



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escola Tècnica
Superior d'Enginyeria
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

Automatización de un sistema de clasificación y almacenaje

Trabajo Fin de Grado

Grado en Ingeniería Informática

Autor: Rubén Morenilla Parada

Tutor: Enrique Jorge Bernabeu Soler

2013-2014

Resumen

Este proyecto tiene como objetivo automatizar un sistema de clasificación y almacenaje mediante la programación de un determinado conjunto de sensores y actuadores. Para su desarrollo se ha hecho uso de la herramienta de programación de automatismos PL7, así como de una célula que contiene un robot constituido por un eje vertical neumático y dos ejes horizontales eléctricos para la simulación a escala de un proceso industrial, prestado por el departamento de sistemas y automática (DISA) de la Universidad Politécnica de Valencia. Como resultado del mismo, se pretende realizar varios modelos de almacenaje a fin de conseguir y observar el correcto comportamiento del autómata en una situación de trabajo. A partir de estas observaciones se podrán realizar conclusiones diversas así como discutir las posibles ventajas e inconvenientes de cada uno de los modelos, desde el punto de vista industrial.

Palabras clave: automatización, PL7, industrial, clasificación, almacenaje, paletizado.

Abstract

This project aims to automate a system of classification and storage by programming a given set of sensors and actuators. To develop it has been used an automatism programming tool called PL7 and a cell containing a robot consisting of a pneumatic vertical axis and two electrical horizontal axes for scale simulation of an industrial process, provided by the systems and automatic department (DISA) of the Polytechnic University of Valencia. As a result, it is intended to make several models of storage to achieve and observe the correct behavior of the robot in a work situation. From these observations it may make different conclusions and discuss the possible advantages and disadvantages of each of the models, from the industrial point of view.

Keywords: automation, PL7, industrial, classification, storage, palletizing.

Lista de contenidos

1. Introducción.....	9
1.1 Objetivo	9
1.2 Metodología	10
2. Materiales	11
2.1 SMC IPC-203.....	11
2.2 PLC Telemecanique Modicon TSX Micro 3722.....	15
2.3 PL7	19
2.4 Lenguaje GRAFCET.....	21
2.5 Lenguaje de contactos o Ladder	23
3. Funcionamiento del autómata	26
4. Proceso	27
4.1 Obtención de las señales	27
4.2 Programación del autómata	33
5. Conclusiones.....	51
6. Bibliografía.....	53
Anexo	54
<i>Inicialización_ejes.stx</i>	54
<i>Modelo_zigzag.stx</i>	56
<i>Modelo_espiral.stx</i>	70



Lista de ilustraciones

Ilustración 1. Estación IPC-203.....	12
Ilustración 2. Foto estación IPC-203.....	12
Ilustración 3. Eje neumático vertical.....	13
Ilustración 4. Arquitectura interna del PLC.....	16
Ilustración 5. Telemecanique Modicon TSX Micro 3722.....	17
Ilustración 6. Módulos de entrada/salida del PLC.....	17
Ilustración 7. Interfaz principal PL7.....	19
Ilustración 8. Etapa inicial en GRAFCET.....	21
Ilustración 9. Etapa en GRAFCET.....	21
Ilustración 10. Transición en GRAFCET.....	21
Ilustración 11. Elementos lógicos de entrada.....	23
Ilustración 12. Elementos lógicos de salida.....	23
Ilustración 13. Temporizador en Ladder.....	24
Ilustración 14. Contado en Ladder.....	24
Ilustración 15. Comparador horizontal en Ladder.....	25
Ilustración 16. Comparador vertical en Ladder.....	25
Ilustración 17. GRAFCET de control.....	29
Ilustración 18. Esbozo inicial del programa.....	34
Ilustración 19. Contenido de la transición %X7->%X8.....	36
Ilustración 20. GRAFCET de un ciclo de almacenamiento.....	37
Ilustración 21. GRAFCET de inicialización.....	38
Ilustración 22. Activación de las señales de habilitación.....	38
Ilustración 23. Contador de tres botes.....	39
Ilustración 24. GRAFCET de tres ciclos de almacenamiento.....	40
Ilustración 25. Contenido de la transición %X5->%X6.....	40
Ilustración 26. Interfaz de programación del PL7.....	42
Ilustración 27. Contenido de la transición %X5->%X99.....	42
Ilustración 28. Contenido de la transición %X99->%X59.....	43
Ilustración 29. GRAFCET de veinticinco ciclos de almacenamiento (Parte 1).....	43
Ilustración 30. GRAFCET de veinticinco ciclos de almacenamiento (Parte 2).....	44
Ilustración 31. Contenido de la transición problemática una vez corregido.....	45
Ilustración 32. Contenido de la transición %X1->%X2 en el modelo zigzag.....	45
Ilustración 33. Inhibición del movimiento "mesa" de recogida en el modelo zigzag.....	46
Ilustración 34. Contador de veinticinco botes.....	47
Ilustración 35. GRAFCET final del modelo zigzag (parte principal).....	47
Ilustración 36. Contenido de la transición %X1->%X2 en el modelo espiral.....	49
Ilustración 37. Inhibición del movimiento "mesa" de recogida en el modelo espiral.....	49

Lista de tablas

Tabla 1. Matriz de posiciones del almacén	14
Tabla 2. Señales de entrada y salida	27
Tabla 3. Acciones de las señales de salida	29
Tabla 4. Relación final de entradas y salidas.....	32
Tabla 5. Modelo de almacenamiento en zigzag.....	41
Tabla 6. Modelo de almacenamiento en espiral	48



1. Introducción

Desde los albores de la humanidad, el hombre ha tratado de facilitar y mejorar la ejecución de tareas pesadas y repetitivas, así como minimizar la peligrosidad de las mismas. Gracias a la automatización, y a las innovaciones relacionadas con este sector que se han dado a lo largo del tiempo, ha sido posible conseguir este objetivo.

Los primeros descubrimientos e inventos relacionados con la automatización fueron creados como objetos animados de culto a los dioses, como en el caso de la civilización egipcia y griega.

Sin embargo, hasta la revolución industrial a mediados del siglo XVIII no se produjo el auténtico avance en el uso de sistemas automatizados y se debió a las innovaciones técnicas producidas en esta época. Un ejemplo de esto es el telar programable con tarjetas perforadas de Joseph Marie Jacquard (1801).

Ya en el siglo XX y con la aparición de los computadores digitales, la automatización alcanzó su máximo esplendor al aumentar la velocidad y poder de cálculo, y disminuir a la vez tanto el precio como el tamaño de los autómatas.

La automatización de procesos se encarga de cumplir con estos requisitos, además de acelerar los tiempos de producción, mejorando así la productividad.

1.1 Objetivo

El objetivo de este proyecto es automatizar un sistema de clasificación y almacenaje, formado por la estación IPC-203 de la marca SMC (véase sección 2.1) de modo que se consiga el correcto almacenamiento o paletizado de veinticinco botes. Este sistema simula un escenario industrial real a escala.

A fin de comparar el comportamiento del sistema en una situación de trabajo, se desarrollarán dos modelos diferentes de almacenaje. Será posible también discutir las posibles ventajas o inconvenientes de uno frente a otro.

De este modo, se pretende desarrollar un proyecto que en palabras del tutor del mismo, Enrique Jorge Bernabeu Soler, es altamente profesional puesto que el trabajo realizado tiene aplicación directa con el mundo real, así como ya más específicamente, en un entorno industrial. Esto viene a expresar que si una empresa solicitara la automatización de un proceso industrial, se podría realizar sin más inconvenientes de los que en este proyecto se exponen, solo que a escala real.

1.2 Metodología

La metodología a seguir en este proyecto es la siguiente.

Primeramente se deberá averiguar e identificar qué señales corresponden a las entradas y salidas del autómata.

Una vez realizado esto, se pasará a programar el autómata de modo que realice el paletizado de un bote a fin de comprobar el funcionamiento del autómata.

Posteriormente, se programará el autómata para que complete el almacenamiento de tres botes con la finalidad de observar que el sistema es capaz de realizar correctamente estos tres ciclos de almacenamiento.

Una vez conseguido el almacenamiento de tres botes, se pasará a automatizar un ciclo de veinticinco almacenamientos, de modo que el sistema trabaje con la cantidad de botes final. Se realizará para ello uno de los dos modelos propuestos.

Si se tiene éxito en este último paso, se podrá realizar la ampliación del programa ya obtenido de manera que se automaticen las situaciones en las que la baliza roja debe encenderse para indicar un estado de alarma. De esta manera se concluirá la programación del primer modelo.

En este momento, y partiendo del primer modelo obtenido ya finalizado, se podrá programar un segundo modelo a fin de comparar ambos, realizando los cambios pertinentes.

Durante todo este proceso se atenderá la aparición de posibles errores y fallos, de manera que su corrección permita llegar al objetivo final de obtener dos modelos válidos, así como la posibilidad de aprender de ellos. Para ello, se realizarán pruebas constantes, tanto para la obtención de señales como para descubrir el motivo de los errores o simplemente observar si el programa realizado hasta el momento es correcto.

2. Materiales

Para el desarrollo de este proyecto se ha hecho uso de los siguientes materiales:

2.1 SMC IPC-203

Es una célula que permite el paletizado de los botes con los que se trabaja en un almacén de veinticinco posiciones.

Esta célula está compuesta por los siguientes elementos:

- PLC de control: Telemecanique Modicon TSX Micro 3722 (ahora Schneider) (Descripción: véase apartado 2.2.)
- Unidad para bus de comunicación.
- Interruptor magnetotérmico.
- Bloque de electroválvulas.
- Botonera de mando.
- Baliza de señalización.
- Unidad de tratamiento de aire.
- Fuente de alimentación.
- Estructura de aluminio anodizado.
- Un eje vertical neumático: Este eje se aproxima a la posición de origen mediante un cilindro neumático en la que se encuentra un bote cerrado, listo para ser almacenado. El sistema de sujeción de los botes esta realizado mediante una ventosa que retiene el bote en su movimiento hasta que es almacenado en la posición deseada. Este eje posee dos finales de carrera en los extremos superior e inferior. (Véase Ilustración 3)
- Dos ejes eléctricos horizontales: Para el almacenaje de los botes, es necesario situar el eje vertical sobre el origen de recogida desde donde los botes van a ser llevados al almacén. Una vez alcanzado el almacén, los botes han de posicionarse en diferentes puntos del mismo. Los dos ejes horizontales cuentan con motores paso a paso para el control de la posición, además de un *driver* por cada eje controlado por las señales digitales del autómeta.
- Un almacén de veinticinco posiciones.
- Dos detectores de presencia en la posición de origen para detectar botes y tapas.

La estación IPC-203 es un sistema basado en tres ejes de coordenadas en el que los botes son recogidos de una posición de origen y son distribuidos sobre cualquier punto del almacén de veinticinco posiciones.

En la Ilustración 2 se muestra, de izquierda a derecha, la baliza roja, el almacén con las veinticinco posiciones ocupadas con botes, el eje vertical neumático, los dos ejes horizontales eléctricos, y la botonera.

En la Ilustración 3 se muestra el eje neumático vertical, así como los dos finales de carrera y la ventosa para la sujeción de los botes mediante succión que posee.

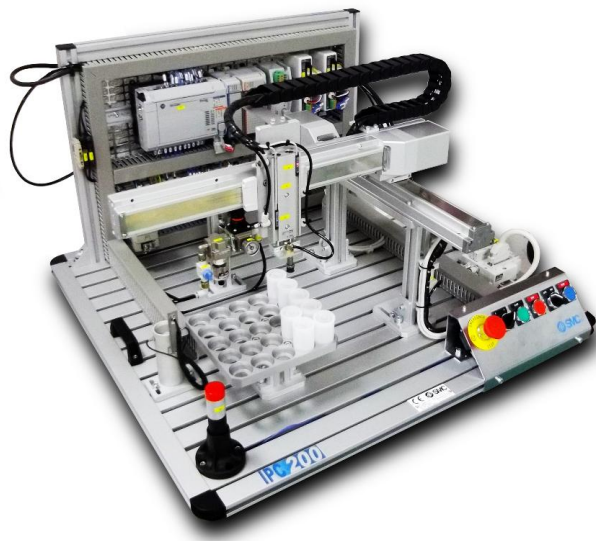


Ilustración 1. Estación IPC-203

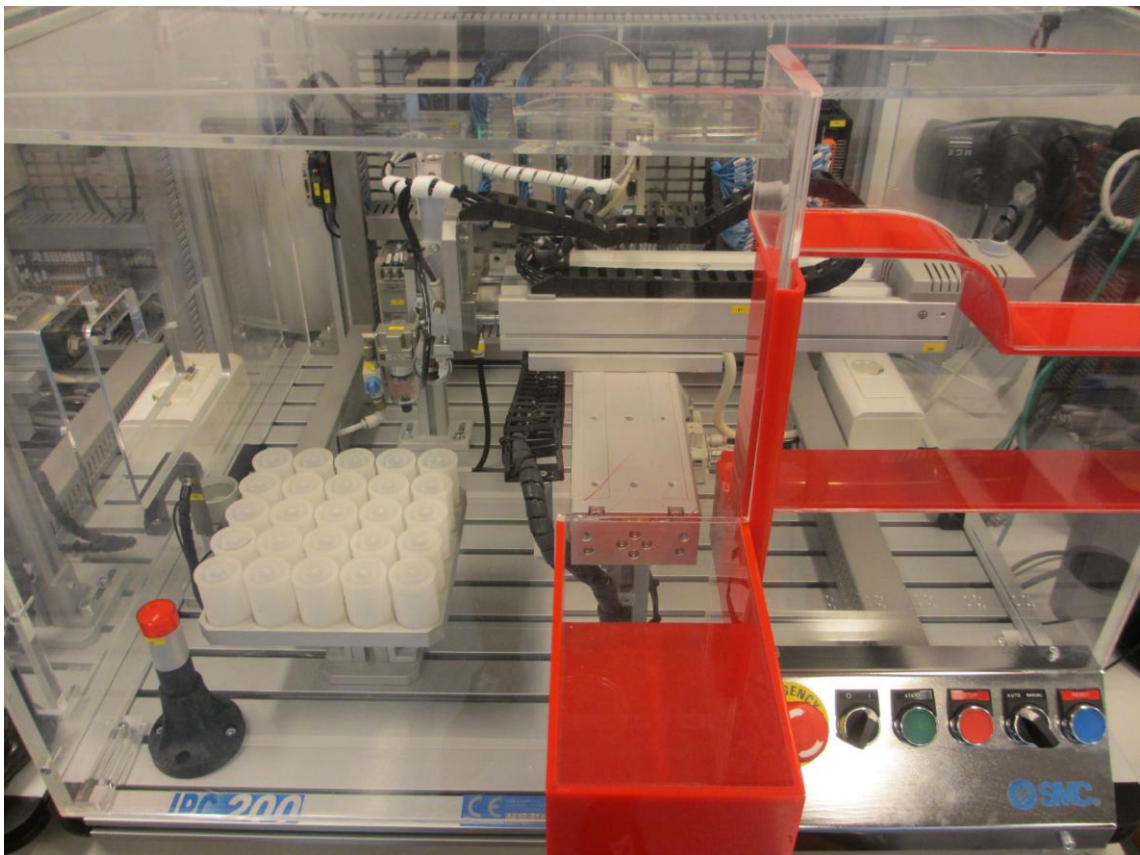


Ilustración 2. Foto estación IPC-203

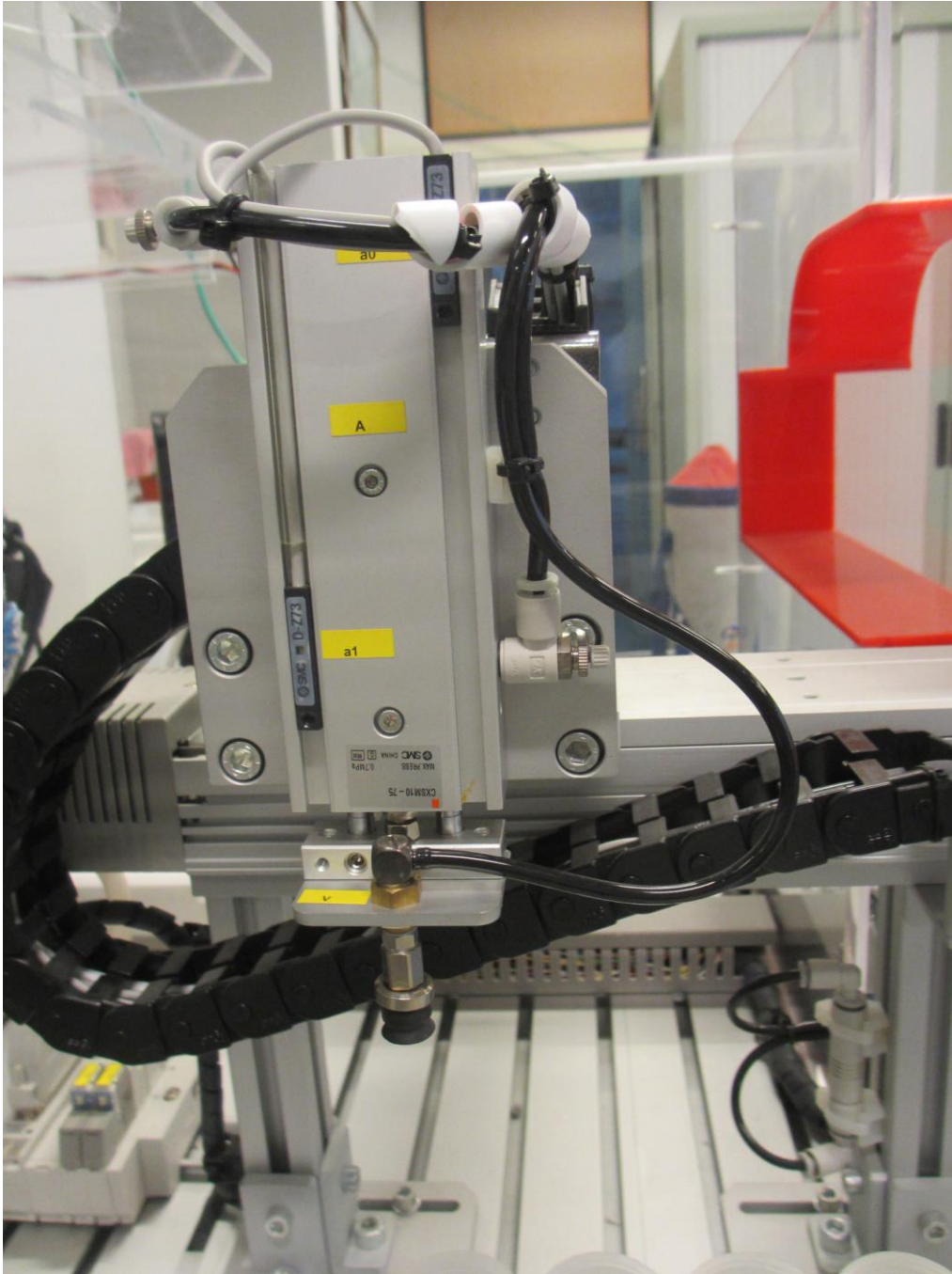


Ilustración 3. Eje neumático vertical

El almacén de veinticinco posiciones, visto desde la perspectiva de la Ilustración 2, tiene codificada cada posición de la siguiente manera:

Tabla 1. Matriz de posiciones del almacén

Mesa 5 Traslación 5	Mesa 5 Traslación 4	Mesa 5 Traslación 3	Mesa 5 Traslación 2	Mesa 5 Traslación 1
Mesa 4 Traslación 5	Mesa 4 Traslación 4	Mesa 4 Traslación 3	Mesa 4 Traslación 2	Mesa 4 Traslación 1
Mesa 3 Traslación 5	Mesa 3 Traslación 4	Mesa 3 Traslación 3	Mesa 3 Traslación 2	Mesa 3 Traslación 1
Mesa 2 Traslación 5	Mesa 2 Traslación 4	Mesa 2 Traslación 3	Mesa 2 Traslación 2	Mesa 2 Traslación 1
Mesa 1 Traslación 5	Mesa 1 Traslación 4	Mesa 1 Traslación 3	Mesa 1 Traslación 2	Mesa 1 Traslación 1

En este punto se va a incluir un estudio sobre los sensores y los actuadores que componen la estación IPC-203.

Los sensores disponibles son los siguientes:

- 2 detectores magnéticos tipo Reed (D-A93L) colocados como finales de carrera para el eje vertical. Pueden apreciarse en la Ilustración 3.
- 2 detectores de origen (E2E-X1C1) de Omron que funcionan como sensores de proximidad para los ejes eléctricos horizontales.
- 1 sensor de fibra óptica (E3X-A41) de Omron, colocado en la posición de recogida de botes a fin de detectar la presencia de botes.
- 1 sensor de fibra óptica (E3X-NA41) de Omron, colocado también en la posición de recogida de botes para detectar la presencia de tapa en los botes.

Los actuadores que componen la estación son:

- 1 cilindro con dos vástagos para manipulación por vacío, con una longitud de carrera de 75 milímetros (CXSM10-75). Dispone de reguladores de caudal. Está controlado por una electroválvula 5/2 monoestable. Colocado como eje vertical neumático. Dispone de espacio para los detectores magnéticos de final de carrera.
- 1 ventosa plana de 10 milímetros de diámetro (ZPT10UN-B5). Está controlada por una electroválvula 5/2 monoestable. Colocada en el extremo inferior del eje vertical para permitir la recogida de botes por vacío.
- 1 eje eléctrico posicionador translación (LXPB2SA-200S-Q). Su función es la de eje horizontal para el movimiento “translación”.
- 1 eje eléctrico posicionador mesa (LXSH2SA-150S-Q). Su función es la de eje horizontal para el movimiento “mesa”.
- 2 drivers posicionadores LC6D-220-AD-Q. Utilizados para controlar el movimiento de los motores paso a paso de los ejes horizontales.

Este conjunto de sensores y actuadores es el que se utilizará para cumplir con los objetivos de este proyecto.

2.2 *PLC Telemecanique Modicon TSX*

Micro 3722

El PLC (en inglés, *Programmable Logic Controller*) o controlador lógico programable, es un computador usado en la automatización de procesos electromecánicos, como el que en este proyecto se desarrolla. Está pensado para controlar múltiples señales de entrada y salida a partir de un programa desarrollado por el usuario. El programa activará determinadas salidas en función del estado de las entradas.

Un PLC está compuesto por las siguientes partes:

- Unidad central de proceso (CPU): Elabora las señales de salida en función del estado de las entradas y de las instrucciones a ejecutar presentes en la memoria de programa.
- Memoria: Esta compuesta por los siguientes bloques:
 - Memoria de datos: encargada de almacenar variables y datos de cálculos que no se reflejan sobre las salidas, así como una imagen de los estados actuales de las entradas y las salidas.
 - Memoria de programa: contiene las instrucciones necesarias para la ejecución del programa, así como los parámetros de configuración del PLC.
- Interfaz de entradas y salidas: establece la comunicación del PLC con el sistema que se desea controlar puesto que adapta el formato de las señales de entrada y salida para ser utilizadas por el PLC.
- Fuente de alimentación: proporciona las tensiones necesarias para el funcionamiento de la parte electrónica del sistema a programar a partir de una tensión externa. En algunos modelos de PLC, existe una batería conectada a la fuente de alimentación a fin de guardar los datos almacenados en caso de desconexión de la tensión externa.
- Bus interno: contiene las conexiones eléctricas necesarias entre la CPU, la memoria y la interfaz de entrada y salida. Los módulos adicionales del PLC deberán conectarse a este bus.

En la siguiente ilustración se puede observar la arquitectura de un PLC, cuyas partes acaban de ser explicadas.

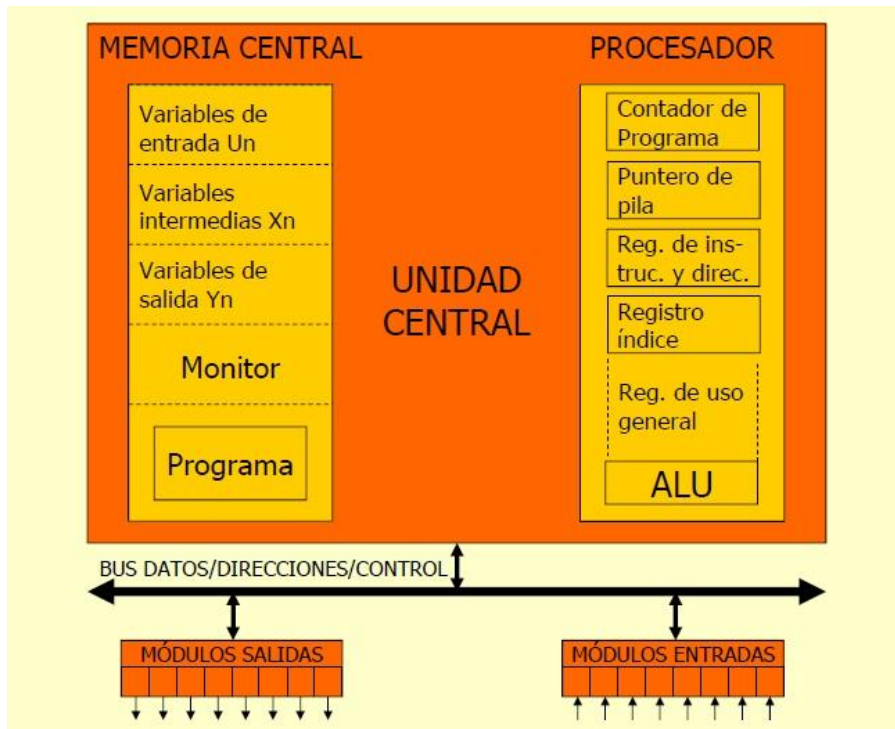


Ilustración 4. Arquitectura interna del PLC

Un PLC realiza las siguientes fases en su funcionamiento:

- Puesta en marcha: donde se comprueba la memoria y los componentes del PLC, así como el programa introducido, las comunicaciones y los módulos de entrada y salida. También se desactivan las salidas en esta fase.
- Ejecución: donde se adquieren los estados de las entradas, se procesan las ecuaciones lógicas y en función de esto se actualizan las salidas. Se realizan también tareas de diagnóstico interno así como las comunicaciones entre las distintas partes del PLC.
- Parada del sistema: donde se cesa la ejecución del programa contenido en el PLC y se desactivan sus salidas.

En este proyecto en concreto se ha utilizado un PLC de la marca Telemecanique (Schneider en la actualidad), modelo Modicon TSX Micro 3722 con 3 módulos de 16 entradas y 12 salidas cada uno, lo que hace un total de 48 entradas y 36 salidas utilizables.



Ilustración 5. Telemecanique Modicon TSX Micro 3722

Los módulos de este PLC usan como notación para los bloques de entradas números impares empezando por uno, y para los bloques de salidas números pares empezando por dos; utilizando %I para las entradas y %Q para las salidas. De este modo, la última entrada del primer bloque de entradas sería la %I1.15, y la primera salida del segundo bloque de salidas sería %Q4.0.

En la Ilustración 5 se puede observar el PLC de la célula a automatizar. Posee una pantalla en la parte frontal que muestra qué señales de entrada y salida hay activas. Esta pantalla tiene tres columnas (una por cada módulo de entrada/salida) divididas en dos mitades. En la parte superior de cada columna se representan las entradas activas, y en la inferior las salidas activas. Así pues, la primera columna empezando por la izquierda contendrá los bloques 1 y 2 de entrada y salida, la segunda los bloques 3 y 4, y la tercera columna los bloques 5 y 6. Esto puede observarse en la siguiente ilustración.



Ilustración 6. Módulos de entrada/salida del PLC

Automatización de un sistema de clasificación y almacenaje

En un bloque de entrada las 16 entradas están numeradas del 0 al 15. Del mismo modo, en un bloque de salida con 12 de ellas, estas están numeradas del 0 al 11.

Para comunicar un ordenador con el PLC se debe conectar el puerto serie del primero a la entrada correspondiente del PLC.

2.3 PL7

El programa PL7 es un software propietario dedicado a los PLC Modicon de la marca Telemecanique, que sirve para desarrollar programas para los autómatas a controlar. Para desarrollar el programa se pueden usar lenguajes como el de contactos, lista de instrucciones, texto estructurado y GRAFCET.

Este es el software usado en la realización de este proyecto debido al PLC que se halla instalado en la célula IPC-203, siendo empleados para ello el lenguaje de contactos y GRAFCET.

Una de las características de esta interfaz es que permite la monitorización del programa cuando el sistema a controlar está conectado al PC.

Durante la programación del autómata se han tenido varias dificultades relacionadas con las características de este software, las cuales serán descritas más adelante en esta memoria (véase sección 4.2).

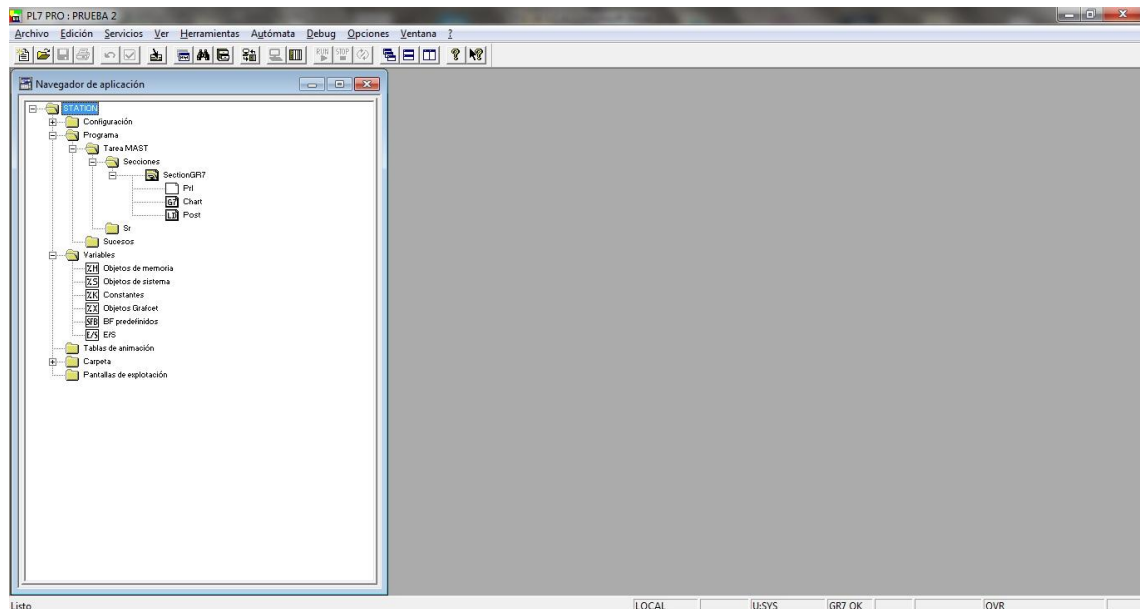


Ilustración 7. Interfaz principal PL7

En la Ilustración 7 se muestran las secciones de las que se compone el programa. Las más importantes son las secciones *Chart* y *Post*.

En la sección *Chart* se realiza el programa principal en GRAFCET. De este modo se compone la estructura del programa que define el comportamiento del autómata.

La sección *Post* es necesaria para completar el programa puesto que define qué etapas del GRAFCET programado en la sección *Chart* activarán las salidas para realizar las acciones que se desean conseguir del sistema conectado al PLC. Esta sección se programa en lenguaje de contactos o Ladder.

Se observa también dentro de la carpeta "Variables", una sección llamada "BF predefinidos". En ella se configura el tiempo a establecer en cada temporizador.

Las referencias a los objetos propios del lenguaje GRAFCET serán explicadas en la siguiente sección *Lenguaje GRAFCET* (véase sección 2.4), de igual modo que las de los objetos propios del lenguaje de contactos, en la sección *Lenguaje de contactos o Ladder* (véase sección 2.5).

2.4 Lenguaje GRAFCET

GRAFCET (en francés, Graphe Fonctionnel de Commande Etape Transition), es un lenguaje de programación que permite describir las diferentes acciones de un sistema lógico caracterizado por sus entradas y salidas, representándolas de manera gráfica.

Este lenguaje dispone de unos elementos gráficos de programación que es necesario conocer a fin de programar con él, los cuales van a ser presentados a continuación.

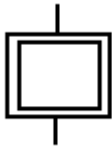


Ilustración 8. Etapa inicial en GRAFCET

En la Ilustración 8 se muestra una etapa inicial, un elemento que indica el comienzo de un programa GRAFCET. Se activa cuando se enciende el autómata. Solo debe de existir una etapa de este tipo en el programa.

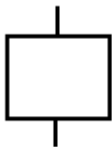


Ilustración 9. Etapa en GRAFCET

En la Ilustración 9 se muestra una etapa normal. La activación de una etapa, ya sea inicial o normal como esta, lleva asociado la realización de una determinada acción.



Ilustración 10. Transición en GRAFCET

En la Ilustración 10 se muestra una transición en el lenguaje GRAFCET y contiene la condición necesaria para desactivar la etapa en ejecución y activar la siguiente.

Estos tres elementos principales son los que conforman un programa en GRAFCET. A fin de entender el comportamiento de un programa en este lenguaje, además de ser capaces de programar correctamente un autómata, hay que tener en cuenta una serie de principios básicos que rigen su comportamiento. Estos principios básicos son los siguientes:

- Las etapas y transiciones siempre se colocan de forma alterna.
- Cada etapa tiene asociada una serie de acciones que solo se realizarán cuando dicha etapa esté activa.
- La activación de una etapa se realiza cuando la transición que la precede contiene una condición cierta y está activa la etapa anterior.
- Cuando la condición de una transición es cierta, se desactiva la etapa precedente y se activa la siguiente.

Teniendo en cuenta estos principios básicos y los elementos de programación anteriormente presentados ya es posible crear un programa en GRAFCET. Sin embargo, para realizar una programación completa del automatismo es necesario tener unos conocimientos del lenguaje de contactos o Ladder. Este lenguaje va a ser descrito en la siguiente sección (véase sección 2.5).

2.5 Lenguaje de contactos o Ladder

El lenguaje de contactos, en escalera o Ladder es un lenguaje gráfico de programación con bastante repercusión en el ámbito de la programación de autómatas.

Del mismo modo que el lenguaje GRAFCET, este lenguaje posee unos elementos básicos que deben conocerse.






Elementos de prueba 		
<u>Instrucciones booleanas</u>		
Contacto de cierre		Contacto establecido cuando el objeto bit que lo controla pasa al estado 1.
Contacto de apertura		Contacto establecido cuando el objeto bit que lo controla pasa al estado 0.
Contacto de detección de cambio de estado		Flanco ascendente: detección del paso de 0 a 1 del objeto bit que lo controla.
		Flanco descendente: detección del paso de 1 a 0 del objeto bit que lo controla.

Ilustración 11. Elementos lógicos de entrada

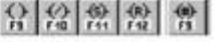





Elementos de acción 		
<u>Asignación booleana</u>		
Bobina directa		El objeto bit asociado toma el valor del resultado del área de prueba.
Bobina inversa		El objeto bit asociado toma el valor inverso del resultado del área de prueba.
Bobina de desactivación		El objeto bit asociado se pone a 1 cuando el resultado del área de prueba es 1.
Bobina de activación		El objeto bit asociado se pone a 0 cuando el resultado del área de prueba es 1.
Bobina de receptividad		El objeto bit asociado se pone a 1 cuando el resultado del área de prueba es 1.

Ilustración 12. Elementos lógicos de salida

En las Ilustraciones 11 y 12 se muestran algunos de los elementos básicos de este lenguaje junto a su función, por lo cual la explicación de esta se omitirá para los elementos aquí presentados.

Además de estos, existen otros elementos que serán de utilidad en la programación del autómata que este proyecto relata.

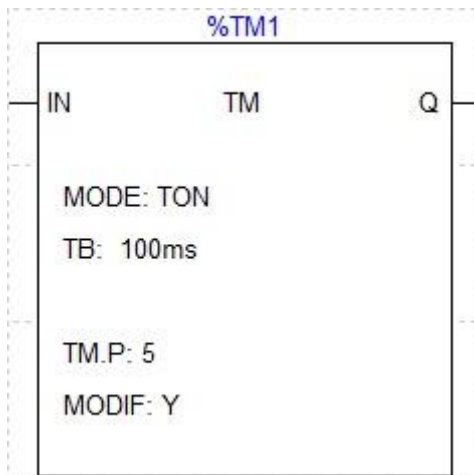


Ilustración 13. Temporizador en Ladder

En la Ilustración 13 se muestra un temporizador. La entrada “IN” del mismo se utiliza para comenzar una cuenta de tiempo, cuyo valor se ha establecido en la sección “BF predefinidos” de la aplicación PL7 (véase sección 2.3). La salida “Q” se activa en el momento dicha cuenta de tiempo expira. En la ilustración, el tiempo establecido de este temporizador es de 0,5 segundos.

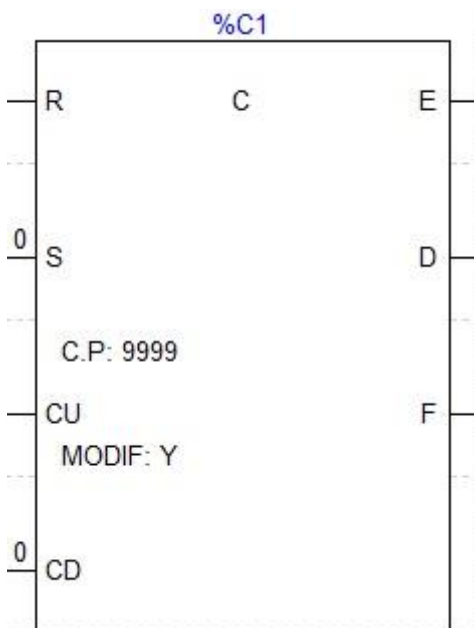


Ilustración 14. Contador en Ladder

En la Ilustración 14 se observa un contador. La entrada “R” se utiliza para poner a cero la cuenta que se esté llevando en ese momento. “S” sirve para establecer un valor en el contador distinto de cero. “CU” aumenta en uno la cuenta interna del contador, mientras que CD la decrementa. La salida “E” indica que el contador está a cero mientras que la salida “F” indica que el contador ha llegado a su número máximo. La salida “D” se activa cuando la cuenta interna del contador ha llegado al número establecido. Sin embargo, para consultar el valor actual del contador se utiliza la variable %C1.V, siendo %C1 la referencia al contador. La variable “V” no se muestra en la ilustración.

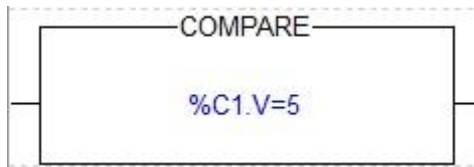


Ilustración 15. Comparador horizontal en Ladder

En la Ilustración 15 se muestra un comparador horizontal. Cuando la condición que contiene el comparador se hace cierta, este se activa, emitiendo de esta manera una señal que confirma este hecho.

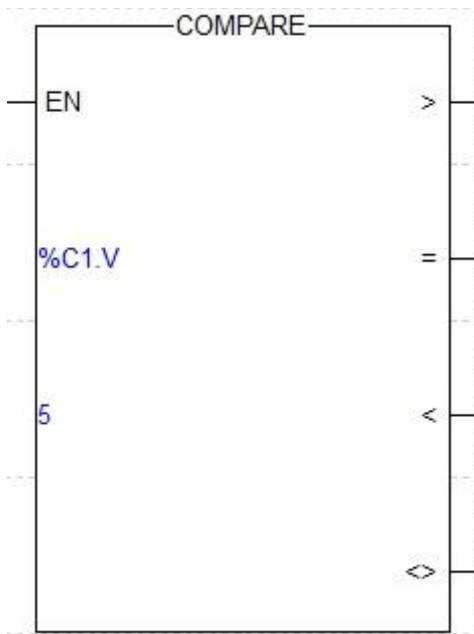


Ilustración 16. Comparador vertical en Ladder

En la Ilustración 16 se muestra un comparador vertical en el lenguaje Ladder. En él se introducen las variables a comparar. Posee una entrada “EN” de habilitación, la cual activa el comparador cuando está conectada. Las salidas que posee se activan cuando la condición entre estas dos variables se hace cierta, de modo que, tomando los valores del ejemplo, si %C1.V es igual a 5, se activará la salida “=” del comparador vertical.

Los programas en lenguaje de contactos crean funciones lógicas de modo que las bobinas de salida contienen el resultado de dichas funciones. Estos programas se leen de arriba abajo y de izquierda a derecha.

Teniendo en cuenta estas condiciones y los elementos presentados, es posible realizar una programación en este lenguaje que permita completar el programa en GRAFCET que define el automatismo objetivo de este proyecto.

3. Funcionamiento del autómatas

El autómatas dispone de una botonera con los siguientes botones (véase Ilustración 2):

- Start
- Stop
- Auto/Man
- Reset
- Un interruptor de encendido
- Una seta para parada de emergencia.

Mediante el uso de estos botones, el autómatas presenta el siguiente funcionamiento:

Cada vez que se pulse el botón de *start* el paletizador se mueve hacia la posición de recogida, coge un bote y lo lleva a su posición correspondiente en el almacén. Con sucesivas pulsaciones de este botón, se irá llenando el almacén.

Una vez se han ocupado todas las posiciones del almacén se enciende la baliza roja de forma fija y no se permitirá introducir más botes hasta que se vacíe el almacén de forma manual y se pulse el botón de *reset*. Volviendo a pulsar *start* después de esto, se podrán introducir más botes.

La baliza con luz intermitente indica que en la posición de origen hay un bote sin tapa o bien no hay bote. Para eliminar esta alarma, se debe poner un bote tapado en esta posición y pulsar el botón de *reset*. Como se ha descrito anteriormente, pulsando *start* se continuará con el normal funcionamiento del autómatas.

4. Proceso

En esta sección se describe el proceso seguido para la automatización de la célula IPC-203.

4.1 Obtención de las señales

Con el fin de programar el autómatas, es necesario saber de qué entradas y salidas se dispone exactamente, además de saber a qué entrada o salida del mismo corresponde cada una.

En el manual de la célula proporcionado por el profesor existe un anexo del cual se ha obtenido una lista de las entradas y salidas de las que dispone el autómatas, la cual se muestra en la siguiente tabla:

Tabla 2. Señales de entrada y salida

	ACCIÓN	SÍMBOLO	TIPO
ENTRADAS	Pulsador Start	m	N.A.
	Pulsador Stop	p	N.C.
	Selector auto/man	A/M	N.A.
	Pulsador Reset	r	N.A.
	Detector de botes	dp1	Sensor fibra óptica PNP
	Detector de botes tapados	dp2	Sensor fibra óptica PNP
	Ventosa arriba	a0	Detector magnético Reed
	Ventosa abajo	a1	Detector magnético Reed
	Salida 1 mesa	S1M	Salida optoacoplada
	Salida 2 mesa	S2M	Salida optoacoplada
	Salida 3 mesa	S3M	Salida optoacoplada
	Salida 4 mesa	S4M	Salida optoacoplada
	Salida 5 mesa	S5M	Salida optoacoplada
	Salida Ready (preparado) mesa	SRM	Salida optoacoplada
	Salida Position (posición) mesa	SPM	Salida optoacoplada
	Salida Busy (ocupado) mesa	SBM	Salida optoacoplada
	Salida Home (origen) mesa	SHM	Salida optoacoplada
	Salida Alarm (alarma) mesa	SAM	Salida optoacoplada
	Salida 1 translación	S1T	Salida optoacoplada
	Salida 2 translación	S2T	Salida optoacoplada
	Salida 3 translación	S3T	Salida optoacoplada
	Salida 4 translación	S4T	Salida optoacoplada
	Salida 5 translación	S5T	Salida optoacoplada
	Salida Ready (preparado) translación	SRT	Salida optoacoplada
	Salida Position (posición) translación	SPT	Salida optoacoplada
	Salida Busy (ocupado) translación	SBT	Salida optoacoplada
	Salida Home (origen) translación	SHT	Salida optoacoplada
	Salida Alarm (alarma) translación	SAT	Salida optoacoplada

SALIDAS	Entrada 0 mesa	E0M	Entrada optoacoplada
	Entrada 1 mesa	E1M	Entrada optoacoplada
	Entrada 2 mesa	E2M	Entrada optoacoplada
	Entrada 3 mesa	E3M	Entrada optoacoplada
	Entrada 4 mesa	E4M	Entrada optoacoplada
	Entrada 5 mesa	E5M	Entrada optoacoplada
	Entrada 6 mesa	E6M	Entrada optoacoplada
	Entrada 7 mesa	E7M	Entrada optoacoplada
	Entrada Enable (habilitación) mesa	EM	Entrada optoacoplada
	Entrada Servo ON mesa	SOM	Entrada optoacoplada
	Entrada Hold (mantenimiento) mesa	HM	Entrada optoacoplada
	Entrada Alarm Reset (borrado) mesa	ARM	Entrada optoacoplada
	Entrada Emergency mesa	EMM	Entrada optoacoplada
	Entrada 0 translación	E0T	Entrada optoacoplada
	Entrada 1 translación	E1T	Entrada optoacoplada
	Entrada 2 translación	E2T	Entrada optoacoplada
	Entrada 3 translación	E3T	Entrada optoacoplada
	Entrada 4 translación	E4T	Entrada optoacoplada
	Entrada 5 translación	E5T	Entrada optoacoplada
	Entrada 6 translación	E6T	Entrada optoacoplada
	Entrada 7 translación	E7T	Entrada optoacoplada
	Entrada Enable (habilitación) trans.	ET	Entrada optoacoplada
	Entrada Servo ON translación	SOT	Entrada optoacoplada
	Entrada Hold (mantenimiento) trans.	HT	Entrada optoacoplada
	Entrada Alarm Reset (borrado) trans.	ART	Entrada optoacoplada
	Entrada Emergency translación	EMT	Entrada optoacoplada
	Baliza roja	BR	Piloto luminoso color rojo
	Avanzar ventosa	A+	EV neumática
	Vacío	V+	EV neumática

Aunque se posee esta tabla de entradas y salidas, no se conoce a qué entrada o salida del autómatas corresponde cada señal de la lista, por lo que primeramente se debe resolver este problema.

El primer paso y el más sencillo es intentar deducir qué señales corresponden a los diferentes botones de la botonera de la célula. Esto puede realizarse pulsando los diferentes botones y viendo el número de la señal que se enciende en la pantalla frontal del PLC. A fin de evitar que la estación se ponga en marcha con la pulsación de cualquiera de estos botones, se debe de asegurar que el sistema no posea ningún programa cargado. Una vez comprobada la inexistencia de un programa cargado con la ayuda de la aplicación PL7, se puede proceder a averiguar las señales correspondientes a estos botones.

Al pulsar el botón de *start* se puede comprobar cómo se enciende la señal número cero de la parte superior de la primera columna de la pantalla del autómatas. De este modo, se puede deducir que esta señal es la entrada %I1.0.

A continuación se realiza lo mismo con el botón de *stop*, y se observa que al pulsar el botón se apaga la señal %I1.1 del autómatas. Se deduce pues el botón de *stop* tiene asociada la señal %I1.1 aunque invertida, puesto que permanece activa a menos que se pulse este botón.

Con el selector de *auto/man* se observa que la señal %I1.2 se activa cuando este está en la posición *man*, y se desactiva en el otro caso. Así pues, esta señal activa indica que la célula trabaja en modo manual.

De igual manera con el botón de *reset* se observa que al pulsarlo se activa la señal %I1.3, por lo que se deduce que esta es la entrada asociada a este botón.

%Q4.2	Desactiva la señal %I3.1 y activa %I1.13 y %I1.14
%Q4.3	Desactiva las señales %I1.13 y %I1.14 y activa %I3.1
%Q4.4	Sin acción
%Q4.5	Sin acción
%Q4.6	Sin acción
%Q4.7	Sin acción
%Q4.8	Sin acción
%Q4.9	Sin acción
%Q4.10	Sin acción
%Q4.11	Sin acción
%Q6.0	Se mueve el brazo robótico hacia la posición de translación 1 y al detenerse activa %I3.10
%Q6.1	Desactiva las señales %I3.7, %I3.8 y %I3.10. Al deshabilitarse vuelven a activarse %I3.7 y %I3.8, pero no %I3.10
%Q6.2	Desactiva la señal %I3.7
%Q6.3	Desactiva la señal %I3.11 y activa %I3.7 y %I3.8
%Q6.4	Desactiva las señales %I3.7 y %I3.8 y activa %I3.11
%Q6.5	Sin acción
%Q6.6	Sin acción
%Q6.7	Sin acción
%Q6.8	Sin acción
%Q6.9	Sin acción
%Q6.10	Sin acción
%Q6.11	Sin acción

En esta primera pasada, se han descubierto las señales asociadas a la baliza roja, al eje vertical y a la ventosa de vacío. Sin embargo, en cuanto a movimientos de los ejes eléctricos horizontales solo se han conseguido realizar dos de ellos, uno de “translación” hacia la posición 1, y otro de “mesa” hacia la posición 5. Al finalizar ambos movimientos, el robot produce un sonido brusco, como si intentara avanzar más y no pudiese hacerlo. Se observa además que el eje vertical no está situado justo en el centro de esta posición del almacén, sino más desplazado hacia la esquina del mismo. Esto lleva a deducir que las señales %Q2.11 y %Q6.0 no son las utilizadas para llevar al robot a esta posición, sino que seguramente deberemos combinar algunas de ellas. Por el momento parece que estas señales habilitan el movimiento de los ejes horizontales, puesto que si la señal de preparado del eje correspondiente (%I1.13 o %I3.7) está desactivada antes de la ejecución de la señal de habilitación, se activa.

Antes de realizar una segunda pasada, se desconecta del suministro de corriente la célula y se lleva el robot manualmente a otra posición con el fin de observar si los ejes realizan algún movimiento en esta ronda.

Una vez alimentada de nuevo la célula, se procede a realizar una segunda pasada del programa de control. En esta nueva pasada se descubre con sorpresa que las señales %Q2.11 y %Q6.0 no producen ahora el movimiento que en la pasada anterior provocaron, teniendo en cuenta que el robot ha sido movido a mano de la posición en la que quedó. Al analizar esta situación y compararla con la anterior, se descubre que las señales %I1.13, %I1.14, %I3.7 y %I3.8 que en la pasada anterior estaban activas no lo están ahora, con lo que finalmente se deduce que estas son las señales que habilitan y deshabilitan los movimientos de mesa y translación.

Analizando ambas situaciones de nuevo, se llega a una nueva conclusión y es que las señales %Q4.2 y %Q4.3 realizan acciones contrarias, al igual que ocurre con las señales %Q6.3 y %Q6.4. Por ejemplo, al analizar las señales que afectan a los movimientos de “mesa”, se

observa que la señal %Q4.3 desactiva el movimiento apagando las señales %I1.13 y %I1.14 y activa la señal %I3.1. Esto hace pensar que la señal %Q4.3 es la de activación de alarma puesto que impide el movimiento del robot y avisa con otra señal (%I3.1) del estado del sistema. Del mismo modo, se deduce que la señal %Q4.2 es la de borrado de alarma, al realizar acciones opuestas. Así pues, las señales %Q6.3 y %Q6.4 realizan las mismas acciones que las dos señales anteriores respectivamente, solo que en este caso afectan a los movimientos del eje de “translación”.

En lo que respecta a la señal %Q4.1, se observa que únicamente desactiva %I1.13, al igual que la señal %Q6.2 que desactiva %I3.7. Por lo tanto parece que estas son las señales de mantenimiento de los ejes “mesa” y “translación”, respectivamente.

En cambio, la activación de las señales %Q4.0 y %Q6.1 desactiva todas las señales referentes a su correspondiente movimiento (“mesa” o “translación”), por lo que se deduce que son las señales de habilitación y que estas están a nivel bajo puesto que los movimientos se inhabilitan con su activación.

Aunque en este punto se han revisado todas las señales, en lo que respecta al movimiento de los ejes horizontales apenas se ha conseguido un movimiento por parte de cada uno de ellos. Es por eso por lo que a raíz de las conclusiones anteriores, se pasa a probar combinaciones de señales para producir los movimientos deseados. La primera idea es probar combinaciones de las señales %Q2.11 y %Q6.0 con las señales de las cuales todavía no se ha descubierto su funcionamiento. De este modo, se encuentran varios nuevos movimientos como los que desplazan el brazo robótico a las posiciones de “mesa 1”, “mesa 2” y “mesa 4” al combinar %Q2.11 con otra señal. Al combinar %Q6.0 con otra señal se observa el desplazamiento del robot hacia las posiciones de “translación 1”, “translación 2” y “translación 4”. En este momento, se reafirma la deducción anterior de que las señales %Q2.11 y %Q6.0 en realidad no llevan a la posición de “mesa 5” y “translación 1” respectivamente, puesto que para llegar a la posición de “translación 1” se ha necesitado la combinación de %Q6.0 con otra señal. Además a diferencia del movimiento anterior generado por una única señal, al finalizar el movimiento el robot para de forma suave sin ningún sonido brusco, y centrado sobre las posiciones del almacén que comparten la componente “translación 1”.

Sin embargo, todavía es necesario averiguar que combinaciones llevan a las posiciones de “mesa 3”, “mesa 5”, “translación 3” y “translación 5”. Se concluye que el movimiento de “translación 6” también será necesario dado que el puesto de recogida del bote se encuentra en la posición “mesa 5, translación 6”. Como ya se han probado combinaciones de dos señales y aun faltan movimientos por descubrir, se pasa a probar con combinaciones de tres señales o más, si fuera necesario.

Después de un tiempo probando todas estas combinaciones, todas las posiciones del almacén incluyendo el puesto de recogida del bote son accesibles por el brazo robótico. La tabla de entradas y salidas queda pues de la siguiente manera:

Tabla 4. Relación final de entradas y salidas

	ACCIÓN	SÍMBOLO	SEÑAL/SEÑALES	
ENTRADAS	Pulsador Start	m	%I1.0	
	Pulsador Stop	p	%I1.1	
	Selector auto/man	A/M	Auto: %I1.2; Man: %I1.2	
	Pulsador Reset	r	%I1.3	
	Detector de botes	dp1	%I1.6	
	Detector de botes tapados	dp2	%I1.7	
	Ventosa arriba	a0	%I1.4	
	Ventosa abajo	a1	%I1.5	
	Salida 1 mesa	S1M		
	Salida 2 mesa	S2M		
	Salida 3 mesa	S3M		
	Salida 4 mesa	S4M		
	Salida 5 mesa	S5M		
	Salida Ready (preparado) mesa	SRM	%I1.13	
	Salida Position (posición) mesa	SPM	%I1.14	
	Salida Busy (ocupado) mesa	SBM	%I1.15	
	Salida Home (origen) mesa	SHM	%I3.0	
	Salida Alarm (alarma) mesa	SAM	%I3.1	
	Salida 1 translación	S1T		
	Salida 2 translación	S2T		
	Salida 3 translación	S3T		
	Salida 4 translación	S4T		
	Salida 5 translación	S5T		
	Salida Ready (preparado) translación	SRT	%I3.7	
	Salida Position (posición) translación	SPT	%I3.8	
	Salida Busy (ocupado) translación	SBT	%I3.9	
	Salida Home (origen) translación	SHT	%I3.10	
	Salida Alarm (alarma) translación	SAT	%I3.11	
	SALIDAS	Entrada 0 mesa	E0M	
		Entrada 1 mesa	E1M	%Q2.3*%Q2.11
Entrada 2 mesa		E2M	%Q2.4*%Q2.11	
Entrada 3 mesa		E3M	%Q2.3*%Q2.4*%Q2.11	
Entrada 4 mesa		E4M	%Q2.5*%Q2.11	
Entrada 5 mesa		E5M	%Q2.3*%Q2.5*%Q2.11	
Entrada 6 mesa		E6M		
Entrada 7 mesa		E7M		
Entrada Enable (habilitación) mesa		EM	%Q4.0	
Entrada Servo ON mesa		SOM	%Q2.11	
Entrada Hold (mantenimiento) mesa		HM	%Q4.1	
Entrada Alarm Reset (borrado) mesa		ARM	%Q4.2	
Entrada Emergency mesa		EMM	%Q4.3	
Entrada 0 translación		E0T		
Entrada 1 translación		E1T	%Q4.4*%Q6.0	
Entrada 2 translación		E2T	%Q4.5*%Q6.0	
Entrada 3 translación		E3T	%Q4.4*%Q4.5*%Q6.0	
Entrada 4 translación		E4T	%Q4.6*%Q6.0	
Entrada 5 translación		E5T	%Q4.4*%Q4.6*%Q6.0	
Entrada 6 translación		E6T	%Q4.5*%Q4.6*%Q6.0	
Entrada 7 translación		E7T		
Entrada Enable (habilitación) trans.		ET	%Q6.1	
Entrada Servo ON translación		SOT	%Q6.0	
Entrada Hold (mantenimiento) trans.		HT	%Q6.2	
Entrada Alarm Reset (borrado) trans.		ART	%Q6.3	
Entrada Emergency translación		EMT	%Q6.4	
Baliza roja		BR	%Q2.0	

	Avanzar ventosa	A+	%Q2.1
	Vacío	V+	%Q2.2

Las entradas que corresponden a las posiciones de “mesa” y “translación” no son visibles puesto que están gestionadas por el driver de cada uno de los ejes horizontales y no se muestran en la pantalla frontal del PLC. Además en el listado aparecen ocho señales de salida por cada eje, sin embargo para el correcto funcionamiento del autómatas no son necesarias más señales de las indicadas en la tabla.

Mientras se realizaban estas pruebas de movimiento, se observa que estas órdenes de los ejes verticales funcionan por impulso. Esto quiere decir que no es necesario mantener continuamente el estado de la señal de movimiento, sino que se podría desactivar dicha señal o cambiar su estado, y el brazo robótico continuaría su movimiento hasta llegar a la posición señalada. Otra observación realizada es que la señal %I1.15 se encendía en la pantalla frontal del PLC cuando el movimiento de “mesa” estaba activo, del mismo modo que la señal %I3.9 con el movimiento de “translación”. Esto lleva a deducir que estas dos señales son las de ocupado de los ejes de “mesa” y “translación” respectivamente.

En este punto, ya es posible comenzar con la programación del autómatas, puesto que ya se conoce el valor de cada una de las señales de entrada y salida que se usarán en este proceso.

4.2 Programación del autómatas

Antes de pasar a programar el autómatas, se meditó tanto sobre el funcionamiento básico del mismo como sobre los requisitos necesarios del programa. De este modo se llegó a una serie de consideraciones iniciales:

- Lo único que varía en el ciclo de almacenamiento de un bote con respecto a otro es la posición de destino del bote, que irá en función del número de botes que ya haya almacenados. Por ejemplo, si el almacén ya tiene dos botes depositados, el siguiente bote a depositar deberá dejarse sobre la posición número tres.
- Es posible, por tanto, que exista una parte del programa común a todos los ciclos de almacenamiento, variando únicamente la parte posterior a la comprobación del número de botes, la cual se encargará de llevar el bote a su posición de destino. A partir de este punto, el programa también puede ser común a todos los ciclos puesto que únicamente resta depositar el bote en la posición correspondiente.
- Se deberá manejar un contador para saber cuántos botes hay depositados ya en el almacén, aumentando en uno la cuenta cada vez que se deposite un bote.
- El contador deberá ser puesto a cero cuando el almacenamiento de los veinticinco botes haya finalizado y el almacén haya sido vaciado.
- Se deberá usar también uno o varios comparadores para saber cuántos botes hay en un momento dado y poder decidir así la posición de destino del bote recogido.
- Las señales más importantes en cuanto al movimiento de los ejes horizontales se refiere deben ser las señales de ocupado y/o preparado de cada eje.

En este momento es posible comenzar con la programación del autómatas, sin embargo se toma la decisión de no comenzar por una programación que incluya el almacenamiento de los



veinticinco botes puesto que aunque ya se conoce cómo mover el robot a todas las posiciones del almacén y el origen, aún no se ha probado un ciclo completo de almacenamiento que incluya la recogida del bote de la posición de origen y su posterior depósito en la posición destino del almacén.

De este modo, se comienza por un esbozo del programa en GRAFCET, el cual deberá realizar el ciclo completo de almacenamiento citado en el párrafo anterior. El uso de la palabra esbozo es intencionado puesto que todavía no se van a incorporar todas las consideraciones iniciales. Por ejemplo, no se va a incorporar todavía ningún contador ni comparador, ni se va a hacer uso de la baliza de alarma.

Con la relación de señales de entrada y salida de la Tabla 4 la creación de este esbozo se realiza de manera sencilla. Sin embargo surge una duda cuya solución deberá ser probada en el programa. Se desconoce qué señal o señales de entrada pueden servir para atestiguar que los ejes eléctricos horizontales han llegado a la posición indicada, aunque se considera que podrían ser útiles las señales de preparado y/o ocupado de cada eje.

El aspecto de este primer esbozo queda reflejado en la siguiente ilustración.

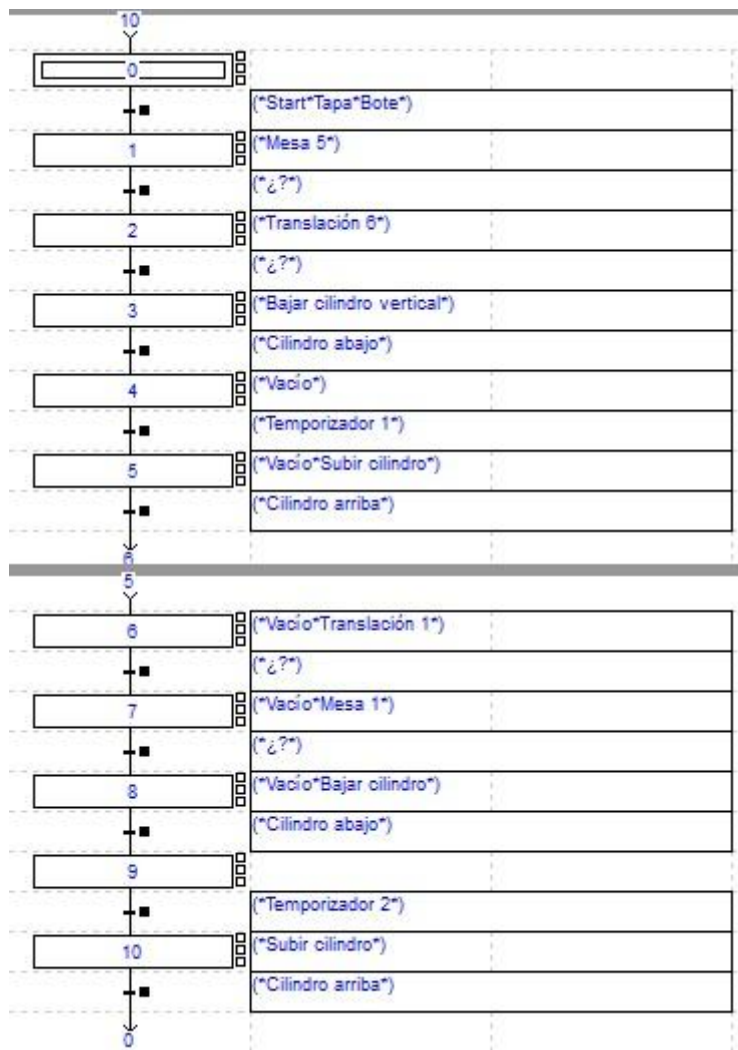


Ilustración 18. Esbozo inicial del programa.

En la Ilustración 18 se muestra el desconocimiento en la adecuada utilización de la señal que verifique el cada eje horizontal ha llegado a la posición correspondiente. Se muestra también el uso de dos temporizadores, establecidos a 0.5 segundos cada uno.

En este momento, es posible comenzar a probar qué señal es la apropiada para la verificación del movimiento que se desea. Puesto que cuando cualquiera de los ejes horizontales está en movimiento se activa una señal de ocupado, la primera idea es introducir en estas transiciones esta señal, pero inversa. Es decir, se considerará que el eje ha llegado a la posición indicada cuando esta señal este desactivada.

Una vez completado el programa con estas señales (%I1.15 para el eje “mesa” y %I3.9 para el eje “translación”) se pasa a comprobar si el resultado obtenido es el esperado. Sin embargo no ocurre así. Todas aquellas acciones que están relacionadas con el movimiento de los ejes horizontales no han surtido ningún efecto, y estos siguen en el mismo sitio. En cambio, todo el resto de acciones como el vacío de la ventosa y el movimiento del eje neumático vertical han realizado lo esperado.

Después de este resultado negativo, se busca en la lista de entradas otra señal que pueda servir para este propósito. Rápidamente, se cae en la cuenta de la existencia de la señal de preparado señales (%I1.13 para el eje “mesa” y %I3.7 para el eje “translación”). Así pues, se pasa a completar el programa de nuevo con esta señal. Es decir, el eje habrá llegado a la posición requerida cuando esta señal este activa. No obstante, con desilusión se obtiene el mismo resultado que en la prueba anterior.

En este punto, se debe buscar el motivo por el cual estas señales no obtienen el funcionamiento esperado, por el que los ejes horizontales no muestran ningún movimiento. Aún sabiendo que el programa realizado hasta el momento no funciona, se repite su ejecución prestando especial atención a la pantalla frontal del PLC donde se muestran las entradas y salidas activas. En este momento se descubre el motivo del error. La señal de ocupado de los ejes horizontales únicamente se activa cuando estos están en movimiento. De este modo, si el eje no empieza a moverse esta señal nunca se activa, con lo cual, al tener como condición esta señal inversa, las transiciones que la contenían saltan directamente a la siguiente etapa, siendo imposible activar el movimiento que se desea en cada eje. Lo mismo ocurre con la señal de preparado, pues esta permanece activa mientras el robot no se mueva.

Así pues, a raíz de estos descubrimientos se deduce que el estado de estas señales en sí mismo no es el que determina si el brazo robótico ha llegado a su posición de destino, sino que probablemente sea el cambio de estado de dichas señales. Es decir, si se toma la señal de ocupado y se tiene en cuenta que esta señal está activa mientras el eje correspondiente se mueve, la deducción adecuada es tener en cuenta el momento de desactivación de esta señal, o el momento en que esta señal pasa de estar activa a desactivarse. Sin duda, este será el momento en el que el eje haya llegado a la posición deseada. De este modo, la señal de ocupado se deberá evaluar por su flanco descendente (de activo a inactivo) y la señal de preparado por su flanco ascendente (de inactivo a activo).

Dicho esto, se pasa a modificar el programa de nuevo, incluyendo esta vez en las transiciones que antes habían dado problemas los contactos por flanco descendente de las señales de ocupado del eje correspondiente, así como los contactos por flanco ascendente de las señales de preparado de dichos ejes, de forma que cuando ocurra cualquiera de estos dos eventos, la condición se haga cierta, activando así la transición. En la siguiente ilustración se

puede observar el contenido de una transición encargada de controlar un movimiento de eje “mesa”.

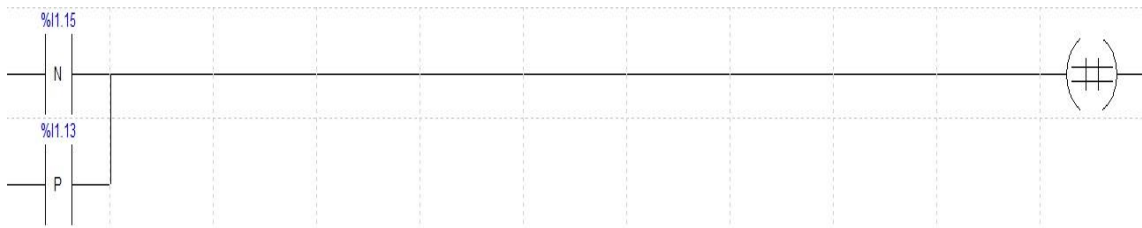


Ilustración 19. Contenido de la transición %X7->%X8

Es en este momento cuando se obtiene el primer resultado correcto, puesto que el ciclo de almacenamiento de un bote se ha realizado con éxito desde el puesto de recogida de botes a la posición de almacenamiento “mesa 1, translación 1”.

Como se puede observar en la Ilustración 18, la orden de movimiento de cada eje se activa de forma separada en las etapas del programa, es decir, la etapa 1 activa la orden de “Mesa 5” y la etapa 2 activa la orden de “Translación 6”. Si bien se ha considerado activar estas dos órdenes en una misma etapa se cae en la cuenta que esto no es posible, puesto que la condición de fin de movimiento en la transición inmediatamente posterior a esta etapa, difícilmente será cierta si se debe evaluar a la vez el flanco descendente de ambas señales de ocupado, o el flanco ascendente de ambas señales de preparado. Esto es así porque prácticamente nunca los dos ejes horizontales dejarán de moverse a la vez, con lo cual la señal de ocupado de cada eje no provocará un flanco descendente al mismo tiempo que el otro, del mismo modo que ocurre con las señales de preparado.

Así pues, el GRAFCET del programa que realiza un ciclo de almacenamiento de un bote queda de la siguiente manera:

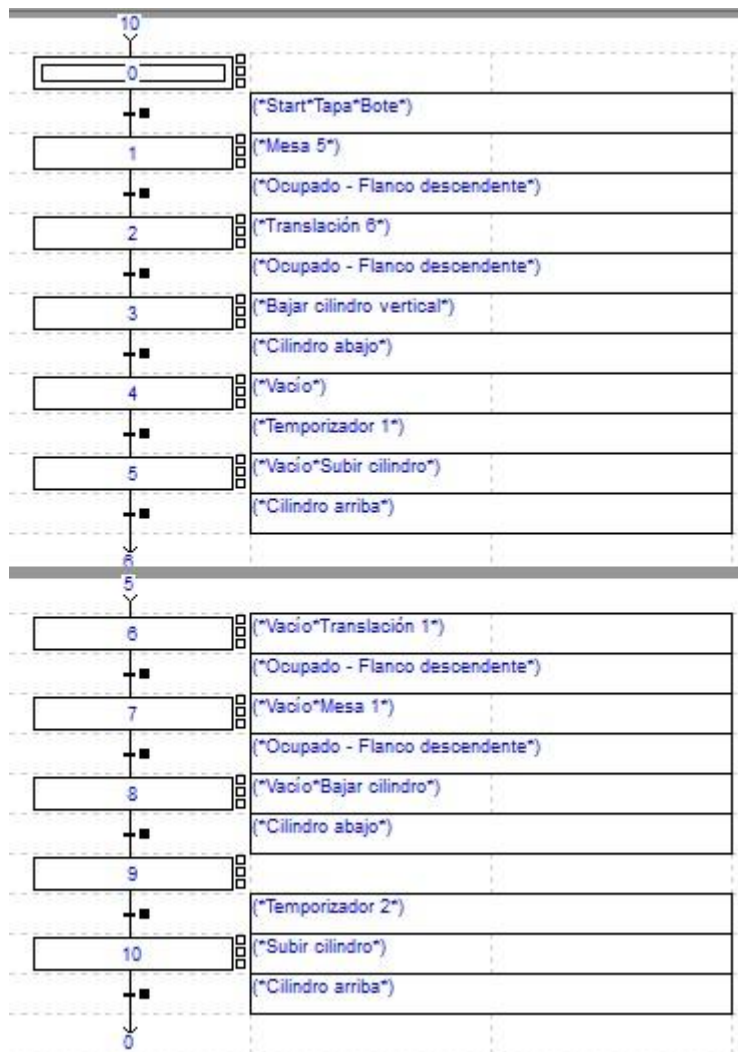


Ilustración 20. GRAFCET de un ciclo de almacenamiento

El siguiente paso, posterior al éxito obtenido en este caso, es probar a incluir un contador y uno o varios comparadores, además de ampliar el programa en GRAFCET para el caso en el que se desee almacenar más de un bote. En concreto, la siguiente prueba se realizará con tres botes.

En este momento se descubre un problema que antes no había surgido. El día en el que se debía comenzar con la programación del autómat para que almacenara tres botes, se pretendía volver a probar previamente el programa con un único ciclo de almacenamiento. Se introdujo de nuevo el programa que previamente había funcionado en la célula y se puso en marcha la misma. Sin embargo, al pulsar el botón de *start* se reprodujo el mismo error en el movimiento de los ejes horizontales que se observó con anterioridad, por el cual ninguno de estos ha sido realizado con éxito. Para corregir el error se revisan las transiciones relativas a los movimientos de los ejes para comprobar que las condiciones existentes en ellas son las correctas, es decir, aquellas que tienen en cuenta el flanco de bajada de la señal de ocupado de cada uno de los ejes. Efectivamente las transiciones se encuentran en el estado correcto. Puesto que el error no está en el programa, se revisa el estado de las señales de entrada y salida en la pantalla frontal del PLC y gracias a esto se descubre el motivo de este nuevo error. En la ejecución del día anterior, la cual había demostrado que el ciclo de almacenamiento de un bote podía realizarse de manera correcta, las señales de origen de cada uno de los ejes se encontraban activas (%I3.0 para el eje



“mesa” y %I3.10 para el eje “translación”). El motivo por el que en esta ocasión se encuentren apagadas, el cual se reafirmará posteriormente, se debe a la desconexión del equipo de la corriente eléctrica en el momento se abandona el aula, puesto que se desconectan todos los interruptores magnetotérmicos del cuadro eléctrico de la misma.

Ya que la activación de las señales de habilitación de los ejes (%Q2.11 para el eje “mesa” y %Q6.0 para el eje “translación”) enciende las señales de origen de cada eje, además de llevar el brazo robótico a una posición que se considera inicial o de origen, se crea un sencillo programa de inicialización para la célula. Este programa es idéntico al de la Ilustración 17, con la salvedad que en la etapa en la que se analizaban las señales, ahora se han introducido las señales de habilitación de ambos ejes para su activación. Esto se muestra en las siguientes dos ilustraciones.

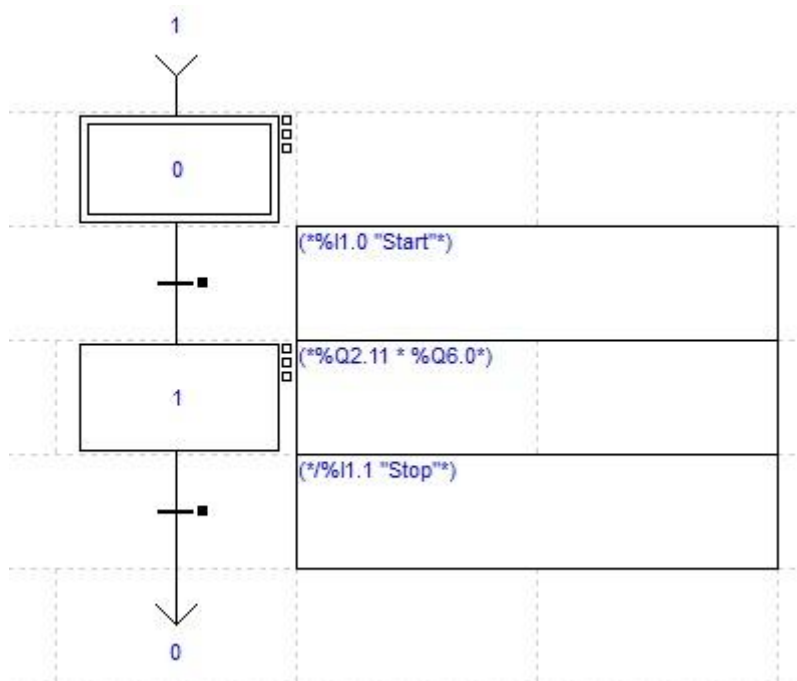


Ilustración 21. GRAFCET de inicialización

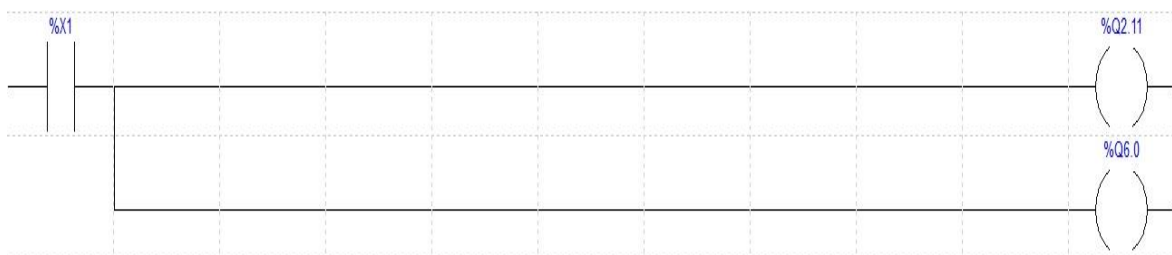


Ilustración 22. Activación de las señales de habilitación

Una vez obtenido el programa de inicialización, se ejecuta para posteriormente comprobar si funciona de nuevo el programa con un ciclo de almacenamiento. Tras comprobar que así es, se guarda este programa de inicialización para posteriores usos.

El fichero generado por la aplicación PL7 que contiene el programa de inicialización tiene el nombre de *inicialización_ejes.stx*.

Tras esto, se pasa a ampliar el programa para que realice tres ciclos de almacenamiento completos. Para ello se debe incluir un contador y tres comparadores que revisen el estado de este con el fin de averiguar cuántos botes hay ya almacenados y por tanto discernir en qué posición debe depositarse el siguiente bote.

En la siguiente ilustración se muestra el contador utilizado en la programación de este programa.

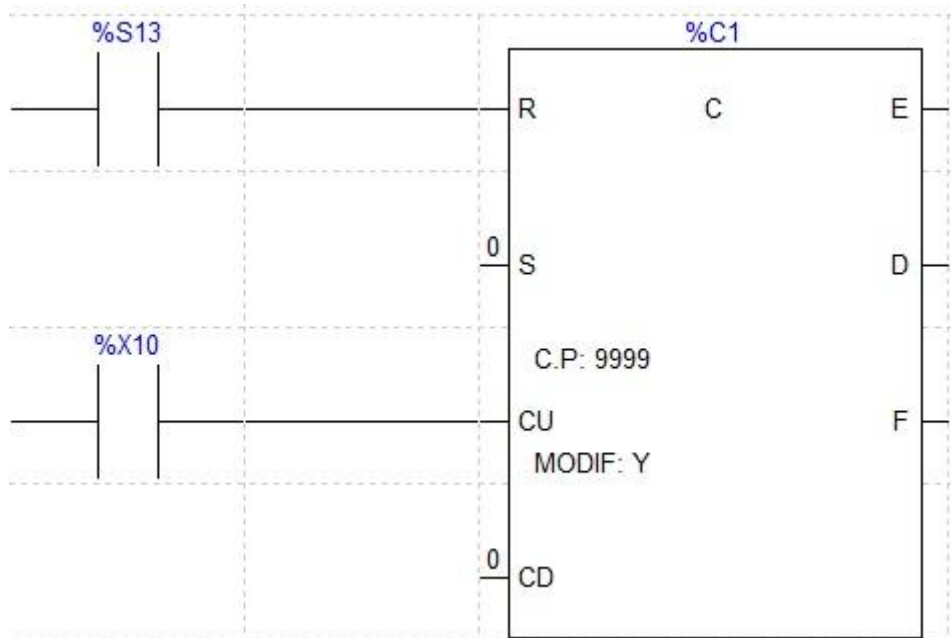


Ilustración 23. Contador de tres botes

Como se puede observar en la Ilustración 23, en el contador se utilizan dos de sus entradas. La primera entrada, indicada con la letra 'R', es la entrada de *reset* o puesta a cero. Se ha asociado esta entrada a la señal de sistema %S13, puesto que esta señal se activa cuando el autómatas pasa al estado de parada. La tercera entrada, indicada como 'CU', es la entrada de incremento del contador. Esta se ha asociado con la etapa número diez, que es la última de cada ciclo de almacenamiento. Por tanto, cada vez que el programa vaya a acabar un ciclo, se incrementará el contador en una unidad, mientras que al desconectar el autómatas este pasará a valer cero de nuevo.

De este modo se pasa a ampliar el GRAFCET anterior que realizaba un único ciclo de almacenamiento para que realice tres. Como se indica en las consideraciones iniciales, al principio de esta sección, existe una parte del programa común a todos los ciclos de almacenamiento. Esto puede observarse en la siguiente ilustración.

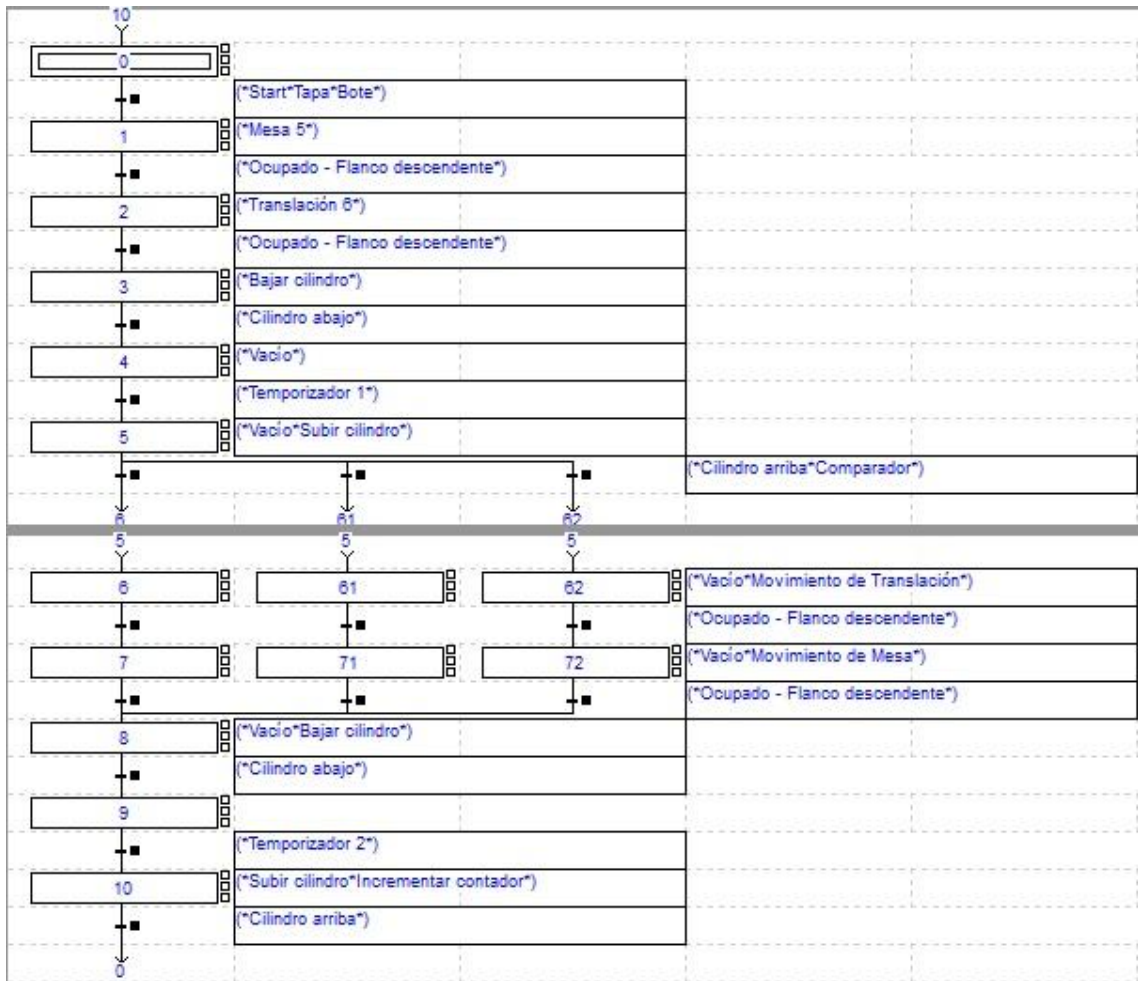


Ilustración 24. GRAFCET de tres ciclos de almacenamiento

En la Ilustración 24 se puede observar como en las transiciones entre las etapas cinco y seis, cinco y sesenta y uno, y cinco y sesenta y dos existe un comentario explicativo donde se lee “Cilindro arriba*Comparador”. Esto es el contenido de estas transiciones y un ejemplo de ello se muestra en la siguiente ilustración.

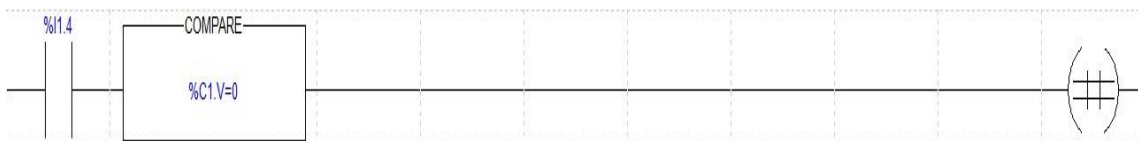


Ilustración 25. Contenido de la transición %X5->%X6

La Ilustración 25 muestra el contenido de la transición entre las etapas cinco y seis. La señal %I1.4 de esta ilustración corresponde a la señal que indica que el cilindro vertical está en posición superior. Lo más importante, sin embargo es la figura que podemos observar a la derecha de esta. Esta figura es un comparador y en su interior se observa la comparación “%C1.V=0”. %C1.V es la variable que se utiliza para acceder al valor almacenado en el contador, mientras que el número cero es el que indica a la célula que debe almacenar el primer bote puesto que el almacén se encuentra vacío. Debido a que el comparador contiene esta comparación, este se activará cuando dicha condición sea cierta, permitiendo e indicando que se debe llevar el primer bote a su posición de almacenamiento final.

De este modo, una vez completado el GRAFCET de tres ciclos de almacenamiento, se procede a la prueba del mismo. El programa deberá depositar los botes en las posiciones “mesa 1, translación 1”, “mesa 2, translación 1” y “mesa 3, translación 1”. Para comprobar que esto es así, se pone en marcha la célula y se introduce el programa en la misma. Se observa como el programa realiza la secuencia de almacenamiento de los tres botes de forma correcta, llevando cada bote a su posición designada. Además al pasar el autómat a modo de parada y volverlo a poner en ejecución se puede comprobar cómo el contador se pone a cero de nuevo, puesto que al empezar de nuevo un ciclo de almacenamiento este deposita el bote en la primera posición asignada.

En este punto se piensa que ya es posible realizar una programación del autómat que incluya los veinticinco ciclos de almacenamiento a fin de llenar el almacén, puesto que se ha sido capaz de almacenar tres botes sin problemas. Debido a esto, se considera que el primer modelo de almacenamiento completo que puede realizarse es un modelo en zigzag en el cual las filas que comparten el componente translación sean rellenas de izquierda a derecha para las filas 1, 3 y 5, y de derecha a izquierda para las filas 2 y 4. Esto es así porque el programa de tres ciclos de almacenamiento depositaba los tres botes sobre la primera fila que comparte el componente de “Translación 1”. El modelo de almacenamiento se muestra en la siguiente tabla, la cual, al igual que en la Tabla 1, está representada en la perspectiva que se muestra en la Ilustración 2.

Tabla 5. Modelo de almacenamiento en zigzag

	Translación 5	Translación 4	Translación 3	Translación 2	Translación 1
Mesa 5	25	16	15	6	5
Mesa 4	24	17	14	7	4
Mesa 3	23	18	13	8	3
Mesa 2	22	19	12	9	2
Mesa 1	21	20	11	10	1

Se procede pues a la programación del GRAFCET que realice el almacenamiento de los veinticinco botes, de manera análoga a como se programó la versión anterior que almacenaba tres botes.

Sin embargo se encuentra un problema antes de empezar a ello. El programa PL7 para la programación de autómatas posee once columnas, como puede verse en la siguiente ilustración.



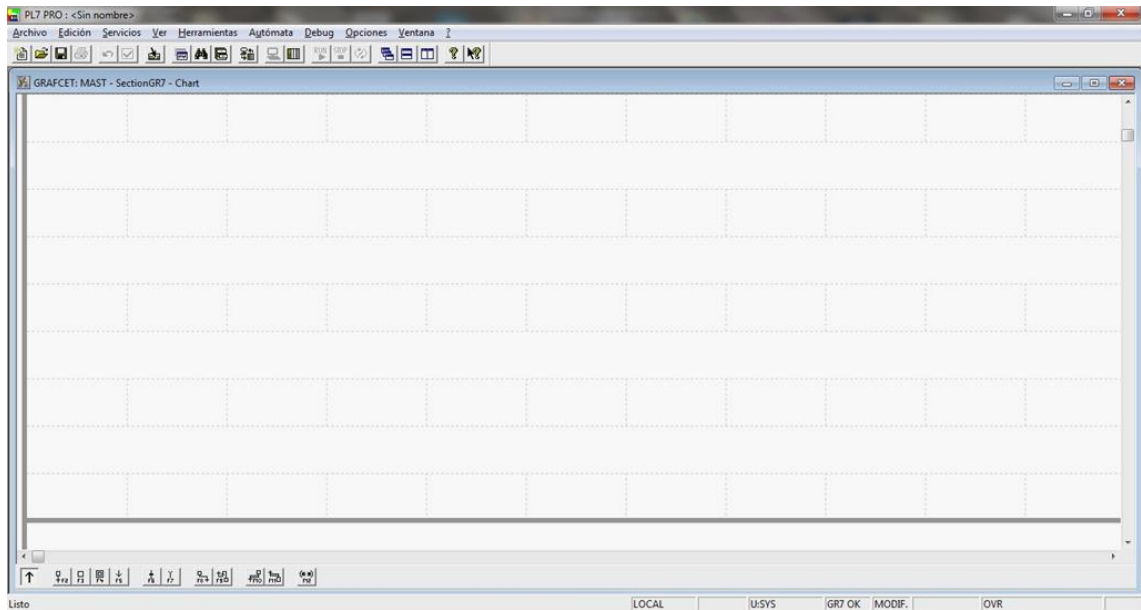


Ilustración 26. Interfaz de programación del PL7

De esta manera, si se sigue programando igual que se ha hecho hasta ahora, solamente se podrá realizar el almacenamiento de once botes, puesto que la interfaz no amplía el número de columnas en caso de necesitar más.

Este problema se soluciona del siguiente modo. El uso de todas las columnas de la interfaz comienza a partir de las transiciones posteriores a la etapa 5, tal y como se muestra en la Ilustración 24. Esto es debido a que estas transiciones son las que comprueban el número de botes ya almacenados para discernir la posición del siguiente bote a depositar (véase Ilustración 25). Así pues, se utilizan las diez primeras columnas normalmente, de modo que se compruebe el almacenamiento de hasta nueve botes, y se complete el almacenamiento del décimo. En la columna undécima, la transición a programar deberá contener una condición que recoja las condiciones de las transiciones que no caben en la interfaz, como se muestra en la siguiente ilustración.



Ilustración 27. Contenido de la transición %X5->%X99

De este modo, en otra página de la interfaz y disponiendo de nuevo de todas las columnas, es posible continuar con la programación del almacenamiento del resto de botes. Esto se realiza utilizando una etapa que recoja la transición anteriormente citada, la cual no realiza ninguna acción, y a partir de aquí volver a analizar el número de botes ya almacenados en otras transiciones para continuar con su almacenamiento del mismo modo que se estaba realizando antes. A fin de completar el proceso de modo correcto, otra etapa que no realice ninguna acción debe recoger todas las transiciones que indican el fin del movimiento del eje “mesa”. Posteriormente a esta, se debe introducir una transición con una condición que sea cierta para todos los movimientos realizados. En este caso, se utilizan las señales de preparado de ambos

ejes (%I1.13 para el eje “mesa” y %I3.7 para el eje “translación”) debido a que en este momento los ejes deberán estar ya en la posición que se les ha ordenado.

Sin embargo volvemos a encontrarnos con el mismo problema, puesto que la siguiente página de la interfaz dispone del mismo número de columnas, de modo que hasta este punto solo es posible almacenar veinte botes. Aun así, es posible aplicar el mismo método de nuevo a fin de disponer de diez columnas más, siendo la undécima columna la que contendrá la transición con la condición adecuada. El contenido de esta transición se muestra en la siguiente ilustración.



Ilustración 28. Contenido de la transición %X99->%X59

Prosiguiendo pues de esta manera, es posible programar el autómata de modo que almacene los veinticinco botes.

Debido a que la explicación del método seguido para la resolución de este problema puede resultar algo engorrosa, se proporcionan las siguientes ilustraciones a fin de aclarar dicho método.

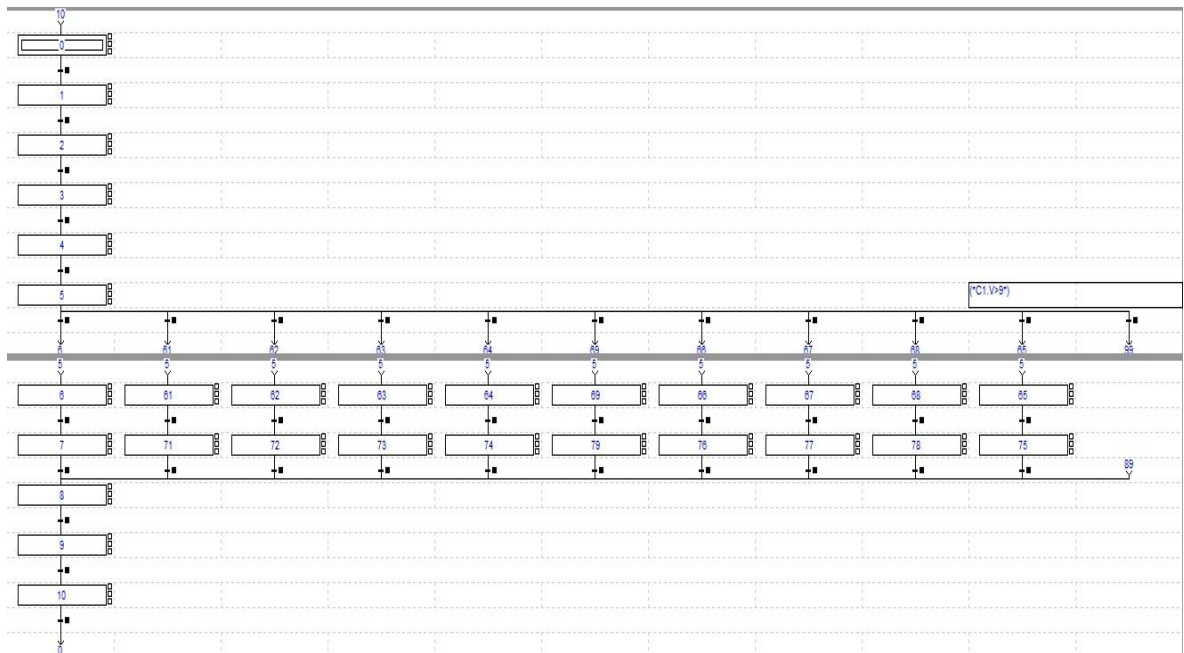


Ilustración 29. GRAFCET de veinticinco ciclos de almacenamiento (Parte 1)

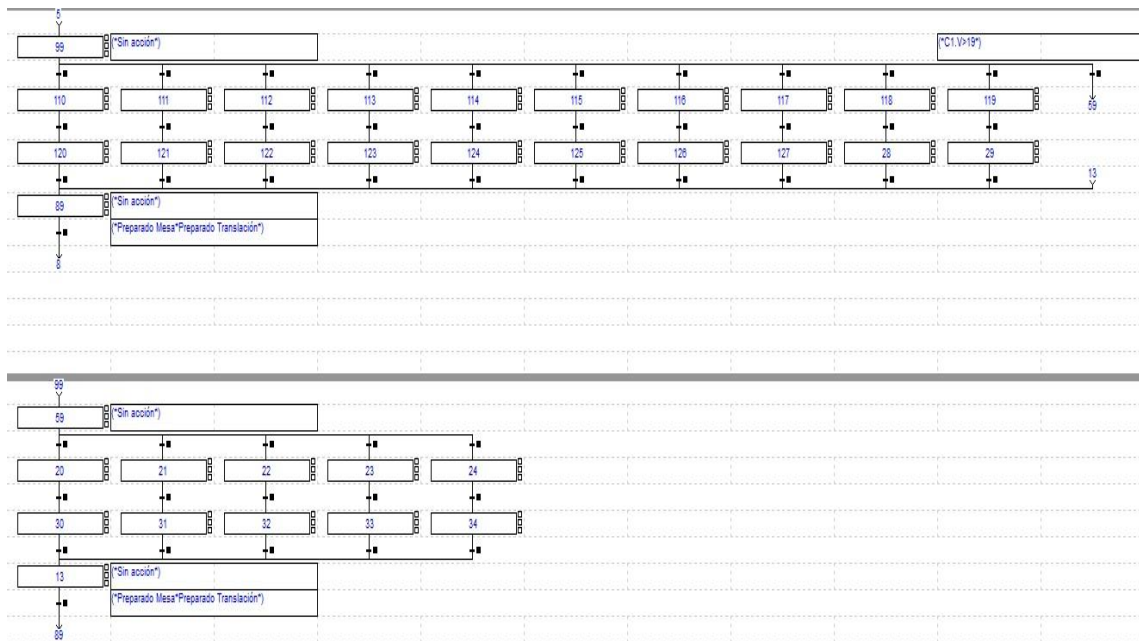


Ilustración 30. GRAFCET de veinticinco ciclos de almacenamiento (Parte 2)

En las Ilustraciones 29 y 30 se muestra el resultado del método empleado para solucionar el problema que arroja la interfaz de programación PL7. En ellas se han incluido unos comentarios a fin de aclarar lo explicado en esta parte de la memoria.

Una vez solucionado este problema, se pasa a comprobar el correcto funcionamiento del programa de veinticinco ciclos de almacenamiento. El proceso empieza desarrollándose de forma correcta, sin embargo ocurre un fallo en el momento se va a depositar el quinto bote. En la etapa 64, se activan las señales necesarias para que el eje horizontal de translación retroceda hasta la posición “Translación 1”, de forma que el bote ya esté sobre la posición de destino. Sin embargo el depósito del bote no llega a ocurrir ya que el cilindro vertical permanece en su posición superior y la ventosa sigue produciendo succión. Mientras permanece esta situación, se revisa el estado del programa, para descubrir el porqué de este error. Gracias a que en el programa PL7 se muestra la evolución del GRAFCET marcando en negro las etapas activas, se descubre que el programa se ha quedado estancado en la etapa 74. Esto se debe claramente a que la condición de la transición posterior a esta etapa no se ha hecho cierta. El contenido de esta transición es el mismo que el mostrado en la Ilustración 19. El motivo por el cual esta no se activa es que el eje “mesa” no debe de moverse en esta ocasión para depositar el bote debido a que ya se encuentra en la fila correspondiente a “mesa 5”, al ser esta la componente “mesa” de la posición de recogida de botes. Esto ocasiona que las señales de ocupado y preparado no varíen de estado, lo que provoca que no se detecten los flancos de bajada y subida de cada señal, respectivamente.

La resolución de este nuevo error es sencilla en este caso, puesto que el problema es similar al que se dio cuando se programaba el GRAFCET de un ciclo de almacenamiento. En este caso, se opta por introducir como condición en esta transición la señal de preparado, de modo que cuando el programa llegue a este punto, se garantice que continúa con su ejecución ya que esta señal estará activa. Esto se muestra en la ilustración que viene a continuación.



Ilustración 31. Contenido de la transición problemática una vez corregido

Antes de pasar a la prueba de funcionamiento, en un momento de reflexión, se cae en la cuenta de que con seguridad, este error se volverá a reproducir en todas las ocasiones en las que se vaya a depositar un bote en una posición cuya componente “mesa” coincida con la de la posición de recogida de botes, como se ha comentado anteriormente. Esto ocurre al depositar los botes quinto, sexto, decimoquinto, decimosexto y vigesimoquinto. Así pues, se pasa a modificar también del mismo modo las transiciones posteriores a las etapas 75, 124, 125 y 34.

Una vez realizadas dichas modificaciones, se ejecuta el programa para comprobar su funcionamiento. Esta vez sí que llega a almacenarse correctamente el quinto bote, sin embargo el autómatas vuelve a quedarse estancado en el momento se presiona el botón de *start* para comenzar el sexto ciclo de almacenamiento. Utilizando de nuevo el programa PL7 se verifica que el programa se ha quedado parado en la etapa 1 en esta ocasión. El motivo es el mismo que antes: el eje “mesa” no debe moverse de la posición “mesa 5” para llegar a la posición de recogida de bote puesto que ya se encuentra ahí.

La solución, sin embargo, no es la misma, dado que en esta ocasión la transición conflictiva es común a todos los ciclos de almacenamiento por lo que no se puede poner como condición de la misma únicamente la señal de preparado del eje “mesa”. Se deduce de forma similar a anteriormente que esta situación se dará cuando se hayan almacenado los botes número cinco, seis, quince y dieciséis.

La solución tomada se muestra en la siguiente ilustración.



Ilustración 32. Contenido de la transición %X1->%X2 en el modelo zigzag

Como se puede observar en la Ilustración 32, se ha optado por incluir cuatro comparadores para detectar los casos en los que haya almacenados cinco, seis, quince y dieciséis botes. Cuando ocurra cualquiera de estas situaciones, el comparador correspondiente se activará, provocando así que la condición de la transición sea cierta aunque no se produzca ningún cambio en las señales de ocupado y preparado, asegurando de este modo la evolución del programa.

Con motivo de la existencia de estas situaciones conflictivas en el proceso de almacenamiento de los botes, se decide inhibir los movimientos correspondientes al eje “mesa” en estas situaciones de conflicto puesto que dicho movimiento no es necesario. De este modo, para las etapas 74, 75, 124, 125 y 34, se eliminan los contactos correspondientes que activaban el movimiento de “mesa” en la sección *Post* del PL7. Se plantea hacer lo mismo para las situaciones en las que el movimiento “mesa” sea innecesario en el momento de recogida del bote de su posición de origen. Esto ocurre en la etapa 1, cuando existen 5, 6, 15 y 16 botes almacenados. Debido a que no se pueden eliminar los contactos correspondientes a la etapa 1 en la sección *Post* del mismo modo que antes, puesto que esta etapa es genérica para todos los ciclos de almacenamiento, se opta por la opción que se muestra en la siguiente ilustración.

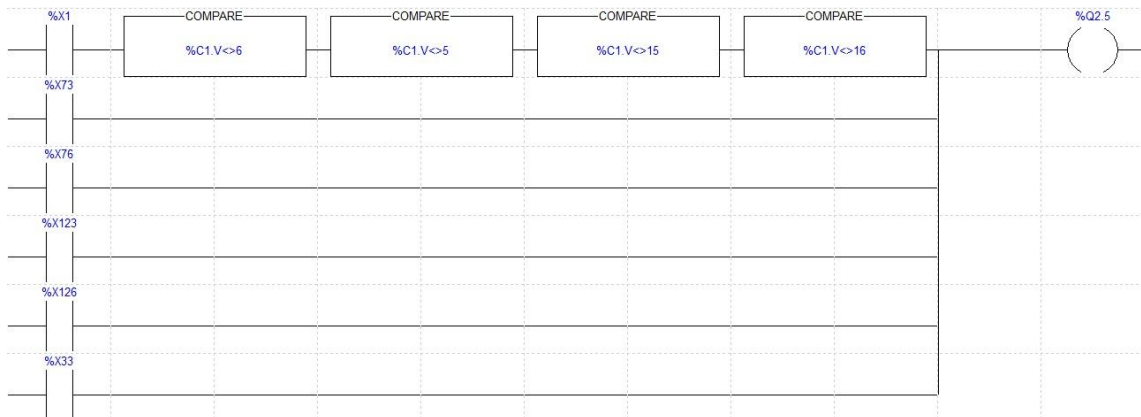


Ilustración 33. Inhibición del movimiento "mesa" de recogida en el modelo zigzag

Como puede observarse en la Ilustración 33, se han añadido al diagrama de contactos que activa la salida %Q2.5 (una de las correspondientes al movimiento “mesa” de recogida de botes) cuatro comparadores para comprobar que no haya 5, 6, 15 o 16 botes almacenados en el momento de recogida de otro bote, de modo que en estos casos se realice la acción correspondiente. En caso contrario, la activación de esta salida por parte de la etapa 1 será inhibida.

En este momento, una vez realizadas las modificaciones pertinentes, se procede a probar el automatismo programado. Finalmente se tiene éxito y el ciclo de almacenamiento de los veinticinco botes se realiza correctamente.

Sin embargo el programa no está completo puesto que se debe de añadir la activación de la baliza roja de forma fija cuando se hayan almacenado los veinticinco botes, además de su activación de forma intermitente en el caso que no se detecte un bote, o el bote no tenga tapa.

De este modo, se pasa a completar el programa mediante unas transiciones que controlen estos eventos de manera posterior al inicio de cada ciclo. En caso de que no haya un bote en la posición de origen o no esté tapado, se activa la baliza roja de forma intermitente utilizando para ello la señal del sistema %S6, que se activa y desactiva de forma intermitente cada segundo. Para desactivar esta alarma se deberá colocar un bote tapado en el origen de recogida y pulsar el botón de *reset*. En caso que se haya completado el almacenamiento de todos los botes, se activa la baliza roja de forma fija, debiendo vaciar manualmente el almacén y pulsar el botón de *reset* para desactivar esta alarma. En este momento es cuando se pone a cero el contador, para comenzar un nuevo ciclo de almacenamiento. Esto puede observarse en la siguiente ilustración.

En este punto se plantea la creación de un segundo modelo de almacenaje para compararlo con el anterior. Se considera frente a otras opciones, el uso de un modelo en espiral, cuya dirección discurra empezando por el exterior hacia el interior de la misma. La tabla que se muestra a continuación muestra el orden de almacenaje de los botes siguiendo este modelo. Al igual que sucede en las Tablas 1 y 5, está representada en la perspectiva que se muestra en la Ilustración 2.

Tabla 6. Modelo de almacenamiento en espiral

	Translación 5	Translación 4	Translación 3	Translación 2	Translación 1
Mesa 5	1	2	3	4	5
Mesa 4	16	17	18	19	6
Mesa 3	15	24	25	20	7
Mesa 2	14	23	22	21	8
Mesa 1	13	12	11	10	9

De esta manera, se puede comenzar con la programación de este modelo, basándose en el GRAFCET del modelo en zigzag, realizando los cambios pertinentes en el programa para pasar de un almacenamiento en zigzag a uno en espiral.

Hay dos formas posibles de realizar esto. Una de ellas es modificar las transiciones que controlan el número de botes ya almacenados a fin de cambiar el orden de almacenado para que se ajuste a este nuevo modelo. La otra es cambiar las etapas que activarán las salidas correspondientes al movimiento de los ejes horizontales en la sección *Post* del programa con el mismo fin. Sin embargo se observa que la primera solución no es viable, puesto que la limitación en el número de columnas que presenta el PL7 ha condicionado el orden en el que se ejecutan las etapas. Esto puede observarse en las Ilustraciones 27 y 28. Al contener estas transiciones las condiciones mostradas en las figuras, en el momento avanzan a la siguiente etapa, se condiciona que haya más de nueve botes para la primera, y más de diecinueve para la segunda, impidiendo que las transiciones que van a continuación contengan un número menor a los dos citados. Se opta, de este modo, por la segunda solución.

Una vez introducidos los cambios para obtener el modelo de almacenamiento en espiral, se pasa a corregir las transiciones del programa que, al igual que ocurrió con el modelo de zigzag, hacen que este se quede estancado. Esto sucede cada vez que se desea depositar un bote en una posición cuya componente “mesa” coincida con la de la posición de recogida de botes. Esta situación se dará cuando se vayan a almacenar los botes primero, segundo, tercero, cuarto y quinto, de modo que se deben modificar las transiciones posteriores a las etapas número 7, 71, 72, 73 y 74 del mismo modo que se realizó con el modelo en zigzag. El contenido de estas transiciones es el mismo que el mostrado en la Ilustración 31. También se debe modificar la transición entre las etapas 1 y 2 de manera similar a antes, esta vez en el momento en el que haya almacenados entre uno y cinco botes puesto que, como se citó antes: el eje “mesa” no debe moverse de la posición “mesa 5” para llegar a la posición de recogida de bote puesto que ya se encuentra ahí. Esto es debido a que la posición de depósito de los cinco primeros botes y la de recogida comparten la componente “mesa”, en concreto “mesa 5”.

El contenido de la citada transición se muestra en la siguiente ilustración.



Ilustración 36. Contenido de la transición %X1->%X2 en el modelo espiral

Del mismo modo que se realizó en el anterior modelo, es necesario inhibir los movimientos correspondientes al eje “mesa” en estas situaciones de conflicto puesto que no se utilizan. En este caso, los contactos que no deben de existir en la activación del movimiento “mesa” son los correspondientes a las etapas 7, 71, 72, 73 y 74. De este modo se eliminan dichos contactos de la sección *Post* del PL7. Para inhibir el movimiento de recogida en este modelo se realiza la modificación que se puede observar en la siguiente ilustración. Como en el caso anterior esta modificación debe realizarse en la activación de aquellas señales que realicen el movimiento de recogida de un bote de la posición de origen, pero en este caso cuando haya almacenados entre uno y cinco botes.



Ilustración 37. Inhibición del movimiento "mesa" de recogida en el modelo espiral

De este modo se impide que el robot realice el movimiento “mesa” innecesario cuando ya hay entre uno y cinco botes almacenados.

Una vez tomadas estas medidas el modelo en espiral tiene asegurado que no se quede estancado en ningún momento de su ejecución, de modo que siempre se garantice la evolución del programa.

Puesto que se ha partido del anterior modelo en zigzag, los procedimientos de alarma en caso de que no haya bote o no esté bien tapado además del caso en que ya se haya completado el almacenamiento ya están implementados. De este modo, el aspecto principal del GRAFCET es el mismo que el que se observa en la Ilustración 35.

Se procede en este punto a probar el programa creado de forma completa, de modo que realice los veinticinco ciclos de almacenamiento así como que demuestre el funcionamiento de

los procedimientos de alarma. Esta prueba se realiza satisfactoriamente, comprobando que todas las acciones se ejecutan según lo esperado.

En este caso, el fichero generado por la aplicación PL7 que contiene el programa creado tiene el nombre de *modelo_espiral.stx*.

5. Conclusiones

En este apartado se van a comentar las conclusiones extraídas de la programación de ambos modelos.

Como puede observarse, las consideraciones iniciales anteriores al comienzo de la programación (véase sección 4.2) han resultado ser correctas. De esta manera, nos hemos encontrado con que el programa posee una parte común para todos los ciclos de almacenamiento, variando únicamente la parte que se encarga de llevar los botes a su destino. Se han utilizado también el contador y los diversos comparadores a fin de llevar la cuenta de los botes almacenados. Por otro lado, las señales que se consideraron de importancia, las señales de preparado y ocupado de cada eje, han demostrado tenerla puesta que gracias a ellas es posible controlar los movimientos del autómat. Sin embargo se ha debido cambiar la idea que se poseía sobre su uso, de manera que estas señales deben ser analizadas por flanco ascendente o descendente y no por nivel alto o bajo.

Se ha creado un programa de inicialización puesto que el robot requería de un estado inicial para su correcto funcionamiento que no siempre estaba presente, de modo que en caso de no encontrarse en este estado, la ejecución del programa de inicialización garantice que se llegue a él.

La idea de realizar dos modelos surgió al comentarlo con el tutor del proyecto, puesto que programar ambos modelos brindaría la oportunidad de compararlos, discutiendo así las ventajas o inconvenientes que cada modelo puede mostrar.

De este modo se pensó que el almacenamiento en zigzag era el más convencional de ambos, sin embargo, posee utilidad real, ya que este modelo puede utilizarse para paletizar o almacenar objetos en un espacio de manera que los objetos vayan siendo almacenados desde el fondo hacia adelante. Esto puede ser útil para despaletizar los objetos por cualquiera de los extremos de dicho espacio. De hecho, si se despaletizan los objetos desde el fondo de este espacio, el proceso puede realizarse con seguridad mientras el autómat trabaja, puesto que no se interferirá en su recorrido. También puede ser útil en motivos de seguridad puesto que un operario humano que trabaje en el ambiente del robot, puede predecir fácilmente dónde irá almacenado el siguiente objeto siguiendo el patrón, de modo que pueda alejarse a tiempo con tal de no interferir en la acción del autómat ni resultar herido por un accidente con este.

De igual modo se dedujo que el almacenamiento en espiral hacia adentro podría ser útil, puesto que el autómat primero delimita su zona de carga, creando el borde exterior, para posteriormente seguir almacenando objetos hacia el interior de este perímetro creado. La utilidad pues de este modelo radica en que un operario humano puede observar cómo el robot delimita primeramente su espacio de carga, para posteriormente ir completándolo, de modo que dicho operario no se interponga de manera no intencionada en la trayectoria de almacenaje de la célula, al no poder introducirse en dicho espacio. De este modo, se puede trabajar de forma segura alrededor de la zona de almacenaje del autómat.

En términos de velocidad, ambos modelos son prácticamente igual de rápidos. En el momento se ha medido el tiempo del ciclo completo de ambos modelos, la diferencia entre



ambos ha sido menor a cuatro segundos siempre. Esta diferencia no puede tenerse demasiado en cuenta puesto que el modo de funcionamiento de la estación obliga a introducir los botes en la posición de recogida de forma manual, así como pulsar el botón de *start* cada vez que se desee comenzar un nuevo ciclo, de modo que estos cuatro segundos pueden deberse a la manipulación no perfecta de un humano.

Se considera que la realización de este proyecto puede dotar de ciertas habilidades requeridas a la hora de enfrentarse a la automatización de un proceso industrial en una empresa o en un escenario real. Dicho de otro modo, realizar este proyecto de forma correcta tiene un alto carácter profesional.

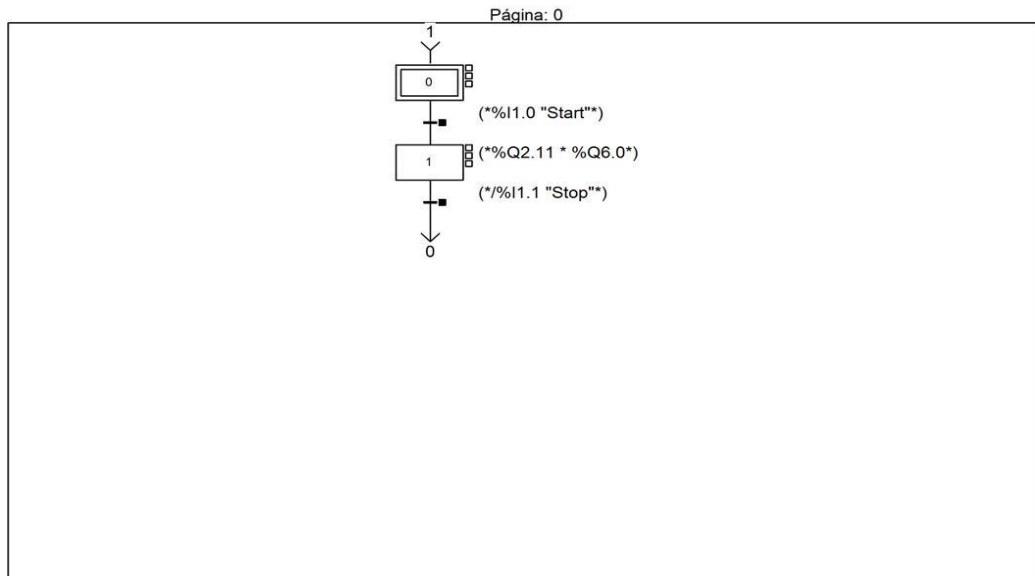
Finalmente, se ha alcanzado satisfactoriamente el objetivo de realizar un programa para automatizar la estación IPC-203 cuyo funcionamiento incluye el paletizado de veinticinco botes en dos modelos distintos, además de las situaciones de alarma cuando el bote a almacenar no tiene tapa o no está presente en la posición de origen, o se ha finalizado el almacenamiento de todos los botes.

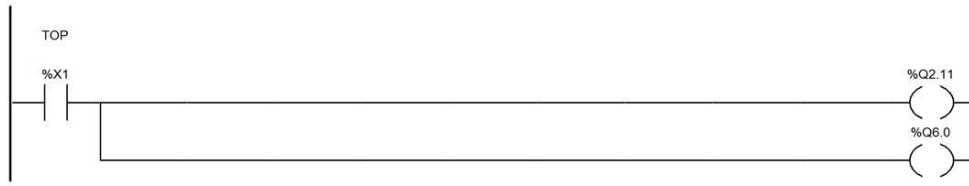
6. Bibliografía

- [1] BERNABEU SOLER, E. J. (2013). *UD2 Tema 6: Arquitectura del autómata programable*. Temario de la asignatura Control por Computador (Código 11583). Valencia: Universidad Politécnica de Valencia.
- [2] Colaboradores de Wikipedia, 'Automatización industrial', *Wikipedia, La enciclopedia libre*, 1 agosto 2014, 15:22 UTC, <http://es.wikipedia.org/w/index.php?title=Automatizaci%C3%B3n_industrial&oldid=75982196> [Consulta: 6 agosto 2014]
- [3] Colaboradores de Wikipedia, 'Controlador lógico programable', *Wikipedia, La enciclopedia libre*, 5 junio 2014, 03:04 UTC, <http://es.wikipedia.org/w/index.php?title=Controlador_l%C3%B3gico_programable&oldid=74849945> [Consulta: 6 agosto 2014]
- [4] Colaboradores de Wikipedia, 'GRAFCET', *Wikipedia, La enciclopedia libre*, 6 julio 2014, 14:51 UTC, <<http://es.wikipedia.org/w/index.php?title=GRAFCET&oldid=75432585>> [Consulta: 7 agosto 2014]
- [5] Colaboradores de Wikipedia, 'Lenguaje Ladder', *Wikipedia, La enciclopedia libre*, 26 mayo 2014, 14:40 UTC, <http://es.wikipedia.org/w/index.php?title=Lenguaje_Ladder&oldid=74643415> [Consulta: 7 agosto 2014]
- [6] MANDADO PÉREZ, E. et al. (2009). *Autómatas programables y sistemas de automatización*. Barcelona: Marcombo S.A.
- [7] OGATA, K. (2010). *Ingeniería de control moderna*. Madrid: Prentice Hall.
- [8] SMC (2007). *IPC-200 Manual de usuario*.

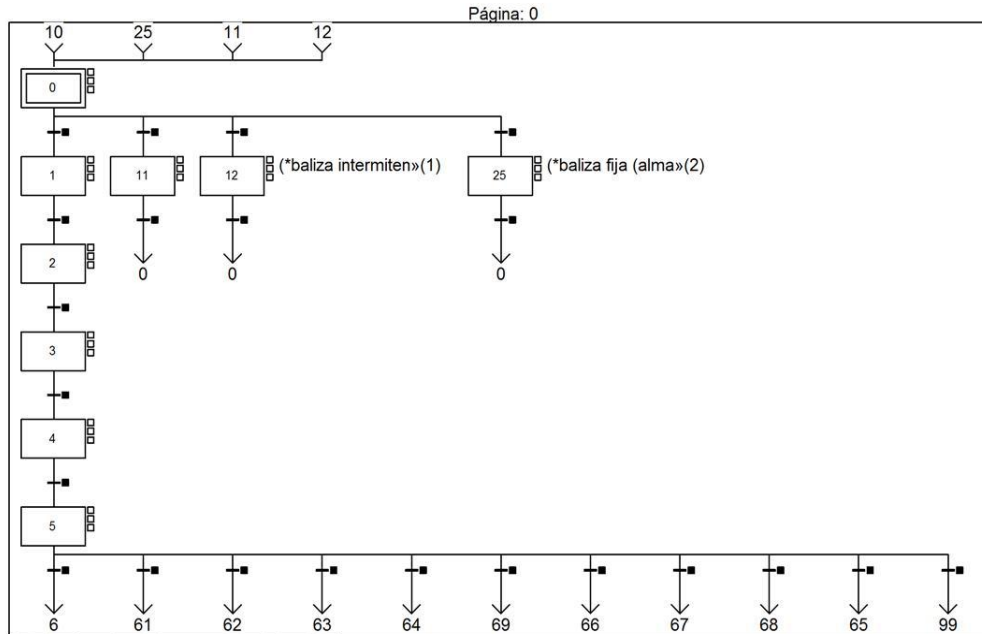
Anexo

Inicialización_ejes.stx





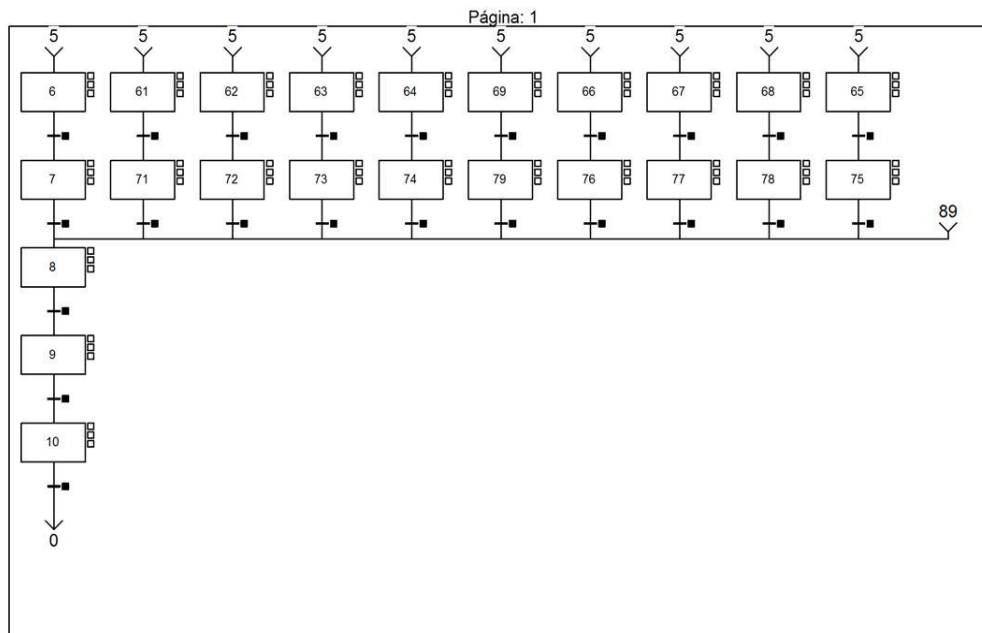
Modelo_zigzag.stx

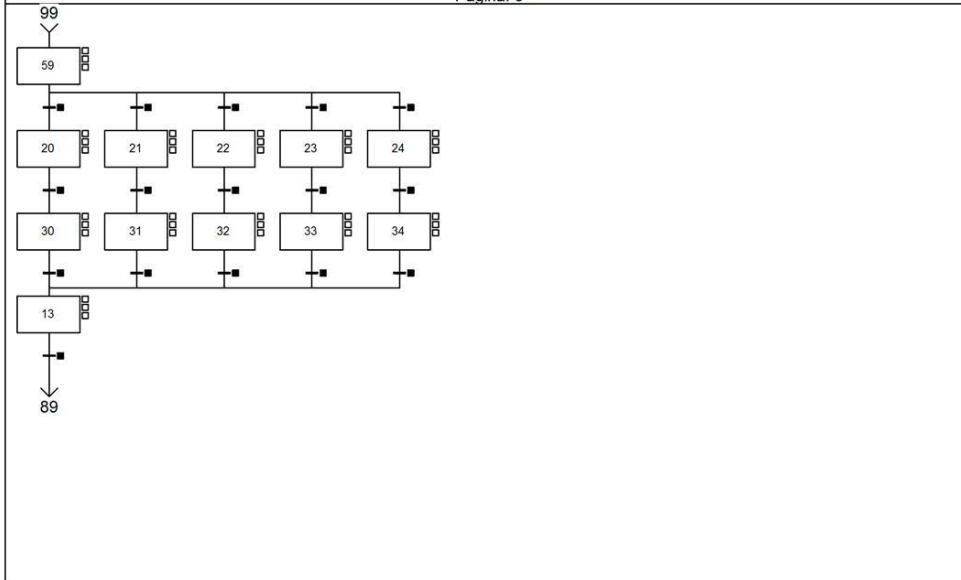
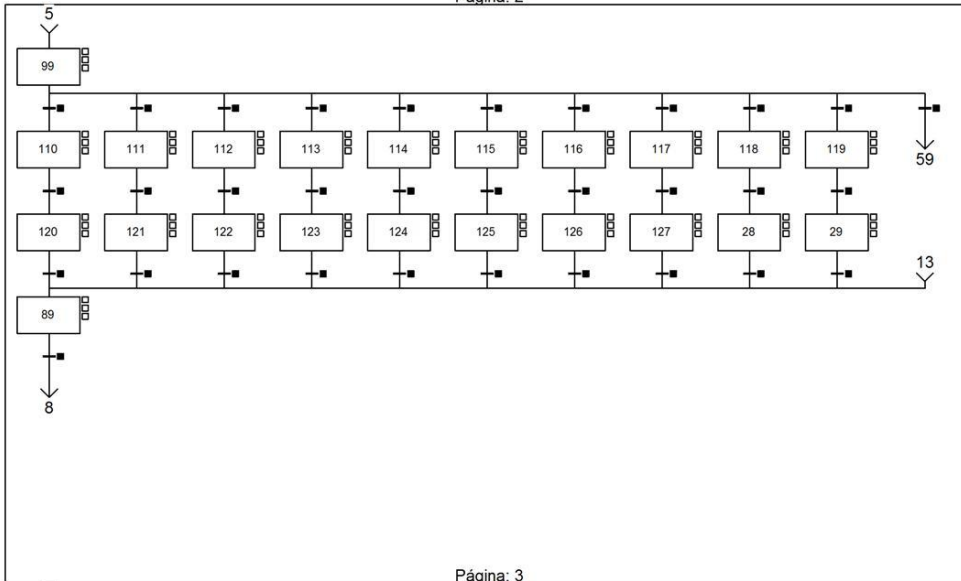


Lista de referencias de los comentarios de la página:

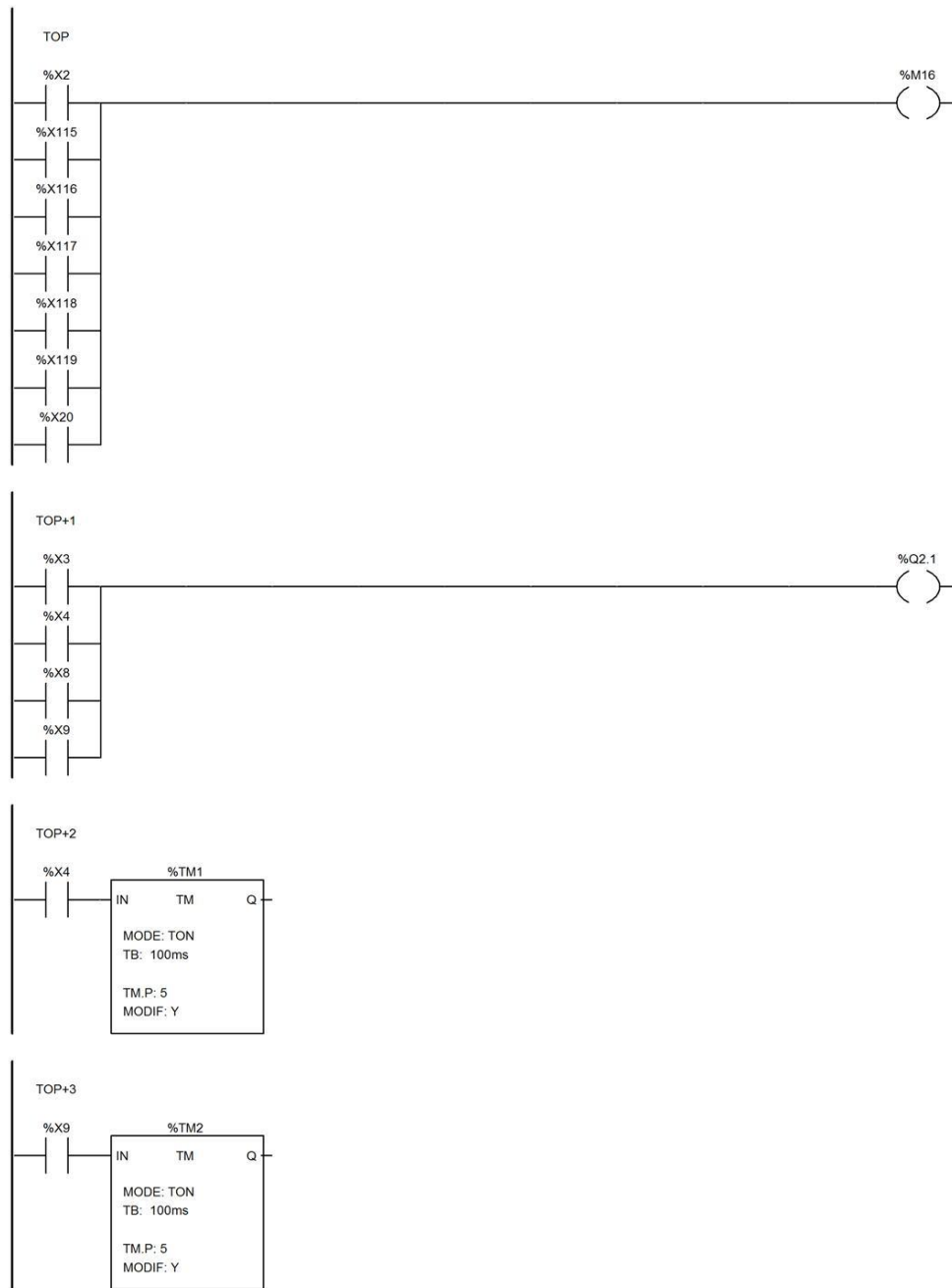
(1):(*baliza intermitente (no bote o no tapa)*)

(2):(*baliza fija (almacén lleno/25 botes)*)





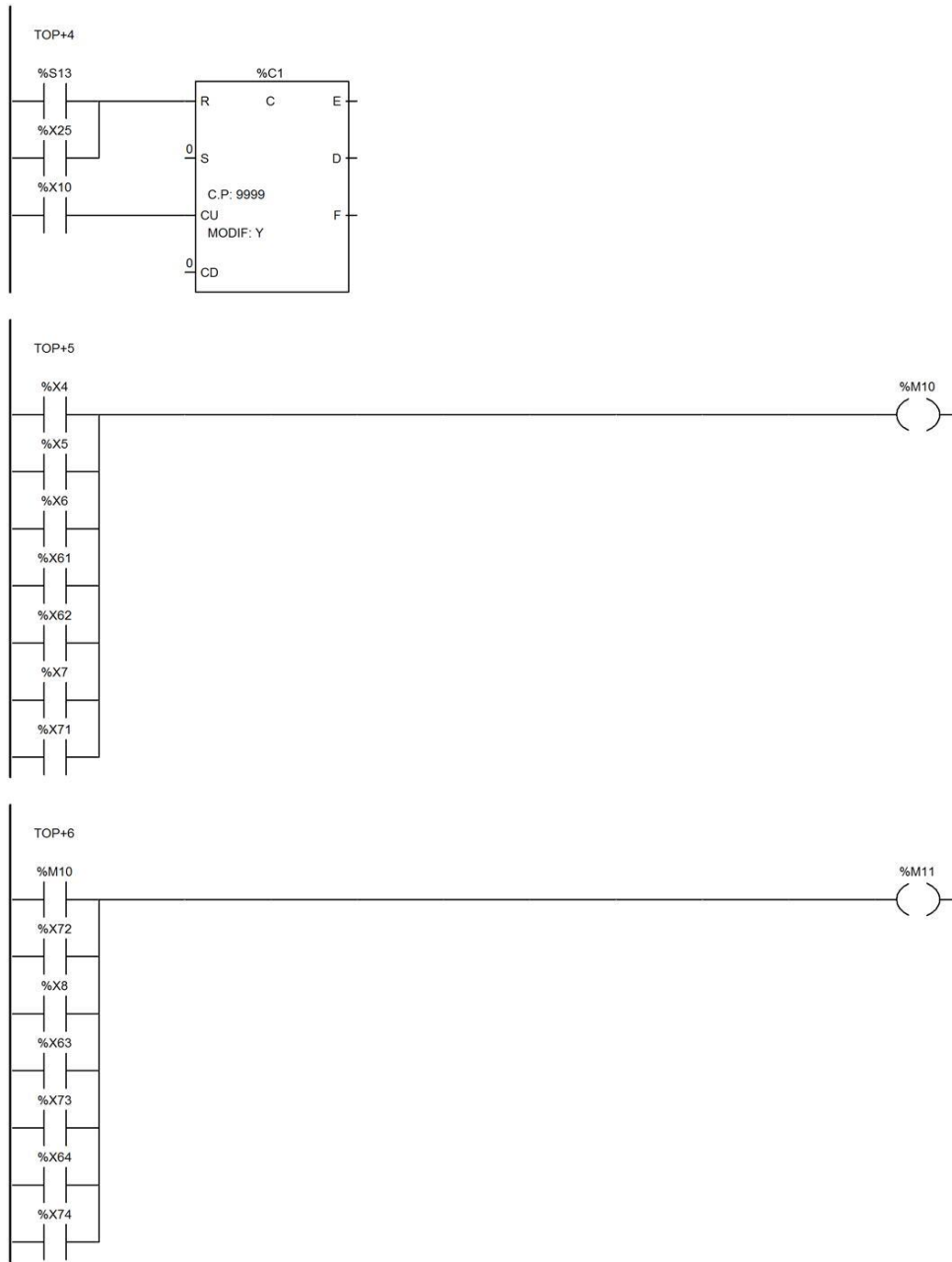
Automatización de un sistema de clasificación y almacenaje



Carpeta: 1

MAST - SectionGR7 - Post

Fecha: 3/9/14



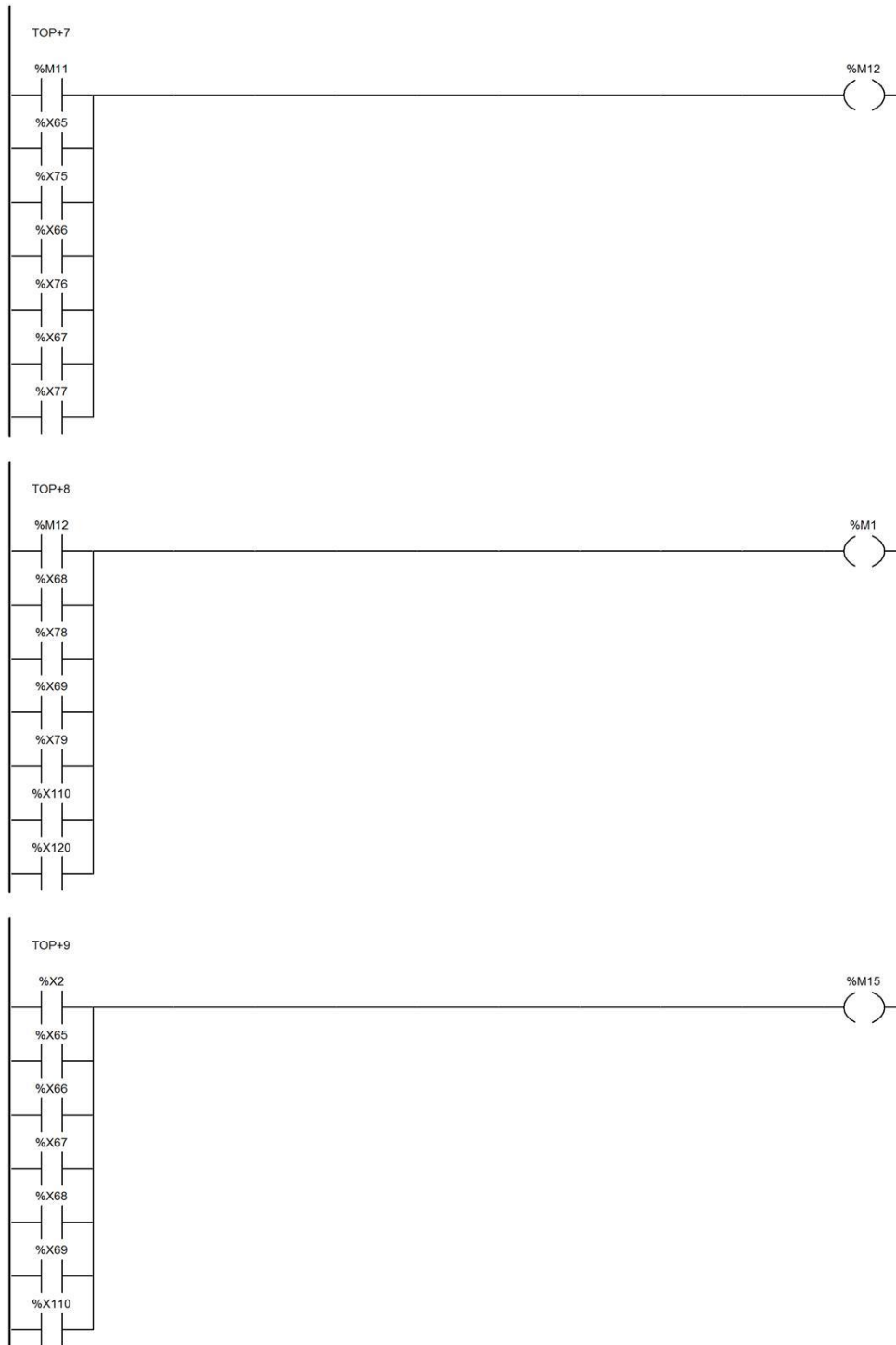
Carpeta: 2

MAST - SectionGR7 - Post

Fecha: 3/9/14



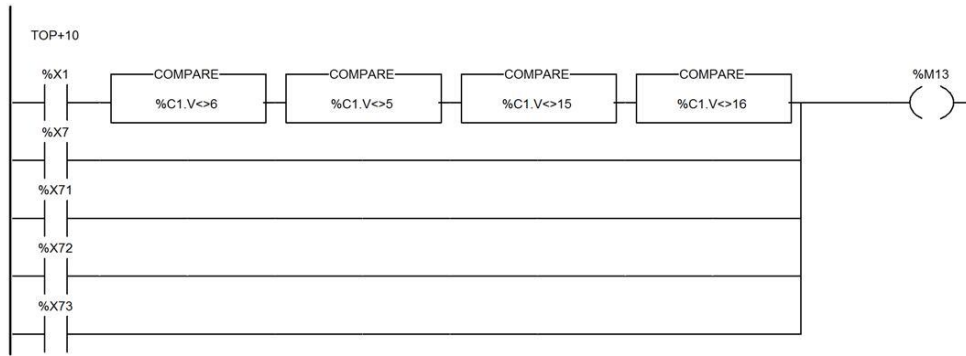
Automatización de un sistema de clasificación y almacenaje



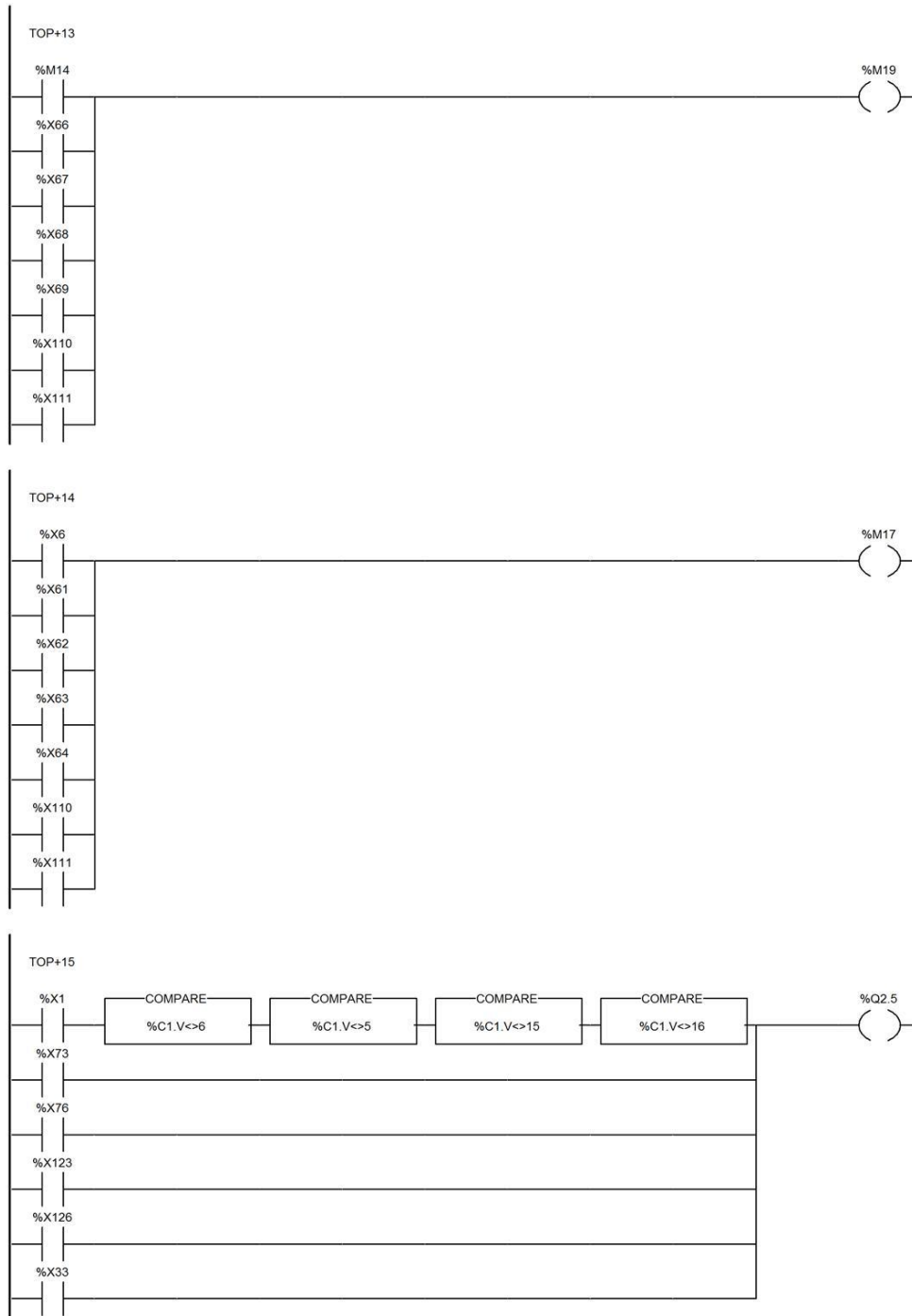
Carpeta: 3

MAST - SectionGR7 - Post

Fecha: 3/9/14



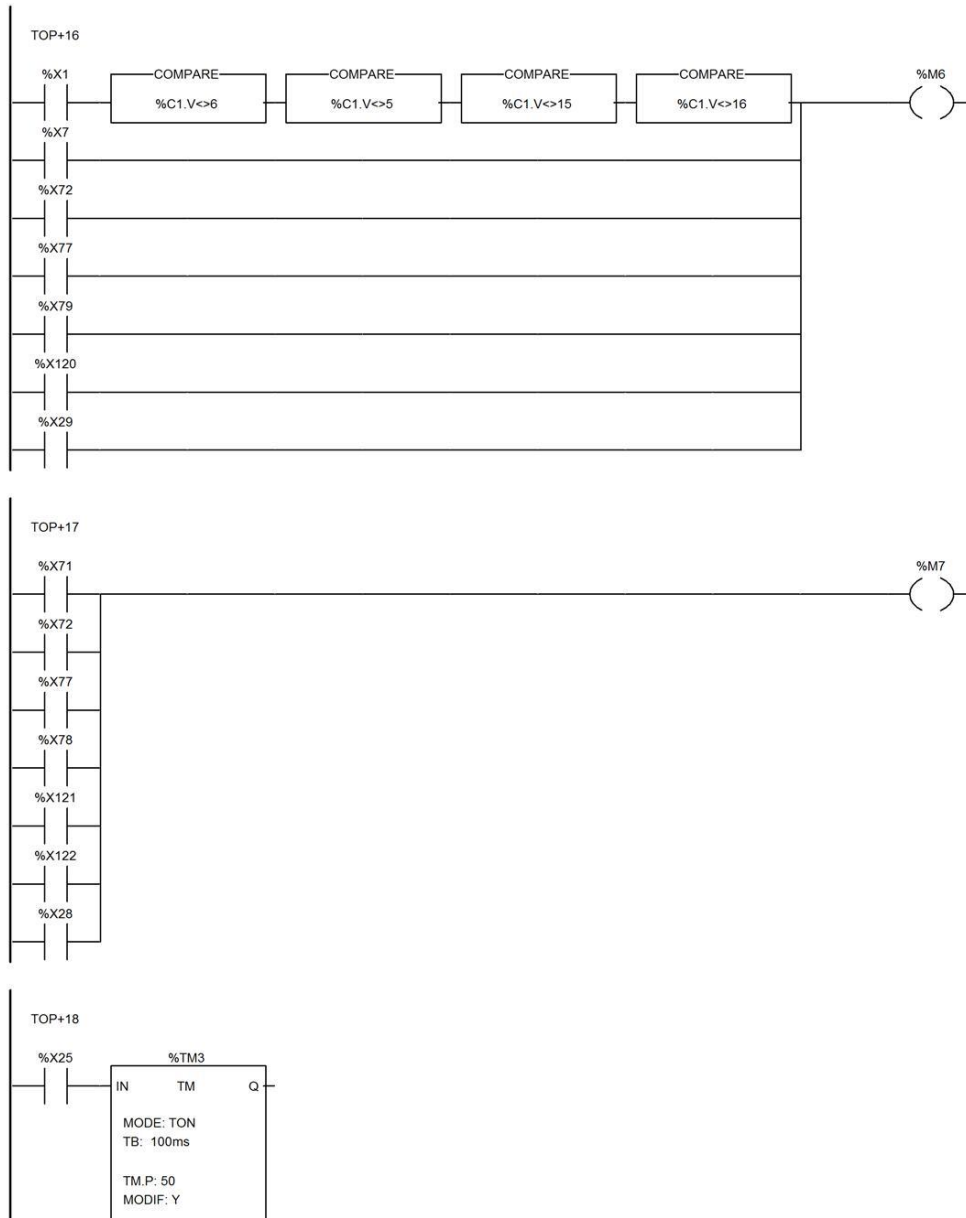
Automatización de un sistema de clasificación y almacenaje



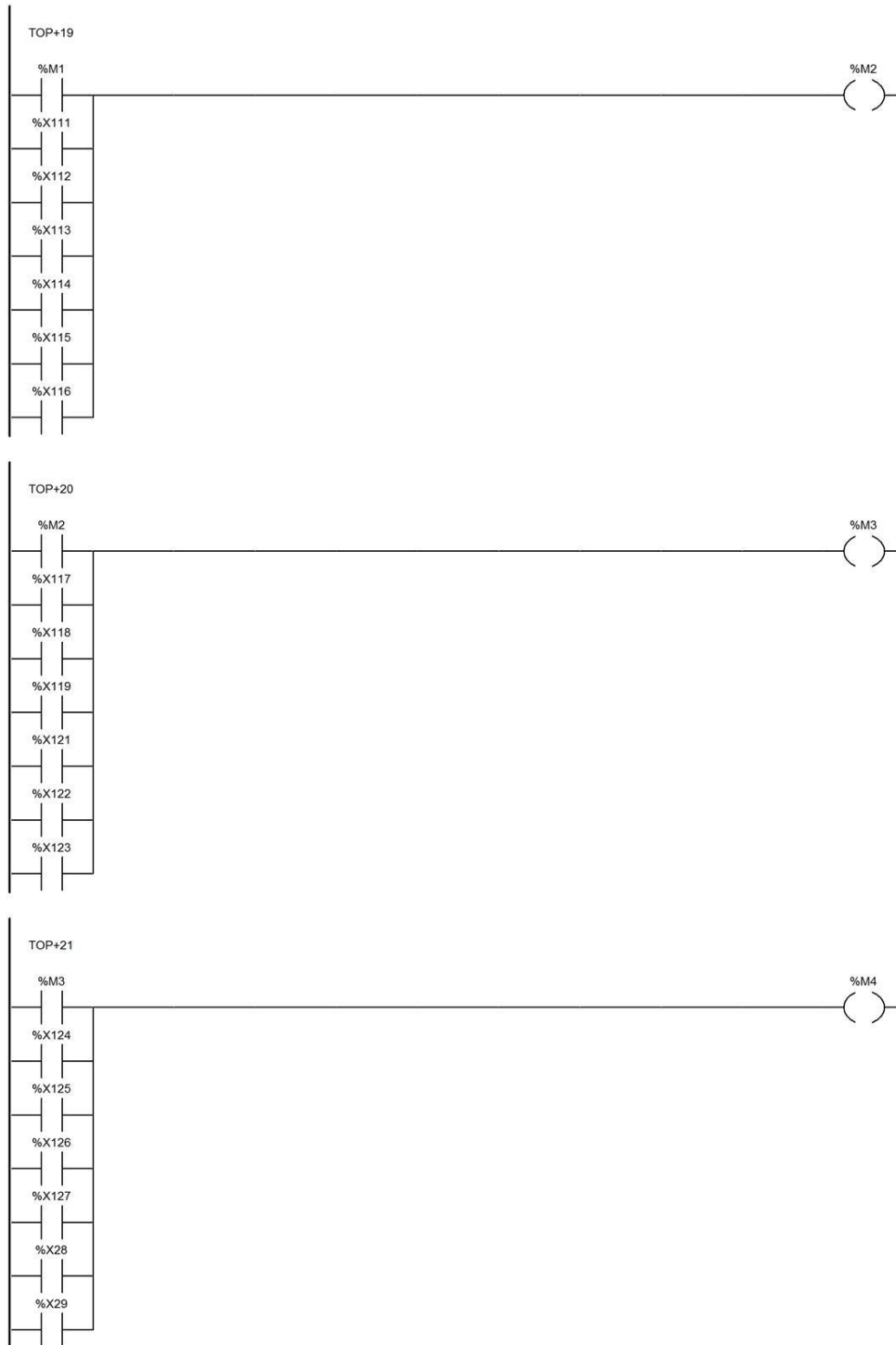
Carpeta: 5

MAST - SectionGR7 - Post

Fecha: 3/9/14



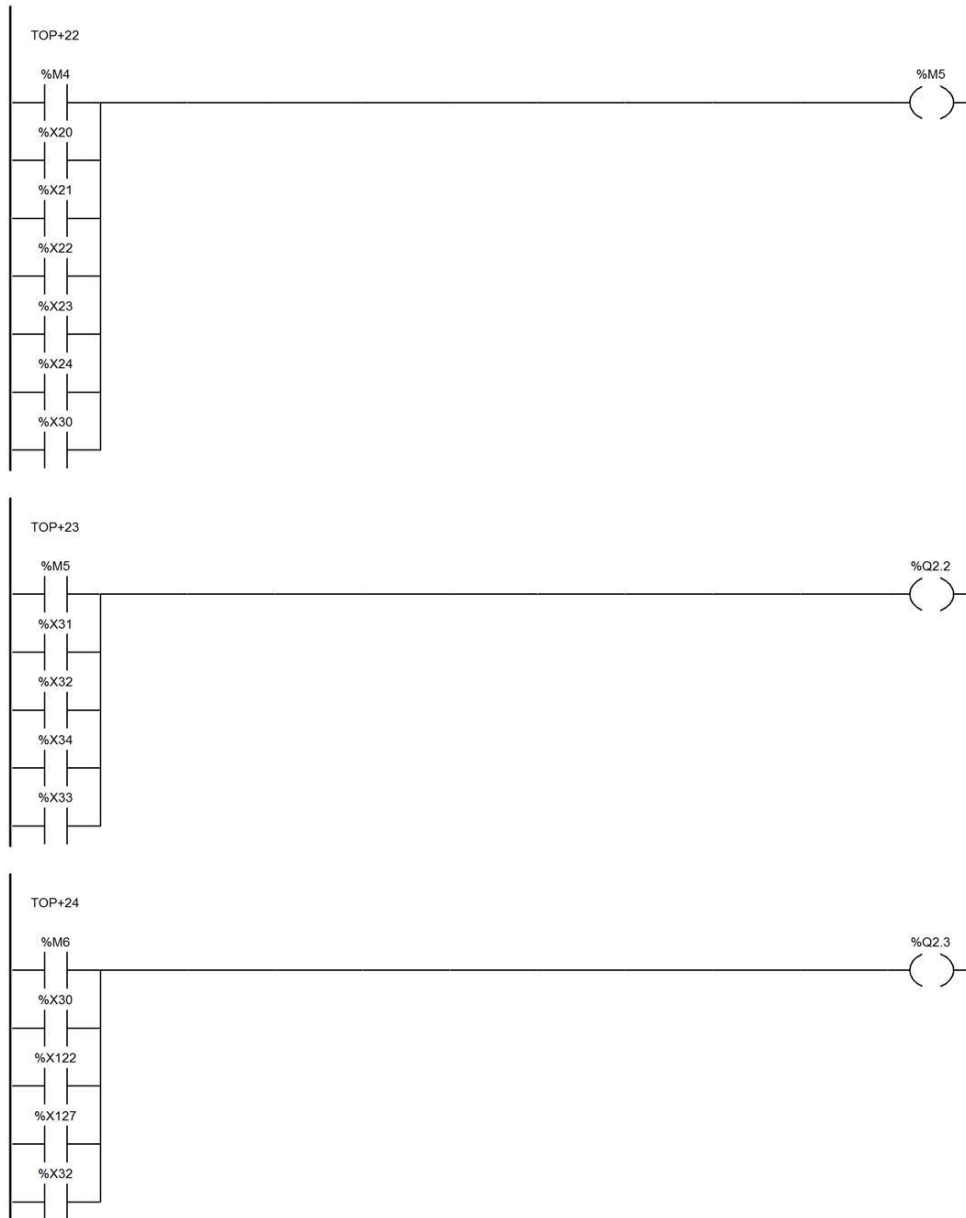
Automatización de un sistema de clasificación y almacenaje



Carpeta: 7

MAST - SectionGR7 - Post

Fecha: 3/9/14

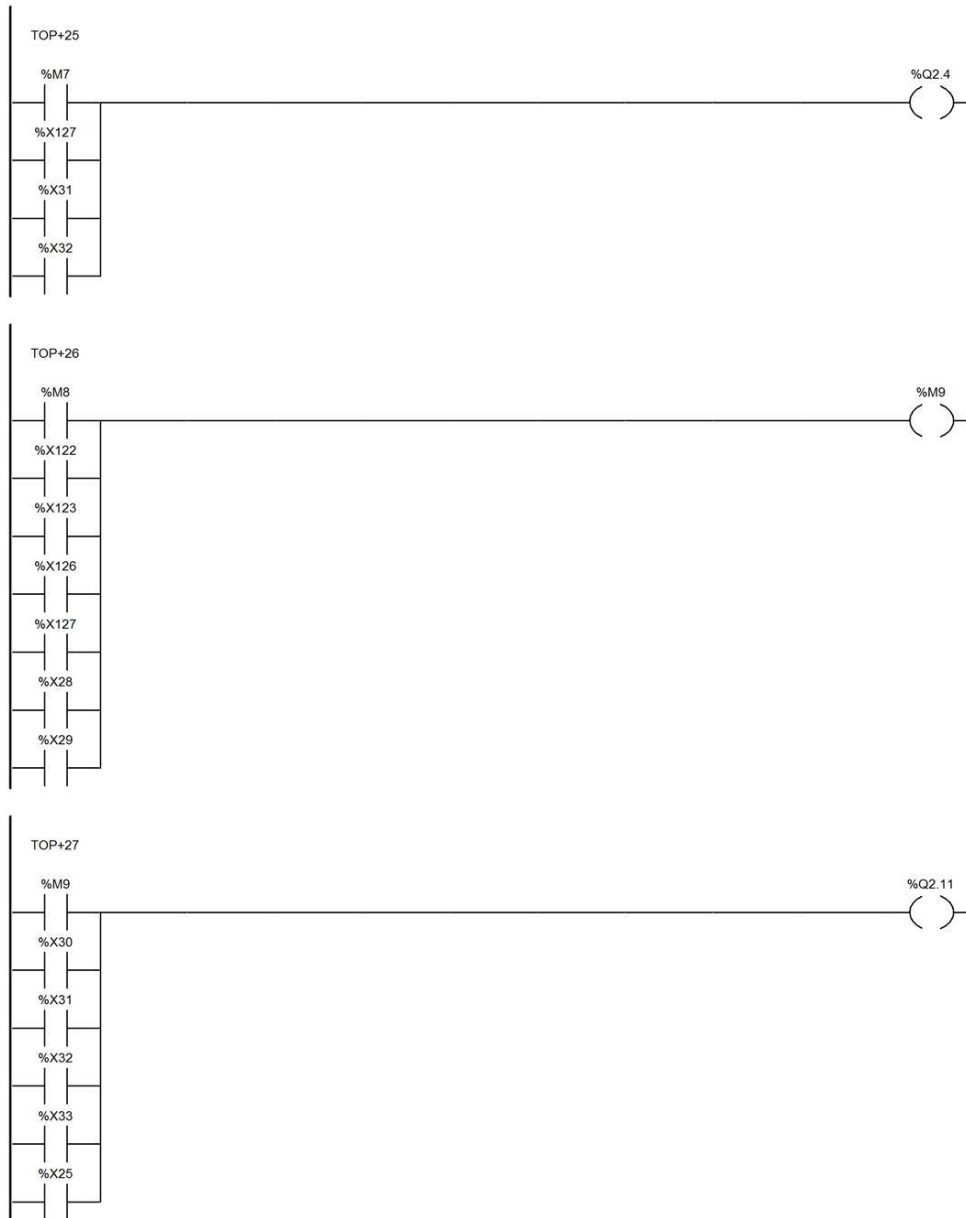


Carpeta: 8

MAST - SectionGR7 - Post

Fecha: 3/9/14

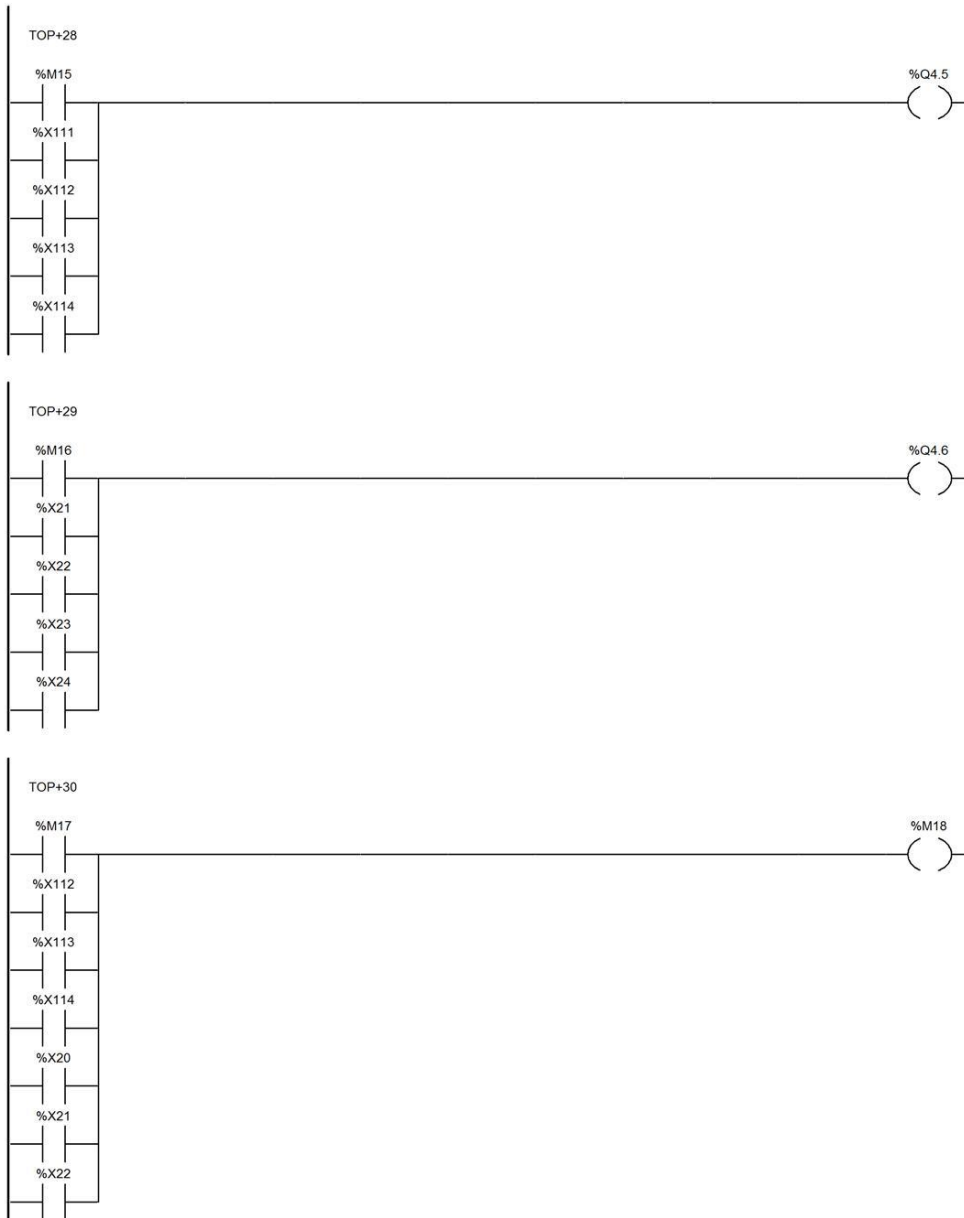
Automatización de un sistema de clasificación y almacenaje



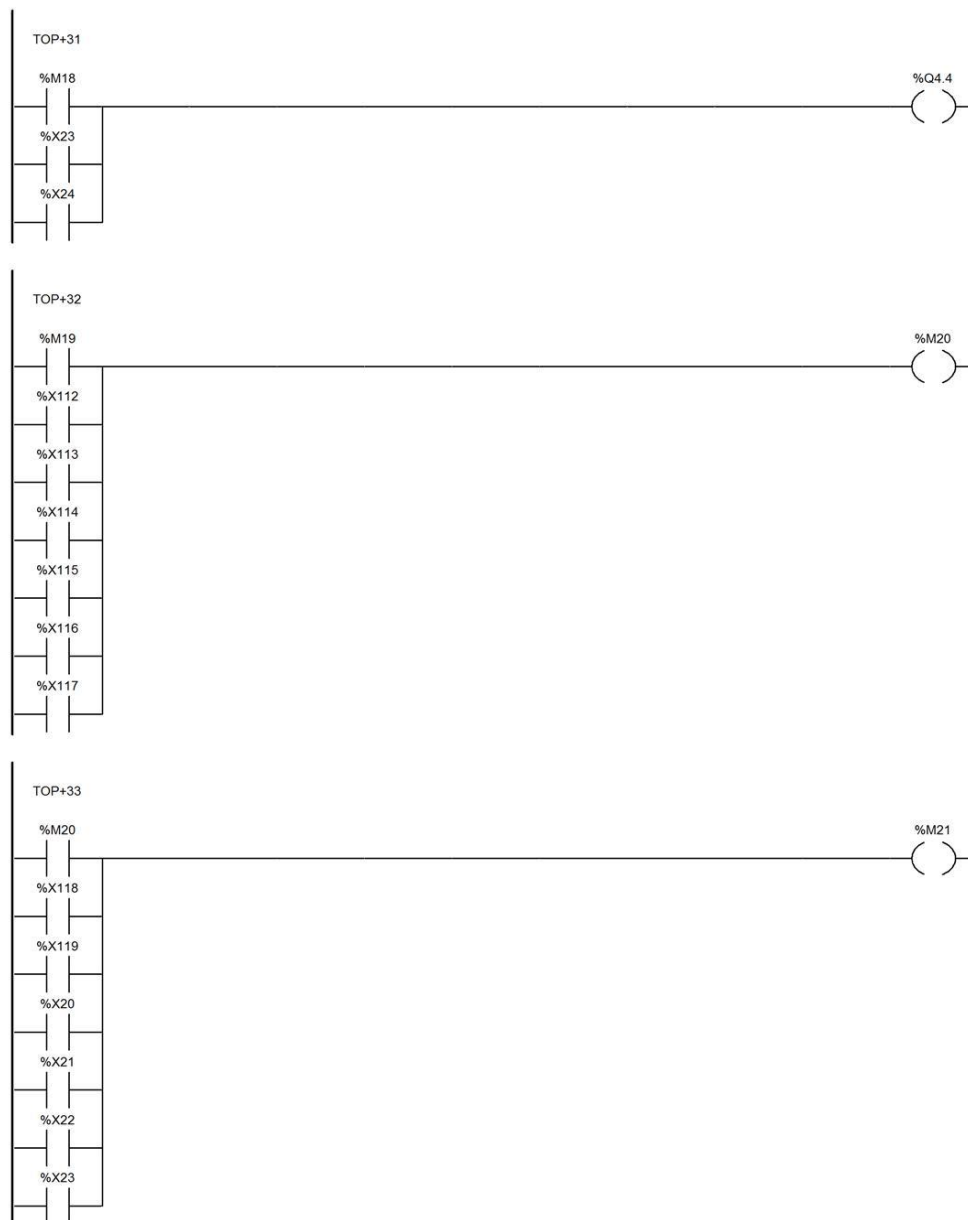
Carpeta: 9

MAST - SectionGR7 - Post

Fecha: 3/9/14



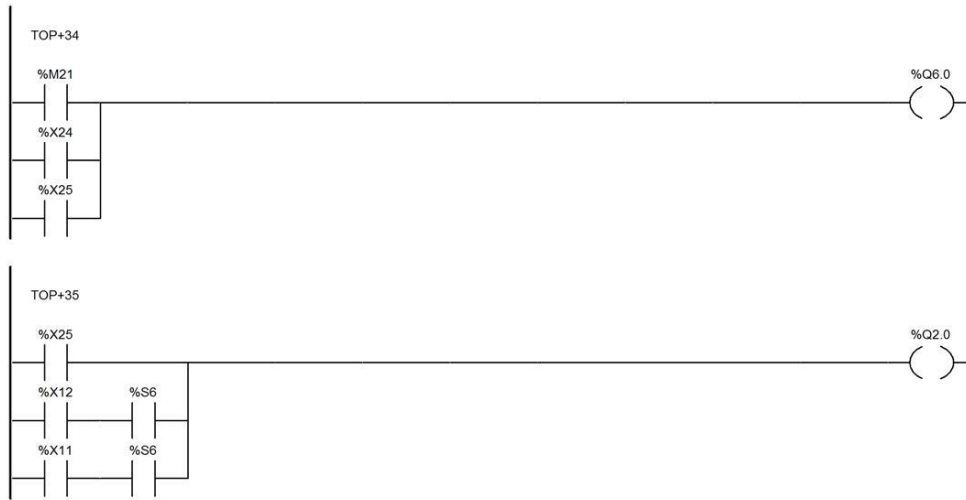
Automatización de un sistema de clasificación y almacenaje



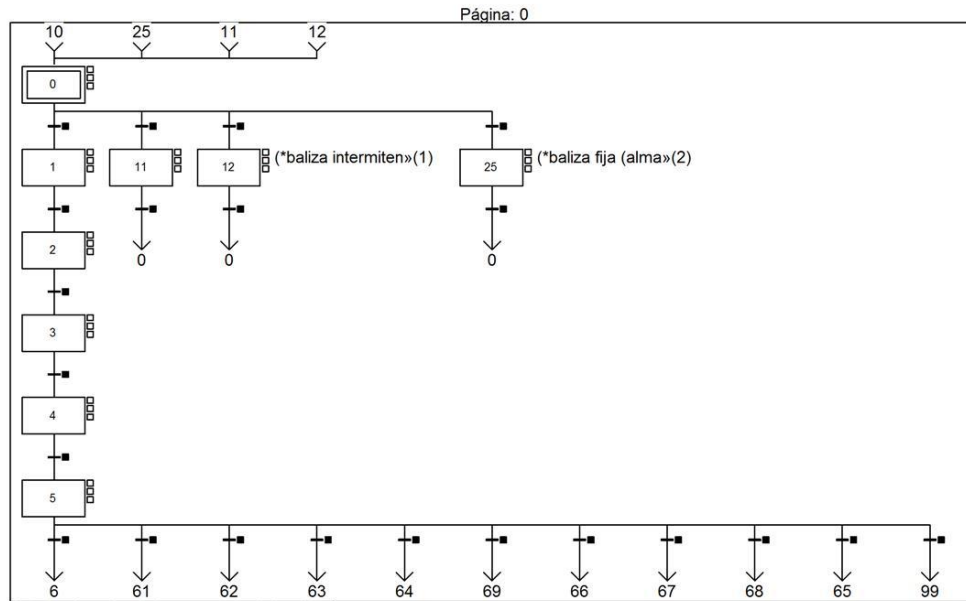
Carpeta: 11

MAST - SectionGR7 - Post

Fecha: 3/9/14



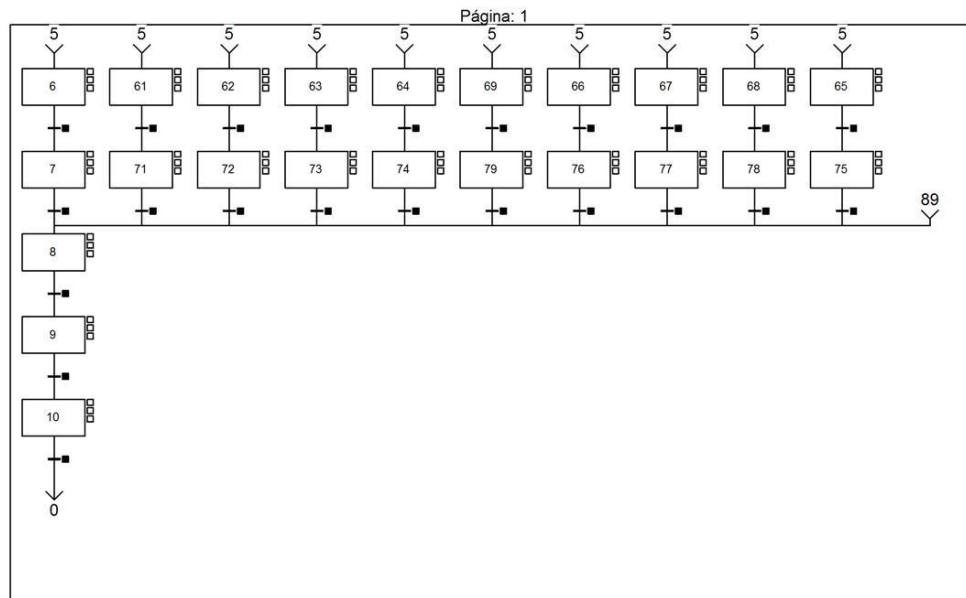
Modelo_espiral.stx

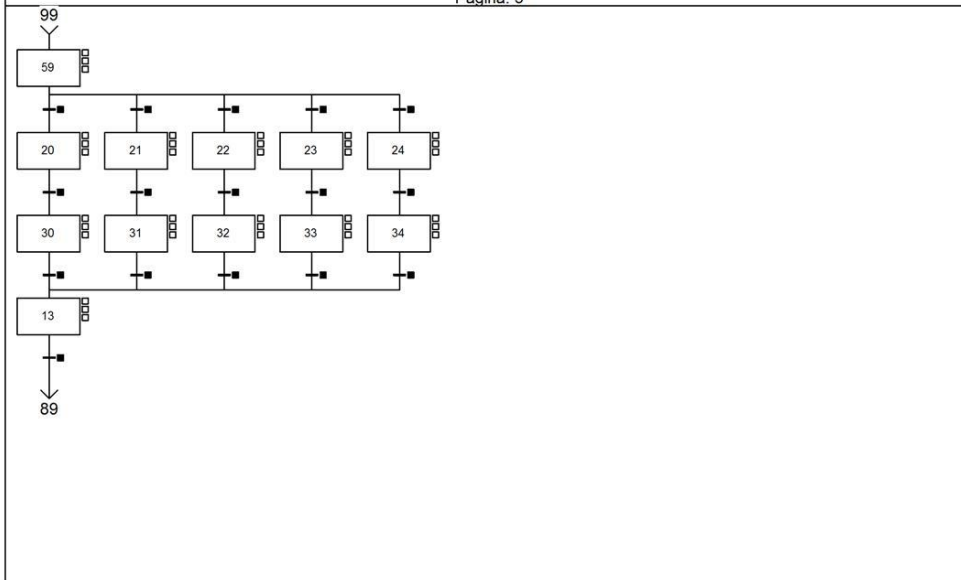
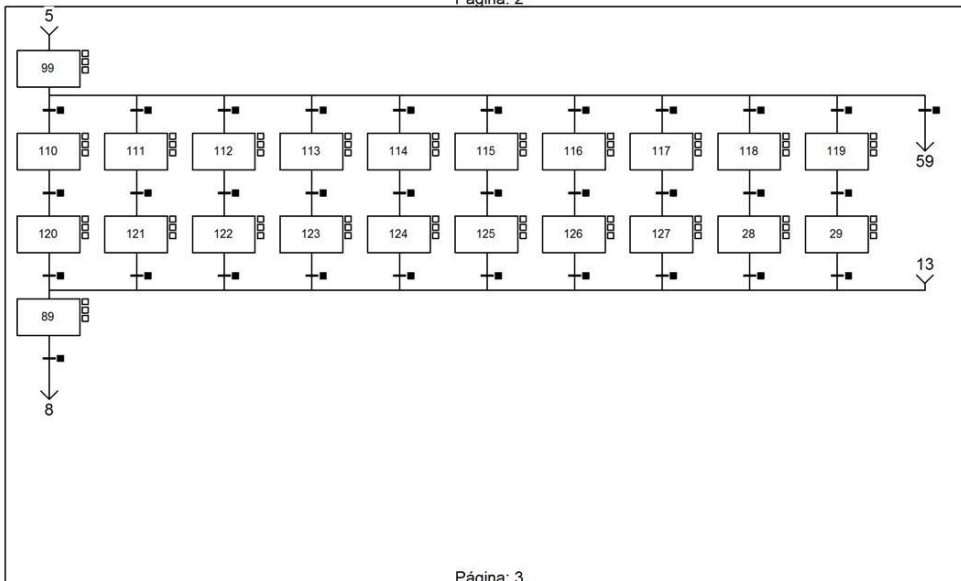


Lista de referencias de los comentarios de la página:

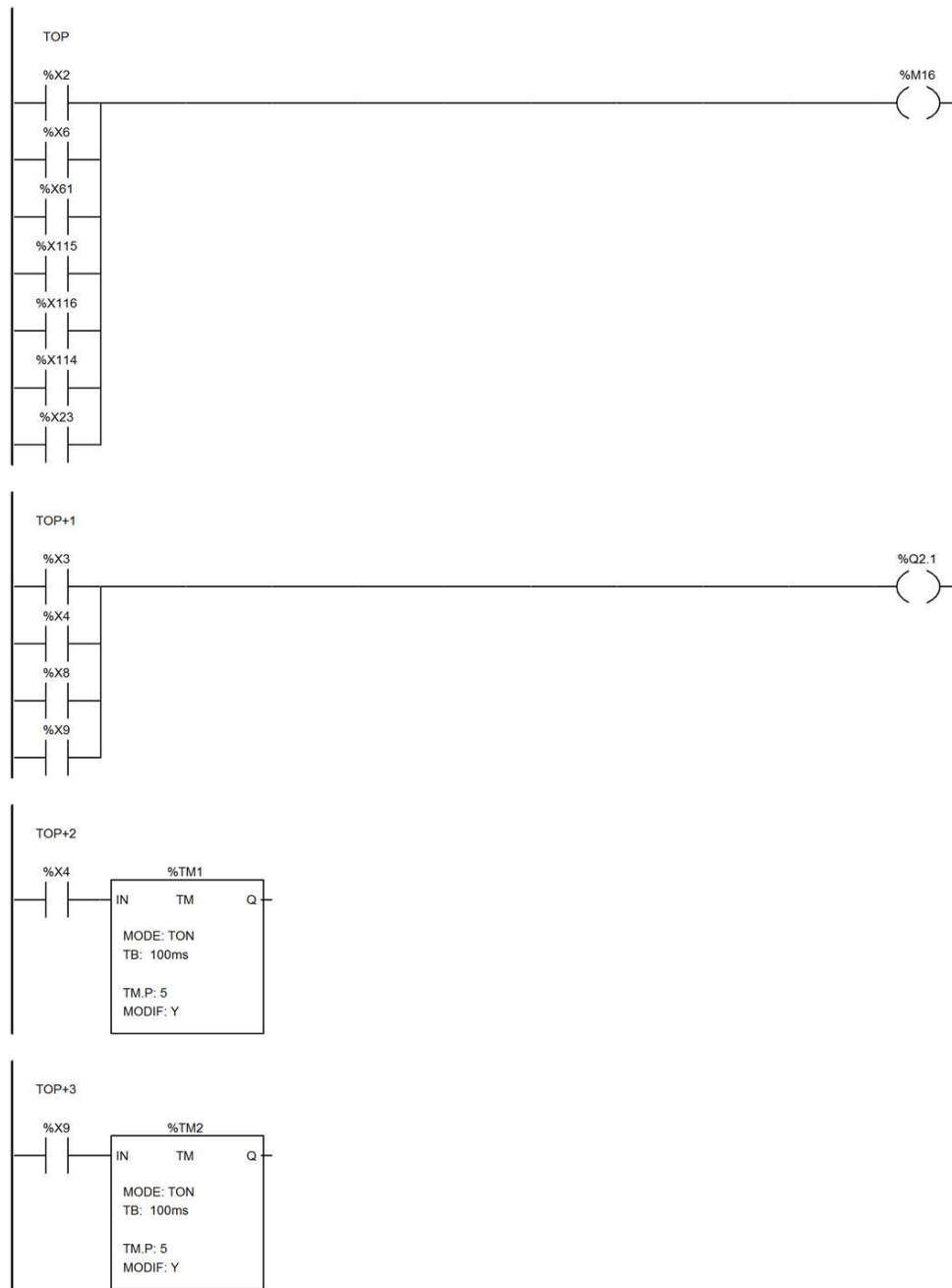
(1):(*baliza intermitente (no bote o no tapa)*)

(2):(*baliza fija (almacén lleno/25 botes)*)





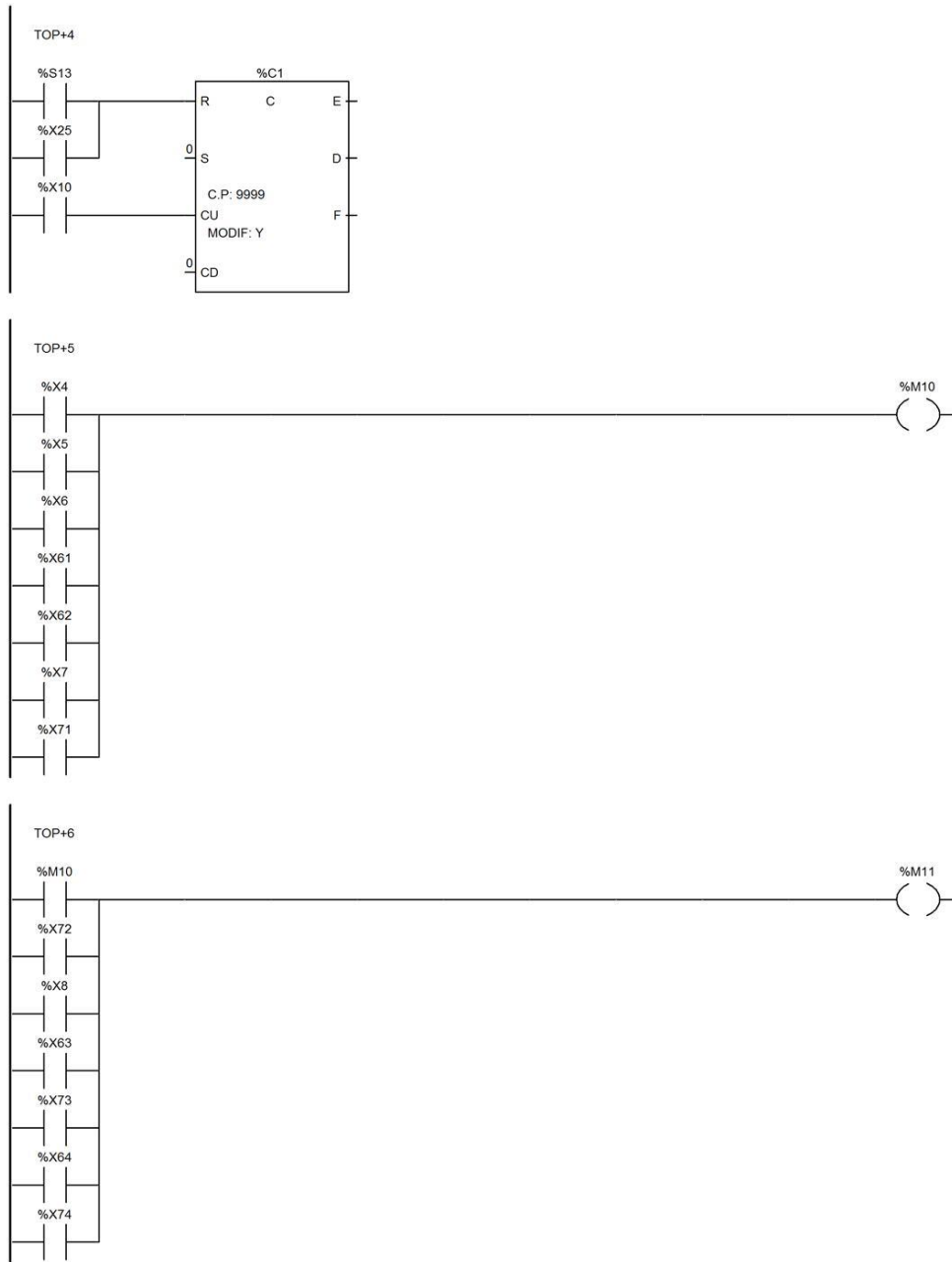
Automatización de un sistema de clasificación y almacenaje



Carpeta: 1

MAST - SectionGR7 - Post

Fecha: 3/9/14



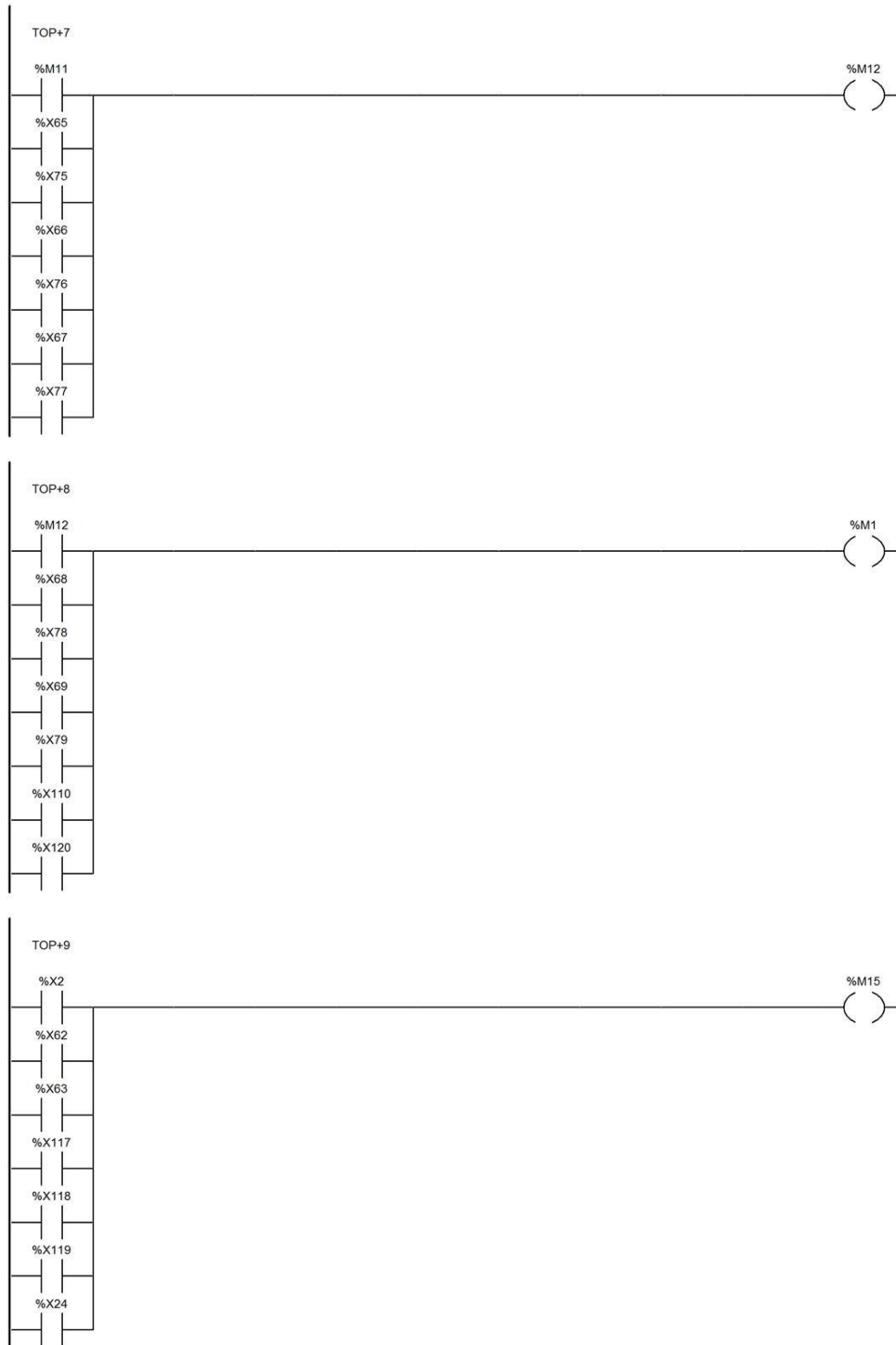
Carpeta: 2

MAST - SectionGR7 - Post

Fecha: 3/9/14



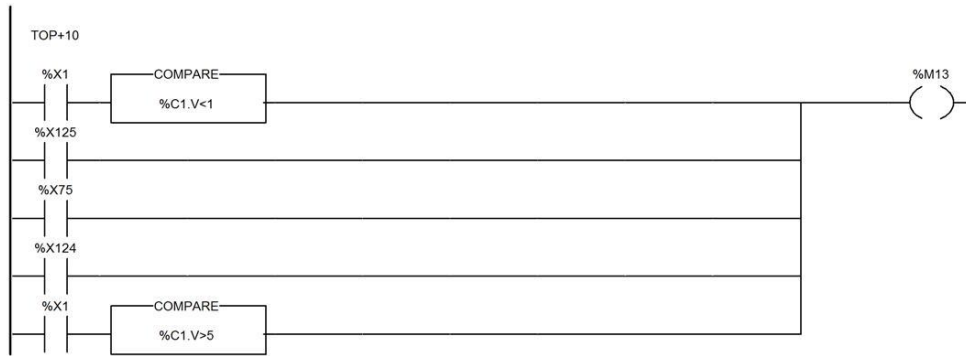
Automatización de un sistema de clasificación y almacenaje



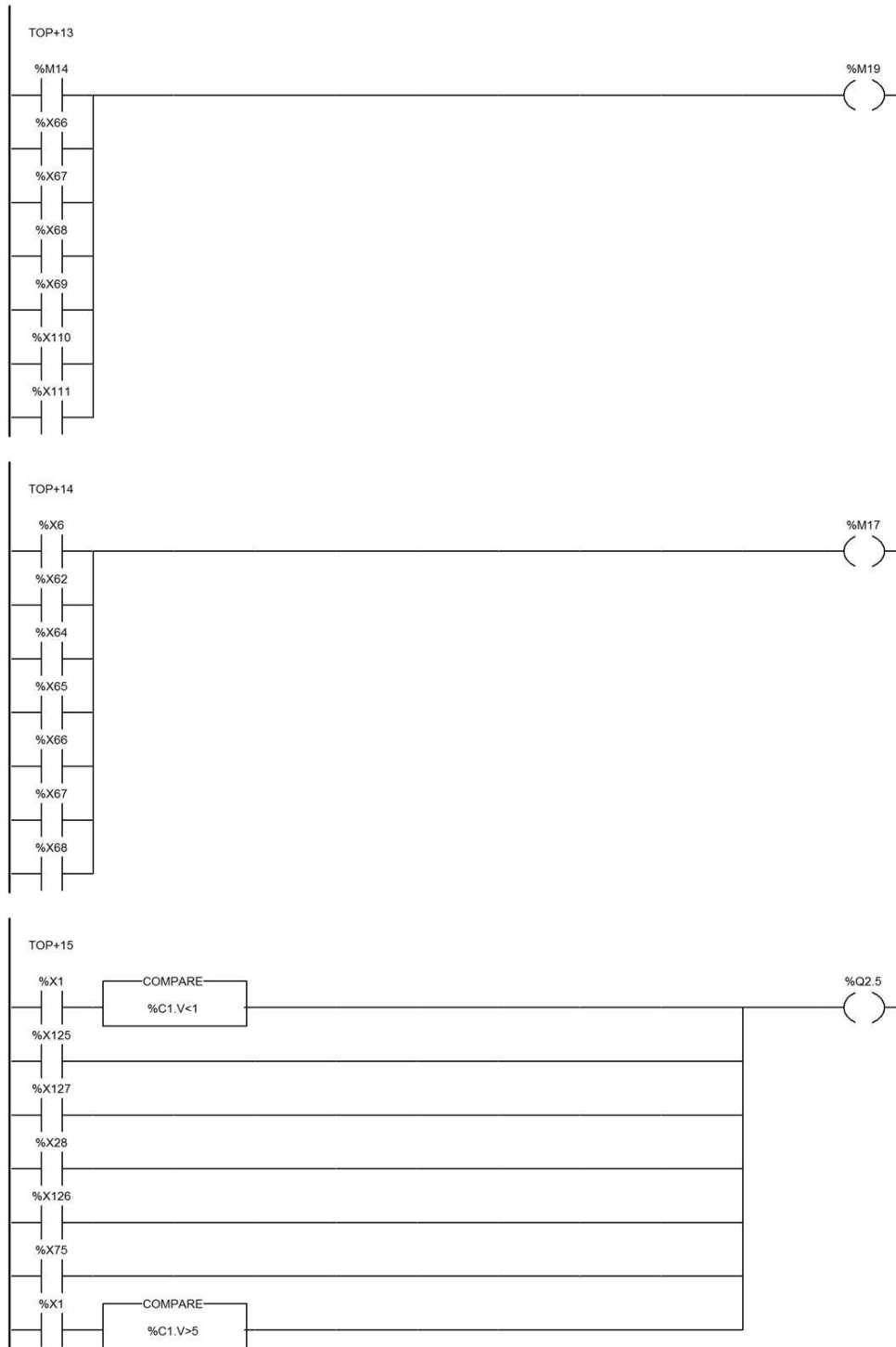
Carpeta: 3

MAST - SectionGR7 - Post

Fecha: 3/9/14



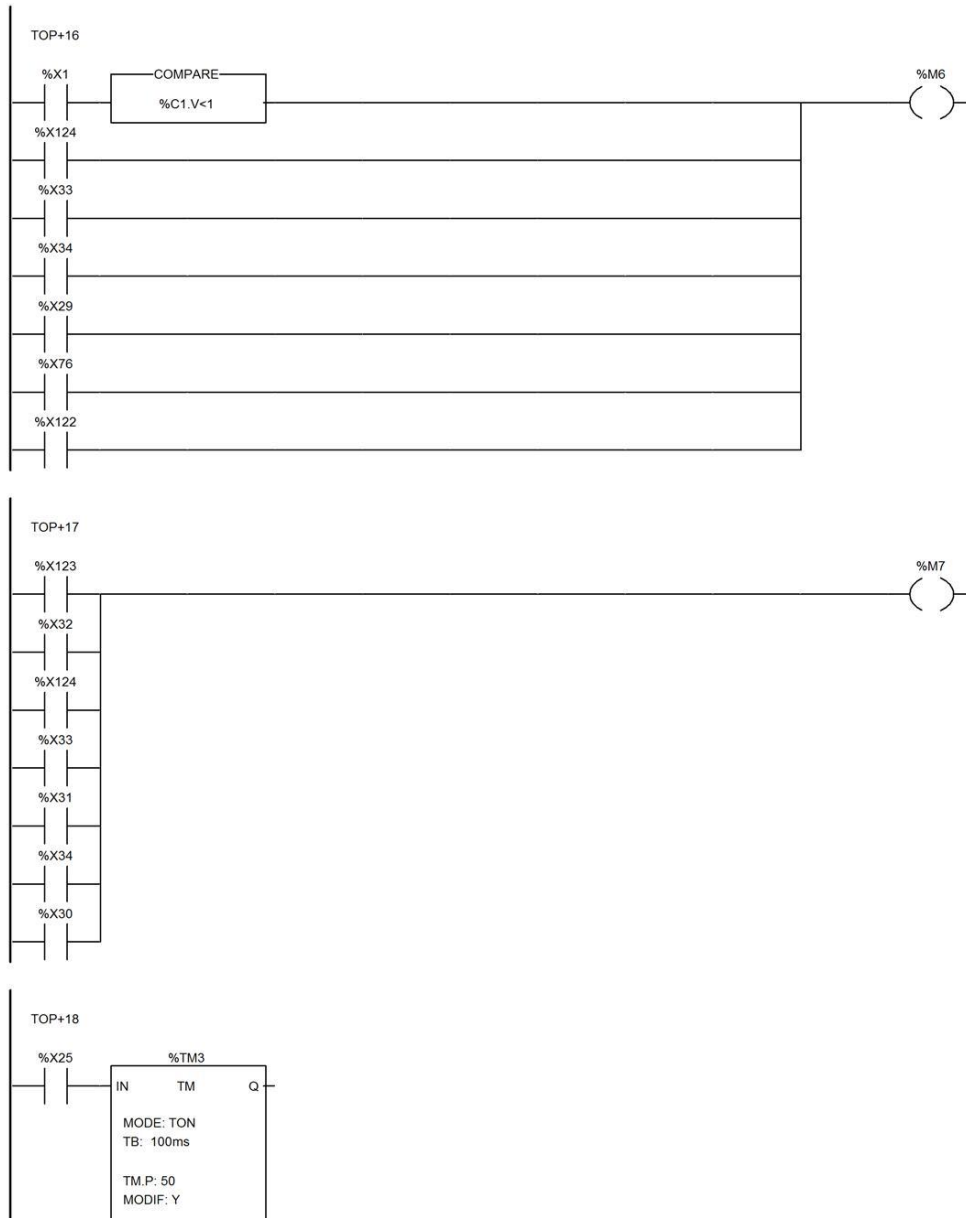
Automatización de un sistema de clasificación y almacenaje



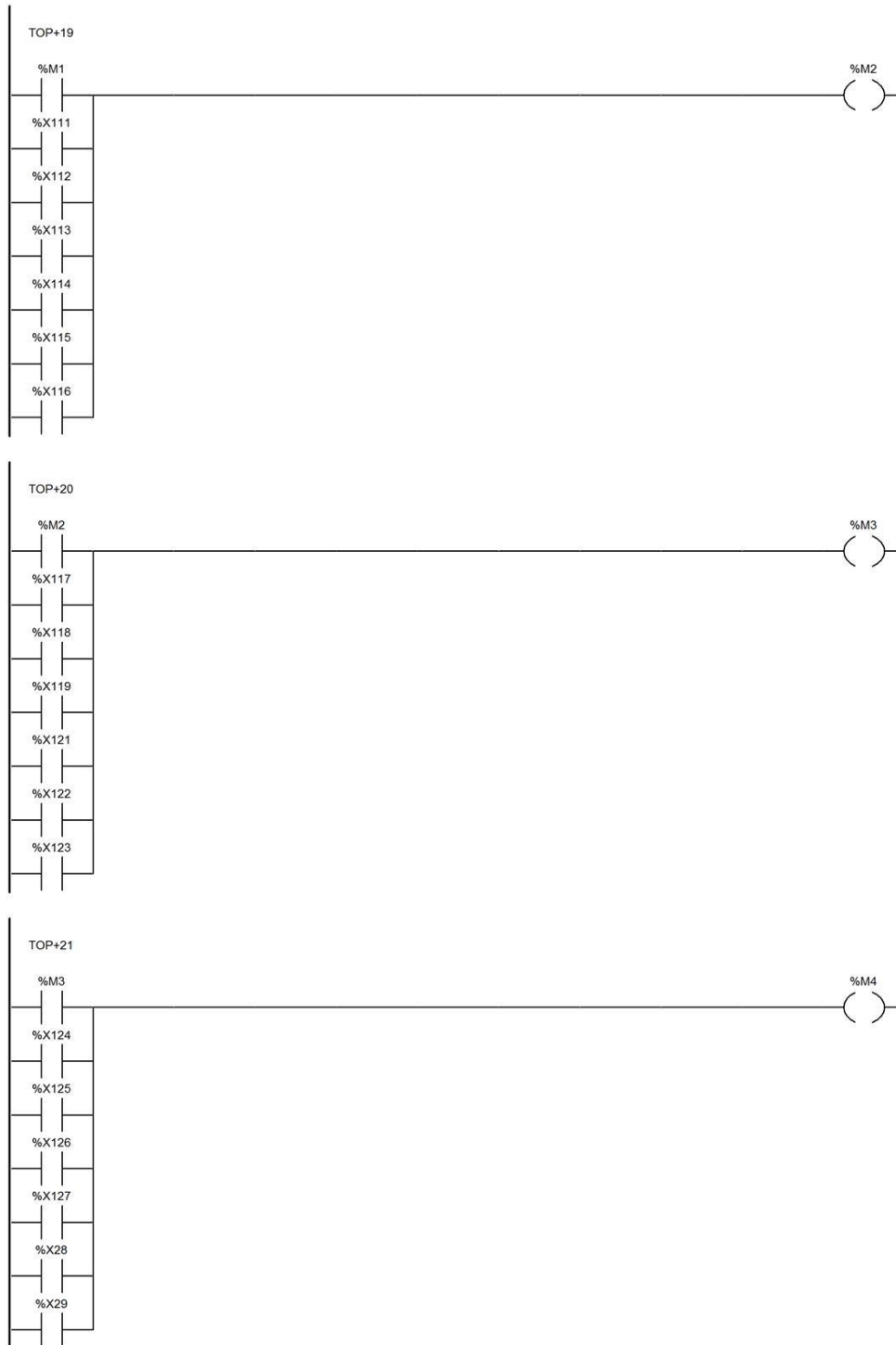
Carpeta: 5

MAST - SectionGR7 - Post

Fecha: 3/9/14



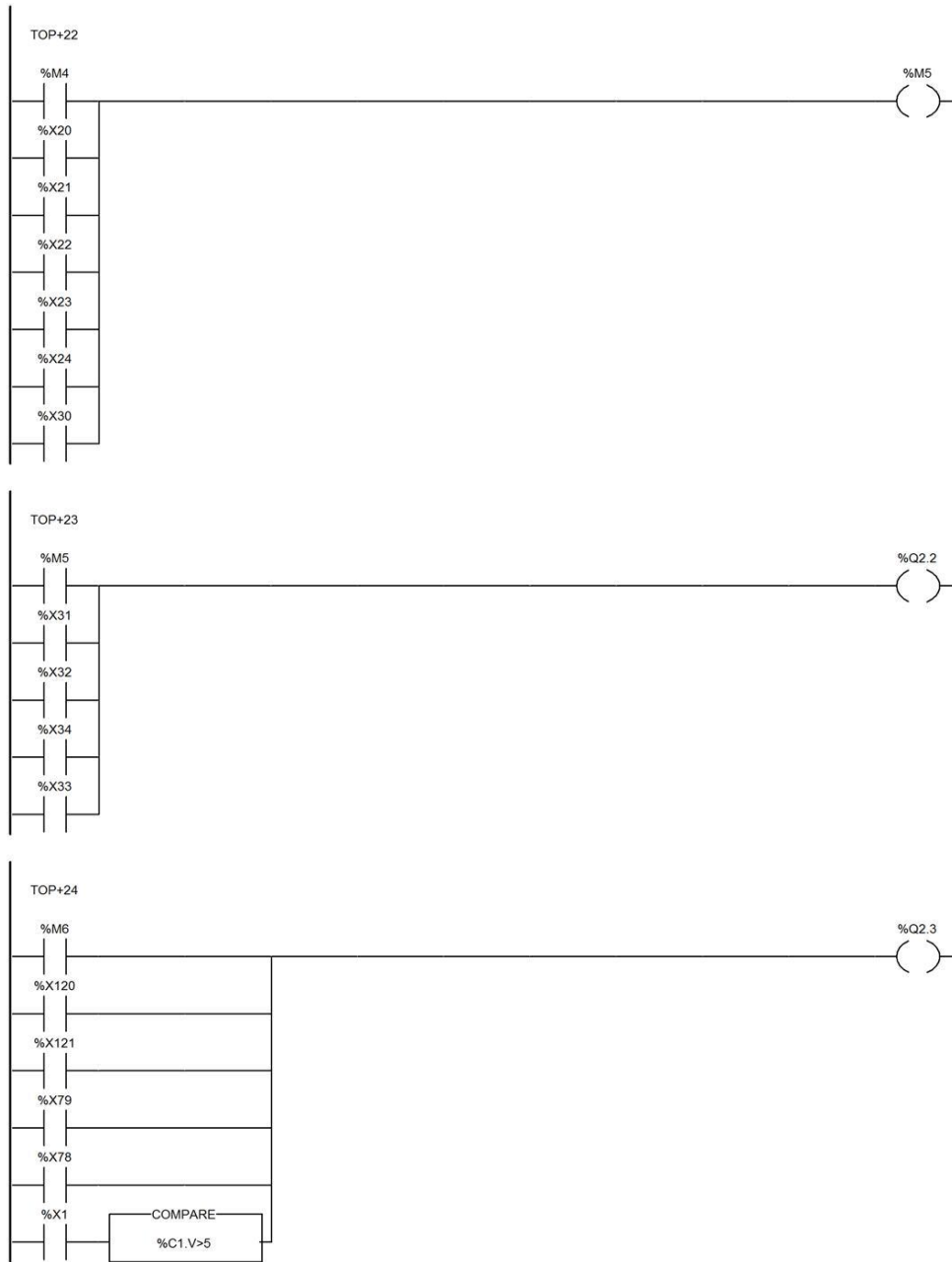
Automatización de un sistema de clasificación y almacenaje



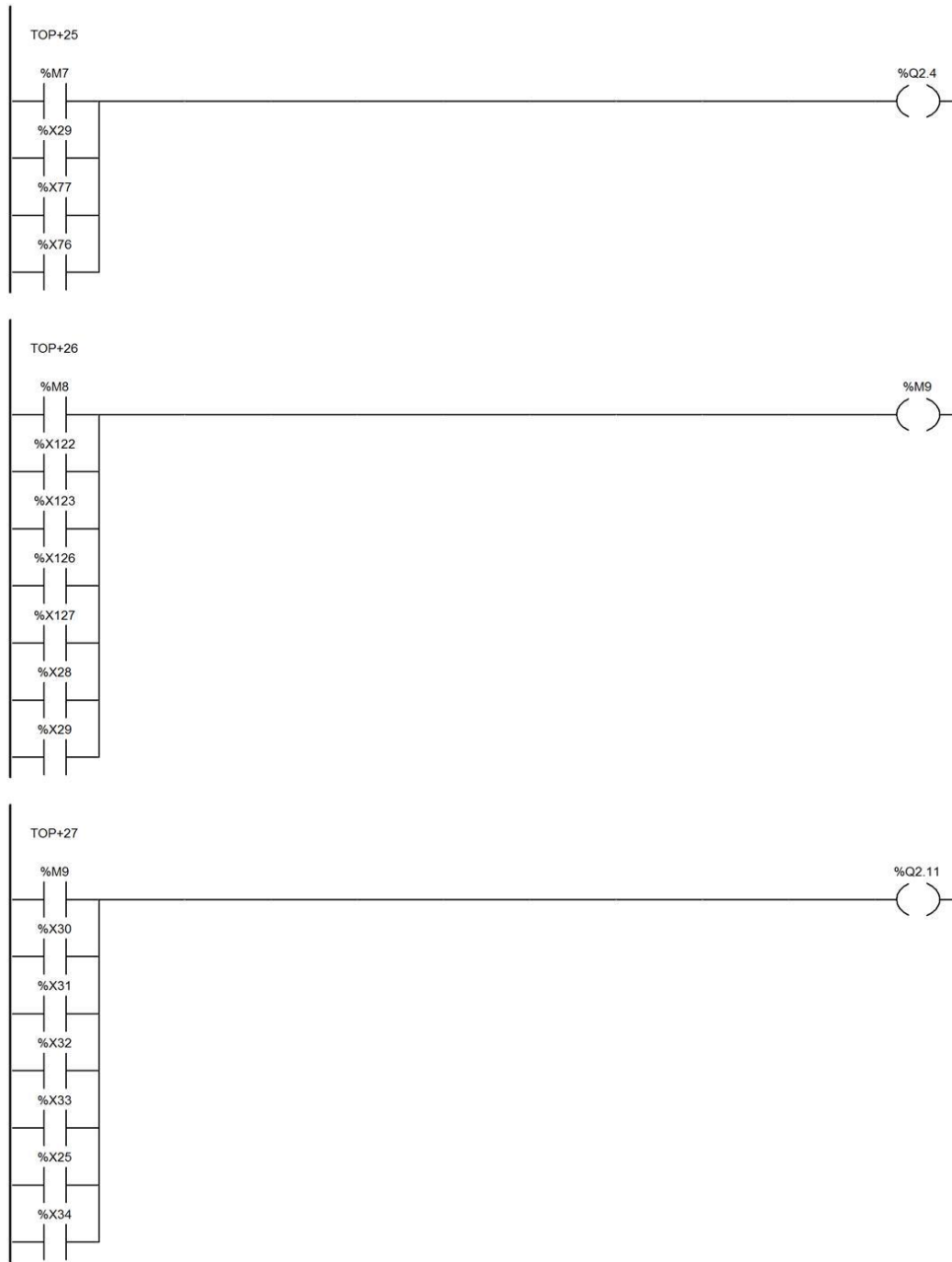
Carpeta: 7

MAST - SectionGR7 - Post

Fecha: 3/9/14



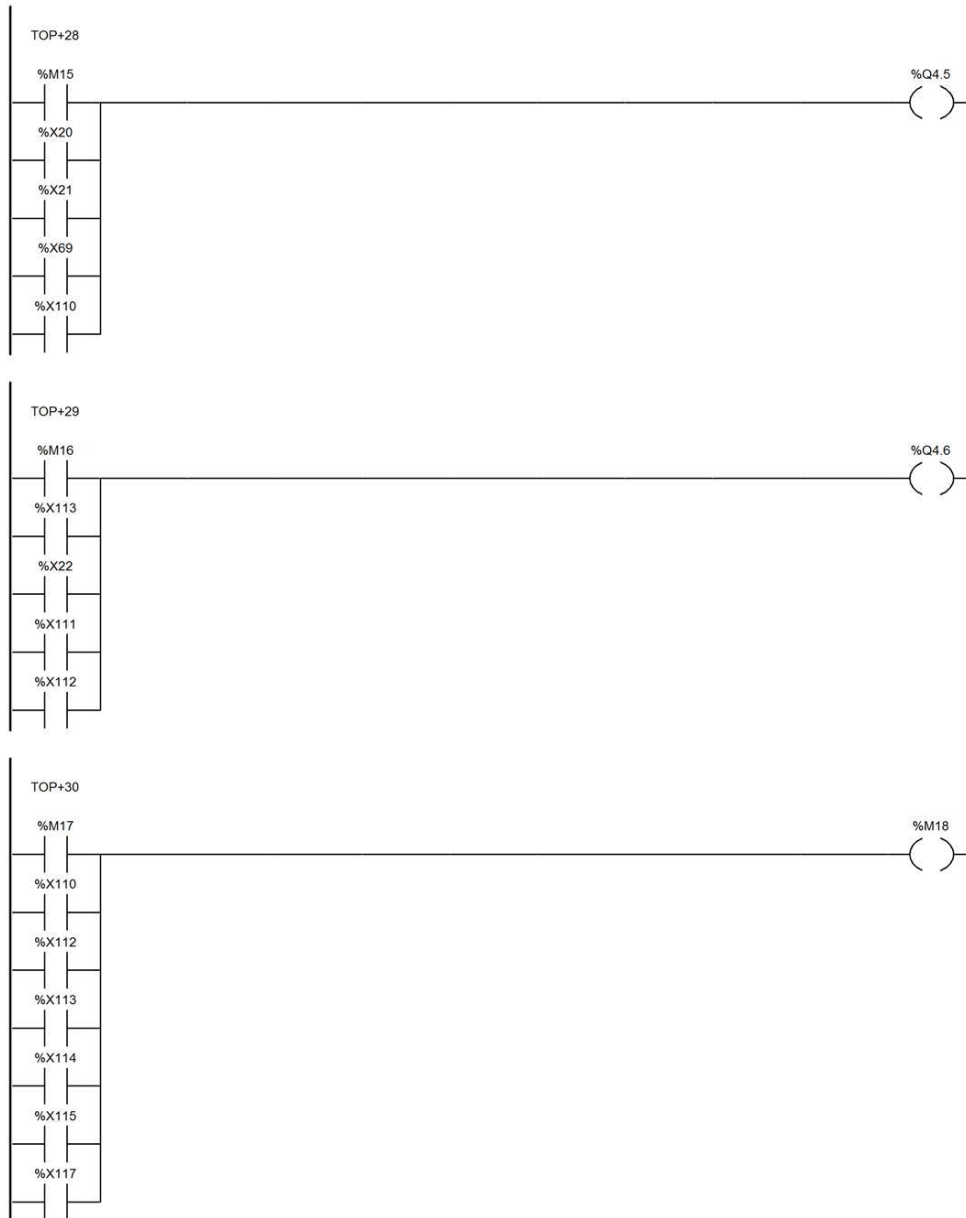
Automatización de un sistema de clasificación y almacenaje



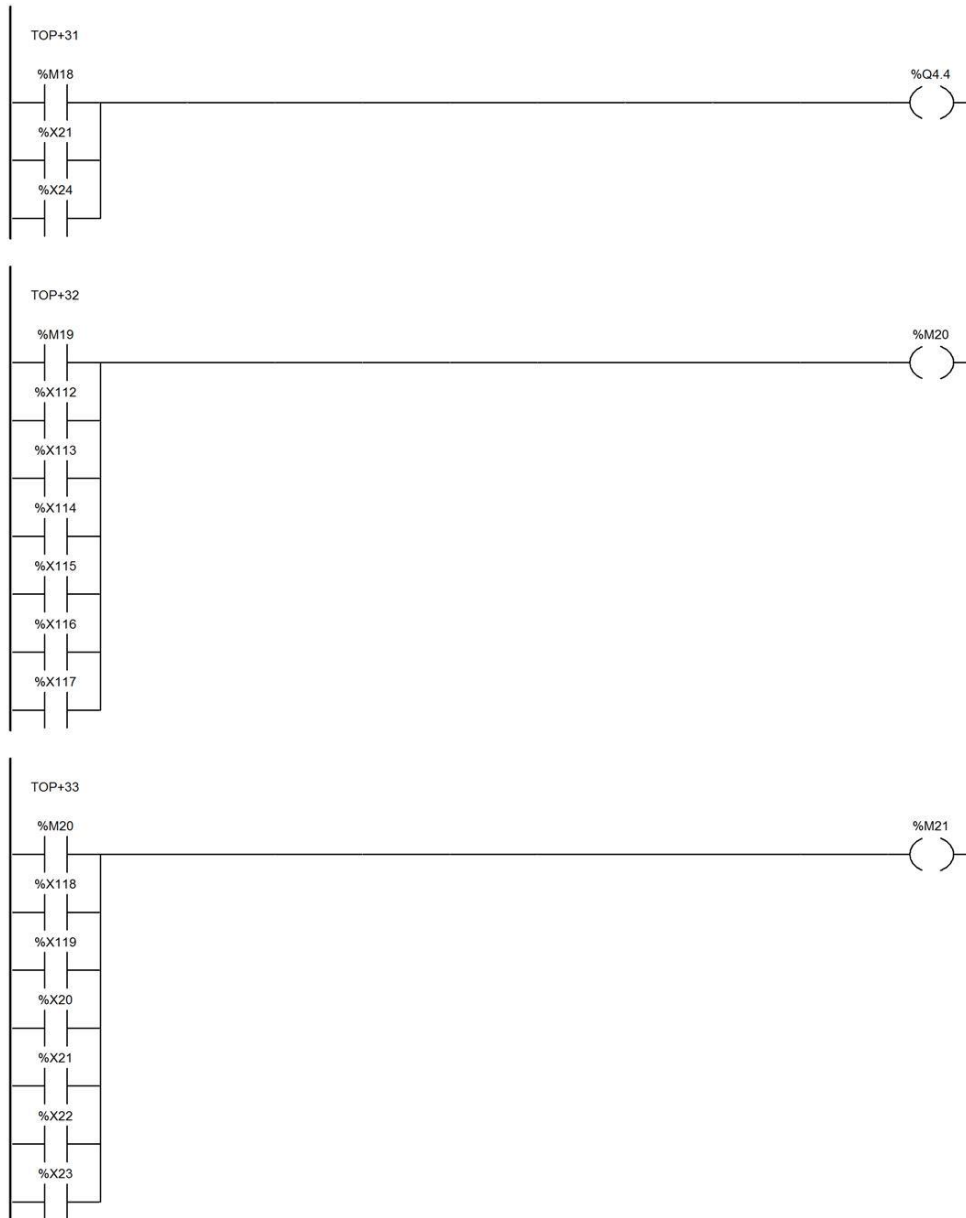
Carpeta: 9

MAST - SectionGR7 - Post

Fecha: 3/9/14



Automatización de un sistema de clasificación y almacenaje



Carpeta: 11

MAST - SectionGR7 - Post

Fecha: 3/9/14

