



Universitat de les
Illes Balears



Trabajo final de grado

GRADO EN INGENIERÍA ELECTRÓNICA INDUSTRIAL Y
AUTOMÁTICA

Análisis y programación de un sistema
basado en el PLC S7-300 de SIEMENS para
las prácticas de automatización industrial

JOSÉ LUÍS OLIVER QUINTANA

Tutor

Joan Pons Mayol

Escuela Politécnica Superior
Universidad de las Islas Baleares
Palma, 16 de mayo de 2017

Dedico este proyecto a mi familia por toda la confianza que han depositado en mi durante los años de estudio. Agradecer todo el apoyo que he recibido.

Agradecer a Joan Pons por todo el apoyo y dedicación durante la realización del TFG. Y a los técnicos del laboratorio, Ginés Valverde y Mateo Fernández, por todas las soluciones que me han aportado a los problemas surgidos con las estaciones de proceso y el entorno SIMATIC S7.

ÍNDICE GENERAL

Índice general	iii
Índice de figuras	v
Índice de cuadros	vii
Acrónimos	ix
Resumen	xi
1 Introducción	1
1.1 Estado del arte	1
1.2 Motivación del proyecto	2
2 Herramientas utilizadas	5
2.1 Componentes S7-200	5
2.2 Componentes S7-300	6
2.3 Interfaz MPI	7
2.4 Red AS-i	9
2.4.1 Maestro AS-i	10
2.4.2 Esclavo AS-i	10
2.4.3 Cable AS-i	10
2.4.4 Fuente de Alimentación	11
2.4.5 Propiedades del sistema	11
2.5 SIMATIC S7	13
2.5.1 Configuración de un proyecto en SIMATIC S7	13
2.5.2 Configuración del Hardware	14
2.5.3 Configuración Software	17
2.5.4 Ajustar interface PG/PC	19
2.5.5 Carga y simulación	20
2.5.6 Bloques del programa de usuario	20
2.6 SFCs de comunicaciones	23
2.6.1 SFC 67 “X_GET”	23
2.6.2 SFC 68 “X_PUT”	24
2.7 Estaciones de proceso	26
2.7.1 Estación de bases	26
2.7.2 Estación de rodamientos	26

2.7.3	Estación de tapas	27
2.7.4	Estación de almacenaje	28
2.7.5	Estación <i>transfer</i>	28
3	Desarrollo de la estación <i>transfer</i>	31
3.1	Funcionamiento de la estación <i>transfer</i>	31
3.2	Programa de usuario	32
3.2.1	Bloques de organización (OB1 y OB100)	33
3.2.2	Datos globales	33
3.2.3	FC3 Marcas	33
3.2.4	FC2 Salidas	36
3.2.5	FC1 Entradas	37
3.2.6	FC4 Comunicación	38
3.2.7	FC6 AS-i	41
3.2.8	FC7 Guía GEMMA de la estación <i>transfer</i>	42
3.2.9	Arranque y parada de la cinta transportadora	43
4	Desarrollo de las estaciones de alimentación	45
4.1	Sincronización con la estación <i>transfer</i>	45
4.2	Estación de bases	47
4.3	Estación de rodamientos	47
4.4	Estación de tapas	48
4.5	Guía GEMMA para las estaciones de alimentación	49
5	Desarrollo de la estación de almacenamiento	53
5.1	Funcionamiento de la estación de almacenamiento	54
5.2	Programa de usuario	54
5.2.1	Bloques de organización OB1 y OB100	54
5.2.2	Datos de usuario (UDT)	55
5.2.3	FC8 “TRATAMIENTO DE LOS MOVIMIENTOS”	55
5.2.4	FC6 “MARCAS”	56
5.2.5	FC5 “SALIDAS”	57
5.2.6	FC4 “ETAPAS”	57
5.2.7	Guía GEMMA de la estación de almacenamiento	57
6	Conclusión	59
A	ANEXO I: Guía GEMMA	61
A.1	Guía GEMMA de las estaciones de procesos	61
A.1.1	Estación de almacenamiento	61
A.1.2	Estación <i>transfer</i>	63
A.2	GRAFCETS	65
A.2.1	GRAFCET nivel 1 de la Estación Transfer-Base	65
A.2.2	GRAFCET nivel 3 de la Estación Transfer-Base	66
A.2.3	GRAFCET nivel 1 de la Estación Transfer-Rodamientos	67
A.2.4	GRAFCET nivel 3 de la Estación Transfer-Rodamientos	68
A.2.5	GRAFCET nivel 1 de la Estación Transfer-Tapas	69

A.2.6	GRAFCET nivel 3 de la Estación Transfer-Tapas	70
A.2.7	GRAFCET nivel 1 de la Estación Transfer-Almacén	71
A.2.8	GRAFCET nivel 3 de la Estación Transfer-Almacén	72
A.2.9	GRAFCET nivel 1 de la Estación de bases	73
A.2.10	GRAFCET nivel 3 de la Estación de bases	74
A.2.11	GRAFCET nivel 1 de la Estación de rodamientos	75
A.2.12	GRAFCET nivel 3 de la Estación de rodamientos	77
A.2.13	GRAFCET nivel 1 de la Estación de tapas	79
A.2.14	GRAFCET nivel 3 de la Estación de tapas	81
A.2.15	GRAFCET nivel 1 de la Estación de almacenamiento	83
A.2.16	GRAFCET nivel 3 de la Estación de almacenamiento	84
Bibliografía		85

ÍNDICE DE FIGURAS

2.1	La configuración de la subred MPI que hay en las estaciones de automatización del laboratorio.	8
2.2	Cables AS-i que dispone la estación <i>transfer</i>	11
2.3	Configuración de la Red AS-i que dispone el laboratorio	12
2.4	A la izquierda se muestra como cargar un proyecto ya creado y a la derecha para abrir un proyecto nuevo	13
2.5	Ventana principal de un proyecto SIMATIC S7	14
2.6	Módulos de la CPU y la fuente de alimentación de los PLC S7-300	15
2.7	Módulos de entrada y salida de los PLC S7-300	16
2.8	Módulo y configuración de los FM 353	16
2.9	Bastidor del PLC S7-300 de la estación <i>transfer</i> configurado	17
2.10	Carpeta contenedora de los bloques de programa de un proyecto en SIMATIC S7	18
2.11	Editor de programas de un proyecto	18
2.12	Configuración del adaptador MPI/USB en windows XP	19
2.13	Configuración para ajustar los parámetros PG/PC	19
2.14	En la ventana <i>Cargar en PG</i> aparecen los dispositivos disponibles en la interfaz MPI	20
2.15	Visualización de la SFC 67 <i>X_GET</i>	24
2.16	Visualización de la SFC 68 <i>X_PUT</i>	25
3.1	Condiciones para que se active <i>BASE_LIBRE</i>	35
3.2	La SFC <i>SET</i> activa las marcas de <i>transfer_on Estación</i>	35
3.3	El retenedor bajará cuando se haya producido el traslado de una base y las demás estaciones estén libres	36
3.4	Lectura de variables de la estación de rodamientos	39
3.5	Envío de variables a la estación de bases	40
3.6	Carga y transferencia de los valores a DB1	41
3.7	Carga de los valores de DB2 a los actuadores <i>AS_I</i>	41
3.8	Condiciones para activar/desactivar <i>MOT</i>	44
4.1	[13] Guía GEMMA que se utiliza para las estaciones de alimentación	51
4.2	Graficet de los estados de la guía GEMMA	52
5.1	Estación de almacenamiento	53
5.2	Inicialización de la tarjeta FM353_X	55

A.1	[13] Guía GEMMA de la estación de almacenamiento	61
A.2	GRAFCET que establece las condiciones exigidas por la guía GEMMA	62
A.3	[13] Guía GEMMA de la estación <i>transfer</i>	63
A.4	GRAFCET que establece las condiciones exigidas por la guía GEMMA	64
A.5	GRAFCET Nivel 1 Estación de transfer-base	65
A.6	GRAFCET Nivel 3 Estación de transfer-base	66
A.7	GRAFCET Nivel 1 Estación de transfer-rodamientos	67
A.8	GRAFCET Nivel 3 Estación de transfer-rodamientos	68
A.9	GRAFCET Nivel 1 Estación de transfer-tapas	69
A.10	GRAFCET Nivel 3 Estación de transfer-tapas	70
A.11	GRAFCET Nivel 1 Estación de transfer-almacén	71
A.12	GRAFCET Nivel 1 Estación de transfer-almacén	72
A.13	GRAFCET Nivel 1 Estación de bases	73
A.14	GRAFCET Nivel 3 Estación de bases	74
A.15	GRAFCET Nivel 1 Estación de rodamientos	75
A.16	GRAFCET Nivel 1 Estación de rodamientos	76
A.17	GRAFCET Nivel 3 Estación de rodamientos	77
A.18	GRAFCET Nivel 3 Estación de rodamientos	78
A.19	GRAFCET Nivel 1 Estación de tapas	79
A.20	GRAFCET Nivel 1 Estación de tapas	80
A.21	GRAFCET Nivel 3 Estación de tapas	81
A.22	GRAFCET Nivel 3 Estación de tapas	82
A.23	GRAFCET Nivel 1 Estación de almacenamiento	83
A.24	GRAFCET Nivel 3 Estación de almacenamiento	84

ÍNDICE DE CUADROS

2.1	[2] Características principales de la CPU 226	5
2.2	[3] Características de la fuente de alimentación Logo!Power	6
2.3	[5] Características de la fuente de alimentación PS 307	7
2.4	[5] Características del módulo de entradas/salidas digitales SM 323	7
2.5	Direccionamiento de los equipos que participan en la subred MPI en este TFG	8
2.6	[1] Características principales de la norma RS-485	9
2.7	[9] Características principales de la fuente de alimentación de AS-i	11
3.1	Direccionamiento de las variables de entrada AS-i	34
3.2	Direccionamiento de las variables de salida de los actuadores	34
3.3	<i>Estación_Libre</i>	34
3.4	<i>X_GET</i>	38
3.5	Las marcas que el programa de usuario envía a las diferentes estaciones . .	40
4.1	Bytes reservados para las variables que lee el <i>transfer</i>	46
4.2	Bytes reservados para la recepción de datos enviados por el <i>transfer</i>	46
4.3	Envío de variables a la estación <i>transfer</i>	46
4.4	Recepción de variables que provienen del <i>transfer</i>	47

ACRÓNIMOS

AS-i AS-Interface

CPU *Central Processing Unit*

DW Double Word: Bloque de memoria formado por 4 bytes

EPS Escuela Politécnica Superior

GEMMA *Guide d'Etudes des Modes de Marches et d'Arrêts*

GRACET *Graphe Fonctionnel de Commande Etape Transition*

PLC Programmable Logic Controller

MPI *Multi Point Interface*

OP Operator Panel

PG Cable para conectar una unidad de programación

TFG Trabajo final de grado

TP *Twisted Pair*

UIB Universitat de les Illes Balears

W Word: Bloque de memoria formado por 2 bytes

RESUMEN

Este Trabajo final de grado (TFG) ha sido diseñado y realizado para comprender la comunicación entre un conjunto de estaciones que simulan un proceso de automatización industrial. Están compuestos por Programmable Logic Controller (PLC) S7 200 y S7 300 de Siemens y está a disposición de los alumnos que cursan las asignaturas de *Automatización Industrial* y *Ampliación de Automatización Industrial* del grado de Ingeniería Industrial de la Universitat de les Illes Balears (UIB).

En este proyecto se describen las herramientas necesarias como el *Multi Point Interface* (MPI), que hace posible la comunicación entre diferentes dispositivos, o la red AS-i, donde se produce el intercambio de datos entre sensores/actuadores y un PLC. Se ha escrito una guía de aprendizaje del software SIMATIC Step7 que detalla la creación de un proyecto, la configuración hardware, y los elementos que hacen posible la programación que posteriormente es cargado en la CPU de un S7 300.

Se ha realizado una breve explicación del funcionamiento de las estaciones de proceso en las que se centra el TFG y los programas de usuario elaborados. En el desarrollo de la memoria se exponen los pasos a seguir para la comunicación entre ellas. En los anexos se muestran los GRAFCETs y la guía GEMMA de cada estación.

Este TFG tiene un fin didáctico para los alumnos que cursan las asignaturas anteriormente mencionadas y tiene la finalidad de establecer un procedimiento común que posibilite la comunicación de cada una de las estaciones individuales con la estación *transfer* que permita la sincronización entre estaciones y la operación conjunta de todos los módulos que forman el proceso.

INTRODUCCIÓN

1.1 Estado del arte

Con el surgimiento de diferentes tecnologías en la primera mitad del siglo XX [1], apareció la dificultad sobre el control de los procesos industriales solamente con la invención humana debido al tamaño de producción y sus costes. Surgió la necesidad de crear tecnología capaz de procesar y memorizar variables físicas.

La creación de dispositivos electrónicos, sensores, facilitó que variables físicas como temperatura, presión, posición, desplazamiento, pudieran ser transformadas a variables eléctricas. Estos dispositivos tienen la precisión para captar valores con mayor rapidez y sensibilidad que la capacidad humana. Los sistemas capaces de procesar estas variables eléctricas y a partir de ellas obtener otras variables binarias reciben el nombre de controladores lógicos.

“Los procesos industriales se pueden definir como un conjunto de acciones, realizadas por una o más máquinas adecuadamente coordinadas que dan como resultado la fabricación de un producto.” (Mandado, Marcos, Fernandez, Armesto(2009))

A principios del siglo XX, se desarrollaron controladores lógicos que carecían de unidad operativa. Por una parte, están los controladores lógicos combinacionales caracterizados por emitir variables de salida que dependen del valor de las entradas en ese preciso momento. No es un sistema adecuado para tomar decisiones en función de la secuencia de estados en sucesivos instantes de tiempo. Por otra parte, existen los controladores lógicos secuenciales, caracterizados por tener la capacidad de memorizar variables de entrada. Pueden emitir una variable de salida en función del valor de las variables de entrada en el pasado. Esto es posible por el uso de biestables.

La desventaja de los controladores lógicos sin unidad operativa es que no disponen de medios para comprobar que operación se está ejecutando en un instante de tiempo

ya que no hay un código de instrucciones donde realizar el seguimiento. Con el progreso de la microelectrónica, se consiguieron fabricar equipos eléctricos y electrónicos para el diseño de controladores lógicos con unidad operativa.

“Los controladores lógicos programables son procesadores digitales secuenciales programables que actúan sobre las variables de salida mediante la ejecución de una secuencia de instrucciones.”(Mandado, Marcos, Fernandez, Armesto(2009))

Al principio, se desarrollaron procesadores capaces de ejecutar ecuaciones lógicas de tipo Y y O emulando sistemas digitales combinacionales o secuenciales encargándose de las variables de entrada y salida a través de la ejecución de instrucciones en una secuencia única. Introduciendo la *Central Processing Unit (CPU)*, se crearon autómatas programables con unidad operativa que contiene una unidad aritmética y lógica que permite realizar operaciones simultaneas de variables binarias y procesar variables analógicas como información numérica.

1.2 Motivación del proyecto

En el laboratorio de robótica de la Escuela Politécnica Superior (EPS), los estudiantes del grado de ingeniería industrial disponen de cinco estaciones de automatización. El objetivo es simular el proceso de creación de una pieza a través de tres elementos y de su almacenamiento. El funcionamiento es el siguiente, una estación dispone de una cinta transportadora que desplaza una serie de palés. Estos se detienen delante de tres estaciones que seleccionan una serie de elementos que forman una pieza completa. Por último, son desplazados hasta una estación encargada de su almacenaje.

De las cinco estaciones, hay tres que se encargan de la selección de un elemento que conforma la pieza. Estas disponen, cada una, de un PLC S7 200 de Siemens. Los programas de usuario son creados a través del software Step7 Microwin. En las asignaturas de *Automatización Industrial* y *Ampliación de Automatización Industrial*, los estudiantes han de realizar una serie de prácticas para el tratamiento de una estación en específico. En cambio, la estación *transfer* y la de almacenamiento son programadas a través de SIMATIC Step7 ya que los PLCs pertenecen a la gama S7 300.

Este TFG ha sido diseñado para comprender la comunicación entre PLCs e indicar a los futuros alumnos como conseguir, mediante una serie de pasos, comunicar sus programas de usuario con la estación *transfer* y poder ver el funcionamiento de todas las estaciones al completo.

En este documento se describen las herramientas, capítulo 2, que hacen posible el funcionamiento del conjunto de las estaciones. Por una parte, se exponen los componentes de los PLCs S7 200 y S7 300 que se disponen en el laboratorio. Por otra parte, se detalla la descripción y características del MPI que permite la comunicación entre los PLCs. También se ha introducido una explicación de la red AS-i ya que es el bus de

comunicación entre los actuadores y sensores y la CPU de la estación *transfer*.

Los programas de usuario del S7 300 son programados en SIMATIC Step7. En este informe se incluye una guía para crear un proyecto desde la configuración hardware y software hasta el proceso de carga en la CPU. Otra finalidad para comprender a utilizar SIMATIC es porque ofrece funciones preprogramadas, *X_GET* y *X_PUT*, que hacen posible el envío de información entre PLCs. En el apartado 2.6 se explica el funcionamiento de dichas funciones ya que se utiliza en el programa de usuario de la estación *transfer*.

En el capítulo 3 se expone el funcionamiento de la estación *transfer*. Se detallan los distintos bloques del programa de usuario con las características principales como los actuadores/sensores que maneja la cinta transportadora, el uso de las marcas que leen el resto de las estaciones, el código principal que se carga en la CPU 314 IFM y el uso de las funciones *X_GET* y *X_PUT*. Es necesario entender su funcionamiento ya que la estación *transfer* es la que comunica con el resto de las estaciones. Es la encargada de avisar cuando cada estación tiene a disposición un palé para realizar el traslado de un elemento y el encargado de comprobar que las demás estaciones están activas.

En este proyecto se han programado todas las estaciones que componen el proceso anteriormente explicado. En el capítulo 4 se hace referencia a las estaciones de bases, rodamientos y tapas. Se muestran las marcas utilizadas para comunicar con la estación *transfer* y los espacios de memoria que se han reservado para el envío/recepción de datos. También, se ha expuesto el funcionamiento de cada una de las tres estaciones detallando cuando y como se realiza el envío de sus elementos a los palés que dispone la cinta transportadora. Se ha realizado un análisis de la *Guide d'Etudes des Modes de Marches et d'Arrêts (GEMMA)* en cada una de las estaciones y en el apartado de anexos se puede consultar el *Graphe Fonctionnel de Commande Etape Transition (GRACET)* de primer y tercer nivel de cada uno de ellos.

Por último, se explica el funcionamiento del programa de usuario cargado en la CPU de la estación de almacenamiento. Se detallan los bloques de programa utilizados y los pasos realizados para conseguir que las piezas creadas se vayan introduciendo en una superficie de treinta posiciones.

En definitiva, este informe ha sido realizado para que futuros alumnos puedan comprender el funcionamiento completo de las estaciones de procesos en el que trabajarán en las sesiones de prácticas. Mediante una serie de pasos podrán conseguir comunicar la estación que programen con la estación *transfer*.

HERRAMIENTAS UTILIZADAS

2.1 Componentes S7-200

Los tres PLCs S7-200 del laboratorio disponen de los siguientes componentes:

- Fuente de alimentación Logo!Power.
- CPU 226
- Módulo de señales analógicas EM 231 AI4/x12Bit (solamente la estación de rodamientos).

El software de programación de los PLC S7-200 es el STEP 7 Micro/WIN [2]. Para la conexión PC/CPU se utiliza el cable de RS-232/PPI multimaestro. En el grado de Ingeniería Industrial de la EPS, los alumnos aprenden a manejar este entorno de programación desde la creación de un programa de usuario, la verificación de los parámetros de comunicación, el establecimiento de la comunicación con el PLC, hasta la carga del programa.

Datos técnicos	CPU 226
Referencia	216-2AD23-0XB0
Memoria de datos	10240 bytes
Entradas integradas digitales	24
Salidas integradas digitales	16
Reloj de tiempo real	Incorporado
Tamaño de la imagen de entradas digitales	128
Tamaño de la imagen de salidas digitales	128

Cuadro 2.1: [2] Características principales de la CPU 226

2. HERRAMIENTAS UTILIZADAS

Datos técnicos	Logo!Power
Referencia	6EP1331-1SH03
Tensión nominal de entrada	100/240V AC
Frecuencia de red	50/60 Hz
Tensión nominal de salida	DC 24 V
Intensidad de salida	1.3 A

Cuadro 2.2: [3] Características de la fuente de alimentación Logo!Power

2.2 Componentes S7-300

Los dos PLC S7-300 del laboratorio disponen de los siguientes componentes:

- Fuente de alimentación PS 307.
- CPU 314 IFM
- Módulo señales digitales 16DI/16DO.
- Módulo de entradas digitales DI4xDC24V
- Módulo de señales analógicas AI4/x12Bit
- CP 342-2 (solamente en la estación *transfer*)

La CPU 314 IFM [4] se caracteriza, entre otras cosas, por ofrecer la función integrada de posicionamiento. Permite trabajar en paralelo al programa de usuario y prolonga mínimamente el tiempo de ciclo de la CPU. Los bloques de datos (DB de instancia) permiten el intercambio de información entre la función integrada y el programa de usuario. La primera es la que escribe y lee en este bloque de datos y accede a las entradas y salidas integradas sin pasar por el programa de usuario.

Las funciones integradas en la estación *transfer* son las entradas y salidas que proporciona la red AS-i para el posicionamiento de piezas sobre la cinta transportadora. Por otra parte, la estación de almacenaje contiene dos servomotores de corriente alterna que son los que mueven el brazo posicionador en los ejes **x** e **y**.

En las siguientes tablas, se muestran características principales varios componentes que forman parte de los PLC. En el apartado 2.4.1 se muestran las características principales del CP 342-2.

Datos técnicos	
Referencia	6ES7307-1BA01-0AA0
Tensión nominal de entrada	120/230V AC
Frecuencia de red	50/60 Hz
Tensión nominal de salida	DC 24 V
Intensidad de salida	2 A

Cuadro 2.3: [5] Características de la fuente de alimentación PS 307

Datos técnicos	
Referencia	6ES7323-1Bl00-0AA0
Número de entradas	16
Número de salidas	16
Tensión nominal de entrada	DC 24 V
Tensión nominal de salida	24 V DC
Intensidad de salida	0,5 A

Cuadro 2.4: [5] Características del módulo de entradas/salidas digitales SM 323

2.3 Interfaz MPI

Interfaz MPI (Multi Point Interface) [6] es un sistema de bus de Siemens desarrollado como interfaz de programación para los Simatic S7. Sirve como medio de comunicación homogénea entre dispositivos de automatización Siemens.

Está diseñado para redes pequeñas, con poca extensión, pocos interlocutores y un volumen de datos a intercambiar reducido. Las características son las siguientes:

- Todos los equipos que se conectan a la red MPI son llamados “estaciones”.
- Se permiten hasta 127 estaciones por cada subred, siendo direccionados desde 0 hasta 126.
- Solamente pueden estar activas un máximo de 32 estaciones a la vez (0 para el PG, 1 para el OP). Para la comunicación entre estaciones, es preciso asignarles una dirección, diferentes entre ellas.
- Hay dos formas de direccionar una subred. Una de ellas es que la CPU obtenga las direcciones MPI de las CP que ha ajustado en STEP 7 o que la CPU adopte por defecto las direcciones MPI de las CP en su instalación.
- La velocidad de transferencia predeterminada es de 187,5 Kbits/s en todas las CPUs. Aunque para los S7-200 se puede ajustar a 19.2 Kbits/s.
- El método de acceso es mediante el bus Token.
- Los equipos que se pueden conectar a la Subred MPI son:
 - PLC S7-300 y S7-400 con interfaz MPI

2. HERRAMIENTAS UTILIZADAS

Estación	Dispositivo	Dirección MPI
	PC	0
<i>transfer</i>	CPU 314 IFM	2
Bases	CPU 226	3
Rodamientos	CPU 226	4
Tapas	CPU 226	5
Almacén	CPU 314 IFM	6
Motor 1	FM 353	7
Motor 2	FM 353	8

Cuadro 2.5: Direccionamiento de los equipos que participan en la subred MPI en este TFG

- S7-200
 - Operator Panel (OP)/ Twisted Pair (TP)
 - Cable para conectar una unidad de programación (PG)/PC
 - Módulos de función
- La máxima longitud de un segmento de una subred MPI puede ser de 50m.
- El medio que conecta las diferentes estaciones es el bus RS-485.

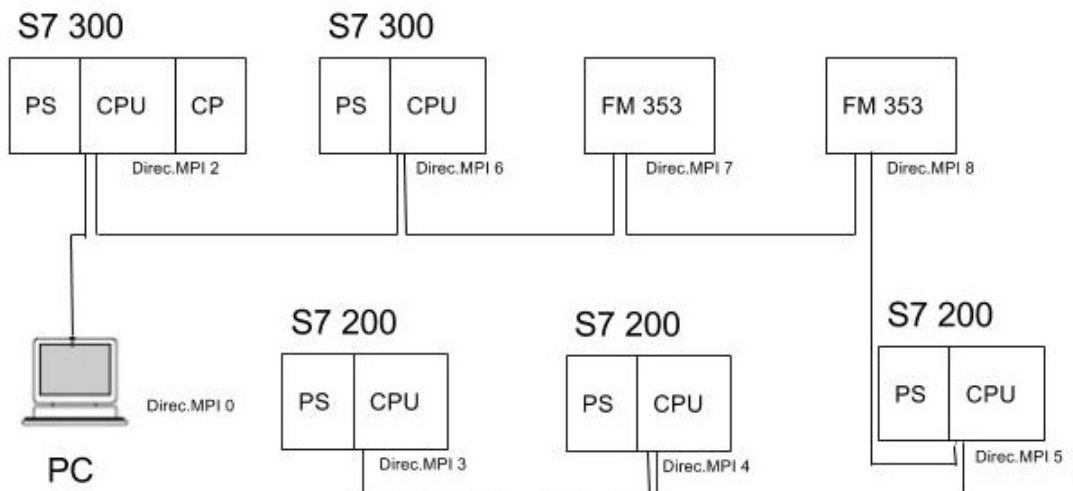


Figura 2.1: La configuración de la subred MPI que hay en las estaciones de automatización del laboratorio.

La comunicación entre las diferentes CPU se realiza mediante la norma RS-485. Es un estándar normalizado para la transmisión de señales, sincronización y detección de errores. Permite conectar hasta 32 transmisores y 32 receptores para una comunicación serie mediante un bus de campo. La comunicación es asíncrona ya que el transmisor y

el receptor contienen señales de reloj independientes.

La transmisión se realiza en banda base digital y se realiza en los dos sentidos. Un procesador de comunicaciones que actúa como principal y el resto como subordinados. Las características principales del RS-485 se muestra en la tabla siguiente.

Datos técnicos	RS-485
Modo de enlace físico	Diferencial
Máximo número de transmisores	32
Máximo número de receptores	32
Distancia máxima (en pies)	4000
Velocidad Máxima	10 Mbits/s
Máx tensión de salida del amplificador de señal	-7 a +12V
Tensión del amplificador de salida con carga	-1,5 a 1,5 V
Tensión del amplificador de salida sin carga	-5 a +5V
Impedancia de los amplificadores de salida	54
Tensión de entrada del receptor	-7 a +12V
Sensibilidad de entrada del receptor	-200mV a +200mV
Impedancia de entrada del receptor	12K min

Cuadro 2.6: [1] Características principales de la norma RS-485

2.4 Red AS-i

AS-Interface (**AS-i**) es un sistema de enlace para el nivel de procesos en instalaciones de automatización. Las características principales son las siguientes[7]:

- Se utiliza para la conexión de actuadores y sensores binarios. Realiza el intercambio de datos entre los sensores/actuadores (esclavos) y el procesador de comunicaciones (maestro) a través del cable AS-i.
- La configuración de la red es de tipo árbol.
- Se permite la conexión mediante un único canal de comunicación.
- La longitud de los mensajes que se transmiten son constantes. El tiempo de ciclo entre el maestro con las estaciones conectadas (se pueden conectar hasta un máximo de 31 esclavos) es de 5ms (Tiempo de ciclo típico de un autómata programable).
- Los esclavos pueden tener conectados hasta 8 sensores/actuadores binarios convencionales.
- En la red AS-i pueden conectarse como máximo 248 entradas y salidas en el caso de módulos estándar.

2.4.1 Maestro AS-i

Permite la conexión entre el S7-300 a la red AS-i [7]. En el laboratorio se dispone del procesador de comunicaciones CP 342-2. Sus características principales son las siguientes:

- Para un máximo de 31 esclavos, el tiempo de ciclo es de 5ms.
- Dispone de 16 bytes de entrada y 16 bytes de salida.
- El acceso de lectura/escritura se produce por Word: Bloque de memoria formado por 2 bytes (**W**) o Double Word: Bloque de memoria formado por 4 bytes (**DW**) a direcciones pares E/S.
- Para configurarlo se necesita, como mínimo, la versión V5.2 del STEP 7.

2.4.2 Esclavo AS-i

Cuando el maestro puede actuar sobre una estación, a este último se le llama esclavo [7]. Cada uno está identificado por una dirección y se establece mediante la conexión de la consola de direccionamiento en el propio bus AS-i. Los esclavos son canales de entrada y salida que se activan cuando son llamados por el maestro.

En el laboratorio se dispone de 6 esclavos estándar. Se tratan de módulos E/S digitales con los siguientes datos técnicos[8]

- 8 canales E/S.
- La tensión de alimentación está entre 26,5V y 31,6V y un consumo total de corriente máxima de 270mA.
- La alimentación se produce mediante la red AS-i.
- Dispone de dos tipos de display de estado. El primer display es para las entradas/salidas y el segundo para el cable AS-i.

2.4.3 Cable AS-i

Es el medio físico por donde transmiten datos y alimentan los sensores/actuadores conectados al módulo AS-i. Se tratan de dos cables bifilares engomados con una sección de 1.5mm² y una longitud máxima de 100m.

Los cables perfilados son los siguientes:

- Amarillo: Es el cable donde se produce el intercambio de datos. Es un cable amarillo con cubierta de caucho con una sección de 1.5mm². El número de referencia es 3RX9010-0AA00.
- Negro: Cable de alimentación auxiliar a 24 VDC. El número de referencia es 3RX9020-0AA00.

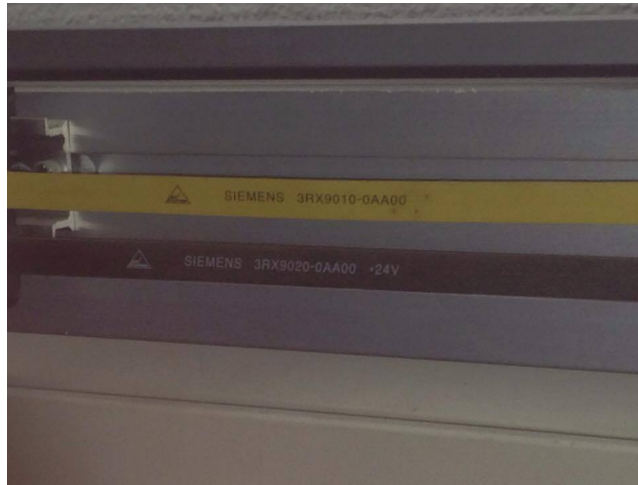


Figura 2.2: Cables AS-i que dispone la estación *transfer*

2.4.4 Fuente de Alimentación

Es el componente que se encarga de suministrar energía a los elementos conectados a la red AS-i.

Datos técnicos	
Referencia	3RX9307-0AA00
Tensión nominal de entrada	115/230V AC
Frecuencia de red	50/60 Hz
Tensión nominal de salida	DC 31 V
Intensidad de salida	2.4 A

Cuadro 2.7: [9] Características principales de la fuente de alimentación de AS-i

2.4.5 Propiedades del sistema

El funcionamiento del AS-i puede ser resumido en tres pasos:

- En la red hay un maestro que realiza intercambio de datos y llama a todos los esclavos, identificándolos por una dirección, esperando una respuesta. Es una comunicación de sondeo periódico.
- El maestro confirma que se efectúa la transmisión de los datos y de tensión en la línea. Detecta los errores de transmisión y, en caso de error, lo comunica a la CPU.
- La CPU y el maestro se transmiten los datos de los procesos y los comandos de parametrización.

2. HERRAMIENTAS UTILIZADAS

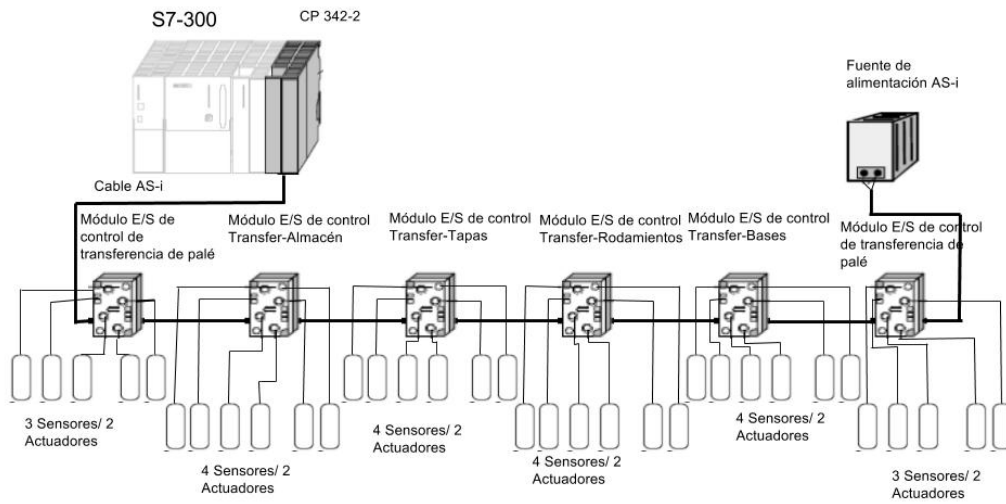


Figura 2.3: Configuración de la Red AS-i que dispone el laboratorio

Si durante el funcionamiento normal se cambia o incorpora un esclavo, no se perturba dicho funcionamiento.

2.5 SIMATIC S7

Step 7 [10] es el software de acceso a la configuración y programación de los PLC S7-300 que se disponen en el laboratorio. Esta herramienta es indispensable para la configuración y la programación de los bloques de los autómatas de la estación *transfer* y almacén.

“El administrador SIMATIC nos permite:

- Crear y gestionar proyectos,
- Configurar y parametrizar el hardware,
- Configurar redes de hardware,
- Programar bloques,
- Cargar y comprobar el funcionamiento del programa,
- Diagnosticar fallos de instalación,”(SIMATIC,2006,p.5-1)

2.5.1 Configuración de un proyecto en SIMATIC S7

Accediendo al “Administrador SIMATIC” aparece la ventana “ Asistente de Step 7 ”. Si se desea crear un proyecto rápido como aparece en la derecha de la figura 2.4, basta con pulsar **Finalizar**. En la ventana inferior derecha se muestra el bloque de función OB1 que se crea de forma predeterminada.

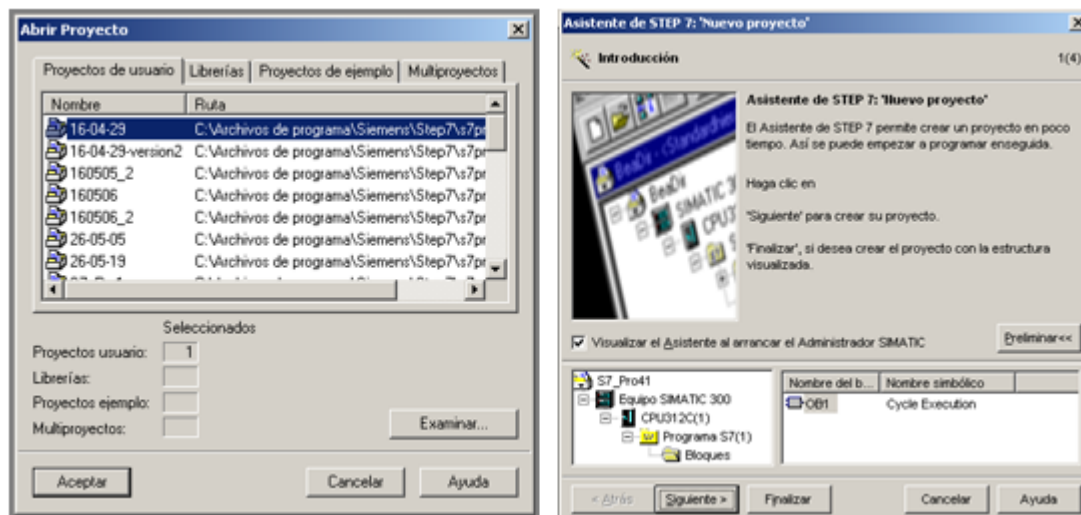


Figura 2.4: A la izquierda se muestra como cargar un proyecto ya creado y a la derecha para abrir un proyecto nuevo

Una vez se ha creado el proyecto, Step 7 muestra la ventana principal 2.5. En la parte izquierda aparece el árbol del proyecto y en la cabeza se muestra el nombre del

2. HERRAMIENTAS UTILIZADAS

proyecto. El segundo nivel lo forman los equipos o subredes. Y por último, los objetos que forman parte del segundo nivel. En la parte derecha de la ventana del proyecto se visualiza el contenido de cada nivel.

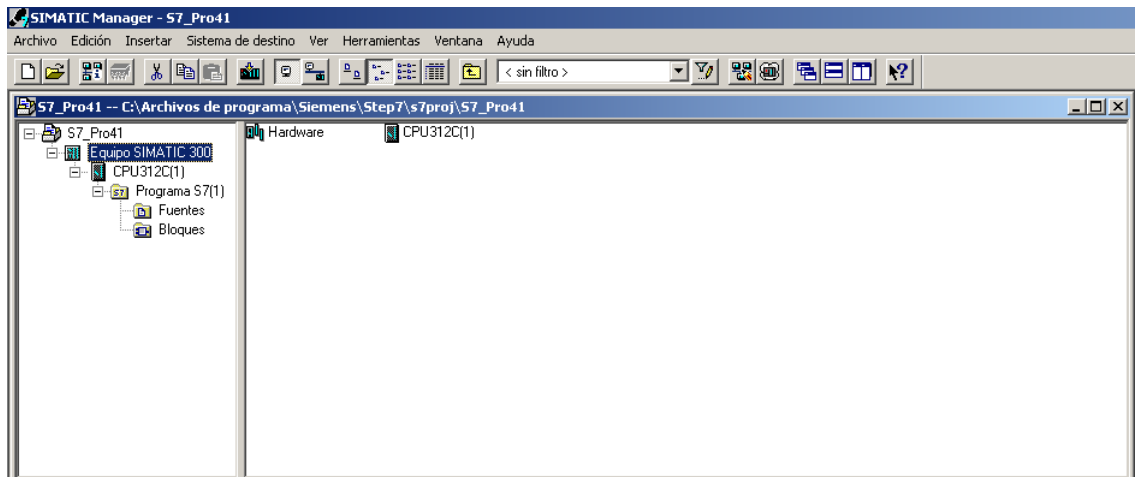


Figura 2.5: Ventana principal de un proyecto SIMATIC S7

Otra forma de crear un proyecto de forma manual es accediendo al comando de menú **Archivo >Nuevo**. También es posible abrir un proyecto ya creado accediendo a **Archivo >Abrir**. Para esta última opción, aparece una ventana como se refleja en la parte izquierda de la figura 2.4, donde se muestran los último proyectos realizados. Para abrir un proyecto de alguna carpeta se realiza a través de la opción **Examinar**.

2.5.2 Configuración del Hardware

Cuando se crea un proyecto, de forma predeterminada, en el árbol de programa aparece el icono de un equipo. Aun así, también se pueden crear más equipos mediante el comando de menú **Insertar >Equipo**. Y se puede introducir un nombre a cada equipo pulsando botón derecho y accediendo a “cambiar nombre”.

Para realizar la configuración hardware del equipo, cada uno tiene un objeto denominado “Hardware”. Pulsando dos veces el objeto, aparece la ventana “HW Config” que está compuesto por tres ventanas. En la ventana superior izquierda es donde se introduce el bastidor que almacenará los componentes del equipo. Haciendo click derecho sobre la ventana y mediante los comandos **Insertar objeto >SIMATIC 300 >BASTIDOR 300 >Perfil soporte** se crea un bastidor. En la figura 2.9 se puede observar cómo queda el bastidor una vez configurado.

En la ventana derecha de “HW Config” aparece el catálogo de Hardware. En el siguiente ejemplo se explican los pasos a seguir para configurar el PLC que dispone la estación *transfer*. El primer paso es introducir la CPU 314 IFM con la referencia 6ES7 314-5AE03-0AB0/V1.2. La figura 2.6 muestra como en la carpeta CPU-300 aparecen diferentes CPU de la familia S7. Una vez seleccionada, la CPU aparece en la segunda

posición del bastidor. Se debe configurar las características de la interfaz MPI para esta CPU, la velocidad de transferencia y la dirección MPI. Además se puede configurar el tiempo de ciclo máximo y propiedades de arranque.

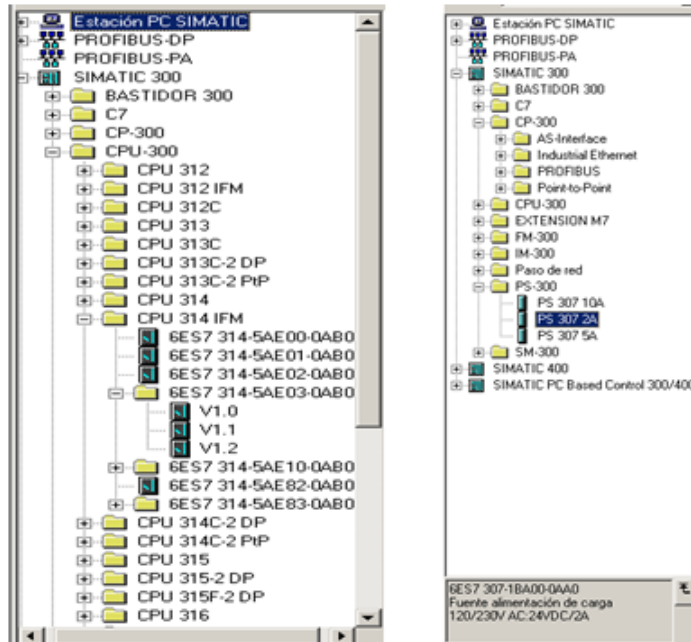


Figura 2.6: Módulos de la CPU y la fuente de alimentación de los PLC S7-300

El segundo paso es añadir la fuente de alimentación del PLC. Como se ha hecho con la CPU, se busca en el catálogo de hardware el PS 307 2A que aparece en la carpeta PS-300. Por último, se añaden los módulos de señales como se muestra en la figura 2.7.

- DI16/DO16xDC24,
- DI4xDC24V (En la configuración se ha introducido el DI16xDC24V).
- AI4/AO1x12Bit (En la configuración se ha introducido el AI4/AO2x12Bit).

La configuración del PLC de la estación *transfer* ya está realizado. La estación Almacén tiene dos componentes más, los motores FM 353 STEPPER. En la figura 2.8 se muestra como configurarlos.

Un ejemplo de hardware configurado es el que se muestra en la figura 2.9. En el bastidor aparecen todos los componentes y en la ventana inferior sus características. Para completar el proceso, hay que guardar y compilar la configuración mediante el comando de menú **Equipo > Guardar y compilar**.

El siguiente paso es configurar la subred MPI. Desde la ventana principal del proyecto y pulsando el botón derecho del ratón aparece una ventana. Mediante el comando **Insertar nuevo objeto > MPI** se accede a la ventana “NetPro” donde se pueden unir a la subred MPI los equipos configurados.

2. HERRAMIENTAS UTILIZADAS

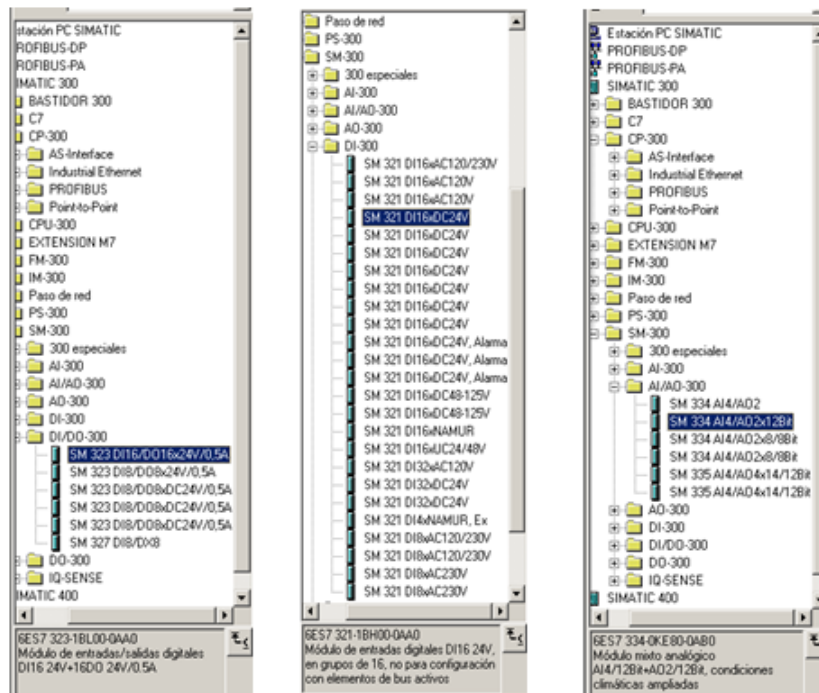


Figura 2.7: Módulos de entrada y salida de los PLC S7-300

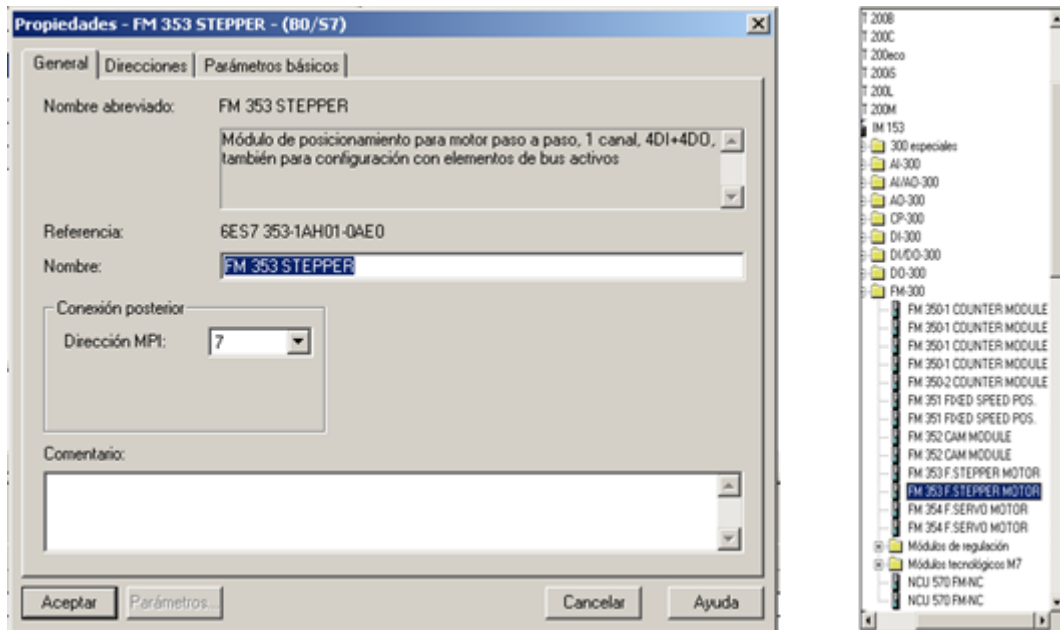


Figura 2.8: Módulo y configuración de los FM 353

2.5. SIMATIC S7

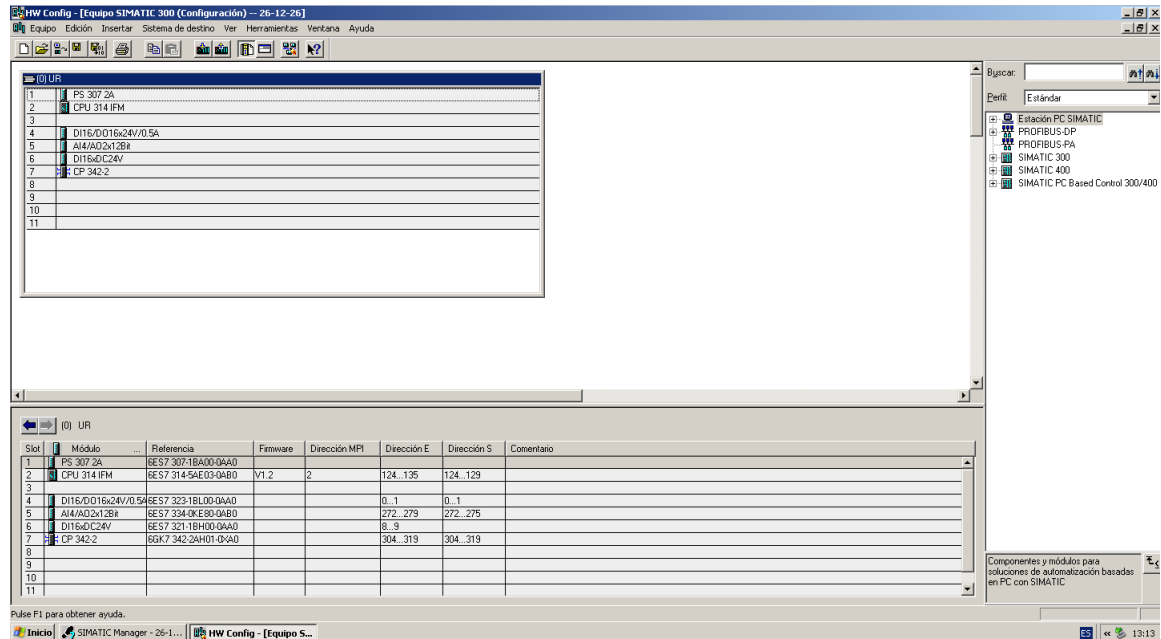


Figura 2.9: Bastidor del PLC S7-300 de la estación *transfer* configurado

2.5.3 Configuración Software

Una vez creada la configuración hardware, volviendo a la ventana del proyecto, el segundo paso es la configuración software. Si se accede al árbol del proyecto, cada equipo dispone de una carpeta donde aparecerán los bloques de programación que se cargarán en la CPU. Accediendo al objeto “Bloques” y pulsando el botón derecho aparece el comando **Insertar Objeto** donde se permite insertar los siguientes objetos:

- Bloque de organización (OB)
- Bloque de función (FB)
- Función (FC)
- Bloque de datos (DB)
- Tipo de datos (UDT)
- Tabla de variables (VAT)

Cuando el usuario se dispone a crear un objeto, hay que introducir el nombre, el nombre simbólico y el lenguaje en el que se va a programar dicho objeto. Respecto a este último, Step 7 permite el uso de los siguientes lenguajes de programación:

- Lista de instrucciones (AWL)
- Esquema de contactos (KOP)
- Diagrama de funciones (FUP)

2. HERRAMIENTAS UTILIZADAS

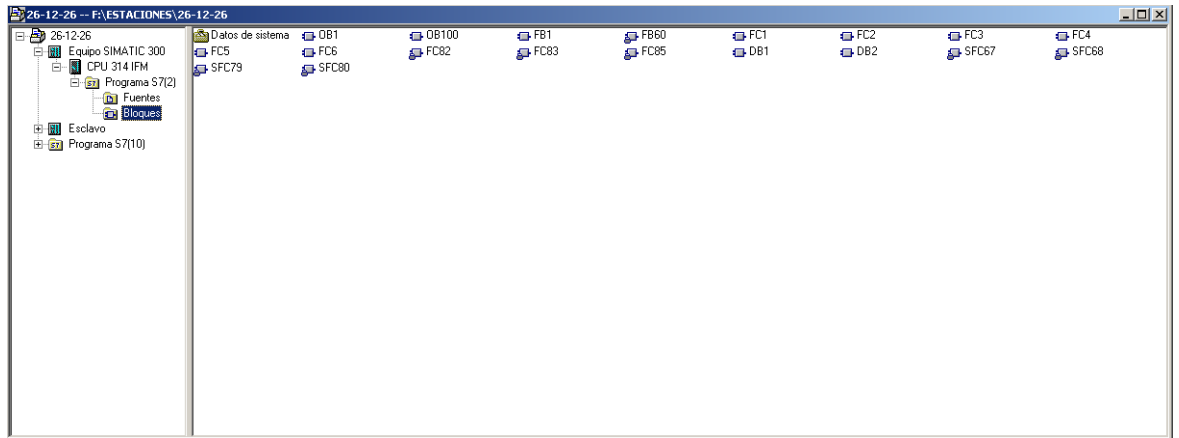


Figura 2.10: Carpeta contenedora de los bloques de programa de un proyecto en SIMATIC S7

Una vez creado un bloque de programa, se puede acceder a programarlo. Clickando sobre un bloque, aparece la ventana del editor de programas. Un ejemplo sería la figura 2.11. En la ventana de la izquierda aparecen los elementos de los lenguajes y bloques de librerías [11]. La ventana central muestra el área de instrucciones. Arriba aparece el título y el comentario del bloque. Cada uno está compuesto por los diferentes segmentos de programación que se diferencian por el título del segmento, el comentario y las instrucciones. Para guardar el trabajo realizado, mediante el comando de menú **Archivo > Guardar** se almacena toda la información del bloque.

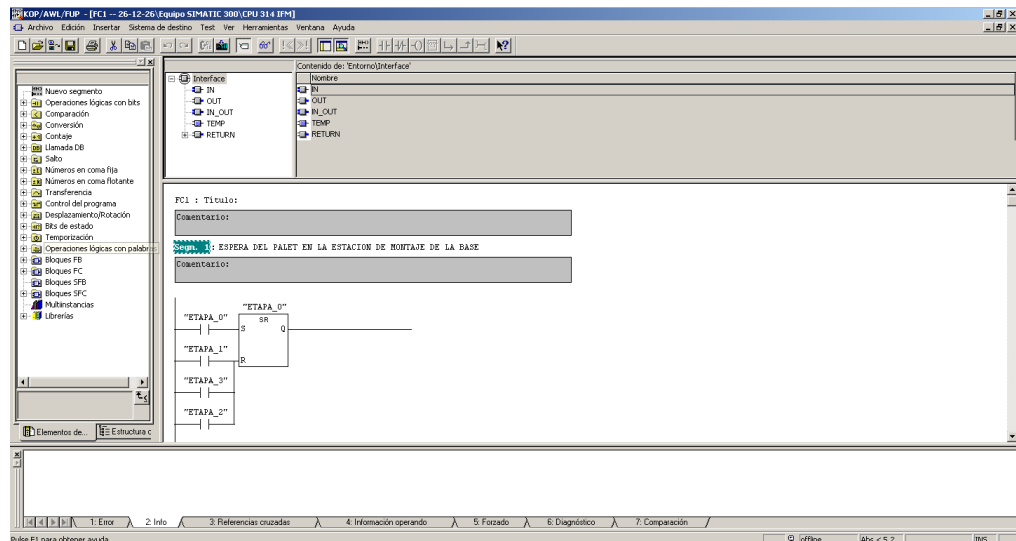


Figura 2.11: Editor de programas de un proyecto

2.5.4 Ajustar interface PG/PC

En el laboratorio de robótica se puede tener acceso al SIMATIC S7 desde Virtual Box. Debido a las dificultades de conexión entre el cable RS-232 y virtual box, se ha decidido utilizar un adaptador MPI/USB. Desde el escritorio y accediendo a los comandos **Inicio >Panel de control >Herramientas administrativas >Administración de equipos >Administrador de dispositivos >Puertos (COM & LPT)**, click derecho en el dispositivo y pulsando en **Propiedades**, se configura el adaptador MPI/USB.

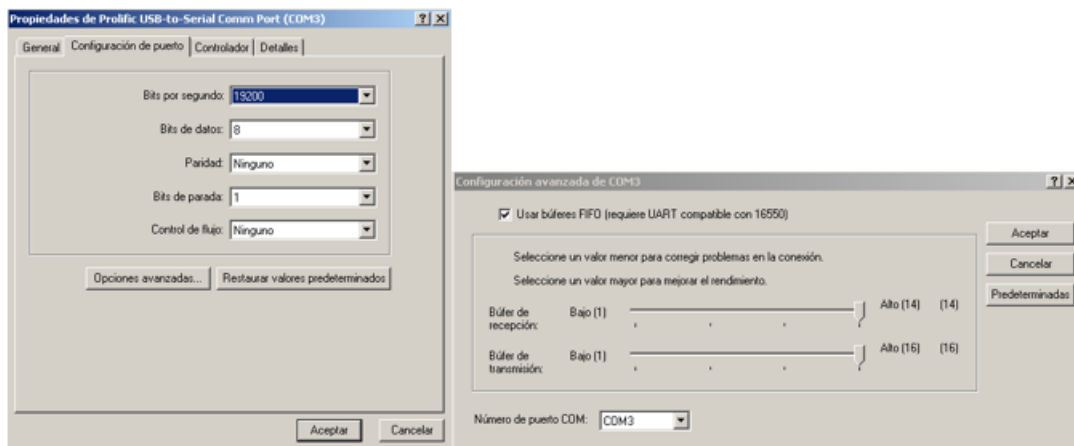


Figura 2.12: Configuración del adaptador MPI/USB en windows XP

Antes de cargar la configuración del proyecto en la CPU, se ha de establecer una comunicación entre el PC y la CPU. Accediendo a **Herramientas >Ajustar interface PG/PC** se puede configurar una comunicación vía MPI. Escogiendo la parametrización *PC Adapter (MPI)* y ajustando las propiedades como se muestra en la figura 2.13.

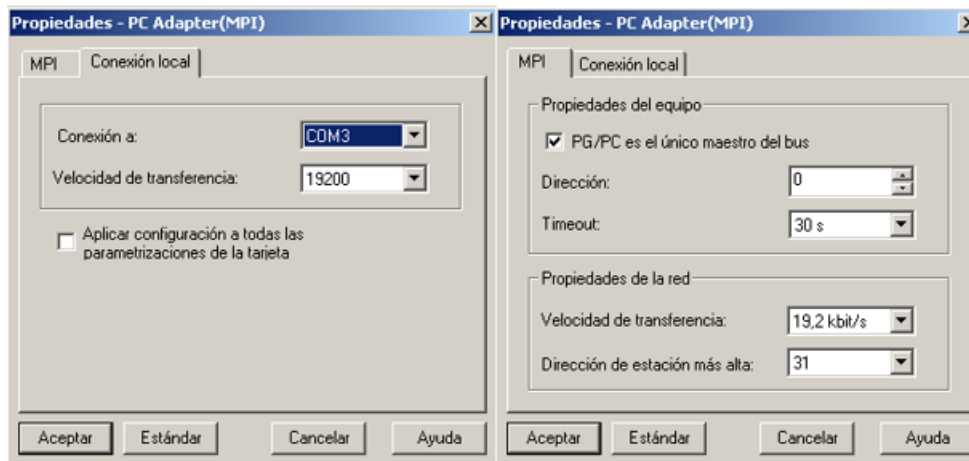


Figura 2.13: Configuración para ajustar los parámetros PG/PC

2.5.5 Carga y simulación

El último paso, antes de ver el sistema de automatización activo, es cargar en la CPU la configuración del hardware y el programa de usuario. Accediendo al comando **Sistema de destino >Cargar en PG** se puede acceder a todas las CPU disponibles. Pulsando **Mostrar**, aparecen todas las CPU accesibles en el Interfaz MPI. Una vez elegida la CPU, se debe pulsar **Aceptar** para cargar toda la configuración. Se establece la conexión entre el PC y la CPU mediante el **PG** y la CPU ha de estar en modo STOP o RUN-P.

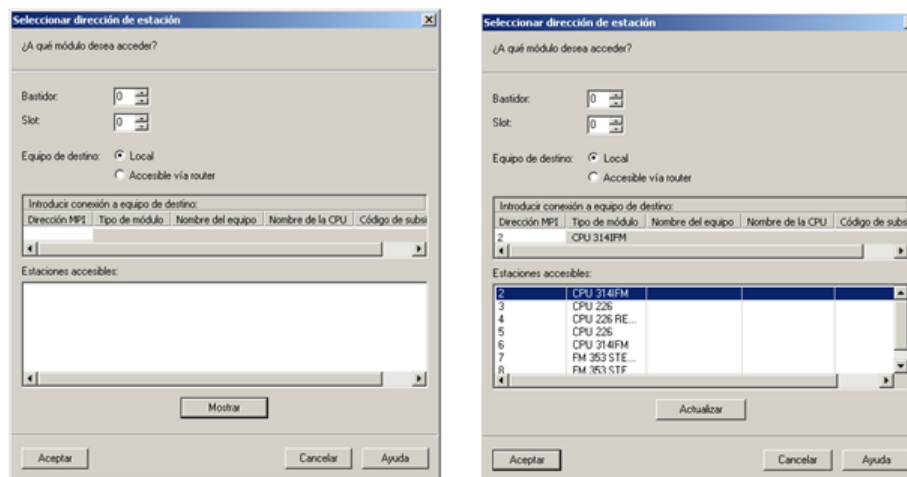


Figura 2.14: En la ventana *Cargar en PG* aparecen los dispositivos disponibles en la interfaz MPI

Por último, pasando el PLC a modo RUN se puede comprobar el funcionamiento de los bloques del programa desde la ventana principal del proyecto y accediendo a los bloques de programa como se muestra en la figura 2.5. Pulsando el icono *Observar si/no* se puede hacer un seguimiento del programa de usuario. Desde el *Editor de programas* es posible verificar el estado de cada variable accediendo a los comandos **Sistema de destino >Observar/forzar variable**. Se accede a la *tabla de variables* y el usuario puede introducir las variables a comprobar a tiempo real pulsando en el icono *Observar variable*.

2.5.6 Bloques del programa de usuario

La CPU se encarga de ejecutar el sistema operativo y el programa de usuario [10]. El sistema operativo organiza las funciones y los procesos que no están ligados a una tarea de control específica. El primero realiza las siguientes funciones:

- Gestión de re arranque del programa.
- Actualización de la imagen de proceso de las entradas y envía la imagen del proceso de las salidas.
- Llamada al programa de usuario.

- Detección y corrección de errores.
- Comunicación con unidades de programación y otras estaciones de comunicación.

El programa de usuario contiene las funciones que realiza el sistema de automatización. Se caracteriza por :

- Mostrar las condiciones de arranque de la CPU.
- Tratar los datos del proceso.
- Tratar las perturbaciones en el desarrollo del programa.

Se puede estructurar el programa en diferentes tipos de bloques[10]:

- **Bloques de organización (OB):** Define la estructura del programa de usuario.
- **Bloques de función del sistema (SFBs) y funciones del sistema (SFCs):** Son funciones integradas en la CPU que le permite acceder a funciones del sistema. Además, las funciones integradas están ligadas a un SFB.
- **Bloques de función (FB):** Son bloques con memoria que puede programar el usuario.
- **Funciones (FC):** Son los bloques que contienen las rutinas del programa más frecuentes.
- **Bloques de datos (DB):** Son áreas de datos que almacenan los datos de usuario.

En los siguientes apartados se hace una descripción más detallada de cada bloque de programa.

Bloques de organización

Es el encargado de interactuar entre el sistema operativo y el programa de usuario y controla [10]:

- El procesamiento cíclico.
- El procesamiento controlado por alarmas del programa.
- El comportamiento de arranque.
- Tratamiento de errores.

Es el que determina cuando se han de ejecutar las diferentes partes del programa de usuario. Pueden disponer de diferentes prioridades que permiten que un OB de mayor prioridad pueda interrumpir a otro OB de menor prioridad. En las CPUs-300 las prioridades están asignadas fijamente. En el programa de usuario de la estación *transfer* y el almacén se hacen servir los OB que se describen a continuación.

OB1 “Bloque de organización para la ejecución cíclica del programa”

OB1 es el bloque de organización que debe utilizar para que el sistema operativo ejecute un ciclo de programa [10]. Las características principales de OB1 son las siguientes:

- Tiene la prioridad más baja de todos los OB y estos pueden interrumpir su ejecución.
- El SO llama al OB1 cuando finaliza la ejecución de arranque o finaliza la ejecución del OB1 del ciclo anterior.
- Cuando finaliza la ejecución del OB1, el SO envía los datos globales. Y antes de que vuelva a arrancar OB1, el SO escribe en los módulos de salida la imagen de proceso de las salidas, actualiza la imagen de proceso de las entradas y recibe los datos globales para la CPU.
- Step7 supervisa el tiempo máximo de ciclo. En caso de que se supere, el SO avisa a OB80 (OB de error de tiempo). Si este último OB no está programado y se sobrepasa el tiempo máximo, la CPU pasa a modo STOP.

OB100 “Rearranque completo”

Antes de que la CPU pase a modo RUN y se realice la ejecución cíclica, el programa parte desde el estado de inicialización OB100. Mediante el código de usuario que se introduce en este bloque de programa, se permite realizar ajustes previos.

Bloques de función del sistema (SFBs) y funciones del sistema (SFCs)

La CPU S7 ofrece bloques programados que se pueden llamar desde el programa de usuario [10].

SFB son bloques de funciones “con memoria” integrados en la CPU S7. Són útiles para:

- Comunicar vía enlaces configurados.
- Funciones especiales integradas.

SFC es una función preprogramada integrada en la CPU. Se puede llamar desde el programa y no se carga como parte integrante del programa. Entre muchas posibilidades ofrecen:

- Transferencia de datos.
- Actualización de imágenes del proceso y tratamiento de campos de bits.
- Direccionamiento de módulos.
- Comunicación por datos globales.

Bloques de función

Son bloques programables “con memoria” que disponen de un bloque de datos de instancia donde se memorizan las variables estáticas. Las variables temporales se memorizan en la pila de datos locales y se pierden al concluir el tratamiento del FB.

Funciones (FC)

Son bloques de programa “sin memoria” que se ejecutan cada vez que son llamados por otro bloque. Se puede utilizar para:

- Devolver un valor de función al bloque invocante.
- Ejecutar una función tecnológica.

Las variables temporales se memorizan en la pila de datos locales y se pierden después de ser tratados. Siempre se han de indicar los parámetros actuales y no se pueden asignar valores iniciales. Al no tener memoria, es necesario proporcionar los parámetros formales. El acceso a parámetros FC se realiza a través de direcciones. El parámetro actual, o variable local, se almacena temporalmente en los datos locales del bloque que realiza la llamada.

Bloques de datos (DB)

Son bloques que contienen datos de usuario utilizables desde otros bloques y no contienen instrucciones. Cada FB, FC o OB pueden leer los datos o escribir en él. Además, los datos contenidos en el DB no se borran.

2.6 SFCs de comunicaciones

Es posible un intercambio de información entre un S7-300 y un S7-200/300 utilizando las funciones X_GET y X_PUT a través del interfaz MPI. En este proyecto, las funciones se realizan siempre desde la CPU 314 IFM de la estación *transfer*.

2.6.1 SFC 67 “X_GET”

La función X_GET se encarga de la lectura de datos de un S7-200/300 [11]. Los parámetros de entrada de la función son las siguientes:

- REQ: Cada vez que se produce un flanco positivo de la marca de entrada del S7-300, se inicia la lectura de los datos.
- CONT: Controlado mediante un BOOLEAN, determina si la comunicación sigue activa tras finalizar el servicio. Si CONT = 0, se desactiva el enlace tras finalizar la transferencia de datos. En caso contrario, el enlace se mantiene activo.
- DEST_ID: Se introduce la dirección MPI de la CPU donde se leen los datos.

2. HERRAMIENTAS UTILIZADAS

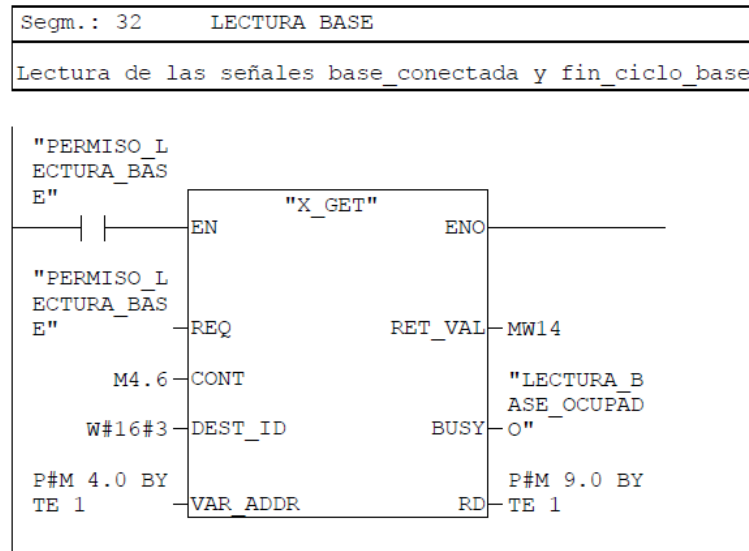


Figura 2.15: Visualización de la SFC 67 X_GET

- VAR_ADDR: Se introduce la referencia de la zona de la CPU que se debe leer. Con el siguiente ejemplo podríamos entender como referenciar el lugar y cuantos datos leer.

P# M4.0 BYTE 1

M4.0 indica la variable y el lugar de memoria donde está la información que se debe leer. BYTE 1 es el número de bytes que se necesitan leer.

Y los parámetros de salida son los siguientes:

- RET_VAL: En caso de que se haya realizado con éxito la transferencia de datos, este parámetro contendrá la longitud en Bytes de los datos copiados del buffer de recepción. En caso contrario, contendrá el código del fallo que se haya producido.

- BUSY: Muestra si ha finalizado el proceso de recepción. Si BUSY=1, no se ha terminado todavía el proceso de recepción. En caso contrario, sí.

- RD: Se introduce la referencia del buffer de recepción. Ha de ser tan largo como la de VAR_ADDR y han de coincidir el tipo variable. Los tipos de variable permitidos son BOOL, BYTE, WORD, DWORD.

Si la CPU pasa al estado *STOP*, se desconecta la comunicación X_GET. En cambio, los datos se leen aunque la CPU del interlocutor pase a *STOP*.

La información de los errores del parámetro RET_VAL están a disposición en las páginas 89-92 del manual [11].

2.6.2 SFC 68 "X_PUT"

La función X_PUT se encarga del envío de datos desde la CPU S7-300 al buffer de recepción de un S7-200/300. Los parámetros de entrada son los siguientes [11]:

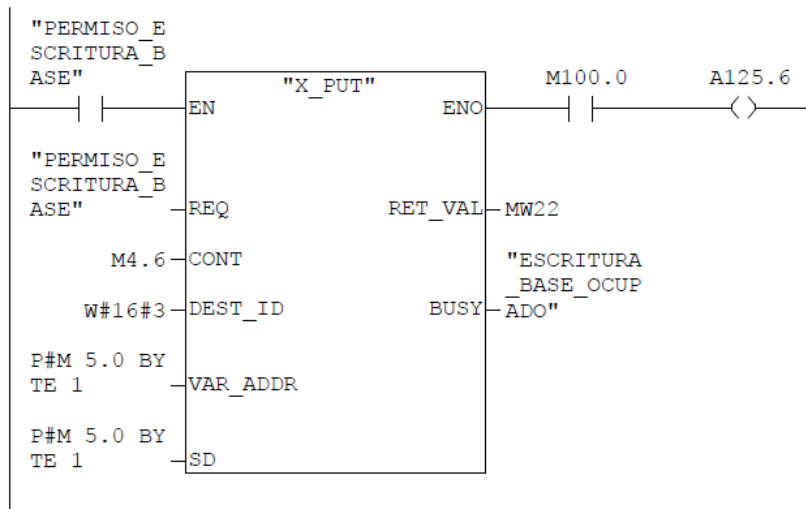


Figura 2.16: Visualización de la SFC 68 X_PUT

- REQ: Cada vez que se produce un flanco positivo de la marca de entrada del S7-300, se inicia la escritura de los datos.
- CONT: Controlado mediante un BOOLEAN, determina si la comunicación sigue activa tras finalizar el servicio. Si CONT = 0, el enlace se desactiva tras finalizar la transferencia de datos. En caso contrario, el enlace se mantiene activo.
- DEST_ID: Se introduce la dirección MPI de la CPU donde se realiza el envío.
- VAR_ADDR: Se introduce la referencia de la zona de la CPU que se debe leer.

Y los parámetros de salida son los siguientes:

- SD: Se introduce la referencia del buffer de recepción. Ha de ser tan largo como la de VAR_ADDR y han de coincidir el tipo variable. Los tipos de variable permitidos son BOOL, BYTE, WORD, DWORD..
- RET_VAL: En caso de que se haya realizado con éxito la transferencia de datos, este parámetro contendrá la longitud en Bytes de los datos copiados del buffer de recepción. En caso contrario, contendrá el código del fallo que se haya producido.
- BUSY: Se mostrará en una marca, si ya se ha terminado el proceso de recepción. Si BUSY=1, no se ha terminado todavía el proceso de recepción. En caso contrario, sí.

Si la CPU pasa al estado STOP, se interrumpe el enlace establecido por la función. En cambio, si la CPU del interlocutor pasa a STOP, los datos emitidos se escriben igualmente.

La información de los errores del parámetro RET_VAL están a disposición en las páginas 89-92 del manual [11].

2.7 Estaciones de proceso

En la introducción de este proyecto se ha expuesto que la EPS dispone de cinco estaciones de procesos en el que se lleva a cabo el montaje de una pieza y su posterior almacenamiento. Ha sido desarrollado por la empresa SMC Internacional Training y tiene el objetivo de simular a pequeña escala un proceso de automatización. En este apartado se realiza una breve explicación sobre el proceso que realiza cada una de las estaciones.

2.7.1 Estación de bases

Es la estación que alimenta el primer elemento de los 3 que completan la pieza final. La finalidad de la estación base es introducir una base encima del palé que facilita la estación *transfer* y está compuesta por 5 módulos:

- Alimentador de la base: Es la zona donde se sitúan las bases que alimentan el proceso. Cada vez que se alimenta una pieza, un cilindro neumático retira la de más abajo de la columna de piezas. Un detector magnético comprueba cuando el cilindro ha llegado a su final de carrera.
- Verificación de la posición: Un cilindro compuesto por una pieza cilíndrica comprueba que la base está colocada correctamente. Un detector magnético comprueba que el cilindro ha llegado a su final de carrera, confirmando la validez de la pieza.
- Desplazamiento al punto de traslado: Un tercer cilindro desplaza la base hasta el lugar donde será recogida para realizar el traslado al palé o desecharla. También se dispone de un detector magnético para confirmar que el cilindro ha llegado a su final de carrera.
- Rechazo base incorrecta: Si el cilindro neumático del verificador de posición detectó que la pieza no estaba colocada correctamente, un cilindro de simple efecto impulsa la pieza por una rampa.
- Inserción de la base en el palet: Si la pieza está introducida correctamente, una plataforma compuesta por 4 ventosas y dos ejes de movimiento, uno para el eje horizontal y otro para el eje vertical, transportan la base hasta el palé.

La estación dispone de una botonera con 3 pulsadores, 2 conmutadores y un botón de emergencia. También dispone de una luminaria con el objetivo de encenderse si falta material.

2.7.2 Estación de rodamientos

La segunda estación aporta un rodamiento a la pieza final y está compuesta por 4 módulos:

- Alimentación del rodamiento: Los rodamientos están almacenados en columna esperando a que un cilindro de doble efecto vaya alimentando el rodamiento de

más abajo en cada ciclo de operación. Cuando el cilindro ha realizado el final de carrera, hay un sensor que se encarga de confirmar que se ha alimentado la pieza.

- Envío a la estación de medida: Una vez confirmado la presencia de rodamiento, un actuador de giro tipo piñón-cremallera, que describe un ángulo de 180° y con una pinza en su base, se encarga de desplazar el rodamiento desde la posición alimentada hasta la base de un elevador.

- Medición de la altura: Cuando el rodamiento está situado en la plataforma del elevador, este asciende hasta contactar con un palpador, potenciómetro lineal, para calcular la altura del rodamiento. Este palpador está compuesto por un cilindro sin vástago y una vez se ha elevado el máximo se activa un detector magnético. Cuando el elevador ha descendido y el rodamiento no tiene la altura indicada por el usuario, un cilindro de doble efecto expulsa la pieza.

- Traslado del rodamiento: Por último, si el rodamiento tiene la altura indicada, una unidad compuesta por un cilindro compacto de movimiento lineal y rotativo, y un brazo de sujeción compuesto por una pinza neumática, se encarga de desplazar la pieza desde la plataforma del elevador hasta el palé.

La estación dispone de una botonera con 3 pulsadores, 2 conmutadores y un botón de emergencia. También dispone de una luminaria con el objetivo de encenderse si falta material.

2.7.3 Estación de tapas

Esta estación tiene la finalidad de facilitar una tapa al palé donde ya hay introducido una base y un rodamiento. Se dispone de 6 combinaciones de tapas. Las hay de aluminio, nylon blanco o negro y de dos alturas distintas. La estación de tapas está compuesta por seis módulos:

- Módulo de alimentación: Las tapas están almacenadas en columna esperando a que un cilindro de doble efecto vaya alimentando la tapa de más abajo en cada ciclo de operación. Cuando el cilindro ha realizado el final de carrera, hay un sensor que se encarga de confirmar que se ha alimentado la pieza.

- Estación de carga: Un manipulador, compuesto por un cilindro compacto de movimiento lineal y rotativo con un brazo de sujeción mediante una pinza, se encarga, mediante la elevación y giro de 180° del manipulador, de desplazar la pieza desde el alimentador hasta un plato divisor.

- Plato divisor: Dos cilindros de tope fijo y móvil y un cilindro empujador que realiza el giro se encargan de mover un plato divisor de ocho posiciones donde se introducen las tapas.

- Estaciones de detección de material y medición: Para diferenciar el tipo de material, color y altura de las tapas, la estación dispone de varios detectores que están situados en diferentes posiciones cercanas al plato divisor. Se dispone de

2. HERRAMIENTAS UTILIZADAS

un detector inductivo, un detector capacitivo y un fotoeléctrico. Para la detección de altura, se utiliza un encoder lineal integrado en un cilindro de doble efecto que se posiciona encima de la pieza.

- Evacuación de tapa incorrecta: Cuando se ha de rechazar una tapa del proceso, se dispone de un manipulador de dos cilindros de vástagos paralelos de doble efecto, eje horizontal y eje vertical, con un brazo de sujeción que dispone de tres ventosas. Este recoge la tapa del plato divisor y la desecha sobre una rampa.

- Inserción de tapa: Para insertar la tapa desde el plato divisor hasta el palé, se dispone de un manipulador formado por un cilindro compacto de movimiento lineal y rotativo con un brazo de sujeción con pinzas y realiza el movimiento de traslado de la pieza.

La estación dispone de una botonera con 3 pulsadores, 2 conmutadores y un botón de emergencia. También dispone de una luminaria con el objetivo de encenderse si falta material.

2.7.4 Estación de almacenaje

Cuando el proceso de montaje de la pieza ha sido completado, solo queda retirarla de la cinta transportadora. La estación de almacenamiento dispone de una superficie de aluminio con espacio para depositar hasta treinta piezas. El actuador que recoge la pieza del palé, y lo almacena, dispone de dos ejes controlados por posición que permite el movimiento en cualquier punto de la superficie, y un eje vertical para levantar y depositar la pieza.

- Ejes posicionadores: La estación dispone de dos actuadores lineales motorizados con el objetivo de llevar la pieza desde el punto de recogida hasta una plataforma. Estos actuadores disponen de servomotores de corriente alterna, con un encoder absoluto que realiza una lectura instantánea del movimiento. Este se realiza mediante el sistema mecánico de tuerca-husillo. El movimiento de giro realizado por el motor se convierte en desplazamiento lineal de los dos ejes.

- Eje vertical: Para la recogida de la pieza, El eje actuador dispone de un cilindro de vástagos paralelos y de cuatro ventosas en su mano. Además, la estación dispone de un sensor de presión de señal digital que indica el nivel de vacío para fijar la pieza.

La estación dispone de una botonera con 3 pulsadores, 2 conmutadores y un botón de emergencia. También dispone de una luminaria con el objetivo de encenderse si falta material.

2.7.5 Estación *transfer*

Es el encargado de transportar los palé hacia las estaciones donde se transfieren los elementos que forman la pieza hasta, finalmente, llegar a la estación de almacenamiento donde se recoge la pieza completa. La estación está compuesta por las siguientes partes:

- Carriles de desplazamiento: La estación *transfer* está formado por una plataforma con dos cintas transportadoras. El primero se sitúa delante de las estaciones y el segundo carril es el de regreso de los palés, una vez las piezas hayan sido almacenadas, para volver a iniciar de nuevo el proceso.

- Topes retenedores y actuadores: Detienen el palé delante de cada estación y un sensor de posición que detecta su presencia. Al haber cuatro estaciones que comunican con la cinta transportadora, hay 4 topes retenedores y cuatro sensores de presencia.

- Elevador de palé: Hay una plataforma que eleva el palé, delante de la estación de rodamientos, para que el manipulador que transfiere el rodamiento pueda introducirlo correctamente.

- Tramo de transferencia de palet: Cada vez que un palé llega al final del carril de la cinta transportadora, un sensor capacitivo detecta la presencia del palé y un cilindro sin vástago de doble efecto se encarga de desplazarlo a la siguiente cinta.

Además, la estación dispone de una botonera con 2 pulsadores y 1 conmutador.

La estación *transfer* es la que comunica con las 4 estaciones. Por tanto, hay 4 lugares donde se retiene un palé para que cualquiera de las tres primeras estaciones pueda introducir una pieza al palé y la última, el almacén, que recoge la pieza. Para explicar con detalle el programa de usuario de la estación *transfer*, en el siguiente capítulo se diferencia cada tramo con los siguientes nombres:

- Transfer-Base
- Transfer-Eje
- Transfer-Tapa
- Transfer-Almacén

DESARROLLO DE LA ESTACIÓN *transfer*

La estación *transfer* es la que comunica con las demás estaciones para que se produzca una sincronización durante el proceso de automatización. Desde la CPU 314 IFM, con las funciones de sistema del programa de usuario que se detalla en este apartado y el interfaz MPI, es posible la comunicación con las CPU 226 de las tres primeras estaciones y la CPU 314 IFM de la estación de almacenamiento.

El programa de usuario ha sido creado desde el SIMATIC Step7 y en este capítulo se analizan los diferentes bloques de programa que se cargan en la CPU. Al final de este capítulo, se detalla la guía GEMMA que se ha utilizado en el *transfer*.

3.1 Funcionamiento de la estación *transfer*

Una vez se ha cargado el programa de usuario y la CPU pasa a modo RUN, a través de las funciones de programa X_PUT y X_GET, el *transfer* puede comunicar a las demás estaciones de su actividad y cuando tienen la oportunidad de hacer el envío de una pieza desde la estación de procesos hasta un palé. Por otra parte, cuando cualquiera de las demás estaciones pasan a modo RUN, también notifican al *transfer* de su estado y cuando están preparados para introducir un elemento a un palé.

Al principio del proceso, la cinta transportadora se pone en funcionamiento hasta que el primer palé llega hasta el tope más cercano a la estación de bases. En el instante que la estación bases comunica al *transfer* que está activo, este último pasa de la etapa de inicialización a la etapa en que espera a recibir el elemento. Cuando el palé dispone de una base y la estación de bases ha notificado la introducción de una base en el palé, se baja el tope que retiene el palé, se pone en marcha la cinta y el palé avanza hasta la segunda zona donde espera recibir un rodamiento. Por otra parte, un segundo palé es retenido delante de la estación de bases.

La estación de rodamientos es la estación que aporta el segundo elemento al palé. El proceso es similar al anterior. Hay una etapa de verificación antes de comenzar con el envío y otro de confirmación de que se ha realizado. El *transfer* dispone de una plataforma para levantar el palé durante la recepción del rodamiento.

Cuando el palé dispone de una base y un rodamiento, espera la confirmación de que las demás partes del *transfer* estén libres. Se baja el tope que retiene el palé y cuando se activa la cinta, el palé pase a la siguiente estación. El tope siempre se levanta siete segundos después de que se haya activado la cinta transportadora, es el tiempo necesario para que el palé haya sobrepasado el tope. La cinta se detendrá cuando las tres primeras estaciones dispongan de palé.

La tercera parada es en la estación de tapas. El proceso es similar a las 2 primeras y la estación se encarga de introducir el último elemento que completa la pieza. Cuando las tres estaciones han realizado los envíos de los elementos que procesan, se vuelve a activar la cinta y es cuando ya se puede comprobar un proceso normal en el que se espera la llegada de palé a cada estación.

A partir de este momento, la pieza completada llega hasta la estación de almacenamiento donde es recogida y almacenada. La estación *transfer* es un circuito cerrado. Una vez el palé haya pasado por la estación de almacenamiento, retornará hasta la estación de bases para volver a repetir de nuevo el proceso.

3.2 Programa de usuario

En este apartado se detallan los bloques de programa que se han hecho servir. La estructura es la siguiente:

Bloques de organización

- OB1
- OB100

Funciones

- FC1 “Etapas”
- FC2 “Salidas”
- FC3 “Marcas”
- FC4 “Comunicación”
- FC5 “Temporizadores”
- FC6 “AS-i”
- FC9 “Guía Gemma”

Funciones del sistema

- SFC 67 “X_GET”
- SFC 68 “X_PUT”
- SFC 79 “SET”
- SFC 80 “RSET”

Datos globales

- DB1 “E_ASI”
- DB2 “ASI_A”

3.2.1 Bloques de organización (OB1 y OB100)

OB 100 es el modo de rearranque completo del programa. Cuando la CPU pasa a modo RUN, el programa comienza por este bloque de inicialización. Se activan las etapas (etapa_0, etapa_10, etapa_20 y etapa_30) de las cuatro partes que dividen el programa, uno para cada estación que comunicará con el *transfer*. Además, se inicializa el contador que verifican los palés que participan en el proceso.

OB 1 llamará en cada ciclo de programa a las seis FC que contiene el programa de usuario.

3.2.2 Datos globales

El programa de usuario dispone de dos bloques de datos globales. En “E_ASI”, aparecen las señales de los sensores y la posición de los actuadores de los escalvos AS-i. Cada vez que se menciona una variable en el programa de usuario, se ha de añadir el prefijo “E_ASI”. Por ejemplo, para escribir la variable del retenedor de la estación de bases, se debe introducir “E_ASI”.ppA. En este programa, todas las variables son de tipo booleano y su valor inicial es “FALSE”. En el cuadro 3.1 se muestran las variables de entrada que reúnen todos los esclavos AS-i y su dirección.

En el bloque de programa DB2, “A_ASI”, se muestran el direccionamiento de las variables de los actuadores. Cada vez que se llama a alguna variable de este bloque, hay que introducir el prefijo “A_ASI”. Por ejemplo, si se quiere elevar la plataforma que facilita la introducción del rodamiento en el palé, se debe escribir “A_ASI”.I.

3.2.3 FC3 Marcas

Antes de analizar las funciones FC1 y FC2, hay una serie de marcas que se deben comentar para entender el funcionamiento del *transfer*.

Estación_Libre

3. DESARROLLO DE LA ESTACIÓN *transfer*

Dirección	Nombre	Dirección	Nombre
0.0	dp2	2.4	dp1
0.1	n1	2.5	m0
0.2	n23	2.6	m_1
1.0	subi	4.0	ppE
1.1	i	4.1	b0E
1.2	ppD	4.2	b1E
1.3	b0D	4.3	b2E
1.4	ppA0	8.4	ppH
1.5	b0A	8.5	b0H
1.6	b1A	8.6	b1H
1.7	b2A	8.7	b2H

Cuadro 3.1: Direccionamiento de las variables de entrada AS-i

Dirección	Nombre	Dirección	Nombre
0.0	n_avance	2.1	I
0.1	n_retroceso	2.4	m avance
1.0	subi	2.5	m retroceso
1.4	a	4.0	e
2.0	D	8.4	h

Cuadro 3.2: Direccionamiento de las variables de salida de los actuadores

Hay cuatro marcas, como se muestra en el cuadro 3.3, que se dividen para cada estación que comunica con el *transfer* y son necesarias para activar la cinta transportadora. Estas marcas se activan en dos momentos determinados del proceso.

- En la etapa de inicialización, cuando todavía no ha llegado el primer palé en el proceso.
- Desde la etapa donde se confirma que el palé ha recibido un elemento hasta que llega el siguiente palé.

Dirección	Nombre
M 2.0	BASE_ LIBRE
M 2.1	EJE_ LIBRE
M 2.2	TAPA_ LIBRE
M 2.3	ALMACEN_ LIBRE

Cuadro 3.3: Estación_ Libre

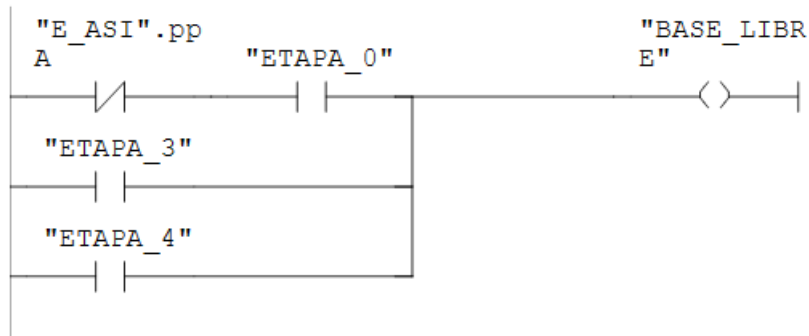


Figura 3.1: Condiciones para que se active *BASE_LIBRE*

Inicio_Ciclo_Estación

Cuando alguna zona del *transfer* recibe un palé, se realiza un SET de la marca *INICIO_CICLO_ESTACIÓN* para comunicar a la estación adyacente que puede realizar el envío de un elemento al palé. Una vez se ha notificado al *transfer* que se ha completado el traslado, se resetea la marca para comunicar de que no se debe hacer otro envío de momento.

transfer_on_Estación

Es otra marca imprescindible para la comunicación entre estaciones. Cuando se ha cargado el programa de la estación *transfer* a la CPU a través del SIMATIC S7, este pasa a RUN. Estas marcas son enviadas a sus respectivas estaciones para comunicar que la estación *transfer* está activa.

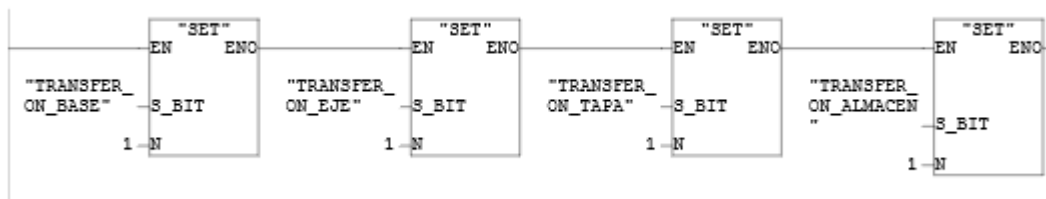


Figura 3.2: La SFC SET activa las marcas de *transfer_on_Estación*

3.2.4 FC2 Salidas

En esta función se detallan las variables de salida que tiene a disposición el *transfer*. Unas salidas pertenecen a los actuadores de los esclavos AS-i y otras se utilizan para poner en funcionamiento el motor de la cinta transportadora, la célula en servicio y el piloto luminoso de defecto.

En cuanto a las salidas que ofrecen los esclavos AS-i, hay tres tipos que facilitan el funcionamiento del proceso:

- Bajada del retenedor del palé: Se hace un set de la salida antes de activar la cinta transportadora para el desplazamiento de los palés.
- Plataforma que eleva el palé para introducir correctamente el rodamiento dentro de la base que ya dispone el palé.
- Avance y retroceso del cilindro de traslado de los palés: Cuando un palé llega al tramo final de una cinta, un sensor detecta su presencia y un cilindro lo desplaza al inicio de la siguiente cinta. Una vez desplazado, el cilindro retrocede.

La estación dispone, también, de las siguientes salidas:

- Al pasar a *RUN*, se activa la dirección que pone la estación en servicio.
- Como se detalla en la sección 3.2.9, *MOT* activa la cinta transportadora.
- *SI* activa el convertidor de frecuencia

Es necesario que los retenedores bajen antes de se active la cinta transportadora para trasladar los palés hasta la siguiente estación. También es importante que la cinta se active cuando todos los palé hayan recibido, anteriormente, un elemento o cuando alguna estación todavía permanece en la etapa de inicialización. Se ha creado una marca, *ESTACIONES_LIBRES*, que condiciona la bajada de los retenedores cuando todas las estaciones cumplan las condiciones comentadas.

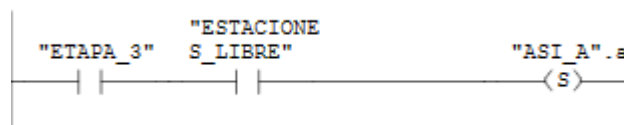


Figura 3.3: El retenedor bajará cuando se haya producido el traslado de una base y las demás estaciones estén libres

3.2.5 FC1 Entradas

En esta función se desarrollan las etapas que hacen posible el proceso de automatización del *transfer*. Las etapas están divididas entre las cuatro zonas donde el *transfer* comunica con otra estación y los graficets están incluidos en el anexo A.2.

- *transfer-bases*: Etapa_ 0 - Etapa_ 6.
- *transfer-rodamientos*: Etapa_ 10 - Etapa_ 16
- *transfer-tapas*: Etapa_ 20 - Etapa_ 26
- *transfer-almacén*: Etapa_ 30 - Etapa_ 36

Las etapas X0 son las de inicialización de cada zona. Se suben los topes retenedores de los palés. Se resetea la marca “INICIO _ CICLO_ ESTACIÓN” y se activa la marca “Estación _ Libre”. Las condiciones para pasar a la siguiente etapa es que la estación adyacente esté conectada y que un sensor detecte la presencia de un palé.

En las etapas X1, el *transfer* comunica a las demás estaciones de la disposición de palé para realizar el envío de un elemento. Cuando le comunican que se procede a transferir un elemento al palé, se pasa a la siguiente etapa.

En las etapas X2, el *transfer* espera hasta que la estación adyacente avise de que ha completado con éxito la introducción del elemento.

En las etapas X3, se bajan los topes retenedores de los palés y el *transfer* comunica que no está disponible para recibir más elementos. El *transfer* se asegura de que todas las zonas están en la etapa inicial o en la etapa X3. Una vez corroborado, se puede pasar a la siguiente etapa.

En las etapas X4, la cinta está en movimiento. Para pasar a la siguiente etapa han de saltar dos temporizadores. Esto es debido a que el palé necesita 7 segundos para pasar desde su estado de reposo hasta superar la posición donde está situado el retenedor. Los temporizadores tienen 5 segundos como tiempo máximo. Empieza a contar primero el temporizador t_ 4. Una vez activo, empieza a contar t_ 5. Cuando este último se activa, el retenedor sube para que detenga el siguiente palé.

En las etapas X5, suben los retenedores. Todas las zonas del *transfer* están a la espera de recibir el siguiente palé. La cinta se detiene cuando todos los palés que parten desde una zona llega hasta el tope de la siguiente, a excepción del palé que estaba en la estación de almacenamiento ya que va a tener bastante recorrido hasta llegar de nuevo a la altura de la estación de bases.

En las etapas X6, se produce un tiempo de espera hasta pasar a las etapas de comienzo de ciclo de cada zona.

3.2.6 FC4 Comunicación

En el apartado 23 se describe el funcionamiento de “X_GET” y “X_PUT”. La CPU del *transfer* tiene reservado varios bytes para almacenar las marcas que lee del resto de PLCs. Por otra parte, también hay un espacio de memoria reservado para los datos de las marcas que serán enviados. Para sincronizar la estación *transfer* con el resto de estaciones, se reservan 8 bytes de memoria. 4 bytes para datos de entrada y 4 para datos de salida.

X_GET

Los siguientes bytes están reservados para la recepción de las variables que se obtienen de las diferentes estaciones.

- El byte 9 para las variables de la estación de bases.
- El byte 10 para las variables de la estación de ejes.
- El byte 11 para las variables de la estación de tapas.
- El byte 12 para las variables de la estación de almacenamiento.

Dirección	Nombre
M 9.0	BASE_ CONECTADA
M 9.1	FIN_ CICLO_ BASE
M 10.0	EJE_ CONECTADA
M 10.1	FIN_ CICLO_ EJE
M 11.0	TAPA_ CONECTADA
M 11.1	FIN_ CICLO_ TAPA
M 12.0	ALMACEN_ CONECTADA
M 12.1	FIN_ CICLO_ ALMACEN

Cuadro 3.4: X_GET

En la figura 3.4 se observa un ejemplo de llamada al SFC 67 “X_GET”. Se trata del momento en que la CPU del *transfer* recibe el valor de dos marcas. **BASE_ CONECTADA** es la variable que, al estar activa, confirma que la CPU de la estación de bases está en modo ON. Y cada vez que la estación de bases tenga una base disponible para introducir en el palé, se activa **FIN_ CICLO_ BASE**.

Siguiendo en el mismo ejemplo. Cada vez que se produce un flanco de subida de de la marca **PERMISO_ LECTURA_ BASE**, la función lee el valor de las dos marcas anteriormente mencionadas. En *DEST_ID* se ha introducido la dirección MPI a la que pertenece la estación de bases. Y en *VAR_ADDR* se introduce la dirección a partir de donde se leen las variables. En este caso, se lee el byte 4 del registro de memoria de la estación de bases. Por otra parte, en la salida *RD* se ha introducido la dirección 9.0, que es donde se van a almacenar el valor de las marcas leídas.

3. DESARROLLO DE LA ESTACIÓN *transfer*

Dirección	Nombre
M 5.0	TRANSFER_ON_BASE
M 5.1	INICIO_CICLO_BASE
M 5.2	STOP_TRANSFER_BASE
M 6.0	TRANSFER_ON_RODAMIENTO
M 6.1	INICIO_CICLO_RODAMIENTO
M 6.2	STOP_TRANSFER_RODAMIENTO
M 7.0	TRANSFER_ON_TAPA
M 7.1	INICIO_CICLO_TAPA
M 7.2	STOP_TRANSFER_TAPA
M 8.0	TRANSFER_ON_ALMACEN
M 8.1	INICIO_CICLO_ALMACEN
M 8.2	STOP_TRANSFER_ALMACEN

Cuadro 3.5: Las marcas que el programa de usuario envía a las diferentes estaciones

- **INICIO_CICLO_BASE**: Confirma la recepción de un palé y que se puede realizar la introducción de una base.

- **STOP_TRANSFER_BASE**: Verifica si el usuario ha hecho una parada del proceso de la estación *transfer*.

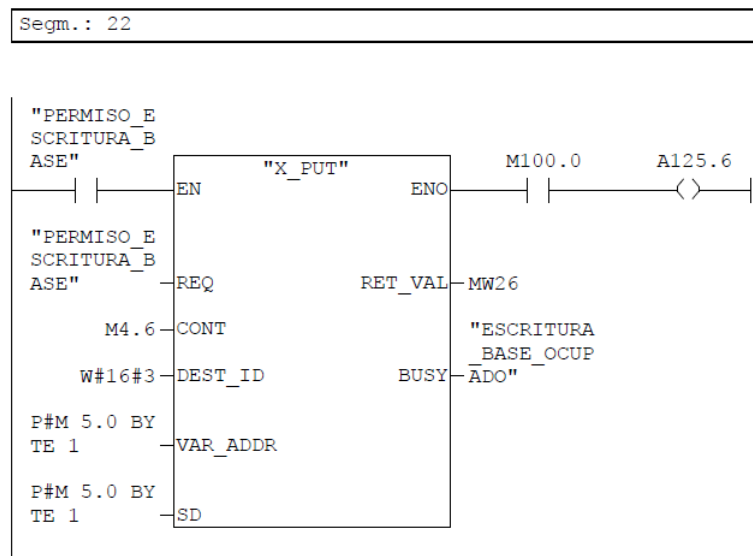


Figura 3.5: Envío de variables a la estación de bases

Este ejemplo también es útil para comprender la comunicación con el resto de estaciones.

3.2.7 FC6 AS-i

Esta función, escrita en lista de instrucciones (AWL), contiene el código necesario para la lectura y escritura de las entradas y salidas digitales del bus AS-i. Durante la configuración Hardware del PLC se determinó un margen de 16 bytes que ocuparía el maestro AS-i para un posible uso de 31 esclavos.

El acceso a las direcciones de los esclavos AS-i se produce mediante las instrucciones de carga y transferencia. La llamada se realiza en forma de W o DW. Como se muestra en la figura 3.6, se carga el DW de entradas de periferia correspondiente a una dirección y se transfiere a un DW del bloque de datos globales DB 1 (E_ASI). De esta forma, el usuario puede gestionar el valor de los sensores de los esclavos para aplicarlo en el programa de usuario.

Para enviar el valor de salida a los actuadores AS-i se ha creado otro segmento de programación, como se muestra en la figura 3.7, para cargar los valores correspondientes al bloque de datos globales DB 2 y transferirlos a las direcciones de los esclavos.

Segm.: 1			
L	PED	256	
T	DB1.DBD		0
L	PED	260	
T	DB1.DBD		4
L	PED	264	
T	DB1.DBD		8
L	PED	268	
T	DB1.DBD		12

Figura 3.6: Carga y transferencia de los valores a DB1

Segm.: 2			
L	DB2.DBD		0
T	PAD	256	
L	DB2.DBD		4
T	PAD	260	
L	DB2.DBD		8
T	PAD	264	
L	DB2.DBD		12
T	PAD	268	

Figura 3.7: Carga de los valores de DB2 a los actuadores AS_I

3.2.8 FC7 Guía GEMMA de la estación *transfer*

GEMMA (Guía de estudio de los modos de marcha y parada) es un método de organización de estados que se deben llevar a cabo en un proceso de automatización (funcionamiento, fallo, parada de emergencia, arranque o final de ciclo) [12]. Es una plantilla cerrada donde el usuario muestra las rutas que debe seguir el programa para un correcto funcionamiento del proceso y priorizando la seguridad de los operarios que van a trabajar cerca de la estación.

En el anexo A.1.2 está a disposición la plantilla de la guía GEMMA y su GRAFCET. Los estados utilizados para el programa de usuario son los siguientes:

- *Parada en el estado inicial (A1)*: Es el primer estado del que parte el programa. Se confirma al resto de estaciones que el *transfer* está activo y se habilitan las marcas **STOP_TRANSFER_ESTACIÓN** para que no inicien todavía sus procesos. Cuando el usuario pulse *MARCHA*, se pasará al siguiente estado. También se puede llegar a A1 desde A5 después de haber realizado una parada de emergencia. En este último caso, es conveniente reinicializar el resto de estaciones.
- *Producción normal (F1)*: El programa de usuario va ejecutando las etapas de procesos que se han mencionado en el apartado 3.2.5. En este estado se activa la marca **I_M**. Por el contrario, se resetan las marcas **STOP_TRANSFER_ESTACIÓN** para que el resto de estaciones de proceso funcionen. Se saldrá de F1 cuando se produzca una parada pedida o se realice una parada de emergencia.
- *Parada de emergencia (D1)*: El usuario puede detener la estación, debido a un estado estado de emergencia, pulsando *PARADA* durante 1.5 segundos. El operario deberá verificar que no haya ningún elemento que pueda ocasionar ningún daño antes de tratar el defecto que ha provocado la parada de emergencia. Pulsando *MARCHA*, se pasará al siguiente estado para corregir la anomalía.
- *Tratamiento de defectos (D2)*: El operario puede solucionar cualquier anomalía que se haya producido. Una vez solucionado, el operario deberá pulsar *MARCHA*.
- *Preparación posterior al defecto (A5)*: Se debe preparar el *transfer* para volver a su funcionamiento normal. El operario deberá retirar los palés que estaban participando en el proceso ya que se volverá a inicializar el proceso. Después, se deberá pulsar *MARCHA* para pasar al siguiente estado.
- *Puesta en el estado inicial (A6)*: Antes de pasar al estado A6, se recomienda volver a reiniciar las estaciones que comunican con el *transfer* ya que pueden estar detenidos en una etapa donde ya se está realizando la transferencia de un elemento. Pulsando *MARCHA*, se pasará al siguiente estado.
- *Parada pedida (A3)*: En cualquier instante, el operario puede parar el proceso pulsando *PARADA* durante menos de 1.5 segundos. también se pararán los procesos del resto de las estaciones.
- *Parada obtenida (A4)*: La estación permanecerá detenida hasta que el usuario pulse *MARCHA*. Se retornará a *Funcionamiento normal*.

3.2.9 Arranque y parada de la cinta transportadora

¿Cuándo se activa la cinta transportadora? Como se ha expuesto en el apartado 3.2.4, la cinta transportadora se activa con la salida “MOT”. Como se puede comprobar en la imagen 3.8, la cinta transportadora tiene las siguientes condiciones para poner en marcha la cinta:

- El *transfer* tiene que permanecer en el estado *Producción normal* de la guía GEMMA. En caso contrario, la cinta no se activará.
- Cuando todas las estaciones estén a la espera de recibir un palé o estén preparadas para recibir el siguiente. Como se ha expuesto en el apartado 4.3, las marcas *ESTACIÓN_ LIBRE* se activan en las etapas de inicialización, cuando una estación ha introducido un elemento en un palé y cuando se han bajado los topes retenedores.
- Sí el operario detiene la cinta transportadora nada más iniciar el desplazamiento y no han pasado más de siete segundos. Las etapas X4 seguirán activas, manteniendo activas las marcas *ESTACIÓN_ LIBRE*, y los palés podrán desplazarse hasta la siguiente estación cuando el operario pulse *MARCHA*.
- Sí el operario detiene la cinta pasados siete segundos después de haberse activado la cinta transportadora y los palés no han llegado a la siguiente estación, los topes retenedores habrán subido y las etapas activas serán las X5. Cuando el operario pulse *MARCHA*, se volverá a activar la cinta.

La cinta transportadora se desactiva en los siguientes casos:

- Cuando la estación no está en el estado *Producción normal* de la guía GEMMA, **I_M** permanece desactivada e impide la activación de la cinta transportadora.
- Cuando los sensores que contiene los retenedores detectan los palés.
- ¿Que sucede al principio del proceso para la espera del primer palé? Al principio, solo participa un palé. La cinta transportadora se ha de detener cuando este llega hasta el primer retenedor. Lo mismo que sucede cuando llega el segundo y el tercer palé. Se ha introducido un contador para detener la cinta transportadora durante los tres primeros casos.

3. DESARROLLO DE LA ESTACIÓN *transfer*

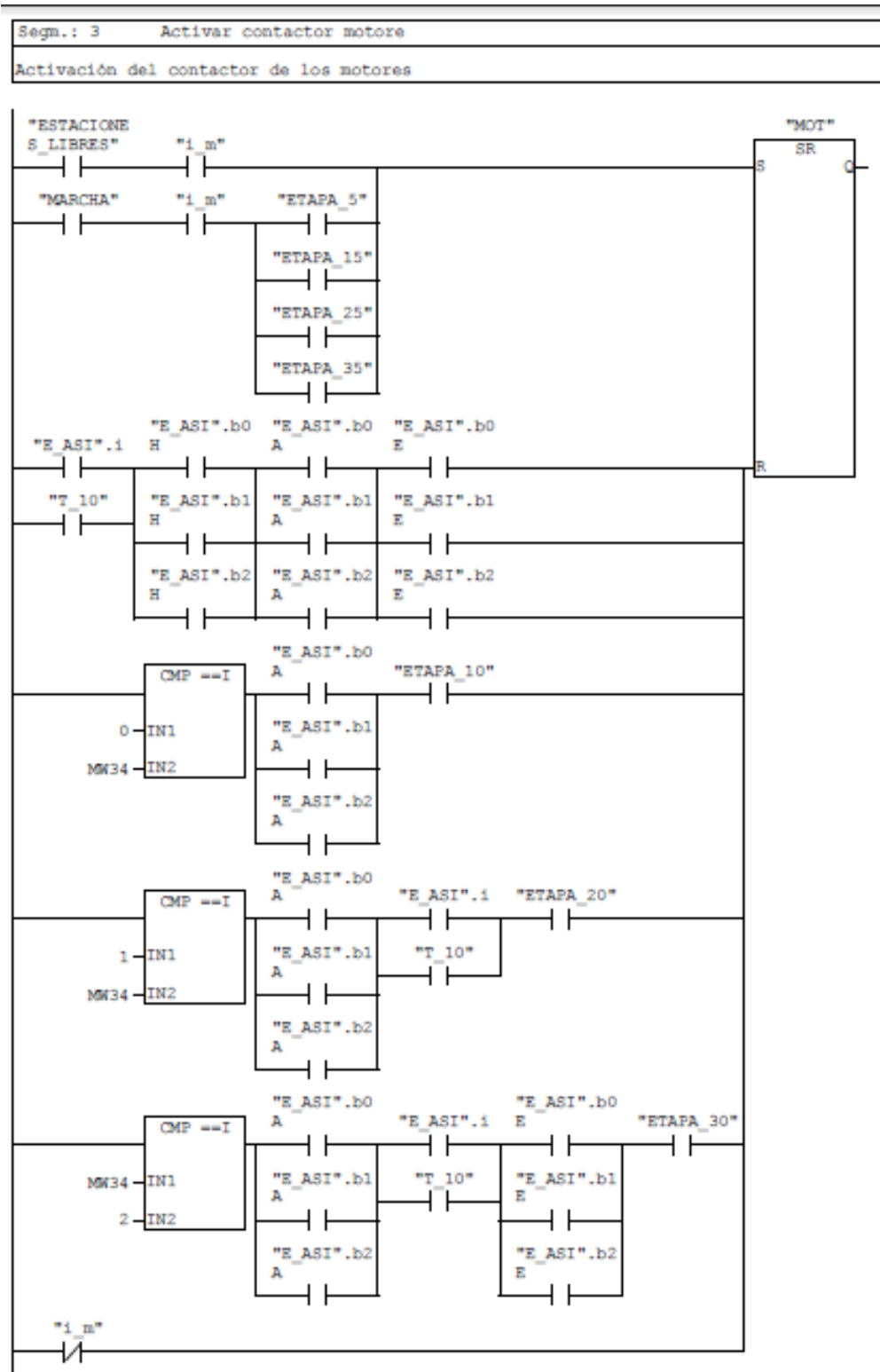


Figura 3.8: Condiciones para activar/desactivar *MOT*

DESARROLLO DE LAS ESTACIONES DE ALIMENTACIÓN

Para la selección de los elementos que conforman la pieza, que finalmente acaba almacenada, en este proyecto se han programado la estación de bases, la estación de rodamientos y la estación de tapas. Se ha prestado especial interés en las marcas que comunican con el *transfer*. Además, se ha aplicado el protocolo de la guía GEMMA. En los apartados de este capítulo se repasa el funcionamiento de cada estación, las variables que son enviadas y recibidas por parte del *transfer*.

4.1 Sincronización con la estación *transfer*

Cada estación de procesos tiene reservados 2 bytes, uno para el envío y otro para la recepción de datos con la estación *transfer*. Avisan a este último cuando pasan a RUN y cuando tienen un elemento la espera de introducir en un palé.

La estación *transfer* ha de verificar que el resto de las estaciones estén conectadas y disponibles para poner en marcha la cinta transportadora. También se le ha de avisar cuando una estación está preparada para introducir un elemento en el palé. Por otra parte, la estación de alimentación inicia su proceso cuando el *transfer* está activo y es avisado cuando hay un palé disponible para introducir un elemento. Las marcas utilizadas son las siguientes:

Variables que lee la estación *transfer* respecto de las estaciones de alimentación:

- **ESTACION_ON**: Confirma a la estación *transfer* que está en modo RUN.
- **FIN_CICLO**: Se activa cuando se dispone a introducir un elemento en el palé. Una vez realizado, se desactiva la marca para confirmar que el envío ha sido completado.

4. DESARROLLO DE LAS ESTACIONES DE ALIMENTACIÓN

VARIABLES QUE RECIBEN LAS ESTACIONES DE ALIMENTACIÓN POR PARTE DEL *transfer*:

- **INICIO_CICLO**: Confirma la llegada de un palé.
- **TRANSFER_ON**: Confirma que la estación *transfer* está activa.
- **STOP_TRANSFER**: Se ha producido una parada de la estación *transfer*.

Las tablas 4.1 y 4.2 muestran los espacios reservados para cada estación. Las tablas 4.3 y 4.4 las marcas de envío y recepción con la estación *transfer*.

Estación	Espacio de memoria
Bases	Byte 4
Rodamientos	Byte 4
Tapas	Byte 22

Cuadro 4.1: Bytes reservados para las variables que lee el *transfer*

Estación	Espacio de memoria
Bases	Byte 5
Rodamientos	Byte 5
Tapas	Byte 23

Cuadro 4.2: Bytes reservados para la recepción de datos enviados por el *transfer*

Estación	Dirección	Nombre
Bases	M 4.0	BASE_ON
Bases	M 4.1	FIN_CICLO
Rodamientos	M 4.0	RODAMIENTOS_ON
Rodamientos	M 4.1	FIN_CICLO
Tapas	M 22.0	TAPAS_ON
Tapas	M 22.1	FIN_CICLO

Cuadro 4.3: Envío de variables a la estación *transfer*

Estación	Dirección	Nombre
Bases	M 5.0	TRANSFER_ON
Bases	M 5.1	INICIO_CICLO
Bases	M 5.2	STOP_TRANSFER
Rodamientos	M 5.0	TRANSFER_ON
Rodamientos	M 5.1	INICIO_CICLO
Rodamientos	M 5.2	STOP_TRANSFER
Tapas	M 23.0	TRANSFER_ON
Tapas	M 23.1	INICIO_CICLO
Tapas	M 23.2	STOP_TRANSFER

Cuadro 4.4: Recepción de variables que provienen del *transfer*

4.2 Estación de bases

La estación de bases es la primera que aporta un elemento al palé. El programa de usuario está compuesto por el bloque de organización OB1 y es el encargado de ejecutar los diferentes subprogramas que componen el programa de usuario.

Cuando se carga el programa en la CPU y se pasa a modo RUN, se parte de una etapa de inicialización. En esta etapa, verifica que el *transfer* está conectado e inicia el proceso de selección de la base. Una vez verificado que el elemento está puesto correctamente y antes de que se activen las ventosas que sujetan la base, se comprueba que se ha activado la marca **INICIO_CICLO** para confirmar que se dispone de palé para introducirle una base.

Por otra parte, la estación de bases ha de informar hasta en tres situaciones al *transfer* para que haya una buena sincronización. Al pasar a RUN, se activa **BASE_ON** mediante la marca SM0.0. Es una marca indispensable para que la zona *Transfer-Base* pase desde el estado de inicialización hasta el estado donde espera recibir la base. La segunda marca que lee el *transfer* es **FIN_CICLO**. Cuando la estación está preparada para realizar el envío de una base, activa la marca para que el *transfer* tenga constancia del suceso. Cuando el traslado se ha producido con éxito, se desactiva.

Por último, si el usuario detiene el *transfer*, se activa la marca **STOP_TRANSFER** para detener la estación.

En el anexo [A.2.9](#) se dispone del graficet de la estación de bases.

4.3 Estación de rodamientos

La estación de rodamientos es la segunda parada donde se detiene el palé para recibir el segundo elemento que compondrá la pieza final. OB1 ejecuta los subprogramas de forma cíclica. El grado de dificultad del programa de usuario es más elevado que respecto de la estación de bases. Hay dos tipos de rodamientos y en este proyecto

se ha decidido que los de mayor grosor sean las que se introducen en la base que ya dispondrá el palé.

Esta estación recoge el rodamiento desde el alimentador hasta la estación de medida para que un elevador acerque el rodamiento hasta un palpador para comprobar su longitud. Si no tiene la medida correcta, la pieza es expulsada. En caso contrario, el segundo manipulador viene a recoger la pieza. Antes de llevar el rodamiento hasta el palé, se comprueba que **INICIO_CICLO** esté activo. Esta marca se desactiva cuando se ha completado el envío.

Cuando el segundo manipulador ha hecho el giro con el rodamiento y está a punto de descender para soltarlo, activa la marca **FIN_CICLO** para que el *transfer* sepa que se está introduciendo el rodamiento en el palé. Al haber finalizado, se desactiva.

Al igual que en la estación de bases, al cargar el programa de usuario en la CPU y pasar a modo RUN, se activa la marca **RODAMIENTO_ON**. Y se iniciará el proceso cuando **TRANSFER_ON** esté activa.

En el anexo [A.2.11](#) se dispone del graficet de la estación de rodamientos.

4.4 Estación de tapas

La última estación alimenta una tapa al palé para completar la pieza. Es la más laboriosa de programar ya que hay que identificar la tapa por su material, color y tamaño. En este proyecto, la tapa que se introduce en el palé es metálica o nylon blanca.

El proceso comienza cuando se activa la marca **TRANSFER_ON**. La alimentación de la primera pieza y la posterior introducción en el plato transcurre desde la *ETAPA_0* hasta la *ETAPA_10*. Cuando la tapa ha llegado a la primera posición del plato, comienza el proceso de verificación. En el programa de usuario se han reservado hasta cuatro bytes para comprobar las características de las tapas que están situadas en el plato:

- **PIEZAS:** Cada vez que se introduce una tapa en el plato, se introduce '1' en el bit menos significativo. Cuando la tapa varía una posición, cada bit rota una posición hacia la izquierda. Si el bit más significativo es un '1', un manipulador desplaza la tapa hasta el palé.
- **METALICA:** Cuando el captador inductivo detecta una tapa de aluminio, el bit menos significativo pasa a ser '1'. Y el bit rota hacia la izquierda por cada movimiento de la tapa.
- **NYLON:** El mismo proceso que el anterior con la diferencia de que ahora es un detector capacitivo el que detecta la tapa de nylon.
- **NEGRO:** Las tapas Nylon negras se van a descartar. Un detector fotoeléctrico es el encargado de detectar cuando una tapa es negra para que un manipulador sea el encargado de retirarla del plato.

Si la tapa no cumple las características para ser transferida, un manipulador, que dispone de un brazo con 3 ventosas, es el encargado de expulsarla. En el programa de usuario, este proceso comprende desde la *ETAPA_ 30* hasta la *ETAPA_ 35*. En caso contrario, otro manipulador caracterizado por una pinza capaz de sujetar una tapa, es la que realiza el trasvase desde el plato hasta el palé. Este proceso comprende desde la *ETAPA_ 24 - ETAPA_ 29* y *ETAPA_ 36 - ETAPA_ 39*.

En el caso de que se realice el envío de la tapa cuando la marca **INICIO_ CICLO** esté activa, el manipulador transfiere la pieza. En el caso de que la marca no esté activa y la tapa esté en la última posición del plato, permanecerá a la espera.

En el anexo **A.2.13** se dispone del graficet de la estación de tapas.

4.5 Guía GEMMA para las estaciones de alimentación

En el apartado **3.2.8** se ha realizado el estudio de la guía GEMMA de la estación *transfer*. Para las estaciones de alimentación también se ha realizado dicho estudio. Aunque se apliquen de forma individual para cada una, disponen de idénticas botoneras y sus procesos de alimentación siguen una misma estructura. En la imagen **4.1** se muestran los estados elegidos y las características de cada uno son las siguientes:

- *Parada en el estado inicial (A1)*: El programa de usuario parte desde un estado de reposo donde se inicializan todos los actuadores y la estación queda pendiente de iniciación del proceso. Pulsando *MARCHA*, el programa pasa al siguiente estado. Por otra parte, si el operario ha tenido que parar el proceso debido a un estado de emergencia, desde *A5* el programa retorna a *A1* donde se vuelven a inicializar los actuadores para que el proceso retorne a su funcionamiento normal.
- *Producción normal (F1)*: El programa de usuario va ejecutando los estados de proceso que parten desde la alimentación de un elemento hasta el traslado al palé. El programa saldrá de la producción normal cuando se pulse la parada de emergencia, parada solicitada o para verificar el proceso de forma manual.
- *Marcha de cierre (F3)*: Si en algún instante el usuario ve oportuno reiniciar el proceso de la estación desde *F1*, deberá pulsar *REARME*. Se detiene el proceso y pulsando *PARADA* se vuelve al estado de *Parada en el estado inicial*. Es recomendable acceder a este estado cuando se produzca una parada de emergencia en la estación *transfer*.
- *Marchas de verificación con orden (F5)*: El proceso de la estación está programado para que actúe de forma automática. Aún así, partiendo desde *F1* y poniendo el selector a modo manual, el usuario puede comprobar paso a paso el funcionamiento de la estación.
- *Parada de emergencia (D1)*: El usuario tiene la opción de parar el proceso debido a una anomalía en la estación pulsando *PARADA* durante 1.5 segundos. Cuando se ha comprobado que el operario no puede sufrir riesgos en cuanto a seguridad, ni de los productos, pulsando *MARCHA* se pasará al siguiente estado

4. DESARROLLO DE LAS ESTACIONES DE ALIMENTACIÓN

donde se podrá corregir la anomalía. El usuario tiene la posibilidad de cortar la alimentación del aire para poder facilitar el diagnóstico.

- *Tratamiento de defectos (D2)*: Después del defecto, el operario puede solucionar cualquier alteración que ha provocado la parada del proceso. Una vez corregido el problema, pulsando *MARCHA*, el programa pasa al estado A5.

- *Preparación posterior al defecto (A5)*: Se debe preparar la estación para volver a su funcionamiento normal. El operario ha de retirar cualquier pieza que pueda interrumpir el proceso de producción. Pulsando *MARCHA* el programa pasa a A6.

- *Puesta en el estado inicial (A6)*: Es el paso anterior a la inicialización de la estación. Se realizará la última comprobación antes de llegar a la parada en el estado inicial. Pulsando *REARME* se accede al A1.

- *Parada pedida (A3)*: En cualquier instante, el operario puede parar el proceso pulsando *PARADA* durante menos de 1.5 segundos.

- *Parada obtenida (A4)*: La estación permanece en reposo hasta el el operario decida pulsar *MARCHA* para retomar F1.

Las tres estaciones disponen cada uno de un pulsador de parada de emergencia para detener completamente el proceso de la estación cortando la alimentación al PLC.

En la figura 4.2 se puede observar el graficet de control de la guía GEMMA. Para que el programa de usuario funcione en los estados *F1* y *F5*, ha de permanecer activo la marca **I_M**. Esta marca está incluida en todas las líneas de código donde se condicionan las etapas de proceso de la estación. Si en algún momento se activa la marca **STOP_TRANSFER_ESTACION**, automáticamente se desactiva **I_M**.

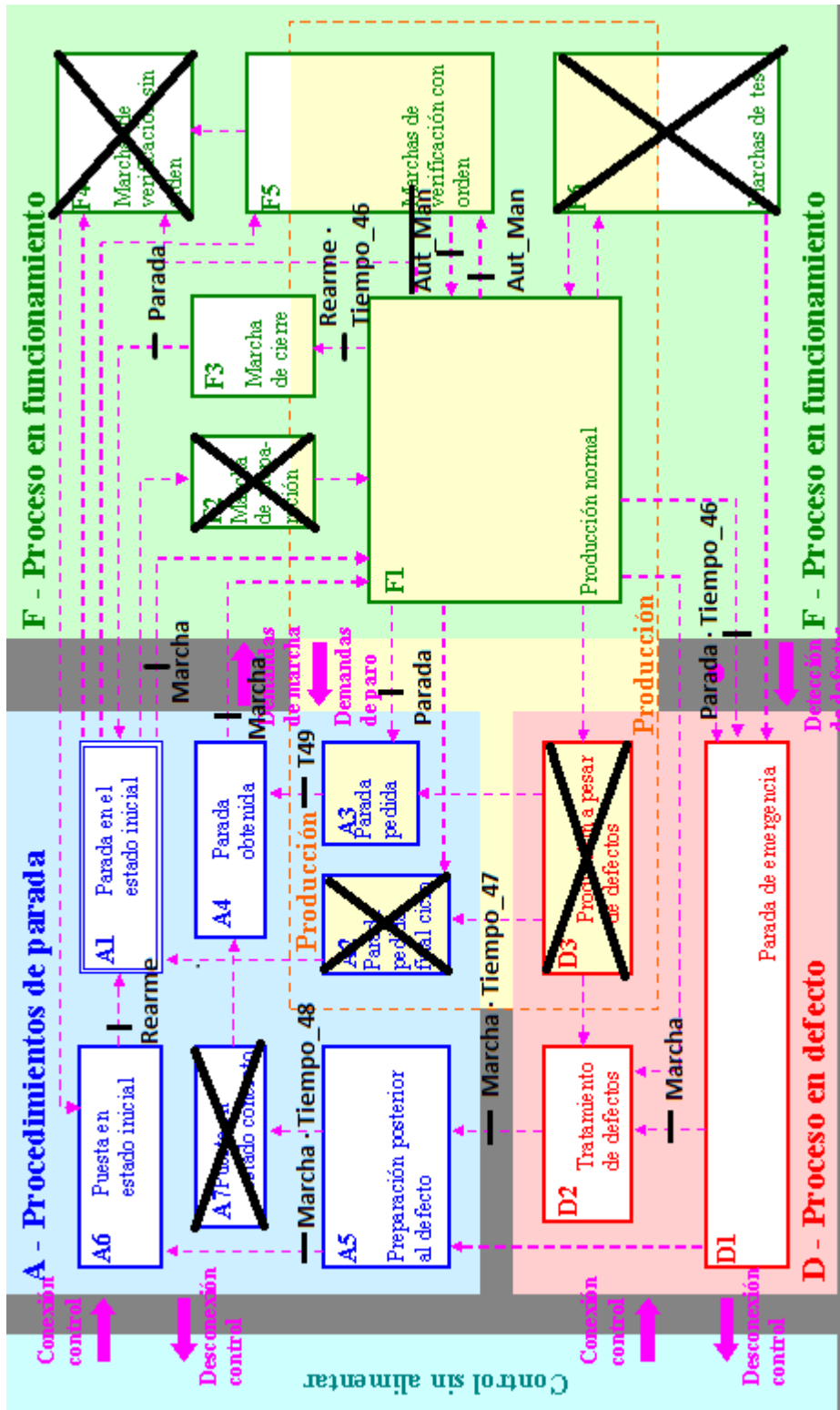


Figura 4.1: [13] Guía GEMMA que se utiliza para las estaciones de alimentación

4. DESARROLLO DE LAS ESTACIONES DE ALIMENTACIÓN

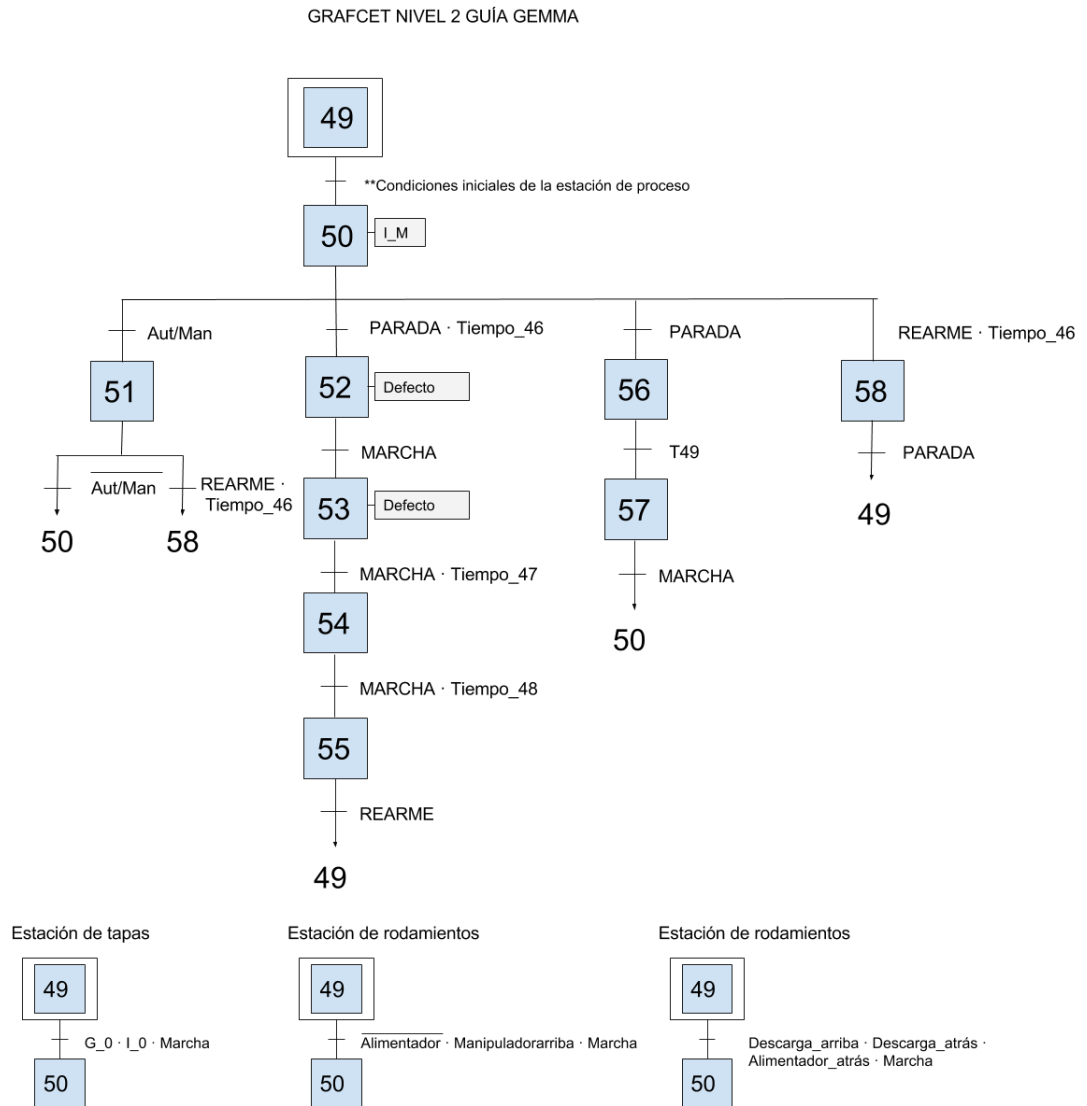


Figura 4.2: Graficet de los estados de la guía GEMMA

DESARROLLO DE LA ESTACIÓN DE ALMACENAMIENTO

La estación de almacenamiento es la encargada de depositar una pieza ya creada. En este apartado se detalla el programa de usuario que se ha hecho servir para el proceso de almacenaje.

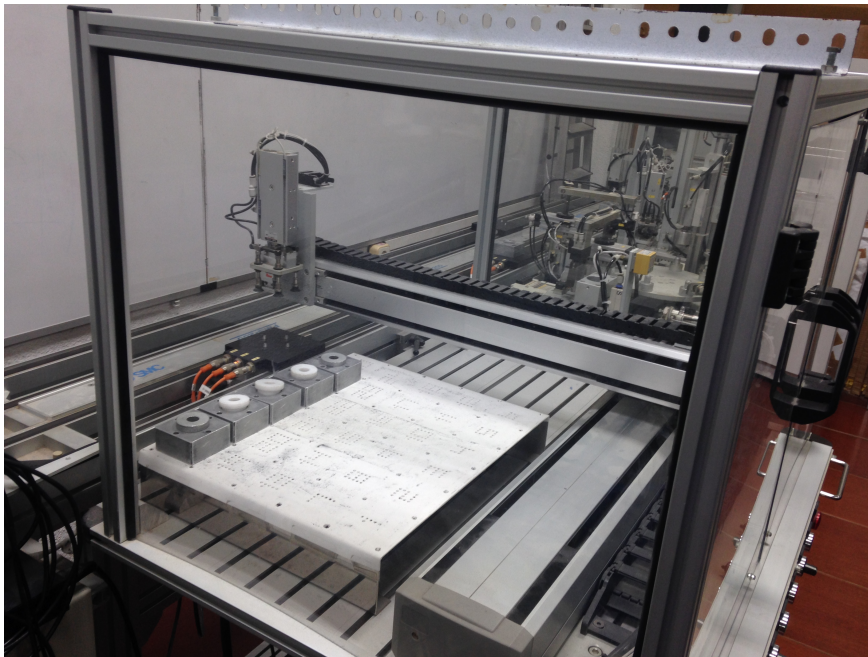


Figura 5.1: Estación de almacenamiento

5.1 Funcionamiento de la estación de almacenamiento

Cuando se ha cargado el programa de usuario y la CPU pasa a modo RUN, la estación de almacenamiento confirma que está activo y preparado para recoger una pieza a la estación *transfer*. Cuando llega un palé con la pieza ya creada, un manipulador controlado por dos ejes motorizados es el encargado de recogerla y depositarla en una superficie de treinta posiciones.

5.2 Programa de usuario

El PLC de la estación de almacenamiento es un S7-300. En este apartado se detallan los bloques de programa que se han hecho servir para el correcto funcionamiento. Los bloques de programa utilizados son los siguientes:

Bloques de organización

- OB1
- OB100

Funciones

- FC0 “POS_INIT”
- FC2 “POS_CTRL”
- FC5 “SALIDAS”
- FC6 “MARCAS”
- FC7 “TEMPORIZADORES”
- FC8 “TRATAMIENTO DE LOS MOVIMIENTOS”
- FC9 “GUIA GEMMA”

Funciones del sistema

- SFC 79 “SET”
- SFC 80 “RSET”

Datos globales

- DB1 “UDT 1”
- DB2 “UDT 2”

5.2.1 Bloques de organización OB1 y OB100

OB1 es el encargado de la ejecución cíclica de las funciones del programa, mientras que OB100 inicializa la tarjeta FM353 del módulo de control “POS_INIT” y activa el modo de control “Búsqueda del punto de referencia”.

En la figura 5.2 se muestra un ejemplo de la llamada a “POS_INIT”. Este módulo introduce los valores de direccionamiento en DB. Tiene tres entradas. **DB_NO** es de

tipo INT y se introduce el número de bloque de datos. **CH_NO** representa el número de ejes y se introduce un 0 porque solo hay un eje en el módulo. En **LADDR** se introduce la dirección del módulo. La salida del módulo es **RET_VAL** y muestra un valor en caso de error [14].

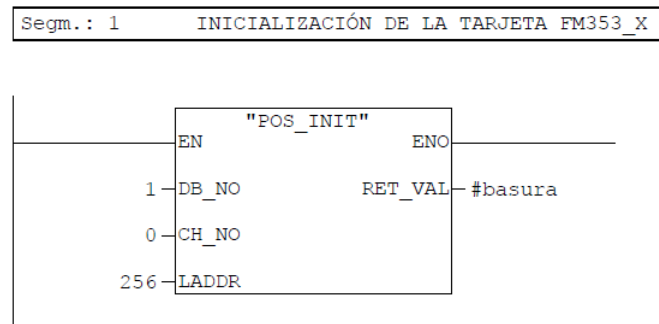


Figura 5.2: Inicialización de la tarjeta FM353_X

5.2.2 Datos de usuario (UDT)

El módulo FM 353 dispone de un bloque de datos globales con las variables que son procesadas por los bloques de funciones que dispone la librería del módulo. El usuario puede acceder a estas señales utilizando direcciones simbólicas. Para ello se ha de crear un AW-DB con estructura UDT.

Los UDT son estructuras de datos creados desde el administrador SIMATIC. En este proyecto se dispone de dos UDT para el control del eje_x y el eje_y.

5.2.3 FC8 “TRATAMIENTO DE LOS MOVIMIENTOS”

En esta función se detallan una serie de pasos para realizar el movimiento de los ejes lineales.

Cuando la tarjeta FM 353 está en modo “Busqueda del punto de referencia” (una puesta en marcha del módulo), se activa el controlador, el servo y el *override*. Después se activa la variable *Start* para realizar la búsqueda del origen de los ejes.

El siguiente paso es cambiar del modo de “Busqueda del punto de referencia” al “Modo manual de datos” (MDI) para habilitar las señales del modo de coordenadas, la consigna de posición y la velocidad de movimiento. Se cargan las coordenadas de recogida de la pieza, la señal MDI enable y, por último, se activa la señal de *start* para realizar el movimiento de los ejes lineales.

La superficie donde se almacena la pieza dispone de treinta posiciones, seis columnas y cinco filas. El proceso de almacenaje es el siguiente. Hay un contador, Z0, que incrementa en uno cada vez que se recoge una pieza. Cuando se ha llenado todas las posiciones de una fila, este contador se resetea y activa la marca **posición_6_x**. En este

momento, un segundo contador, Z1, incrementa su valor para que la siguiente pieza se coloque en la siguiente fila. Cuando el valor de Z1 es 5, indica que se ha completado el proceso de almacenaje de la plataforma y se activa **posición_5_** y para detener el proceso.

En la última línea de código se intercambian las señales de control de movimiento mediante "POS_CTRL". Esta función, que pertenece a la librería del módulo FM 353, se encarga de procesar las peticiones de lectura y escritura de las variables de los datos globales. Se caracteriza por sincronizar con los módulos para el intercambio de señales/datos.

5.2.4 FC6 "MARCAS"

Como se ha comentado en el apartado anterior, hay una serie de marcas que controlan las posiciones de almacenamiento de las piezas:

- *posicion_6_x*: Las piezas se van depositando de izquierda a derecha en las diferentes filas que conforma la superficie. Al inicio del proceso, esta marca es reseteada. Cada vez que el contador "Z0" ha incrementado seis veces, se activa la marca para que la próxima pieza se coloque en la siguiente fila. Se vuelve a resetear *posicion_6_x* para volver a repetir el proceso.
- *posicion_5_y*: Una vez se han completado todas las filas que dispone la superficie, esta marca es la encargada de confirmar al usuario de que el proceso ha sido completado. El programa de usuario dispone de un segundo contador, "Z1". Este comprueba las filas que han sido ocupadas. Cuando se ha introducido la última pieza en la última fila, Z1 se activa. El programa de usuario pasa a una línea de código reservada donde se confirma el almacenamiento completado. Cuando el usuario pulse **REARME**, se resetea la marca y se reinicia el proceso.

Las marcas que son enviadas al *transfer* son las siguientes:

- **ALMACEN_ON**: Permanece activa desde que la CPU pasa a modo ON.
- **FIN_CICLO**: Se activa justo antes de descender las ventosas que recogen la pieza. Por otra parte, se desactiva en la etapa de inicialización y en la etapa de desplazamiento de la pieza hasta su posición de almacenamiento.

Las marcas que el almacén recibe por parte del *transfer* son las siguientes:

- **INICIO_CICLO**: Confirma la llegada de un palé.
- **TRANSFER_ON**: Confirma que la estación *transfer* está activa.
- **STOP_TRANSFER**: Se ha producido una parada de la estación *transfer*.

5.2.5 FC5 “SALIDAS”

Las salidas que más se utilizan durante el proceso son las siguientes:

- *A+*: Se activa para descender las ventosas que dispone el manipulador para recoger la pieza.
- *V+*: Se activa para realizar el vacío de las ventosas desde que se recoge la pieza hasta que se ha colocado en la posición de almacenamiento.
- *piloto_de_defecto*: Una vez la superficie está completa se enciende el piloto luminoso.

5.2.6 FC4 “ETAPAS”

En el anexo [A.23](#) se muestra el GRAFCET de primer y tercer nivel de la estación de almacenamiento. En este apartado se destaca el contenido más importante de FC4.

- La etapa_0 es de inicialización del programa de usuario. Se muestran las condiciones iniciales de los actuadores y la mano se dirige al eje de coordenadas (0,0).
- En la etapa_1, la mano se dirige a la posición de recogida de pieza.
- En la etapa_2, se comprueba que la estación *transfer* esté activa y que llegue un palé con una pieza al punto de recogida.
- Desde la etapa_3 hasta la etapa_7, se recoge la pieza y es almacenada.
- En la etapa_9 se pueden producir dos condiciones, que haya espacio para almacenar la siguiente pieza o no quede espacio. Mientras *posición_5* permanezca negada, la mano recogerá la siguiente pieza. En caso contrario, pasa a la etapa_10.
- En la etapa_10, el usuario deberá pulsar **REARME** para volver a retomar el proceso desde el principio.

5.2.7 Guía GEMMA de la estación de almacenamiento

En el anexo [A.1](#) se muestran la guía GEMMA. Los estados utilizados son los mismos que se detallan en el apartado [4.5](#). El graficet que representan los estados GEMMA están incluidos en el anexo [A.2](#).

CONCLUSIÓN

Este documento ha sido elaborado para comprender y conseguir que las estaciones automatización que dispone el laboratorio de robótica de la EPS puedan funcionar en conjunto. Cada año, una serie de alumnos tienen su primera toma de contacto con unas estaciones a escala reducida de una cadena de montaje. Siguiendo una serie de pasos, en las sesiones de prácticas se podrá comprobar su completo funcionamiento pudiendo intervenir los programas de usuario de diferentes compañeros.

Se han descrito cada una de las estaciones de proceso, la subred de comunicación MPI, la red de entradas y salidas que dispone la estación *transfer* para la comunicación con sus sensores y actuadores, y las funciones X_GET y X_PUT para la recepción y envío de datos entre estaciones. Además, se ha realizado la programación de cada una de las estaciones y se ha enfatizado en el entendimiento de la estación *transfer* para comprender la comunicación y sincronización con el resto.

En este TFG se ha debido comprender el entorno SIMATIC para interactuar con los S7-300. Se ha escrito una guía para la creación, configuración, programación y carga de un proyecto en SIMATIC.

En definitiva, se ha realizado este TFG para entender la comunicación entre las estaciones de proceso que dispone el laboratorio de robótica y facilitar a los futuros alumnos una serie de pasos para conseguir, mediante sus programas, el proceso de creación de una pieza y su almacenamiento.

De cara entorno profesional, la automatización es un sector que en la actualidad ofrece una gran demanda de empleo. Hay diversas infraestructuras que requieren el control de procesos industriales para diversas soluciones. El trabajo realizado en este TFG se podría entender como una práctica de aprendizaje de cara a una posible salida laboral.



ANEXO I: GUÍA GEMMA

A.1 Guía GEMMA de las estaciones de procesos

A.1.1 Estación de almacenamiento

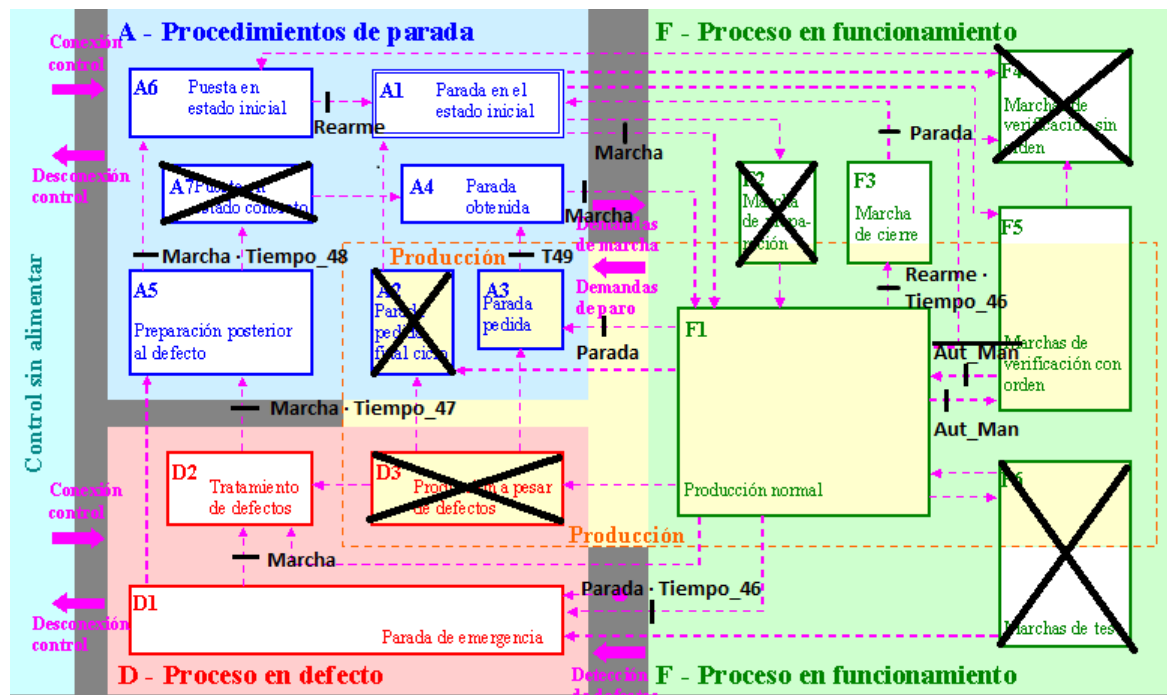


Figura A.1: [13] Guía GEMMA de la estación de almacenamiento

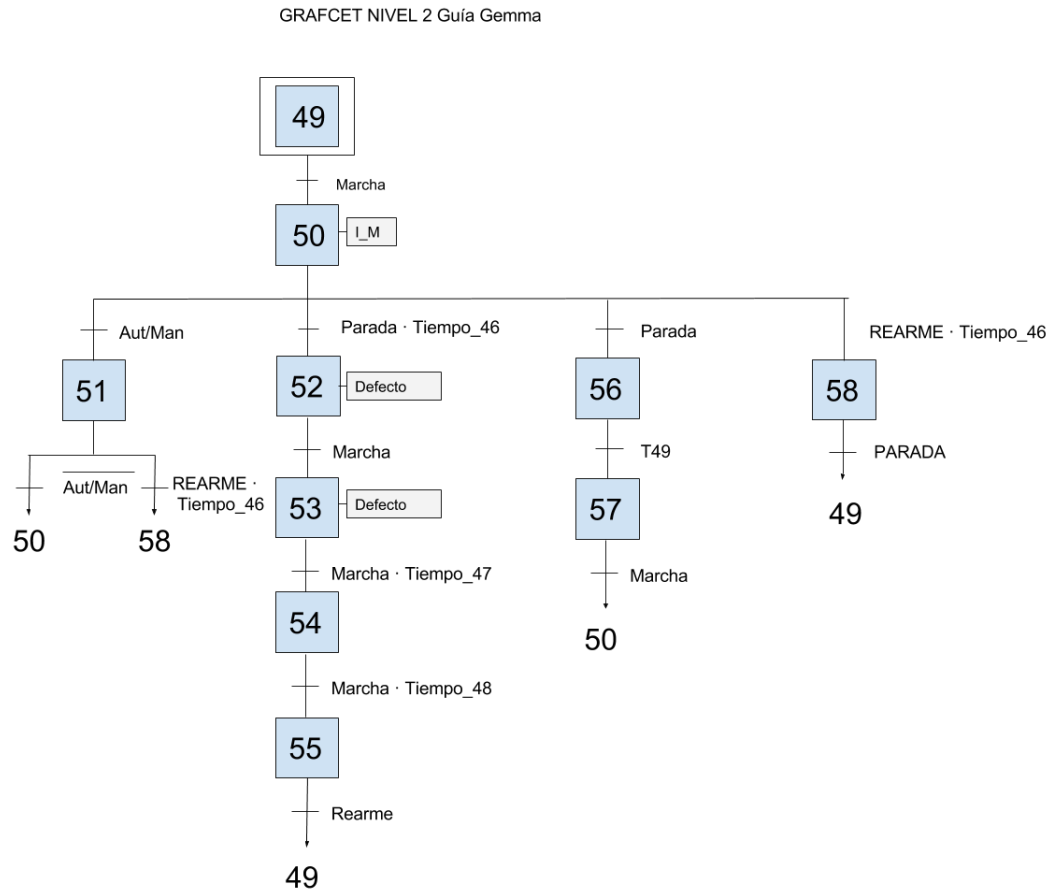


Figura A.2: GRAFCET que establece las condiciones exigidas por la guía GEMMA

A.1.2 Estación *transfer*

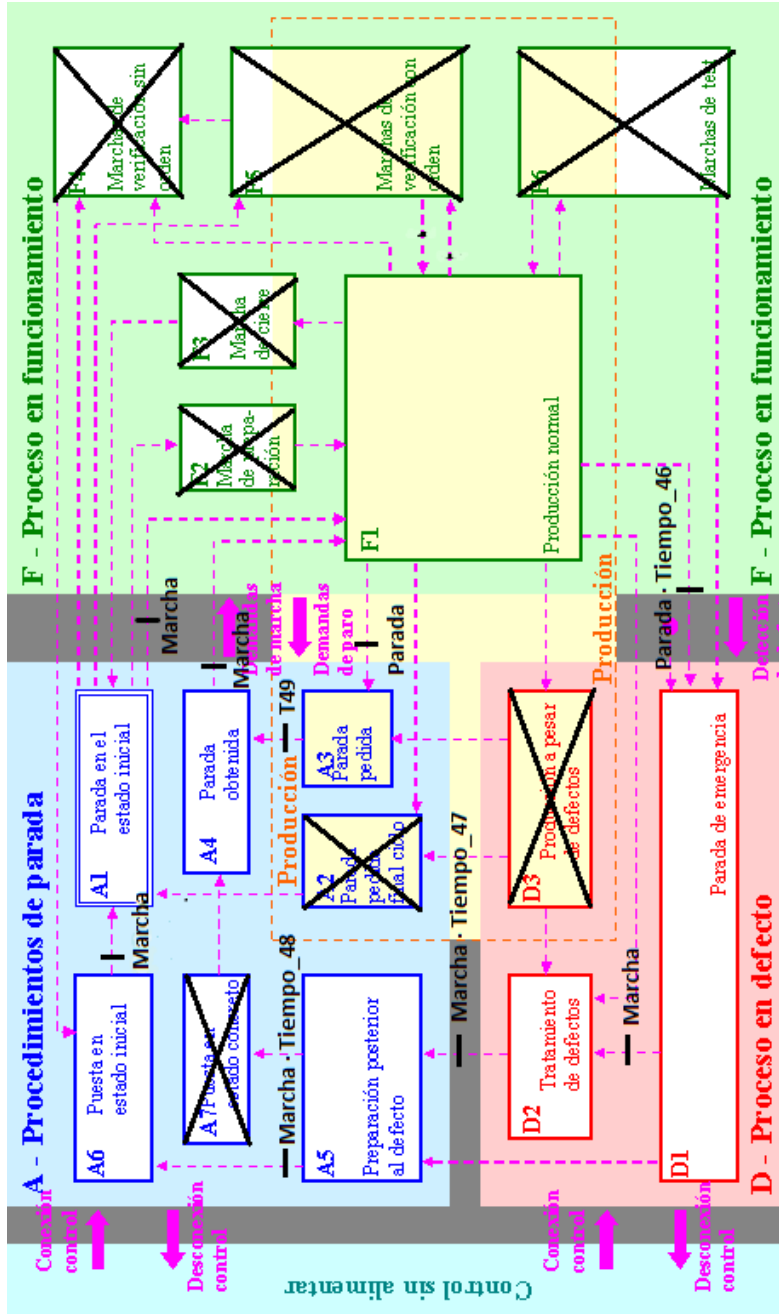


Figura A.3: [13] Guía GEMMA de la estación *transfer*

GRAFSET NIVEL 2 Guía Gemma

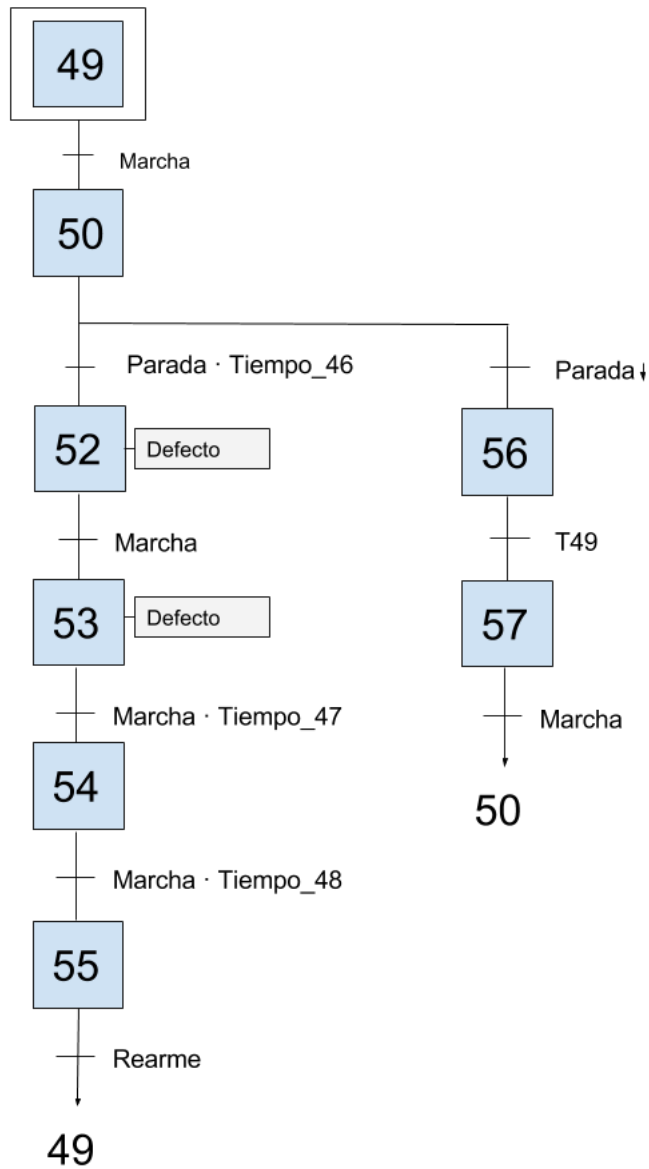


Figura A.4: GRAFCET que establece las condiciones exigidas por la guía GEMMA

A.2 GRAFCETS

A.2.1 GRAFCET nivel 1 de la Estación Transfer-Base

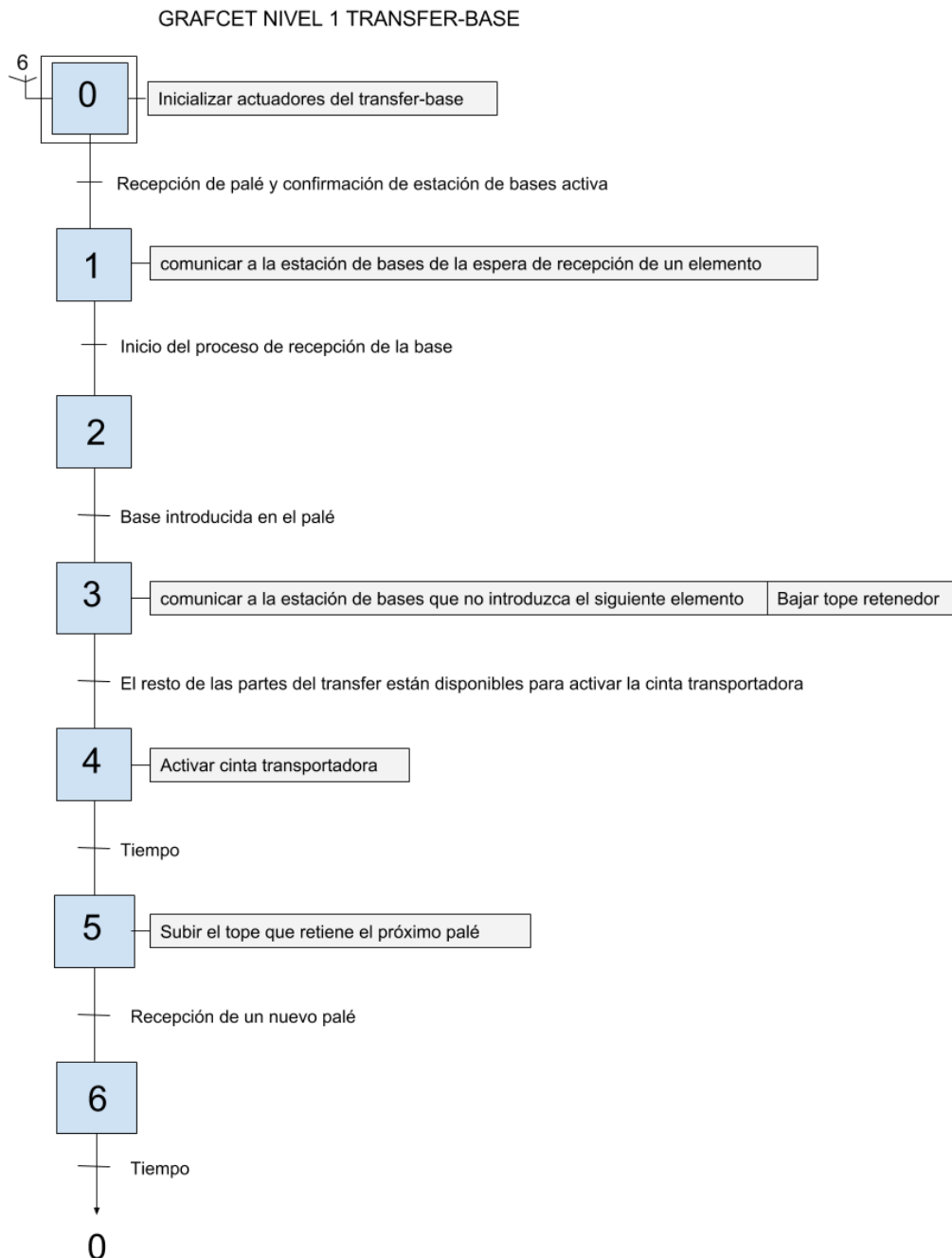


Figura A.5: GRAFCET Nivel 1 Estación de transfer-base

A.2.2 GRAFCET nivel 3 de la Estación Transfer-Base

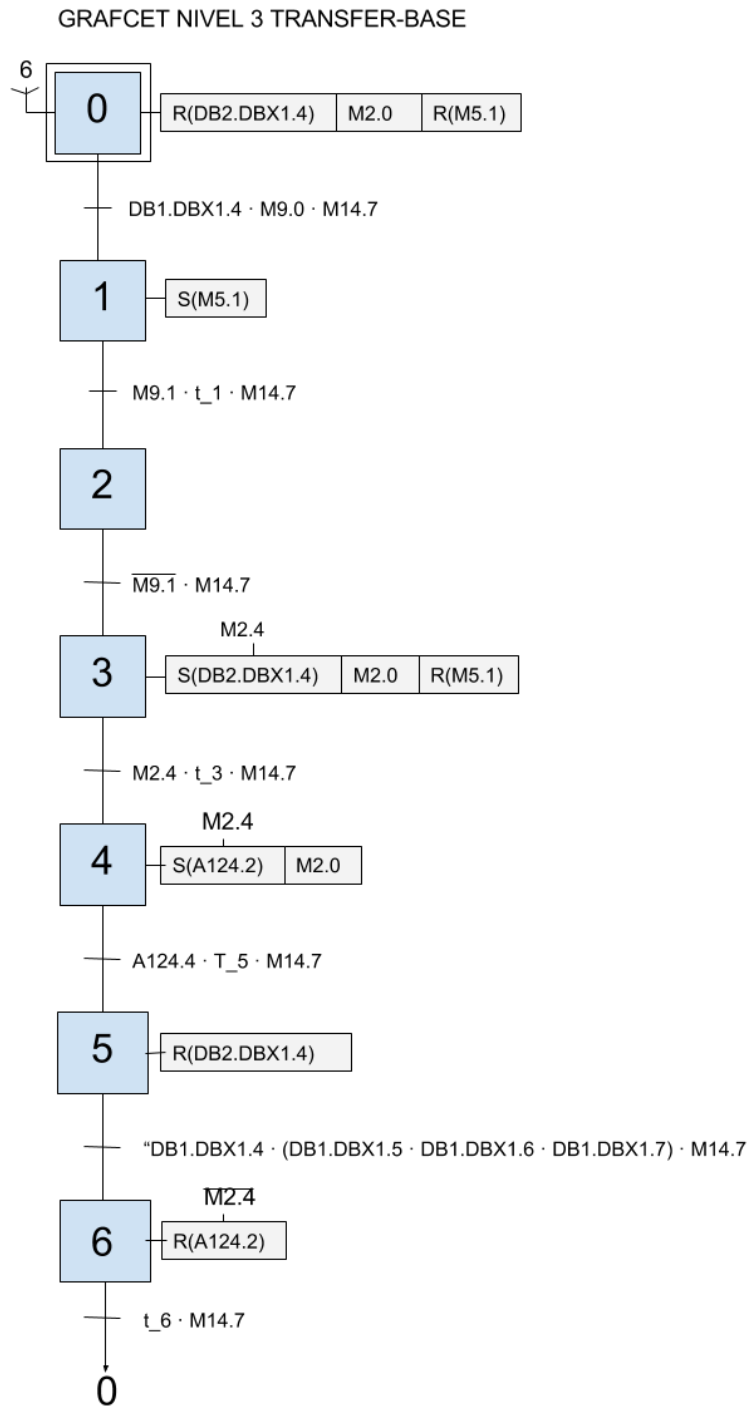


Figura A.6: GRAFCET Nivel 3 Estación de transfer-base

A.2.3 GRAFCET nivel 1 de la Estación Transfer-Rodamientos

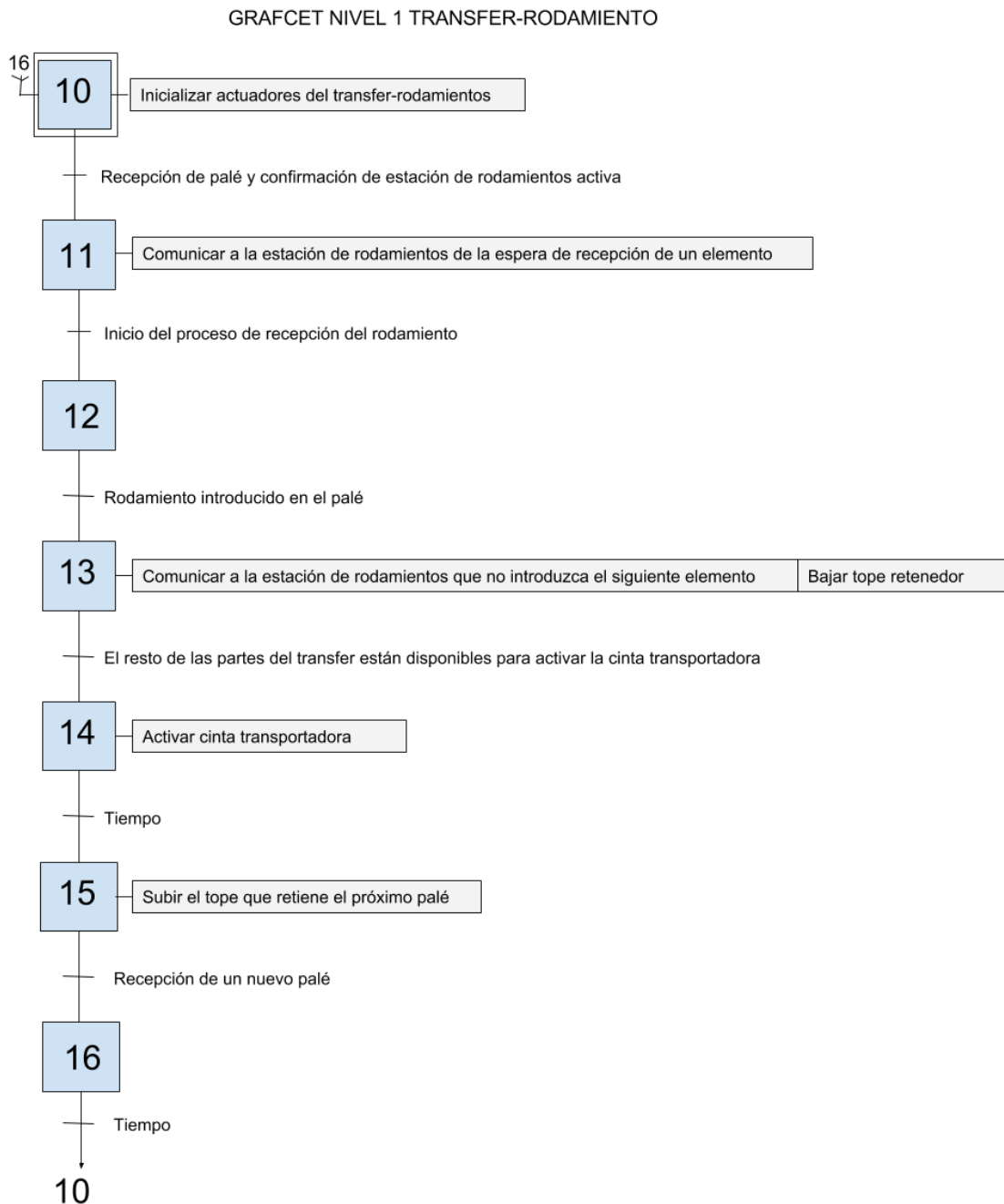


Figura A.7: GRAFCET Nivel 1 Estación de transfer-rodamientos

A.2.4 GRAFCET nivel 3 de la Estación Transfer-Rodamientos

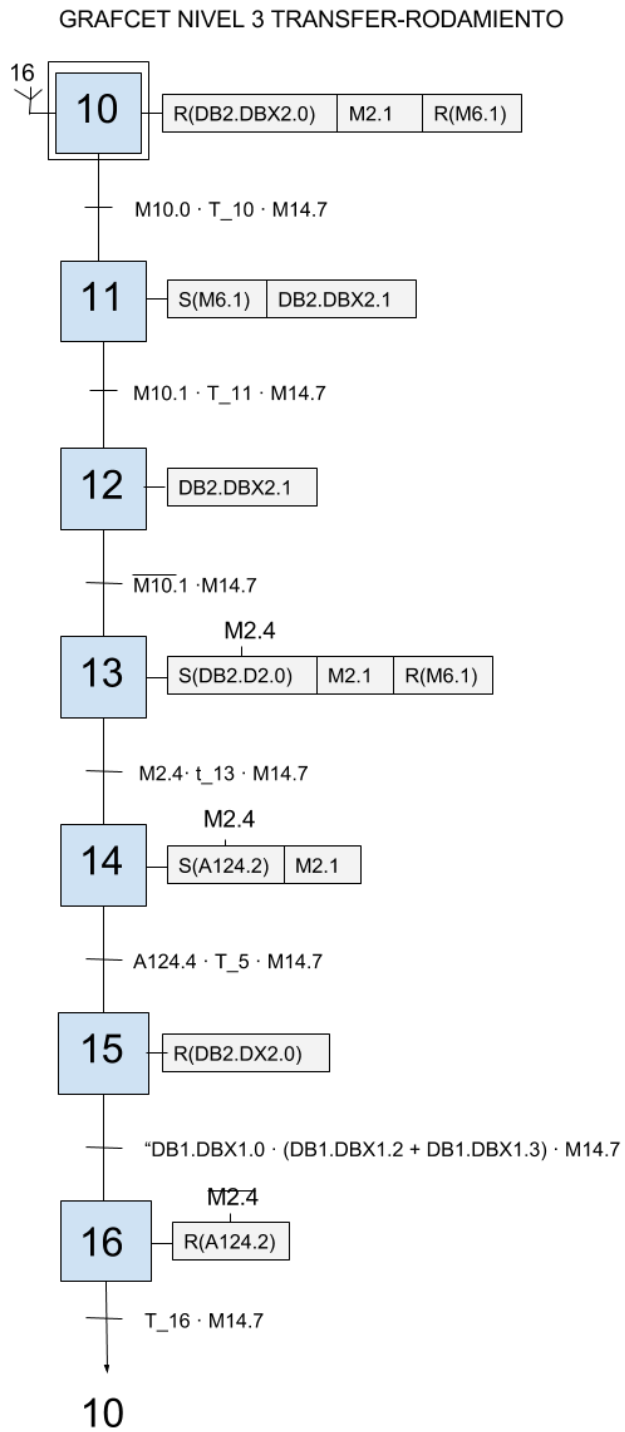


Figura A.8: GRAFCET Nivel 3 Estación de transfer-rodamientos

A.2.5 GRAFCET nivel 1 de la Estación Transfer-Tapas

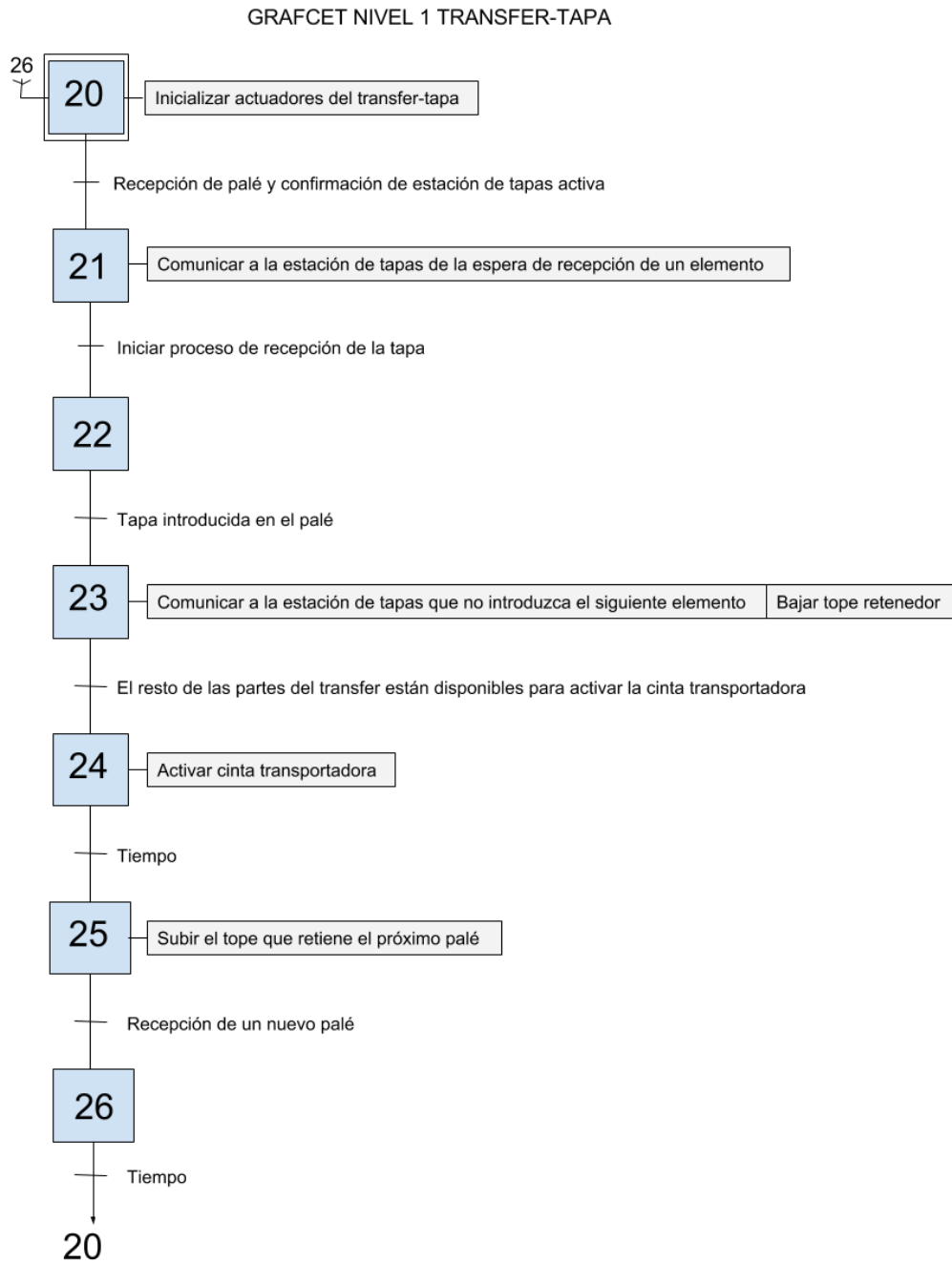


Figura A.9: GRAFCET Nivel 1 Estación de transfer-tapas

A.2.6 GRAFCET nivel 3 de la Estación Transfer-Tapas

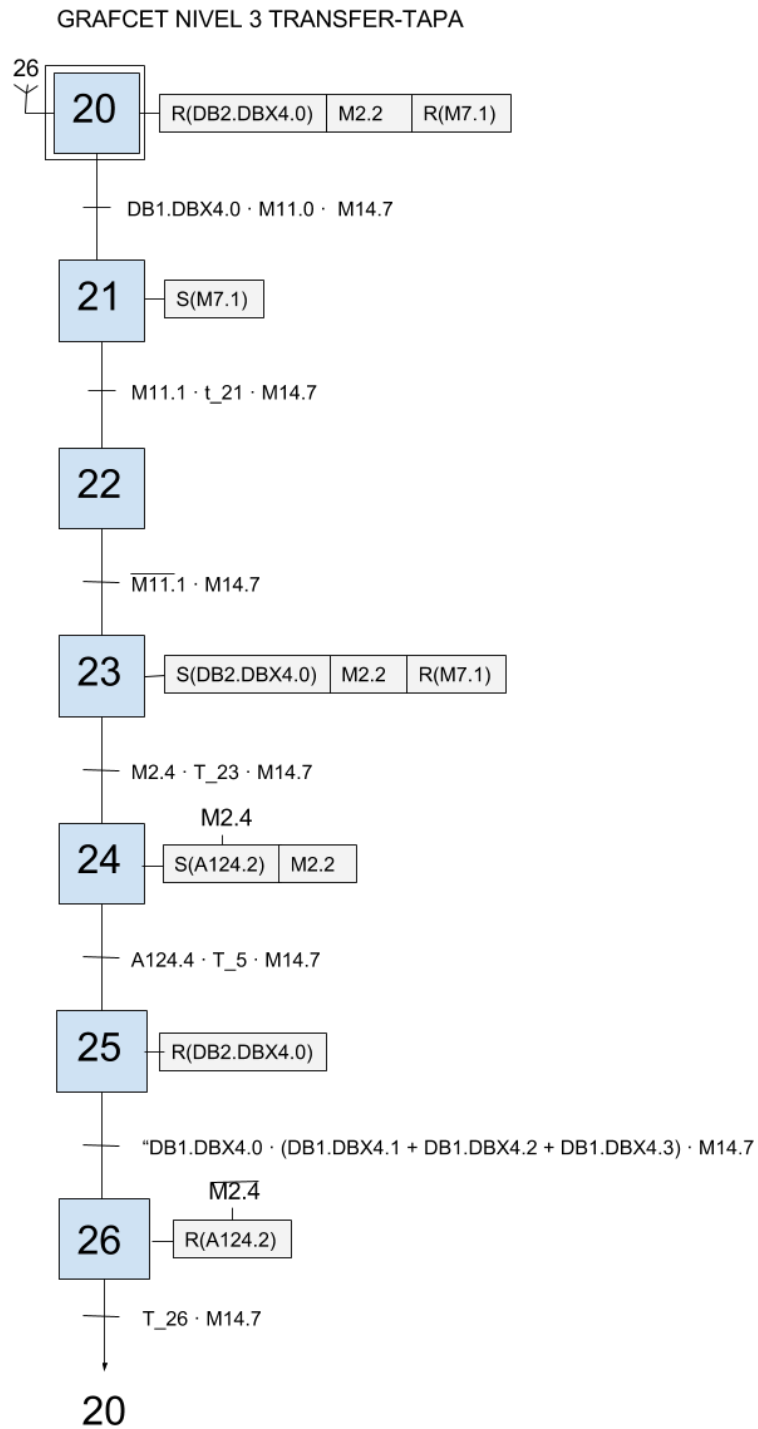


Figura A.10: GRAFCET Nivel 3 Estación de transfer-tapas

A.2.7 GRAFCET nivel 1 de la Estación Transfer-Almacén

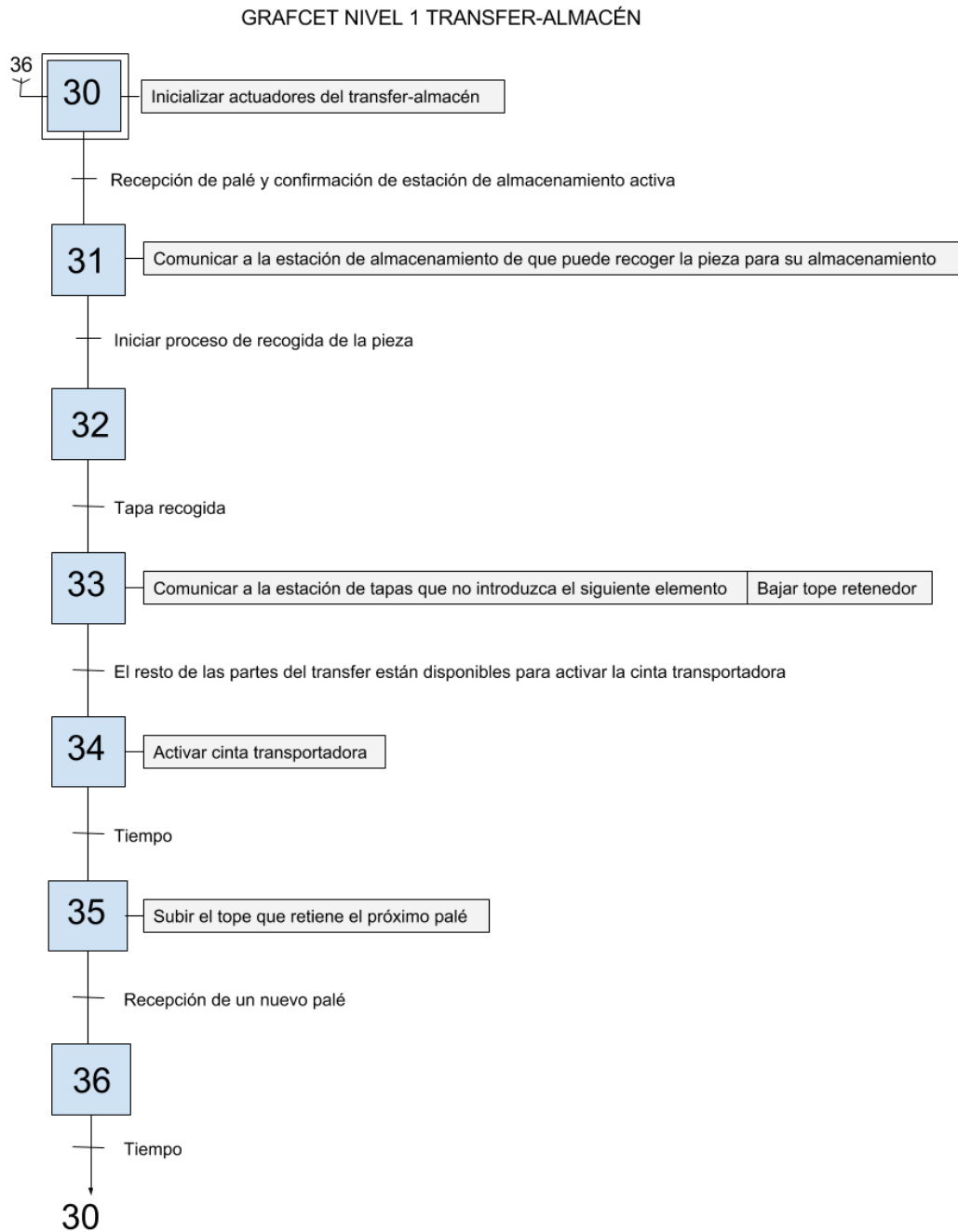


Figura A.11: GRAFCET Nivel 1 Estación de transfer-almacén

A.2.8 GRAFCET nivel 3 de la Estación Transfer-Almacén

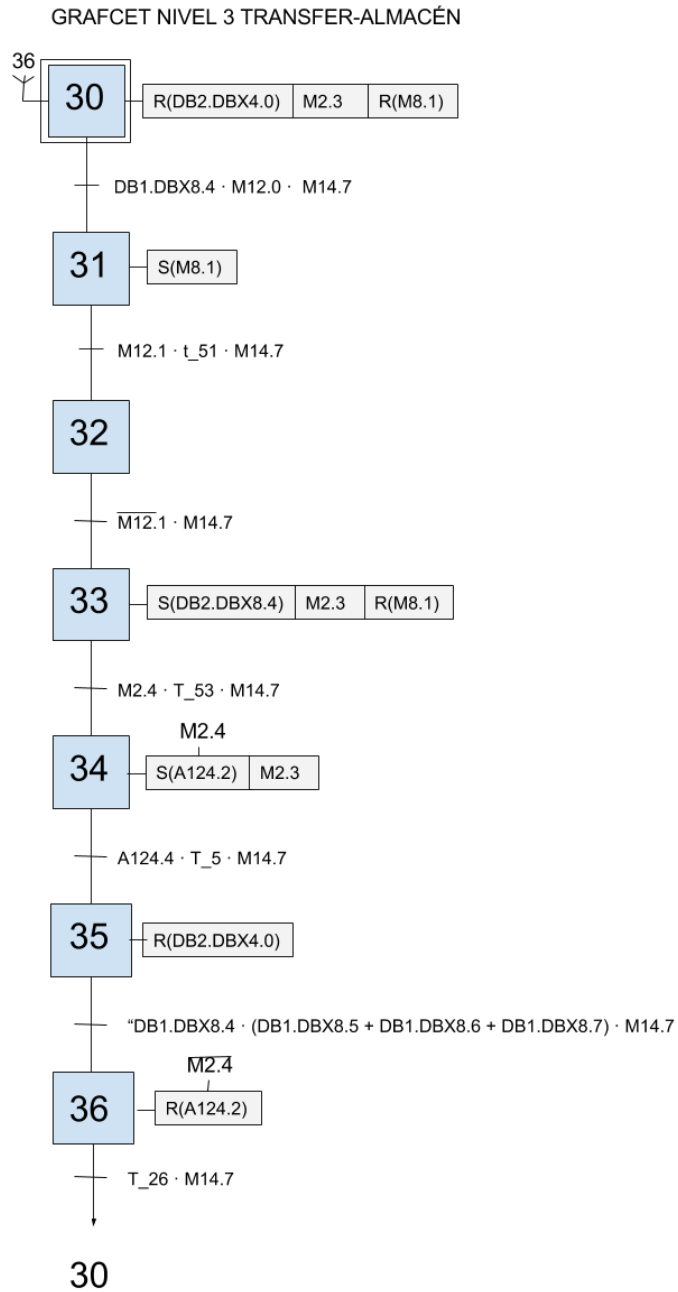


Figura A.12: GRAFCET Nivel 1 Estación de transfer-almacén

A.2.9 GRAFCET nivel 1 de la Estación de bases

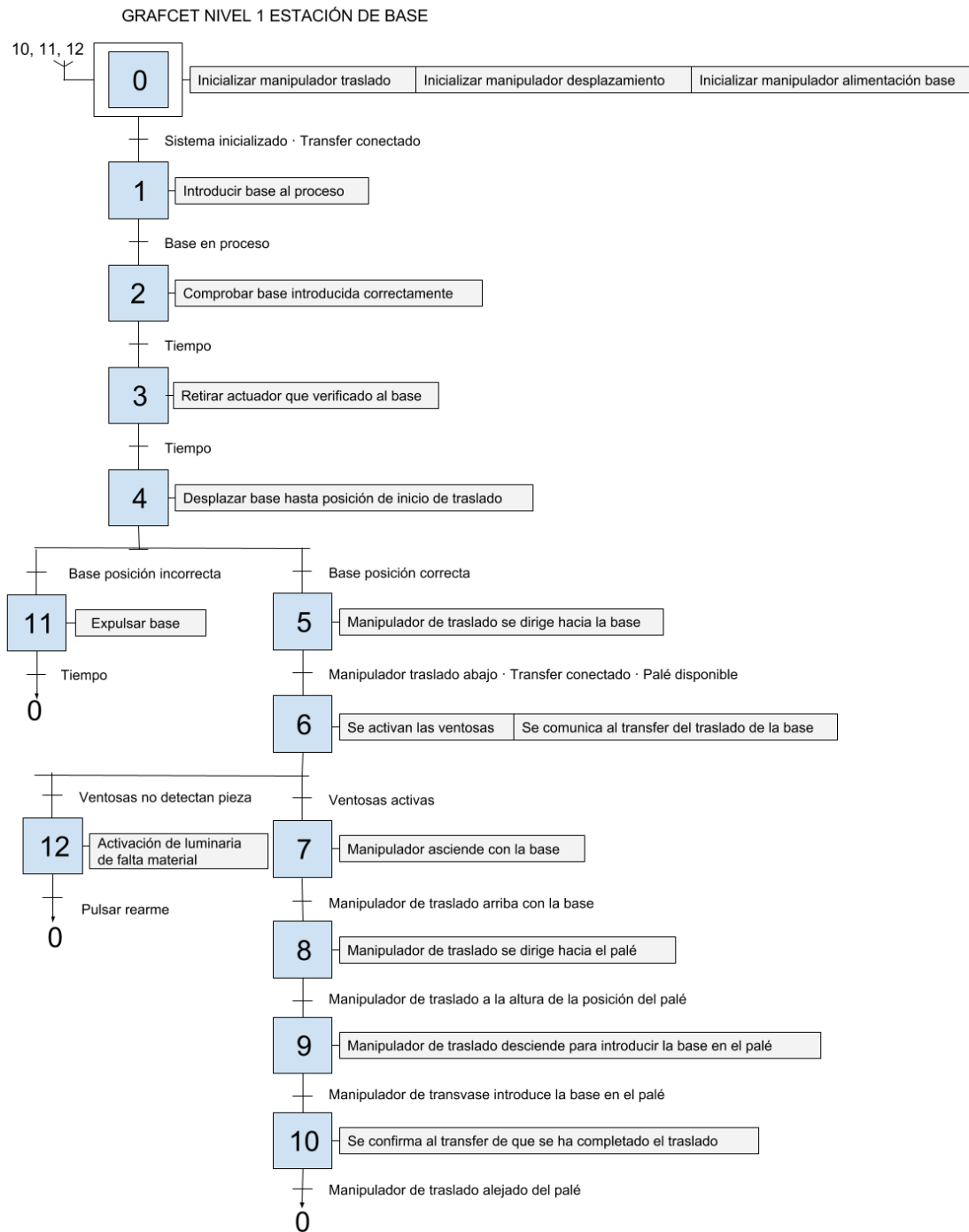


Figura A.13: GRAFCET Nivel 1 Estación de bases

A.2.10 GRAFCET nivel 3 de la Estación de bases

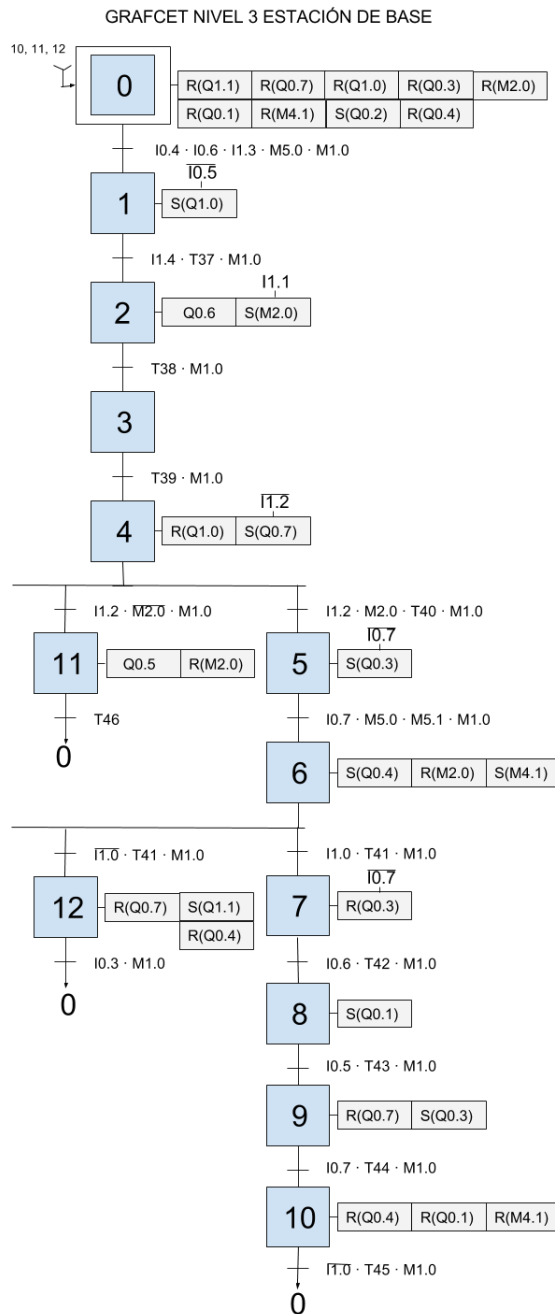


Figura A.14: GRAFCET Nivel 3 Estación de bases

A.2.11 GRAFCET nivel 1 de la Estación de rodamientos

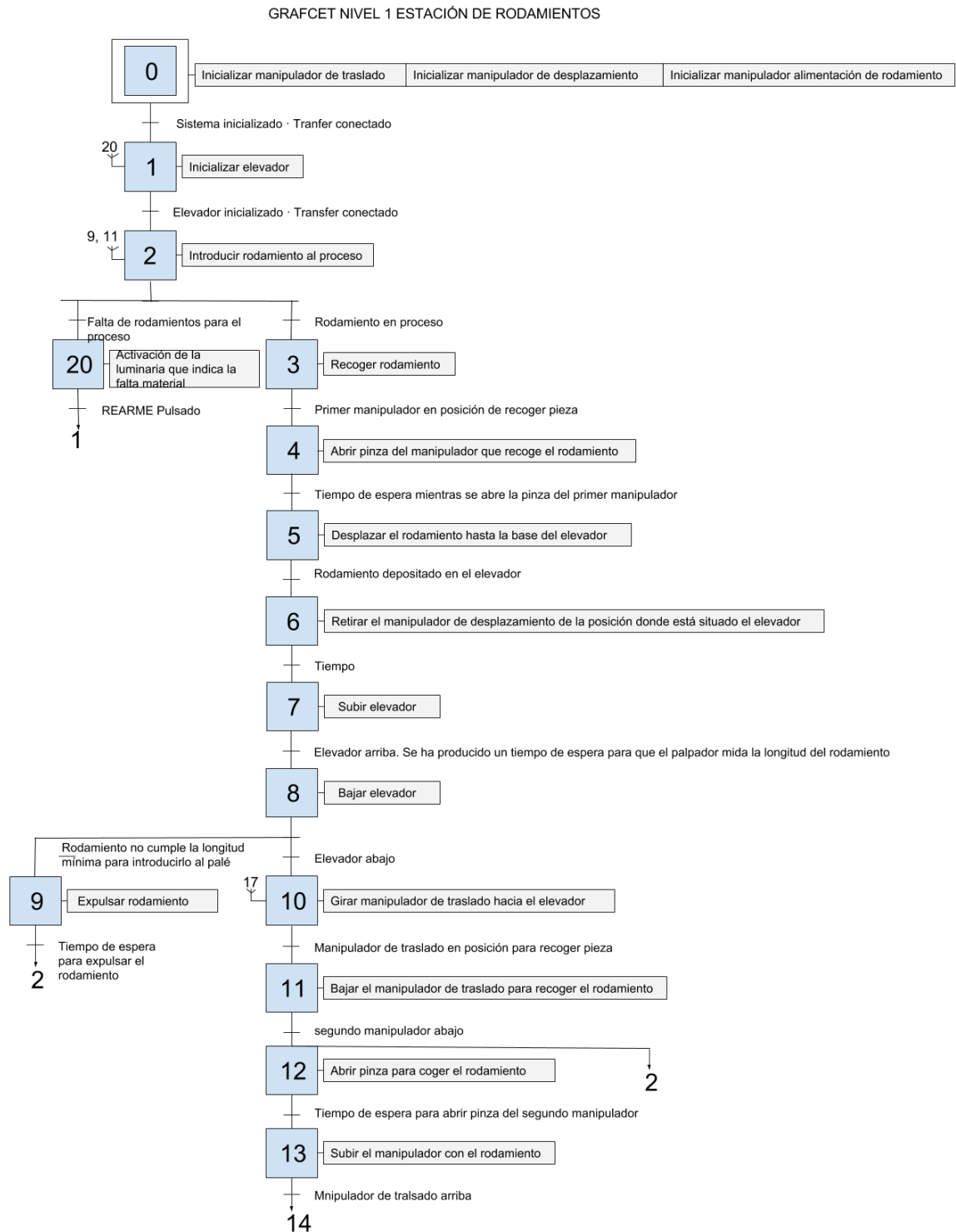


Figura A.15: GRAFCET Nivel 1 Estación de rodamientos

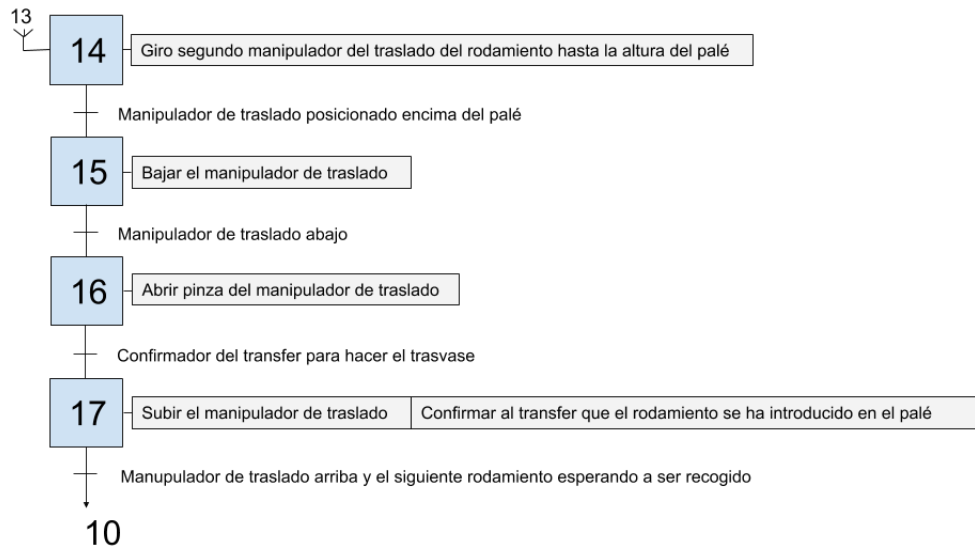


Figura A.16: GRAFCET Nivel 1 Estación de rodamientos

A.2.12 GRAFCET nivel 3 de la Estación de rodamientos

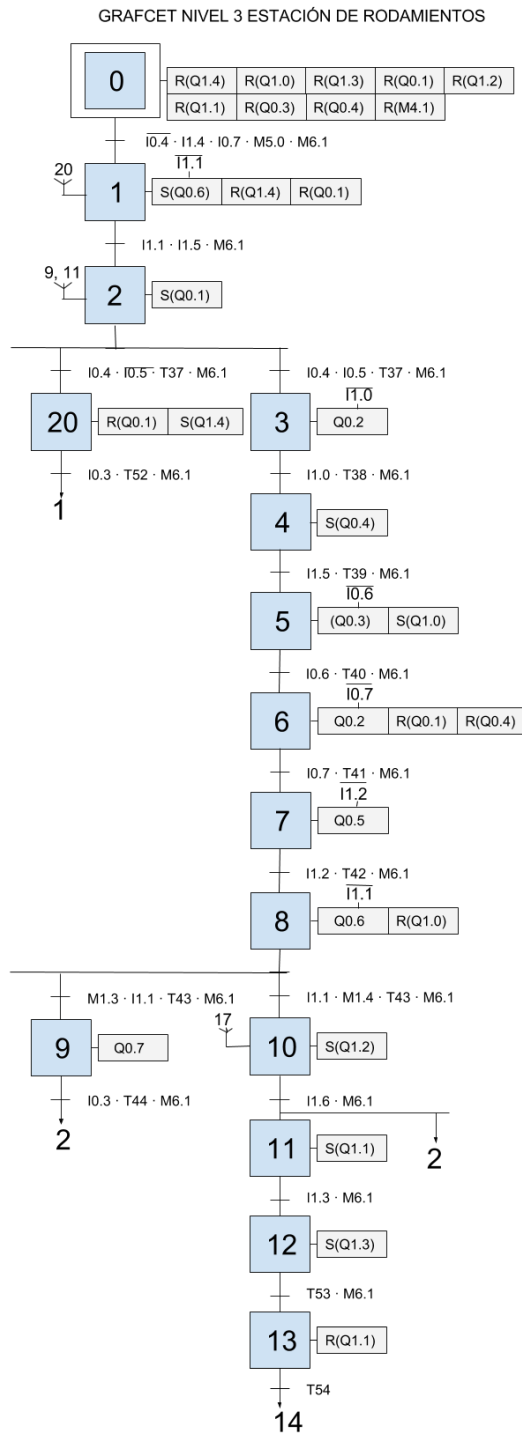


Figura A.17: GRAFCET Nivel 3 Estación de rodamientos

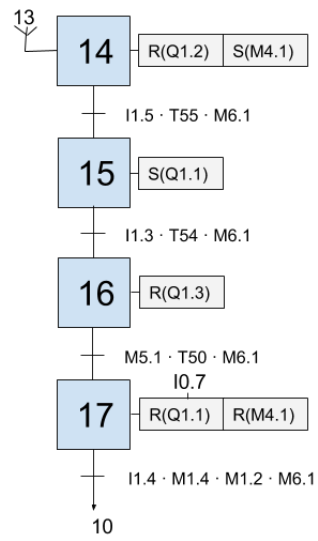


Figura A.18: GRAFCET Nivel 3 Estación de rodamientos

A.2.13 GRAFCET nivel 1 de la Estación de tapas

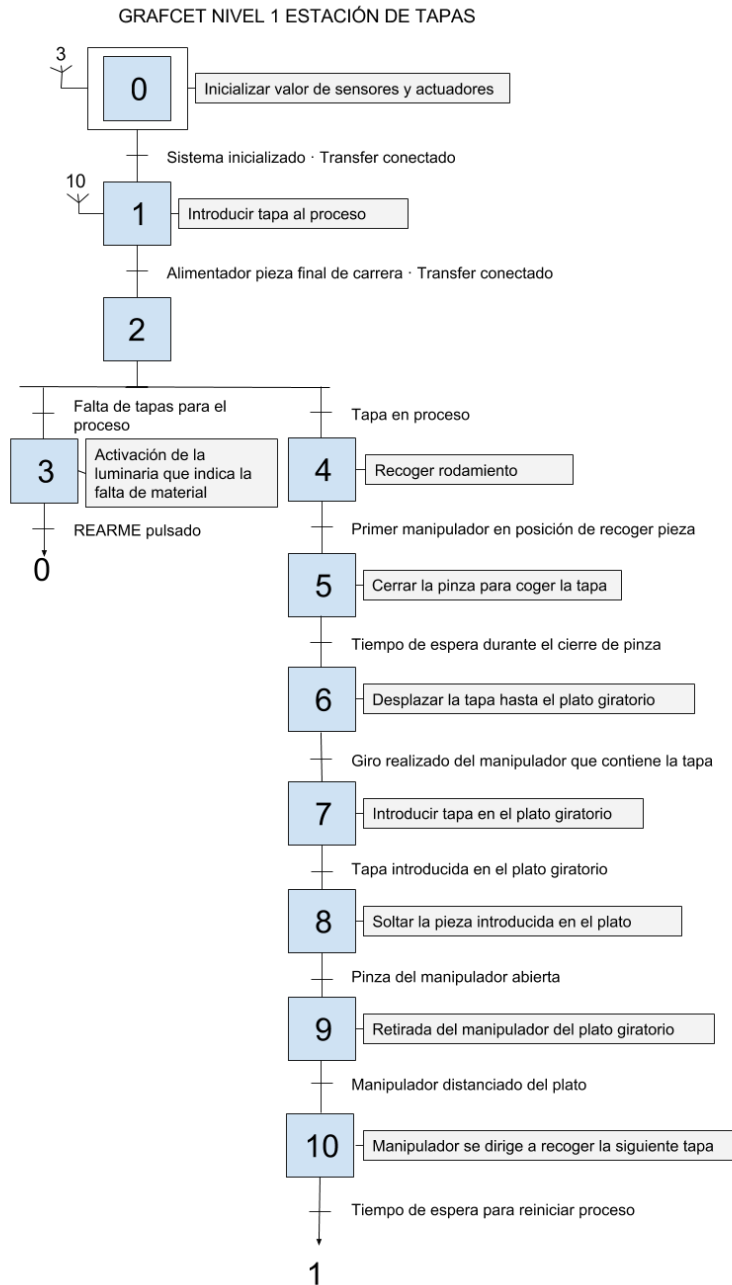


Figura A.19: GRAFCET Nivel 1 Estación de tapas

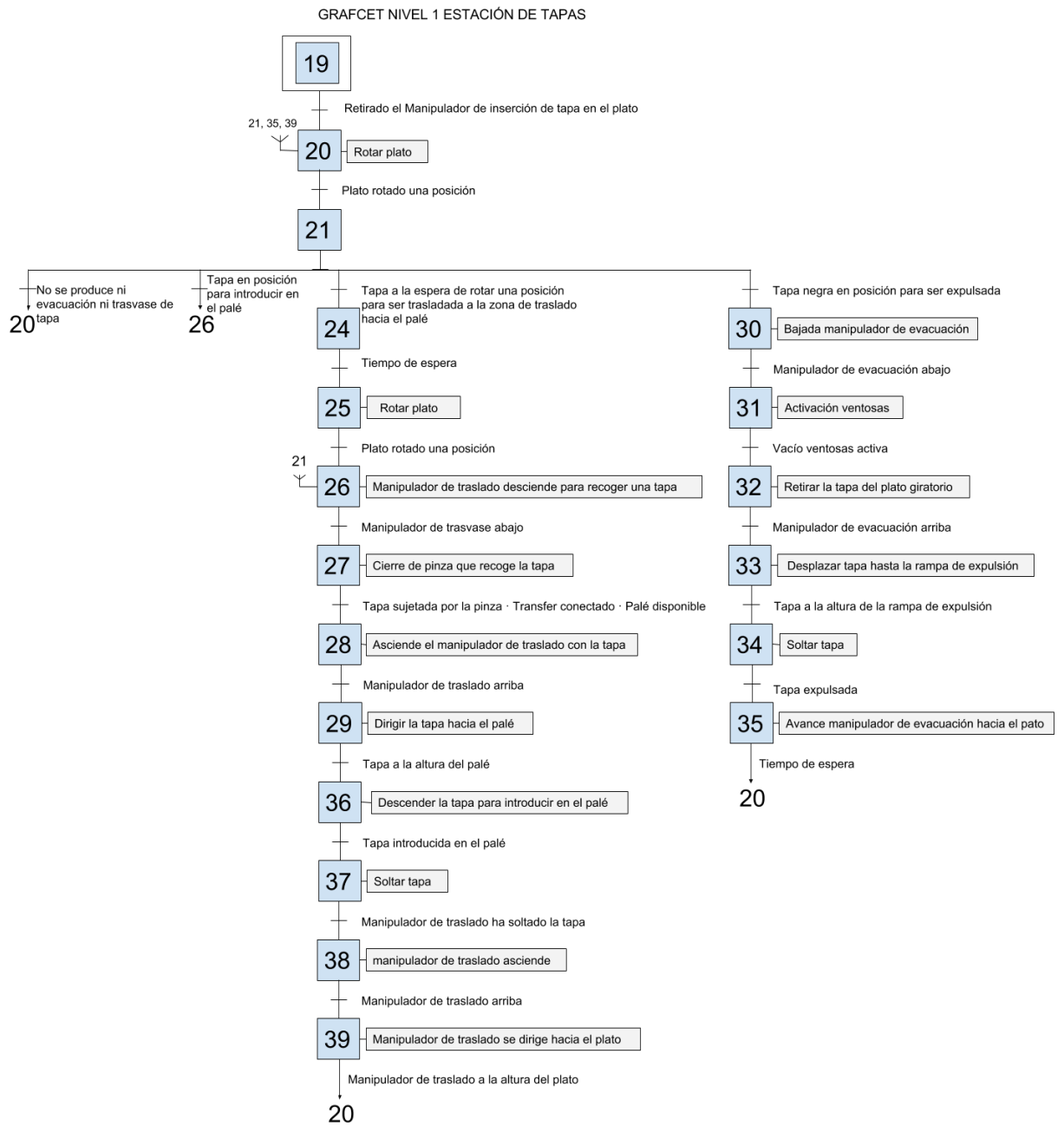


Figura A.20: GRAFCET Nivel 1 Estación de tapas

A.2.14 GRAFCET nivel 3 de la Estación de tapas

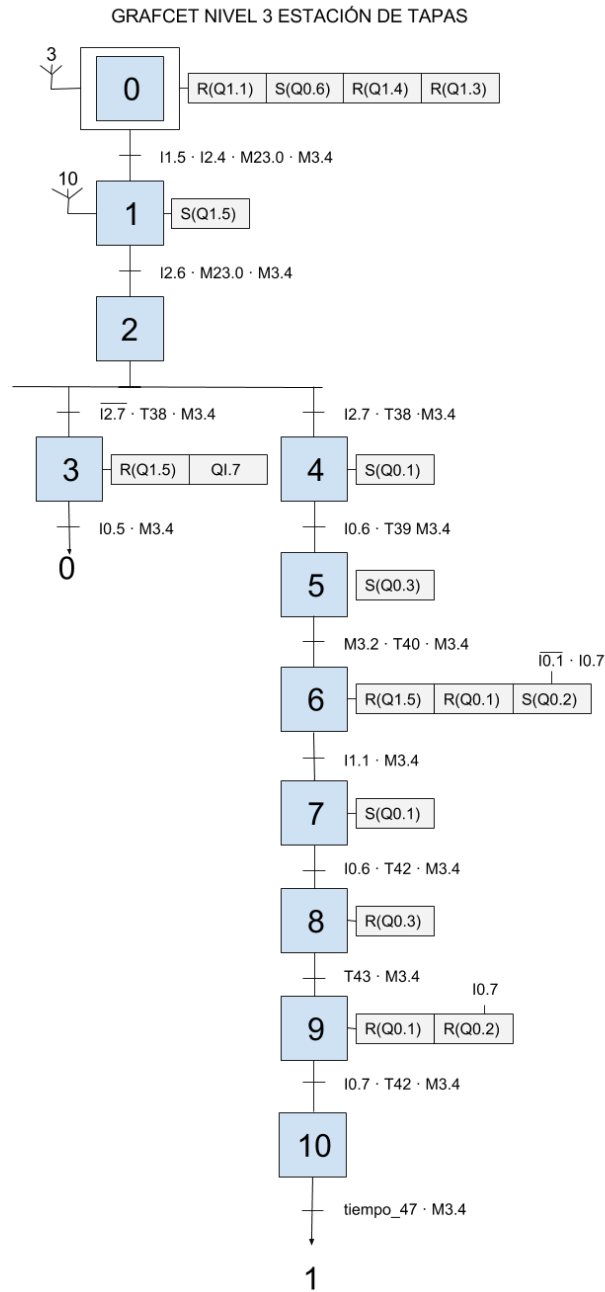


Figura A.21: GRAFCET Nivel 3 Estación de tapas

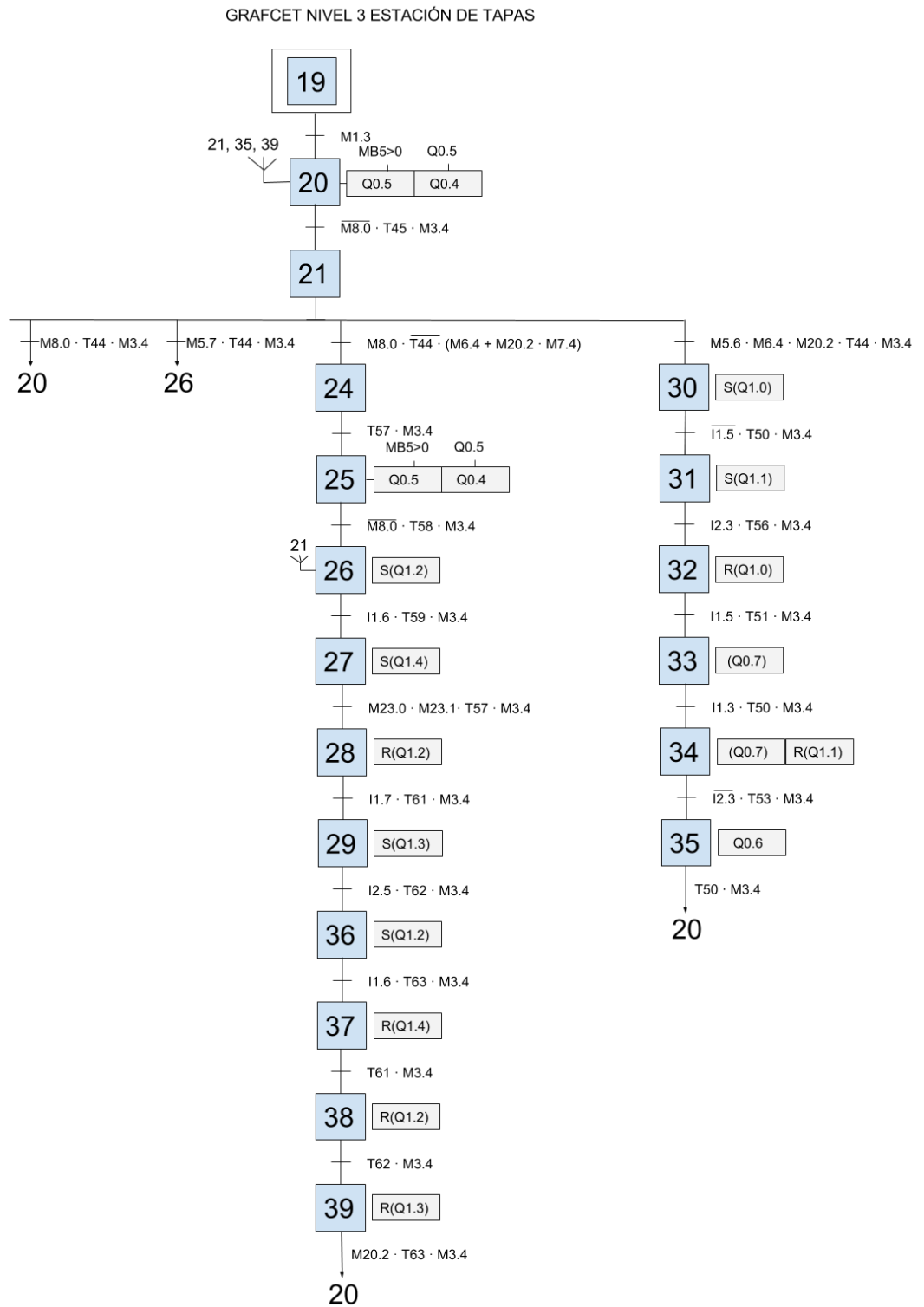


Figura A.22: GRAFCET Nivel 3 Estación de tapas

A.2.15 GRAFCET nivel 1 de la Estación de almacenamiento

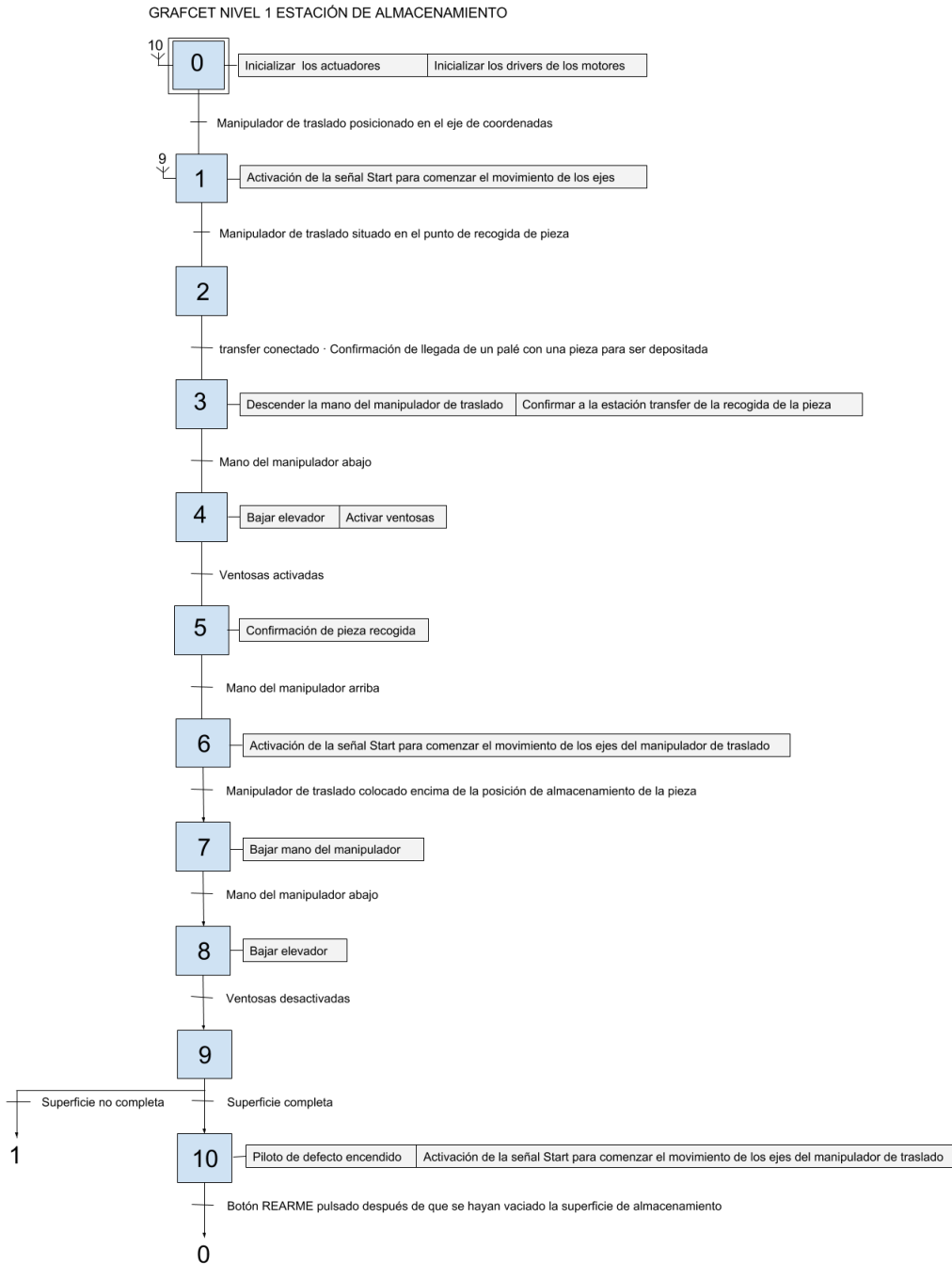


Figura A.23: GRAFCET Nivel 1 Estación de almacenamiento

A.2.16 GRAFCET nivel 3 de la Estación de almacenamiento

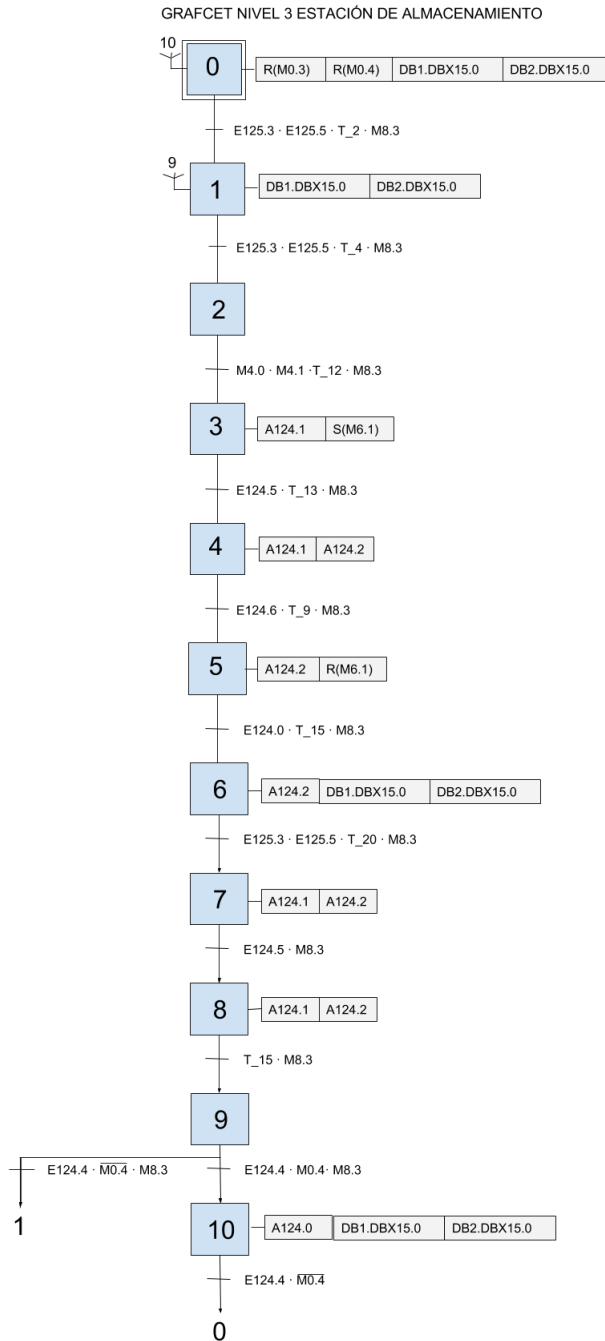


Figura A.24: GRAFCET Nivel 3 Estación de almacenamiento

BIBLIOGRAFÍA

- [1] E. M. Pérez, J. M. Acevedo, and C. F. Silva, *Autómatas programables y sistemas de automatización*. Marcombo, 2009. 1.1, 2.6
- [2] S. Simatic, “Manual del sistema de automatización s7-200,” *Siemens AG*, 1998. 2.1, 2.1
- [3] Siemens, “Hoja de datos logo!power 24 v/1,3 a,” 2017. [Online]. Available: <http://www.tme.eu/es/Document/ff7eba7e4bbaca81a2c7bb2149a3bffe/6EP1331-1SH03.pdf> 2.2
- [4] —, “Autómata programable s7-300 funciones integradas cpu 312 ifm/314 ifm,” *Siemens AG*, 1996. [Online]. Available: https://cache.industry.siemens.com/dl/files/951/1086951/att_96459/v1/S73if_s.pdf 2.2
- [5] —, “S7-300 sistema de automatización s7-300 datos de los módulos,” *Siemens AG*, 2013. 2.3, 2.4
- [6] —, “S7-300 cpu 31xc y cpu 31x: Configuración,” *Siemens AG*, 2008. [Online]. Available: <https://w5.siemens.com/spain/web/es/industry/automatizacion/simatic/Documents/S7300-FIRMWARE.PDF> 2.3
- [7] —, “Simatic net as-interface - introducción y fundamentos,” *Siemens AG*, 1999. [Online]. Available: <http://linux0.unsl.edu.ar/~rvilla/c3m10/apt01.pdf> 2.4, 2.4.1, 2.4.2
- [8] —, “Product data sheet 3rk1200-0dq00-0aa3,” *Siemens AG*, 2008. [Online]. Available: https://www.sistemamid.com/panel/uploads/biblioteca/1/349/1259/1260/1261/5124/5126/5127/5176/5178/Datasheet_3RK1200-0DQ00-0AA3_EN.pdf 2.4.2
- [9] —, “Data sheet power supply unit 2,4 a,” *Siemens AG*, 2017. [Online]. Available: <https://mall.industry.siemens.com/mall/en/uk/Catalog/Product/3RX9307-0AA00> 2.7
- [10] A. SIEMENS, “Simatic programar con step 7 manual,” 2006. [Online]. Available: https://cache.industry.siemens.com/dl/files/056/18652056/att_70833/v1/S7prv54_s.pdf 2.5, 2.5.6
- [11] —, “Software de sistema para s7-300 y s7-400 - funciones estándar y funciones de sistema tomo 1/2,” 2010. [Online]. Available: https://cache.industry.siemens.com/dl/files/604/44240604/att_67020/v1/s7sfc__d.pdf 2.5.3, 2.6.1, 2.6.1, 2.6.2, 2.6.2

BIBLIOGRAFÍA

- [12] A. Burguera, “La guía gemma,” UIB, Tech. Rep., 2012. **3.2.8**
- [13] O. Boix, “Descripción de la guía gemma.” [Online]. Available: <http://recursos.citcea.upc.edu/grafcet/gemma/descrip.html> **4.1, A.1, A.3**
- [14] A. SIEMENS, “Fm 353 stepper drive positioning module,” 2008. [Online]. Available: http://www.efesotomasyon.com/html/siemens/FM353_0808_b.pdf **5.2.1**