

Memoria
1ª Parte de Robótica Y Automatización
Ascensor Simple Con Memoria, Puerta y Sistema de
Emergencia.
Ascensor Doble

Jesús Carrascosa Carro
jescarcar5@alum.us.es

5 de mayo de 2025

Índice

| | |
|---|----------|
| 1. Sobre el proyecto | 1 |
| 2. Ascensor Simple de 4 Plantas con Memoria, Puerta y Sistema de Evacuación: | 2 |
| 2.1. Funcionalidad: | 2 |
| 2.2. Componentes | 3 |
| 2.2.1. Acciones | 3 |
| 2.2.2. Bloques funcionales | 3 |
| 3. Ascensor Doble Con Memoria | 4 |
| 3.1. Funcionalidad: | 4 |
| 3.2. Componentes: | 4 |
| 3.2.1. Bloques funcionales: | 4 |
| 3.2.2. Funciones: | 5 |

1. Sobre el proyecto

El presente trabajo, realizado por Jesús Carrascosa Carro para la primera parte de la asignatura de Robótica y Automatización, consiste en la realización de un sistema de control para uno o dos ascensores de 4 plantas, dejando a elección del alumno:

- El número de ascensores.
- El número de plantas.
- La utilización de botoneras con memoria.
- La adición de otros sistemas adicionales como puertas, sistemas de emergencia . . .

Con el fin de maximizar la posible puntuación obtenida, se entregan 2 variantes, ambas de 4 plantas y con memoria:

1. Un ascensor simple de 4 plantas con puerta, memoria y sistema de evacuación.
2. Un ascensor doble de 4 plantas con memoria y sistema de asignación.

2. Ascensor Simple de 4 Plantas con Memoria, Puerta y Sistema de Evacuación:

2.1. Funcionalidad:

- Acceso a las cuatro plantas.
- Memoria, pudiendo recordar las llamadas.
- Parada al vuelo, si se pulsa el botón de una planta (sin distinción de cabina o botonera externa) y dicha planta se encuentra en la trayectoria actual del ascensor (sin haber pasado todavía por dicha planta), el ascensor se parará en la planta solicitada.
- Sistema de puertas, representada en la simulación por un cambio de color(verde) de la cabina del ascensor, estas se abren durante un tiempo determinado(3 segundos, modificable mediante la constante global¹) al llegar a la planta estacionada. Si el ascensor se encontrase en una planta estacionado, con las puertas cerradas y sin destino asignado, las puertas se pueden abrir pulsando el botón de la planta en la que se encuentre estacionado.
- Sistema de evacuación en caso de emergencia: El ascensor, en caso de emergencia(simulado con un botón), inmediatamente borrará todos los destinos de su memoria, apagando las luces de planta y descenderá a la planta cero donde mantendrá sus puertas abiertas hasta que se desactive la emergencia(mediante el botón de rearme², que solo surte efecto cuando el ascensor se encuentre en la planta cero). Inspirado en las medidas contra incendio de los ascensores [fain](#)
- Preservación del sentido. En caso de múltiples destinos se escogerá aquel que continúe el sentido del ascensor(subida o bajada). Es decir, si el ascensor habiendo descendido a la planta dos, tiene como opciones subir a la planta tres o bajar a la planta uno, bajará a la planta uno.
- Diferenciación de pulsaciones internas y externas: Las luces de la cabina y de la botonera externa indican si se ha presionado un botón en la cabina o en la botonera externa, encendiéndose la luz correspondiente. Si se presiona un botón en la cabina y también uno externo para la misma planta, se iluminarán las luces correspondientes a esa planta, tanto en el interior como en el exterior.

¹Curiosamente en una de las prácticas entregables, codesys no me dejó crear una "variable" de tipo constante, sin embargo, esta vez sí me ha dejado

²La conexión del botón de rearme a un detector de flanco de bajada es intencional, ya que evita que, mientras el ascensor desciende en modo de emergencia, alguien mantenga el botón de rearme presionado. Esto garantiza que la emergencia no se desactive automáticamente al llegar a la planta cero, sino que se requiera volver a presionar el botón.

2.2. Componentes

2.2.1. Acciones

Se han definido 7 acciones:

1. **inicio:** Escrita en ST, se encarga de instanciar el decodificador de planta, y pulsador de planta.
2. **Comp_Boton_Planta:** Escrita en ST, define el funcionamiento de los botones, es decir, determina si al pulsar el botón correspondiente a una planta, esta se añade a la lista de destinos. Tal y como está programado no añadirá una planta si esta es coincidente con la actual o hay una emergencia.
3. **Comp_btn_emer:** Escrita en LD, describe el funcionamiento de los botones de rearme y emergencia.
4. **Seleccion:** Escrita en ST, describe el funcionamiento del algoritmo de selección de sentido.
5. **Luces:** Escrita en LD, describe el funcionamiento de las luces de cabina y de la botonera externa
6. **llegada:** Escrita en ST, y activado al llegar a una planta, se encarga del apagado de las luces correspondientes y de quitar de la lista de destinos dicha planta.
7. **EmerAction:** Escrito en ST, describe parte de las acciones que se realizan en caso de emergencia (vaciado de la lista de destinos/memoria, apagado de luces ...)

2.2.2. Bloques funcionales

Se han descrito 3 bloques funcionales:

1. **Controlador:** Escrito en grafcet, tiene como función controlar el funcionamiento del ascensor, abarcando las funciones principales:
 - Subida o bajada del ascensor.
 - Comportamiento en caso de emergencia.
 - Selección de sentido.
 - Apertura de puertas.
 - Interpretación de la botonera.

El controlador se instancia en el programa Ascensor.

2. **PulsaPlanta:** Escrito en LD, se emplea para determinar si se ha pulsado el botón (tanto el de cabina como el de la botonera externa) correspondiente a la planta en la que se encuentra el ascensor estacionado y abrir la puerta en caso de que estuviera cerrada.
3. **PlantaDec:** Escrito en ST, se utiliza para averiguar en que planta está el ascensor. Debido a que el margen de detección de los sensores de planta es estricto (el ascensor ha de estar enteramente en el área de la planta para activarlo), si por cualquier razón ninguno de los sensores está activo devuelve una planta inexistente.

3. Ascensor Doble Con Memoria

3.1. Funcionalidad:

- Acceso a las cuatro plantas.
- Memoria, pudiendo recordar las llamadas.
- Parada al vuelo, si se pulsa el botón de una planta (sin distinción de cabina o botonera externa) y dicha planta se encuentra en la trayectoria actual del ascensor (sin haber pasado todavía por dicha planta), el ascensor se parará en la planta solicitada.
- Preservación del sentido a nivel individual. En caso de múltiples destinos posibles, el ascensor escogerá aquel que continúe el sentido del ascensor(subida o bajada). Es decir, si el ascensor habiendo descendido a la planta dos, tiene como opciones subir a la planta tres o bajar a la planta uno, bajará a la planta uno.
- Diferenciación de pulsaciones internas y externas: Las luces de la cabina y de la botonera externa indican si se ha presionado un botón en la cabina o en la botonera externa, encendiéndose la luz correspondiente. Si se presiona un botón en la cabina y también uno externo para la misma planta, se iluminarán las luces correspondientes a esa planta, tanto en el interior como en el exterior.
- Balanceo de asignaciones externas. En el caso de que ambos o ninguno de los ascensores estuvieran disponibles³, se le asignará la planta a aquel que no le fuera asignada anteriormente una planta(se dan turnos).

Se ha planteado un sistema de emergencia similar al descrito para el ascensor simple, pero no se ha implementado.

3.2. Componentes:

3.2.1. Bloques funcionales:

1. **Ascensor:** Escrito en grafcet, muy similar al controlador del ascensor simple, pero con su funcionalidad limitada a lo siguiente:
 - Selección del sentido.
 - Control de los motores.
 - Encendido o apagado de las luces de cabina.
 - Control de la botonera interna, ya que la asignación de plantas de la botonera externa es realizada por otro bloque funcional.

Incluye las siguientes acciones:

- a) **Cod_Sent_act:** Escrita en ST, describe el algoritmo de selección de sentido.
- b) **Comp_btn:** Escrita en ST, se encarga de la adición de nuevos destinos, tanto los provenientes de la botonera interna como del controlador de asignación.
- c) **inicialización e Inicializacion2:** Escritas en ST y LD respectivamente, se encargan tanto de la instanciación del bloque funcional planta actual como del comportamiento de las luces.

³Se dice que un ascensor está disponible si no tiene destinos o su sentido es favorable

- d) llegada_act*: Escrita en ST, y activada al pararse el ascensor en una planta, apaga las luces de la cabina correspondientes a dicha planta.
- 2. **Controller**: Escrito en Grafset, se encarga de la asignación de plantas provenientes de la botonera externa como del balanceo de asignaciones según la disponibilidad anteriormente descrita. Incluye las siguientes acciones:
 - **comp_luces, inicialización e ini2**: Se encargan tanto del comportamiento de las luces como la instanciación de los PulsadorDecoder.
 - **add_lista**: Añada a la lista de destinos interna de controller la planta solicitada.
- 3. **PulsadorEncoder y PulsadorDecoder**: Ambos escritos en ST, PulsadorEncoder si está activo, convierte el botón pulsado en un entero. PulsadorDecoder realiza la función contraria, convirtiendo un entero en un "botón".
- 4. **PlantaActual_FB**: Escrito en ST, tiene la misma función que PlantaDec del ascensor simple.

3.2.2. Funciones:

- 1. **ya_destino**: Escrita en ST, indica si alguno de los ascensores ya tiene asignada una planta en su lista de destinos.
- 2. **disponible**: Escrita en ST, indica si un ascensor está disponible⁴, es decir, si no tiene destinos o si su sentido es favorable.

⁴Técnicamente, aunque no causa ningún error grave, la forma de calcular la disponibilidad no es del todo correcta, ya que si el ascensor se encuentra en movimiento, su planta actual será la inexistente planta 4.