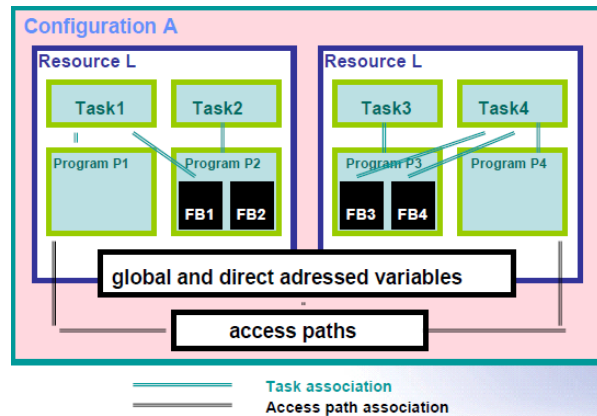
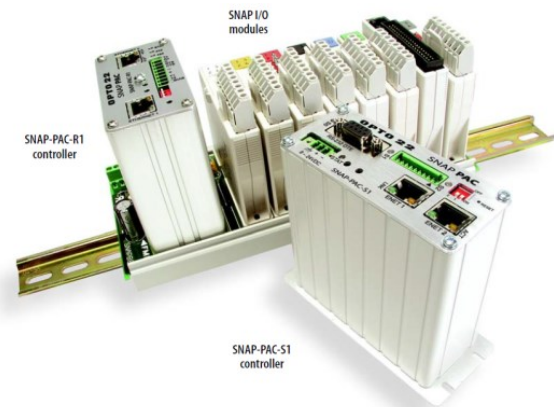


Unidad 3

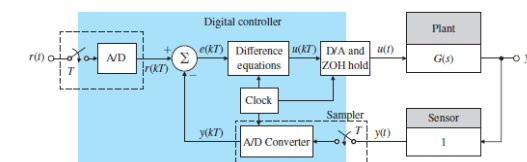
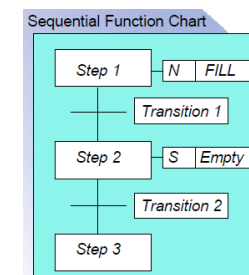
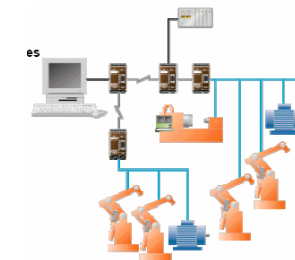
Diseño de Sistemas de Control Secuencial con Autómatas Programables



AUTÓMATAS y CONTROL DISCRETO

Unidades Temáticas:

- 1. Autómatas, Sistemas Discretos y Sistemas Híbridos**
Aplicaciones y Fundamentos. FSM. STATECHARTS. PETRI Nets. GRAFCET. Sistemas Híbridos.
- 2. Autómatas Programables de Control – IEC 61131: Arquitecturas y Lenguajes**
PLCs, arquitecturas. Seguridad Funcional. Entornos de desarrollo. Configuración. Lenguajes ST/ SFC.
- 3. Diseño de Sistemas de Control Secuencial con Autómatas Programables**
Métodos clásicos. Métodos sistemáticos y modernos para sistemas complejos. Aplicación.
- 4. Arquitecturas Distribuidas de Control y Comunicaciones Industriales:**
Arquitecturas distribuidas. Buses de Campo, Redes y Protocolos industriales de comunicación.
- 5. Control Discreto de Sistemas Continuos, Control Híbrido**
Sistemas muestreados. Simulación y Control. Control y Simulación Híbridos. Aplicación.



AUTÓMATAS y CONTROL DISCRETO

3. Diseño de Sistemas de Control Secuencial

A. Métodos Clásicos para Sistemas Simples

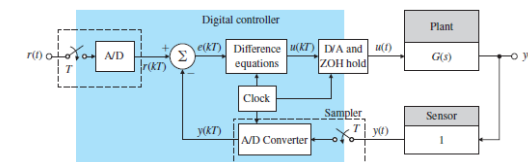
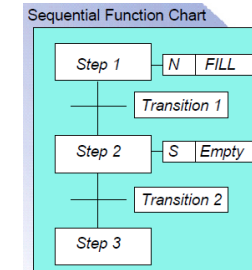
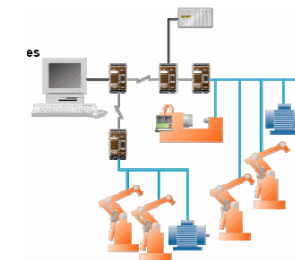
- Proyectos de Automatización. Entornos de Desarrollo (IDE) y Métodos de Diseño.
- Métodos de Diseño Clásicos. Sistemas Combinacionales. Sistemas Secuenciales de Control: Emulación de Biestables S-R. Emulación del Diagrama de Transición de Estados.

B. Métodos Sistemáticos y Modernos para Sistemas Complejos

- Descomposición Jerárquica.
- Partición de Algoritmo en Fases.
- SFC (GRAFCET): Diagramas de Función Secuenciales.

C. Aplicaciones. Buenas Prácticas de Programación y Configuración

- Uso de Entornos Integrados de Desarrollo (IDE). Creación de un Proyecto.
- Configuración de Hardware. Estructuración de Software.
- Uso de CODESYS, TIA Portal, etc. Recomendaciones: PLCopen, etc.
- Ejemplos y Ejercicios de Aplicación.



Diseño de Sistemas de Control Secuencial

*basados en **Autómatas Programables (PLCs, PACs, etc.)** →*

Combinar:

*a. Utilización de **Herramientas o Entornos de Desarrollo:***

*Herramientas de Diseño Asistido por Computador (**CAD/CAE Tools**) →*

*→ Entornos integrados de desarrollo (**IDE**, Integrated Development Environments).*

Gestión o administración de información →

*Facilitan y organizan tareas de diseño: **Proyecto de Automatización** (configuración de Hardware y Software).*

*Suministradas por cada fabricante (ej. **TIA Portal/Step 7** [Siemens], **PC Worx** [Phoenix Contact], etc.) vs. genéricas (**CODESYS**, **OpenPCS**).*

*b. Aplicación de **Métodos de Diseño:***

***Especificaciones de Requisitos** →*

*→ Desarrollo del **Programa de Aplicación de Control y Automatización***

*Uso de distintos **Lenguajes de Programación y Elementos Comunes IEC 61131-3**;*

*aplicando **Métodos Clásicos** vs. **Métodos Sistemáticos**.*

Etapas básicas de Proyecto de Automatización



Figura 5.1. *Etapas del proyecto de un sistema electrónico de control basado en autómatas programables.*

Entornos de Desarrollo, IDE (ej. STEP7)

Configuración de Hardware:

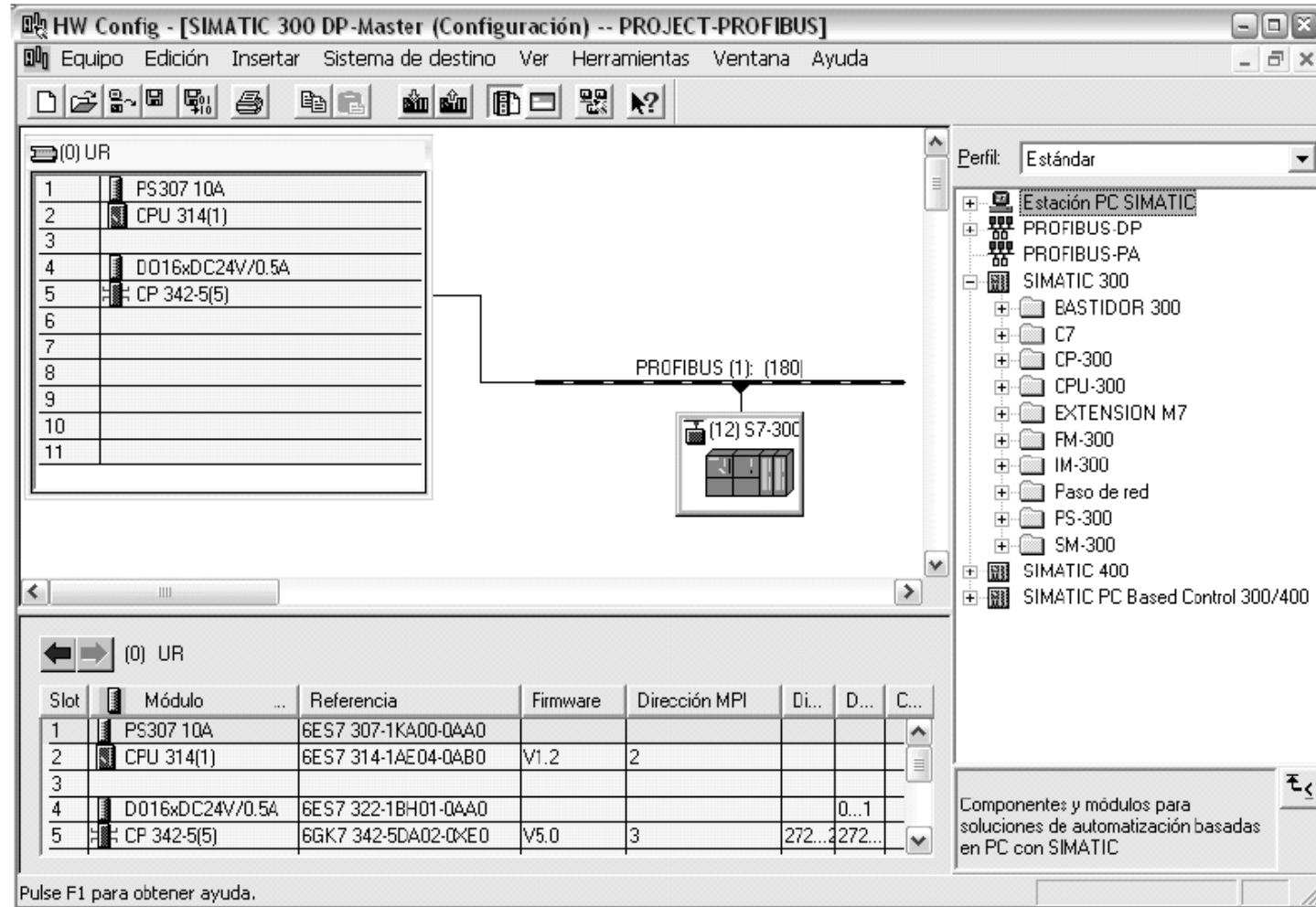


Figura 5.2. Programa configurador de hardware de STEP7.

Ciclo de Operación básica (ej. S7-300, etc.)

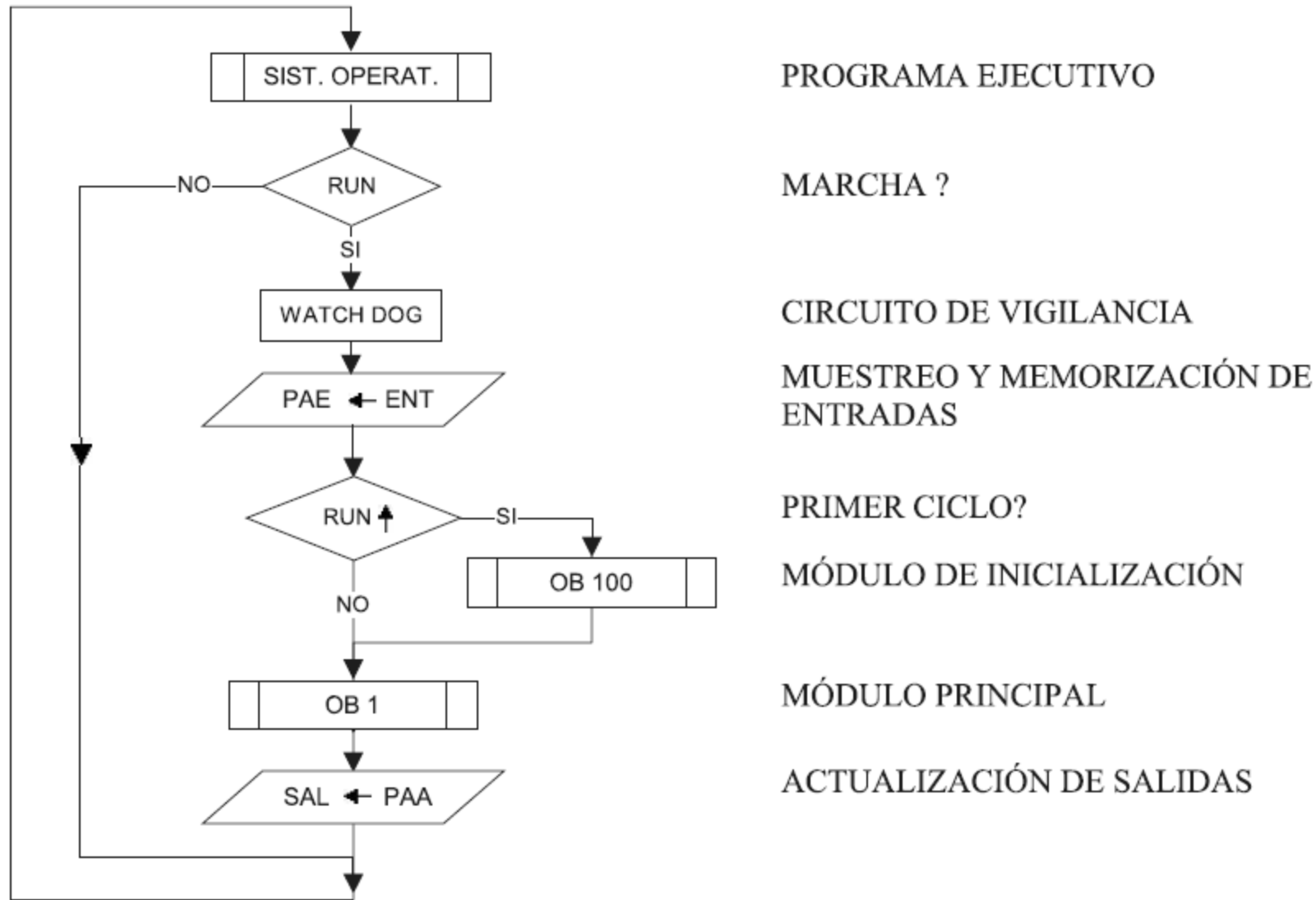


Figura 5.3. Organigrama del funcionamiento cíclico de un autómata programable S7.

1. Emulación de biest. S-R (prueba-error)

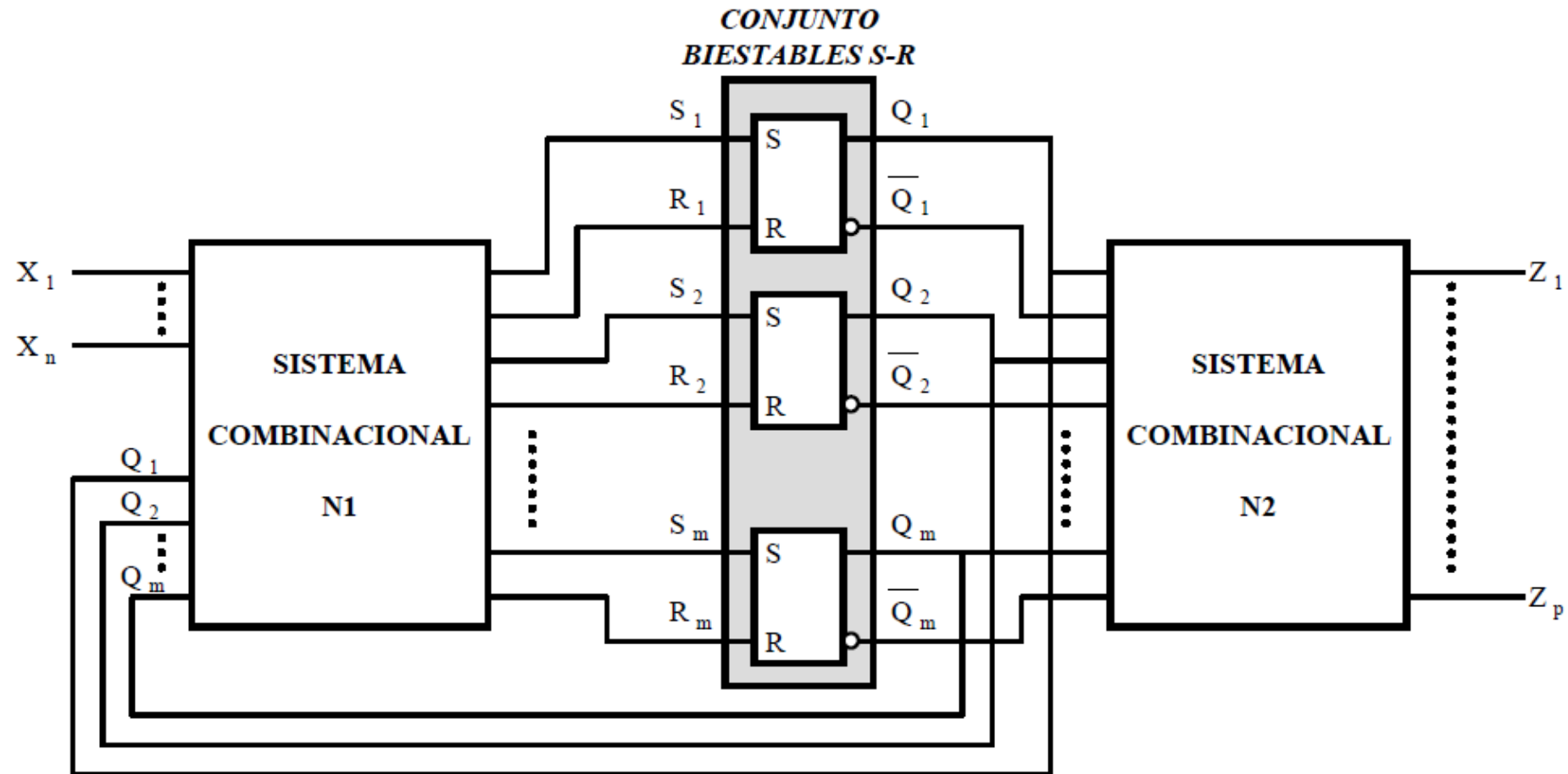


Figura 5.7. Diagrama de bloques de un sistema secuencial asíncrono realizado con biestables R-S.

1. Emulación de biest. S-R (prueba-error)

- Prueba y error (modificación paulatina)
- Distintas alternativas (falta de sistematización)
- Concebido originalmente para lenguaje **Ladder** (esquemas de contactos)

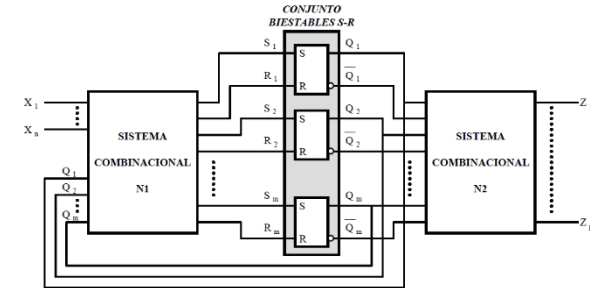


Figura 5.7. Diagrama de bloques de un sistema secuencial asincrónico realizado con biestables R-S.

a. Diseño orientado hacia las **variables de salida**:

- Si una variable de salida depende solo de valor **instantáneo** actual de entradas → lógica **combinacional** (“paradigma de Mealy” parcial);
- Si una variable de salida depende del valor o **secuencia** de valores de entradas **anteriores** → **biestable SR de salida** (“paradigma de Moore” parcial);
- Estados internos adicionales → **biestable SR interno** para memorizar secuencias que finalmente activan biestables SR de salida.

b. Diseño orientado hacia las **variables de estado interno**:

- Especificaciones → Definir **variables de estado interno** (discreto) para memorizar secuencias de variables de entrada → Cada estado interno: **biestable SR interno**;
- **Variables de salida**: a partir de variables de estado interno y/o de variables de entrada (paradigma de Moore vs. Mealy).

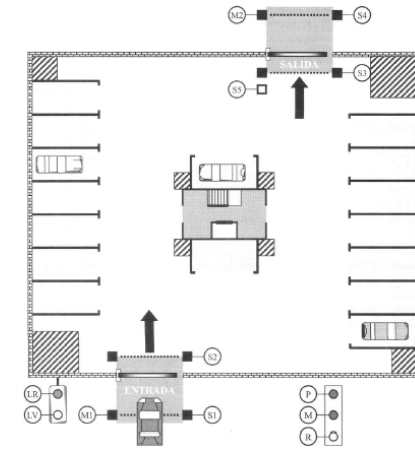
1. Emulación de biest. S-R: Práctica

a. Diseño orientado hacia las **variables de salida**:

- Mandado Autómatas Programables, p. 295-304

Ejemplo 5.3:

Control de Cochera de Estacionamiento.

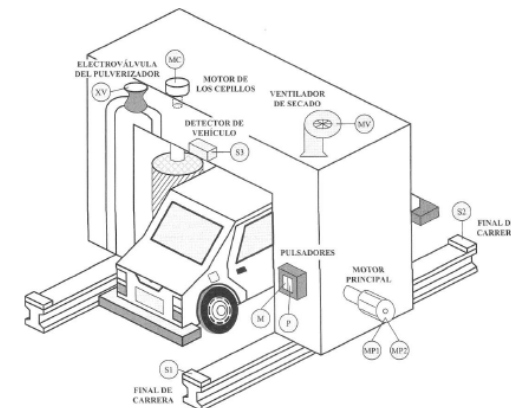


b. Diseño orientado hacia las **variables de estado interno**:

- Mandado Autómatas Programables, p. 304-309

Ejemplo 5.4:

Control de Máq. Lavadora de autos.



Práctica: Ej. 5.3 Control Automático de Cochera de Estacionam.

El garaje dispone de un acceso de entrada y otro de salida controlados por sendas barreras que se accionan mediante los motores eléctricos M1 y M2 respectivamente.

A ambos lados de las dos barreras se instalan sensores de presencia de vehículo: S1 y S2 en la de entrada y S3 y S4 en la de salida. Dichos sensores permanecen activados mientras hay un vehículo ante ellos y por su situación física nunca se activan simultáneamente S1 y S2 ni S3 y S4. Se utiliza un sensor S5 para detectar la presencia de ficha en el control de salida.

La capacidad del garaje es de 10 vehículos y el sistema electrónico debe controlar la ejecución de las siguientes acciones:

- *Apertura y cierre automático de las barreras*

La barrera de entrada debe abrirse si en el interior del garaje hay menos de 10 vehículos y se produce un flanco de subida (paso de desactivado a activado del sensor S1). Dicha barrera debe cerrarse si se produce un flanco de bajada (paso de activado a desactivado del sensor S2). La barrera de salida debe abrirse si se activa S5 (presencia de ficha) y se produce un flanco de subida en S3 y debe cerrarse al producirse un flanco de bajada en S4.

- *Señalización a la entrada, mediante una luz verde LV, de que existen plazas libres en el garaje*
- *Señalización a la entrada, mediante una luz roja LR, de que el garaje está completo y no pueden entrar más coches.*

El sistema de control debe poseer además los siguientes elementos de entrada:

- Un pulsador M para ponerlo en marcha. A partir del instante de dar tensión al automático no se permite la entrada o salida de vehículos hasta que no se accione este pulsador.
- Un pulsador de paro P para dejarlo fuera de servicio. Si se acciona este pulsador queda impedida la entrada y salida de vehículos hasta que se accione el pulsador M. En el caso de que P y M se accionen simultáneamente, el primero debe predominar sobre el segundo.
- Un pulsador R para poner a 10 el contador de vehículos en el instante de dar tensión al automático.

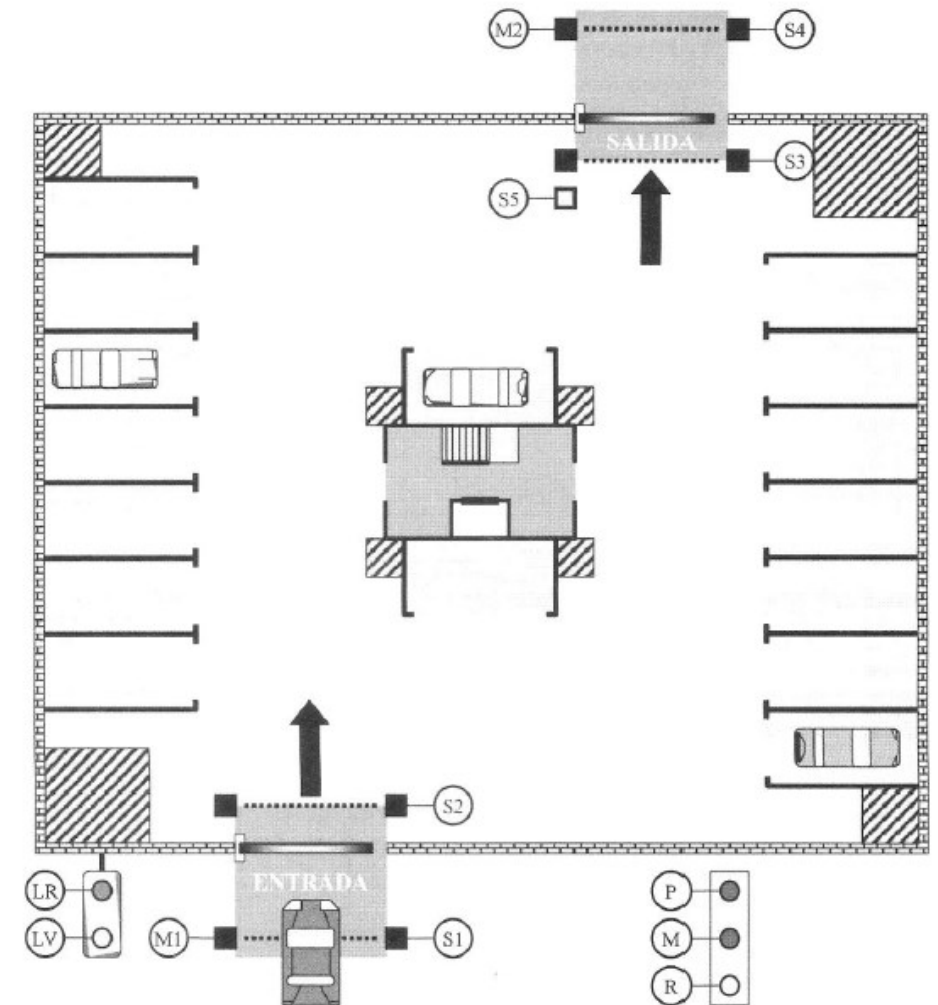
Hacer:

0. Emulación de biestables SR salida LADDER

1. Autómata: plantear **GRAFCET**

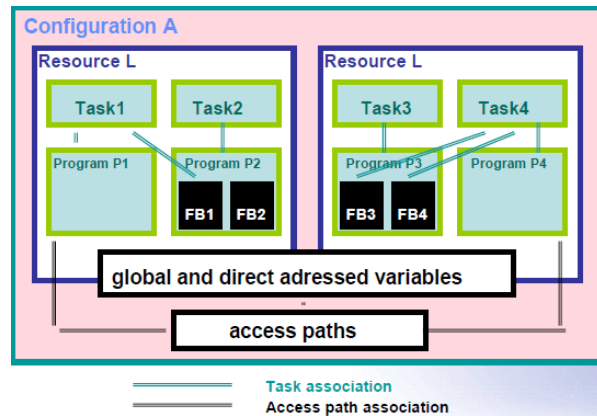
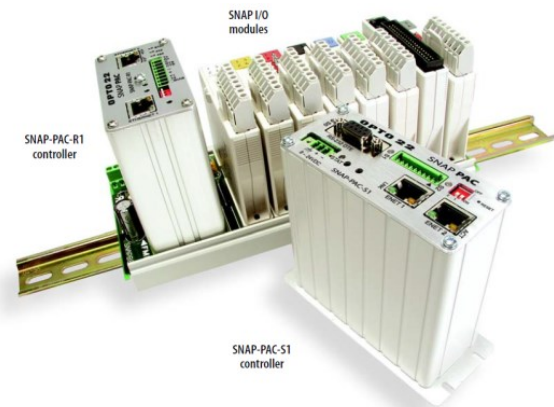
2. Partición en fases/etapas → **ST (CODESYS)**

3. Implementación automata directamente en **SFC (CODESYS)**



Unidad 3

Diseño de Sistemas de Control Secuencial con Autómatas Programables



AUTÓMATAS y CONTROL DISCRETO

3. Diseño de Sistemas de Control Secuencial

A. Métodos Clásicos para Sistemas Simples

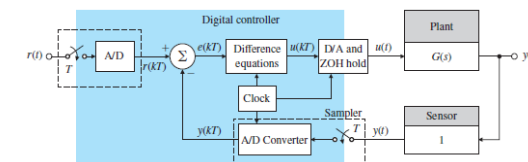
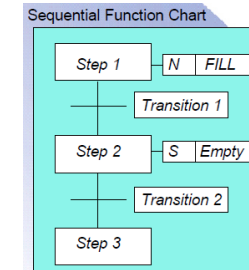
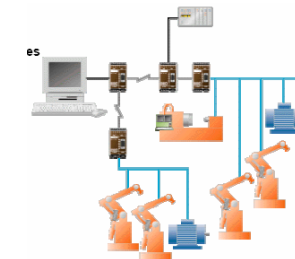
- Proyectos de Automatización. Entornos de Desarrollo (IDE) y Métodos de Diseño.
- Métodos de Diseño Clásicos. Sistemas Combinacionales. Sistemas Secuenciales de Control: Emulación de Biestables S-R. Emulación del Diagrama de Transición de Estados.

B. Métodos Sistemáticos y Modernos para Sistemas Complejos

- Descomposición Jerárquica.
- Partición de Algoritmo en Fases.
- SFC (GRAFCET): Diagramas de Función Secuenciales.

C. Aplicaciones. Buenas Prácticas de Programación y Configuración

- Uso de Entornos Integrados de Desarrollo (IDE). Creación de un Proyecto.
- Configuración de Hardware. Estructuración de Software.
- Uso de CODESYS, TIA Portal, etc. Recomendaciones: PLCopen, etc.
- Ejemplos y Ejercicios de Aplicación.



Diseño de Sistemas de Control Secuencial

*basados en **Autómatas Programables (PLCs, PACs, etc.)** →*

Combinar:

*a. Utilización de **Herramientas o Entornos de Desarrollo:***

*Herramientas de Diseño Asistido por Computador (**CAD/CAE Tools**) →*

*→ Entornos integrados de desarrollo (**IDE**, Integrated Development Environments).*

Gestión o administración de información →

*Facilitan y organizan tareas de diseño: **Proyecto de Automatización** (configuración de Hardware y Software).*

*Suministradas por cada fabricante (ej. **TIA Portal/Step 7** [Siemens], **PC Worx** [Phoenix Contact], etc.) vs. genéricas (**CODESYS**, **OpenPCS**).*

*b. Aplicación de **Métodos de Diseño:***

***Especificaciones de Requisitos** →*

*→ Desarrollo del **Programa de Aplicación de Control y Automatización***

*Uso de distintos **Lenguajes de Programación y Elementos Comunes IEC 61131-3**);*

*aplicando **Métodos Clásicos** vs. **Métodos Sistemáticos**.*

2. Emulación del Diagrama de Estados: a partir de GRAFCET

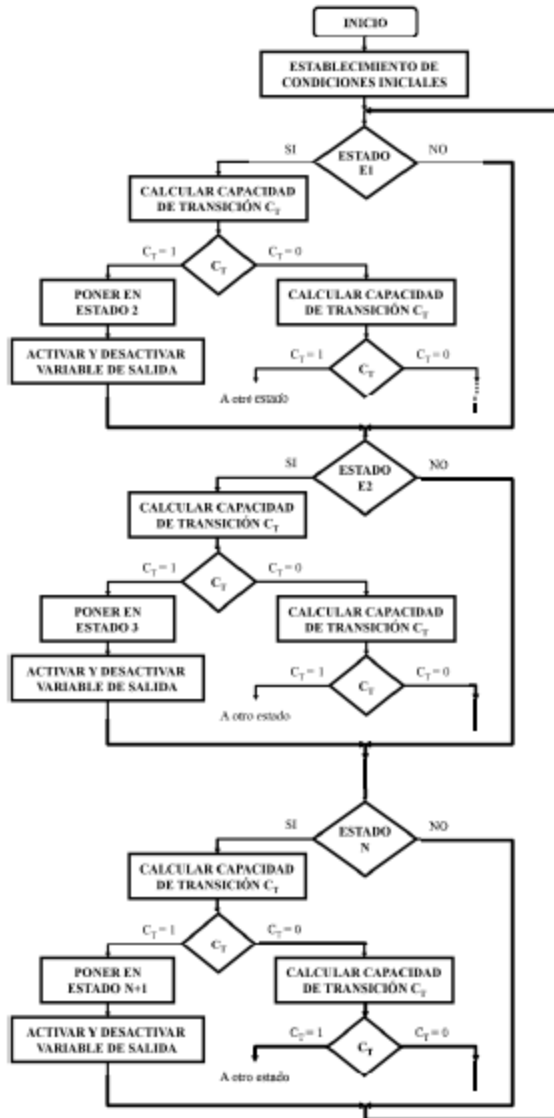


Figura 5.30. Algoritmo compacto básico de emulación de un diagrama de estados.

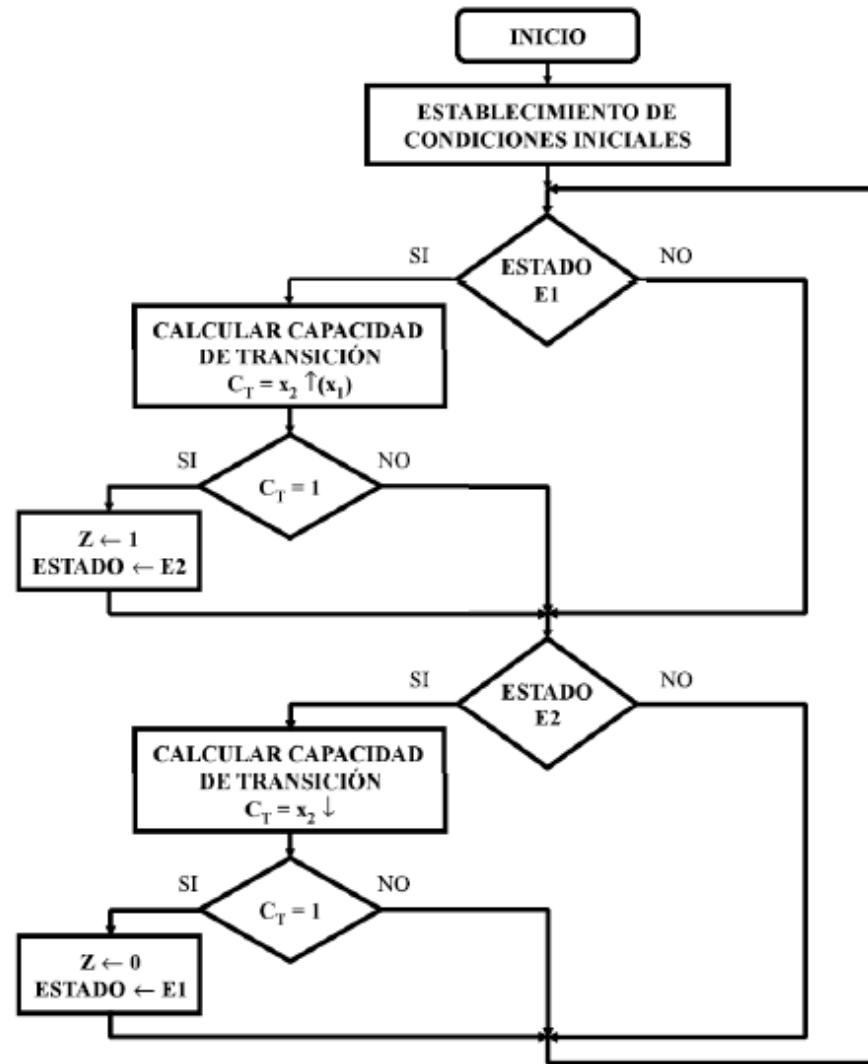


Figura 5.32. Algoritmo compacto que describe el comportamiento que debe tener el autómatas programable que controla el sistema de selección de barras de la figura 5.31.

1. Emulación de Diag. Estados : Práctica

a. Diseño por *Emulación de Diagrama de Estados*:

– Mandado Autómatas Programables, p. 311-314

Ejemplo 5.5:

Control para Selección de Barras de acuerdo a su longitud.

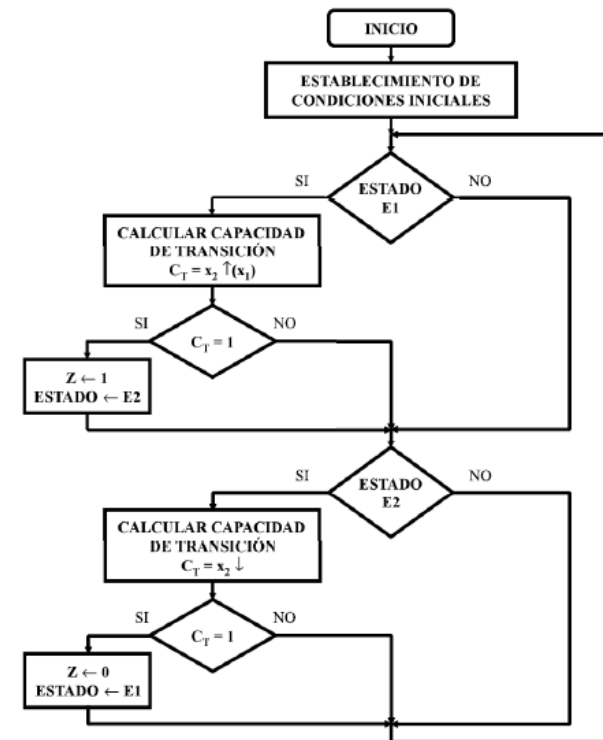
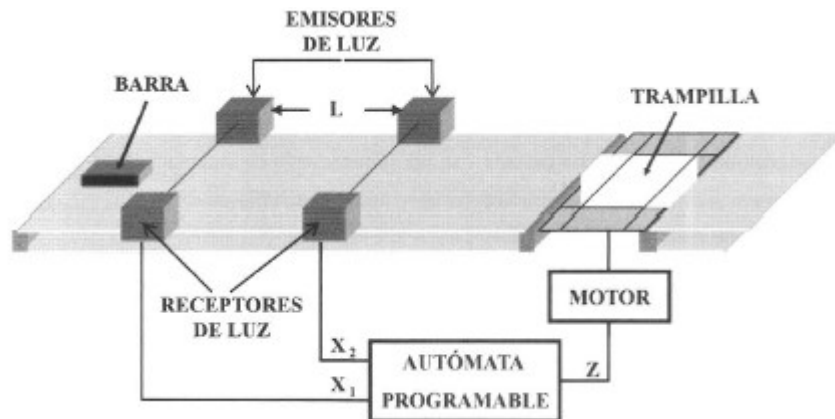


Figura 5.32. Algoritmo compacto que describe el comportamiento que debe tener el autómata programable que controla el sistema de selección de barras de la figura 5.31.

Modelo VDI 2221: Descomposición Jerárquica

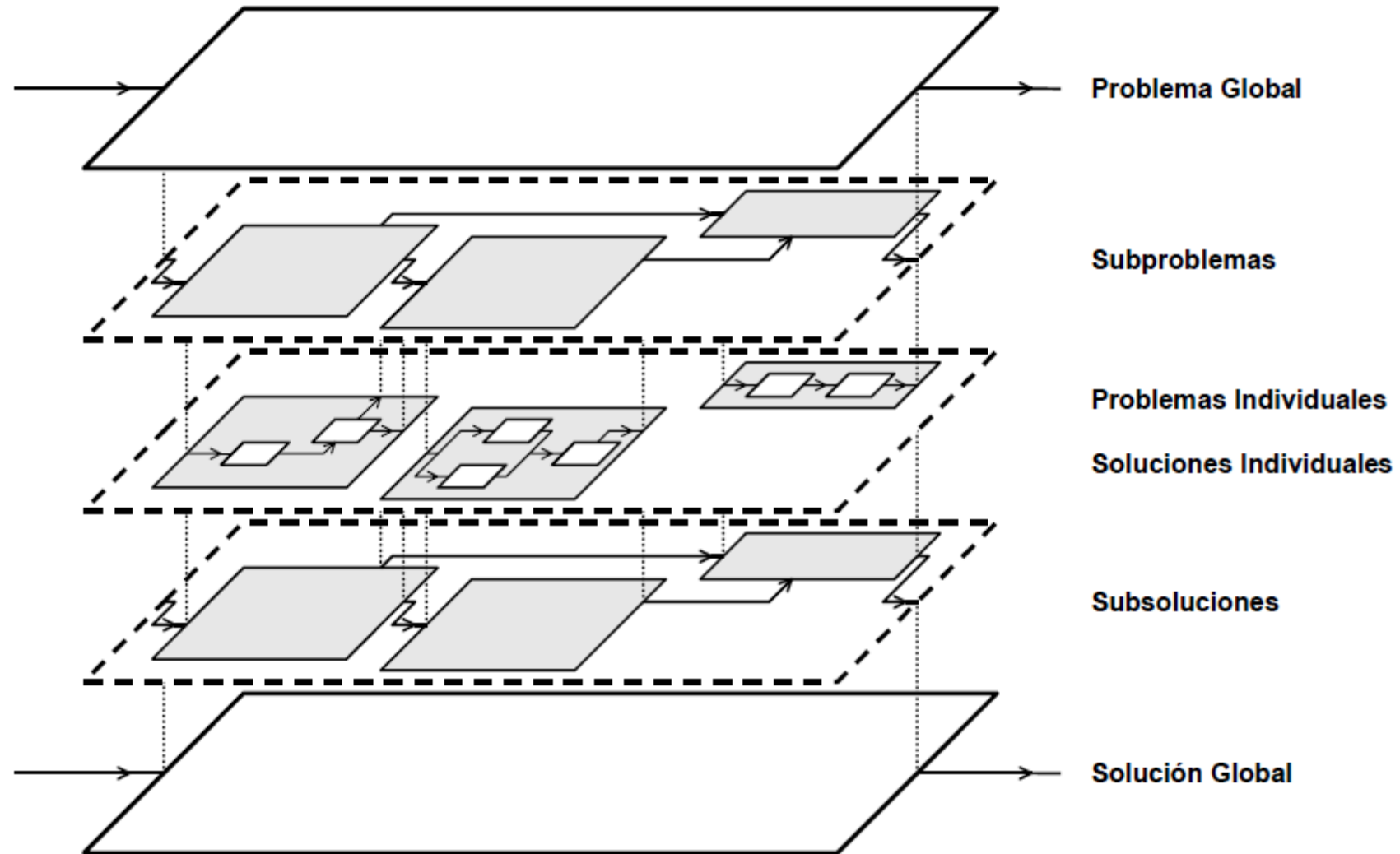


Figura 5.33. Descripción gráfica del modelo de diseño VDI 2221.

Partición del Algoritmo en Fases /Etapas

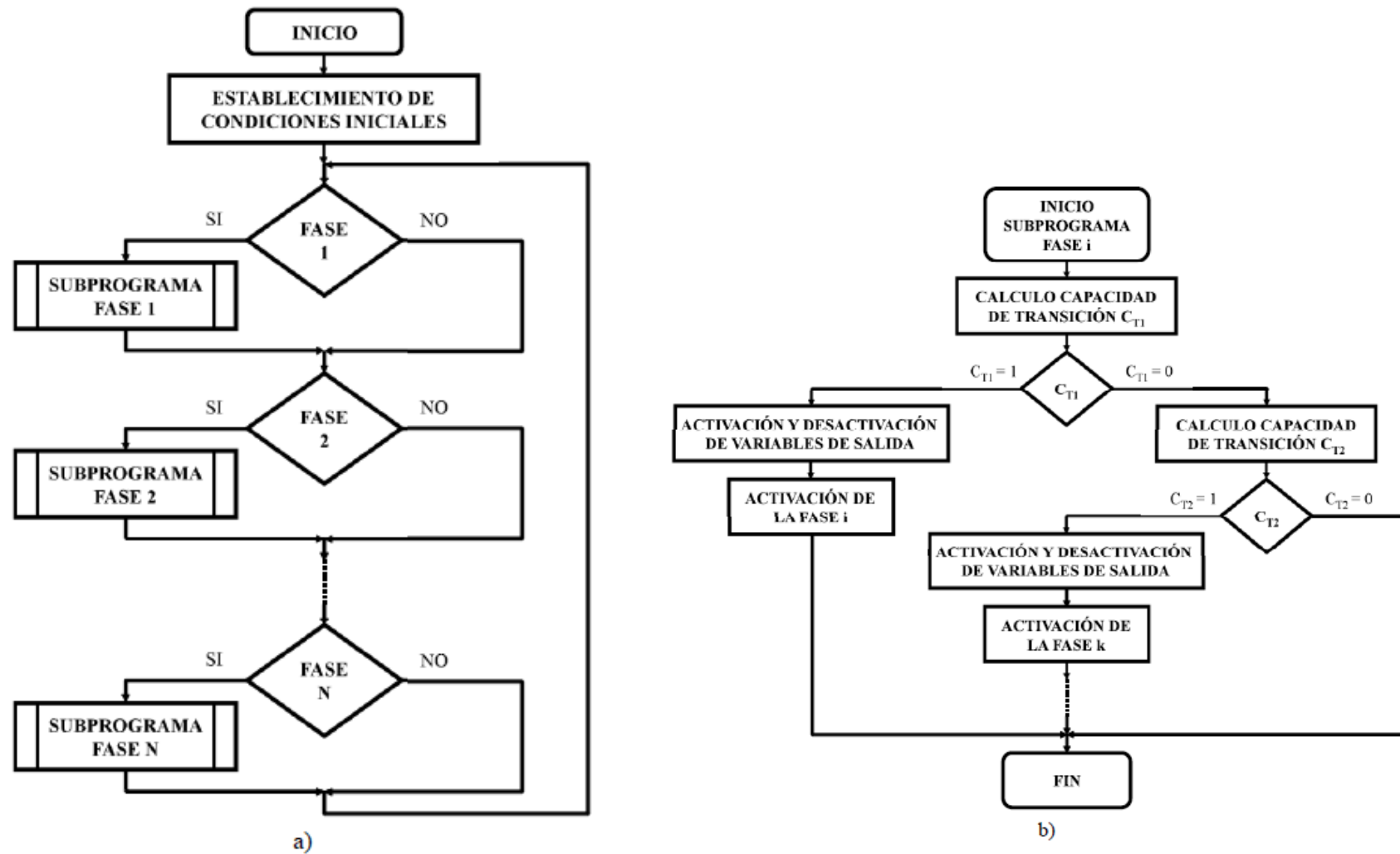


Figura 5.34. Algoritmo general del método de diseño basado en la partición del algoritmo en fases: a) Programa principal; b) Algoritmo de cada fase.

Sequential Function Chart (SFC/Grafcet)

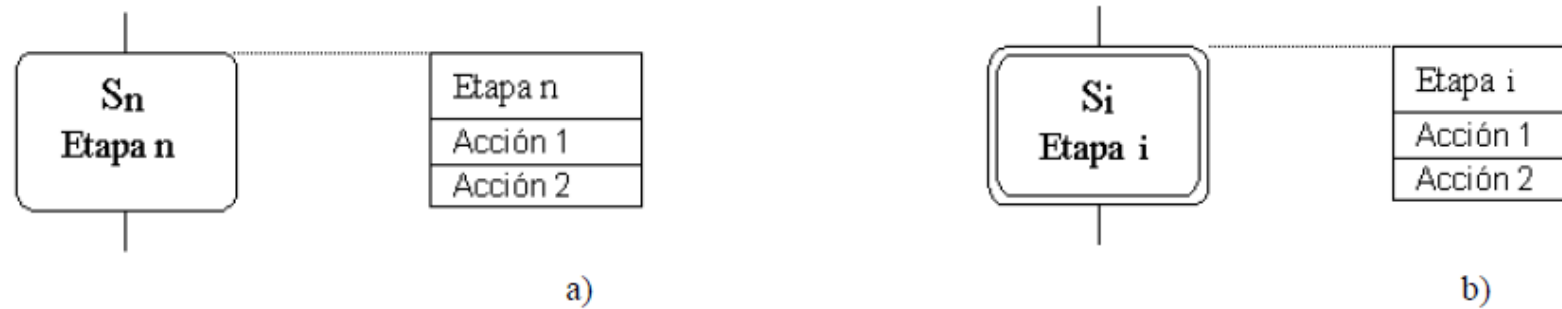


Figura 5.40. Representación gráfica en el lenguaje S7-GRAPH: a) Una etapa cualquiera; b) Una etapa inicial.

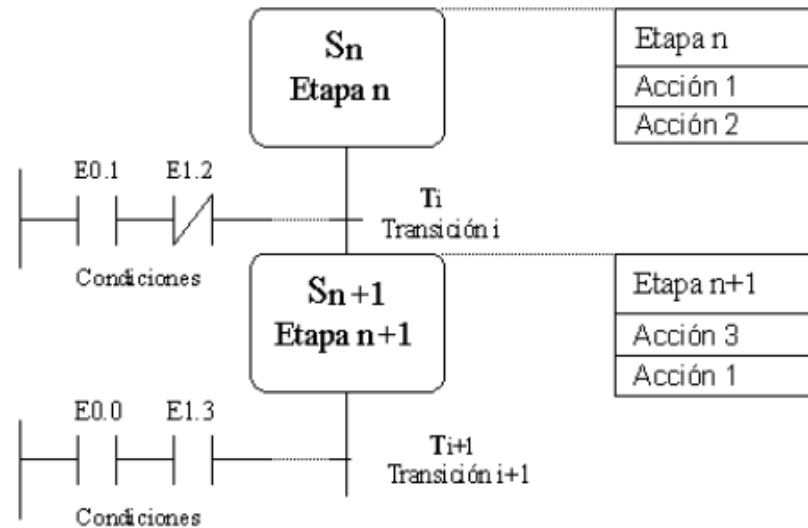


Figura 5.41. Representación gráfica en el lenguaje S7-GRAPH de las etapas y de las transiciones entre ellas.

Sequential Function Chart (SFC) Ejemplo

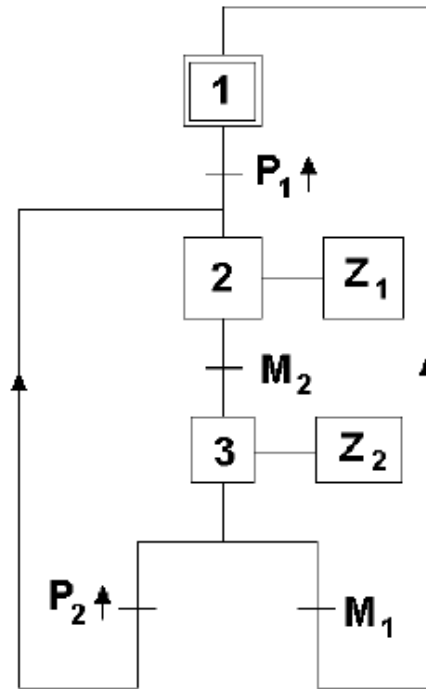


Figura 5.45. Diagrama SFC del ejemplo 5.9.

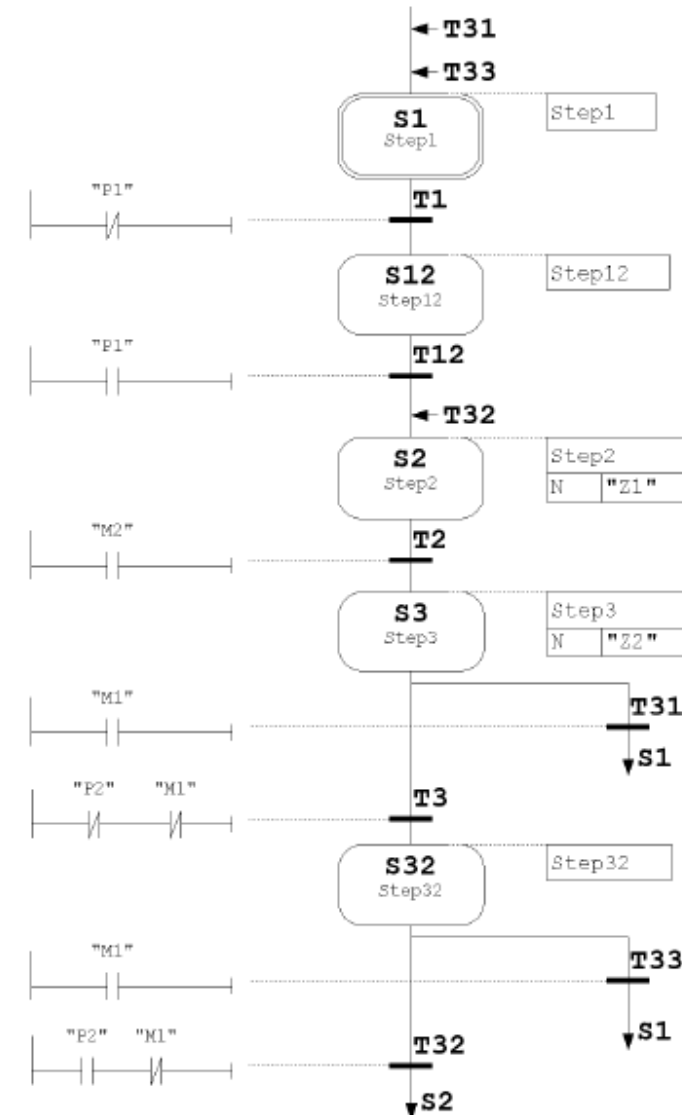


Figura 5.46. Programa en el lenguaje S7-GRAPH del ejemplo 5.9.

Sequential Function Chart (SFC) Errores

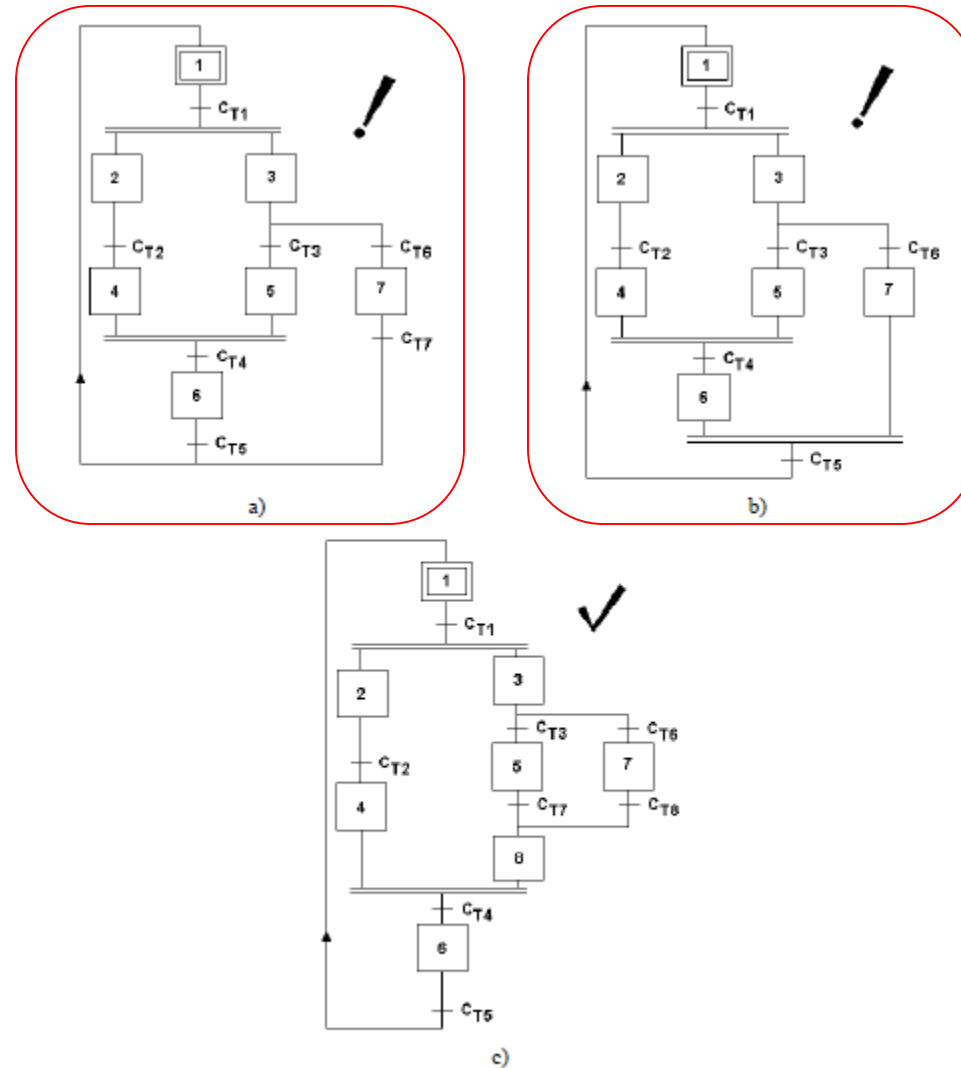


Figura 5.51. Errores de programación en un diagrama SFC/S7-GRAPH: a) Inseguro; b) Inalcanzable; c) Posible.

SFC Sincronización

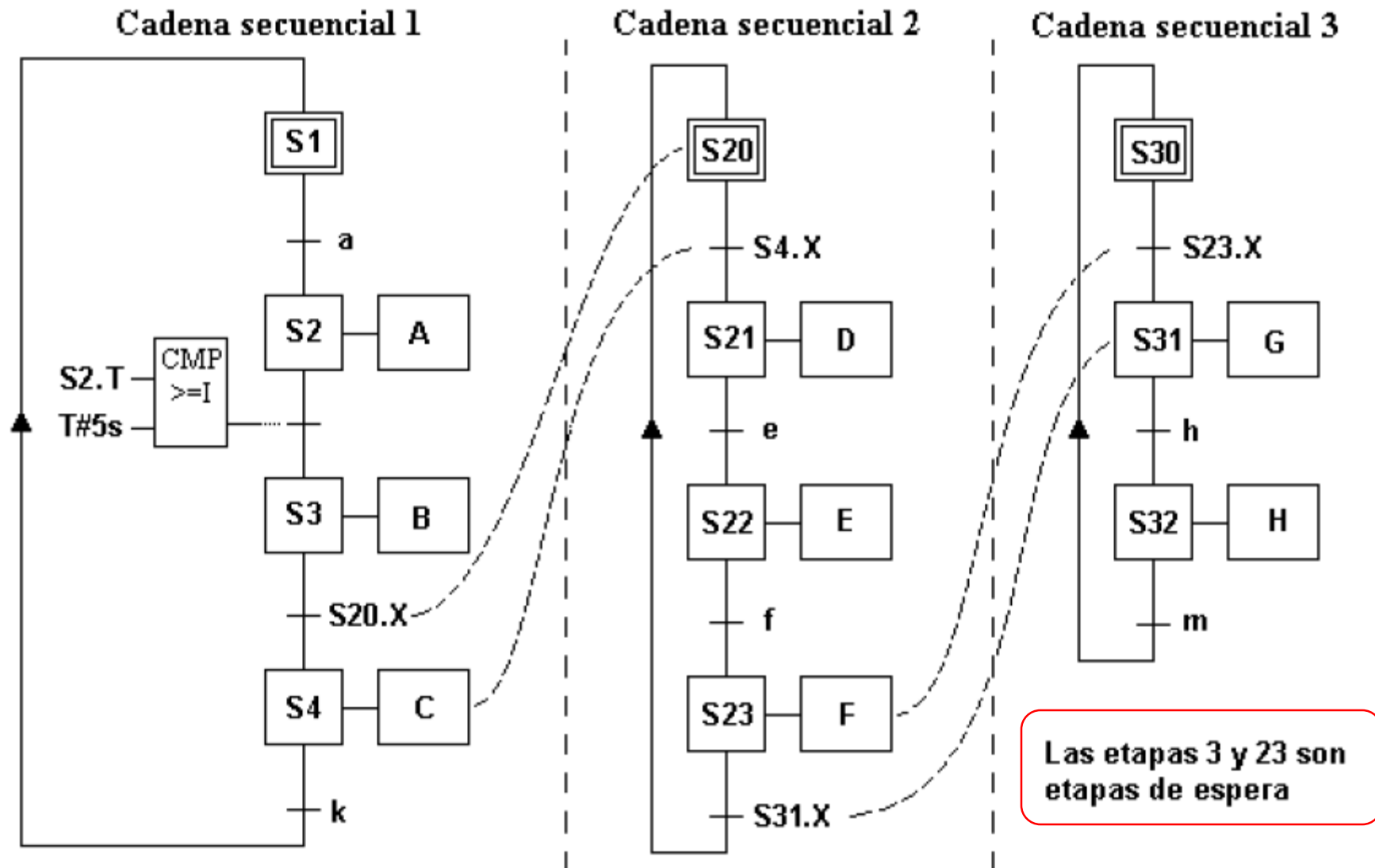


Figura 5.57. Posibilidades de sincronización entre cadenas secuenciales.

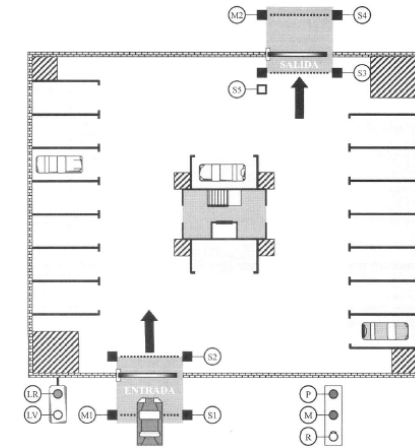
Práctica

a. Partición del Algoritmo en Fases o Etapas:

- Mandado Autómatas Programables, p. 295-304

Ejemplo 5.3:

Control de Cochera de Estacionamiento.

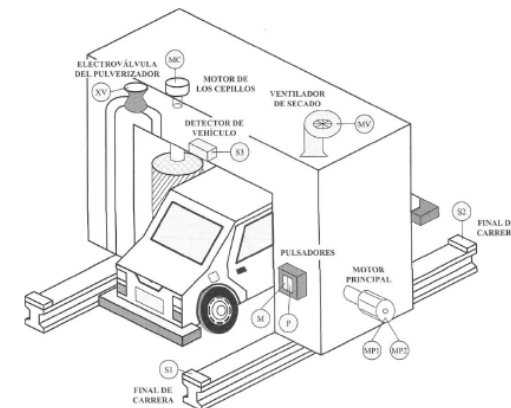


b. Sequential Function Chart (SFC):

- Mandado Autómatas Programables, p. 304-309

Ejemplo 5.4:

Control de Máq. Lavadora de autos.



Práctica: Ej. 5.3 Control Automático de Cochera de Estacionam.

El garaje dispone de un acceso de entrada y otro de salida controlados por sendas barreras que se accionan mediante los motores eléctricos M1 y M2 respectivamente.

A ambos lados de las dos barreras se instalan sensores de presencia de vehículo: S1 y S2 en la de entrada y S3 y S4 en la de salida. Dichos sensores permanecen activados mientras hay un vehículo ante ellos y por su situación física nunca se activan simultáneamente S1 y S2 ni S3 y S4. Se utiliza un sensor S5 para detectar la presencia de ficha en el control de salida.

La capacidad del garaje es de 10 vehículos y el sistema electrónico debe controlar la ejecución de las siguientes acciones:

- *Apertura y cierre automático de las barreras*

La barrera de entrada debe abrirse si en el interior del garaje hay menos de 10 vehículos y se produce un flanco de subida (paso de desactivado a activado del sensor S1). Dicha barrera debe cerrarse si se produce un flanco de bajada (paso de activado a desactivado del sensor S2). La barrera de salida debe abrirse si se activa S5 (presencia de ficha) y se produce un flanco de subida en S3 y debe cerrarse al producirse un flanco de bajada en S4.

- *Señalización a la entrada, mediante una luz verde LV, de que existen plazas libres en el garaje*
- *Señalización a la entrada, mediante una luz roja LR, de que el garaje está completo y no pueden entrar más coches.*

El sistema de control debe poseer además los siguientes elementos de entrada:

- Un pulsador M para ponerlo en marcha. A partir del instante de dar tensión al automático no se permite la entrada o salida de vehículos hasta que no se accione este pulsador.
- Un pulsador de paro P para dejarlo fuera de servicio. Si se acciona este pulsador queda impedida la entrada y salida de vehículos hasta que se accione el pulsador M. En el caso de que P y M se accionen simultáneamente, el primero debe predominar sobre el segundo.
- Un pulsador R para poner a 10 el contador de vehículos en el instante de dar tensión al automático.

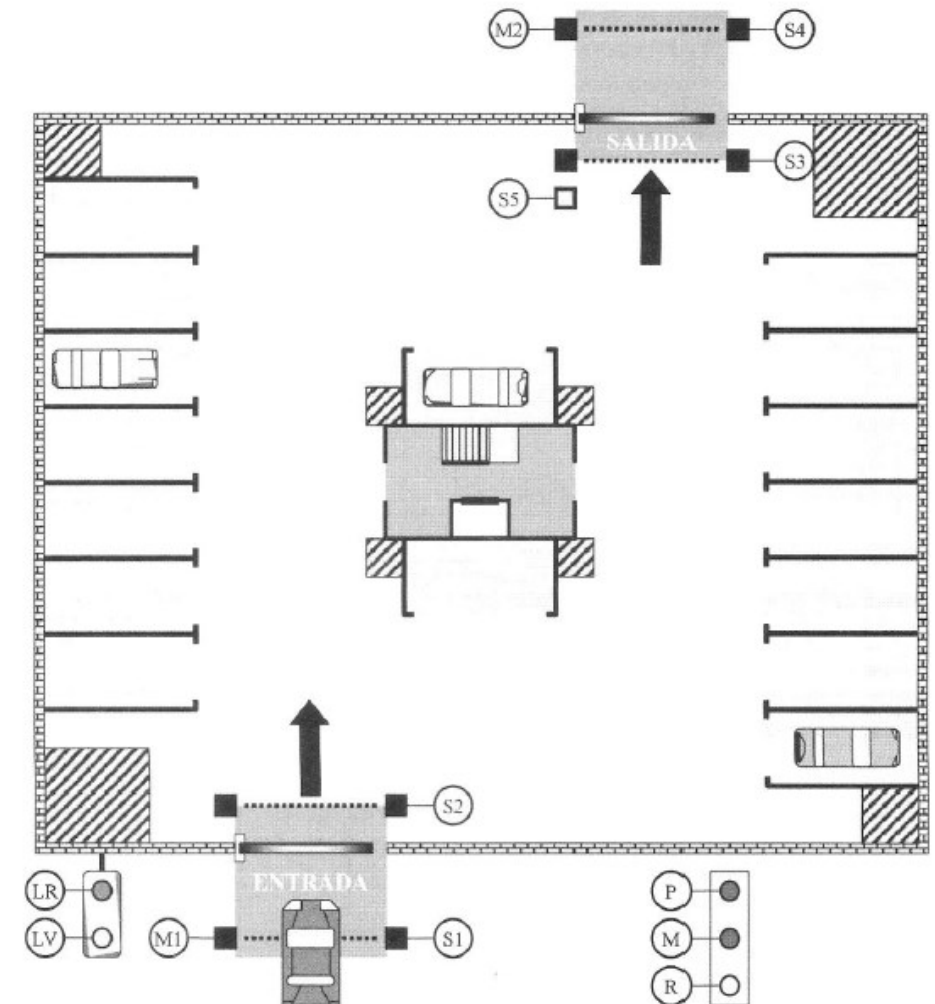
Hacer:

0. Emulación de biestables SR salida LADDER

1. Autómata: plantear **GRAFCET**

2. Partición en fases/etapas → **ST (CODESYS)**

3. Implementación automata directamente en **SFC (CODESYS)**



AUTÓMATAS y CONTROL DISCRETO

3. Diseño de Sistemas de Control Secuencial

A. Métodos Clásicos para Sistemas Simples

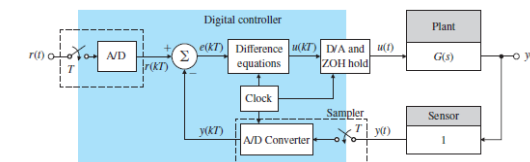
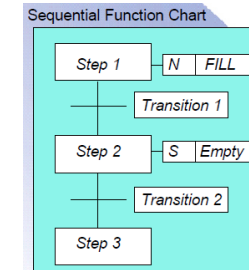
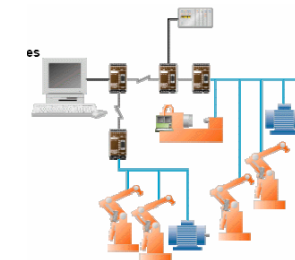
- Proyectos de Automatización. Entornos de Desarrollo (IDE) y Métodos de Diseño.
- Métodos de Diseño Clásicos: Sistemas Combinacionales. Sistemas Secuenciales de Control: Emulación de Biestables S-R. Emulación del Diagrama de Transición de Estados.

B. Métodos Sistemáticos y Modernos para Sistemas Complejos

- Descomposición Jerárquica.
- Partición de Algoritmo en Fases.
- SFC (GRAFCET): Diagramas de Función Secuenciales.

C. Aplicaciones. Buenas Prácticas de Programación y Configuración

- Uso de Entornos Integrados de Desarrollo (IDE). Creación de un Proyecto.
- Configuración de Hardware. Estructuración de Software.
- Uso de CODESYS, TIA Portal, etc. Recomendaciones: PLCopen, etc.
- Ejemplos y Ejercicios de Aplicación.



GEMMA: Especificaciones Sistemáticas

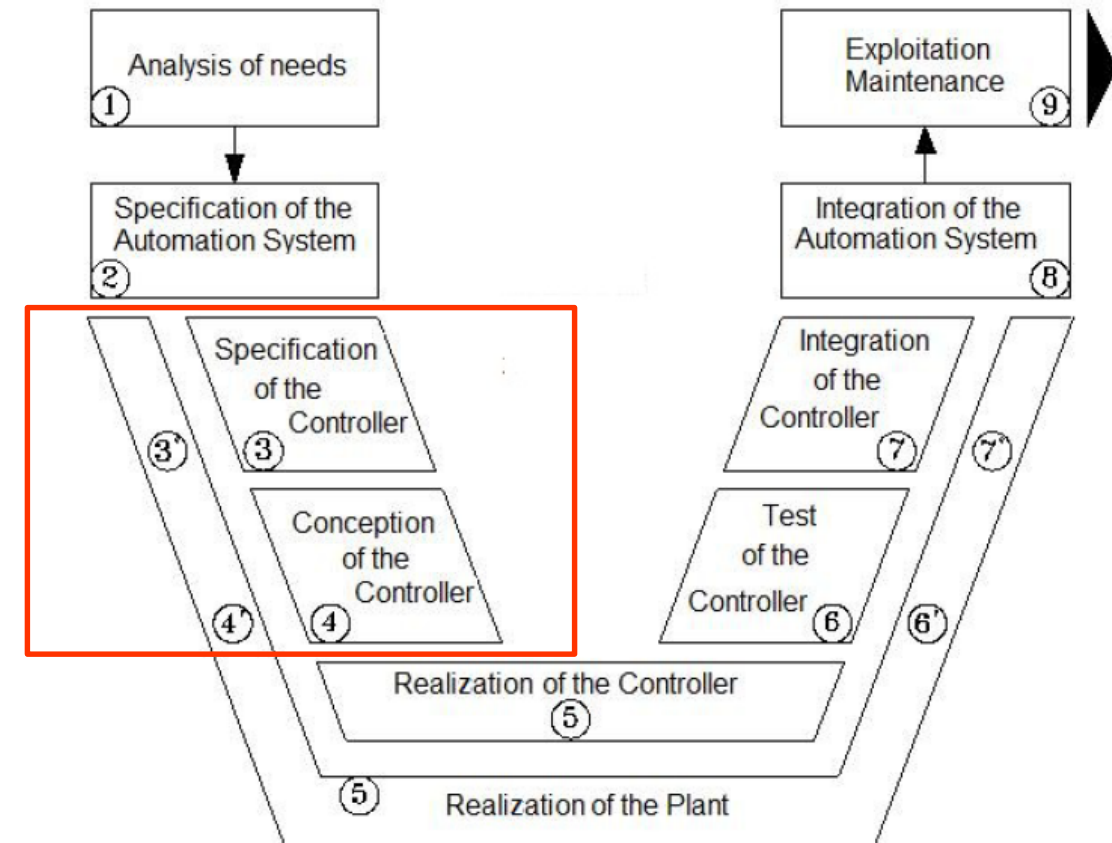
GEMMA: Guide d'Etude des Modes de Marches et d'Arrêts =

Guía de Estudio de **Modos** (estados discretos) de **Arranque y Paradas**

Enfoque **SISTEMÁTICO** para obtener **ESPECIFICACIONES Confiables** de Controlador para Planta Híbrida

Modelo V de
Ciclo de Vida de Desarrollo
de Sistema de Control
(enfoque “mecatrónico”)

Especificación y Concepción del Comportamiento del Controlador →



Diseño de Sist. de Automatización

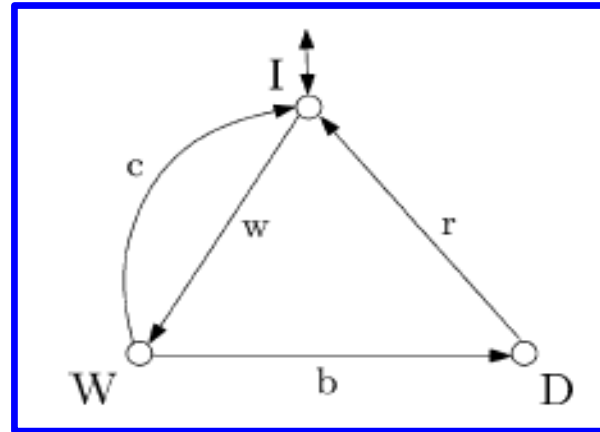
Ref.: [AyCD_01d-03a_Grafcet-SFC_GEMMA_A systematized approach to obtain dependable controller specifications for hybrid plants.pdf](#)

GEMMA: Especificaciones Sistemáticas

GEMMA: Guide d'Etude des Modes de Marches et d'Arrêts =
Guía de Estudio de **Modos** (estados discretos) de **Arranque y Paradas**

Enfoque **SISTEMÁTICO** para obtener **ESPECIFICACIONES Confiables** de Controlador para Planta Híbrida

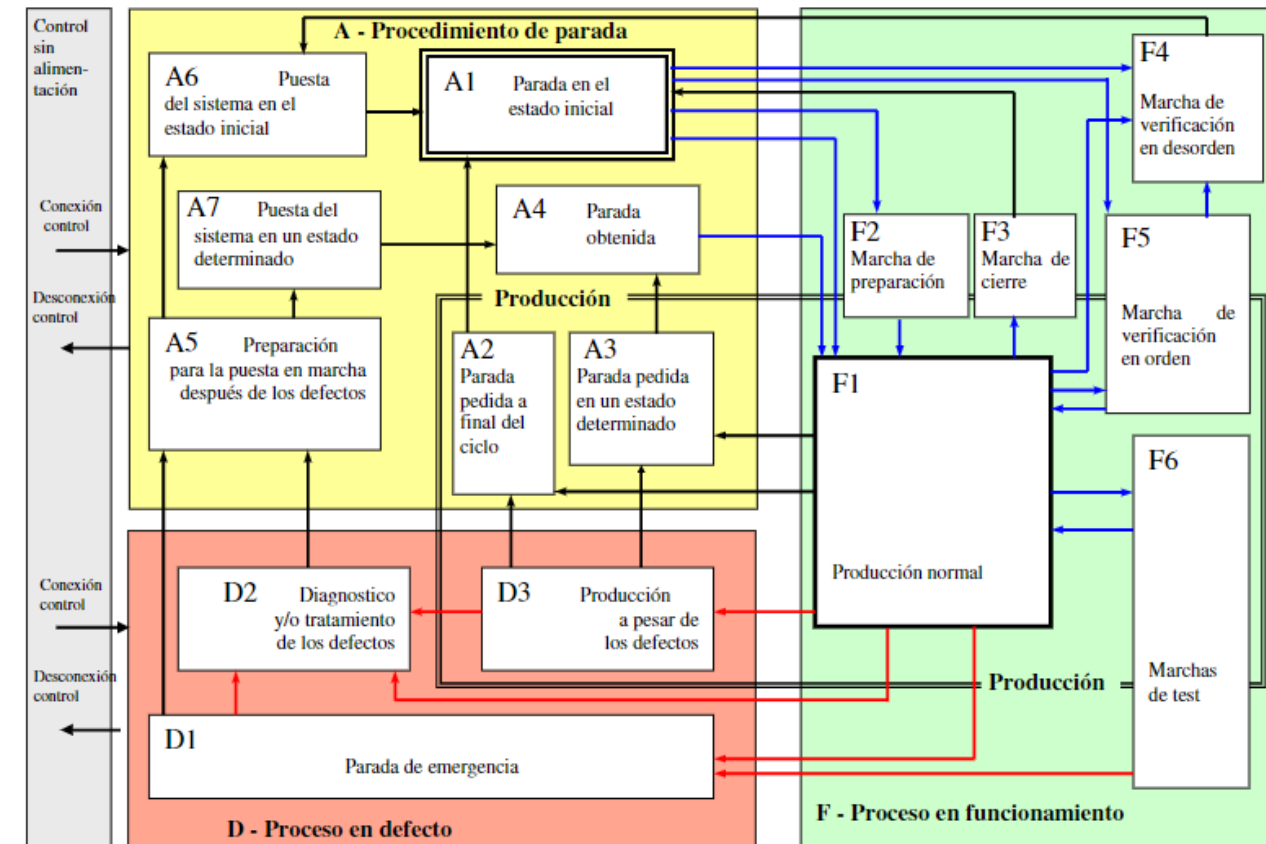
GEMMA:
Modos de Funcionamiento
y
Estados de Parada
Considerados en el diseño
de un
Sistema de Automatización



Ref.: Curso de GRAFCET y GEMMA
<https://web.archive.org/web/20110709021515/http://edison.upc.edu/curs/grafcet/>

Diseño de Sist. de Automatización

Ref.: AyCD_01d-03a_Grafcet-SFC_GEMMA_A systematized approach to obtain dependable controller specifications for hybrid plants.pdf



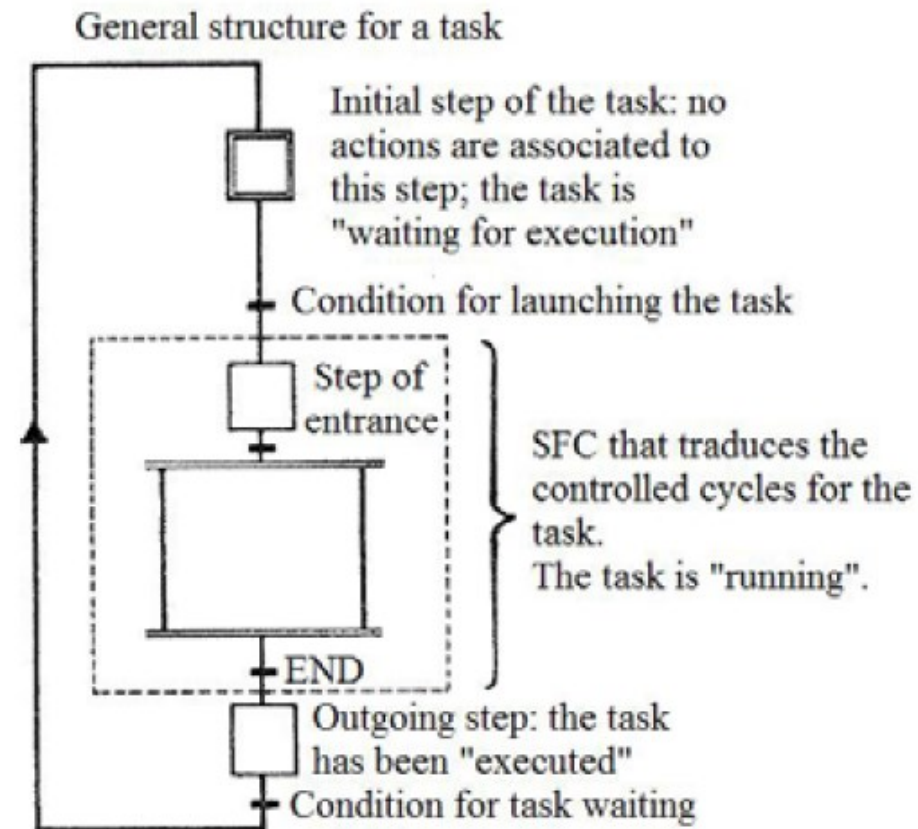
GEMMA: Especificaciones Sistemáticas

GEMMA: Guide d'Etude des Modes de Marches et d'Arrêts =

Guía de Estudio de **Modos** (estados discretos) de **Arranque y Paradas**

Enfoque **SISTEMÁTICO** para obtener **ESPECIFICACIONES Confiables** de Controlador para Planta Híbrida

Estructura general de una TAREA



Diseño de Sist. de Automatización

Ref.: [AyCD_01d-03a_Grafcet-SFC_GEMMA_A systematized approach to obtain dependable controller specifications for hybrid plants.pdf](#)

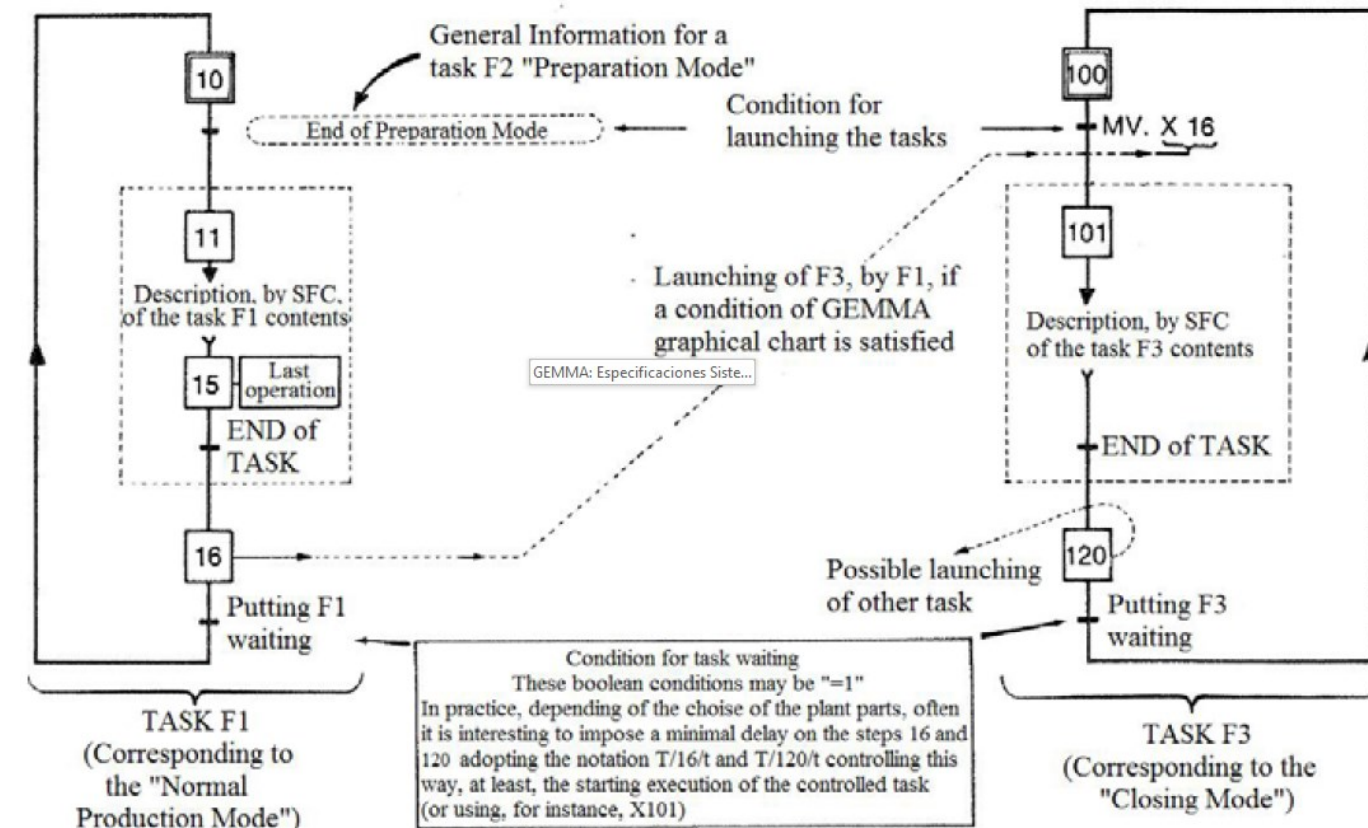
GEMMA: Especificaciones Sistemáticas

GEMMA: Guide d'Etude des Modes de Marches et d'Arrêts =

Guía de Estudio de **Modos** (estados discretos) de **Arranque y Paradas**

Enfoque **SISTEMÁTICO** para obtener **ESPECIFICACIONES Confiables** de Controlador para Planta Híbrida

Coordinación Horizontal



Diseño de Sist. de Automatización

Ref.: AyCD_01d-03a_Grafcet-SFC_GEMMA_A systematized approach to obtain dependable controller specifications for hybrid plants.pdf

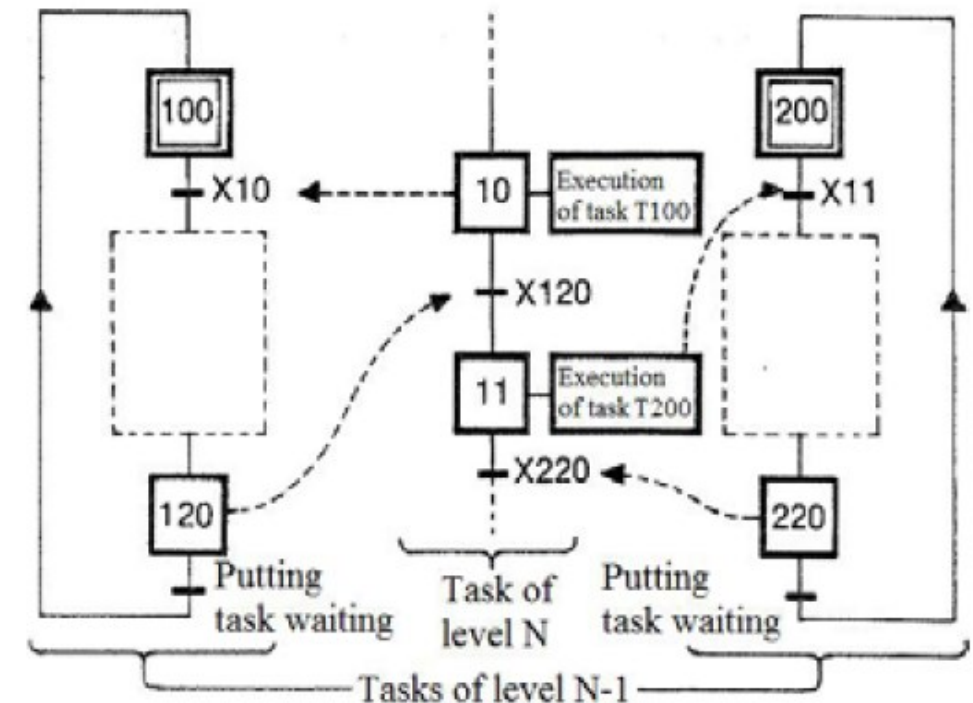
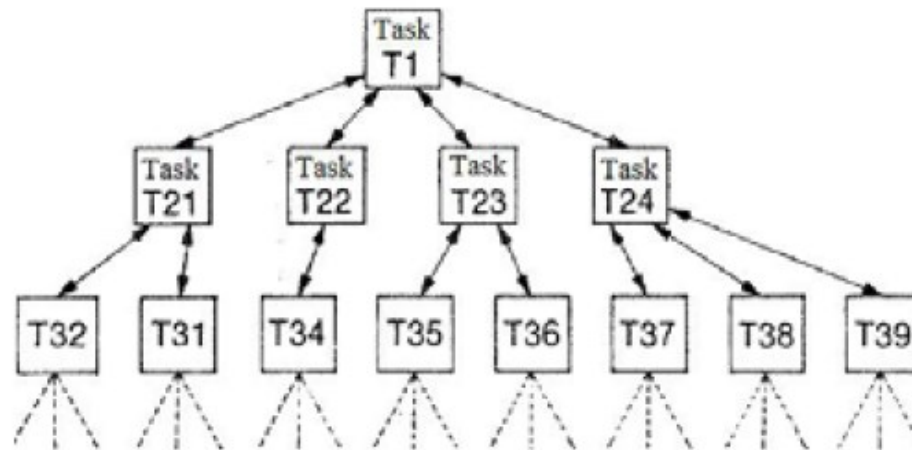
GEMMA: Especificaciones Sistemáticas

GEMMA: Guide d'Etude des Modes de Marches et d'Arrêts =

Guía de Estudio de **Modos** (estados discretos) de **Arranque y Paradas**

Enfoque **SISTEMÁTICO** para obtener **ESPECIFICACIONES Confiables** de Controlador para Planta Híbrida

Coordinación Vertical



Diseño de Sist. de Automatización

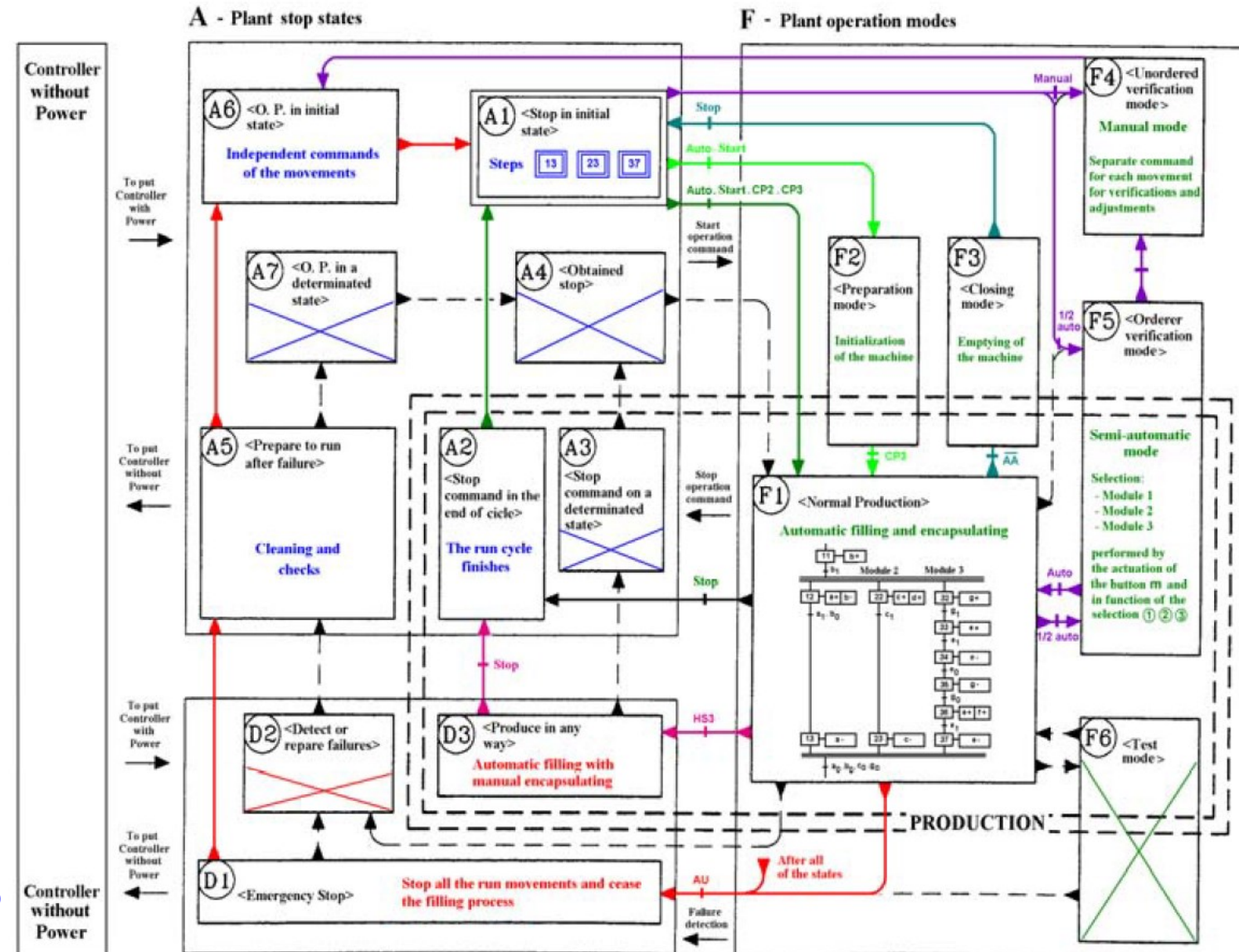
Ref.: [AyCD_01d-03a_Grafcet-SFC_GEMMA_A systematized approach to obtain dependable controller specifications for hybrid plants.pdf](#)

GEMMA: Especificaciones Sistemáticas

Ejemplo:

Ver en doc. Ref.:

AyCD_01d-03a_Grafcet-SFC_GEMMA_A systematized approach to



IEC 61131-3: Entornos de Desarrollo

CODESYS: Plataforma de desarrollo IEC 61131-3

<https://www.codesys.com>

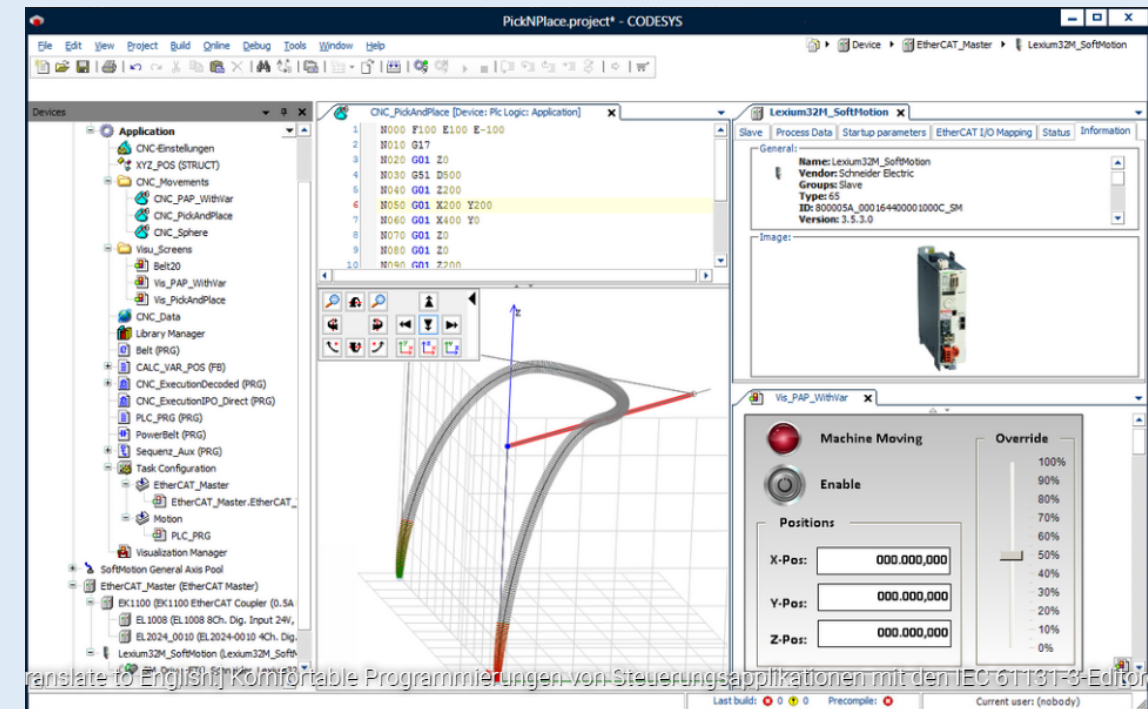
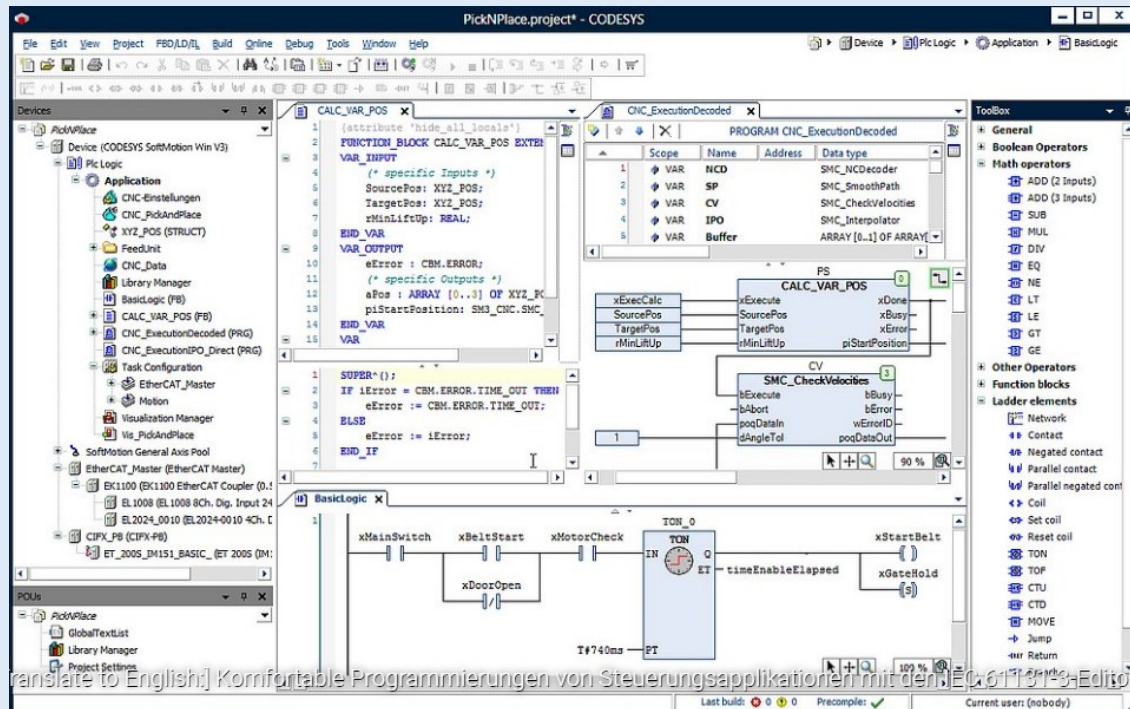
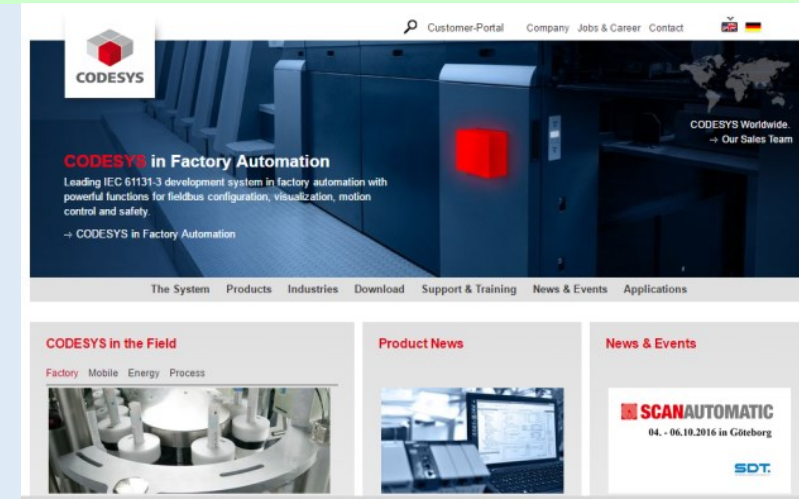
Instalar: **CODESYS 64 3.5.20.20.exe**

Ver documento Instalación e inicio CODESYS:

1. CODESYS_Installation & Start_User Doc 18.0.pdf

2. Ver **CODESYS ONLINE HELP**

<https://www.helpme-codesys.com/>



IEC 61131-3: Entornos de Desarrollo

Otros entornos similares:

Step 7/TIA Portal (Siemens)

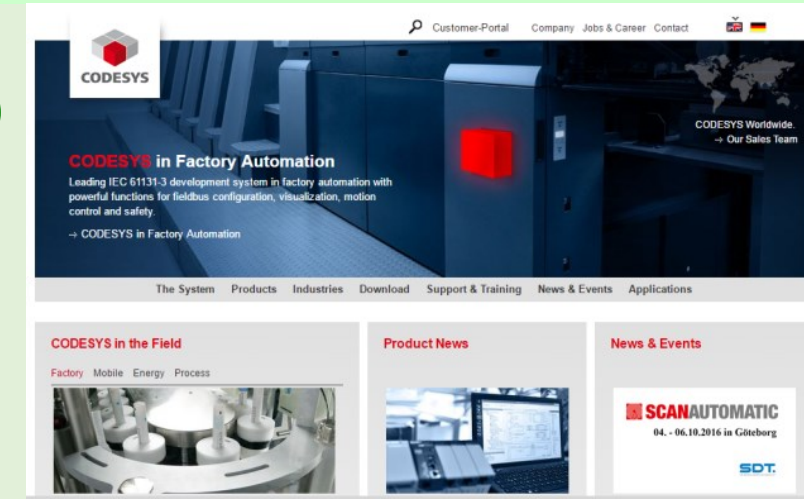
PC Worx / PLCnext (Phoenix Contact)

Unity Pro (Schneider)

OpenPCS (infoteam)

<https://infoteam.de/en/our-know-how/plc-programming-systems>

Etc...



IEC 61131-3: desde Simulink PLC Coder

<https://www.mathworks.com/products/simulink-plc-coder.html>

Generación de Código IEC 61131-3:
Structured Text (and **Ladder Diagrams**)
para PLCs y PACs
(independiente del hardware)

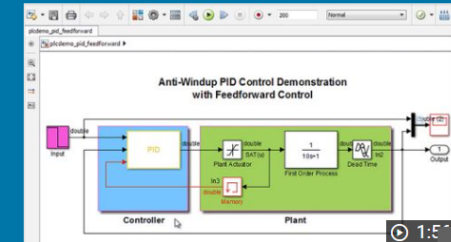
Simulink PLC Coder

Generate IEC 61131-3 Structured Text and Ladder Diagrams for PLCs and PACs

Request a free trial

Request a quote

Simulink PLC Coder™ generates hardware-independent IEC 61131-3 Structured Text and Ladder Diagrams from Simulink® models, Stateflow® charts, and MATLAB® functions. Structured Text is generated in PLCopen XML and other file formats supported by widely used integrated development environments (IDEs), including 3S-Smart Software Solutions CODESYS®, Rockwell Automation Studio 5000, Siemens TIA Portal, and Omron® Sysmac® Studio. Ladder diagrams are generated in file formats



Advanced techniques for customizing IEC 61131 code generation for PLCs

<https://www.mathworks.com/videos/advanced-techniques-for-customizing-iec-61131-code-generation-for-plcs-1607094156796.html>

