



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



GRADO EN INGENIERÍA MECÁNICA

**CONFIGURACIÓN Y DESARROLLO
DE APLICACIÓN SCADA EN
PANTALLA HMI DE OMRON PARA
ACCEDER A DATOS DE AUTÓMATAS
DE SIEMENS Y SCHNEIDER**

AUTOR: Diego Suárez Bethencourt

TUTOR: José Vicente Salcedo Romero de Ávila

CURSO ACADÉMICO: 2016/2017

AGRADECIMIENTOS

A mi tutor José Vicente Salcedo.

A toda mi familia.

A mi gran amigo David Panadero.

A mi querida novia Mari Carmen Teruel.

Sin ellos no hubiera sido tan amena la realización de este trabajo.

RESUMEN

El documento que se expone a continuación surge de la motivación de ampliar los conocimientos adquiridos durante el grado en el sector de la automatización industrial. Para ello se ha automatizado una maqueta de un centro neumático de mecanizado con el uso de dos autómatas de diferentes casas. En primer lugar, se ha empleado el autómata S7-1214 de Siemens, para llevar a cabo la programación de la maqueta, y posteriormente se ha llevado a cabo la automatización con el uso del PLC M241, de la marca Schneider.

Para ir un paso más allá, se ha adentrado en un área de la automatización que resultaba ser desconocido hasta el momento, la programación de una pantalla HMI que, tras comunicarse con un autómata, permita el control sobre un proceso industrial.

Para ello, se ha hecho uso del software NB-Designer, con el que se ha programado una HMI de la marca Omron, de forma que permita su correcta comunicación con cada uno de los autómatas empleados.

De este modo, se ha diseñado un SCADA que, con el uso de una pantalla táctil, y a través de la programación de los autómatas, permite interactuar de forma intuitiva con el proceso de mecanizado, pudiendo elegir entre tres modos de funcionamiento distintos con los que trabajar.

ABSTRACT

The following document arises from the motivation to expand the knowledge about the industrial automation sector, acquired during the degree. For this purpose, a model of a pneumatic machining center has been automated with the use of two automata from different brands. First of all, the Siemens S7-1214 was used to carry out the programming of the model, and then the automation was carried out using the M241 PLC, branded by the Schneider brand.

To go a step further, it has entered an area of automation that was unknown until now, the programming of an HMI screen that, after communicating with an automaton, allows control over an industrial process. For this, NB-Designer software has been used, with which an Omron HMI has been programmed, so that it can be correctly communicated with each of the PLCs.

In this way, a SCADA has been designed that, with the use of a touch screen, and through the programming of the automatons, allows to interact intuitively with the machining process, being able to choose between three different operating modes with which to work.

RESUMEN VLC

El document que s'exposa a continuació sorgeix de la motivació d'ampliar els coneixements adquirits durant el grau en el sector de l'automatització industrial. Per a açò s'ha automatitzat una maqueta d'un centre neumàtic de mecanitzat amb l'ús de dos autòmats de diferents cases. En primer lloc, s'ha emprat el autòmat S7-1214 de Siemens, per a dur a terme la programació de la maqueta, i posteriorment s'ha dut a terme la automatització amb l'ús del PLC M241, de la marca Schneider.

Per anar un pas més enllà, s'ha endisat en un àrea de l'automatització que resultava ser desconegut fins al moment, la programació d'una pantalla HMI que, després de comunica-rse amb un autòmat, permeta el control sobre un procés industrial. Per a açò, s'ha fet ús del software NB-Desogner, amb el qual s'ha programat una HMI de la marca Omron, de manera que permeta la seua correcta comunicació amb cadascun dels autòmats emprats.

D'aquesta manera, s'ha dissenyat un SCADA que, amb l'ús d'una pantalla tàctil, i a través de la programació dels autòmats, permet interactuar de forma intuïtiva amb el procés de mecanitzat, podent triar entre tres maneres de funcionament distintes amb els quals treballar.

ÍNDICE DEL TRABAJO

- A. MEMORIA
- B. PRESUPUESTO
- C. ANEXO. CÓDIGO DE PROGRAMACIÓN



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escuela Técnica Superior de Ingeniería del Diseño

GRADO EN INGENIERÍA MECÁNICA

**CONFIGURACIÓN Y DESARROLLO
DE APLICACIÓN SCADA EN
PANTALLA HMI DE OMRON PARA
ACCEDER A DATOS DE AUTÓMATAS
DE SIEMENS Y SCHNEIDER**

A. MEMORIA

AUTOR: Diego Suárez Bethencourt

TUTOR: José Vicente Salcedo Romero de Ávila

CURSO ACADÉMICO: 2016/2017

ÍNDICE DE LA MEMORIA

1.	OBJETIVO	1
2.	INTRODUCCIÓN	1
2.1	Antecedentes	1
2.2	Motivación	2
2.3	Alcance del trabajo	2
2.4	Tareas a realizar	2
3.	Situación y emplazamiento	3
4.	Estudio del proyecto.....	3
4.1	Definiciones	3
4.2	Elementos empleados.....	5
4.2.1	Autómata Siemens S7-1214 AC/DC/Rly	5
4.2.2	Autómata Schneider Modicon M241	7
4.2.3	Pantalla HMI Omron	10
4.2.4	Maqueta del proceso de producción	11
4.2.5	Sensores.....	11
4.2.6	Actuadores.....	14
4.3	Funcionamiento del proceso	18
4.4	Lenguajes de programación.....	19
4.4.1	Diagrama Funcional Secuencial (SFC).....	19
4.4.2	Lenguaje Ladder (LD)	20
4.5	Variables y direccionamiento de memoria.....	22
4.5.1	Distribución de memoria	22
4.5.2	Direccionamiento de la memoria	23
4.6	Softwares empleados	25
4.6.1	Siemens TIA Portal V13.....	25
4.6.2	SoMachine V4.1	29
4.6.3	NB-Designer	33
5.	Comunicaciones.....	36
5.1	Comunicación entre los autómatas y la maqueta	36
5.2	Comunicación entre los autómatas y el PC	37
5.3	Comunicación entre pantalla HMI y PC	37

5.4	Comunicación entre pantalla HMI y PLC	37
6.	Desarrollo del proyecto	38
6.1.	Funcionamiento de la maqueta del centro neumático	38
6.2	De Diagrama Funcional Secuencial a Lenguaje Ladder	39
6.3	Primeros pasos con Tia Portal V13 y SoMAchine	40
6.4	Programar los distintos Hardware	40
6.4.1	Configuración S7-1200 en Tia Portal V13	40
6.4.2	Configuración Modicon 241 en SoMachine	42
6.5	Introducción de las variables	44
6.5.1	Introducción de variables TIA Portal V13	44
6.5.2	Introducción de variables SoMachine V4.1	46
6.6	Implementación del código e programación	47
6.6.1	Implementación del lenguaje en TIA Portal V13	47
6.6.2	Implementación del lenguaje en SoMachine	50
6.7	Transferir a PLC y comprobación del funcionamiento	52
6.7.1	Transferencia del código en TIA Portal V13	52
6.7.2	Transferencia del código en SoMachine	53
6.8	Configurar comunicación de NB-Designer con los PLC's	54
6.9	Diseño de la pantalla HMI con NB-Designer	55
6.9.1	Ventana de inicio	55
6.9.2	Ventana de funcionamiento automático	62
6.10	Transferir en NB-Designer	66
6.11	Puesta en marcha y verificación	66
7	POSIBLES MEJORAS	67
8	PLIEGO DE CONDICIONES	68
8.1	Elaboración del proceso	68
8.2	Características del material empleado	69
8.3	Legislación	70
9	Conclusión	71
10	Bibliografía	72

ÍNDICE ILUSTRACIONES

Ilustración 1: PLC S7-1214 AC/DC/Rly (Fuente: Elaboración propia)	5
Ilustración 2: PLC Schneider Modicon M241 (Fuente: Elaboración propia)	7
Ilustración 3: Pantalla HMI Omron (Fuente: Elaboración propia).....	10
Ilustración 4:Proceso Mecanizado (Fuente: Elaboración propia)	11
Ilustración 5: Fototransistor cinta transportadora (Fuente: Elaboración propia)	12
Ilustración 6:Sensores final de carrera (Fuente: Elaboración propia).....	13
Ilustración 7: Interruptor Start/Stop (Fuente: Elaboración propia)	13
Ilustración 8: Motor cinta transportadora (Fuente: Elaboración propia)	15
Ilustración 9: Motor mesa giratoria (Fuente: Elaboración propia)	15
Ilustración 10: Cilindro de doble efecto (Fuente: Elaboración propia)	16
Ilustración 11: Cilindro con resorte (Fuente: Elaboración propia)	16
Ilustración 12: Compresor (Fuente: Elaboración propia)	17
Ilustración 13: Diagrama Funcional Secuencial (SFC) (Fuente: Doc Lenguaje de programación PL´s).....	20
Ilustración 14:Código LD (Fuente: Doc Lenguaje para programación de PLC´s)	21
Ilustración 15:Vista del portal (Fuente: Elaboración propia)	25
Ilustración 16:Vista del proyecto (Fuente: Elaboración propia)	26
Ilustración 17:Árbol de funciones (Fuente: Elaboración propia)	26
Ilustración 18:Bloque de programas (Fuente: Elaboración propia)	28
Ilustración 19:Carga de dispositivo (Fuente: Elaboración propia)	29
Ilustración 20:Ventana SoMachine Central (Fuente: Elaboración propia)	30
Ilustración 21:Ventana Logic Builder (Fuente: Elaboración propia)	30
Ilustración 22:Agregar POU (Fuente: Elaboración propia)	31
Ilustración 23:Variables globales (Fuente: Elaboración propia).....	32
Ilustración 24:NB-Designer (Fuente: Elaboración propia).....	33
Ilustración 25:Propiedades de Lámparas de Bit (Fuente: Elaboración propia).....	34
Ilustración 26: Conexión DB37 (Fuente: Elaboración propia)	36
Ilustración 27: Conexión Ethernet (Fuente: Elaboración propia).....	37

Ilustración 28:Selección del dispositivo (Fuente: Elaboración propia)	41
Ilustración 29:Signal Board y Dirección IP (Fuente Elaboración propia).....	42
Ilustración 30: Selección PLC M241 (Fuente: Elaboración propia)	42
Ilustración 31: Insertar cartridge (Fuente: Elaboración propia).....	43
Ilustración 32: Comunicación SoMachine - M241 (Fuente: Elaboración propia)	43
Ilustración 33: Declaración de variables TIA Portal (Fuente: Elaboración propia).....	44
Ilustración 34: Variables de marca TIA Portal (Fuente: Elaboración propia)	46
Ilustración 35: Declaración de variables SoMachine (Fuente: Elaboración propia).....	47
Ilustración 36:Introducir Bloque de Función (Fuente: Elaboración propia).....	48
Ilustración 37: Llamada a Bloque_1. TIA Portal (Fuente: Elaboración propia)	49
Ilustración 38:Área de trabajo Bloque_1 en TIA Portal (Fuente: Elaboración propia) ..	50
Ilustración 39:Llamada a los POU's en SoMachine (Fuente: Elaboración propia)	51
Ilustración 40: Área de trabajo POU_TRANSICIONES, SoMachine (Fuente: Elaboración propia)	51
Ilustración 41: Transferencia del código de programación en TIA Portal (Fuente: Elaboración propia)	52
Ilustración 42: Transferencia del código de programación en SoMachine (Fuente: Elaboración propia)	53
Ilustración 43: Comunicaciones NB-Designer (Fuente: Elaboración propia)	55
Ilustración 44:Fecha/Hora en NB-Designer (Fuente: Elaboración propio)	56
Ilustración 45: Configuración contraseña (1) NB-Designer (Fuente: Elaboración propia)	57
Ilustración 46: Configuración contraseña (2) (Fuente: Elaboración propia).....	58
Ilustración 47:Biblioteca de Textos. Idiomas. (Función: Elaboración propia).....	59
Ilustración 48:Activación Biblioteca de Textos en el botón (Fuente: Elaboración propia)	59
Ilustración 49: Botón de comando / Idioma (Fuente: Elaboración propia)	60
Ilustración 50: Propiedades Botón de Función, NB-Designer (Fuente: Elaboración propia)	61
Ilustración 51:Pantalla principal HMI (Fuente: Elaboración propia).....	62
Ilustración 52: Pantalla de modo automático HMI (Fuente: Elaboración propia)	63
Ilustración 53: Pantalla de modo manual HMI (Fuente: Elaboración propia).....	64
Ilustración 54: Pantalla de modo contador HMI (Fuente: Elaboración propia)	65
Ilustración 55: Transferir programa NB-Designer (Fuente: Elaboración propia)	66

ÍNDICE DE TABLAS

Tabla 1:Datos técnicos generales de familia S7-1214C (Fuente: Manual S7-1200).....	6
Tabla 2:Datos técnico, Características generales de la CPU 1214C (Fuente: Manual S7-1200).....	6
Tabla 3:Memoria PLC M124 (Fuente: Doc EIO0000000143503M241).....	8
Tabla 4:Características Modicon M241 TM241C40R (Fuente: Doc EIO0000000143503M241)	9
Tabla 5:Características NB7W-TW01B (Fuente: Catálogo producto).....	10
Tabla 6:Entradas (Fuente: Elaboración propia).....	14
Tabla 7:Salidas (Fuente: Elaboración propia)	17
Tabla 8:Elementos del lenguaje LD (Fuente: elaboración propia)	21
Tabla 9:Memoria de datos (Fuente: Elaboración propia)	22
Tabla 10:Distribución de memoria (Fuente: Elaboración propia).....	23
Tabla 11:Ejemplo direccionamiento (Fuente: Elaboración propia)	23
Tabla 12: Direccionamiento I/Q/M (Fuente: Elaboración propia)	24

1.OBJETIVO

El siguiente proyecto tiene como objeto comunicar los datos programados en los autómatas Siemens y Schneider con una pantalla HMI de Omron. Para ello se ha realizado la automatización de un centro neumático de mecanizado mediante los diferentes PLC's citados, de forma que se pueda supervisar y controlar el proceso mediante la aplicación SCADA de la pantalla táctil. Para establecer la conexión entre todos los dispositivos en planta, se ha utilizado una red de protocolo Profinet.

2.INTRODUCCIÓN

A continuación, se expondrá como surge la idea de poner en marcha este proyecto, su alcance y los pasos previos a realizar para el posible desarrollo del mismo.

2.1 Antecedentes

La introducción de los autómatas en el ámbito industrial ha significado una importante mejora en la eficiencia de las empresas, la seguridad de sus trabajadores y el control sobre sus producciones. Un proceso de producción automatizado libra a los trabajadores de tener que realizar monótonas, complejas y peligrosas tareas que son imprescindibles para la producción.

En este proyecto se ha querido tener en cuenta, no solo el trabajo con autómatas en un proceso industrial, sino la posibilidad de tener que trabajar con componentes de diferentes marcas para la completa automatización del proceso.

En la sociedad en la que vivimos, cuando sale a la luz un nuevo producto enfocado a cubrir unas necesidades determinadas, inmediatamente comienza la producción por parte de otras marcas de productos similares al original. El hecho de la existencia de productos procedentes de distintas marcas genera un problema de comunicación entre ellos. Cuando se habla de automatización, se pueden distinguir distintos autómatas y pantallas HMI procedentes de distintas familias (Siemens, Omron, Schneider, Rockwell...) y como se ha mencionado anteriormente, la comunicación entre los softwares y elementos de los distintos competidores no son igual de cómodas que si trabajamos con la misma marca.

2.2 Motivación

Como consecuencia del interés generado tras haberse cursado la asignatura Automatización de Máquinas y Procesos, del grado en Ingeniería Mecánica, y la escasez de conocimientos adquiridos por el resto de materias impartidas en dicho grado, se ha querido profundizar más sobre la completa funcionalidad de los autómatas y aplicaciones SCADA en pantallas HMI.

La puesta en práctica de controlar de forma automatizada una maqueta de un centro neumático de mecanizado, ha proporcionado los conocimientos necesarios para poder afrontar un problema similar en la vida laboral lo cual cumple con los objetivos de la selección del proyecto que se expone en este trabajo.

2.3 Alcance del trabajo

El proyecto debe cumplir con los siguientes requisitos para su correcta elaboración:

- Correcto funcionamiento del centro de mecanizado de forma automática al ser conectado con ambos PLC's.
- Establecer comunicación de datos entre los distintos autómatas y la pantalla HMI de Omron.
- Correcto funcionamiento del centro de mecanizado de forma manual con la ayuda de la pantalla táctil.
- Generar un modo de selección en la pantalla HMI en el que se decida el número de piezas a mecanizar de forma automática.
- Generación de una interfaz de la pantalla que permita actuar sobre la misma de una forma guiada e intuitiva.

2.4 Tareas a realizar

Para poder poner en marcha el funcionamiento del proyecto en cuestión ha sido necesaria la realización de las siguientes tareas:

- Elaboración del lenguaje de diagrama de implementación Grafcet.
- Traducción al lenguaje de diagrama de contactos para poder ser interpretado por los softwares de programación.
- Estudio del funcionamiento de los distintos softwares (Siemens TIA Portal V13 y SoMachine V4.1)
- Programación de los autómatas Siemens y Schneider.
- Estudio del funcionamiento del software empleado para la interfaz y la programación de la pantalla HMI.

- Estudio de las comunicaciones entre los distintos elementos del sistema.
- Diseño y programación de la interfaz de la pantalla HMI y comunicación con los autómatas.
- Ajuste y comprobación del correcto funcionamiento del centro de mecanizado y los requisitos planteados.

3.Situación y emplazamiento

El presente proyecto se ha realizado en el Departamento de Ingeniería de Sistemas Automatizados (DISA), edificio 5D de la Universidad Politécnica de Valencia.

4.Estudio del proyecto

Durante el próximo apartado se profundizará en cada uno de los elementos, tanto físicos como de programación, necesarios para la realización del proyecto.

4.1 Definiciones

Las siguientes definiciones, sobre los elementos básicos del entorno de la automatización, han sido sacadas del documento “Diseño y automatización industrial” del autor Pere Ponsa Antoni Granollers y de documentos localizados en la sección de “PolifomaT”, de la Universidad Politécnica de Valencia, aportados por R.P. Fernández.

- **Automatización:** La Real Academia de las Ciencias Físicas y Exactas define la automática como el conjunto de métodos y procedimientos para la sustitución del operario en tareas físicas y mentales previamente programadas. De esta definición original se desprende la definición de la automatización como la aplicación de la automática al control de procesos industriales. [1]
- **Autómata:** también llamado PLC (Programmable Logic Controller), se trata de un equipo electrónico programable diseñado para ser utilizado en un entorno industrial, que utiliza una memoria programable para el almacenamiento interno de instrucciones, con el fin de controlar procesos industriales secuenciales. [2]

- **Implementación:** acción o efecto de poner en funcionamiento, aplicar los métodos y medidas necesarios para llevar algo a cabo. [3]
- **Grafcet:** diagrama de implementación basado en la secuencia de etapas y transiciones, empleado para la programación de los autómatas. Se basa en generar acciones (Salidas) en función de las entradas que se detecten.
- **Etapas:** las etapas determinan el estado en el que se encuentra el programa en un momento dado. Una etapa suele ir acompañada de alguna acción a realizar por el programa, activando una salida al activarse la misma.
- **Transición:** se trata del paso entre una etapa y la posterior. Las transiciones vienen dadas por una o varias condiciones que han de cumplirse para activarse, lo cual permite el paso a la siguiente etapa. Estas condiciones vendrán dadas por la activación de entradas, temporizadores, contadores u otras condiciones que se puedan establecer.
- **Diagrama de contactos:** tiene la misma funcionalidad que el Grafcet, pero esta vez se trata de un diagrama de contactos de entradas y salidas, en lugar de esta formado por etapas y transiciones. Es el lenguaje que se ha empleado para la programación de los autómatas.
- **Sensor:** Se tratan de las entradas más comunes en los autómatas. Cuando un sensor del proceso es activado, se manda una señal digital de valor 1 o 0 al programa, el cual actuará de una forma u otra según las condiciones establecidas en la programación.
- **Actuador:** mecanismo del proceso que se activa cuando la salida del autómata, correspondiente a la dirección de dicho actuador, es activada.
- **Pantalla HMI:** conocido como Interfaz Hombre Maquina, se trata de una pantalla táctil mediante la cual se monitoriza y se controla la supervisión de un proceso industrial. Las señales del proceso se transfieren a la HMI mediante una computadora, un PLC, una RTU o Drive's. Se debe establecer una comunicación lógica entre los distintos dispositivos para su correcto funcionamiento.

4.2 Elementos empleados

A continuación, se exponen los elementos físicos con los que se ha tenido que tratar a lo largo del desarrollo del proyecto.

4.2.1 Autómata Siemens S7-1214 AC/DC/Rly

Uno de los dos autómatas a programar para el control de la maqueta del proceso de mecanizado ha sido el autómata de Siemens de la familia S7-1200, más concretamente el S7-1214 AC/DC/Rly.

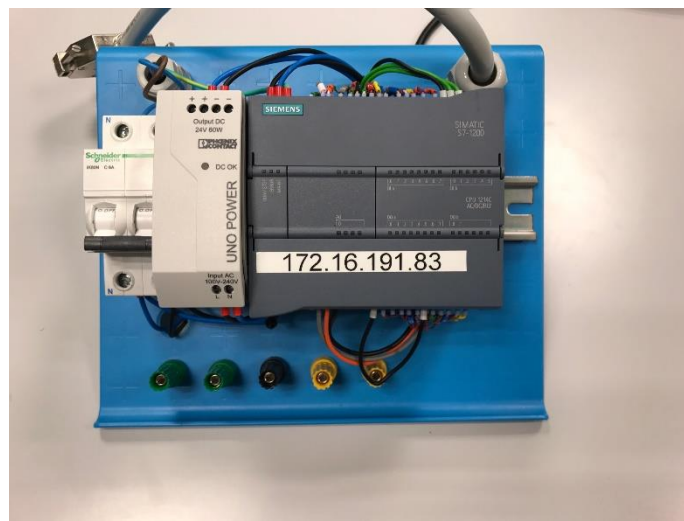


Ilustración 1: PLC S7-1214 AC/DC/Rly (Fuente: Elaboración propia)

Las características de dicho autómata y los datos técnicos de las tablas 1 y 2, presentadas a continuación, se encuentran reflejadas en el “Manual de sistema de SIMATIC S7 Controlador Programable S7-1200”, proporcionado por SIEMENS.

El controlador S7-1200 ofrece la flexibilidad y potencia necesarias para controlar una gran variedad de dispositivos para las distintas necesidades de automatización. Gracias a su diseño compacto, configuración flexible y amplio juego de instrucciones, el S7-1200 es idóneo para controlar una gran variedad de aplicaciones.

La CPU incorpora un microprocesador, una fuente de alimentación integrada, circuitos de entrada y salida, PROFINET integrado, E/S de control de movimiento de alta velocidad y entradas analógicas incorporadas, todo ello en una carcasa compacta, conformando así un potente controlador. Una vez cargado el programa en la CPU, ésta contiene la lógica necesaria para vigilar y controlar los dispositivos de la aplicación. La CPU vigila las entradas y cambia el estado de las salidas según la lógica del programa de usuario, que

puede incluir lógica booleana, instrucciones de contaje y temporización, funciones matemáticas complejas, así como comunicación con otros dispositivos inteligentes.

Referencia del autómata de trabajo: 6ES7 214-1BG40-0XB0

Descripción:

- Memoria de trabajo 75KB.
- Fuente de alimentación 120/240V AC con DI14 x 24V DC SINK/SOURCE, DQ10 x relé y AI2 integradas.
- 6 contadores rápidos y 4 salidas de impulso integradas.
- Signal Board amplia con I/O integradas.
- 3 módulos de comunicación para comunicación serie.
- 8 módulos de señales para ampliación I/O.
- 0,04ms/1000 instrucciones.
- Conexión PROFINET para programación, HMI y comunicación PLC-PLC.

Datos técnicos	CPU 1214C AC/DC/relé	CPU 1214C DC/DC/relé	CPU 1214C DC/DC/DC
Referencia	6ES7 214-1BG40-0XB0	6ES7 214-1HG40-0XB0	6ES7 214-1AG40-0XB0
Dimensiones A x A x P (mm)	110 x 100 x 75		
Peso de envío	475 gramos	435 gramos	415 gramos
Disipación de potencia	14 W	12 W	
Intensidad disponible (SM y bus CM)	1600 mA máx. (5 V DC)		
Intensidad disponible (24 V DC)	400 mA máx. (alimentación de sensores)		
Consumo de corriente de las entradas digitales (24 V DC)	4 mA/entrada utilizada		

Tabla 1: Datos técnicos generales de familia S7-1214C (Fuente: Manual S7-1200)

Datos técnicos	Descripción	
Memoria de usuario (Consulte "Datos técnicos generales" (Página 1149), "Remanencia de la memoria interna de la CPU".)	Trabajo	100 KB
	Carga	4 MB, interna, ampliable hasta tamaño de tarjeta SD
	Remanente	10 KB
E/S digitales integradas	14 entradas/10 salidas	
E/S analógicas integradas	2 entradas	
Tamaño de la memoria imagen de proceso	1024 bytes de entradas (I)/1024 bytes de salidas (Q)	
Área de marcas (M)	8192 bytes	
Memoria temporal (local)	<ul style="list-style-type: none"> • 16 KB para arranque y ciclo (incluyendo los FB y FC asociados) • 6 KB para cada uno de los niveles de prioridad de alarma (incluidos los FB y FC asociados) 	
Ampliación con módulos de señales	8 SM máx.	
Ampliación con SB, CB o BB	1 máx.	
Ampliación con módulos de comunicación	3 CM máx.	

Tabla 2: Datos técnico, Características generales de la CPU 1214C (Fuente: Manual S7-1200)

4.2.2 Autómata Schneider Modicon M241

Citadas las características del autómata de Siemens, pasaremos a nombrar las del otro autómata con el que se automatizó el proceso, el autómata de Schneider Modicon M241 Logic Controller.

Para ello se ha sacado información del documento “EIO0000000143503M241” publicado en 2015, enfocado a los controladores lógicos de Schneider, donde se describen las características del autómata en cuestión.



Ilustración 2: PLC Schneider Modicon M241 (Fuente: Elaboración propia)

Descripción general del autómata

El M241 Logic Controller tiene una amplia variedad de potentes funciones y puede servir para una amplia gama de aplicaciones.

La configuración, programación y puesta en funcionamiento del software se lleva a cabo con el software SoMachine.

Lenguajes de programación

M241 Logic Controller se configura y programa con el software SoMachine, compatible con los siguientes IEC 61131-3 lenguajes de programación:

- IL: Lista de instrucciones
- ST: textos estructurados
- FBD: diagrama de bloques de funciones
- SFC: diagrama funcional secuencial
- LD: diagrama de contactos

Fuentes de alimentación

La fuente de alimentación de M241 Logic Controller es de 24 V CC o de 100-240 V CA.

Reloj en tiempo real

El M241 Logic Controller incluye un sistema de reloj en tiempo real (RTC).

Run/Stop

El M241 Logic Controller se puede utilizar externamente mediante los métodos siguientes:

- Un conmutador Ejecutar/Detener de hardware.
- Una operación de Ejecutar/Detener de una entrada digital dedicada, definida en la configuración del software.
- Un comando del software SoMachine

Memoria

En la siguiente tabla se describen los tipos de memoria del PLC.

Tipo de memoria	Tamaño	Uso a
RAM	64 Mbytes, 8 Mbytes de los cuales están disponibles para la aplicación	ejecutar la aplicación.
Flash	128 MB	guardar el programa y los datos en caso de corte de electricidad.

Tabla 3: Memoria PLC M124 (Fuente: Doc EIO0000000143503M241)

Entradas/Salidas incrustadas

Los tipos de E/S incrustadas que aparecen a continuación están disponibles, en función de la referencia del controlador:

- Entradas normales
- Entradas rápidas asociadas con contadores
- Salidas transistorizadas normales (común negativo y positivo)
- Salidas transistorizadas rápidas (común negativo y positivo) asociadas con generadores de pulsos
- Salidas de relés

Almacenamiento extraíble

Los M241 Logic Controller incorporan un slot para tarjetas SD.

Usos Principales de la tarjeta SD:

- Inicialización del controlador con una aplicación nueva
- Actualización del firmware del controlador
- Aplicación de archivos de configuración de Post al controlador
- Aplicación de fórmulas
- Recepción de archivos de registro de datos

Funciones de comunicación incorporadas

Hay disponibles los siguientes tipos de puertos de comunicación según la referencia del controlador:

- CANopen maestro
- Ethernet
- USB mini-B
- Línea serie 1
- Línea serie 2

Características TM241C40R

En la siguiente se contemplan las características del modelo de referencia TM241C40R.

Referencia	Entradas digitales	Salidas digitales	Puertos de comunicación	Tipo de terminal	Fuente de alimentación
TM241C40R	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	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 4: Características Modicon M241 TM241C40R (Fuente: Doc EIO0000000143503M241)

4.2.3 Pantalla HMI Omron

La pantalla táctil con la que se ha dispuesto para realizar el proyecto es una pantalla de la serie NB de Omron, más concretamente le modelo NB7W-TW01B, cuyas características se exponen en la *tabla 5*. Permite una sencilla comunicación con los autómatas programables de Omron, aparte de existir la posibilidad de poder trabajar con autómatas de otras marcas, como se ha demostrado con este proyecto.



Ilustración 3: Pantalla HMI Omron (Fuente: Elaboración propia)

La comunicación entre la pantalla HMI y los diferentes autómatas se ha llevado a cabo a través de los puertos Ethernet de los dispositivos, y el diseño de la interfaz de la pantalla a través del software NB-Designer, en el que se profundizará más adelante.

Serie del Fabricante	NB
Tipo de Display	TFT LCD
Tamaño del Display	7 pulg.
Resolución del Display	800 x 480pixels
Color del Display	Color
Número de Puertos	3
Tipo de Puerto	Ethernet, Host USB, RS-232C
Memoria Integrada	128 MB
Retroiluminación	Sí
Tensión de Alimentación	20,4 → 27,6 V dc
Anchura	148mm
Profundidad	46mm
Temperatura de Funcionamiento Mínima	0°C
Dimensiones	202 x 148 x 46 mm
Longitud	202mm
Índice de Protección IP	IP65
Temperatura de Funcionamiento Máxima	+50°C

Tabla 5:Características NB7W-TW01B (Fuente: Catálogo producto)

4.2.4 Maqueta del proceso de producción

La maqueta empleada para realizar el proyecto (*Ilustración 4*) simula un centro de mecanizado que se ayuda de un compresor de aire para realizar gran parte de sus tareas. Con esta maqueta se logra poner en práctica los conocimientos adquiridos en el campo de la automatización de procesos, sin necesidad de tener que interactuar con un proceso a escala real.

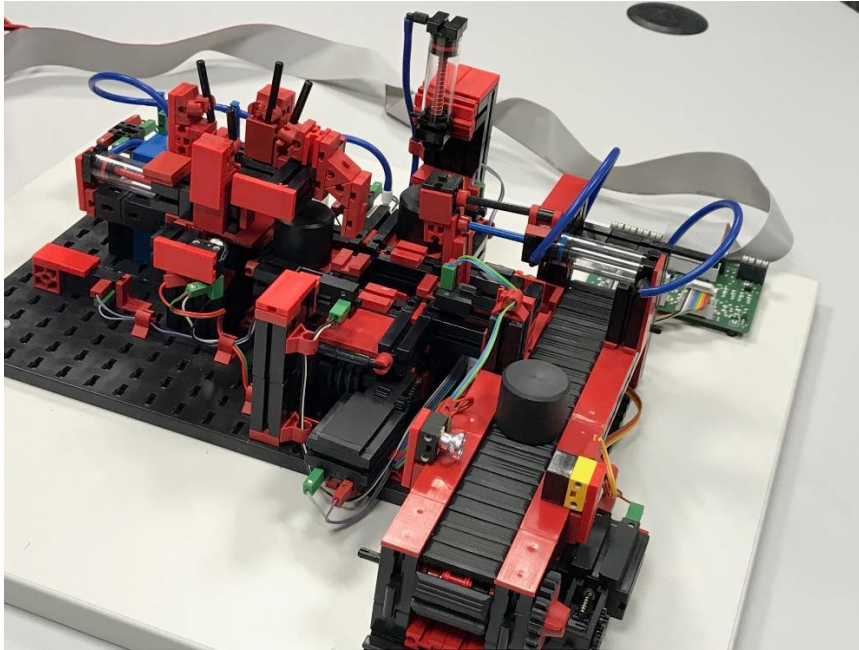


Ilustración 4:Proceso Mecanizado (Fuente: Elaboración propia)

La maqueta consta de los siguientes elementos que hacen posible el mecanizado de las piezas que intervienen en el proceso:

- Un sistema de impulsión con resorte
- Mesa giratoria
- Prensa
- Sistema de impulsión con cilindro de doble efecto
- Cinta transportadora

4.2.5 Sensores

La maqueta de trabajo contiene una serie de sensores que permiten que se le comunique al autómatas, en forma de entradas, el estado en el que se encuentra el proceso en cada momento. De esta forma, el autómatas responderá a estas señales según la programación previamente configurada, activando o desactivando los actuadores necesarios para el mecanizado de las piezas.

A continuación, se muestran ilustraciones de los diferentes tipos de sensores que se encuentran a lo largo de la cadena de mecanizado:

- 2 fototransistores: uno de ellos se encuentra en el comienzo del recorrido, para identificar la presencia de una nueva pieza a mecanizar. El otro fototransistor tiene como función informar de que la pieza ha llegado al final del recorrido, por lo que se encuentra en el último tramo de la cinta transportadora.

El funcionamiento de estos sensores se basa en la emisión de un haz de luz y un receptor que lo detecte. Cuando el receptor detecta el haz de luz, se envía un 1 al PLC, mientras que, cuando hay un objeto que impide el paso del haz de luz, el sensor mandará un 0 al autómata. De esta forma es como estos sensores detectan la presencia de algún elemento en su posición.

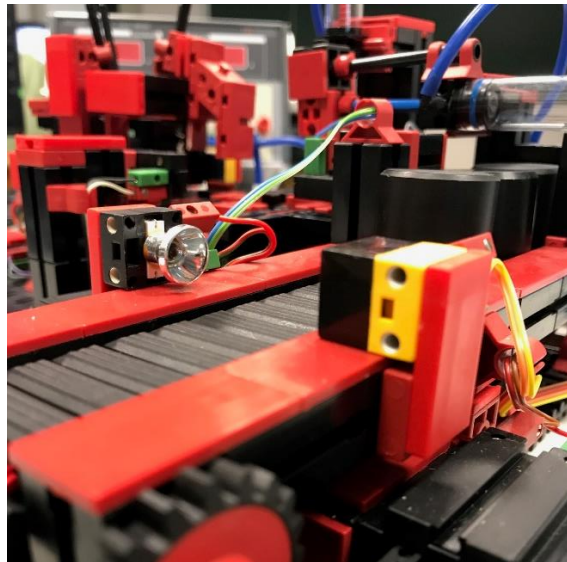


Ilustración 5: Fototransistor cinta transportadora (Fuente: Elaboración propia)

- 3 finales de carrera: dos de ellos se emplean para detectar la presencia de la pieza en los siguientes puntos del recorrido, uno bajo el cilindro de la presa y al llegar al puesto de intercambio entre la mesa giratoria y la cinta transportadora. El otro sensor se emplea para detectar cuando la mesa giratoria se encuentra colocada de forma alineada con las distintas postas del proceso. Gracias a este sensor se informa de cuando la mesa esta lista para recibir una pieza, para que se active la función de prensado o para intercambio de la pieza entre la mesa y la cinta transportadora.

Estos sensores son simples pulsadores que son activados cuando la pieza ejerce presión sobre ellos. Mientras se encuentren activos, estos sensores se ponen a 1. A diferencia de los fototransistores, estos sensores se ponen a 1 con la presencia de la pieza, mientras que los transistores trabajan de forma inversa.

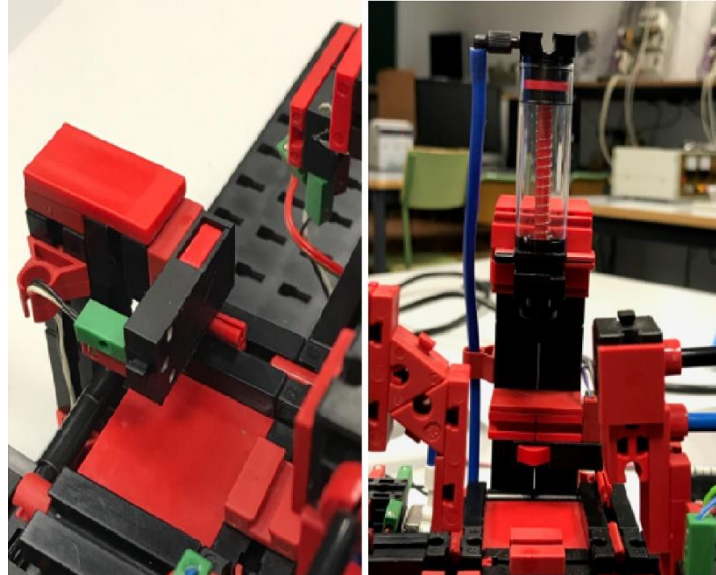


Ilustración 6: Sensores final de carrera (Fuente: Elaboración propia)

- 1 interruptor Start/Stop: es un sensor del mismo tipo que los empleados para los finales de carrera, la única diferencia es que, con la ayuda de una pequeña placa, se hará presión de forma permanente en el sensor para mantenerlo en 1 cuando se quiera poner en funcionamiento el proceso de mecanizado.

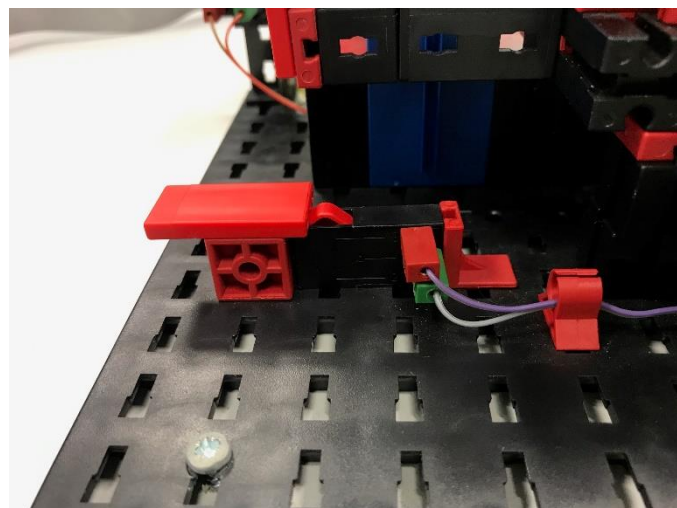


Ilustración 7: Interruptor Start/Stop (Fuente: Elaboración propia)

En la siguiente *tabla 6* se muestran cada una de las entradas, activadas por los sensores, que intervienen en el proceso, además de las direcciones correspondientes para la correcta comunicación con los distintos autómatas.

Entradas	Descripción	Dirección S7-1200	Dirección M241
I1	Fototransistor cinta transportadora	%I0.2	I%X0.2
I2	Final carrera intercambio mesa giratoria	%I0.3	I%X0.3
I3	Final carrera presa	%I0.5	I%X0.5
I4	Final carrera mesa giratoria	%I0.6	I%X0.6
I5	Fototransistor entrada de pieza	%I0.7	I%X0.7
I6	Interruptor Start/Stop	%I1.0	I%X1.0

Tabla 6:Entradas (Fuente: Elaboración propia)

4.2.6 Actuadores

Las señales que recibe el autómata desde los sensores en forma de entradas, generan una respuesta que se verá reflejada en la maqueta a través de los actuadores. Los actuadores intervendrán sobre la pieza a mecanizar de forma automática, asegurando su transporte de principio a fin del proceso y la ejecución del prensado a la que se someten las piezas durante el proceso.

Se pueden identificar actuadores con distintas funciones a lo largo del mecanizado, algunos encargados del empuje de la pieza, otro encargado del prensado y otros para el transporte de las mismas. Se muestran a continuación los distintos actuadores que intervienen en el proceso:

- Motor de la cinta transportadora: al ser activado mediante el código de programación, activa el funcionamiento de la cinta transportadora para el desplazamiento lineal de la pieza.

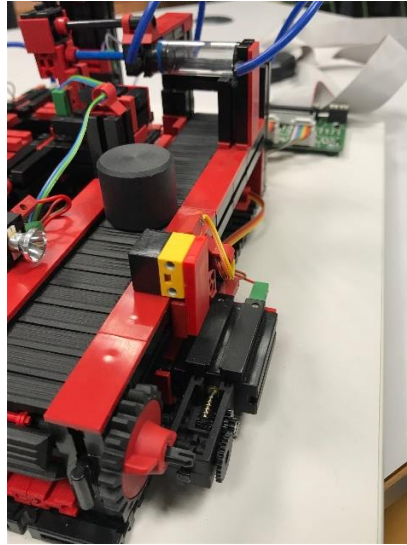


Ilustración 8: Motor cinta transportadora (Fuente: Elaboración propia)

- Motor de la mesa giratoria: al recibir la orden desde el autómata y activarse, provoca el movimiento circular de la mesa giratoria.



Ilustración 9: Motor mesa giratoria (Fuente: Elaboración propia)

- Cilindro de doble efecto: con la ayuda del compresor, el cilindro de doble efecto se emplea para el intercambio de la pieza entre la mesa giratoria y la cinta transportadora. Según se active una de las entradas de aire del cilindro, éste se desplazará en un sentido u otro.

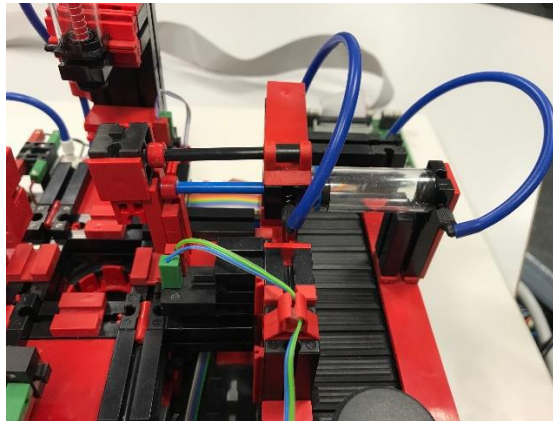


Ilustración 10: Cilindro de doble efecto (Fuente: Elaboración propia)

- Cilindro con resorte: se pueden encontrar dos en todo el proceso, uno para efectuar el primer empuje de la pieza hacia la mesa giratoria, y el otro para accionar la prensa cuando la pieza esté en la posición adecuada. Este último se puede contemplar en la ilustración 6 del apartado de sensores.

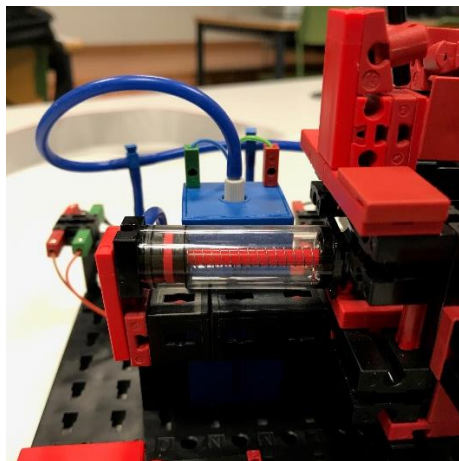


Ilustración 11: Cilindro con resorte (Fuente: Elaboración propia)

- Compresor: el compresor se emplea para activar cada uno de los cilindros que componen el proceso. Tanto el de doble efecto como los que funcionan con resorte.

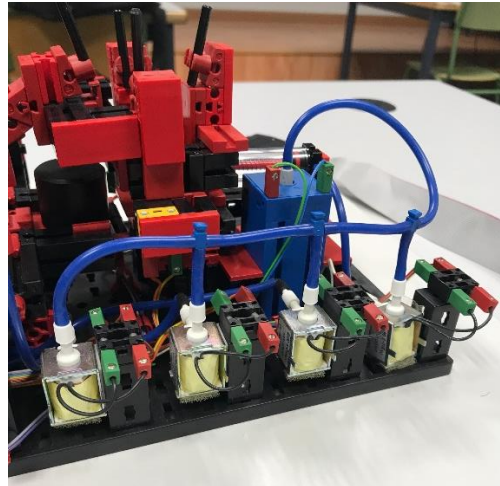


Ilustración 12: Compresor (Fuente: Elaboración propia)

La respuesta generada por los autómatas ante las entradas que recibe, son transmitidas a los actuadores a través de las llamadas *salidas*, las cuales se muestran a continuación junto a las direcciones de comunicación correspondientes a cada autómata:

Salida	Descripción	Dirección S7-1200	Dirección M241
Q1	Motor cinta transportadora	%Q0.2	%QX0.6
Q2	Motor mesa giratoria	%Q0.4	%QX1.0
Q3	Salida Cilindro intercambio de pieza	%Q0.5	%QX1.1
Q4	Entrada cilindro intercambio de pieza	%Q0.6	%QX1.2
Q5	Cilindro de prensa	%Q0.7	%QX1.3
Q6	Cilindro de la entrada de piezas	%Q1.0	%QX1.4
Q7	Compresor	%Q1.1	%QX1.5

Tabla 7: Salidas (Fuente: Elaboración propia)

4.3 Funcionamiento del proceso

La maqueta tiene un funcionamiento lineal encargado de transportar de un punto a otro las piezas que intervengan y realizar un prensado sobre las mismas en uno de los apartados del recorrido.

Con la ayuda de la pantalla HMI se dará la orden al autómata de poner en marcha el centro de mecanizado, lo cual, estando en el modo automático, conllevará a la siguiente sucesión de acciones:

- 1 Cuando una pieza es detectada por el sensor I5 (*Fototransistor entrada de piezas*) se activarán las salidas Q7 y Q6, encargadas de activar el compresor y el cilindro de entrada, respectivamente. De esta forma la pieza será empujada a una de las plataformas que forman parte de la mesa giratoria.
- 2 Cuando ésto sucede, y pasado un tiempo de unos segundos, controlado por un temporizador, se activará la mesa giratoria (Q2) hasta que el sensor I3 (*Final de carrera presa*) sea activado. Este sensor nos permite saber que tenemos una pieza debajo de la prensa.
- 3 Una vez que se activa I3, los autómatas están programados para activar las salidas Q7 y Q5, de esta forma se efectúa la acción de prensado sobre las piezas.
- 4 Con la ayuda de otro temporizador se limita a que el proceso dure unos segundos.
- 5 Se vuelve a activar la mesa giratoria, esta vez hasta que el sensor I2 (*Final de carrera intercambio mesa giratoria*) sea el que se ponga a 1. La pieza se encontrará en el lado opuesto de donde ha comenzado.
- 6 El autómata dará la orden de activar los actuadores Q7 y Q4, con el fin de colocar la pieza sobre la cinta transportador. Una vez pasado un tiempo determinado se activarán Q7 y Q3, para volver a colocar el cilindro de doble efecto en su lugar inicial.
- 7 Cuando este tiempo transcurre se activa Q1 y como consecuente el motor que mueve la cinta transportadora para desplazar la pieza.
- 8 Al producirse un flanco de subida en I1 (*Fototransistor cinta transportadora*) la cinta se detendrá y la pieza, ya mecanizada, estará lista para ser recogida.
- 9 El funcionamiento de la mesa giratoria vendrá controlado por el sensor I4 (*Final de carrera mesa giratoria*) y un temporizador que la mantenga parada durante unos 3 segundos. De esta forma se permite trabajar con tres piezas dentro del proceso de forma simultánea, evitando solapamiento de información que pueda ser enviada al autómata.

4.4 Lenguajes de programación

La única forma de lograr que un PLC controle el correcto funcionamiento de un proceso automatizado determinado, es mediante la implementación de un código lógico que determine la forma de actuar del autómata. A este código se le llama lenguaje de programación.

Podemos distinguir varios tipos de lenguaje de programación que, en función del modelo del autómata, podrán ser o no empleados para la programación del mismo.

A continuación, se exponen algunas de las distintas formas de lenguaje existentes:

- Lenguaje Ladder (LD)
- Diagrama funcional secuencial (SFC)
- Lenguaje booleano lista de instrucciones (IL)
- Diagrama de funciones (FBD)
- Lenguaje de texto estructurado (ST)

Como consecuencia de que para la realización de este proyecto se han empleado los lenguajes LD y SFC, serán en los que se profundice a continuación:

4.4.1 Diagrama Funcional Secuencial (SFC)

Se trata de un lenguaje gráfico (Grafcet) que se representa con un diagrama de secuencias de control. Se considera un lenguaje sumamente útil para operaciones de control secuencial en el que se pasa de un punto al otro del programa una vez que se satisfaga una condición dada.

En los Grafcet se pueden distinguir tres elementos principales:

- Etapas
- Transiciones
- Acciones

Teniendo en cuenta la *Ilustración 13*, en la que se representa un ejemplo de un Grafcet sencillo, se contemplan las etapas, representadas en forma de cuadrados enumerados; entre cada una de las etapas se encuentra un trazado horizontal, las transiciones; para finalizar, en forma rectangular a la derecha de las etapas, se ubican las acciones que se realizarán cuando una etapa está activa.

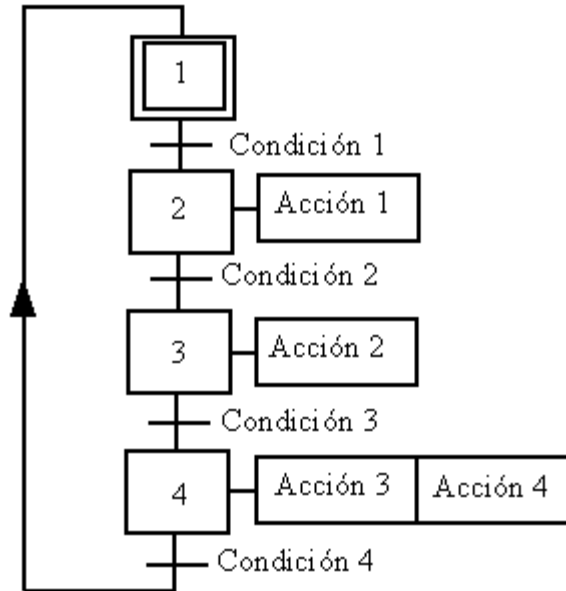


Ilustración 13: Diagrama Funcional Secuencial (SFC) (Fuente: Doc Lenguaje de programación PL's)

Para que una transición se active deben satisfacerse unas condiciones, que son programadas en una pantalla ajena a la que se contempla. Esta programación se ha realizado con el lenguaje Ladder.

Una vez que se active la etapa deja el paso al “flujo” permitiendo la activación de la siguiente etapa. Cuando una etapa está activa, en caso de tener asociada una acción, ésta se llevará a cabo, hasta que se cumplan las condiciones de la siguiente transición.

De esta forma se completa el ciclo, según las condiciones que se vayan cumpliendo a lo largo del proceso. Con la posibilidad de ser un ciclo cíclico o cerrado.

4.4.2 Lenguaje Ladder (LD)

También conocido como lenguaje de diagrama de contactos o de escalera, está basado en el esquema del funcionamiento eléctrico básico de los circuitos de conmutación, simulando el pase de corriente a través del mismo, según estén “abiertas” o “cerradas” las condiciones que lo componen.

Este lenguaje se encuentra normalizado según normas NEMA, por lo que cualquier PLC será capaz de interpretarlo y trabajar con él. A continuación, se mostrarán los elementos esenciales que componen el lenguaje LD, junto con su respectivo funcionamiento:

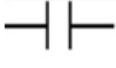
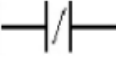
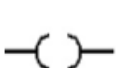
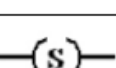
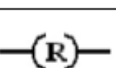
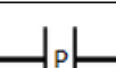
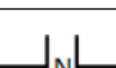
	Contacto	Será activada, permitiendo el pase de "corriente" cuando el elemento que lo compone se pone a 1.
	Contacto negado	Será activada, permitiendo el pase de "corriente" cuando el elemento que lo compone se pone a 0.
	Bobina	Se activa poniéndose a 1 cuando la combinación de requisitos previos, que suelen ser contactos, temporizadores o contadores, da un uno lógico. Suele representar elementos de salida.
	Bobina Set	Se comporta igual que la bobina normal, solo que requiere de la activación de una bobina Reset con la misma dirección para ponerse a 0.
	Bobina Reset	Al ponerse a 1 desactiva la bobina Set que tenga la misma dirección.
	Flanco de subida	Se pone a 1 cuando detecta el pase de 0 a 1 del bit que lo controla.
	Flanco de bajada	Se pone a 1 cuando detecta el pase de 1 a 0 del bit que lo controla.

Tabla 8: Elementos del lenguaje LD (Fuente: elaboración propia)

Una vez aclarado el comportamiento de los diferentes elementos que componen el diagrama de contactos, se muestra una parte de un código LD (Ilustración 14) para comprender como se comunican unos elementos con otros.

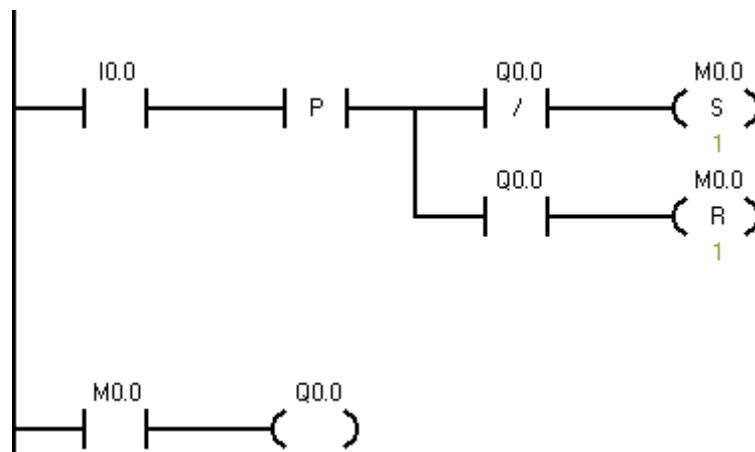


Ilustración 14: Código LD (Fuente: Doc Lenguaje para programación de PLC's)

El orden de lectura de este lenguaje se efectúa de izquierda a derecha y de arriba hacia abajo. Deben cumplirse las condiciones previas a cada bobina, para que ésta se ponga a 1, por lo que, para que se active la bobina Set de la variable M0.0, previamente ha de estar a 1 el contacto I0.0, el flanco de subida y el contacto negado de Q0.0.

En caso de que esté a 1 del contacto no negado de Q0.0, será la bobina Reset la que se ponga a 1 y de esta forma desactivaría la bobina Set de M0.0.

4.5 Variables y direccionamiento de memoria

Como se ha podido contemplar en el apartado anterior, cada uno de los elementos que componen el lenguaje de programación está acompañado de una variable a la que hace referencia. En la *ilustración 14* se puede ver que el primer contacto de todos tiene declarada la variable I0.0, al igual que el resto de elementos. El flanco de subida también está asociado a una variable, a pesar de que no se visualice en este ejemplo.

En este apartado se verán los distintos tipos de variables con los que se ha trabajado en el proyecto, cómo se distribuye la memoria del sistema y el direccionamiento de la misma para poder interpretar las variables empleadas.

4.5.1 Distribución de memoria

Se pueden distinguir tres apartados dentro de cómo se distribuye la memoria del sistema:

- Memoria de programa: en la que se almacenan las operaciones de esquema de contactos, que serán ejecutadas por el autómata para el correcto funcionamiento del centro de mecanizado.
- Memoria de parámetros: permite el almacenamiento de parámetros configurables. En este proyecto se ha empleado alguno de estos parámetros, como son las direcciones de módulo, de las que se hablará posteriormente.
- Memoria de datos: es el área a la que accede el programa de usuario para la lectura de cada una de las variables de datos y objetos.

Memoria de datos	
Área de datos	Objetos
Memoria de variables (V)	Temporizadores (T)
Imágenes de proceso de entradas (I)	Contadores (Z)
Imágenes de proceso de salida (Q)	Entradas analógicas (AI)
Marcas internas (M)	Salidas analógicas (AQ)
Marcas especiales (SM)	Acumuladores (ACU)

Tabla 9: Memoria de datos (Fuente: Elaboración propia)

En la siguiente tabla se muestra la longitud de los elementos que componen la distribución de la memoria:





Nombre	Longitud
 BIT	L = 1 BIT
7 6 5 4 3 2 1 0  BYTE	L = 8 BITS
7 6 5 4 3 2 1 0 7 6 5 4 3 2 1 0  PALABRA	L = 16 BITS
7 6 5 4 3 2 1 0 7 6 5 4 3 2 1 0 7 6 5 4 3 2 1 0 7 6 5 4 3 2 1 0  DOBLE PALABRA	L = 32 BITS

Tabla 10: Distribución de memoria (Fuente: Elaboración propia)

Se debe tener en cuenta que un BIT trabajará con dos tipos de estados, “1” o “0”.

4.5.2 Direccionamiento de la memoria

Las direcciones que se le deben dar a una variable varían según el autómata con el que se trabaje. La información sobre el direccionamiento de las variables expuesta a continuación está referida a la empleada para el autómata de S7-1200.

Para este apartado se comenzará con un ejemplo a modo de aclaración. Dada la variable I0.6 con direccionamiento %I0.6, en la *tabla 11* se muestra la función de cada uno de los componentes de este direccionamiento:

I	Identificador de área (I=entrada)
0	Dirección del byte (Byte 0 será el primer byte)
.	Punto decimal que separa la dirección del byte del número de bit
6	Bit del byte o número de bit: bit 7 de 8 (0 a7)

Tabla 11: Ejemplo direccionamiento (Fuente: Elaboración propia)

Como se ha visto anteriormente, es posible el acceso a las distintas áreas de memoria de la CPU. A continuación, se citarán las más relevantes de este proyecto y la estructura de su direccionamiento:

Direccionamiento	Bit	Byte	Ej.
Imagen entradas	I [módulo].[dir. del bit]	I [tamaño].[dir. del byte inicial]	I0.1 IB4
Imagen salidas	Q [módulo].[dir. del bit]	Q [tamaño].[dir. del byte inicial]	Q1.1 QB5
Área de marcas	M[módulo].[dir. del bit]	M [tamaño].[dir. del byte inicial]	M26.7 MD20

Tabla 12: Direccionamiento I/Q/M (Fuente: Elaboración propia)

Cuando se trabaja con las direcciones del autómata M241, es necesario añadir una X después de cada uno de los identificadores de área, para la correcta lectura por parte del autómata. De esta forma, un identificador de área de entrada sería %IX0.2.

El direccionamiento de las marcas especiales también varía según el autómata empleado, pero la funcionalidad que aportan se puede encontrar de una manera u otra en todos los tipos de autómatas. Los direccionamientos expuestos a continuación hacen referencia de nuevo al autómata S7-1200, pudiéndose emplear estas marcas, con otro direccionamiento, en el resto de los autómatas:

- %M100.0 (Arranque): este bit es activado únicamente durante el primer ciclo. Suele ser empleado para el arranque del programa.
- %M100.1 (DiagStatusUpdate): diagrama de diagnóstico modificado.
- %M100.2 (AlwaysTRUE): este bit siempre permanece activo.
- %M100.3 (AlwaysFALSE): este bit siempre permanece negado.
- %M0.0 - %M0.7: ofrecen un reloj que estará activo e inactivo por un período de tiempo determinado, el cual variará según el número de bit especificado.

En este proyecto solo ha sido necesario el uso de la marca %M100.0, para declarar unas condiciones tras el arranque del programa. En el caso del autómata M241 esta marca viene dada por la dirección *IsFirstMastCycle*.

Las marcas especiales deben ser activadas en los softwares correspondientes para poder ser empleadas.

4.6 Softwares empleados

Durante este apartado se profundizará sobre la estructura de los distintos softwares que han sido necesarios para la elaboración de este proyecto. Por un lado, los dos softwares enfocados a la programación de los distintos autómatas, y por otro lado el empleado para el diseño y programación de la pantalla HMI.

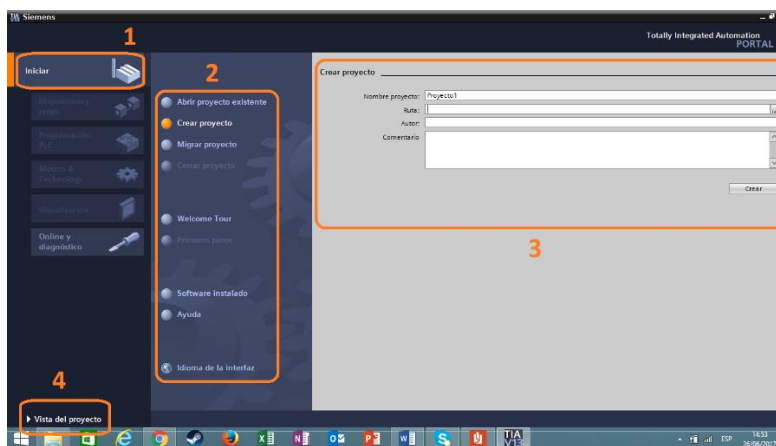
4.6.1 Siemens TIA Portal V13

Anteriores gamas de los PLC de Siemens, tal como el S300, tienen la opción de ser programados a través de dos programas, el Step7 V5.5 y el Tia Portal V13. Según han salido nuevos modelos de autómatas, se han ido enfocando únicamente al programa Tia Portal V13, dadas las mejores prestaciones que ofrece en comparativa con el otro programa.

Por ello, el software que se ha empleado para la programación del PLC de la gama S7-1200, ha sido el TIA Portal V13. Esta plataforma ha pretendido unificar todos los softwares que emplea Siemens para sus diferentes dispositivos, logrando así que la programación de alguna aplicación determinada sea lo más compacta posible.

Existen dos vistas principales del programa que pueden ser seleccionadas por el usuario, según sus preferencias de trabajo. Es posible cambiar entre una vista y otro con un solo click.

Vista del portal

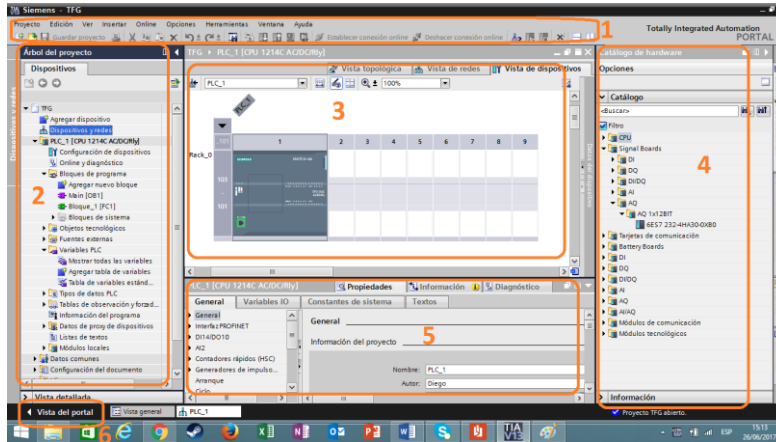


- 1) Portales para las diferentes tareas
- 2) Tareas del portal seleccionado
- 3) Panel de selección para la acción seleccionada
- 4) Cambia a la vista de proyecto

Ilustración 15: Vista del portal (Fuente: Elaboración propia)

Se recomienda el uso de esta vista, únicamente para la selección del autómata con el que se va a trabajar, al igual que su referencia. Una vez seleccionado, pasar a la vista de proyecto.

Vista del proyecto



- 1) Menús y barra de herramientas
- 2) Árbol de proyecto
- 3) Área de trabajo
- 4) Task Cards
- 5) Ventana de inspección
- 6) Cambia a la vista del portal

Ilustración 16: Vista del proyecto (Fuente: Elaboración propia)

Dentro del árbol de proyectos de la vista proyectos se selecciona qué parte del programa es la que se quiere mostrar en el área de trabajo, para así poder interactuar con la misma. En la Ilustración 17 se muestra más claramente en qué partes se descompone el árbol de trabajo, y posteriormente se nombran cuáles son los apartados frecuentados durante este proyecto.

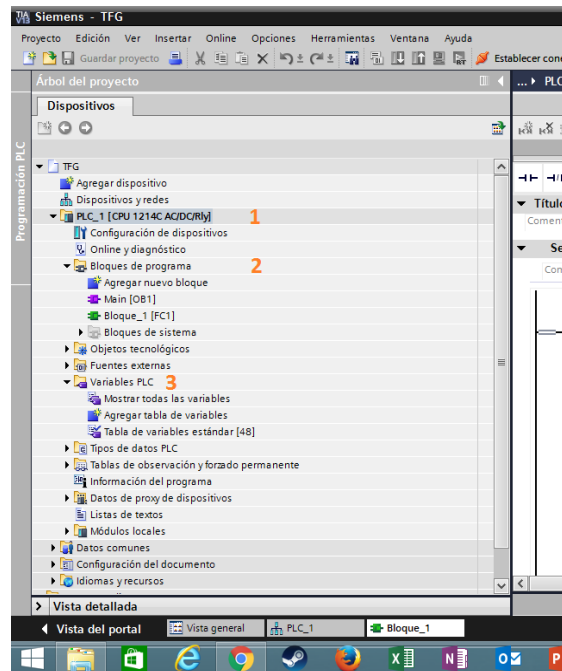


Ilustración 17: Árbol de funciones (Fuente: Elaboración propia)

1) En el apartado *PLC_1 -> Configuración de dispositivos* es donde se puede efectuar cualquier cambio relacionado con la configuración del PLC de trabajo. En este apartado también permite realizar la selección del autómata, que se ha realizado en la vista del portal.

A demás, también nos permite seleccionar la Signal Board con la que cuenta el autómata y poder agregar elementos externos al autómata, en caso de que éste conste de ellos.

2) El apartado *Bloque de programas* es el que permite al usuario insertar el código de programación que se le quiere transferir al autómata.

Este programa, a pesar de tener varios bloques de programas, únicamente se centra en leer el bloque de programa *Main [OB1]*. Por lo que es necesario realizar una llamada desde el bloque *Main* al resto de bloques, si se quiere que el programa los lea.

Se verá más adelante que toda la programación del autómata se ha realizado en la función *Bloque_1 [FC1]*, por lo que ha sido necesario hacer una llamada del mismo desde el *Main [OB1]*.

3) Por último, se tiene el apartado *Variables PLC*, en el que se han de introducir todas aquellas variables que sean utilizadas en el lenguaje empleado, acompañadas de sus direcciones correspondientes. En caso de que no se declare una variable que aparezca en el lenguaje de programación la compilación del proyecto no será permitida por parte del programa.

Por otro lado, es oportuno aclarar cómo funciona TIA Portal V13 a la hora de introducir el lenguaje de programación, ya que trabaja de una forma muy similar a la de SoMachine, lo cual sirve de ayuda para la comprensión de ambos softwares. Se utilizará la Ilustración 18 como ejemplo para ello.

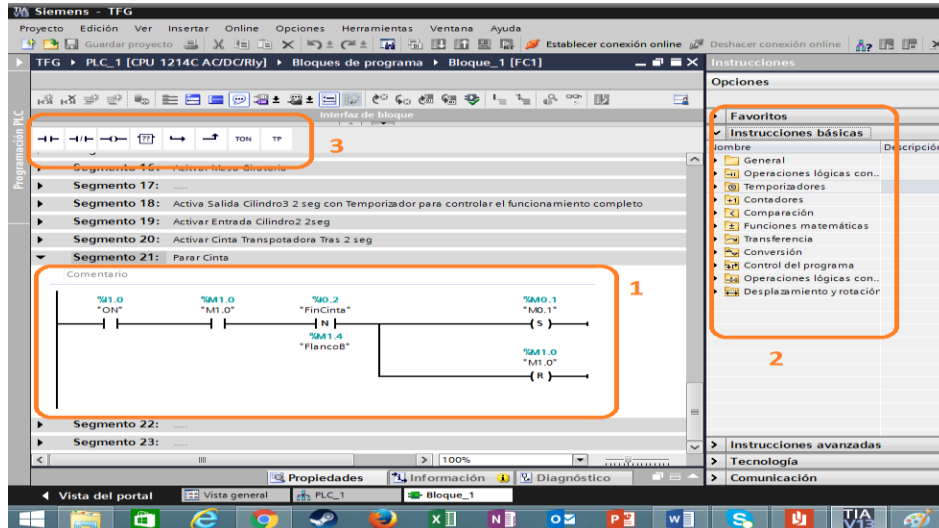


Ilustración 18: Bloque de programas (Fuente: Elaboración propia)

1) En este apartado del programa es donde se debe realizar la implementación del lenguaje de programación que se ha diseñado previamente.

2) Dentro de *Instrucciones básicas* se encuentran cada uno de los elementos que hacen falta para la implementación del lenguaje (Contactos, bobinas, temporizadores, contadores, comparadores...).

3) Es recomendado utilizar la barra *Favoritos* para ubicar en ella todos aquellos elementos que se utilicen con alta frecuencia. De esta forma se agiliza el trabajo de programación.

Para finalizar, en la barra de herramientas, se encuentran las funciones de *Compilar* y *Cargar en dispositivo*, que tienen como objetivo transferir correctamente el programa al PLC.

Una vez que se ha terminado de introducir todo el código de programación, se debe compilar el programa para identificar que no existe ningún error en la programación. Compilado el programa se selecciona *Cargar en dispositivo*, lo que abre una ventana representada en la *Ilustración 19*.

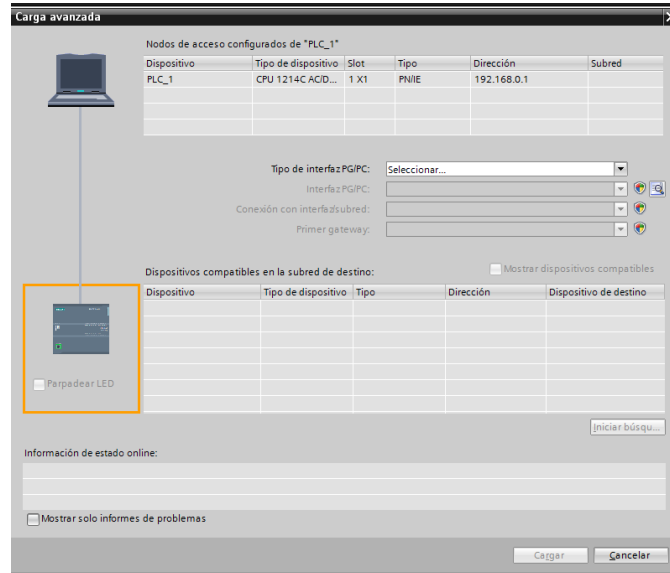


Ilustración 19: Carga de dispositivo (Fuente: Elaboración propia)

Previo a transferir el programa al PLC es necesario establecer una comunicación entre el PC y el autómata. En este caso se ha hecho mediante los puertos Ethernet de cada uno de los dispositivos.

Para la correcta comunicación entre ambos, es necesario ir al apartado *PLC_1 -> Configuración de dispositivos* y una vez ahí, dentro de la *Ventana de inspección -> Interfaz PROFINET -> Protocolo IP* declarar la dirección IP correcta del autómata.

Una vez configurado, volviendo a la *Ilustración 19*, se inicia la búsqueda del Dispositivo, se selecciona, y se transfiere el programa al PLC.

4.6.2 SoMachine V4.1

Para la programación de los autómatas diseñados por Schneider se emplea el software SoMachine, en este caso se ha trabajado con la versión SoMachine V4.1, pero existen otras versiones más básicas que se frecuentan para la programación de muchos autómatas, como es SoMachine Basic.

Tiene una estructura similar a TIA Portal V13, lo cual facilita el estudio de su funcionamiento una vez que se tiene los conocimientos sobre este otro software.

Una de las ventajas que tiene SoMachine frente a TIA Portal es que permite el uso del lenguaje de programación por diagrama funcional secuencial (SFC), por lo que, en caso

de haber trabajado únicamente con este software, no hubiera sido necesario la conversión de lenguaje SFC al lenguaje por diagrama de contactos LD.

En las ilustraciones mostradas a continuación se puede ver las dos ventanas principales de las que consta SoMachine: *SoMachin Central*, donde se hace la selección del dispositivo con el que se va a trabajar, y la ventana *Logic Builder*, estructurada de forma similar que la *Vista de proyecto* del TIA Portal V13.

SoMachine Central

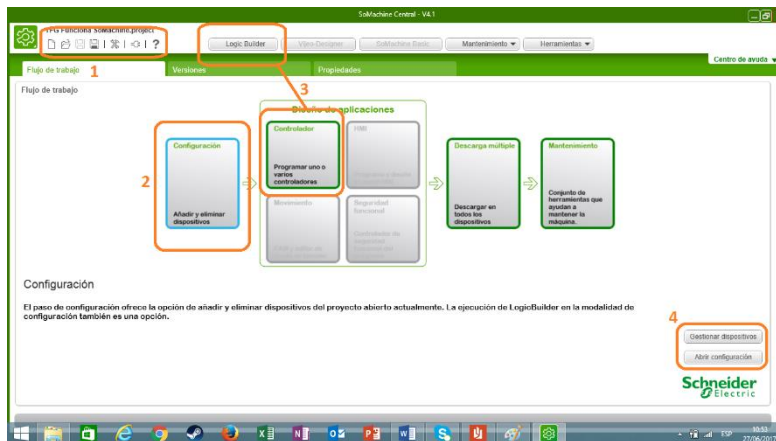


Ilustración 20: Ventana SoMachine Central (Fuente: Elaboración propia)

- 1) Barra de herramientas
- 2) Selección de los dispositivos a programar
- 3) Pasar al modo Logic Builder (Interfaz de programación)
- 4) Tarea según el flujo de trabajos seleccionado

Logic Builder

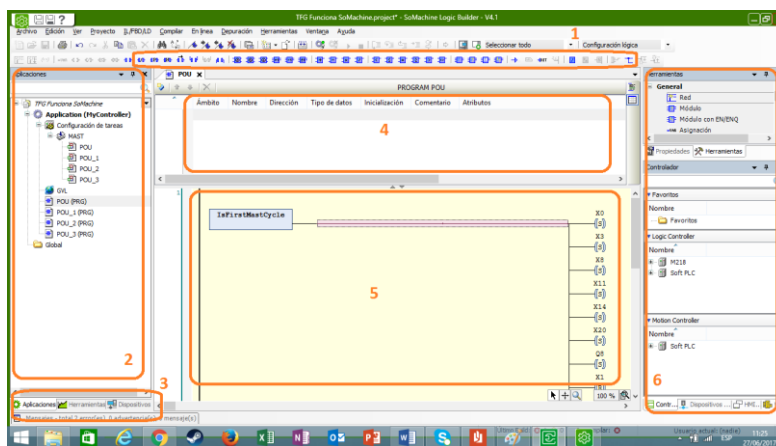


Ilustración 21: Ventana Logic Builder (Fuente: Elaboración propia)

- 1) Barra de herramientas de programación.
- 2) Navegador del proyecto
- 3) Pestañas de navegación
- 4) Ventana declaración de variables
- 5) Área de trabajo
- 6) Herramientas de trabajo

Con el uso de las *pestañas de navegación* el programa permite al usuario variar entre los tres modos de navegadores de proyecto existentes.

- En primer lugar, se tiene el modo de navegación *Dispositivos*, empleado para la configuración del hardware de los controladores elegidos.
- La pestaña de *Aplicaciones* permite la realización de la programación de la aplicación, y el acceso a todos los elementos que se requieren para su elaboración (lista de variables globales, tareas, estructura de datos, visualizadores, POU's).
- La pestaña *Herramientas* tiene como apartado principal el administrador de bibliotecas, además de las visualizaciones que sean creadas en el proyecto.

Para que se pueda comenzar con la elaboración del programa, es necesario seleccionar un POU en el que se pueda trabajar. Cuando se crea un nuevo POU se selecciona el tipo de lenguaje de programación que se va a emplear para trabajar en él (*Ilustración 22*). En un mismo programa se pueden declarar varios POU's en los que ir programando distintas partes del proyecto, con la posibilidad de emplear un lenguaje distinto de programación en cada uno de ellos.

Para la creación de un nuevo POU se debe ir a la pestaña *Aplicaciones*, seleccionar el símbolo (+) que se encuentra a la derecha de *Application (MyController)* y hacer click en *POU....*

De esta forma se accede a la ventana que se muestra en la siguiente ilustración, donde solo queda hacer la elección del idioma deseado para la programación del autómata.

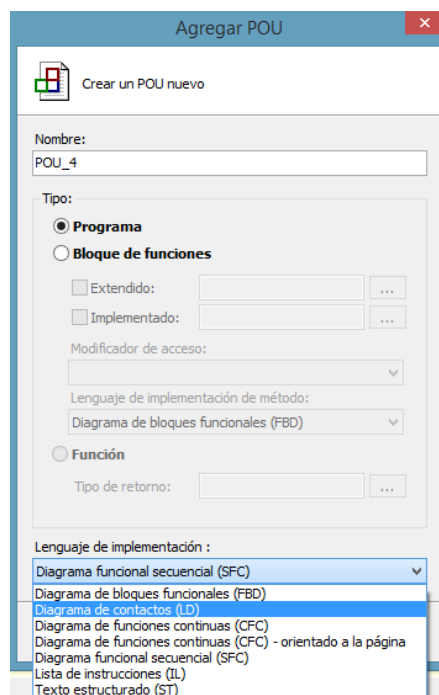


Ilustración 22: Agregar POU (Fuente: Elaboración propia)

En caso de que se tenga seleccionada la pestaña *Aplicaciones*, el software facilita un rápido y sencillo acceso a los elementos necesarios para la redacción del lenguaje de programación. La barra de *herramientas de programación* mantiene siempre a la vista los elementos más comunes a la hora de programar, funcionando como la sección de *Favoritos* del Tia Portal V13. En caso de necesitar algún elemento que no se encuentre en esta vista rápida, se recurrirá al apartado *herramientas de trabajo*, en el que se pueden encontrar todos y cada uno de ellos.

A diferencia de Tia Portal, SoMachine presenta la opción de declarar variables de la programación de forma global o de forma que sean declaradas únicamente en una ventana de trabajo determinada.

Si se quiere declarar una variable, que únicamente se quiere que sea reconocida en uno de los POU empleados, se hará uso de la ventana de declaración de variables ubicada en la parte superior del área de trabajo (Punto 4 de la *Ilustración 21*). Mientras que, si se desea declarar una variable que pueda ser identificada en cada uno de los POU del proyecto se debe recurrir a la sección *GVL*, ubicada en el navegador del proyecto de la pestaña *Aplicaciones*, tal como se muestra a continuación:

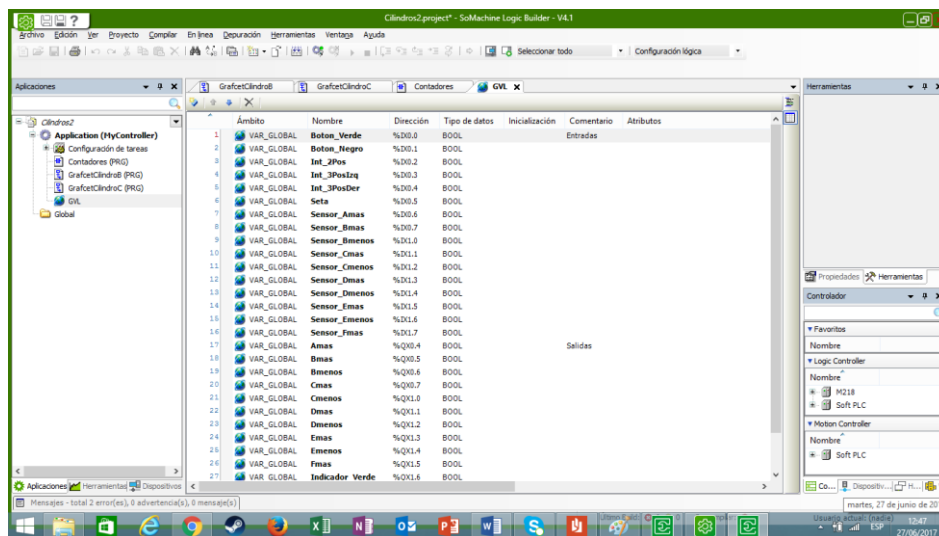


Ilustración 23: Variables globales (Fuente: Elaboración propia)

Una vez realizada la programación, ésta debe ser transferida al autómata desde el programa. Al igual que con el de Siemens, el autómata Módicon M241 Logic Controller consta de una entrada Ethernet por la que se establece la comunicación con el dispositivo.

Una vez conectado, se debe ir a la pestaña *Dispositivos*, debajo de la ventana de trabajo, en *Modalidad de conexión*, seleccionar *Dirección IP* e introducir la correspondiente al dispositivo deseado. Ya solo quedaría, en la barra de herramientas, seleccionar el

dispositivo en la ventana de trabajo, hacer click en *Compilar* y posteriormente en *Iniciar la sesión*.

4.6.3 NB-Designer

El software NB-Designer está enfocado a la realización del diseño de la interfaz y la programación de las pantallas HMI NB de Omron. Este programa permite al usuario personalizar el funcionamiento de pantallas táctiles, amoldándolo al ámbito de trabajo en el que vayan a ser empleadas. De esta forma se puede generar una interfaz intuitiva que permita la interacción con un autómata, encargado de controlar un sistema de producción, sin necesidad de recurrir a conocimientos de programación de automatización.

La *Ilustración 24* muestra la ventana principal de NB-Designer, junto a una breve aclaración del funcionamiento de cada una de las partes.

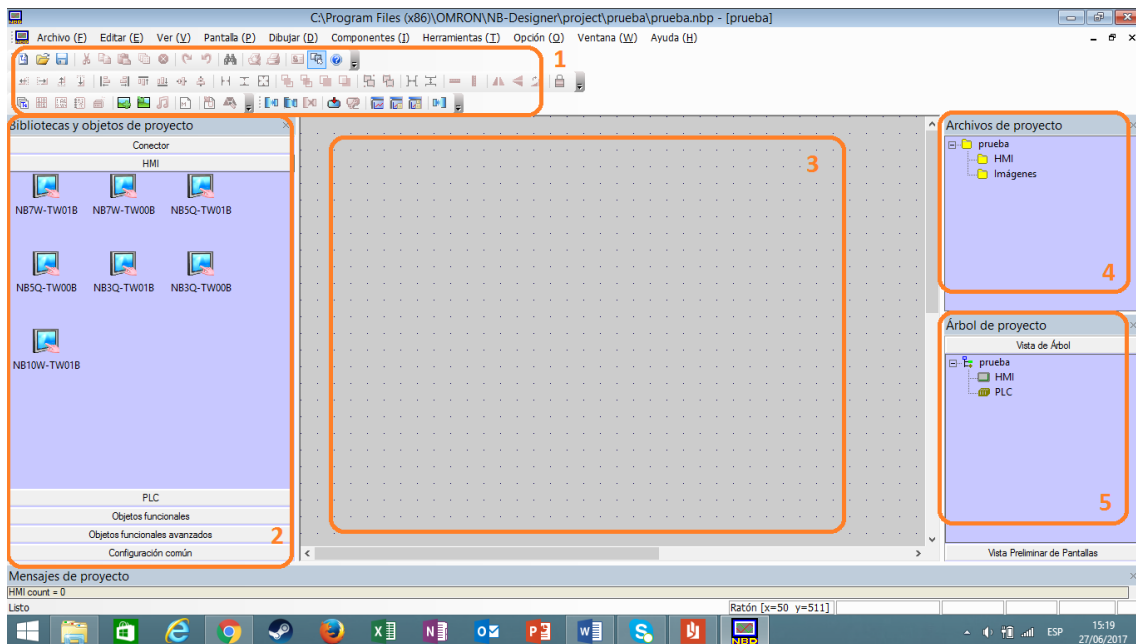


Ilustración 24:NB-Designer (Fuente: Elaboración propia)

1) Barra de herramientas: los comandos más frecuentes en esta barra a lo largo del proyecto han sido los de *Compilar*, *Descargar* y *Simulación Offline*. Con la opción *Simulación Offline*, el usuario puede ver en la pantalla del PC cómo funcionaría la interfaz de la pantalla táctil, sin necesidad de tener que disponer de ella en ese momento.

2) Ventana de elementos gráficos: a través de esta ventana se realiza la selección del tipo de conexión que se va a establecer entre los dispositivos (Puerto serie o Ethernet), el modelo de la pantalla HMI con la que se va a trabajar y el del

autómata a la que será conectada.

Por otro lado, en la ventana de elementos gráficos, también se seleccionan los objetos funcionales que vaya a participar en el diseño de la HMI. Las lámparas y los interruptores de bits han sido fundamentales para la interfaz de este proyecto.

3) Ventana de diseño: Se arrastran desde la ventana de elementos gráficos aquellos elementos con los que se quiera interactuar para poder trabajar con ellos en esta ventana. Su función principal es el diseño de la conexión entre los dispositivos y el diseño de la interfaz de la pantalla HMI.

4) Ventana de archivos de proyecto: Donde se almacenan cada uno de los archivos empleados a lo largo del diseño.

5) Ventana de estructura de proyecto: Permite al usuario navegar entre las distintas ventanas que componen el diseño de la interfaz de la pantalla, además de la creación de las mismas.

Cuando se arrastra un elemento funcional, como las lámparas o los interruptores de bits, a la ventana de diseño, se refleja en pantalla una ventana como la mostrada en la *Ilustración 25*, en la que se configura el diseño y las referencias del elemento en cuestión. A continuación, se detallan cada una de las partes fundamentales que componen esta pantalla.

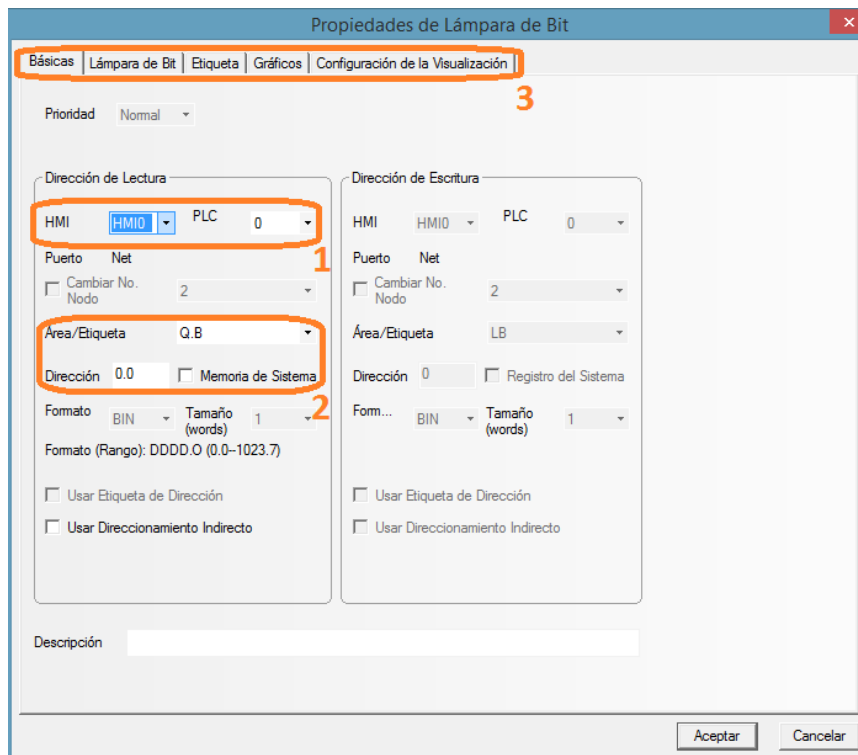


Ilustración 25: Propiedades de Lámparas de Bit (Fuente: Elaboración propia)

1) Selección de los dispositivos en los que a interactuar el objeto funcional. Previamente, al hacerse la selección de los dispositivos, se les asocia una identificación, que es la que se selecciona en este lugar para hacerle referencia.

2) Determinación del área a la que va asociado el objeto en cuestión. Para asociar un objeto funcional a una variable que haya sido transferida al autómata, se debe especificar la dirección con la que se ha programado dicha variable en el software de programación. En primer lugar, aclarar el identificador de área, y posteriormente la dirección a la que esté asociada.

3) El resto de configuraciones que se pueden hacer sobre estos elementos se reparten el resto de pestañas que se muestran en la ilustración:

- *Lámpara de bit o conmutador*: en el caso de los interruptores permite determinar si se quiere un interruptor alterno, momentáneo, en ON o en OFF.
- *Etiqueta*: selección del texto de la etiqueta que aparecerá en la interfaz, además de la opción de utilizar el modo *Biblioteca de texto*, recomendado si se quiere disponer de diferentes idiomas en la interfaz.
- *Gráficos*: importación de la imagen que se empleará para la representación del objeto funcional.
- *Configuración de la visualización*: ajustar el tamaño y posición del objeto en la interfaz de la pantalla HMI.

Una vez conectado el dispositivo, si se quiere transferir el diseño a la pantalla, se debe previamente hacer click en *Compilar todo* y posteriormente en *Descargar*.

5. Comunicaciones

Durante este apartado se mostrarán los tipos de conexiones que se han empleado para establecer la comunicación entre los distintos dispositivos que han intervenido en el proyecto.

5.1 Comunicación entre los autómatas y la maqueta

La maqueta con la que se ha trabajado consta de una conexión de tipo DB37, visible en la Ilustración 26, que facilita la conexión con cualquiera de los autómatas con los que se ha trabajado.

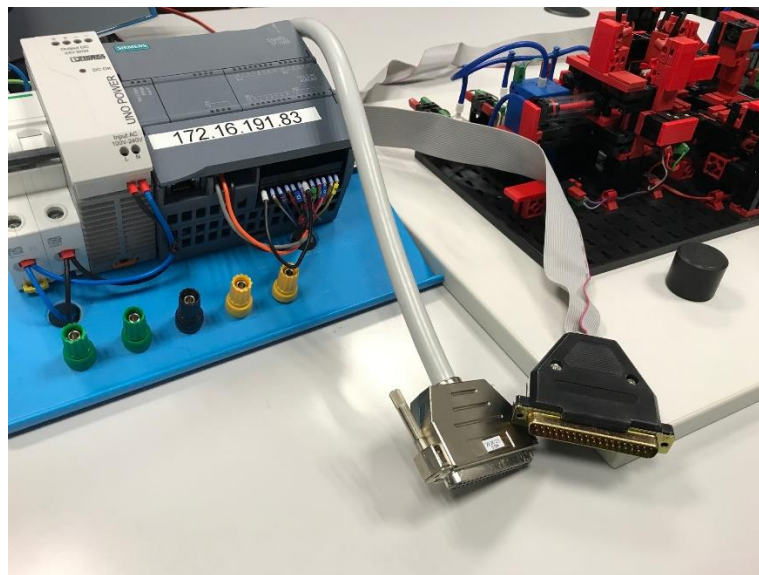


Ilustración 26: Conexión DB37 (Fuente: Elaboración propia)

Se trata de un tipo de comunicación rápida y realmente cómoda, ya que no requiere de ninguna configuración previa, por parte de los softwares de programación del autómata, para que se pueda establecer. Únicamente con realizar la comunicación macho-hembra del conector DB37, queda establecida la comunicación entre el autómata y la maqueta.

5.2 Comunicación entre los autómatas y el PC

Tanto los autómatas S7-1200 y el ModiconM241 como cualquier PC, disponen de una tarjeta de conexión Ethernet. De esta forma se permite la comunicación PLC-PC con el uso de un cable Ethernet para cada uno de los dispositivos. En la siguiente ilustración se muestra la entrada de puerto Ethernet de los dispositivos.



Ilustración 27: Conexión Ethernet (Fuente: Elaboración propia)

Previo a establecer la conexión, es necesario asignar a la tarjeta ethernet del PC una dirección IP que pertenezca a la misma red que se ha asignado el autómata. Como se ha mencionado anteriormente, cada uno de los softwares de programación ofrecen la opción de configurar la dirección IP deseado según el PLC que se vaya a emplear.

5.3 Comunicación entre pantalla HMI y PC

La pantalla HMI tiene un tipo de conexión con el PC igual que la que tienes los autómatas. Establecen la comunicación con el ordenador vía Ethernet, y dependen de una dirección IP para poder comunicarse correctamente. En este caso, la asignación de la dirección IP se hace a través del software de diseño de la pantalla, el NB-Designer.

5.4 Comunicación entre pantalla HMI y PLC

Como consecuencia de que todos los dispositivos mencionados poseen una tarjeta Ethernet, y la comunicación entre ellos con el PC se ha realizado a través de la misma, resulta de mayor comodidad hacer los mismo para establecer la comunicación entre HMI-PLC.

A la hora de trabajar con NB-Designer se ha seleccionado la opción de comunicación Ethernet, por lo que este software se encarga de establecer esta comunicación con el PLC de la misma forma que lo ha hecho con el PC.

6.Desarrollo del proyecto

Una vez que se han expuesto cada uno de los elementos participantes en el proyecto y se ha indagado en el funcionamiento de cada uno de los softwares, necesarios para la programación de cada uno de los dispositivos, se procede a desarrollar el procedimiento que se ha llevado a cabo para la automatización de la maqueta con cada uno de los autómatas, y su monitorización a través de la pantalla HMI.

6.1. Funcionamiento de la maqueta del centro neumático

En el ámbito laboral, cualquier centro de mecanizado, por no decir cualquier cadena de producción automatizada, ofrecerá la opción de poder seleccionar, con la ayuda de una pantalla HMI, varios modos funcionamiento con los que se quiera trabajar.

Por ello, en este proyecto, se ha querido diseñar tres posibles modos de funcionamiento con los que pueda trabajar la maqueta:

- Por un lado, se dispondrá del modo **automático**, en el que el trabajador únicamente tiene que activar el arranque del sistema, y dar la orden de que empiece a trabajar mediante el pulsador de la pantalla *Start*. Con este modo, el centro neumático comenzará a producir un número indefinido de piezas sin descanso. Es necesario volver a seleccionar el botón *Start* para desactivarlo, y a su vez, esto hará que el proceso se detenga al instante.

En el apartado 4.3 *Funcionamiento del proceso* se ha explicado cómo actúan cada uno de los sensores y actuadores para la correcta mecanización de las piezas en este modo de funcionamiento.

- Además del modo automático, es fundamental que un proceso de producción cuente con un modo **manual** de trabajo, ya que de esta forma se puede controlar cada una de las acciones de forma separada. El modo manual, además de ofrecer una producción con el control total sobre cada uno de los actuadores del sistema, permite poder hacer revisiones sobre cada una de las partes del proceso sin que sea necesaria la presencia de una de las piezas a mecanizar.

Supóngase que la producción está dando algún tipo de problema y se quiere saber cuál es el actuador que lo está produciendo. Para ello se puede hacer uso del modo manual, ya que permite activar cada uno de los actuadores sin la necesidad de la activación de los sensores del sistema. De esta forma, tras probar cada una de las partes, será sencillo dar con el causante del problema.

Para el modo manual, se ha creado una ventana en la pantalla HMI con un botón de bit para cada uno de los actuadores que se encuentran en el proceso. Activando el botón de bit deseado, se pondrá el funcionamiento el actuador correspondiente.

- Para finalizar, se ha creado el modo **número de piezas**. Este modo de funcionamiento es ideal para aquellas empresas que se dediquen a hacer un número de mecanizados determinado al día, ya que, con un solo click, se le da a la orden al sistema de que se detenga cuando finalice el mecanizado del número de piezas seleccionado.

Se ha diseñado una nueva ventana en la pantalla HMI que dispone de varios botones de bits. Cada uno de los botones representa un número determinado de piezas a producir, por lo que, al ser seleccionado, y pulsar *Start*, el proceso de producción comenzará a trabajar hasta que mecanice la cantidad de piezas seleccionadas.

Los valores de cada uno de los botones que determinan el número de piezas a mecanizar podrían ser ajustados según las dimensiones de producción de cada una de las empresas. A modo de ejemplo se han creado pulsadores de producción de 100, 75, 50, 25 y 2 piezas.

6.2 De Diagrama Funcional Secuencial a Lenguaje Ladder

Durante el curso previo a la elaboración de este proyecto, se han adquirido conocimientos sobre automatización de procesos, en los que se empleaba el lenguaje de diagrama funcional secuencial (Grafcet) para realizar la programación de los autómatas con los que se trabaja.

Como se ha mencionado anteriormente, el software Tia Portal V13 no permite una completa programación de sus autómatas a través de este tipo de lenguaje, por lo que ha sido necesario expandir los conocimientos sobre este tema, con el estudio del tipo de lenguaje Ladder, también conocido como diagramas de contactos.

En primer lugar, se ha redactado a mano el código de programación con el uso de diferentes Grafcet dedicados a cada una de las funciones del proceso. Un Grafet enfocado a la recolocación previa de todo el sistema, otro para el funcionamiento de la mesa giratoria, otro para la cinta de transporte y otros encargados del accionamiento de cada uno de los cilindros que participaban en la cadena de mecanizado, tanto los de empuje, como el de la prensa.

Una vez diseñado este código, y con el fin de poder implementarlo en los softwares deseados, se ha pasado al lenguaje Ladder con la ayuda del documento *PROGRAMACIÓN DE UN GRAFCET EN LENGUAJE LD (ESQUEMA DE CONTACTOS)*. En este documento, con la ayuda de un ejemplo que se plantea, se explica paso a paso las pautas a seguir para, dado un código con estructura de Grafcet, poder pasarlo a diagrama de contactos, respetando las acciones y condiciones que se imponen en este código.

Con el lenguaje de programación Ladder diseñado, el siguiente paso se trataba de realizar un intensivo estudio sobre el funcionamiento de cada uno de los softwares.

6.3 Primeros pasos con Tia Portal V13 y SoMAchine

Este es uno de los apartados que más breves de la redacción, pero uno de los de mayor peso a lo largo de este proyecto.

La familiarización con los softwares de programación de cada uno de los autómatas ha sido una pieza fundamental para los lograr los objetivos planteados.

Ha sido necesario un intensivo estudio de los manuales de funcionamiento de cada uno de los softwares, manuales de formación y guías de programación, al igual que el uso de tutoriales que aportaran información relevante para poder proyectarla en este caso particular.

Se trata de un apartado de formación que igual no se ve reflejado de forma directa en el proyecto, pero es de gran importancia por los conocimientos que se han adquirido a lo largo de su elaboración.

6.4 Programar los distintos Hardware

Para la identificación de los PLC por parte de cada uno de sus softwares de programación es necesario realizar una configuración previa desde el mismo programa en la que se determine el PLC a programar, el tipo de comunicación y elementos complementarios del autómata en caso de contar con ellos.

Para la redacción de este apartado se estudiará la configuración de cada uno de los autómatas por separado.

6.4.1 Configuración S7-1200 en Tia Portal V13

Como el software de programación empleado para la programación de este dispositivo es el Tia Portal V13, será en este programa donde se deba configurar el PLC con las mismas características que con el de trabajo.

En primer lugar, es necesario seleccionar aquel controlador que tenga las mismas características que con el que se va a comunicar, incluyendo la referencia. Para ello, se ha seleccionado, dentro de *Vista del portal, Dispositivos y redes -> Agregar dispositivo -> Controladores -> Simatic S7-1200 -> CPU -> CPU 1214 AC/DC/Rly* y la referencia *6ES7 214-1BG40-0XB0*. Posteriormente se ha hecho click en *Agregar (Ilustración 29)*.

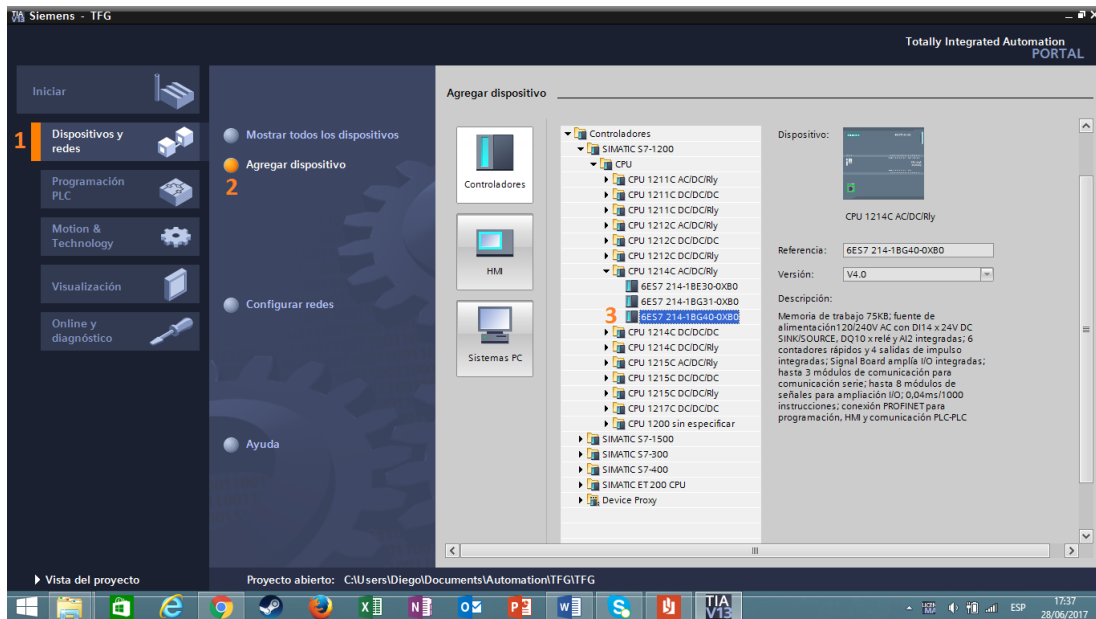


Ilustración 28: Selección del dispositivo (Fuente: Elaboración propia)

Dado que el autómata empleado consta de una expansión de entradas y salidas del tipo 6ES7 232-4HA30-0XB0, se ha añadido al autómata seleccionado en el programa de la siguiente manera:

Dentro de *Vista del proyecto*, seleccionando *Configuración de dispositivos* del Árbol del proyecto, se ha recurrido al catálogo ubicado a la derecha, y seleccionado *Signal Board* -> *AQ* -> *AQ 1X12BIT* -> *6ES7 232-4HA30-0XB0*. Debe ser arrastrado al recuadro vacío que se encuentra en la ilustración del autómata.

Puestos en este punto, únicamente queda la configuración de la comunicación entre el software y el autómata. Para ello, seleccionando el puerto Ethernet, representado de verde en el autómata, se ha accedido al apartado *Protocolos IP*, dentro de *Direcciones Ethernet*, en que se ha introducido la dirección IP del autómata (172.16.191.83)

En la siguiente Ilustración se muestran los apartados mencionado previamente, en los que selecciona la *Signal Board* deseada, y se accede al *Protocolo IP*:

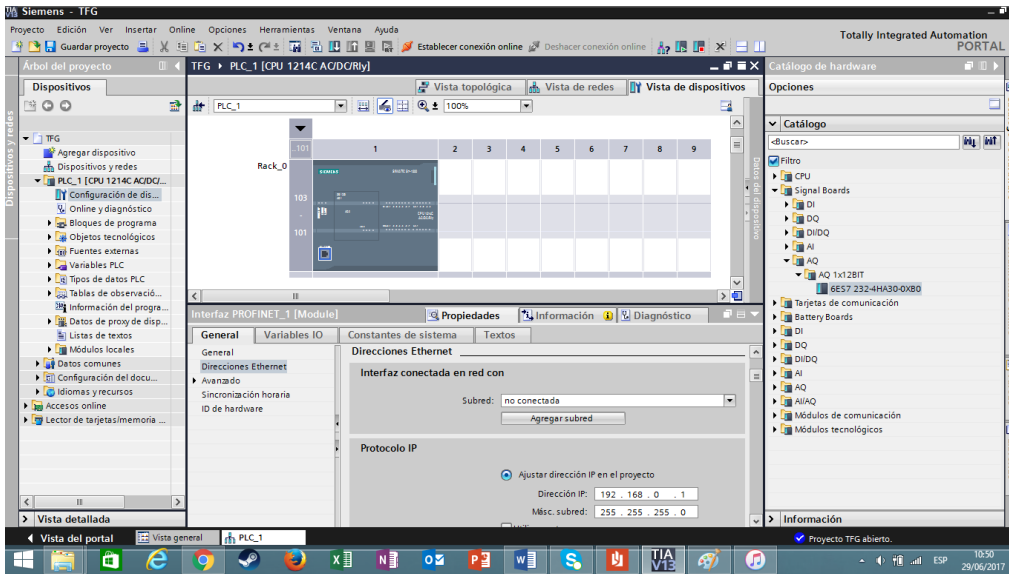


Ilustración 29: Signal Board y Dirección IP (Fuente Elaboración propia)

6.4.2 Configuración Modicon 241 en SoMachine

Al igual que con el software anterior, el primer paso a dar para comenzar con el nuevo proyecto es la selección del dispositivo deseado, de forma que cumple con las mismas características que el autómata con el que se va a establecer comunicación.

Para ello, desde la ventana *SoMachine Central*, seleccionando *Añadir y eliminar dispositivos*, se ha accedido a una ventana en la que se facilitan los distintos modelos de autómatas disponibles para trabajar con este software. Para este proyecto se ha seleccionado Controlador lógico -> M241 -> TM241CE40R y con el uso de la flecha que aparece en mitad de pantalla y haciendo click en *Aceptar*, ha sido añadido el autómata en cuestión.

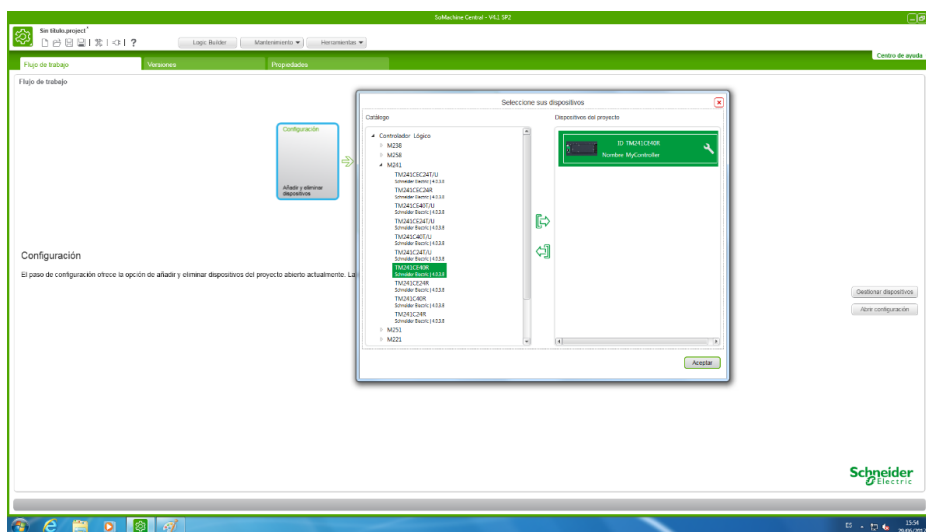


Ilustración 30: Selección PLC M241 (Fuente: Elaboración propia)

De forma similar que ha sido necesario añadir la Signal Board en Tia Portal, en este caso se deben añadir los dos cartridge con los que consta el PLC. Desde la ventana *Logic Builder*, se han seguido los siguientes pasos: Dispositivos -> I/O Modules -> Cartridges (TMC4AI2 en Cartridge_1 y TMC4AQ2 en Cartridge_2). Se hace arrastrandolos con el ratón hasta el dispositivo del proyecto. (Ilustración 31).

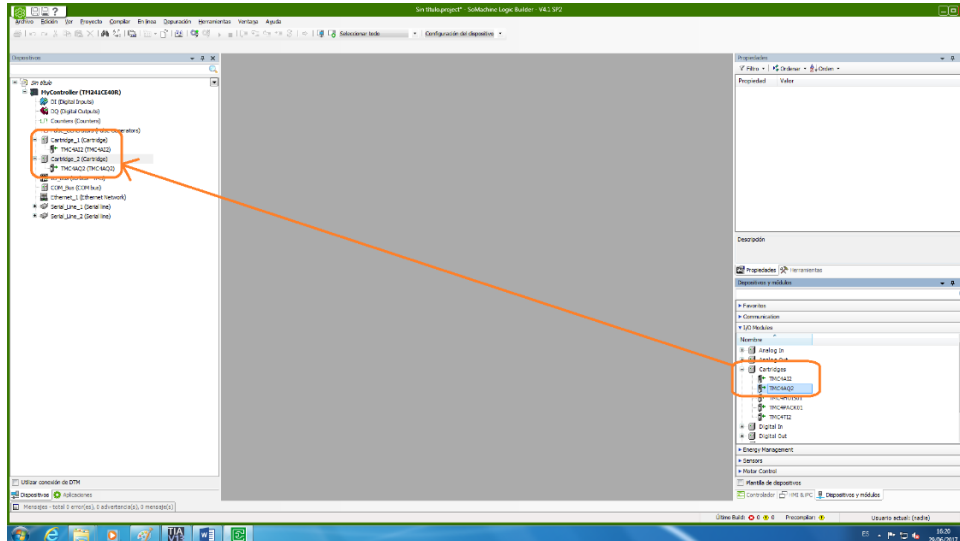


Ilustración 31: Insertar cartridge (Fuente: Elaboración propia)

Con este último pase, ya solo quedaría por completar la configuración de la comunicación entre el Modicon 241 y SoMachine.

Manteniendo la Ventana Logic Builder, se ha seleccionado, en el navegador del proyecto, MyController (TM241CE40R) -> Modalidad de conexión -> Dirección IP e introducido la dirección IP 172.16.191.104.

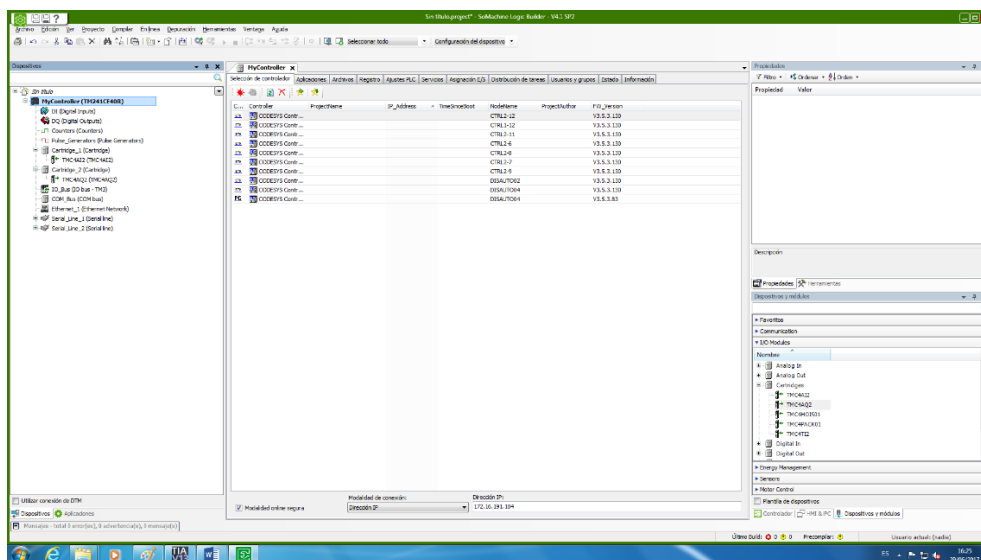


Ilustración 32: Comunicación SoMachine - M241 (Fuente: Elaboración propia)

6.5 Introducción de las variables

Una vez configurados cada uno de los dispositivos en sus correspondientes programas, el siguiente paso a realizar ha sido la declaración de todas las variables que participan en el lenguaje de programación creado.

Resulta de mayor comodidad introducir y declarar cada una de las variables antes de implementar el lenguaje en el software, ya que de este modo se pueden ordenar al gusto del usuario, y pueden ser arrastradas desde la ventana de variables hasta el código en cuestión para asignarlas a los elementos deseados.

6.5.1 Introducción de variables TIA Portal V13

Para realizar la declaración de variables en TIA Portal V13 se ha hecho uso de la sección del *Árbol del Proyecto* -> *Variables PLC* -> *Agregar tabla de variables*. En esta nueva tabla de variables es donde se han declarado todas y cada una de las variables introducidas de forma manual. Interesa trabajar en la sección *Mostrar todas las variables*, ya que es donde se pueden encontrar algunas variables, como las marcas especiales mencionadas en el apartado 4.5.2 *Direccionamiento de la memoria*, que no se encontraran en la tabla agregada por el usuario.

En la siguiente ilustración se puede contemplar la vista del proyecto en la que el usuario se encuentra a la hora de definir las variables con las que trabajar:

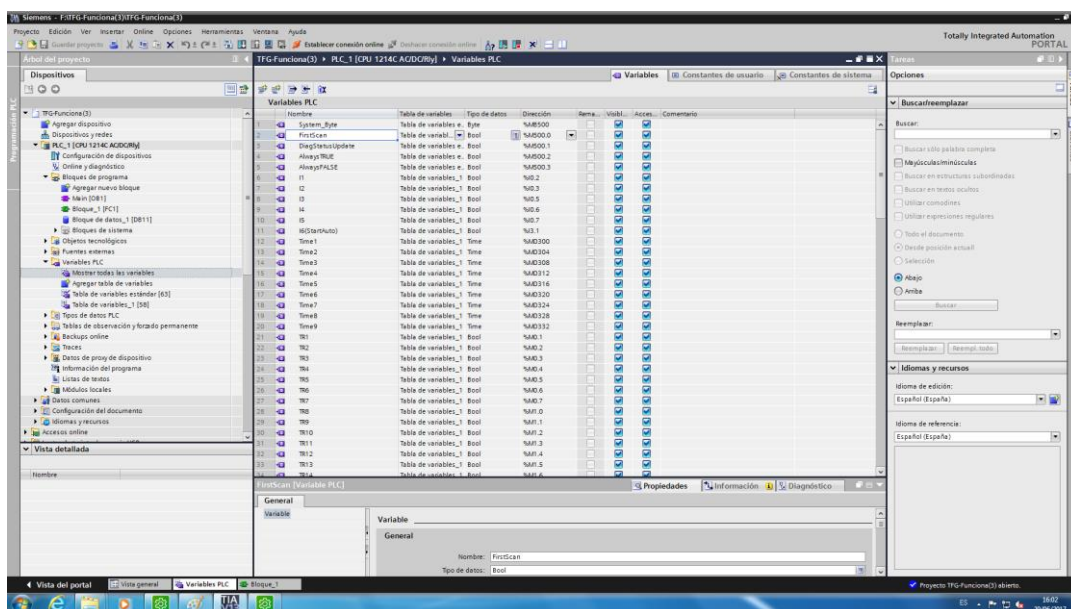


Ilustración 33: Declaración de variables TIA Portal (Fuente: Elaboración propia)

Para la correcta declaración de una variable, hay que tener en cuentas cada uno de los apartados que se muestran en la tabla de variables reflejada en el área de trabajo:

- **Nombre:** En esta columna se les da nombre a cada una de las variables. De esta forma, se pueden utilizar referencias que ayuden al usuario a identificar el funcionamiento de cada una de las variables con solo ver su nombre. A la hora de implementar el código de programación, TIA Portal identificará las variables que hayan sido declaradas, o bien por su dirección o bien por el nombre que se les ha asociado.

En este proyecto se han llamado a las variables por el orden de aparición a lo largo de los distintos Grafset que se han diseñado, tanto para las transiciones (TR1, TR2, TR3...), como para las etapas (X1, X2, X3...). El nombre de las entradas (I1, I2, I3...) y salidas (Q1, Q2, Q3...) se ha mantenido tal como se encontraban en el informe dedicado a las direcciones de los sensores y actuadores de la maqueta.

- **Tipo de datos:** Para la elaboración de este proyecto, únicamente ha sido necesario el uso de tipos de datos de carácter booleano, es decir, que únicamente puedan obtener dos valores, 0 y 1. Se trata del tipo de datos más frecuente en la programación de sistemas de producciones industriales tales como la tratada en este caso.
- **Dirección:** Es donde se declaran las direcciones de las variables empleadas. El tipo de dirección seleccionado en relación a la función de la variable se ha explicado previamente en el apartado *4.5.2 Direccionamiento de la memoria*.

Para declarar la dirección de las marcas especiales, al igual para activarlas o desactivarlas, se ha de ir a Configuración de dispositivos -> General -> Marcas de sistemas y de ciclo y activar la utilización del byte de marcas de sistema. Debajo se facilita un apartado donde introducir la dirección del byte de las mismas.

En este caso, como consecuencia de seleccionar una dirección de byte de marcas de sistemas igual a 500, se tiene como marca especial de función de primer ciclo %M500.0 (FirstScan).

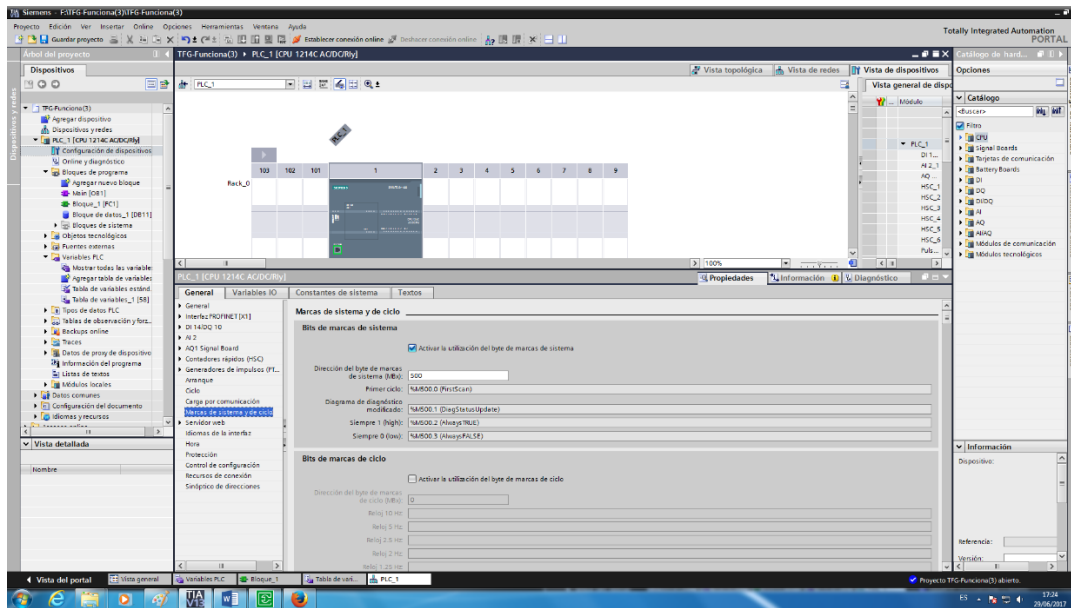


Ilustración 34: Variables de marca TIA Portal (Fuente: Elaboración propia)

6.5.2 Introducción de variables SoMachine V4.1

Como se ha explicado en el apartado 4.6.2 *SoMachine V4.1*, este software ofrece la opción de declarar las variables de forma global, o de forma que únicamente sean reconocidas por uno de los POU con los que se trabaje.

Durante este proyecto se han declarado todas las variables de forma global, por tener una forma de trabajo similar a la empleada con TIA Portal. En primer lugar, se han declarado todas las variables, y posteriormente se ha pasado a la elaboración de cada uno de los POU's.

En caso de SoMachine, la declaración de las variables no se realiza completando una tabla que es introducida por el usuario, sino que se hace con código de escrita que respeta el siguiente orden: [Nombre] AT [Dirección] : [Tipo de dato].

La funcionalidad de cada una de estas partes del código es misma que la explicada en el apartado anterior sobre las variables de TIA Portal.

En la siguiente ilustración se puede contemplar cómo sería la pantalla en la que se realiza la introducción este código. Para acceder a ella se debe hacer click en Aplicaciones -> Application (MyController: TM241CE40R) -> GVL.

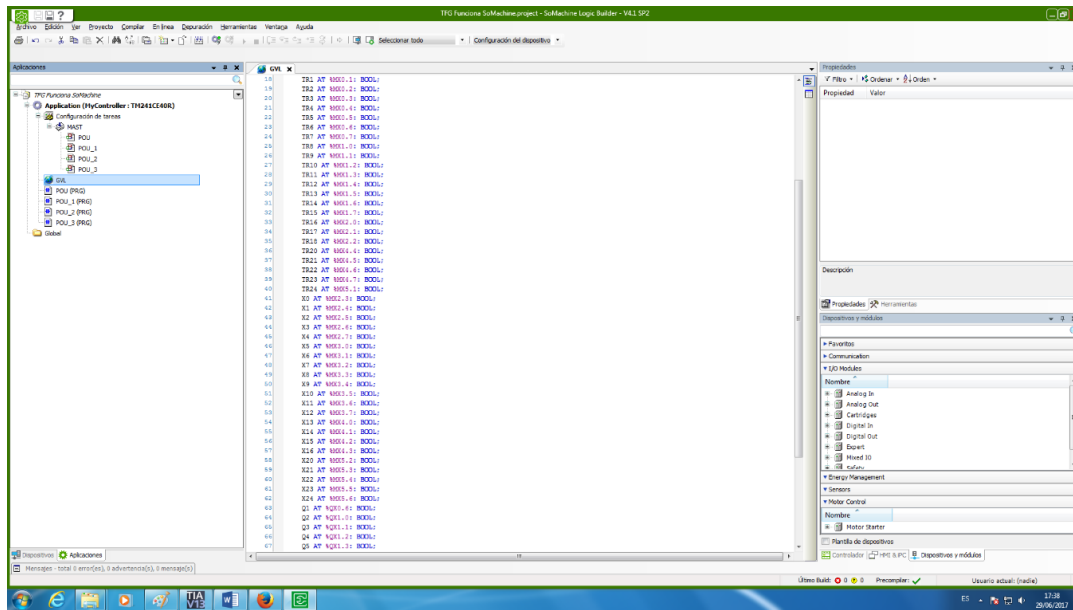


Ilustración 35: Declaración de variables SoMachine (Fuente: Elaboración propia)

En el caso de SoMAchine no es necesario activar las marcas especiales para poder hacer uso de ellas. Por lo que únicamente ha sido necesario, a la hora de implementar el código, crear un nuevo elemento al que se le llamó *IsFirstMastCycle*. De esta forma el software ya asimila que esa variable debe activarse nada más arrancarse el proceso.

6.6 Implementación del código e programación

Una vez declaradas todas las variables que participan en la programación de los autómatas el siguiente paso a dar ha sido la implementación del lenguaje de programación, realizado por diagrama de contactos, a cada uno de los softwares. A pesar de que SoMachine permite la lectura de un código de programación basado en Grafnets, se ha decidido mantener el lenguaje LD para ambos programas, ya que, de esta forma, resulta más cómodo poder trabajar de forma simultánea con ambos programas. Además, en caso de tener que realizar algunos cambios, ayuda tener un único lenguaje que retocar.

En general, los programas dedicados a la programación de autómatas, mantienen una estructura y un orden similar a la hora de introducir el código de programación. Pero lógicamente existen diferencia al hacerlo con cada uno de los softwares.

6.6.1 Implementación del lenguaje en TIA Portal V13

Como se ha planteado previamente el código de programación previamente a mano, en este apartado lo único que se ha debido es copiarlo en el programa introduciendo cada uno de los elementos y las variables que lo componen.

Los autómatas programados por TIA Portal V13, para leer el código de programación, se centran en la función Main [OB1], que se puede encontrar dentro de *Bloques de programas*. Por lo que, si se quiere crear un nuevo bloque de función en el que introducir el código, se debe hacer una llamada de dicho bloque desde el bloque Main, para que el autómata lo lea.

En este proyecto se ha preferido crear un nuevo bloque de función para trabajar en él, por lo que los pasos a seguir para importar el código han sido los expuestos a continuación.

En primer lugar, se ha creado un nuevo bloque de programa de tipo *Función*. Para ello se ha seleccionado, dentro de *Árbol del proyecto* -> *PLC_1 [CPU 1214 AC/DC/Rly]* -> *Bloques de programa* -> *Agregar nuevo bloque*. Dando lugar a la siguiente ventana:

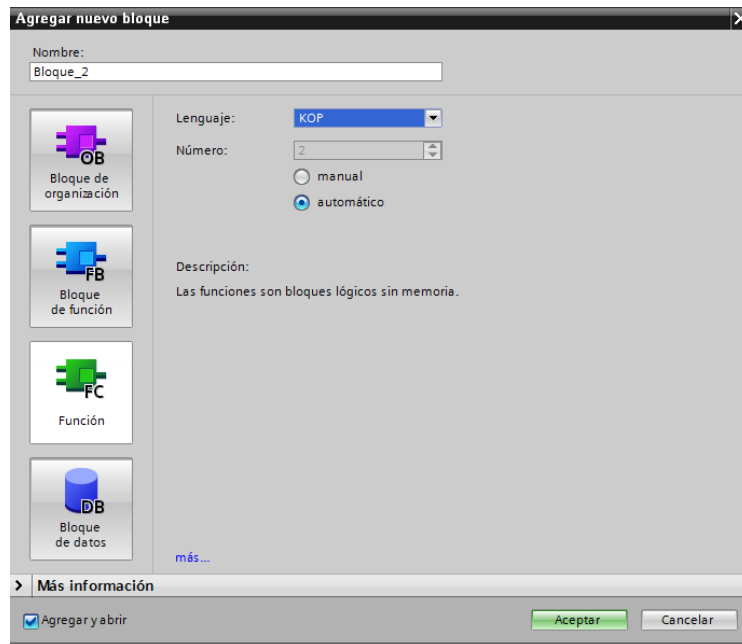


Ilustración 36: Introducir Bloque de Función (Fuente: Elaboración propia)

En esta ventana, se ha seleccionado el icono en verde *Función* y el lenguaje de tipo KOP (Diagrama de contactos).

Una vez que creado, y desde el bloque de programa Main [OB1], se ha creado un cuadro vacío con el uso de la barra *Favoritos* en la que se encuentran los elementos. A este cuadro vacío se le ha asignado el nombre de Bloque_1 (*Ilustración 37*). De esta forma el programa, al acceder al bloque Main, será enviado directamente al Bloque_1, en donde hará la lectura de todo el código que se introduzca en él.

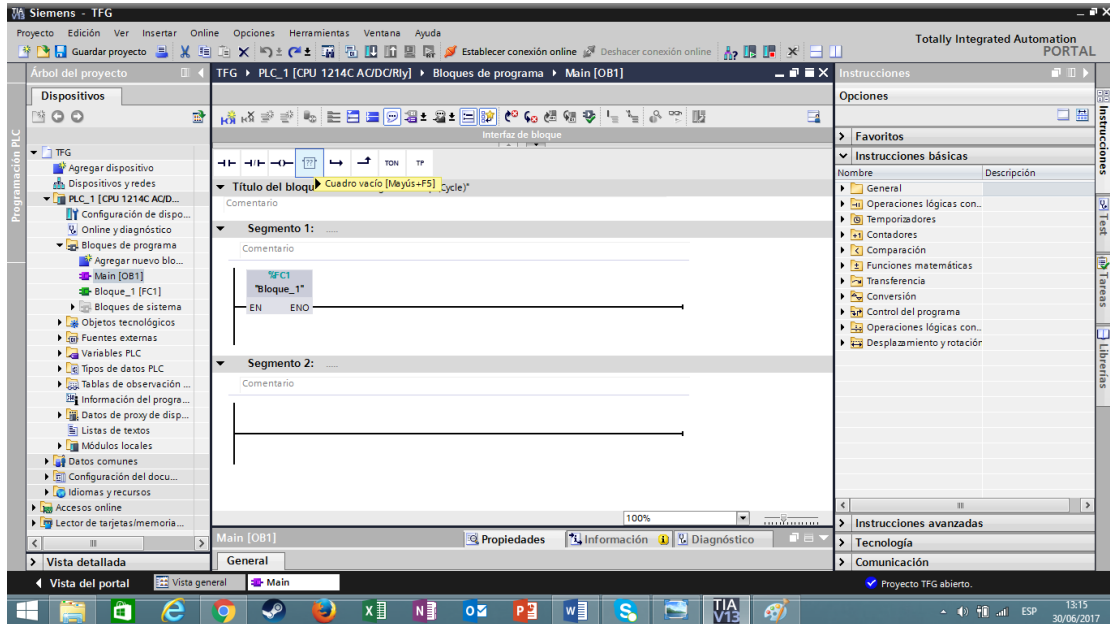


Ilustración 37: Llamada a Bloque_1. TIA Portal (Fuente: Elaboración propia)

Una vez hecha la llamada del Bloque_1, haciendo click sobre el mismo en el Árbol del proyecto, se accede al área de trabajo en la que se ha introducido el código.

TIA Portal facilita la opción de dividir el código por segmentos a modo de organización. Se han empleado cinco segmentos diferentes para organizar el código empleado. En primer lugar, un segmento en el que se han introducido los contadores empleados para el modo de trabajo *número de piezas*, otro segmento para determinar el estado de arranque del proceso, otro para imponer las condiciones de activación de cada una de las variables, el siguiente para declarar qué transiciones activan qué etapas de las que se encontraban en el Grafcet (recuérdese que se ha trabajado con un lenguaje de diagrama de contactos basado en uno diseñado previamente en Grafcet) y por último un segmento dedicado a establecer cuáles son las etapas que activan los distintos actuadores.

En la *Ilustración 38* se muestra un ejemplo de una parte de uno de los segmentos utilizados, el dedicado a las condiciones de la activación de las transiciones.

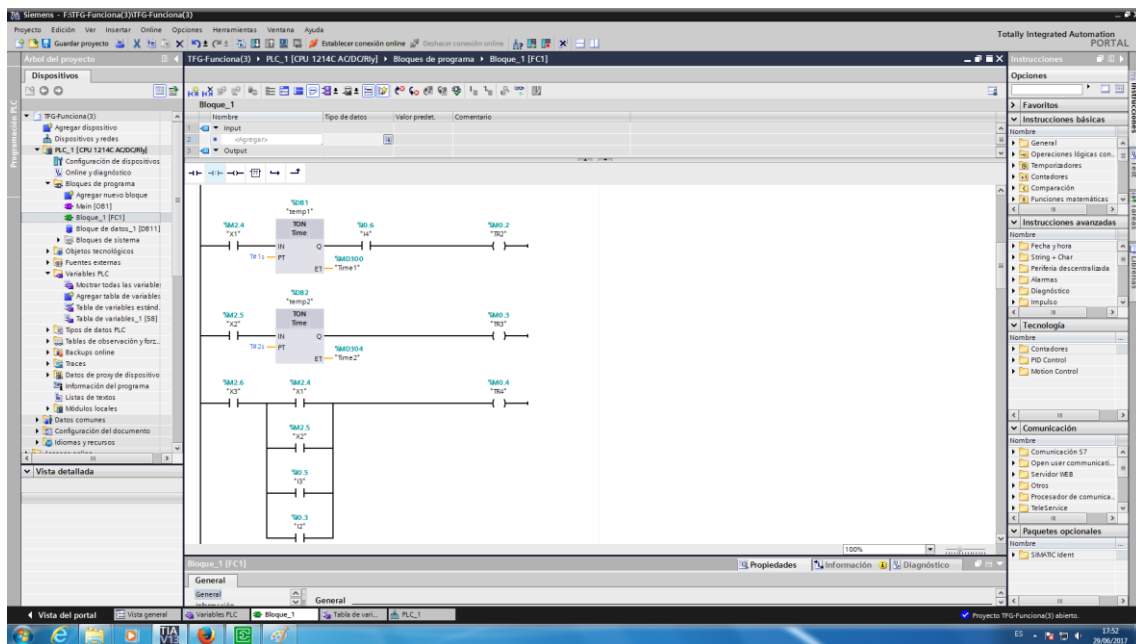


Ilustración 38: Área de trabajo Bloque_1 en TIA Portal (Fuente: Elaboración propia)

Una vez copiado todo el código en el Boque_1, el proyecto está listo para ser transferido al autómata.

6.6.2 Implementación del lenguaje en SoMachine

Como se ha mencionado anteriormente, el código de programación empleado tanto TIA Portal como en SoMachine es el mismo, por lo que la estructura de los distintos segmentos explicada en el apartado anterior se ha mantenido con este programa. La principal diferencia es que SoMachine no ofrece una división por segmentos dentro de un mismo POU, por lo que se ha creado un nuevo POU por cada segmento utilizado en TIA Portal.

En el caso de los autómatas programados por SoMachine, la lectura de la programación se efectúa en el *MAST*, ubicado en *Aplicaciones -> Configuración de tareas*. Por ello, se debe arrastrar cada uno de los POU creados hasta este apartado, para que, de esta forma, el autómata reciba la llamada de los mismos a través del *MAST* (*Ilustración 39*).

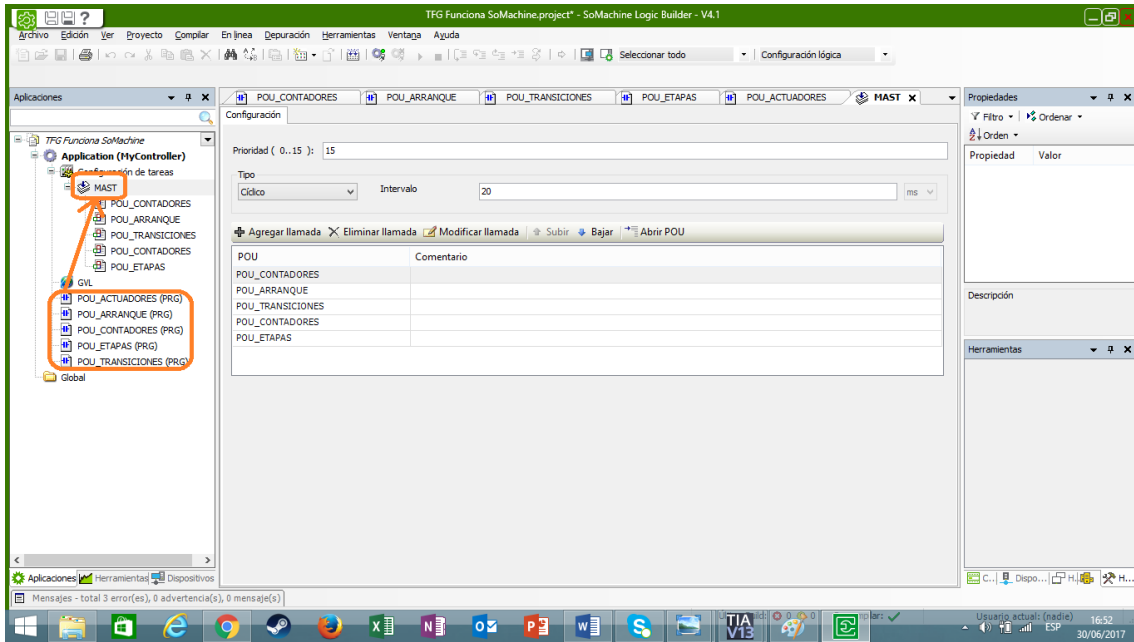


Ilustración 39: Llamada a los POU's en SoMachine (Fuente: Elaboración propia)

Como consecuencia de la similitud en la estructura de los programas empleados, una vez realizado este apartado con TIA Portal, no se han prestado complicaciones a la hora de hacerlo con SoMachine.

En este caso, los elementos a incluir en el código se han ido añadiendo desde la barra de herramientas de programación, en donde se han podido encontrar todos y cada uno de ellos.

A continuación, se muestra un apartado de uno de los POU's a modo de ejemplo.

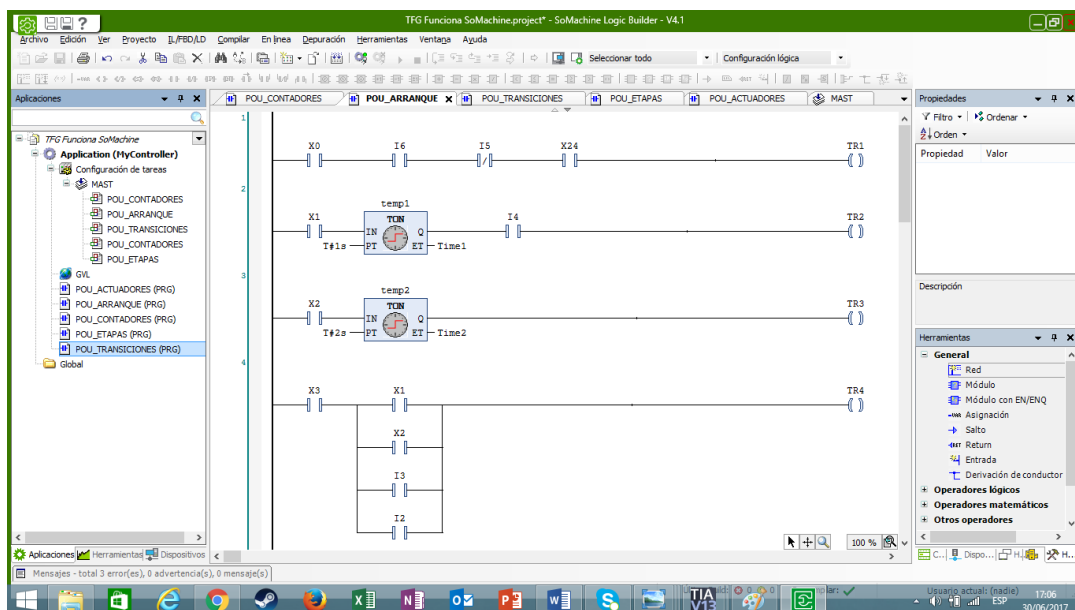


Ilustración 40: Área de trabajo POU_TRANSICIONES, SoMachine (Fuente: Elaboración propia)

Una vez pasado todo el código al software se ha comenzado con el proceso de transferencia del mismo al autómata.

6.7 Transferir a PLC y comprobación del funcionamiento

Una vez que se ha implementado el código en cada uno de los softwares y se han conectado los autómatas con el PC, tal como se ha mencionado en el apartado 5.2 *Comunicación entre los autómatas y el PC*, el código ya está listo para ser transferido al autómata correspondiente. Como la forma de comunicarse con el autómata no es la misma para los distintos programas, se ha dividido este apartado en dos partes.

6.7.1 Transferencia del código en TIA Portal V13

Para transferir el programa al autómata de Siemens, se ha hecho uso de los comandos *Compilar* y *Transferir* que se encuentran en la barra de herramientas del programa. Es necesario compilar previamente el programa para que el propio software haga una comprobación de si existe algún error de programación en el código. Una vez compilado y seleccionada la herramienta *Transferir*, se ha accedido a la ventana que se muestra en la siguiente ilustración:

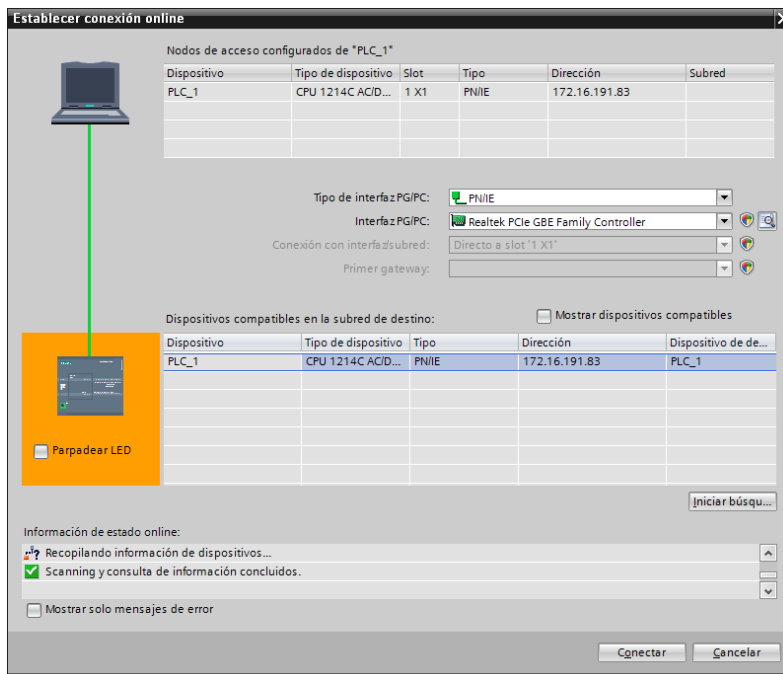


Ilustración 41: Transferencia del código de programación en TIA Portal (Fuente: Elaboración propia)

Desde ésta venta, para lograr establecer comunicación con el autómata, es necesario seleccionar el tipo de interfaz *PN/IE* y la interfaz *Realtek PCIe GBE Family Controller* de los desplegables que se muestran en la ilustración. A continuación, haciendo click en

Iniciar búsqueda... se ha detectado el autómata correspondiente a la dirección IP que se ha introducido previamente.

Una vez detectado, y seleccionando *Conectar*, se ha transferido el código del programa al autómata, estando listo para ser puesto en funcionamiento con la maqueta.

6.7.2 Transferencia del código en SoMachine

En el caso de SoMachine, la conexión con el autómata de Schneider se logra desde la pestaña *Dispositivos*, de la pestaña de navegación. En el apartado *Modalidad de conexión* se ha seleccionado el modo de conexión *Dirección IP*, y a continuación se ha introducido la dirección IP propia del autómata empleado (IP 172.16.191.104), tal como se muestra en la *Ilustración 42*.

En el listado de controladores que aparecen en pantalla, se ha seleccionado el TM241CE40R, se ha compilado el programa desde la barra de herramientas, y se ha seleccionado la opción de transferir.

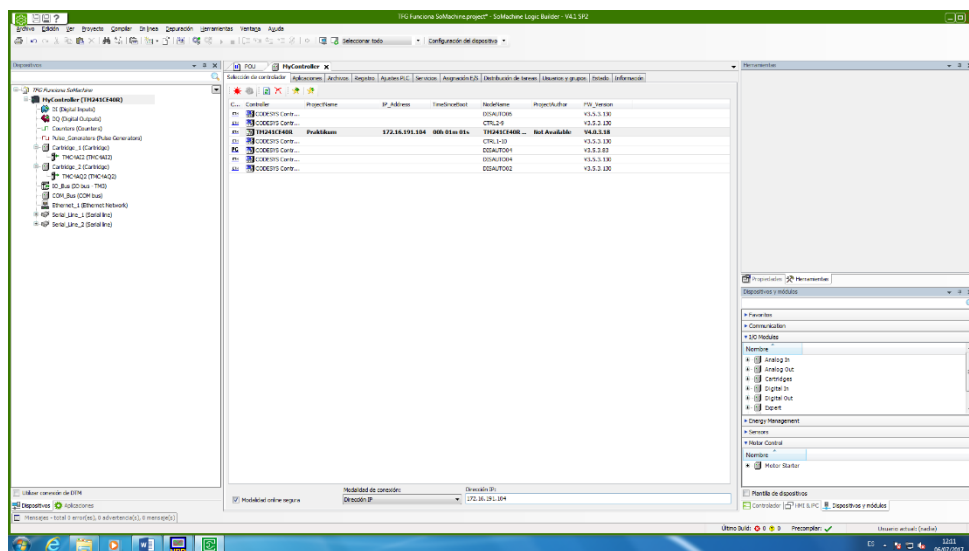


Ilustración 42: Transferencia del código de programación en SoMachine (Fuente: Elaboración propia)

En el listado de controladores que aparecen en pantalla, se ha seleccionado el TM241CE40R, se ha compilado el programa desde la barra de herramientas, y se ha seleccionado la opción de transferir.

Han existido determinados problemas a la hora de establecer la conexión con los autómatas, ya que en caso de introducir incorrectamente la dirección IP de los PLC, estos se desconfiguran. En numerosas ocasiones ha sido necesaria la ayuda de los técnicos de laboratorio para configurar de nuevo los autómatas correctamente.

Una vez que se ha transferido correctamente, se ha pasado a comprobar el correcto funcionamiento de la maqueta según el código impuesto, y realizado algunos ajustes en el código para corregir pequeños errores de programación.

Finalizado este apartado, el autómata ya está configurado para que el proceso de mecanizado funcione correctamente con ambos autómatas.

6.8 Configurar comunicación de NB-Designer con los PLC's

NB-Designer ofrece una forma muy sencilla de establecer el tipo de conexión entre la pantalla HMI y el autómata en cuestión, basada en arrastrar los elementos deseados a la ventana de diseño y especificar la dirección IP de cada uno de ellos, ya que la conexión empleada es vía Ethernet.

A continuación, y con la ayuda de alguna ilustración, se mencionarán los pasos dados para configurar los dispositivos a utilizar y la comunicación entre ellos:

- 1) En el apartado *Bibliotecas y objetos de proyecto*, en la pestaña *HMI* se ha seleccionado y arrastrado a la ventana de diseño la pantalla *NB7W-TW01B*. Se ha seleccionado el modo de visualización horizontal, ya que se adapta mejor a la representación de la maqueta.
- 2) En la pestaña *PLC* se ha seleccionado el modelo del autómata deseado, por ejemplo, para el caso del Siemens, se ha seleccionado y arrastrado el *SIEMENS S7-1200 Ethernet (TCP Slave)*.
- 3) El siguiente paso ha sido seleccionar el tipo de conexión que se establece entre los dispositivos, para ello, nuevamente en la biblioteca se ha accedido a la pestaña *Conector*, de donde se ha arrastrado a la ventana de diseño el tipo de conexión Ethernet.
- 4) Una vez que se arrastra la comunicación Ethernet se abre una ventana de configuración de la comunicación, en la que se añaden los dispositivos, junto a sus direcciones IP correspondientes.

En primer lugar, se ha añadido la pantalla HMI, para ello, en el modo *Tipo* de la nueva pantalla, se ha seleccionado *HMI*. A continuación, se ha añadido la dirección IP 172.16.191.54, propia de la pantalla. Para finalizar, en el apartado *Protocolo*, se ha seleccionado el modelo de autómata con el que se va a comunicar esta pantalla.

En segundo lugar, se ha añadido el PLC haciendo click en *PLC*, dentro del apartado *Tipo*, e introduciendo la dirección IP 172.16.191.83, correspondiente al autómata empleado.

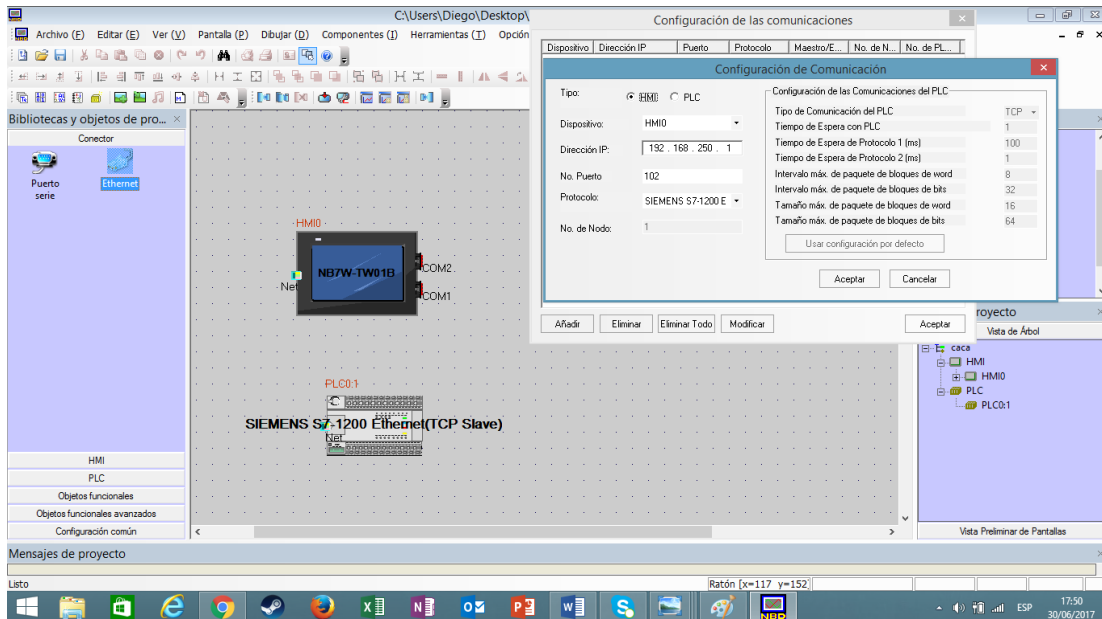


Ilustración 43: Comunicaciones NB-Designer (Fuente: Elaboración propia)

- 5) Haciendo click en Aceptar ya queda configurada la comunicación entre la pantalla HMI y cada uno de los autómatas.

6.9 Diseño de la pantalla HMI con NB-Designer

Una vez configurada la conexión entre la pantalla HMI y los autómatas, se ha comenzado con el diseño de cada una de las ventanas que formarán parte de la interfaz de la pantalla táctil, con la que se controlará el sistema de mecanizado.

En este apartado se detalla cómo se han diseñado cada una de las ventanas, al igual que las distintas funciones que se le han añadido al diseño para completar la interfaz del sistema, como la introducción de una contraseña para poder acceder a las distintas pantallas, cambiar el idioma de trabajo o disponer de la fecha y hora en cada una de las ventanas.

6.9.1 Ventana de inicio

La ventana de inicio tiene como función principal, facilitar al usuario la selección del modo de trabajo al que quiera acceder en un momento dado. Por ello, se han creado tres botones de función, uno para cada modo de trabajo. Además, se ha decidido crear una contraseña que debe ser introducida por el usuario para poder acceder a los mismos, un selector de idiomas basado en las ilustraciones de las banderas correspondientes al idioma a seleccionar, y la configuración de la fecha y hora expuestas

en la parte superior derecha de la ventana. Todo ello rodeando el nombre del sistema de producción que es controlado por esta pantalla: *Centro Neumático de Mecanizado*.

6.9.1.1 Fecha y hora

Para introducir la fecha y la hora a cada una de las ventanas, NB-Designer consta del objeto funcional *Fecha/Hora*. Para hacer uso del mismo se ha accedido a la pestaña *Objetos funcionales avanzados*, de la biblioteca del software, y se ha arrastrado el objeto funcional a la ventana de diseño, abriéndose la siguiente ventana de configuración:

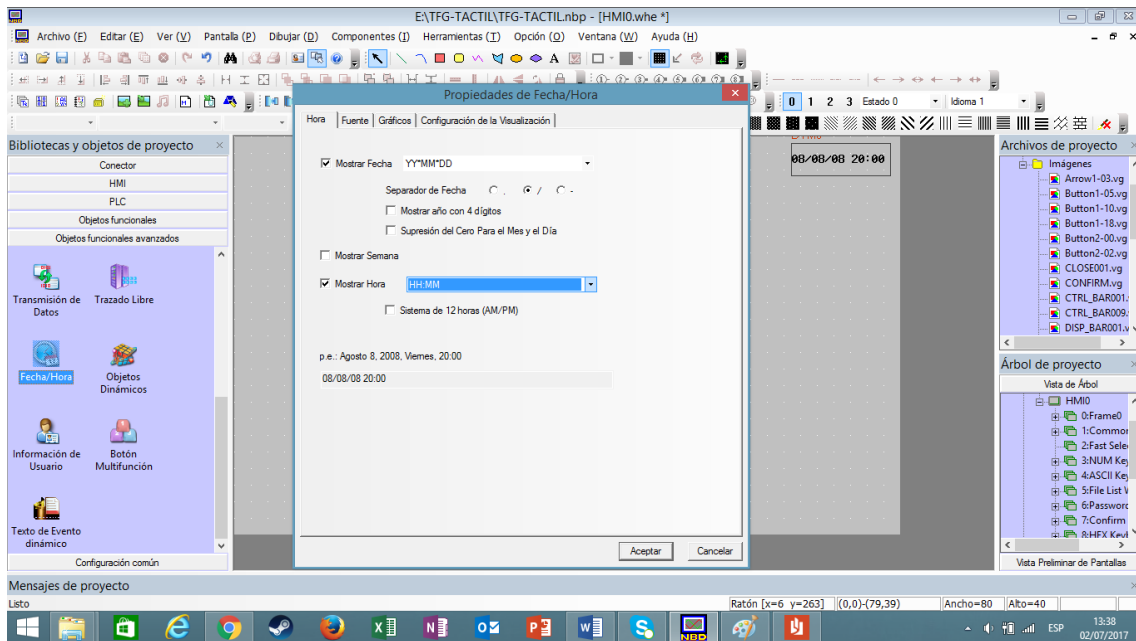


Ilustración 44: Fecha/Hora en NB-Designer (Fuente: Elaboración propio)

En esta ventana se ha marcado el separador de fecha mediante el icono “/” y la opción *Mostrar Hora*, con la estructura *HH:MM*. En cuanto a la configuración de la fecha se ha mantenido la que venía predeterminada.

Este mismo procedimiento se ha repetido para cada una de las ventanas diseñadas en este proyecto.

6.9.1.2 Contraseña

A modo de completar el diseño de la pantalla y simular una situación real en el mundo laboral, se ha configurado una contraseña, que sería impuesta por la empresa en cuestión, para que los trabajadores pudieran acceder a las funciones de la pantalla HMI.

Para el diseño de esta contraseña, en primer lugar, se ha accedido al primer desplegable de la ventana de proyectos, que tiene el nombre que se le ha dado al proyecto en este software. A continuación, haciendo doble clic sobre la pantalla HMI que se ha incluido en el área de diseño previamente, se ha accedido a las propiedades de la misma. Dentro de esta ventana, en *Configuración de niveles de seguridad* se ha introducido el valor 2

en el apartado *No. De niveles de seguridad* y posteriormente, en *Contraseña 1* se ha introducido el código *12345*. Este código será la contraseña que pedirá el programa cuando se quiera hacer uso de la pantalla HMI.

El siguiente a paso a dar es declarar un objeto funcional en la ventana de inicio que sea el encargado de pedir la contraseña mencionada anteriormente.

Para ello, desde la pestaña *Objetos funcionales*, del apartado *Biblioteca y objetos funcionales del proyecto*, se ha arrastrado el objeto *Entrada numérica* a la ventana de diseño de la ventana de inicio. Tras seleccionar la opción *Memoria de Sistemas*, en el desplegable *Área/Etiquetas*, se ha seleccionado la opción: *LW9040: Contraseña de nivel de seguridad (2 Words E)*. (Ilustración 45).

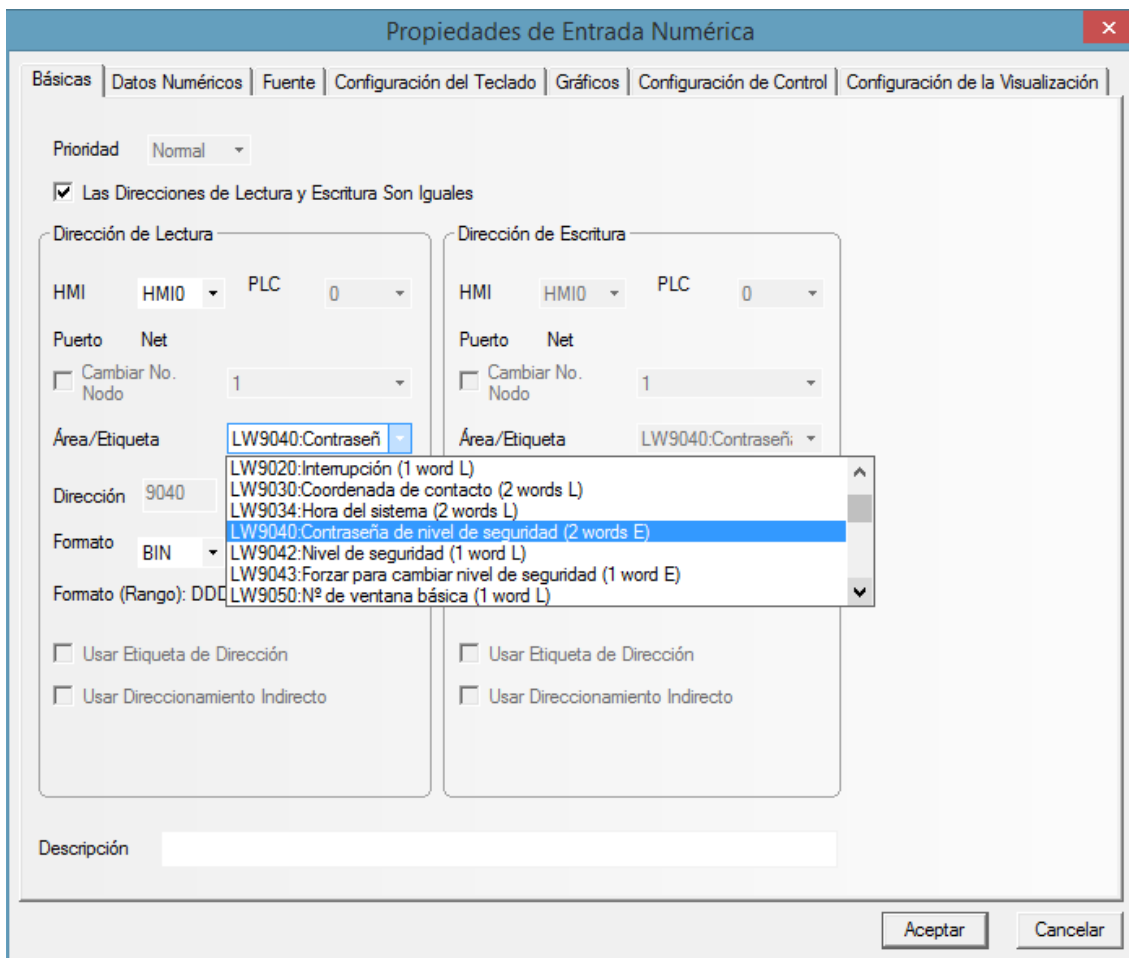


Ilustración 45: Configuración contraseña (1) NB-Designer (Fuente: Elaboración propia)

Posteriormente, desde la pestaña *Datos Numéricos* de esta misma ventana se ha seleccionado *Tipo de Datos* -> *Contraseña*.

Para finalizar, desde la configuración de aquellos botones que se quiera que estén restringidos por la contraseña diseñada, se ha de acceder a la pestaña *Configuración de Control* y marcar la casilla *Nivel de seguridad*. A su derecha se especifica el nivel de seguridad que se quiere imponer, por lo que, como consecuencia de que en este proyecto únicamente se ha configurado una contraseña, se ha seleccionado la opción

“1” de este desplegable. En la siguiente ilustración, se muestra la ventana de configuración de aquellos botones a los que se les ha impuesto la contraseña para poder acceder a ellos.

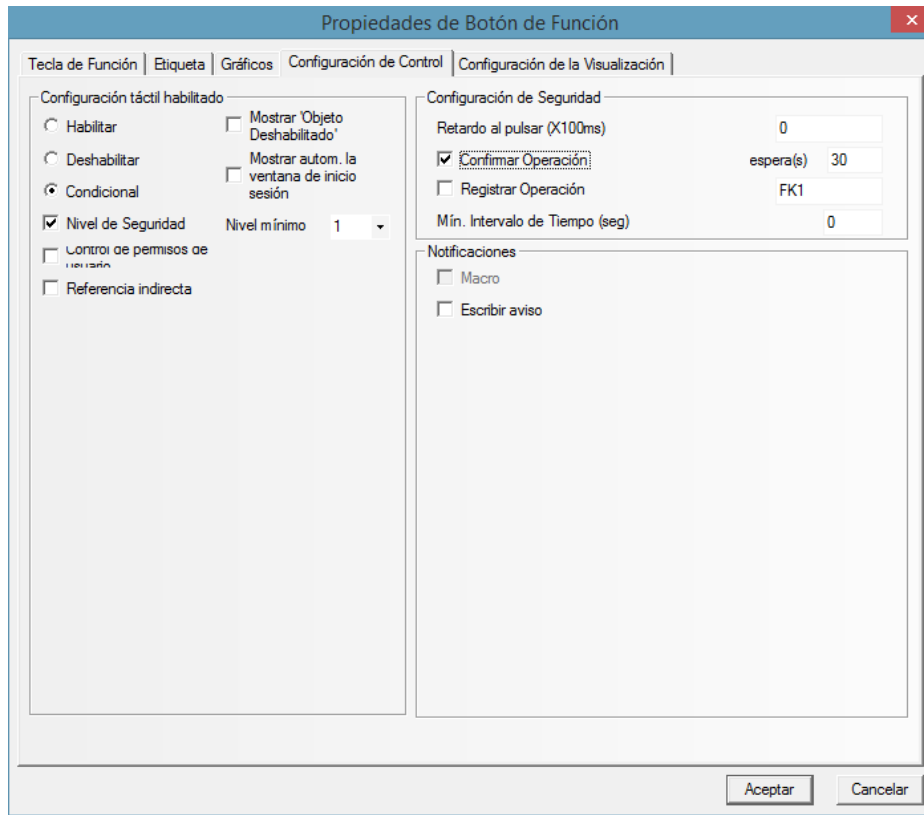


Ilustración 46: Configuración contraseña (2) (Fuente: Elaboración propia)

6.9.1.3 Selector de idiomas

Un dispositivo de estas características, que está diseñado para poder ser controlado por distintos usuarios, debe constar de un selector de idiomas que permita elegir el idioma con el que se quiera trabajar en un momento dado. En el caso de este proyecto, se ha decidido configurar el programa de forma que ofrezca los textos que lo componen tanto en inglés como en español. Para ello, el usuario cuenta, en la ventana principal, con dos banderas con las que se puede seleccionar el idioma deseado.

Para realizar esta configuración se ha comenzado accediendo a la *Biblioteca de Textos* que se encuentra en la pestaña *Configuración común* de la ventana *Bibliotecas y objetos de proyecto*.

Una vez en la biblioteca, haciendo click en *Añadir*, se han introducido los nombres, a modo de identificadores, de las palabras o frases que aparecen en la interfaz de la pantalla. Una vez elegido el nombre, se ha introducido en *lenguaje 1* lo que se ha querido que aparezca en pantalla al estar activo el idioma castellano, y en *lenguaje 2* lo que se quiere que se vea cuando el lenguaje activado es el inglés.

En la siguiente ilustración se muestra la biblioteca de textos con algunos ejemplos de textos que se han empleado en el proyecto.



Ilustración 47: Biblioteca de Textos. Idiomas. (Función: Elaboración propia)

Una vez introducidos los textos en el idioma 1 (español) y el idioma 2 (inglés), es necesario configurar cada uno de los botones que se le quiera introducir un texto que aparezca en pantalla en el idioma seleccionado. Para ello, se accede a las propiedades del botón en cuestión haciendo click sobre el mismo, y dentro de la pestaña *Etiqueta*, se activa la casilla *Usar Biblioteca de Texto*. A continuación, se ha hecho uso del desplegable que se encuentra inmediatamente debajo y se ha seleccionado el nombre correspondiente al texto que se desea asociar a este botón. Este procedimiento se ha realizado para cada uno de los botones a los que se les ha asignado un texto a mostrar en pantalla.

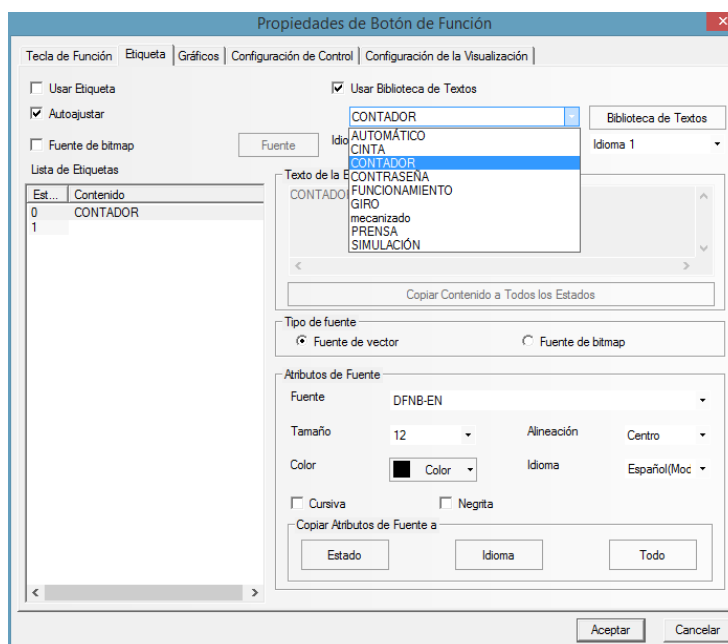


Ilustración 48: Activación Biblioteca de Textos en el botón (Fuente: Elaboración propia)

Por último, es necesario introducir dos objetos que, de cara a su visualización en la interfaz, estén representados por las banderas correspondientes al idioma a seleccionar.

Para ello se ha hecho uso del objeto *Botón de Comando*, dentro de la pestaña *Objetos Funcionales*. Al ser arrastrado a la ventana de diseño se ha seleccionado, en la pestaña *Básicas*, la casilla *Registro del Sistema* y a continuación, en el desplegable de *Áreas/Etiquetas* la opción *LW9130: Modificar dinámica de la biblioteca de textos. Índice (1 word L/E)*.

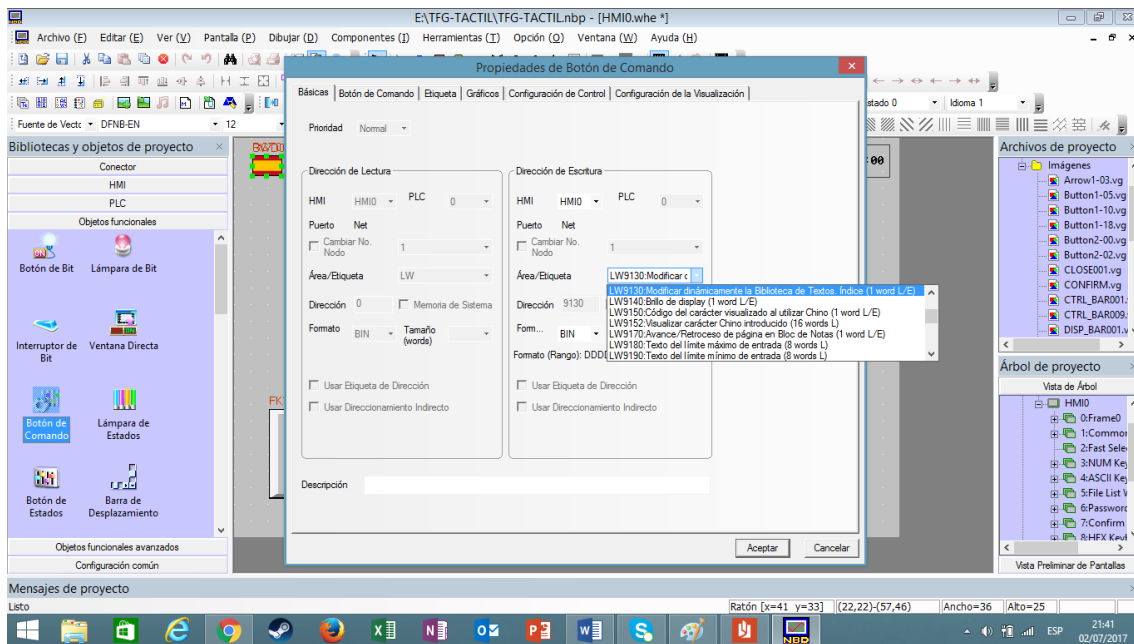


Ilustración 49: Botón de comando / Idioma (Fuente: Elaboración propia)

Para finalizar, en la pestaña *Botón de Comando*, se ha seleccionado el tipo *Establecer valores*, y en la casilla *Establecer* se ha introducido el valor 0 para el botón de comando que se quiere que configure el idioma en español, y el valor 1 en el botón dedicado a poner el idioma de la pantalla en inglés.

Se debe tener en cuenta que, desde la pestaña *Gráficos*, se permite configurar la imagen que representa el botón en la interfaz de la pantalla. En este caso se han seleccionado las banderas correspondientes a cada uno de los idiomas para la representación de los botones de comando.

6.9.1.4 Pulsadores modo de trabajo

El objetivo de la ventana principal gira en torno a los tres botones de función de los que se hablará a continuación, ya que son los encargados de permitir al usuario elegir la modalidad de trabajo con la que quiere interactuar.

Se han declarado tres modos de trabajo diferentes, tal como se mencionó en el apartado *6.1 Funcionamiento de la maqueta del centro neumático*, por lo que han sido necesarios

tres botones de función diferentes, que guíen al usuario a la ventana correspondiente al modo de trabajo seleccionado.

Previo a configurar los botones de función que nos lleven a las distintas ventanas de modo de trabajo, es necesario crear dichas ventanas. Para ello, desde *Árbol de proyecto* -> *Vista de árbol*, y haciendo click derecho sobre *HMIO*, se han añadido tres ventanas nuevas a las que se les ha dado los nombres de *Automático*, *Manual* y *Contador*.

Una vez creadas las ventanas que se van a emplear, el siguiente paso ha sido introducir los botones de función en la ventana principal, para ello se ha accedido a la pestaña *Objetos funcionales avanzados* y se ha arrastrado el elemento *Botón de Función* a la ventana de diseño. Dentro de las propiedades del botón de función se ha activado la casilla *Cambiar ventana*, en el desplegable que se encuentra a su lado se ha mantenido *Cambiar pantalla* y a su derecha, con el uso del otro desplegable, se selecciona la ventana a la que se quiere acceder con el uso de este botón.

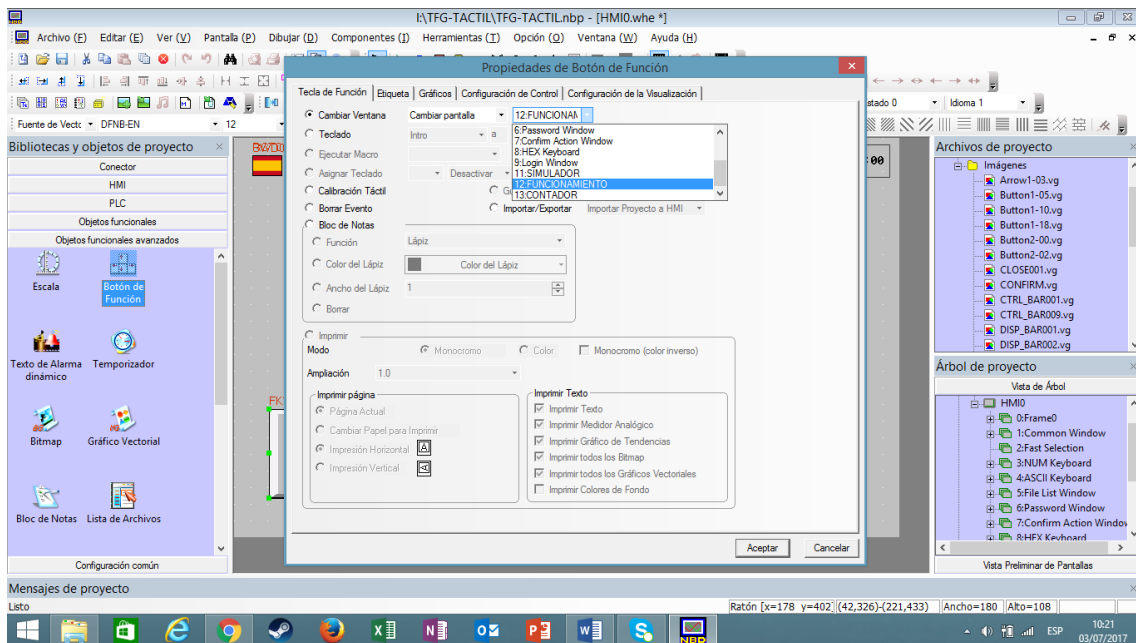


Ilustración 50: Propiedades Botón de Función, NB-Designer (Fuente: Elaboración propia)

Por lo tanto, se han creado tres botones de función, cada uno de ellos configurado de forma que lleven al usuario a cada una de las ventanas creadas.

Finalmente, con los distintos elementos mencionados en estos subapartados, se ha diseñado una ventana de inicio con una interfaz tal como la mostrada en la siguiente ilustración:



Ilustración 51: Pantalla principal HMI (Fuente: Elaboración propia)

6.9.2 Ventana de funcionamiento automático

Tanto en esta ventana, como en las próximas que se mencionarán, se han organizado los interruptores y las lámparas en un orden tal que simulen su distribución en el centro de mecanizado a controlar. En la *Ilustración 52* se puede contemplar que, con la ayuda de algunas herramientas de dibujo y la distribución de los objetos funcionales, se simula una vista en alzado del centro neumático. Esto ayuda al trabajador a la hora de interpretar la interfaz de la pantalla con la que debe operar.

La pantalla de funcionamiento automático consta de una lámpara de bit por cada actuador y cada sensor que se encuentran en la maqueta. Además ofrece un interruptor de Bit de arranque (Start) y otro de recolocación, que sitúa el sistema en su estado inicial.

Para introducir los interruptores, se ha accedido a la pestaña de *Objetos funcionales* y se ha arrastrado el icono *Interruptor de Bit* al lugar deseado de la ventana de diseño. En la pestaña *Básicas* de las propiedades del interruptor, dentro de la sección *Área/Etiqueta* se ha introducido el área I.B y las direcciones 3.1 y 3.0 para los botones Start y Est In respectivamente. Estas direcciones vienen dadas por las direcciones empleadas en el software de programación. De esta forma, al poner a 1 estos interruptores con la pantalla HMI, el autómata activará las variables I3.1 o I3.0 de la programación, actuando así sobre el comportamiento de la maqueta.

Par su correcto funcionamiento, ha sido necesario ir a la pestaña *Conmutador de bit* y seleccionar el tipo *Alternó*.

En el caso de las lámparas de Bit, su única función es iluminarse en verde (ponerse a 1) cuando el autómata detecte que la variable asociada a la lámpara está activa. Por ello, para añadirlas a la ventana de funcionamiento automático, únicamente ha de ser arrastrado el icono *Lámpara de Bit* al lugar deseado de la ventana de diseño, y en la

pestaña *Básicas*, de las propiedades de la lámpara, seleccionar el área y la dirección correspondiente.

En el caso de los sensores se ha seleccionado en *Área/Etiqueta* el área *I.B*, y para los actuadores el área *Q.B*. En cuanto a las direcciones, se ha accedido a la tabla de variables del software de programación y comprobado las direcciones de cada uno de los sensores y actuadores que se han querido reflejar en la pantalla para posteriormente introducirlos en la sección *Direcciones* de las propiedades de cada objeto funcional.

Por último, es necesario que la pantalla facilite al usuario un pulsador que le permita volver a la pantalla principal por si se quiere cambiar de modo de trabajo de nuevo. Para ello se ha introducido un *Objeto de Función*, tal como los empleados para la selección de modo de trabajo, pero en este lugar, al configurarlo, se ha seleccionado la ventana de inicio como el destino al que se accede al pulsarlo. Se ha representado con una flecha de retroceso ubicada en la parte inferior izquierda de la pantalla.

En la siguiente ilustración se puede contemplar la interfaz de la ventana de funcionamiento automático y la distribución seleccionada para cada uno de sus componentes.

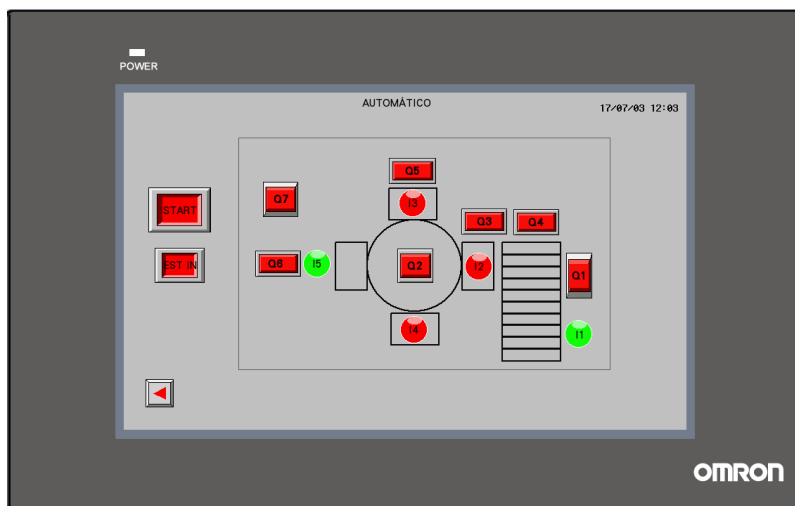


Ilustración 52: Pantalla de modo automático HMI (Fuente: Elaboración propia)

6.9.3 Ventana de funcionamiento manual

Se trata de dos ventanas visualmente muy similares, pero que tienen diferentes funciones de trabajo. La principal diferencia entre esta ventana y la de modo de funcionamiento automático, es que los objetos funcionales que representan los actuadores de la maqueta, en lugar de ser lámparas de Bit, pasan a ser interruptores de Bit. El motivo de este cambio se debe a que, con esta pantalla, los objetos de función que representan los actuadores de la maqueta no tienen el objetivo de informar sobre la activación de los mismo, como en el caso de la anterior pantalla, sino que su función es activar estos actuadores para hacerlos trabajar de forma manual.

La configuración de estos interruptores se ha efectuado de la misma manera que los interruptores de arranque y recolocación mencionados en el apartado anterior. En este

caso se han utilizado las direcciones correspondientes a los actuadores que, tras hacer uso de su interruptor correspondiente, han de ponerse en marcha en la maqueta del proceso de mecanizado.

Es importante recordar que se ha de seleccionar el tipo *Altern* en la pestaña *Conmutador de bit*.

Por último, de la misma forma que en la pantalla de modo automático, se ha añadido en la parte inferior izquierda, un objeto de funcionamiento que permita al usuario volver a la pantalla de inicio.

La interfaz de la ventana de funcionamiento manual ha quedado diseñada como se muestra en la *Ilustración 53*.

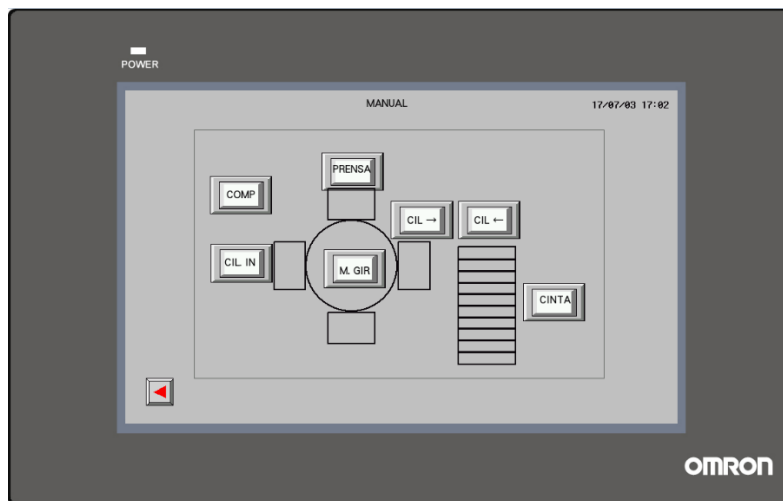


Ilustración 53: Pantalla de modo manual HMI (Fuente: Elaboración propia)

6.9.4 Ventana de funcionamiento en modo contador

Esta ventana de trabajo está diseñada con una estructura prácticamente exacta a la de la ventana de trabajo automático, con la única diferencia de que posee cinco interruptores de más. Estos cinco interruptores son los que utilizará el usuario para señalar al autómata el número de piezas que se quiere mecanizar antes de que se produzca una parada automática del proceso.

Para introducir estos pulsadores también se han empleado interruptores de Bit de tipo *Altern*. Configurados con un área del tipo I.B y con unas direcciones tales que activan los contadores correspondientes en el código de programación. Cada pulsador activará un contador dado, el cual está previamente programado para ponerse a 1 cuando se haya mecanizado el número de piezas que le corresponde. Al ponerse el contador a 1, avisa al programa de que debe detenerse.

Por todo ello, existen tantos contadores como interruptores dedicados al conteo, cinco, y se han programado para que se activen con los valores 100, 75, 50, 25 y 2. A los interruptores enfocados a esta función se les ha configurado con las direcciones 4.1, 4.2, 4.3, 4.4 y 4.5 respectivamente.

A continuación, se muestra como ha quedado el diseño de la interfaz de la pantalla Contador.

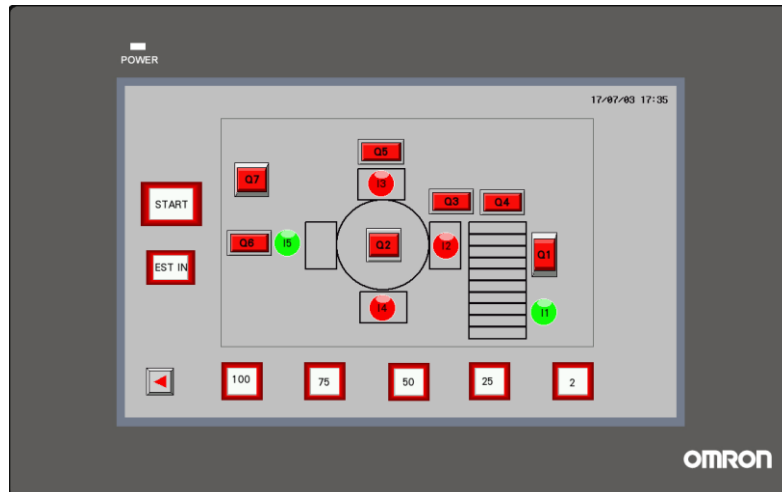


Ilustración 54: Pantalla de modo contador HMI (Fuente: Elaboración propia)

6.10 Transferir en NB-Designer

Una vez diseñadas todas las pantallas que componen la interfaz de la pantalla HMI, y configurados todos los elementos que se encuentran en ellas, únicamente falta transferir el programa a la pantalla HMI de Omron. Al hacerlo, la maqueta estará lista para funcionar según el autómata con el que se encuentre comunicada, manteniendo un control sobre su funcionamiento a través de la pantalla HMI.

NB-Designer, al igual que los softwares de programación, cuenta con un comando de compilar y otro de transferir el programa, que se encuentran en la barra de herramientas ubicado en la parte superior. Haciendo click en compilar y acto seguido en transferir, se accede a la ventana que se muestra en segundo plano en la *Ilustración 55*. Puestos en esta ventana, se ha seleccionado la pantalla HMI0 y se ha seleccionado la opción *Transferir*. El software tardará un tiempo en cargar el programa, pero al finalizar la pantalla HMI ya estará lista para ser utilizada.

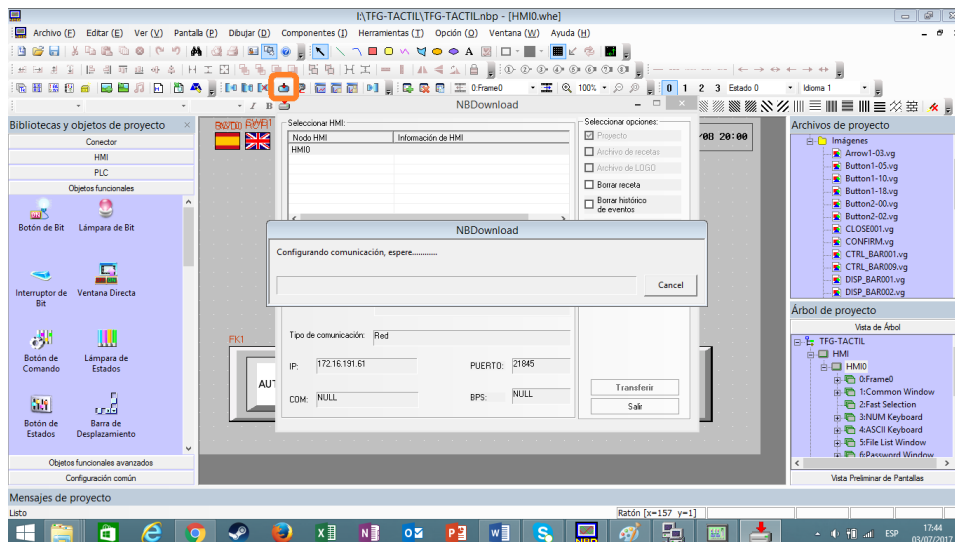


Ilustración 55: Transferir programa NB-Designer (Fuente: Elaboración propia)

6.11 Puesta en marcha y verificación

Una vez finalizada la configuración de cada uno de los softwares empleados, se ha establecido la conexión entre la maqueta, los autómatas, cada uno por separado, y la pantalla HMI. Establecida la comunicación, únicamente faltaba transferir cada uno de los programas a su dispositivo correspondiente y poner en marcha la fase de pruebas.

Como es común en programaciones de este tipo, no todo ha marchado a la perfección en el primer intento. Ha sido necesario retocar algunos parámetros tanto en la programación del código, como en la configuración de algunos elementos de la pantalla HMI, pero finalmente se ha logrado que la maqueta trabaje correctamente en sus tres modos de funcionamiento, y que el usuario pueda interactuar sobre la misma, tal como se había planeado en los objetivos de este proyecto.

7 POSIBLES MEJORAS

Existen alternativas y mejoras que pueden hacer de este proyecto un proyecto más eficiente y cómodo a la hora de trabajar con él. En este apartado se citarán alguna de las mejoras que se han considerado de interés para un mejor funcionamiento del proceso.

Por un lado, con el fin de realizar una comunicación más intuitiva entre la pantalla HMI y el trabajador, se ha pensado en que resultaría de gran ayuda el diseño de un SCADA capaz de hacer un seguimiento al completo del avance de cada una de las piezas por el proceso. En este proyecto se puede hacer ese seguimiento, únicamente sabiendo cuándo la pieza ha llegado a cada uno de los sensores, pero resultaría interesante que se supiera su ubicación en todo momento, con una representación del proceso en la pantalla HMI.

Por otro lado, la optimización del tiempo de espera entre las distintas tareas que se realizan en el mecanizado siempre resulta una clara mejora en un proceso de producción. En este proyecto se ha intentado ajustar los temporizadores de forma que el tiempo de espera entre las distintas tareas sea lo más corto posible, pero siempre dejando un margen de seguridad que garantice que el mecanizado se realice correctamente, evitando, por ejemplo, que una pieza comience a rotar en la mesa de giro, sin haber sido prensada el tiempo necesario.

Un estudio sobre la optimización de este tiempo de espera ayudaría al ritmo de producción de cualquier proceso industrial.

La programación de un nuevo modo de trabajo podría significar otra mejora a tener en cuenta. Por ejemplo, un modo de trabajo que, en lugar de seleccionar dentro las opciones ofrecidas por el programa, el número de piezas a mecanizar, que este número sea introducido en pantalla por el propio trabajador. De esta forma se puede mecanizar un número más específico de piezas.

Por último, el hecho de emplear un sistema de producción de estas dimensiones, únicamente para un tipo de pieza con unas dimensiones dadas, restringe a gran escala la funcionalidad de este proceso. Una solución a este problema, que permita poder trabajar con piezas de distintas dimensiones, sería poder seleccionar, desde la pantalla HMI, el tipo de pieza con el que se va a trabajar. En función del modelo de pieza seleccionado, los elementos que componen el proceso de mecanizado se adaptarían para poder realizar el mecanizado de dicho modelo de pieza. En caso de tratarse de una pieza de menor tamaño, las plataformas de la mesa giratoria formarían un cuadrado de menor tamaño, y la prensa tendría un recorrido más largo de funcionamiento.

8 PLIEGO DE CONDICIONES

En el presente apartado se trata de reflejar las diferentes condiciones técnicas requeridas en relación a los dispositivos empleados para la ejecución de este proyecto, además de los pasos que se deben llevar a cabo para el montaje del sistema al completo.

8.1 Elaboración del proceso

Es importante tener en cuenta que este proyecto deber ser dirigido y realizado por un personal cualificado, que posea con los conocimientos necesarios para afrontarlo y con capacidad de reacción ante posibles problemas técnicos que puedan surgir durante el montaje, la conexión entre dispositivos, la programación o la puesta en marcha del sistema.

El orden de trabajo a seguir para para la realización del proyecto se rige por la sucesión de pasos expuestos a continuación:

- 1 Montaje de la maqueta del centro neumático de mecanizado.
- 2 Montaje de cada uno de los autómatas, junto a los módulos correspondientes, citados previamente en la memoria.
- 3 Conexión entre la maqueta y los distintos autómatas.
- 4 Interconexión y visualizado de entradas y salidas de la maqueta con cada uno de los PLC's.
- 5 Elaboración del código de programación realizado a mano.
- 6 Implementación de dicho código en los softwares TIA Portal V13 y SoMachine.
- 7 Conexión entres loa autómatas y el PC empleado.
- 8 Programación de los autómatas al transferirles el código.
- 9 Comprobación del correcto funcionamiento del código empleado con la maqueta.
- 10 Diseño de la interfaz de la pantalla HMI con NB-Designer para comunicarse con cada uno de los programas.
- 11 Conexión entre el PC y la HMI de Omron.
- 12 Transferencia del diseño realizado a la pantalla.

- 13 Conexión de todo el montaje PLC-PC-HMI.
- 14 Comprobación del correcto funcionamiento del diseño de la pantalla HMI junto a cada uno de los autómatas.

8.2 Características del material empleado

Para poner en funcionamiento el proceso que se ha programado durante este proyecto, se recomienda el uso exclusivo de los materiales expuestos a lo largo de esta memoria.

Si se utilizan elementos de marcas que no se han mencionado en la memoria no se garantiza el correcto funcionamiento del proceso de mecanizado, ya que la programación empleada se ha llegado a cabo exclusivamente para las marcas de las que se ha hecho mención. Por ello, el director del proyecto no se hace responsable del mal funcionamiento del proceso en caso de que se emplee material que no se encuentra expuesto en esta memoria.

No debería presentar un problema el hecho de encontrar recambios de los materiales empleados en este proyecto, ya que se tratan de marcas, tanto de los autómatas como de la pantalla, de gran peso en el sector de la automatización industrial.

Las características de los PLC y de la pantalla HMI han sido expuestas en el apartado 4.2 *Elementos empleados* de la memoria.

Se debe tener en cuenta que, para poder realizar la programación de los distintos dispositivos, es necesario un PC que cumpla con unos requisitos mínimos necesarios para el trabajo. En este caso se ha empleado un ordenador de sobremesa del laboratorio donde se ha realizado el proyecto, con el que se ha empleado la programación de los distintos softwares y se ha establecido la comunicación con los tres dispositivos. El PC empleado debe contar con las características mínimas que se muestran a continuación:

Características del Hardware

- Microprocesador Pentium II con velocidad de reloj de 333MHz o superior
- Memoria RAM 256MB
- Disco duro de 2GB o superior
- Conexión a la red
- Ratón y teclado

Características del Software

- Programa TIA Portal V13
- SoMachine V4.1
- NB-Designer
- Sistema operativo Windows 8 o superior

En relación a la maqueta empleada, está formada por los elementos citados en el apartado 4.2.4 Maqueta del proceso de producción. Se trata de un diseño que no cuenta con elementos punzantes o cortantes, por lo que no presenta ningún tipo de riesgo para el encargado de ponerla en funcionamiento.

8.3 Legislación

A continuación, se detallan aquellas legislaciones a tener en cuenta para la elaboración del proyecto expuesto:

- RD 177/2008, de 8 de febrero, por el que se establece el título de Técnico de Instalaciones Eléctricas y Automáticas y se fijan sus enseñanzas mínimas.
- RD 1589/2011, de 4 de noviembre, por el que se establece el Título de Técnico en Mantenimiento Electromecánico y se fijan sus enseñanzas mínimas.
- RD 560/2011, de 20 de abril, por el que se complementa el Catálogo Nacional de Cualificaciones Profesionales, mediante el establecimiento de tres cualificaciones profesionales de la familia profesional electricidad y electrónica.
- RD 7/1988, de 8 de enero, relativo a las exigencias de seguridad del material eléctrico destinado a ser utilizado en determinados límites de tensión.
- EN ISO 164884:2003: Automatización de edificios y sistemas de control.
- Norma IEC-1131 sobre la estandarización de los lenguajes de programación y sobre los diferentes tipos de autómatas programables y sus periféricos.
- Manual de funcionamiento TIA Portal V13.
- Manual de funcionamiento SoMachine V4.1.
- Manual de funcionamiento NB-Designer.
- Catálogo de cada uno de los dispositivos empleados.

9 Conclusión

Una vez finalizado este proyecto y alcanzados los objetivos propuestos, se puede echar la vista atrás y contemplar todos los conocimientos adquiridos a lo largo de su elaboración. La decisión de realizar un proyecto con estas características se ha tomado por la motivación de querer ampliar los conocimientos en el campo de la automatización de procesos industriales y poder trabajar con softwares, como TIA Portal, que hoy en día resultan punteros en el mundo laboral.

Además de afianzar y completar lo aprendido sobre los lenguajes de programación, este proyecto ha generado la necesidad de tener que ampliar los estudios con los que ya se contaba, teniendo que aprender sobre el uso de los softwares empleados, además de partir de cero en la temática de la configuración de las pantallas HMI y las prestaciones que ofrecen para el control de los procesos automatizados.

A pesar de que en este proyecto se haya trabajado con una maqueta de un proceso de mecanizado, para lograr su correcta automatización y control mediante la pantalla HMI se ha actuado de forma muy similar a cómo se haría con un proceso industrial, cubriendo una de las principales motivaciones por las que se ha decidido afrontar un proyecto como el presente en esta memoria.

10 Bibliografía

- YOUTUBE, lista de reproducción “Curso TIA Portal Siemens S7-1200 1/20 – 20/20” en Youtube.
<https://www.youtube.com/watch?v=xFAYA0Q-N9A&list=PLYv8upwzfANzzIToYoQdCWNv8OX7Wtw8G>
 - YOUTUBE, “canal de youtube de Martha Lucia Rodriguez Lopez” en Youtube.
<https://www.youtube.com/channel/UCx28CgjHoITVsaWCJm4sHpA>
 - [1] Pere Ponsa Antoni Granollers (2007). “Diseño y automatización industrial”, Universidad Politécnica de Cataluña. Barcelona.
<https://www.epsevg.upc.edu/hcd/material/lecturas/interfaz.pdf>
 - [2] y [3] R.P. Fernández (2015), Documentos “PolifomaT”, Universidad Politécnica de Valencia. Valencia.
- SIMATIC (2009). “S7-1200 Getting Started del S7-1200”, SIEMENS.
- Schneider Electric (2014). “Modicon M241 Logic Controller Guía de programación”, www.schneider-electric.com.
 - Schneider Electric (2015). “SoMachine Introducción”, www.schneider-electric.com.
 - Schneider Electric (2014). “SoMachine Manual de formación”, www.schneider-electric.com.
 - Omron (2016). “Programmable Terminals, Host connection manual”, Omron.
 - Omron (2012). “Programmable Terminals, NB-Designer Operation manual”, Omron.



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escuela Técnica Superior de Ingeniería del Diseño

GRADO EN INGENIERÍA MECÁNICA

**CONFIGURACIÓN Y DESARROLLO
DE APLICACIÓN SCADA EN
PANTALLA HMI DE OMRON PARA
ACCEDER A DATOS DE AUTÓMATAS
DE SIEMENS Y SCHNEIDER**

B. PRESUPUESTO

AUTOR: Diego Suárez Bethencourt

TUTOR: José Vicente Salcedo Romero de Ávila

CURSO ACADÉMICO: 2016/2017

ÍNDICE PRESUPUESTO

1.	Presupuesto.....	1
1.1	Cuadro de precio nº1: Hardware	1
1.2	Cuadro de precio nº2: Software	2
1.3	Cuadro de precio nº3: Mano de obra	2
1.4	Cuadro de precio nº4: Presupuesto total	3

1.Presupuesto

En el presente documento se procede al desglose del presupuesto de cada uno de los elementos que han sido necesarios para la realizar el proyecto. Tanto elementos materiales como softwares empleados para la programación de cada uno de los autómatas y de la pantalla HMI.

A demás, hay que tener en cuenta la mano de obra requerida para la correcta instalación y programación de todo el sistema.

1.1 Cuadro de precio nº1: Hardware

Se han tomado precios estimados de cada uno de los elementos que cumplen con las características necesarias para la elaboración del proyecto.

HARDWARE			
DESCRIPCIÓN	CANT.	PRECIO/U	IMPORTE
Monitor PC	1	100,00 €	100 €
Ratón	1	10,00 €	10 €
Teclado	1	25,00 €	25 €
Torre del PC	1	400,00 €	400 €
Maqueta del centro de mecanizado	1	917,50 €	917,50 €
PLC S7-1214 AC/DC/RLY	1	361,62 €	361,62 €
Módulo SM 1221 DC (16ID)	1	144,93 €	144,93 €
Módulo SM 1222 RLY (16DO)	1	153,51 €	153,51 €
PLC Modicon M241	1	386,27 €	386,27 €
Módulo de E/S TMC4AI2	1	116,52 €	116,52 €
Módulo de E/S TMC4AQ2	1	135,04 €	135,04 €
Pantalla táctil HMI, Omron, 7 pulg	1	765,55 €	765,55 €

Importe Bruto	Base Imponible	%	I.V.A	
3.515,94 €	3.515,94 €	21	738,35 €	4.254,29 €

1.2 Cuadro de precio nº2: Software

Para poder hacer uso de los softwares mencionados a lo largo de la memoria es necesario contar con la licencia de los mismos, por lo que hay que tener en cuenta el coste de las mismas.

SOFTWARE			
DESCRIPCIÓN	CANT.	PRECIO/U	IMPORTE
SIMATIC STEP 7 - V13	1	427,43 €	427,43 €
SoMachine V4.1	1	-	-
NB-Designer	1	-	-

Importe Bruto	Base Imponible	%	I.V.A	
427,43 €	427,43 €	21	89,76 €	517,19 €

1.3 Cuadro de precio nº3: Mano de obra

El coste de la mano de obra ha sido tomado de los salarios, en función de la categoría profesional, que se encuentran publicados en el Boletín Oficial del Estado.

PRESUPUESTO INGENIERO TÉCNICO			
DESCRIPCIÓN	h	€/h	IMPORTE
Análisis	10	22,50 €	225,00 €
Diseño	20	22,50 €	450,00 €
Implementación	30	22,50 €	675,00 €
Evaluación	10	22,50 €	225,00 €

Importe Bruto	Base Imponible	%	I.V.A	
1.575,00 €	1.575,00 €	21	330,75 €	1.905,75 €

PRESUPUESTO TÉCNICO MONTAJE			
DESCRIPCIÓN	h	€/h	IMPORTE
Instalación	10	10,00 €	100,00 €
Montaje	5	10,00 €	50,00 €
Evaluación	5	10,00 €	50,00 €

Importe Bruto	Base Imponible	%	I.V.A	
200,00 €	200,00 €	21	42,00 €	242,00 €

1.4 Cuadro de precio nº4: Presupuesto total

RESUMEN	
DESCRIPCIÓN	IMPORTE
Hardware	4.254,29 €
Software	517,19 €
Presupuesto Ingeniero Técnico	1.905,75 €
Presupuesto Técnico Montaje	242,00 €
TOTAL	6.919,23 €



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



GRADO EN INGENIERÍA MECÁNICA

**CONFIGURACIÓN Y DESARROLLO
DE APLICACIÓN SCADA EN
PANTALLA HMI DE OMRON PARA
ACCEDER A DATOS DE AUTÓMATAS
DE SIEMENS Y SCHNEIDER**

C. ANEXO. CÓDIGO DE PROGRAMACIÓN

AUTOR: Diego Suárez Bethencourt

TUTOR: José Vicente Salcedo Romero de Ávila

CURSO ACADÉMICO: 2016/2017

ÍNDICE ANEXO

1.	Códigos de programación.....	1
1.1	Diagrama funcional secuencial (Grafcet).....	1
1.2	Lenguaje Ladder (Diagrama de contactos)	5

1. Códigos de programación

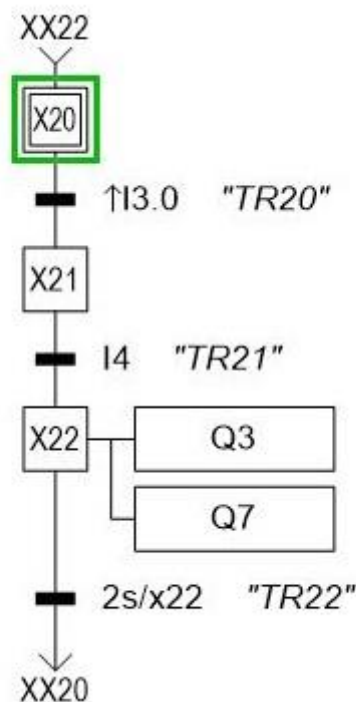
El presente anexo tiene como objetivo mostrar el código empleado para la programación de cada uno de los autómatas. Se ha dividido en dos apartados, uno para cada uno de los lenguajes empleados durante el desarrollo del proyecto. Se recuerda que en primer lugar se ha diseñado el código con un diagrama funcional secuencial para, posteriormente, traducirlo al lenguaje Ladder, también llamado, diagrama de contactos.

Respetando el orden de diseño de los mismos, se muestran a continuación cada uno de los códigos mencionados.

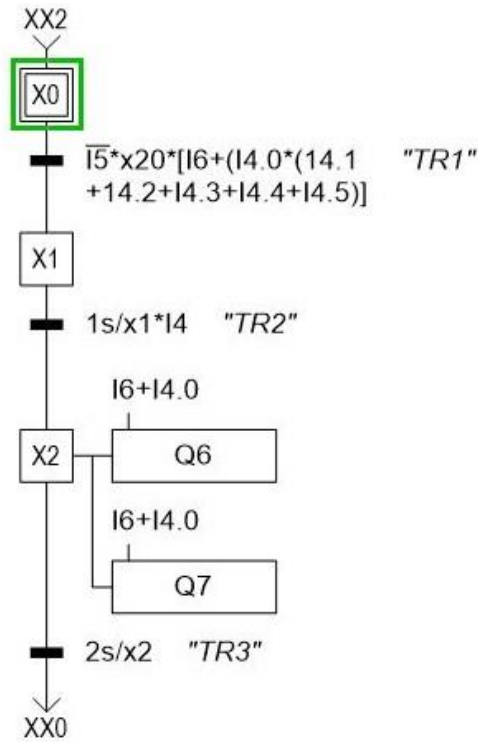
1.1. Diagrama funcional secuencial (Grafcet)

Para el desarrollo de este diagrama se han diseñado diferentes bloques de Grafcet capacitados de trabajar de forma simultánea.

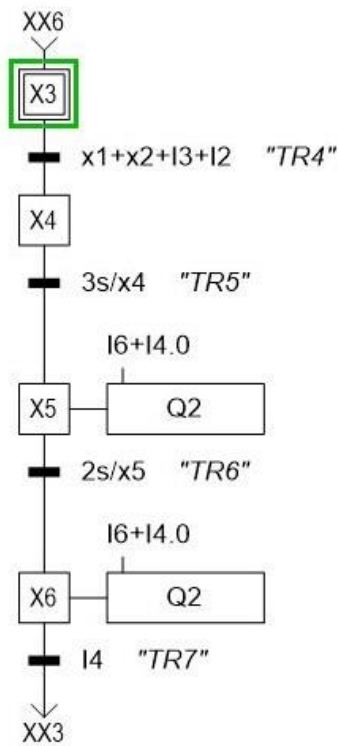
Grafcet Recolocación al estado inicial



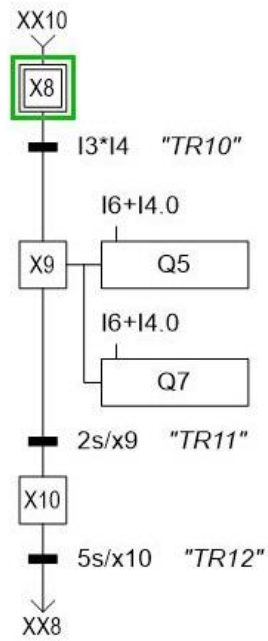
Grafcet Primer empuje



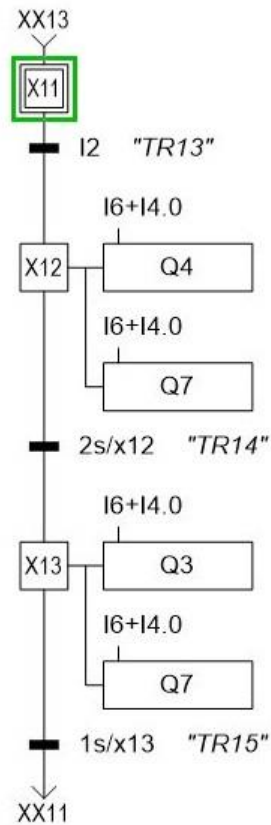
Grafcet Mesa giratoria



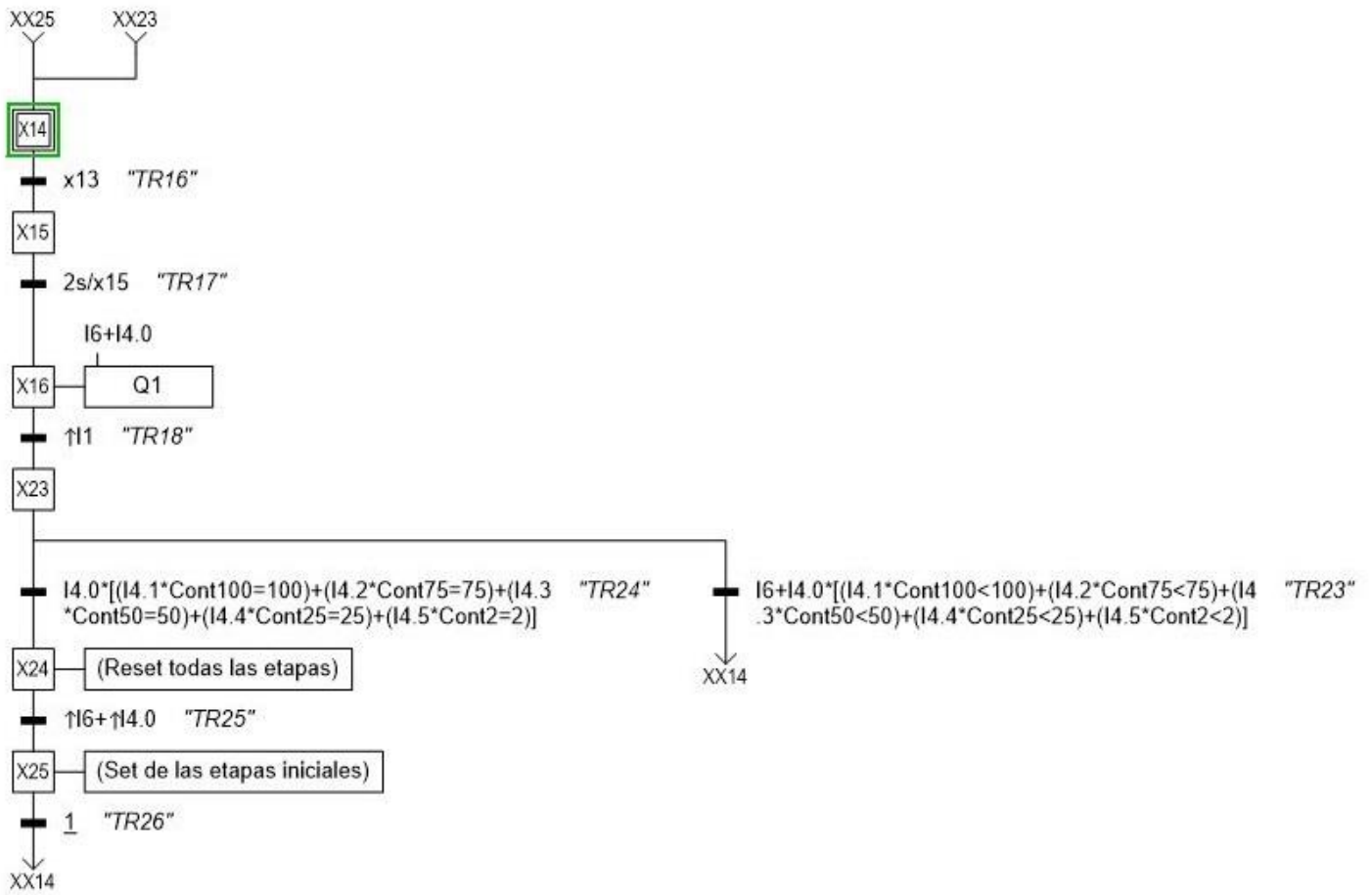
Grafcet Prensa



Grafcet Cilindro doble efecto

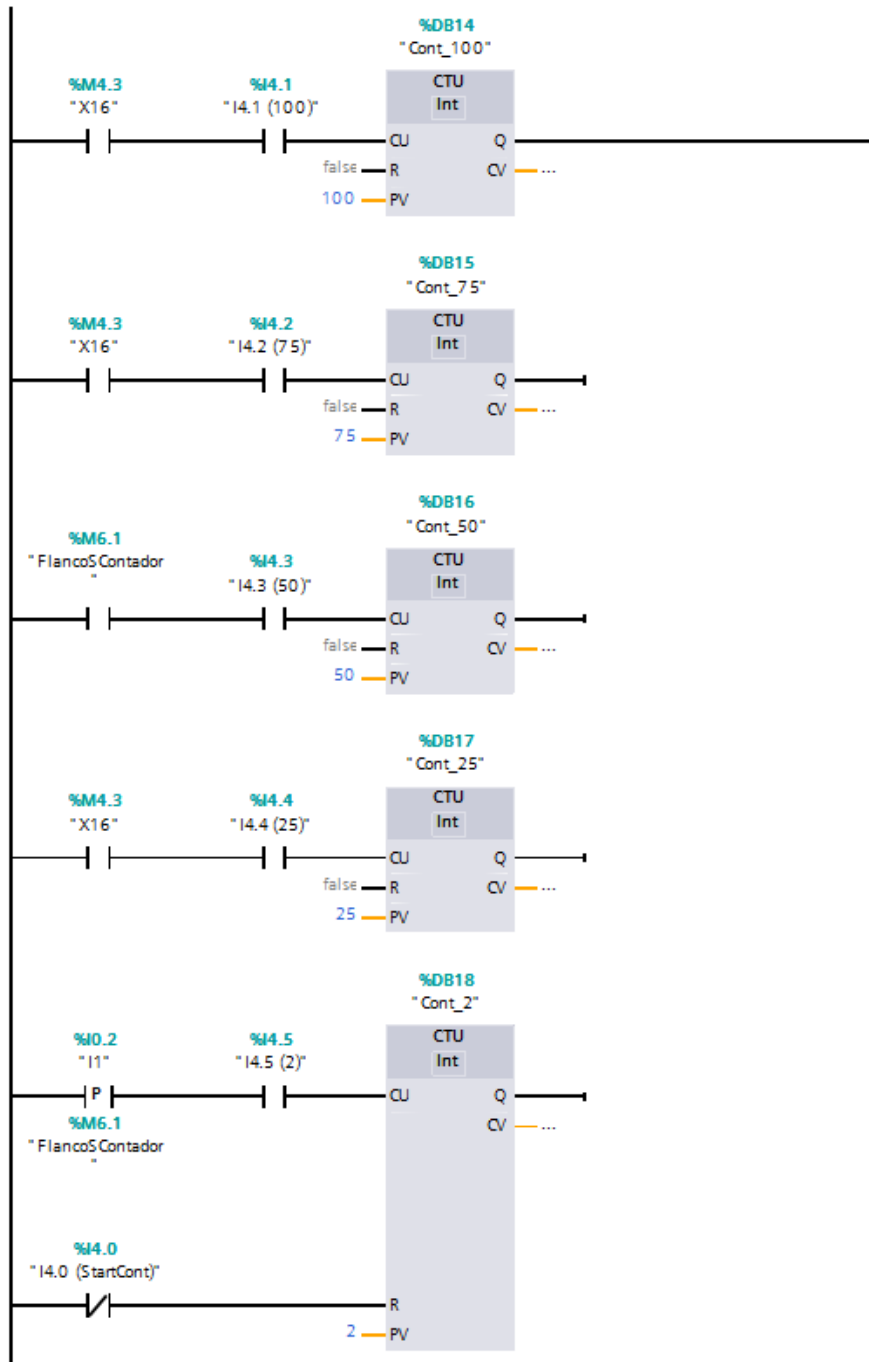


Grafcet Cinta transportadora

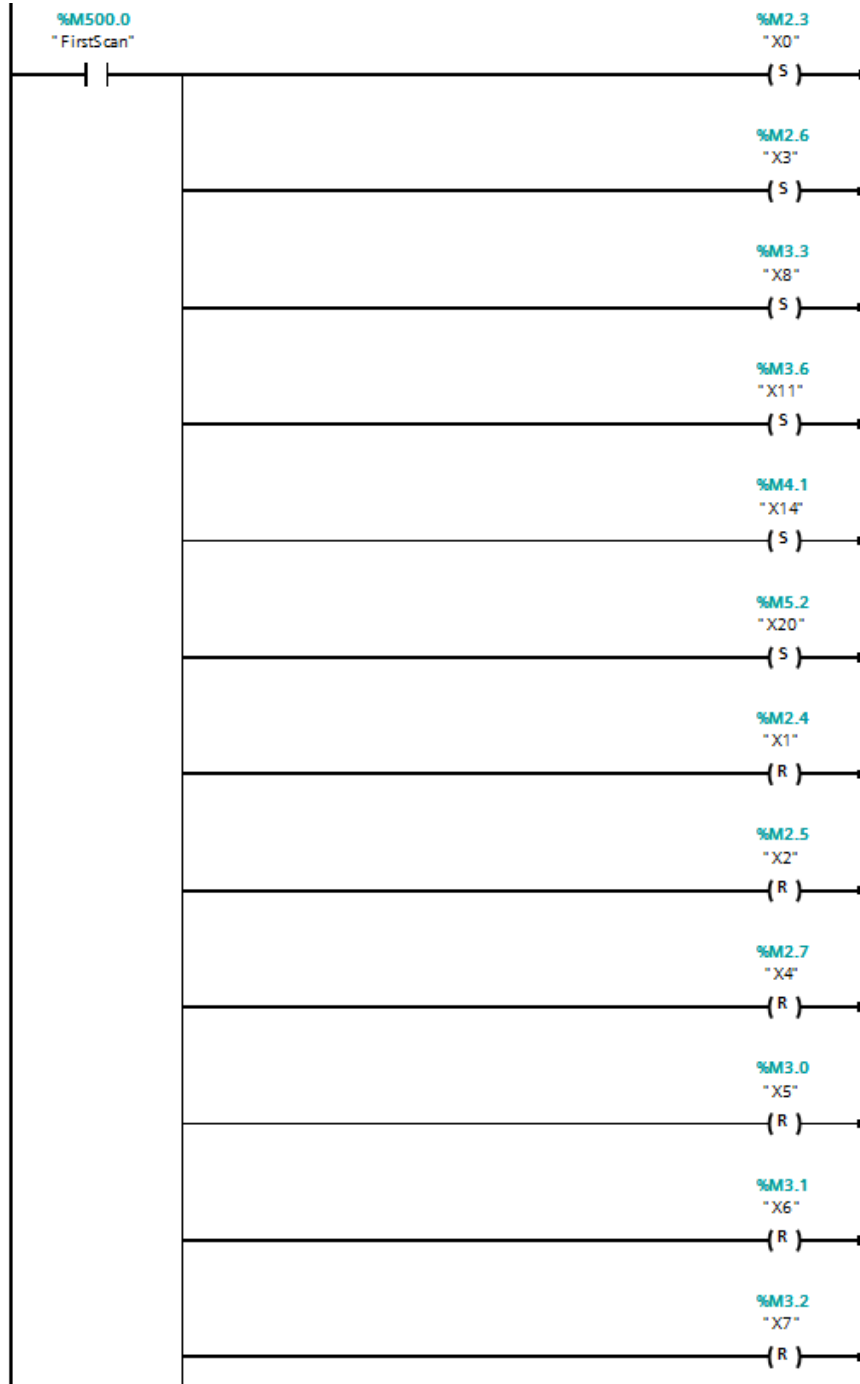


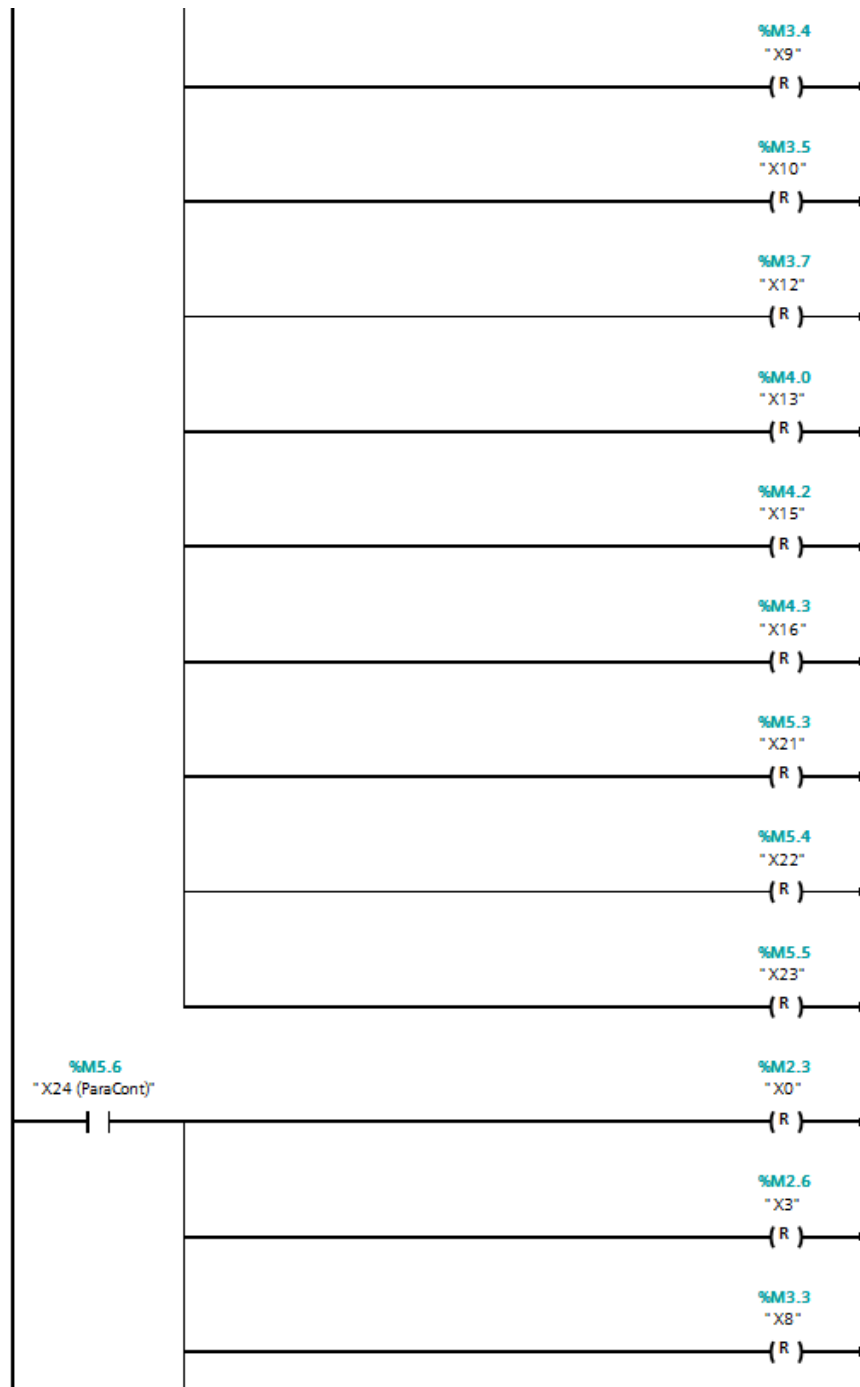
1.2 Lenguaje Ladder (Diagrama de contactos)

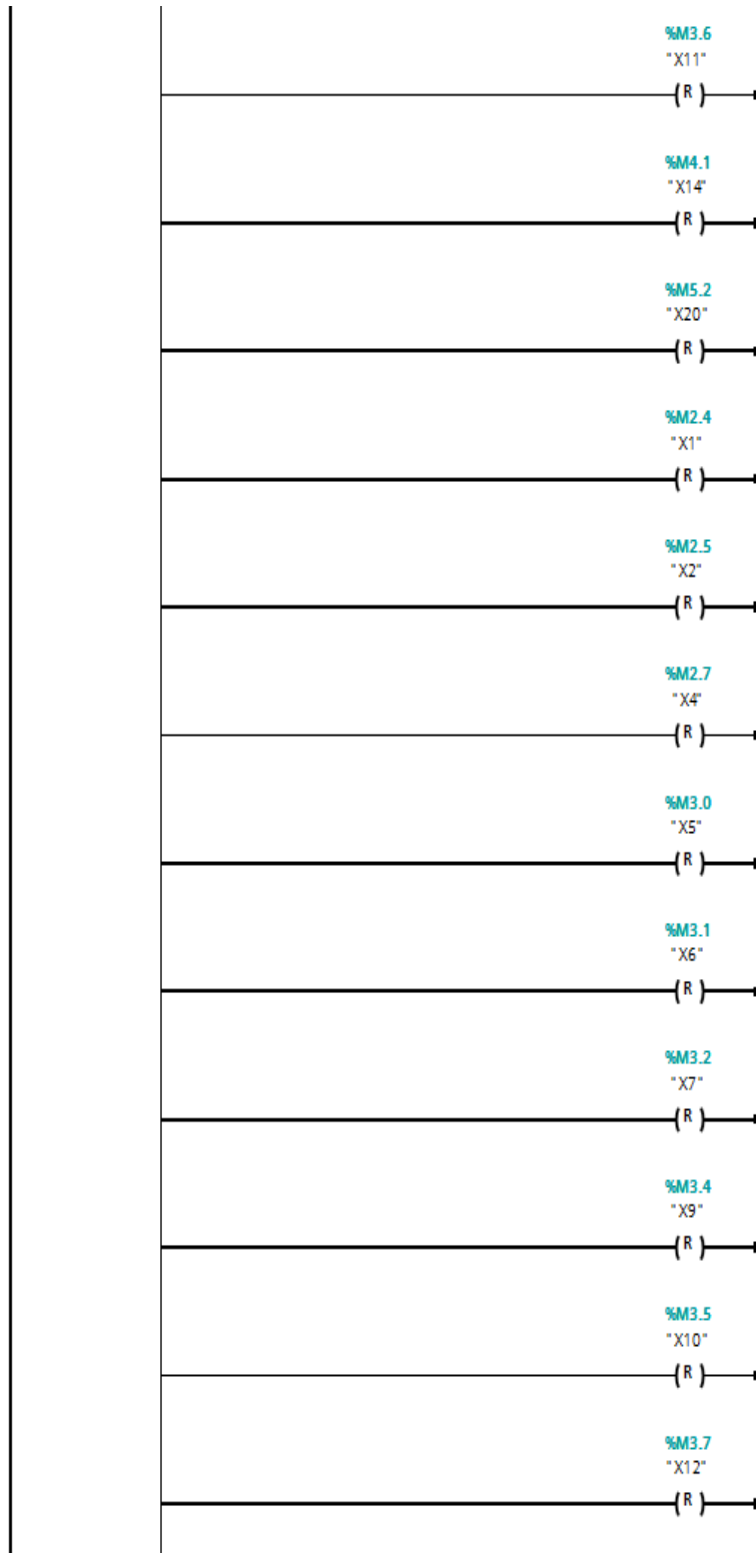
Segmento 1: Contadores

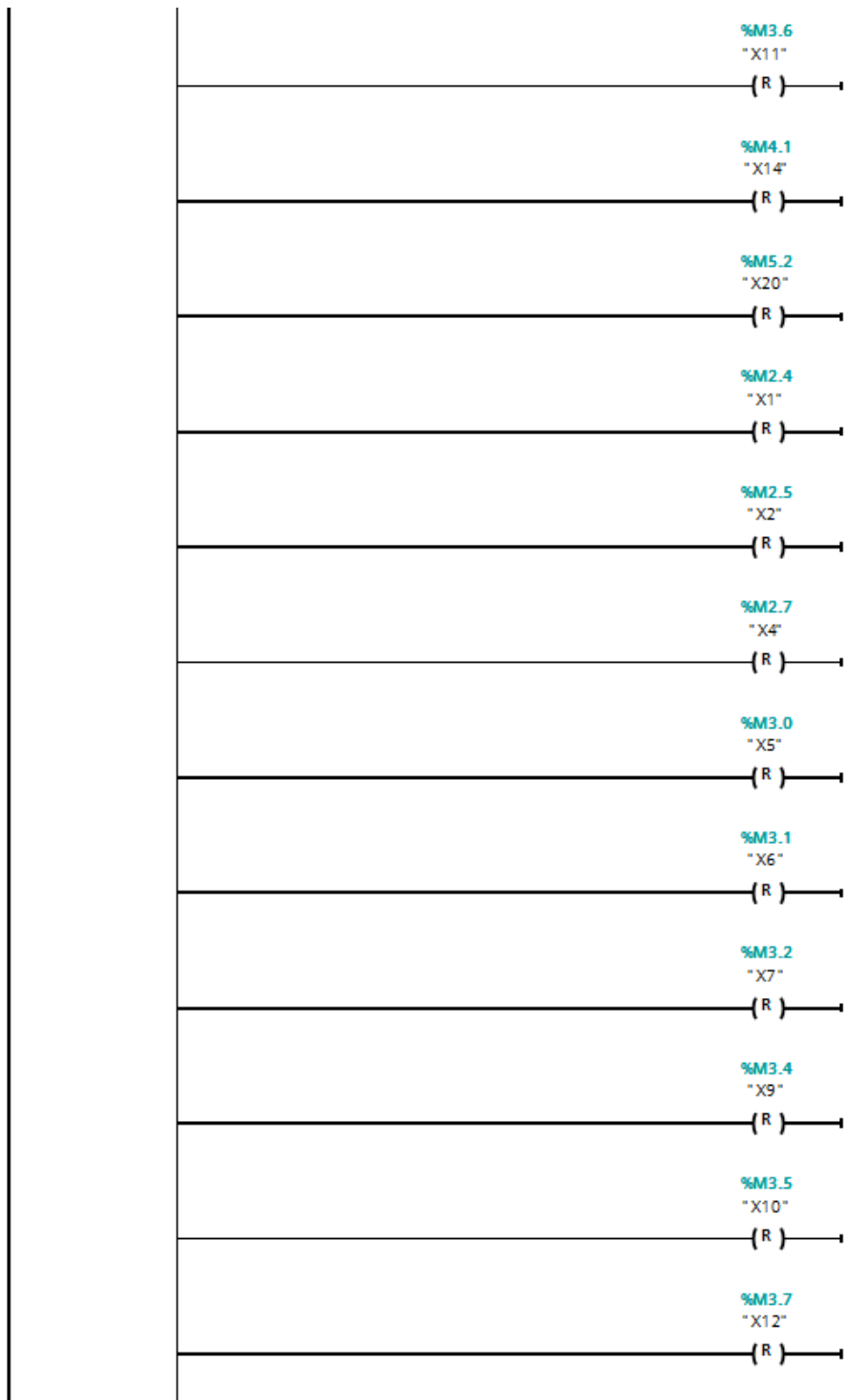


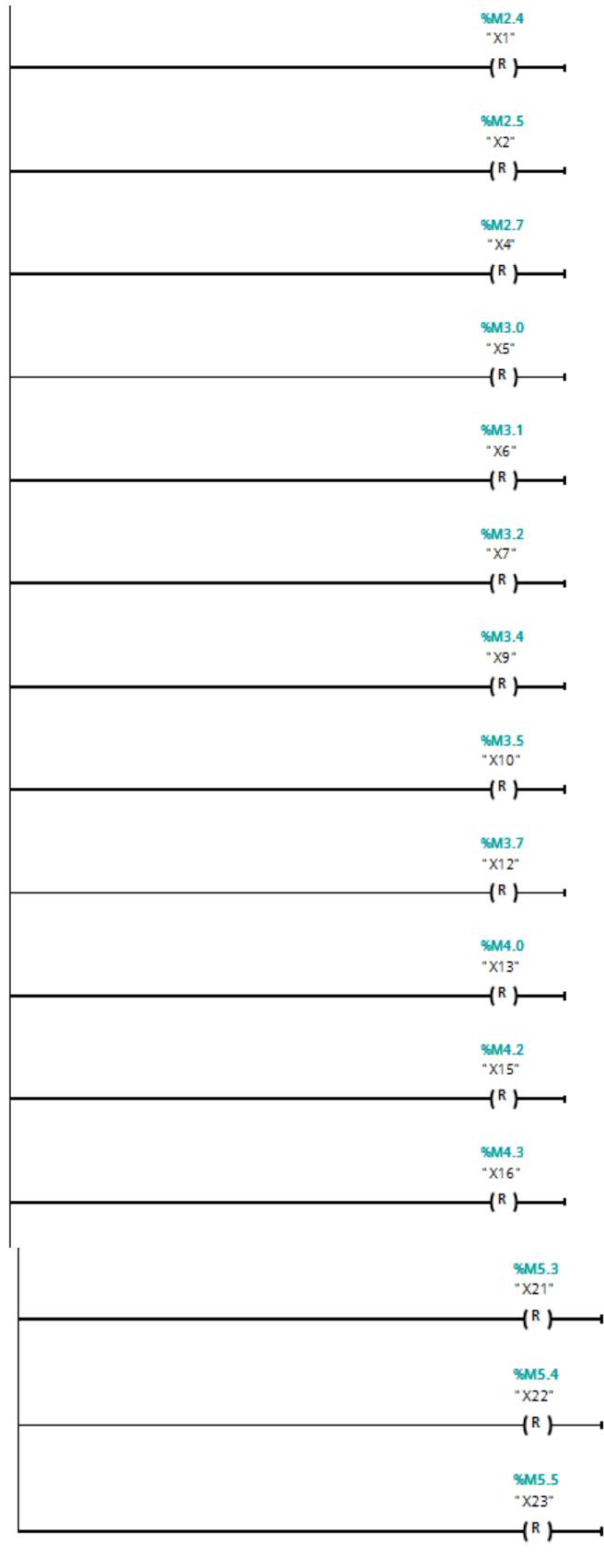
Segmento 2: Estado inicial y Reset



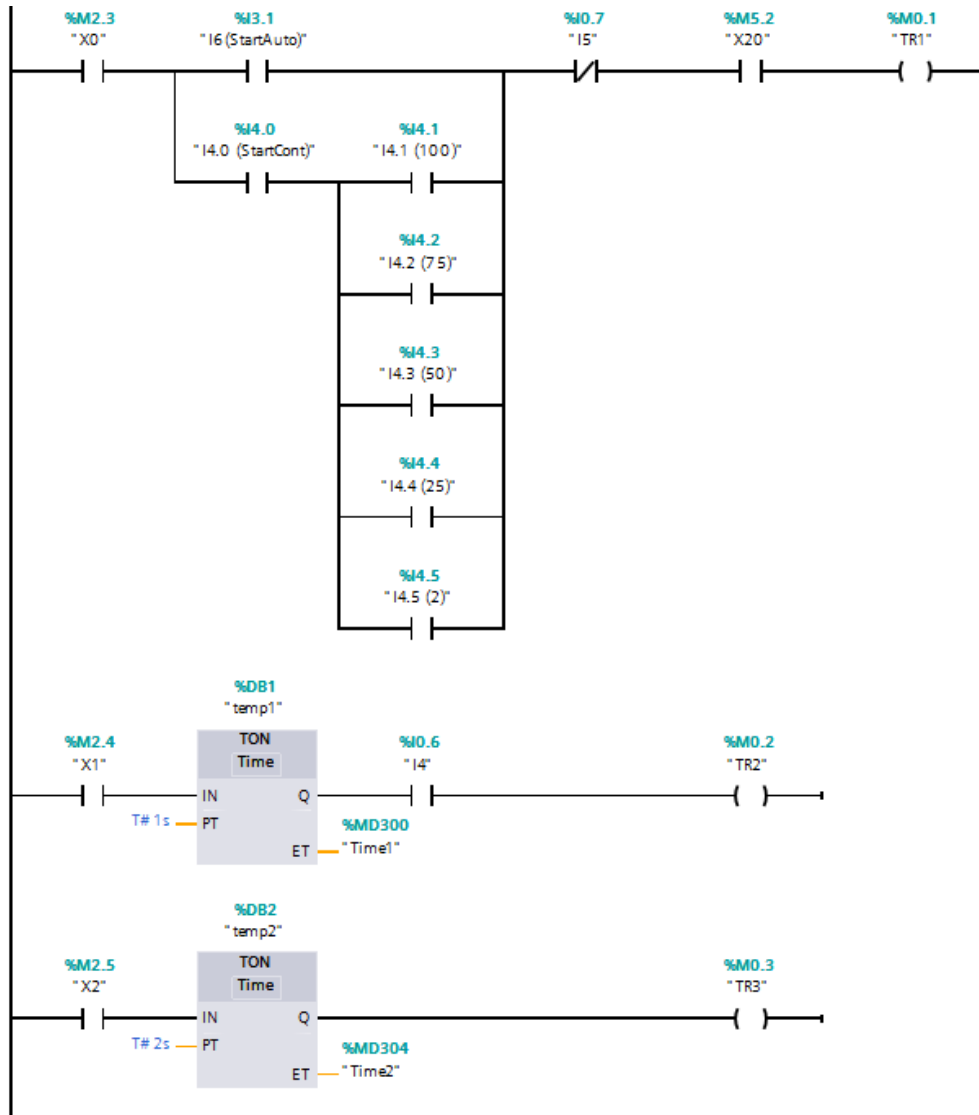


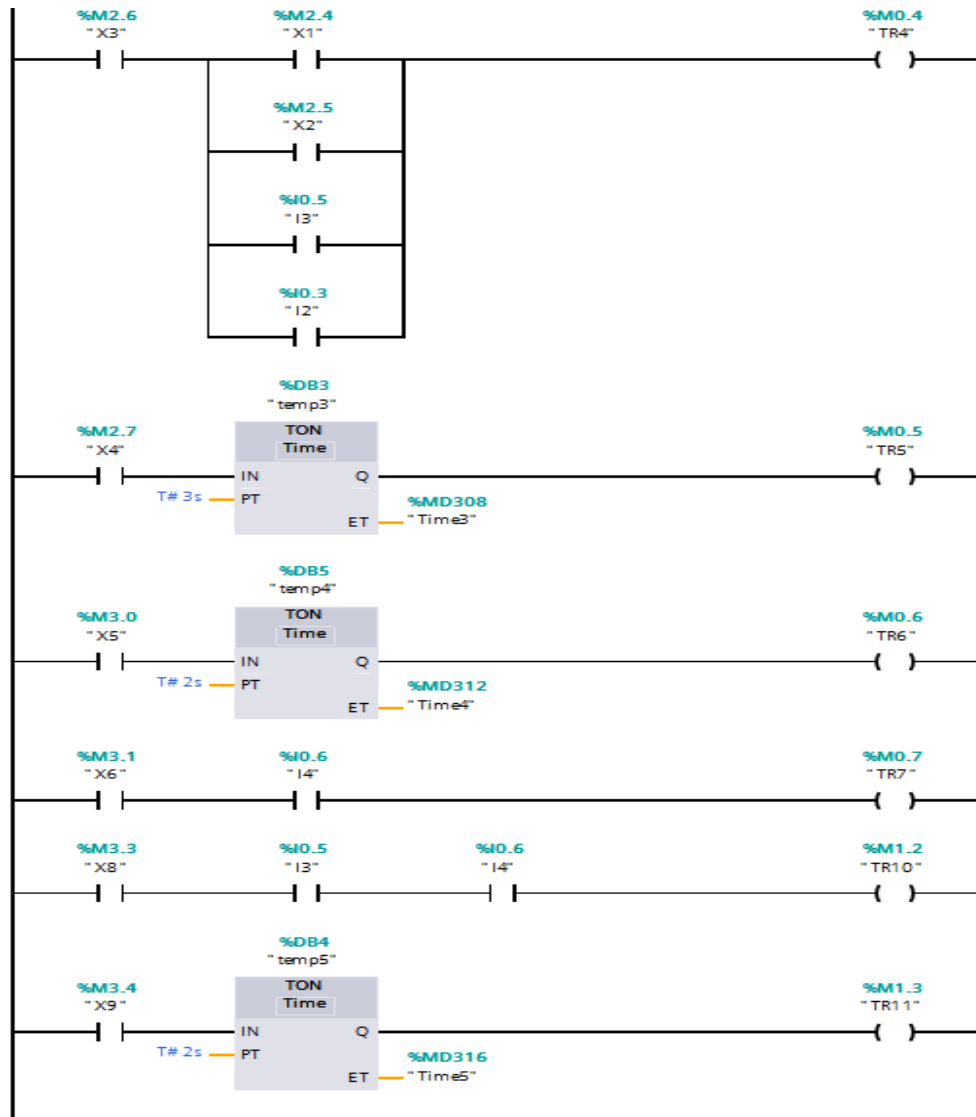


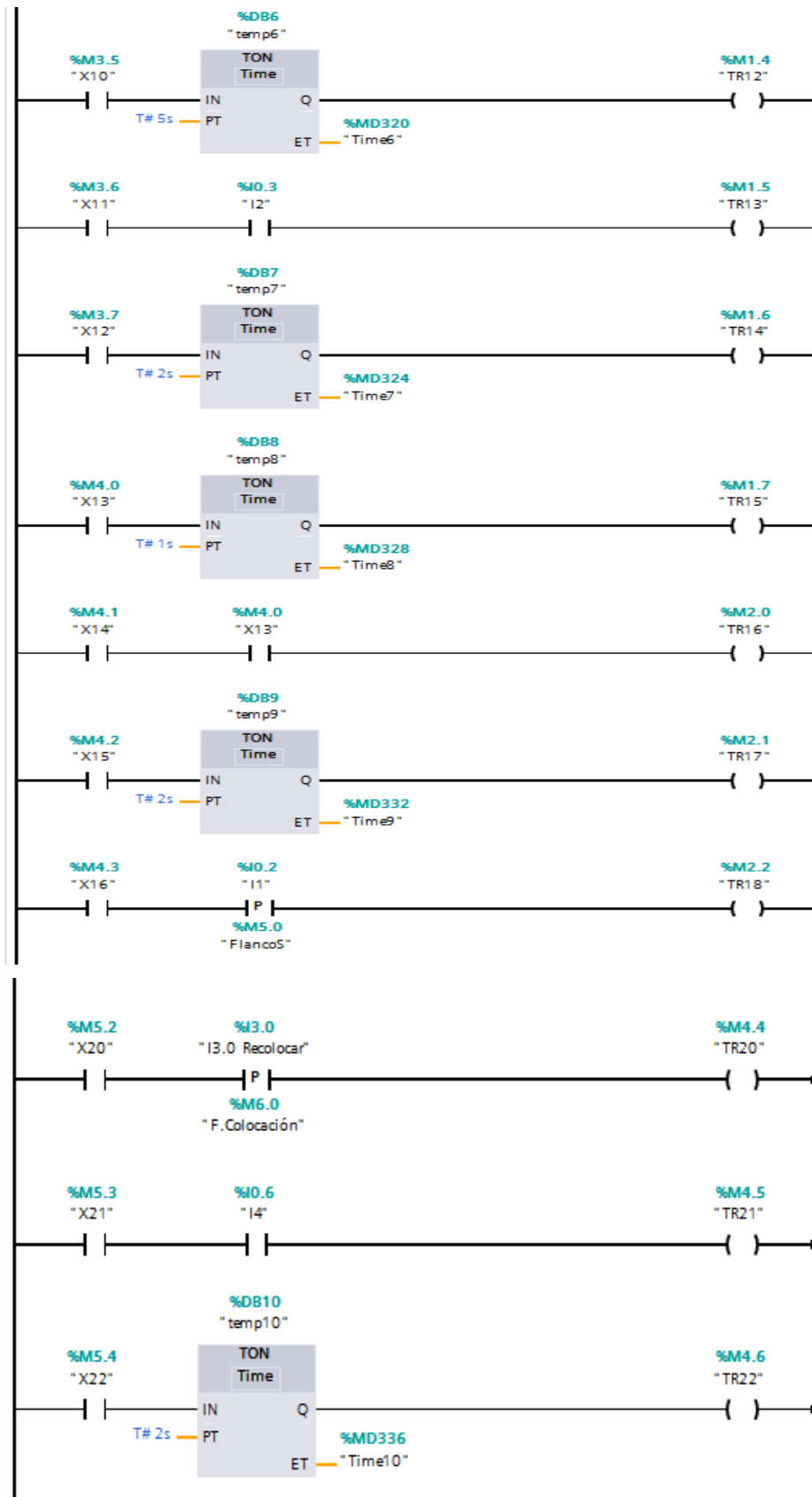


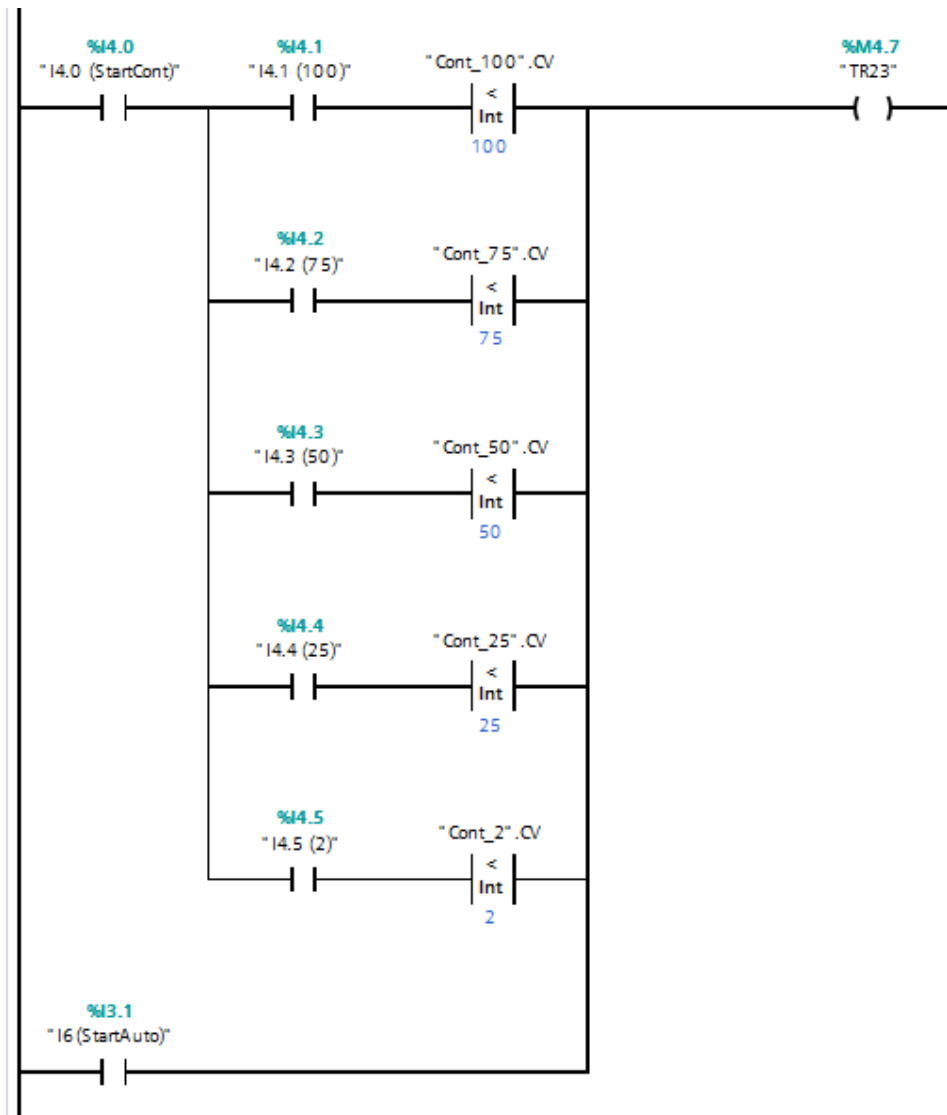


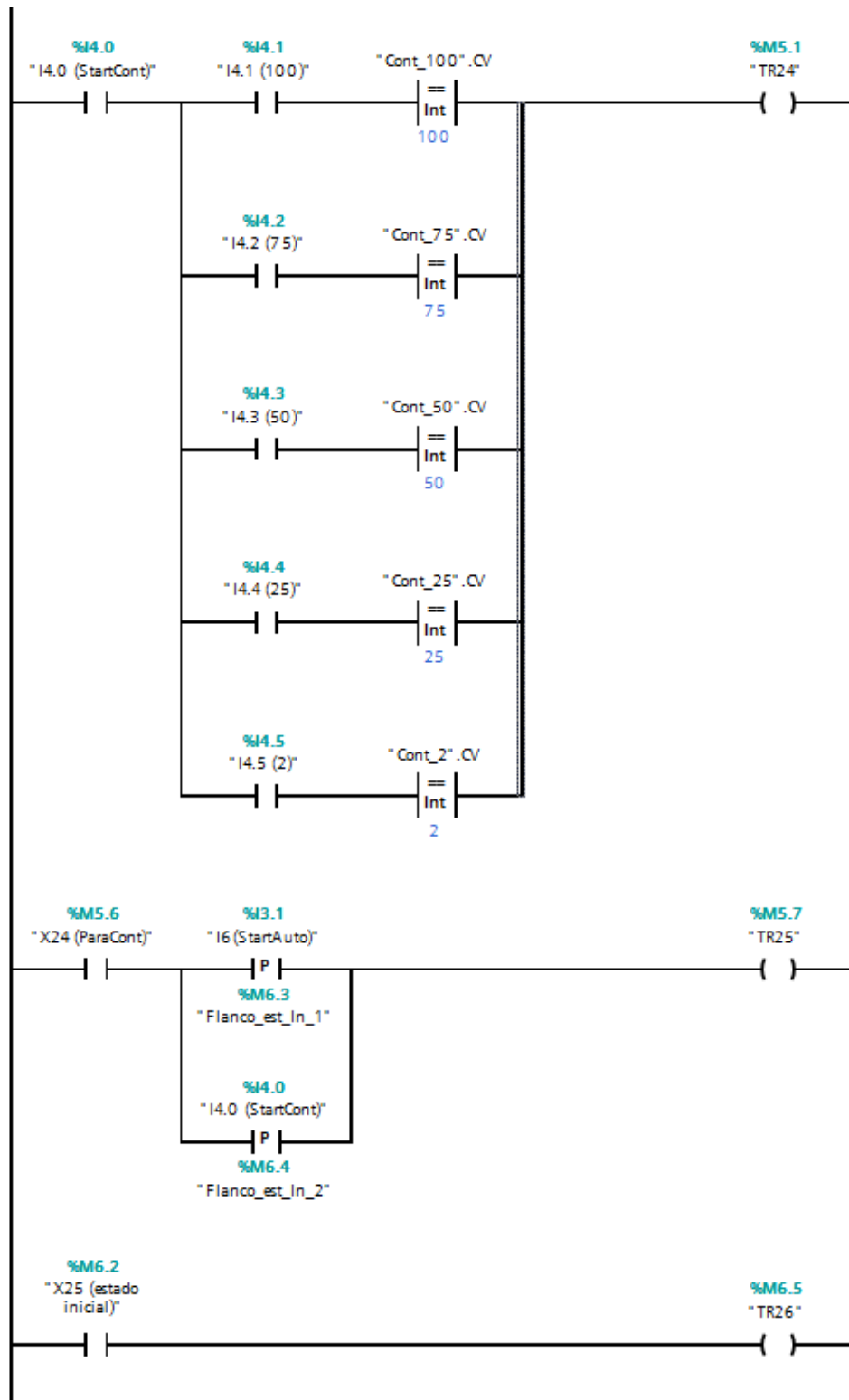
Segmento 3: Transiciones



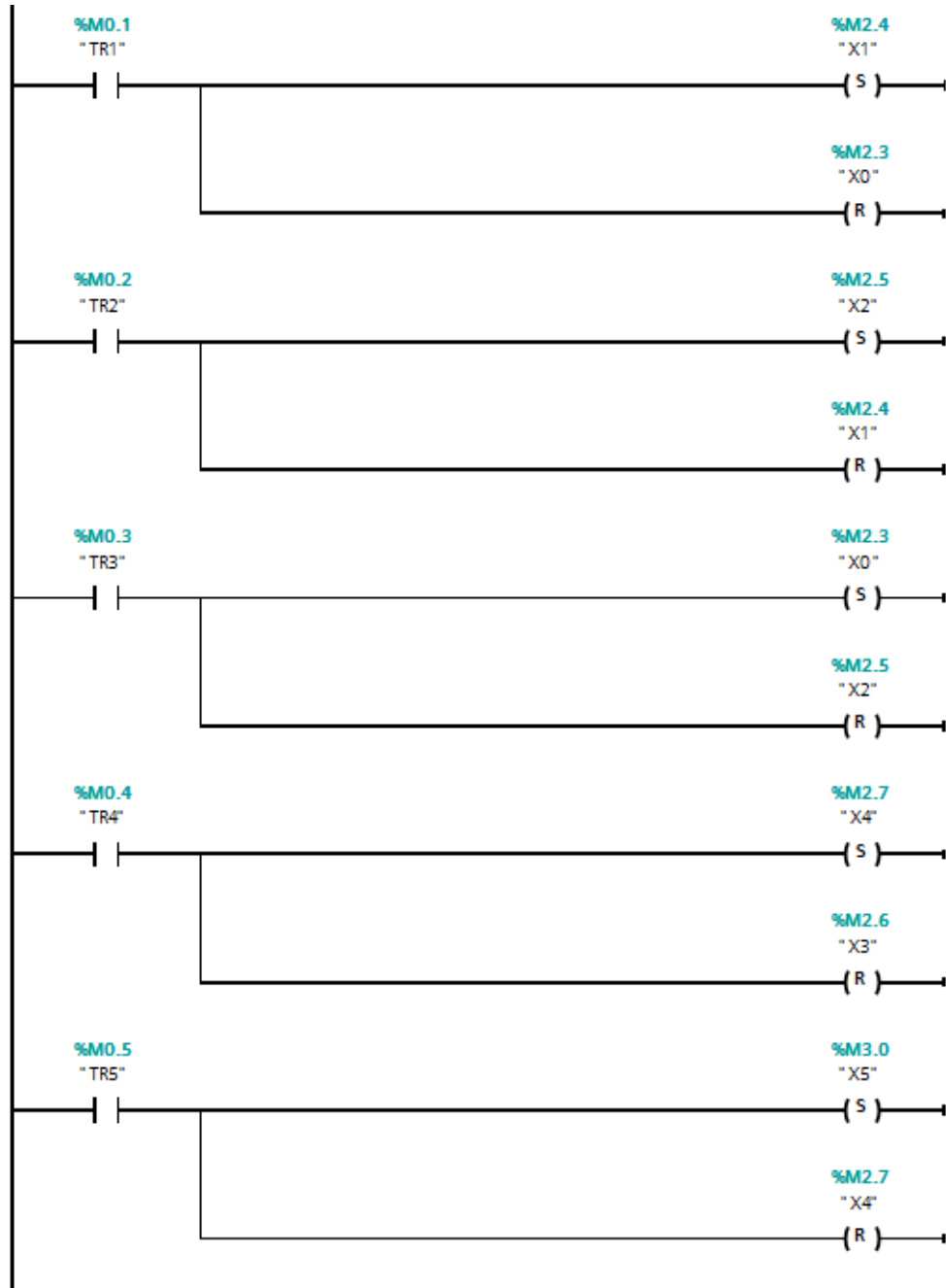


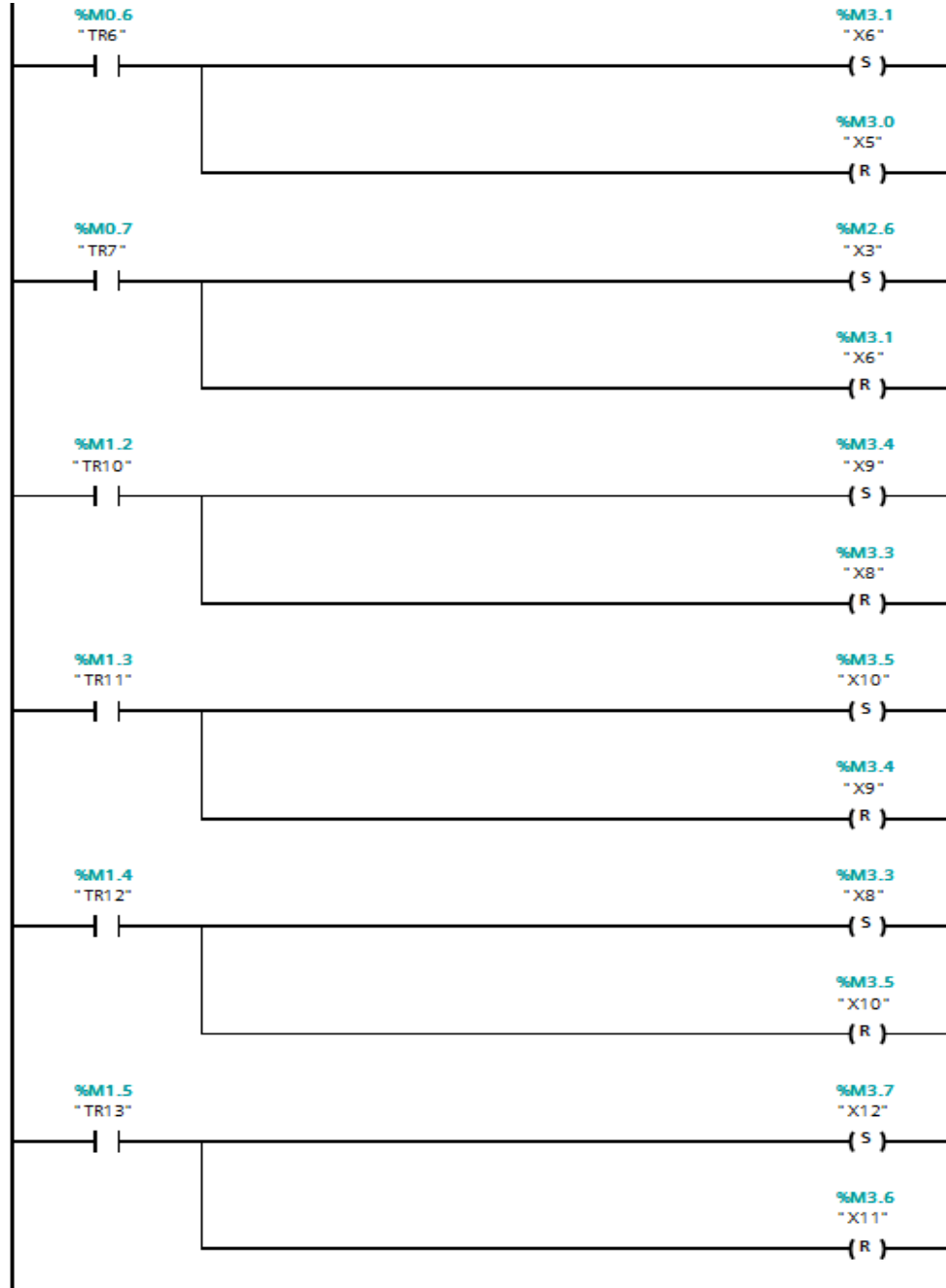


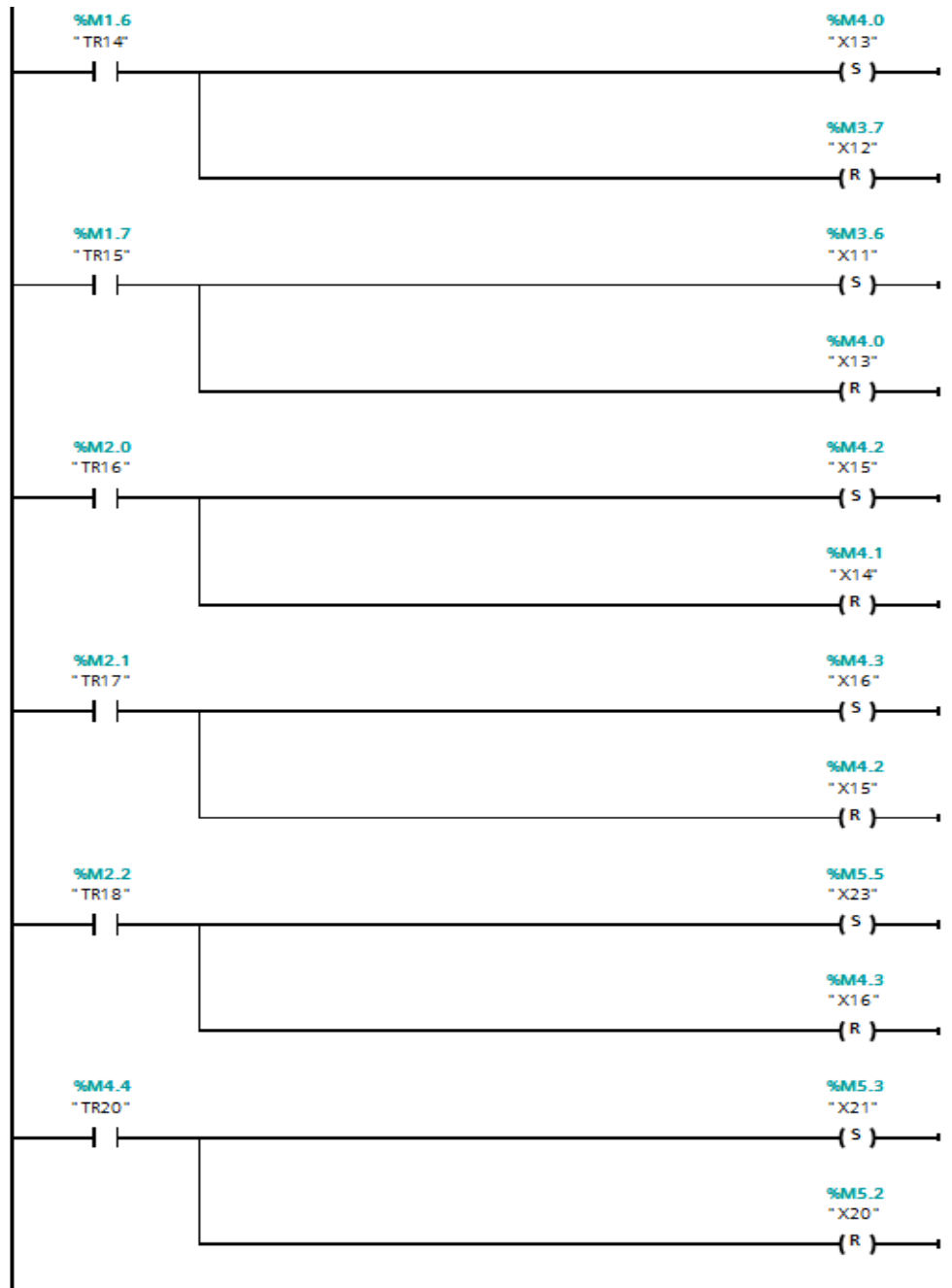


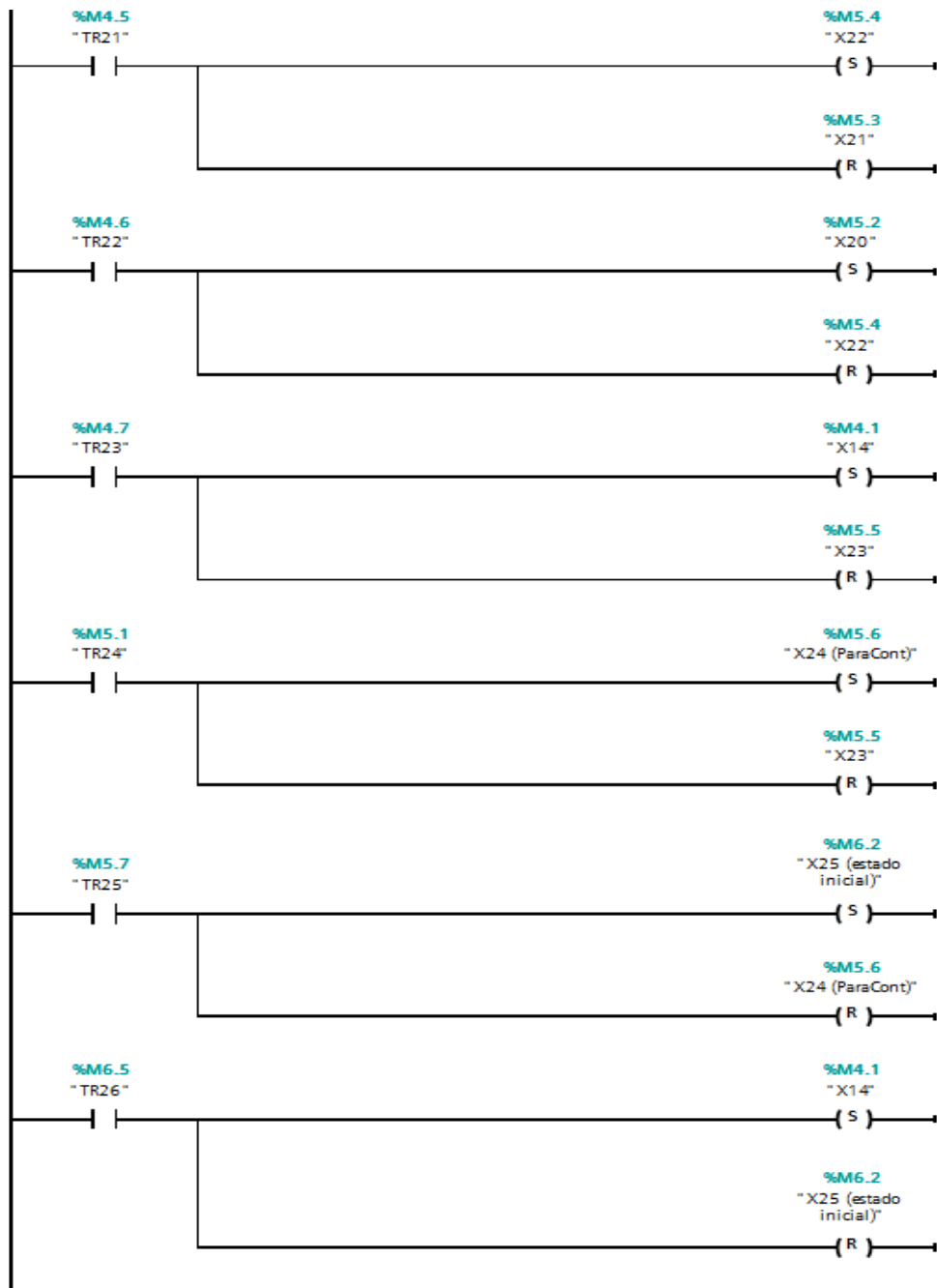


Segmento 4: Etapas









Sección 5: Actuadores

