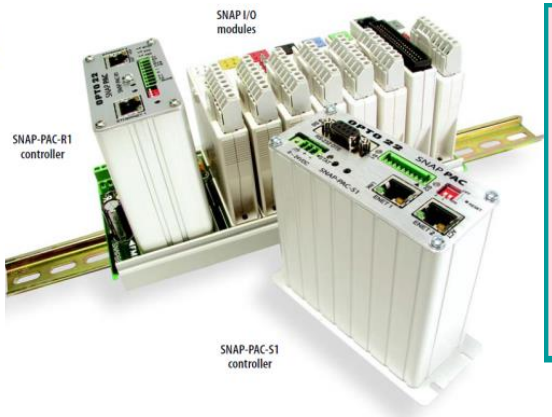
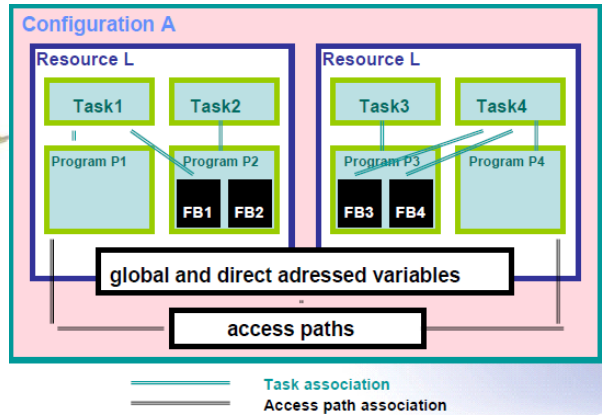
Unidad 3:

Diseño de Sistemas de Control Secuencial
Norma IEC 61131-3





Diseño de Sistemas de Control Secuencial

Plan General:

1. **Introducción: Autómatas y Control Discreto**
   - Aplicaciones y Fundamentos. Autómatas / FSM. STATEFLOW. GRAFCET.
2. **Autómatas Programables – IEC61131: Arq. y Lenguajes**
   - PLCs, arquitecturas. Entorno de desarrollo. Configuración. Lenguajes ST / SFC.
3. **Diseño de Sists. Control Secuencial**
   - Métodos clásicos. Métodos sistemáticos. Aplicación.
4. **Comunicaciones Industriales: Redes y Protocolos**
   - Arquitecturas distribuidas. Buses de Campo y Protocolos industriales.
5. **Control Discreto de Sistemas Continuos e Híbridos**
   - Sistemas muestreados. Simulación y Control. Sistemas Híbridos. Aplicación.

Diseño de Sistemas de Control Secuencial

Plan de la Unidad 3:

A. Métodos Clásicos

- Proyectos de Automatización. Entornos de Desarrollo 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

- Descomposición jerárquica.
- Partición del Algoritmo en Fases.
- SFC/Grafcet: Diagramas de Función Secuenciales

C. Aplicaciones

- Uso de entornos de Desarrollo.
- Creación de un Proyecto. Configuración de Hardware. Estructura de Software.
- Resumen y Consultas.
- Próximos Pasos...

Diseño de Sists. 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 (config. Hw y Sw).*
   - *Suministradas por cada fabricante (ej. **Step 7** [Siemens], **PC Worx** [Phoenix Contact], etc.) vs. genéricas (**CODESYS**, **OpenPCS**).*

b. *Aplicación de **Métodos** de Diseño:*
   - ***Especificaciones** de funcionamiento → **Programa** de Control (lenguajes de Programación y elementos commune **IEC 61131-3**);*
   - ***Métodos Clásicos** vs. **Métodos Sistemáticos**.*

Etapas 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:

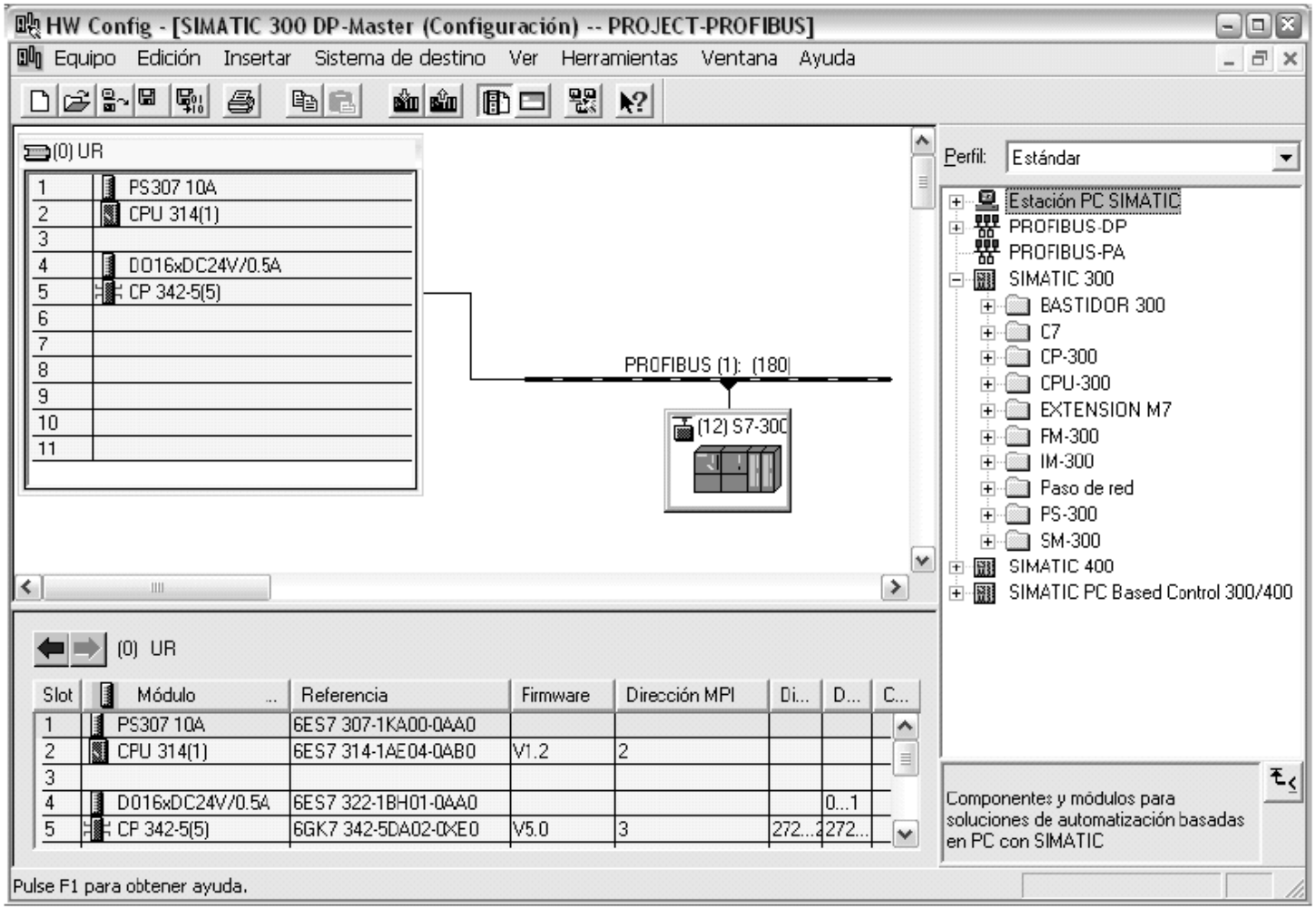
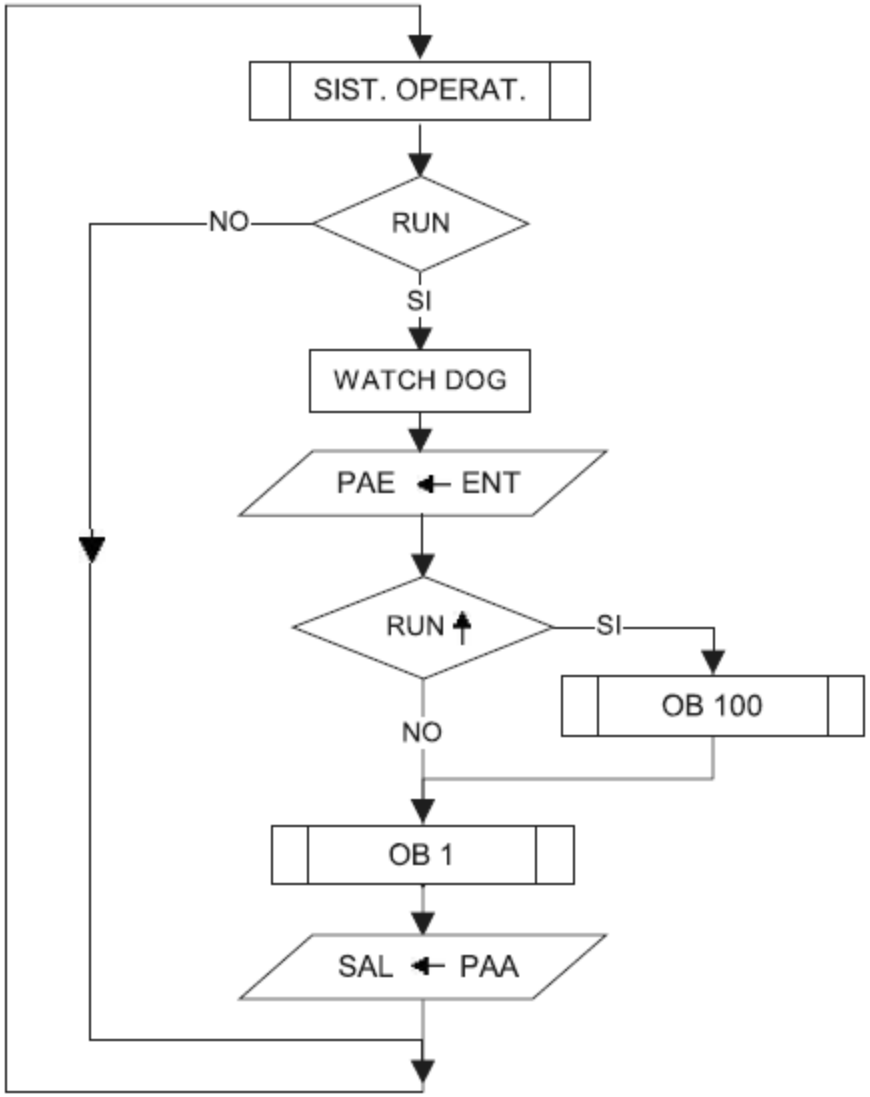


Figura 5.2. *Programa configurador de hardware de STEP7.*

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



PROGRAMA EJECUTIVO

MARCHA ?

CIRCUITO DE VIGILANCIA

MUESTREO Y MEMORIZACIÓN DE ENTRADAS

PRIMER CICLO?

MÓDULO DE INICIALIZACIÓN

MÓDULO PRINCIPAL

ACTUALIZACIÓN DE SALIDAS

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

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

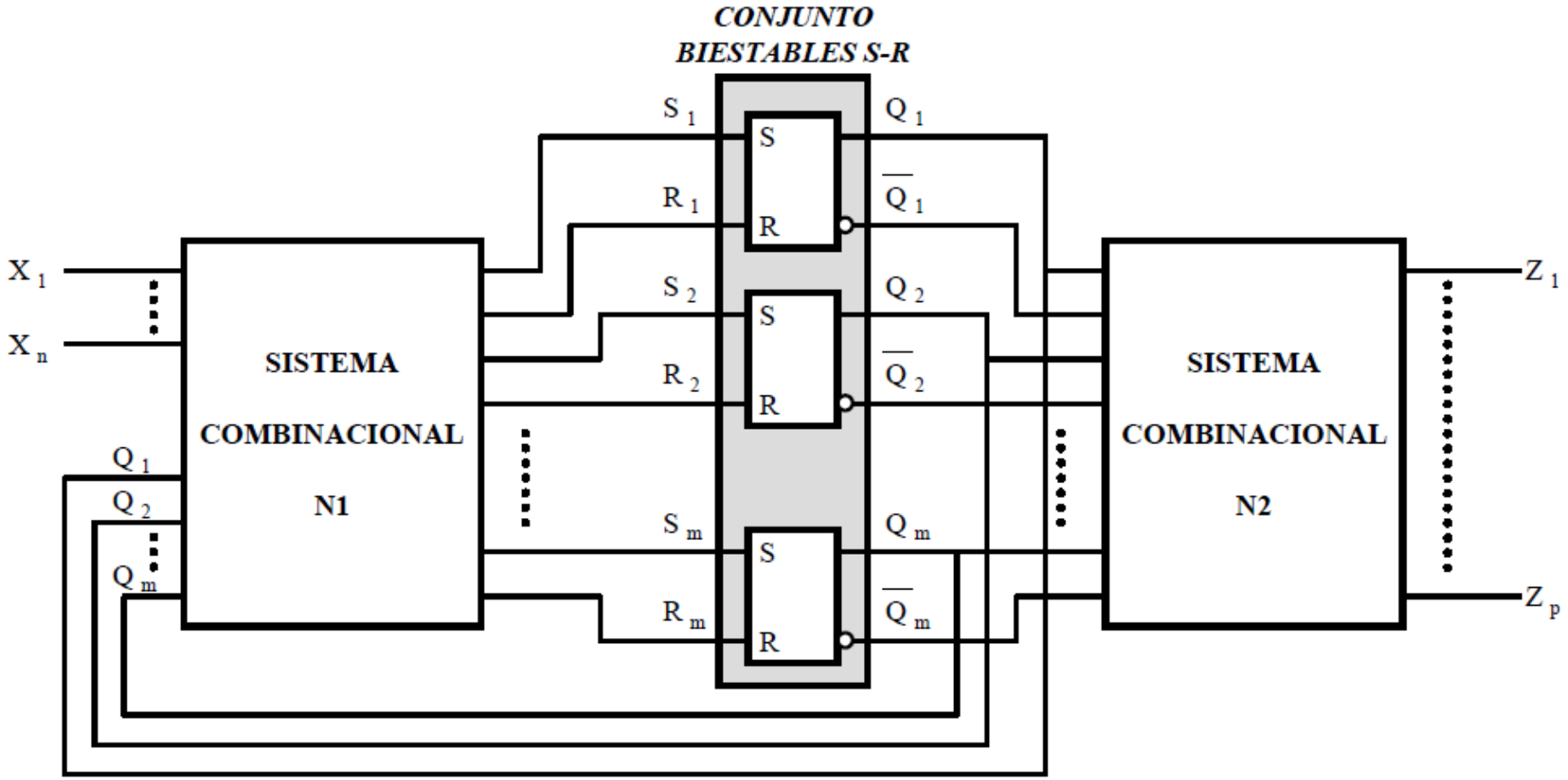


Figura 5.7. *Diagrama de bloques de un sistema secuencial asíncrono realizado con biestables R-S.*

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)*

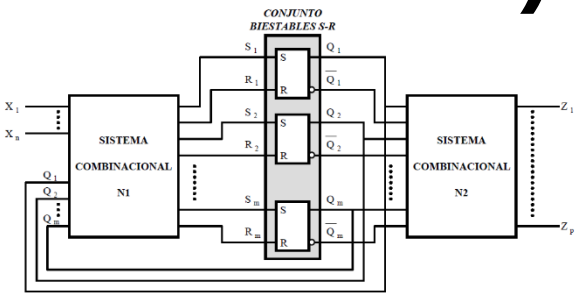


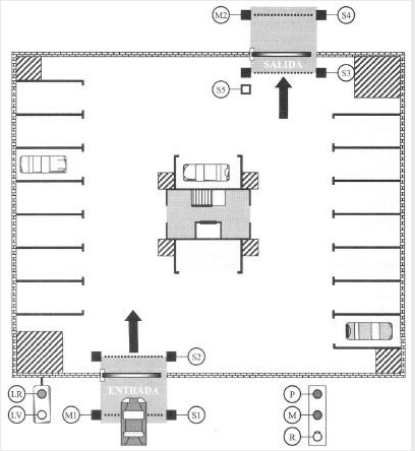
Figura 5.7. *Diagrama de bloques de un sistema secuencial asíncrono realizado con biestables R-S.*

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.*

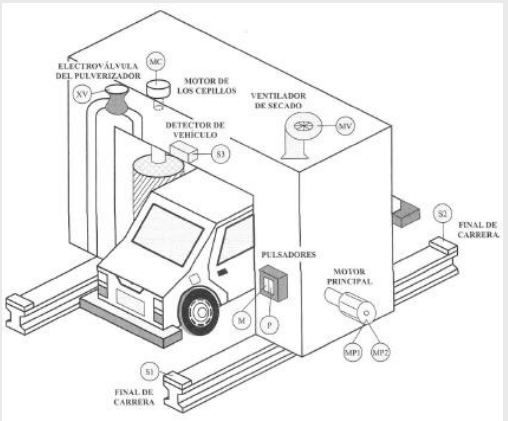
b. *Diseño orientado hacia las **variables de estado interno**:*
   - *Especificaciones → Definir **variables de estado interno** (discreto) para memorizar secuencias de variables de entrada;*
   - *Cada estado interno: **biestable SR interno**;*
   - *Variables de salida: a partir de variables de estado 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.



2. Emulación del Diagrama de Estados

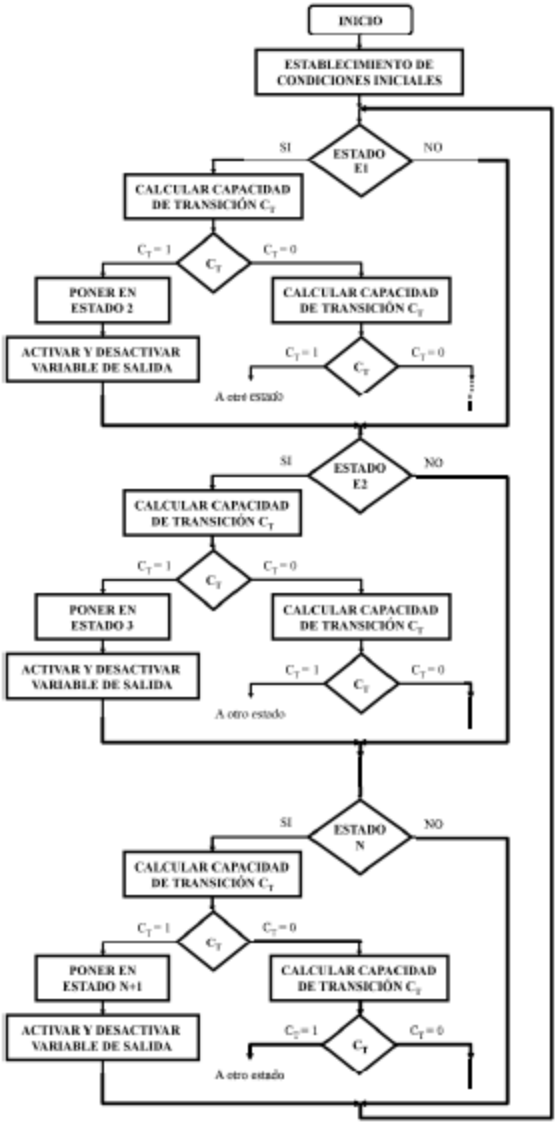


Figura 5.30. *Algoritmo compacto básico de emulación de un diagrama de estados.*

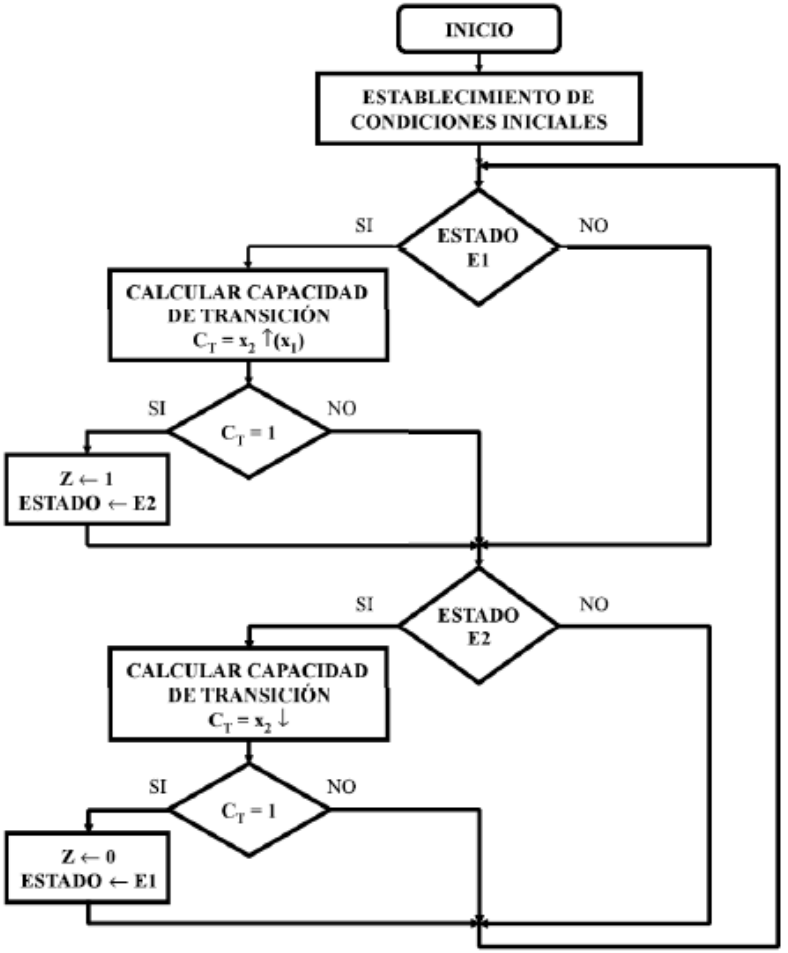
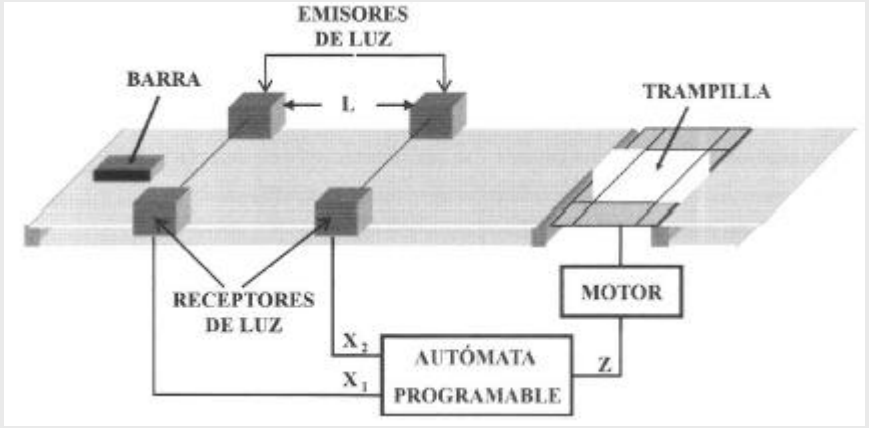


Figura 5.32. *Algoritmo compacto que describe el comportamiento que debe tener el autómata programable que controla el sistema de selección de barras de la figura 5.31.*

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.



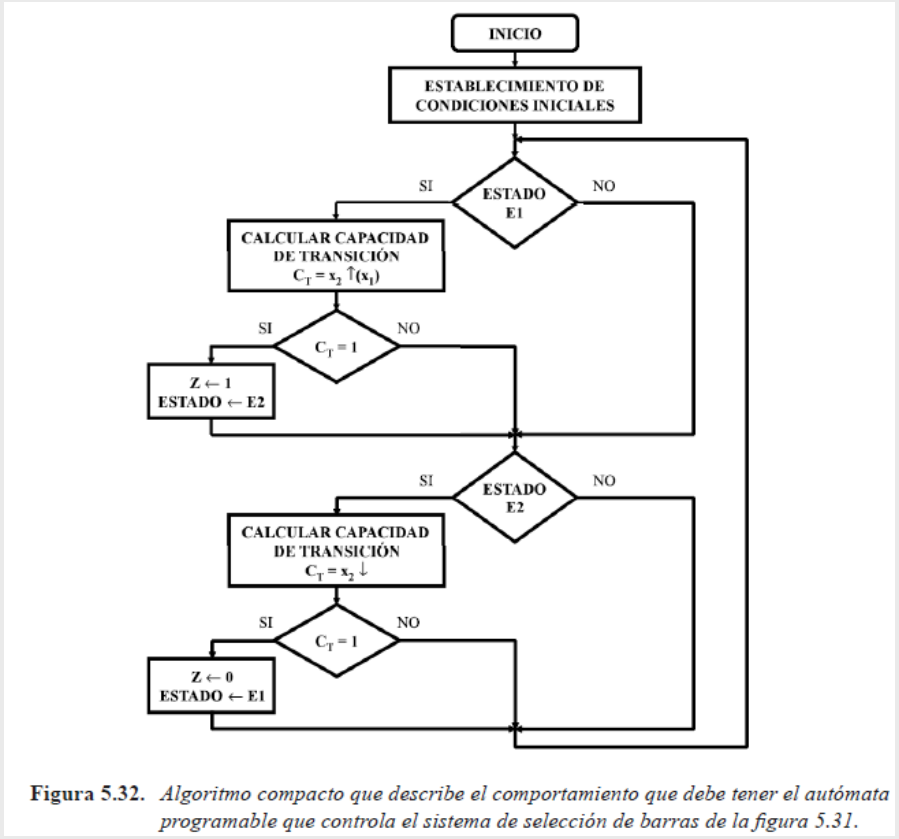


Figura 5.32. *Algoritmo compacto que describe el comportamiento que debe tener el autómata programable que controla el sistema de selección de barras de la figura 5.31.*

Modelo VDI 2221: descompos. jerárquica

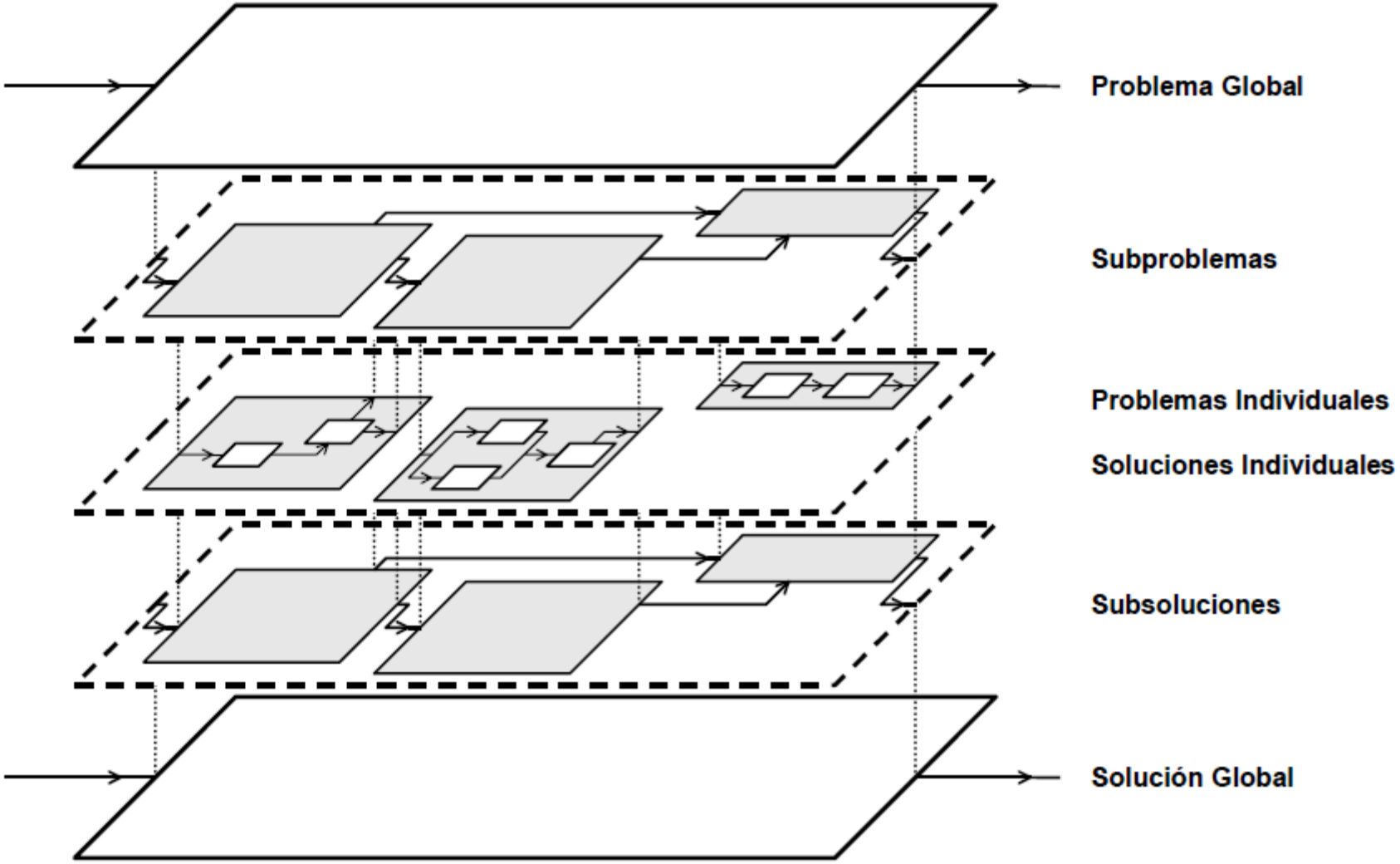


Figura 5.33. *Descripción gráfica del modelo de diseño VDI 2221.*

Partición del Algoritmo en Fases /Etapas

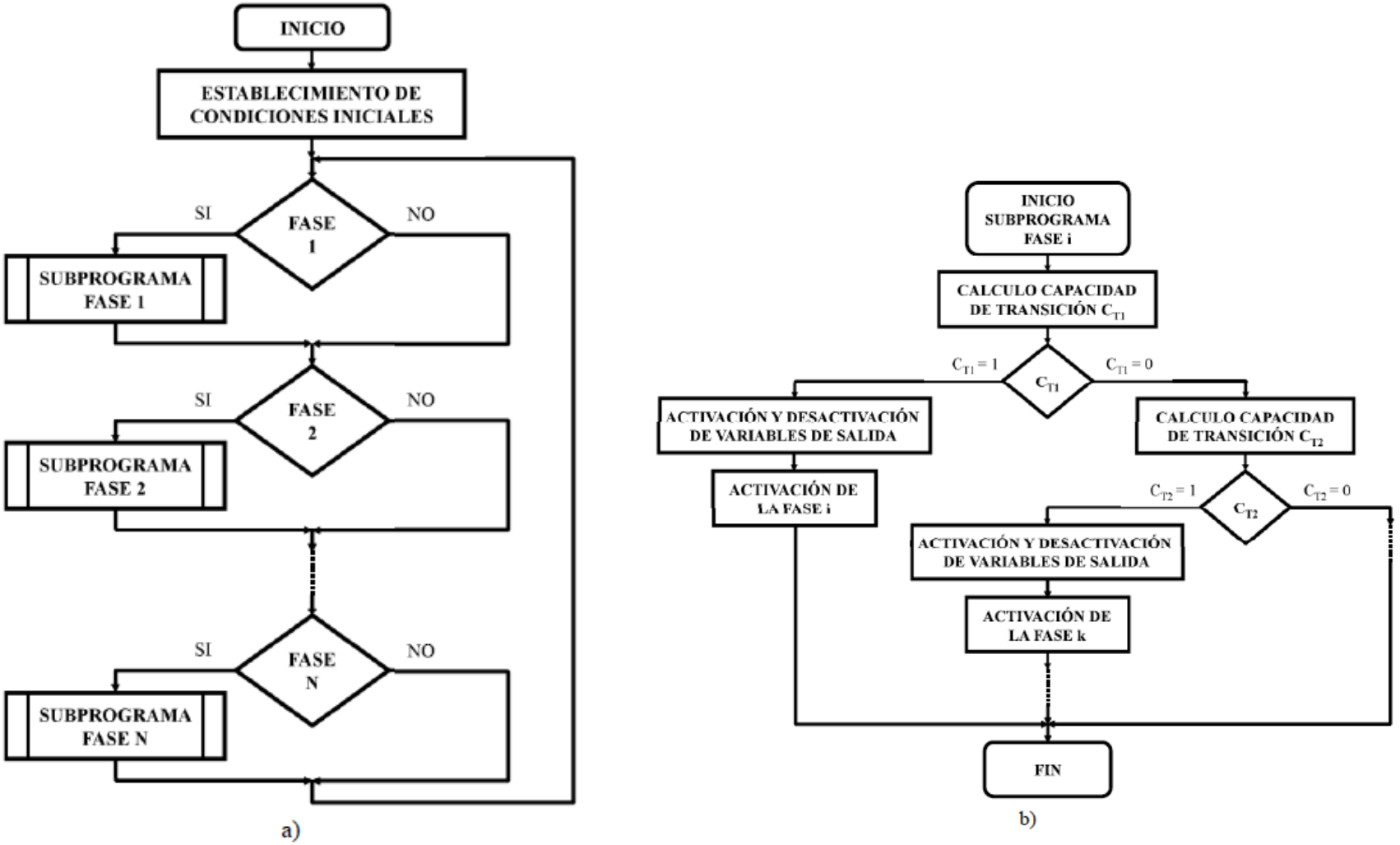


Figura 5.34. *Algoritmo general del método de diseño basado en la partición del algoritmo en fases: a) Programa principal; b) Algoritmo de cada fase.*

Sequential Function Chart (SFC/Grafcet)

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

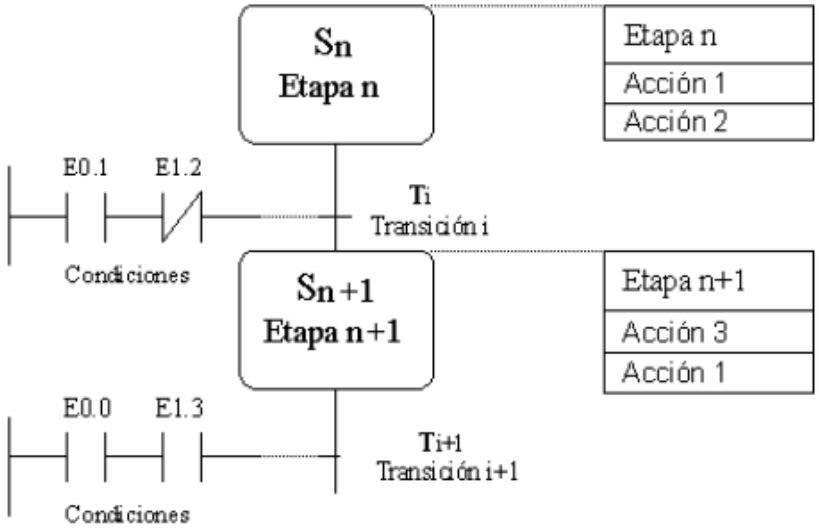


Figura 5.41. *Representación gráfica en el lenguaje S7-GRAPH de las etapas y de las transiciones entre ellas.*

Sequential Function Chart (SFC) Ejemplo

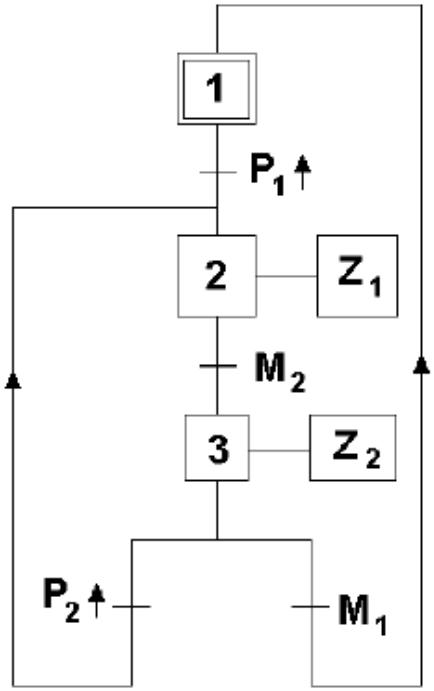


Figura 5.45. *Diagrama SFC del ejemplo 5.9.*

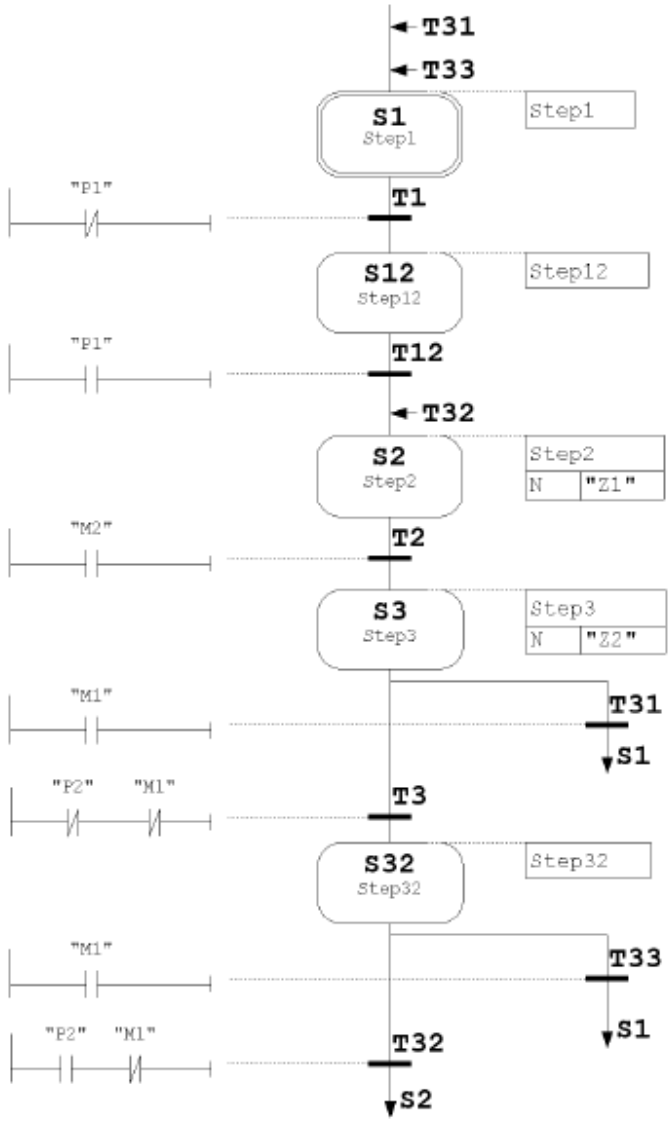


Figura 5.46. *Programa en el lenguaje S7-GRAPH del ejemplo 5.9.*

Sequential Function Chart (SFC) Errores

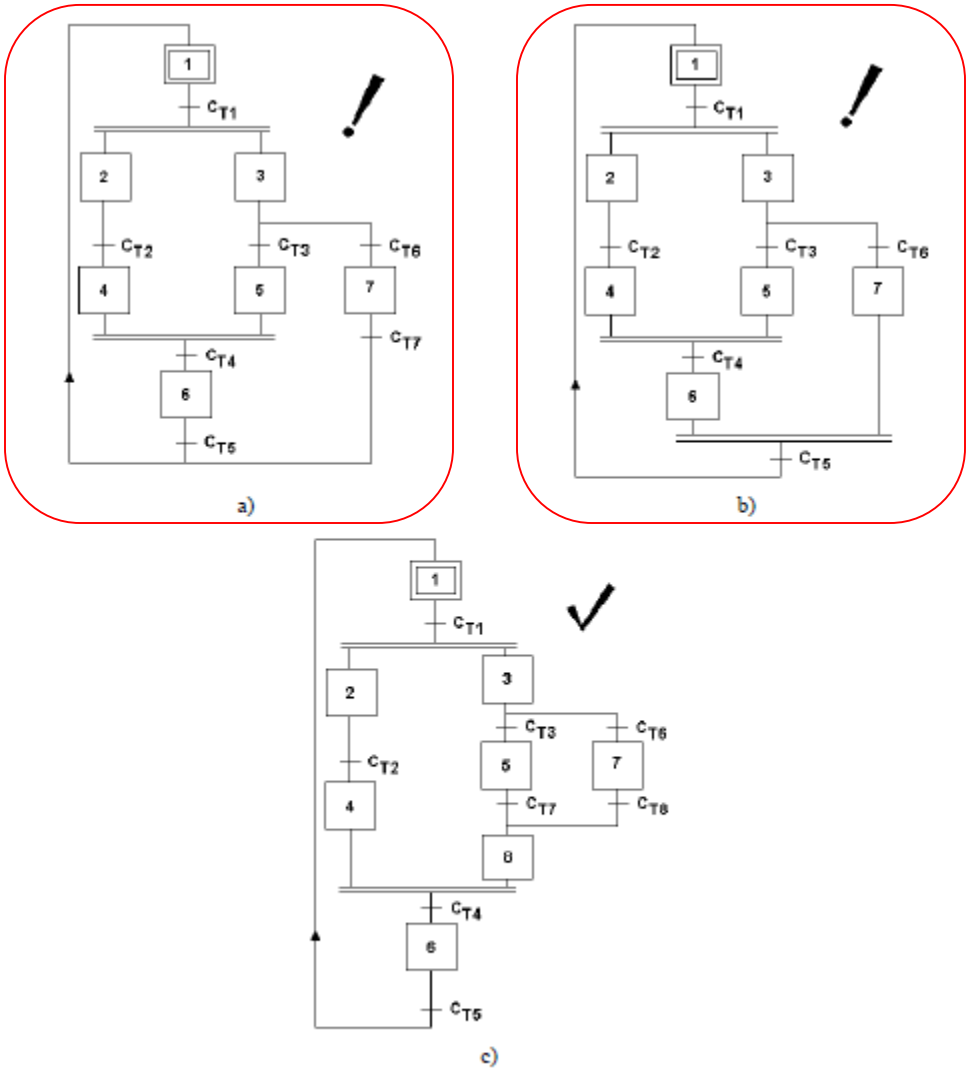


Figura 5.51. *Errores de programación en un diagrama SFC/S7-GRAPH: a) Inseguro; b) Inalcanzable; c) Posible.*

SFC Sincronización

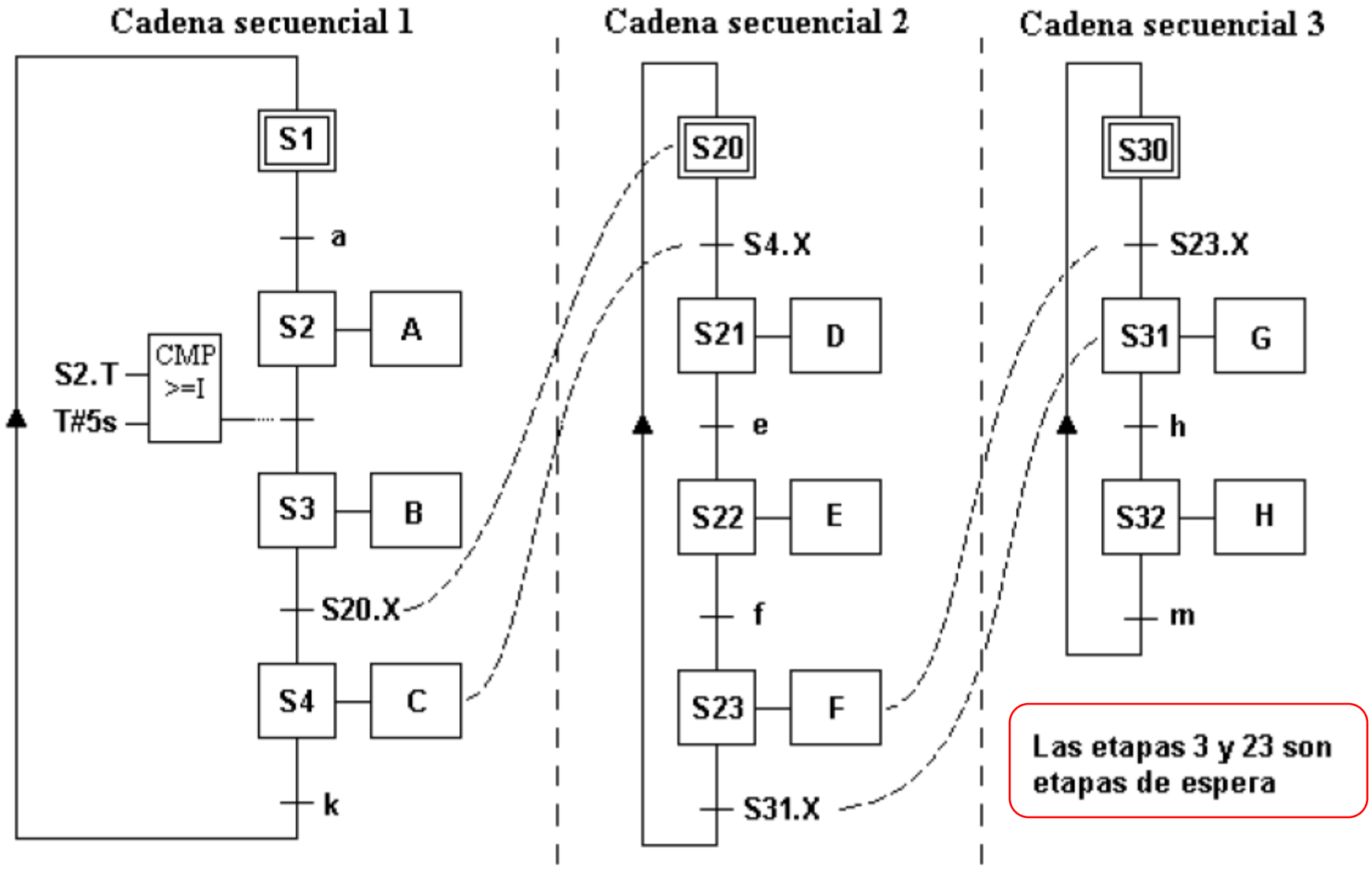
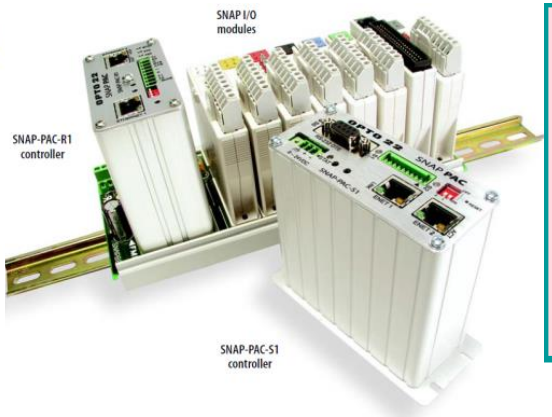
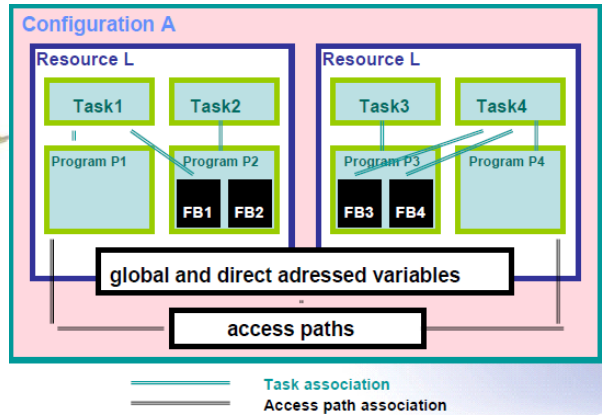


Figura 5.57. *Posibilidades de sincronización entre cadenas secuenciales.*

Unidad 3:

Diseño de Sistemas de Control Secuencial
Norma IEC 61131-3





Diseño de Sistemas de Control Secuencial

Plan de la Unidad 3:

A. Métodos Clásicos

- Proyectos de Automatización. Entornos de Desarrollo 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

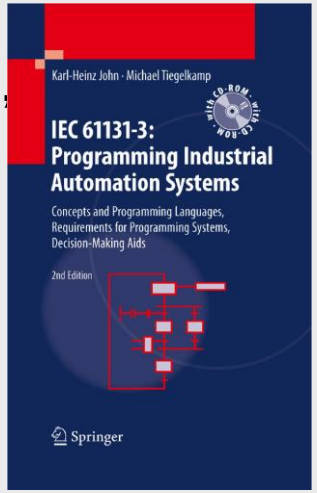
- Descomposición jerárquica.
- Partición del Algoritmo en Fases.
- SFC/Grafcet: Diagramas de Función Secuenciales

C. Aplicaciones

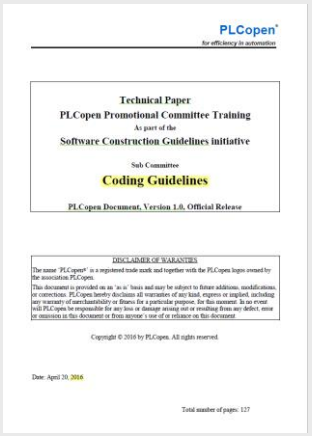
- Uso de entornos de Desarrollo.
- Creación de un Proyecto. Configuración de Hardware. Estructura de Software.
- Resumen y Consultas.
- Próximos Pasos...

Programación c/ IEC 61131-3 (docum.)

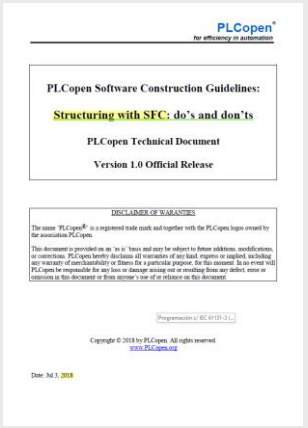
- **Libro:** K.-H. JOHN- M. TIEGELKAMP, **IEC 61131-3: Programming Industrial Automation Systems**, 2nd Ed. 2010.

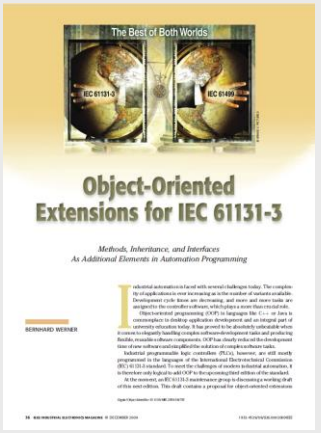


- Documento Técnico **PLC open**: **Coding Guidelines**, Version 1.0, 2016.



- Documento Técnico **PLC open**: **Structuring with SFC: do's & don'ts**, Version 1.0, 2018.

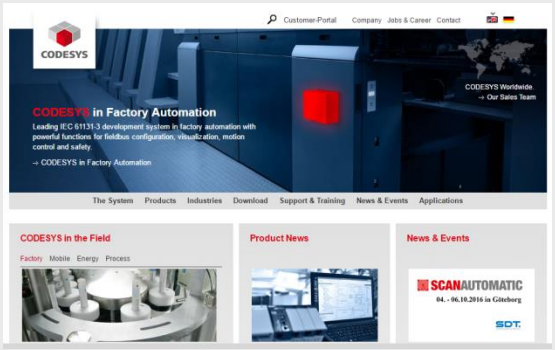




- B. Werner, **Object-Oriented Extensions** for IEC 61131-3, IEEE Industrial Electronics Magazine, Dec. 2009.

IEC 61131-3: Entornos de Desarrollo

CODESYS: Plataforma de desarrollo IEC 61131-3
https://www.codesys.com
Instalar: CODESYS 64 3.5.17.10.exe



- → **CODESYS V3.5 SP17** → **Version: 3.5.17.10**

Ver documento Instalación e inicio CODESYS:

1. CODESYS_Installation & Start_User Doc 18.0.pdf
2. Ver **CODESYS ONLINE HELP**
   https://help.codesys.com/webapp/_cds_f_development_system_introduction;product=codesys;version=3.5.17.0
3. CODESYS Corp. EDUCATION: Deep dive – **Getting Started with CODESYS**
   https://www.youtube.com/watch?v=HlV0Nb__UQM (Video 2021-10-29, 1 h 14 min)

CODESYS Development System V3

The CODESYS Development System is an **IEC 61131-3 programming tool** for the industrial controller and automation technology sector, available in a 32-bit and a 64-bit version.

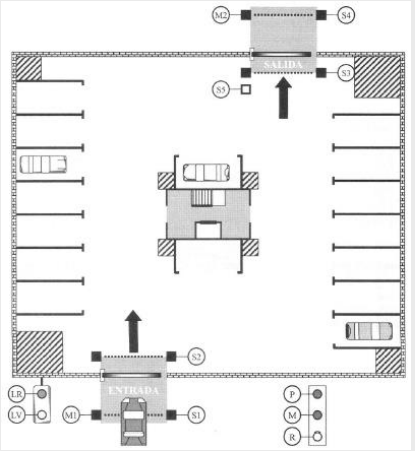
The CODESYS Development System **engineering tool** integrates various support functions in every phase of development:

- **Project tree** for structuring project configuration, for example to divide the entire application into **objects** and **tasks**
- **Configurator** for integrating and describing various devices and fieldbus systems
- **Editors** for typical application development in all graphical and text-based implementation languages defined by IEC 61131-3
- **Compilers** for building applications in lean and powerful machine code
- **Debugger, simulator, and SoftPLC** (as trial target system) for direct user testing of the created applications

Práctica

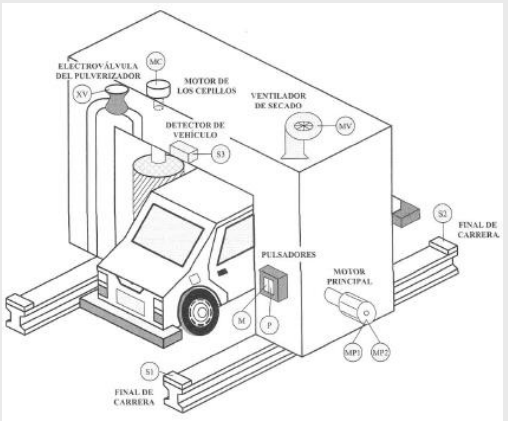
a. ***Partición del Algoritmo en Fases o Etapas***:

– *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.

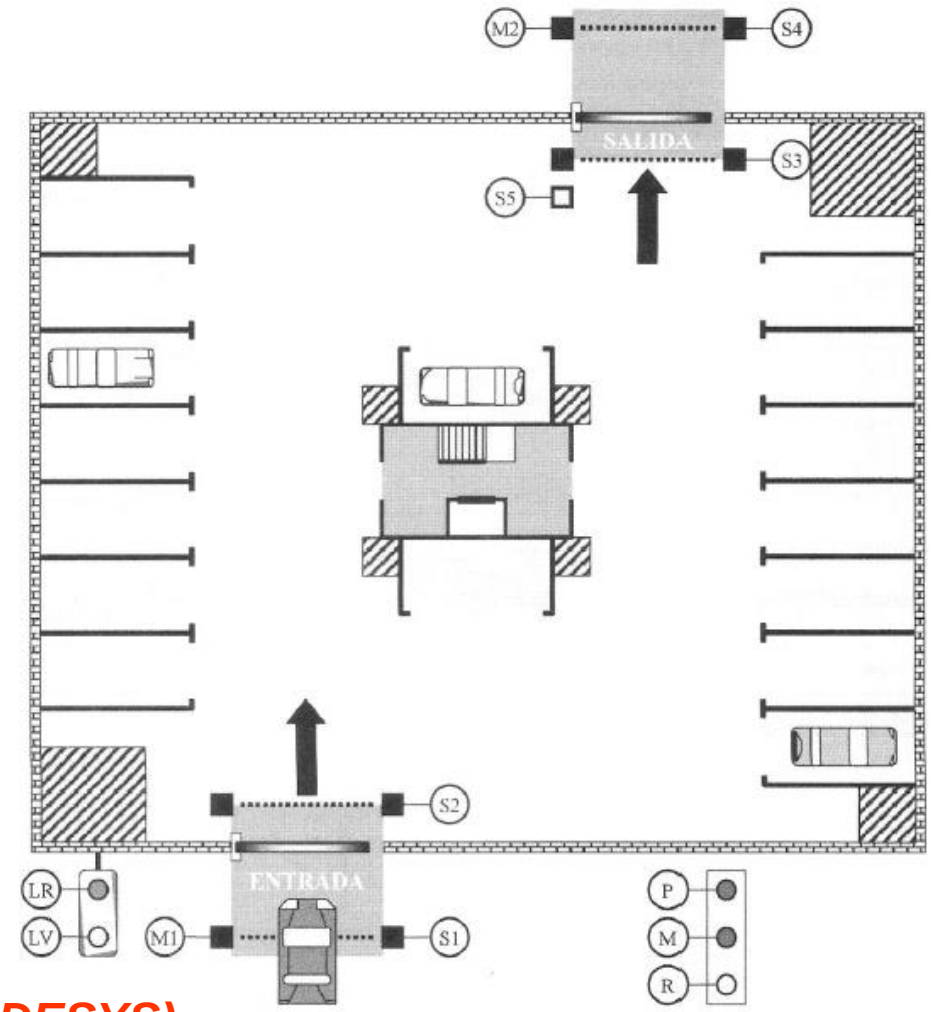
La capacidad del garaje es de 10 vehículos y el sistema electrónico debe controlar la ejecución de las siguientes acciones:

- *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 autómata 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 autómata.



Hacer:

0. Emulación de biestables SR salida

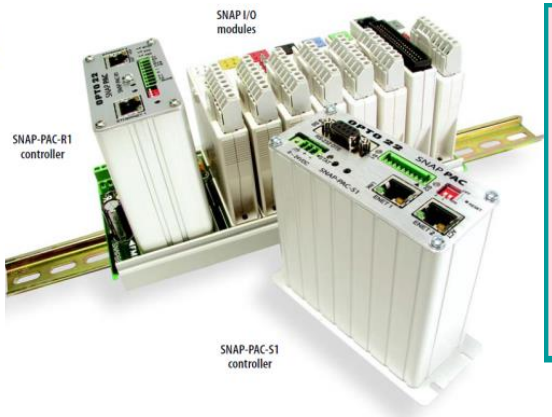
*1. Autómata: plantear **GRAFCET***

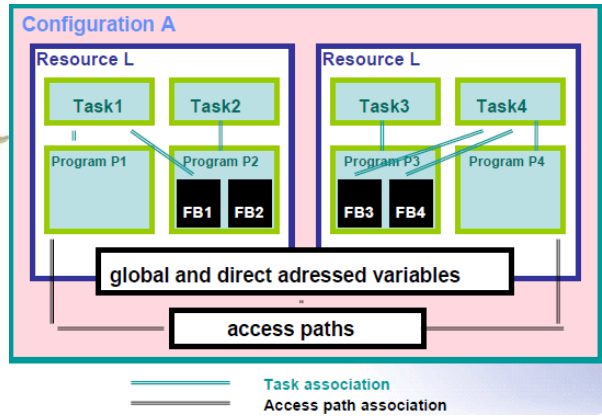
*2. **Partición en fases/etapas** → **ST (CODESYS)***

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

Unidad 3:

Diseño de Sistemas de Control Secuencial
Norma IEC 61131-3





Diseño de Sistemas de Control Secuencial

Plan de la Unidad 3:

A. Métodos Clásicos

- Proyectos de Automatización. Entornos de Desarrollo 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

- Descomposición jerárquica.
- Partición del Algoritmo en Fases.
- SFC/Grafcet: Diagramas de Función Secuenciales

C. Aplicaciones

- Uso de entornos de Desarrollo.
- Creación de un Proyecto. Configuración de Hardware. Estructura de Software.
- Resumen y Consultas.
- Próximos Pasos...

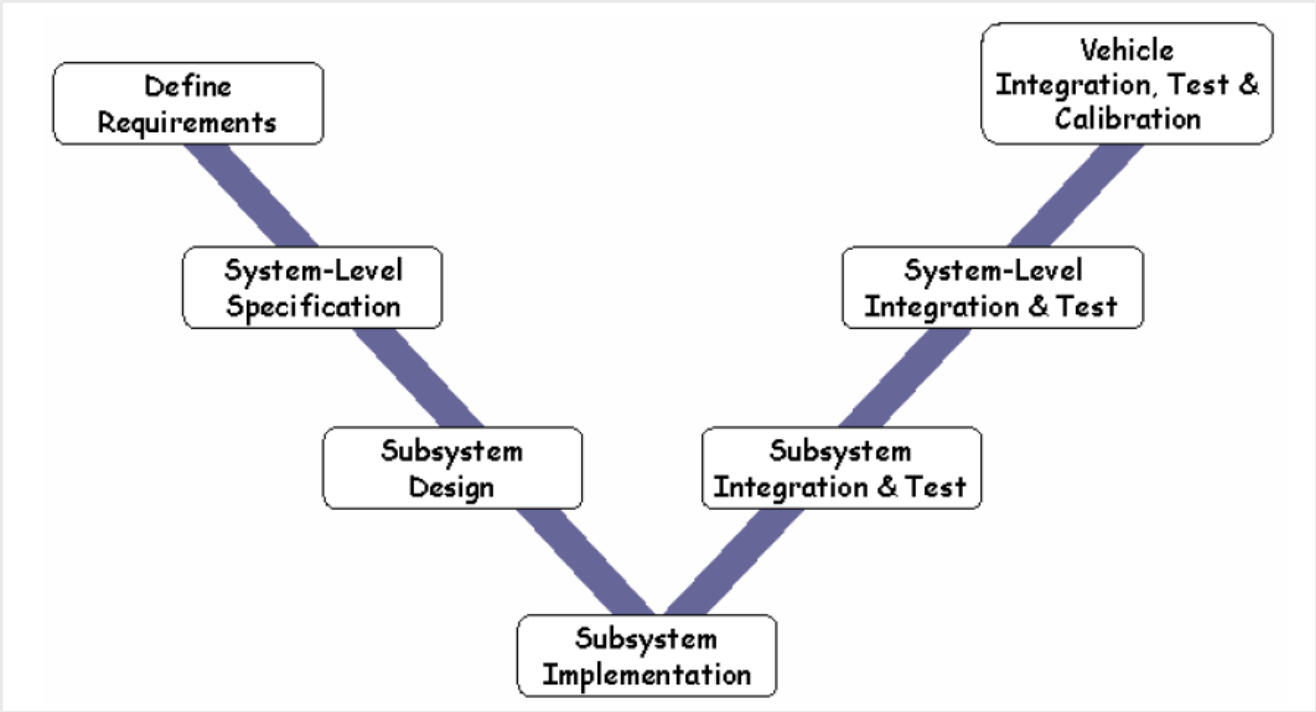
Práctica – Ejemplo: Model-Based Design

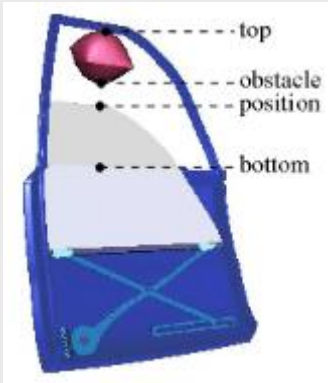
c. ***Sist. Control de Accionamiento para Ventanilla de Auto***:

- *The Interesting Engineering behind your **Car Window**!*
  Lesics: Video 2022-02-26 (4:40 min)
  https://www.youtube.com/watch?v=Yl40Uj8kCWU&ab_channel=Lesics

The interesting engineering behind your Car Window!

- ***Model-Based Design (V model –Simulink/Stateflow)***
  MathWorks





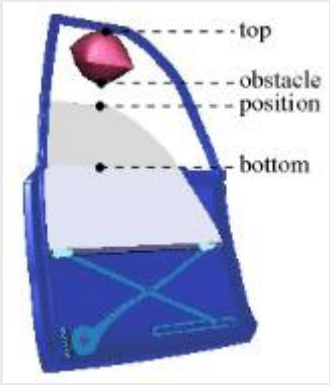
Práctica – Ejemplo: Model-Based Design

c. Sist. Control de Accionamiento para Ventanilla de Auto:

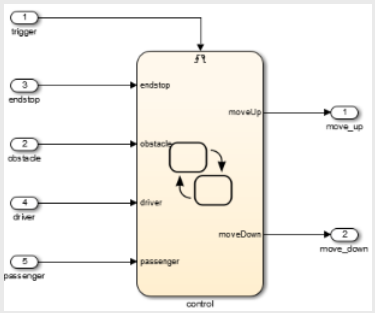
- *The Interesting Engineering behind your* ***Car Window****!*
  Lesics: Video 2022-02-26 (4:40 min)
  https://www.youtube.com/watch?v=Yl40Uj8kCWU&ab_channel=Lesics

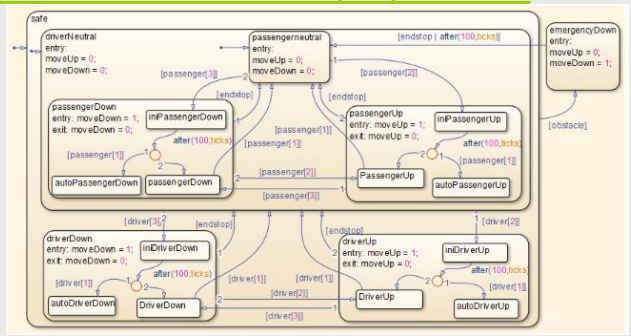
The interesting engineering behind your Car Window!

- ***Model-Based Design (V model –Simulink/Stateflow)***
  MathWorks: **Caso de Estudio - Proyecto**
  POWER WINDOW *System*
  https://www.mathworks.com/help/simulink/ug/power-window-example-case-study.html
  POWER WINDOW ***CONTROL*** *Project*
  https://www.mathworks.com/help/simulink/slref/power-window-control-project.html



Desarrollar en:
Simulink / Stateflow
Implementar en:
CODESYS SFC / ST





Referencias

- Mosterman, P., J. Sztipanovits, and S. Engell, "**Computer-Automated Multiparadigm Modeling** in **Control Systems Technology**," IEEE Transactions on Control Systems Technology, Vol. 12, Number 2, 2004, pp. 223–234.
- Prabhu, Sameer M. and Mosterman, Pieter J., "**Model-Based Design of a Power Window System: Modeling, Simulation, and Validation**"