

Controlaores lógicos Programables

Profesor: Carlos E. Morán Garabito

Alumnos:

Medina Rodríguez Francisco Javier

Martínez Noyola Moisés Emanuel

5°A Ing. Mecatrónica

Reporte Práctica 3

Sistema de Control de Acceso Vehicular

**Introducción:**

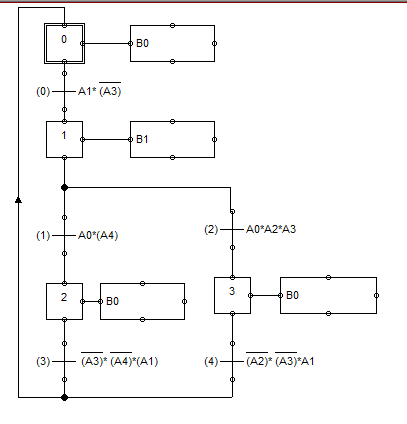
[](https://www.google.com.mx/url?sa=i%26rct=j%26q=%26esrc=s%26source=images%26cd=%26cad=rja%26uact=8%26ved=2ahUKEwi59dqylOvgAhVNXKwKHbYPCdkQjRx6BAgBEAU%26url=https:/articulo.mercadolibre.com.mx/MLM-556298587-barrera-vehicular-de-control-de-acceso-pluma-de-45-metros-_JM%26psig=AOvVaw0iiyXPLP-f4f00NNQU2boH%26ust=1551880710486843)

|  |  |
| --- | --- |
| [IMG_256](https://www.google.com.mx/url?sa=i%26rct=j%26q=%26esrc=s%26source=images%26cd=%26cad=rja%26uact=8%26ved=2ahUKEwjczt6nlOvgAhUBbK0KHY3CCzoQjRx6BAgBEAU%26url=http:/wikifab.dimf.etsii.upm.es/wikifab/index.php/Sistema_de_Control_de_un_Aparcamiento_08001%26psig=AOvVaw0iiyXPLP-f4f00NNQU2boH%26ust=1551880710486843) | [IMG_256](https://www.google.com.mx/url?sa=i%26rct=j%26q=%26esrc=s%26source=images%26cd=%26cad=rja%26uact=8%26ved=2ahUKEwj9-KL-levgAhVwmK0KHVVyDJ0QjRx6BAgBEAU%26url=http:/itsmyblogmechatronics.blogspot.com/2015/04/car-park-barrierscoin-counters.html%26psig=AOvVaw2CjaFyB-VdiV575Ue4S674%26ust=1551881159203505) |

**Desarrollo:**

Diseño del Grafcet correspondiente.

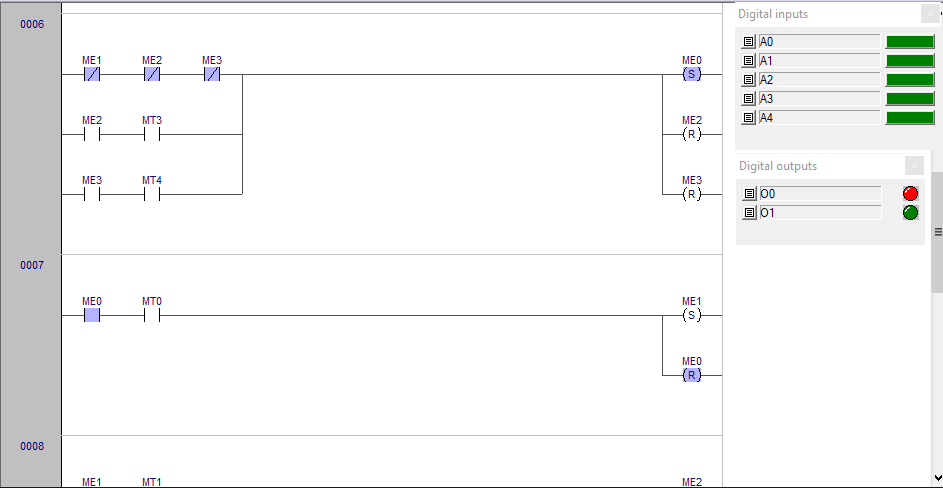
|  |  |  |  |
| --- | --- | --- | --- |
| Identificador | Sensores | Identificador | Actuadores |
| A0 | Pluma Abajo | O0 | Levantar pluma |
| A1 | Pluma Arriba | O1 | Bajar pluma |
| A2 | Sensor tag |  | |
| A3 | Sensor de coche |
| A4 | Botón del guardia |



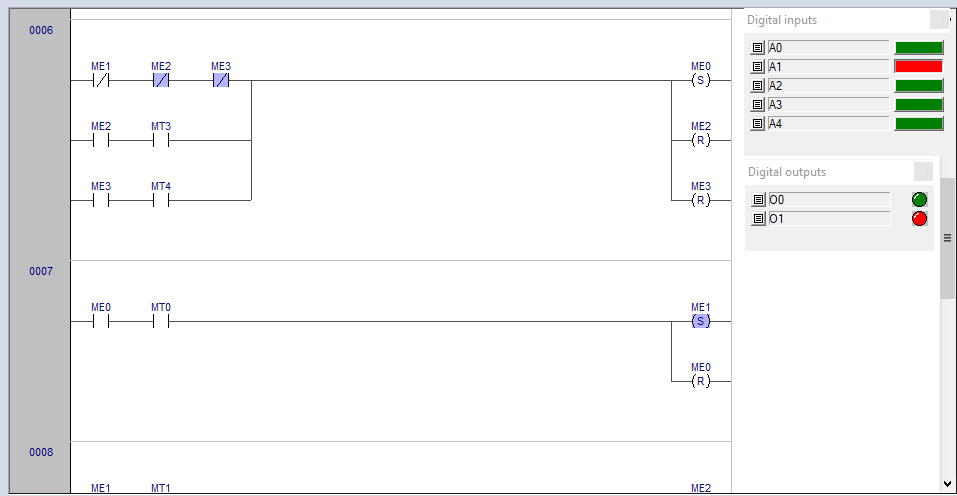
Este diagrama resulta muy útil a la hora de simular. Gracias a él nos damos cuenta en cuál estado y transición nos encontramos.

**Simulación**

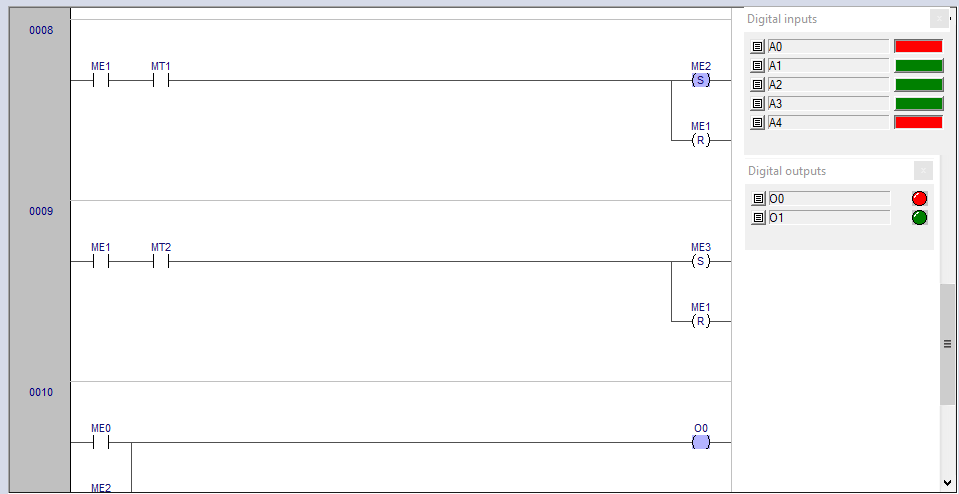
Estado 0 (Home): Apreciamos la salida en el actuador O0 (levantar pluma).



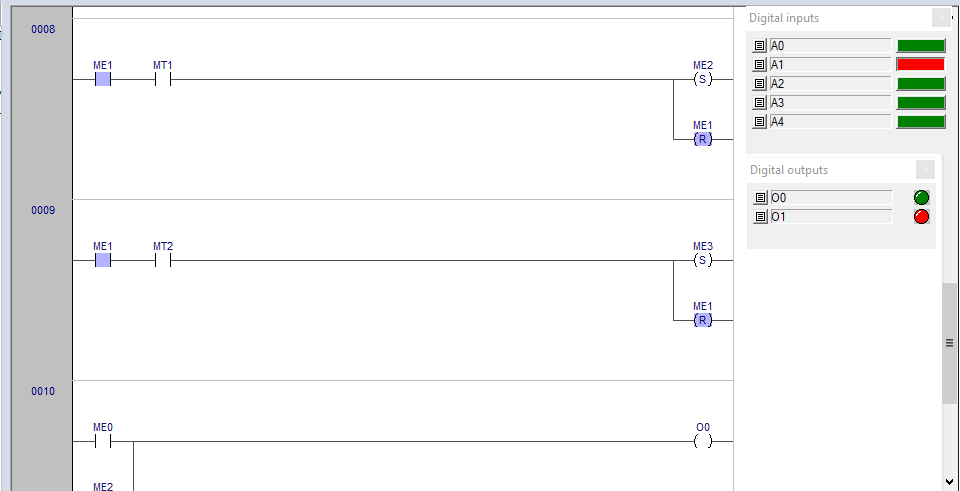
Estado 1: Activar el sensor A1 y negar el A3, da paso al estado 1, el cual activa el actuador B1 (Bajar pluma).



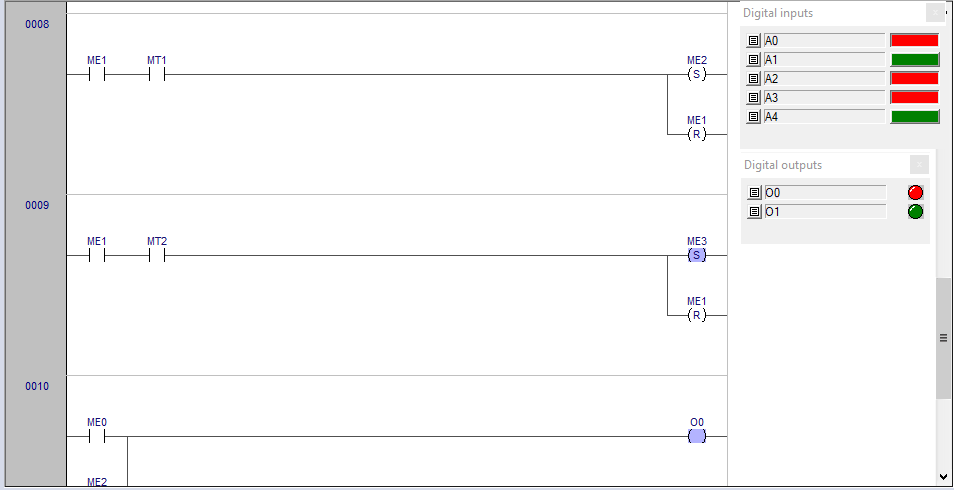
Estado 2: Este representa al guardia activando el botón (A4), y si la pluma está abajo (A0), Entonces subirá la pluma (actuador O0)



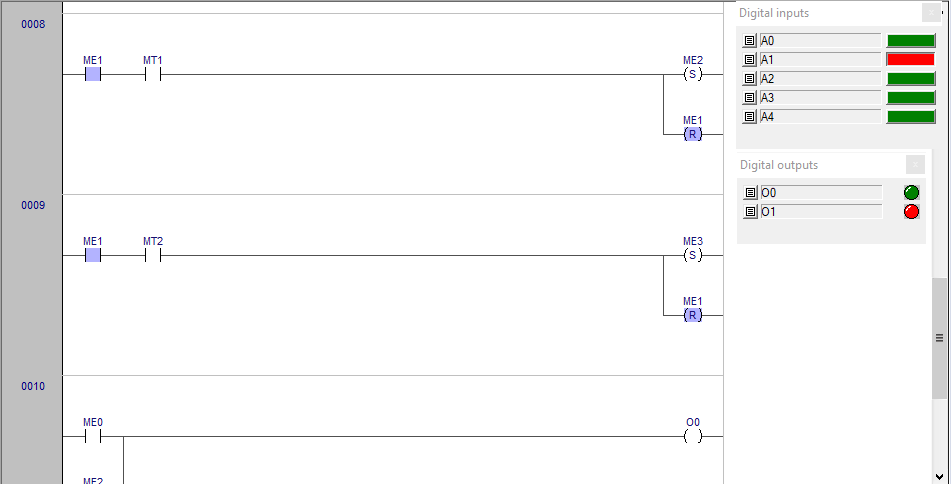
Regreso a estado 1: cuando la pluma esté arriba (A1), El guardia deje de presionar el botón (A4 inactivo) y no haya coche (sensor A3 desactivado), entonces bajará la pluma (actuador O0). De esta forma se podrá reiniciar el ciclo.



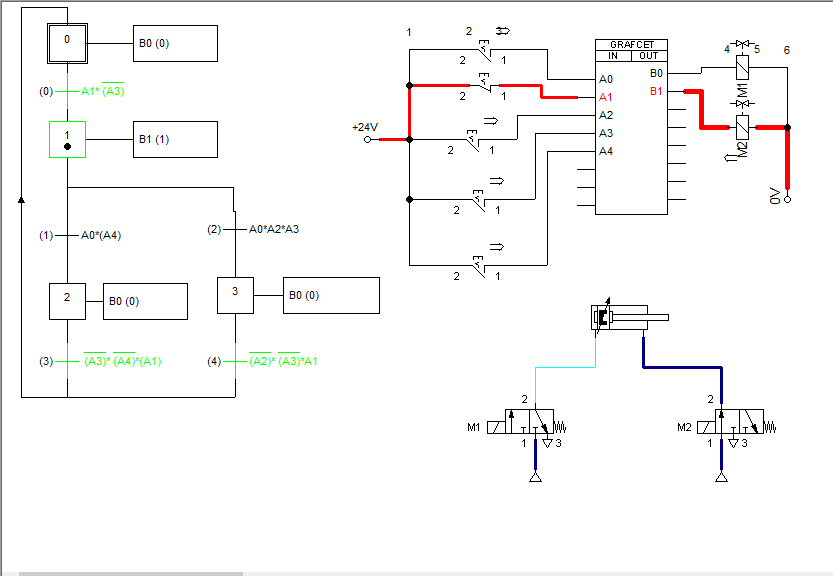
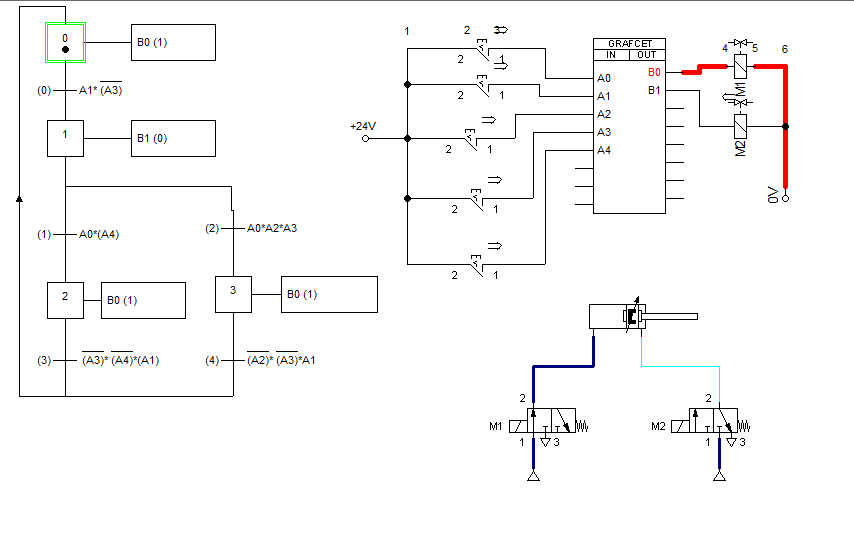
Estado 3: Para este caso, se necesitará un tag en un coche residente (A2), sensor de proximidad del mismo (A3) y que la pluma esté abajo (A0) para poder levantarla (O0). Nótese que se hace automáticamente sin necesitar un guardia para presionar el botón.

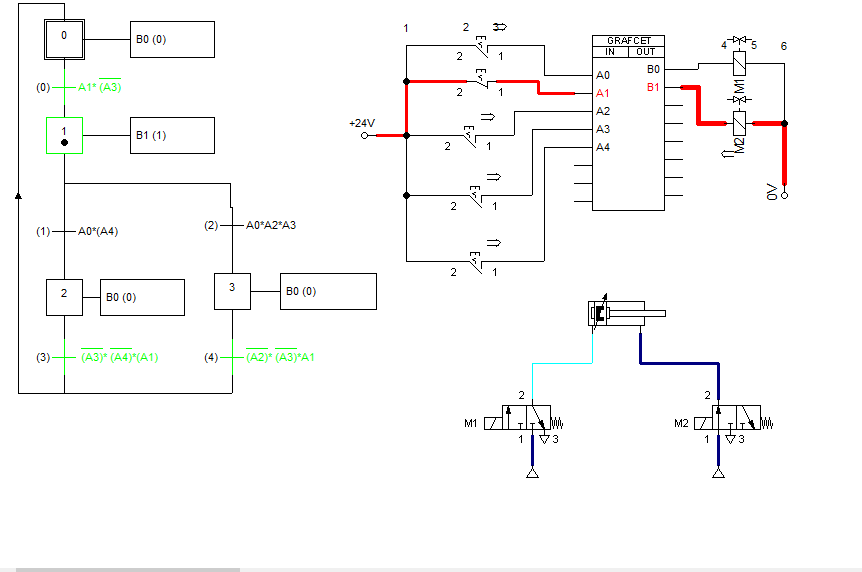
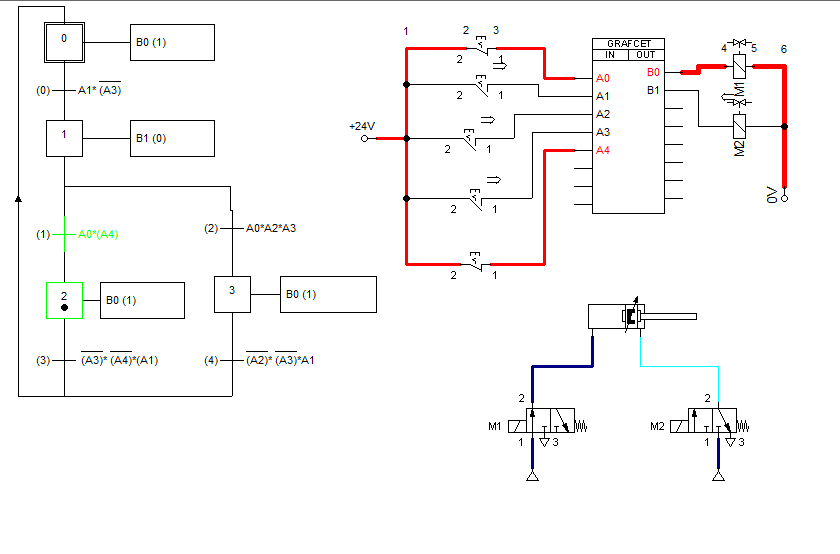


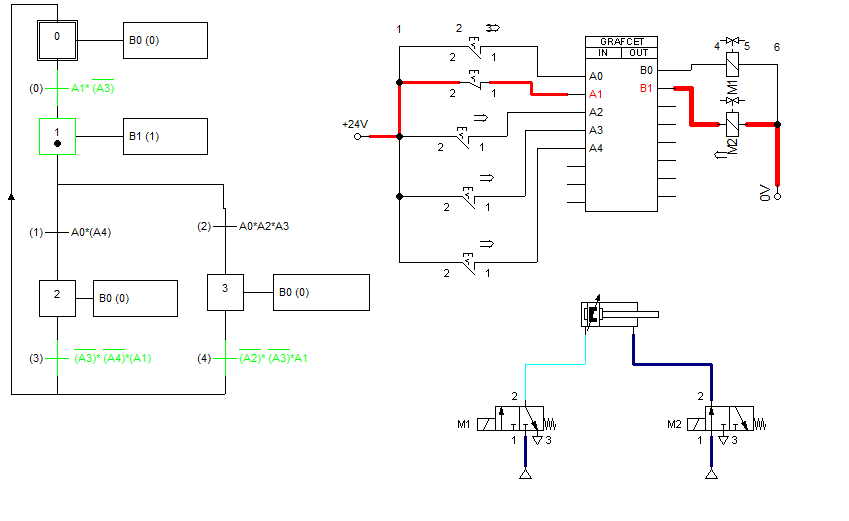
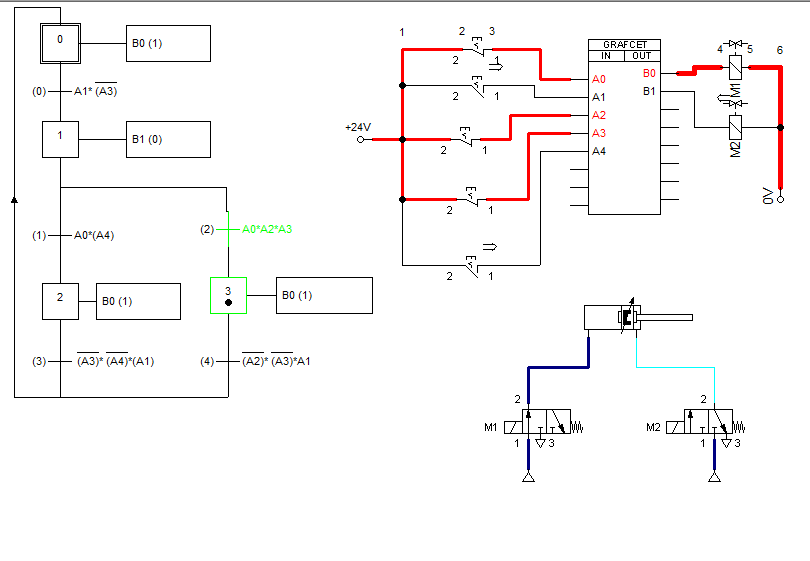
Regreso a estado 1: Para este se necesita que los sensores de tag (A2) y el de proximidad del coche (A3) estén desactivados y que la pluma esté arriba (A1)



Capturas de la simulación en FluidSim:







**Conclusiones**

Básicamente, el GRAFCET es un modelo de representación gráfica, de los sucesivos comportamientos de un sistema lógico, predefinido por sus entradas y salidas. También es un [grafo](https://es.wikipedia.org/wiki/Grafo), o diagrama funcional normalizado, que permite hacer un modelo del proceso a automatizar, contemplando entradas, acciones a realizar, y los procesos intermedios que provocan estas acciones. Inicialmente fue propuesto para documentar la etapa secuencial de los sistemas de control de procesos a eventos discretos. No fue concebido como un lenguaje de programación de autómatas, sino un tipo de grafo para elaborar el modelo pensando en la ejecución directa del automatismo o programa de autómata.

Varios fabricantes en sus autómatas de gama alta hacen este paso directo, lo que lo ha convertido en un potente lenguaje gráfico de programación para [autómatas](https://es.wikipedia.org/wiki/Aut%C3%B3mata_programable), adaptado a la resolución de sistemas secuenciales. En la actualidad no tiene una amplia difusión como lenguaje, puesto que la mayoría de los autómatas no pueden programarse directamente en este lenguaje, a diferencia del lenguaje Ladder. Pero se ha universalizado como herramienta de modelado que permite el paso directo a programación, también con Ladder.

