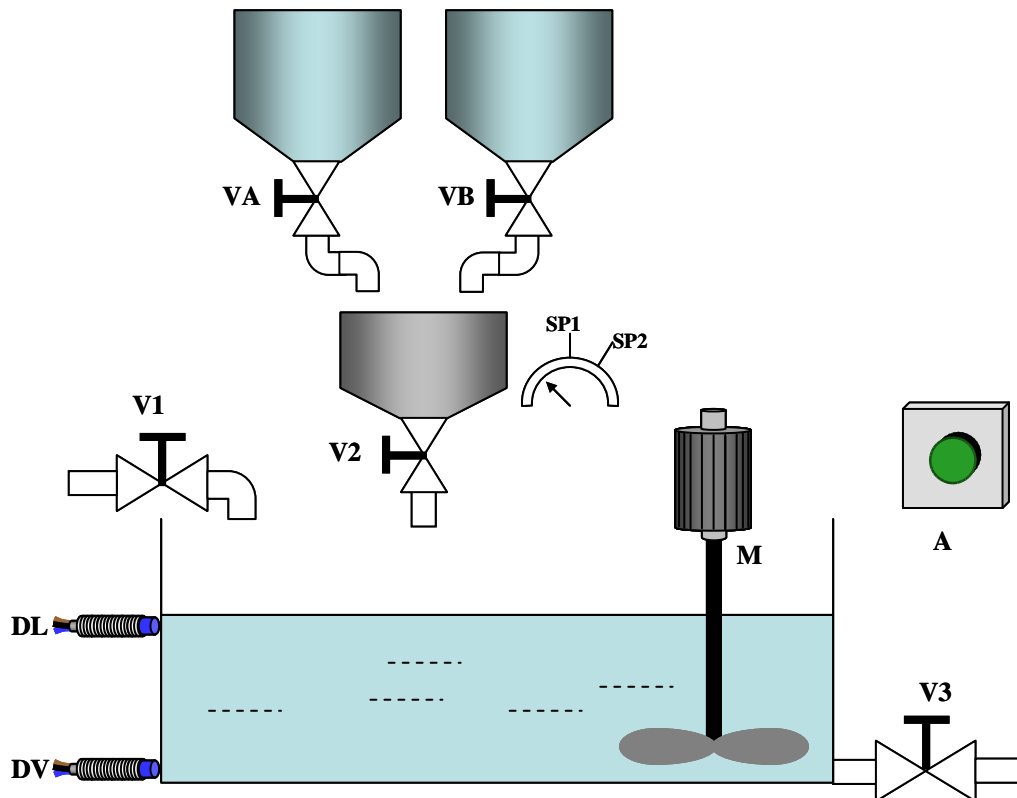


PROBLEMA: MEZCLADORA

Se quieren mezclar 2 productos con agua. Se llena el depósito de agua abriendo la válvula V1. La dosificación de los dos productos se realiza con una tolva acumulativa, se vierte el producto A sobre la tolva hasta que se alcanza un peso SP1 y a continuación se añade el producto B para conseguir el peso total de los 2 productos, SP2. Se abre la válvula de la tolva durante 10 segundos para dejar caer el contenido. Se realiza el proceso de mezclado durante 30 segundos accionando el agitador y se vacía el depósito para poder iniciar un nuevo ciclo. El proceso se activa con un interruptor P.



Se pide Realizar la programación en GRAFCET del controlador del automatismo, programando sus transiciones y las acciones de las etapas en Lenguaje de Contactos.

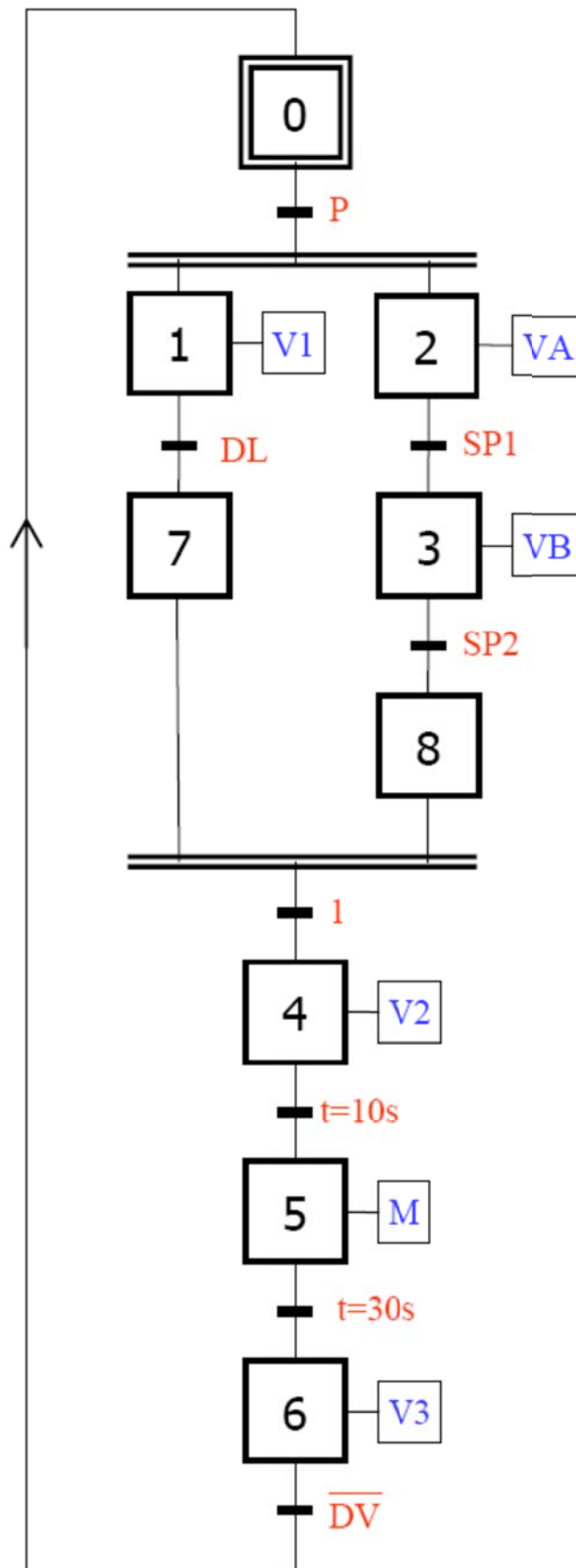
Solución 1:

Entradas:

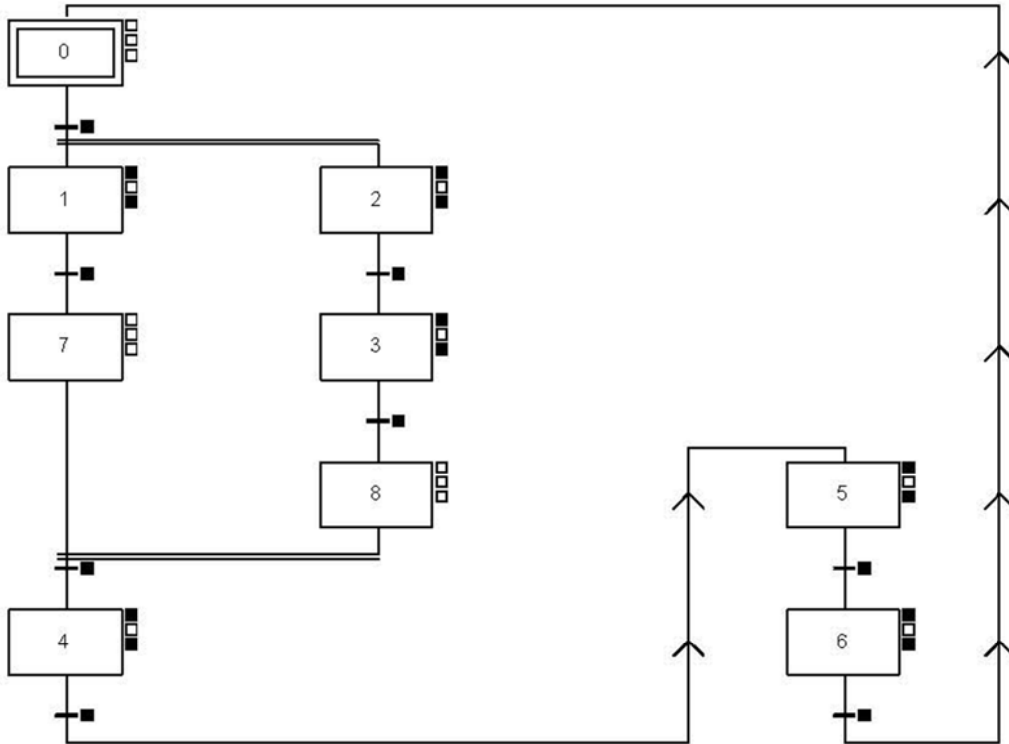
P %I1.0
DL %I1.1
DV %I1.2
SP1 %I1.3
SP2 %I1.4

Salidas:

V1 %Q2.0
V2 %Q2.1
V3 %Q2.2
VA %Q2.3
VB %Q2.4
M %Q2.5

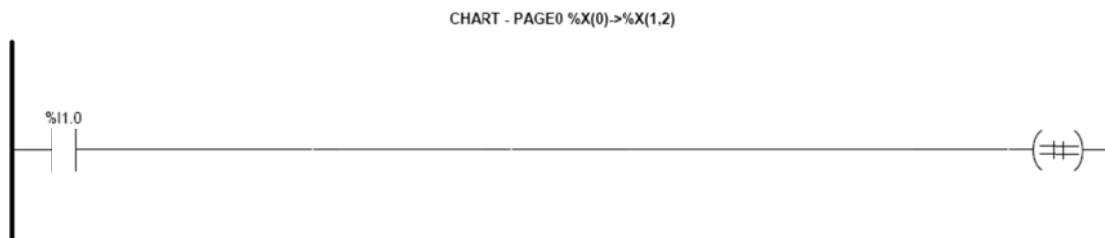


En primer lugar debemos programar el diagrama funcional en grafcet. Para ello tomaremos los elementos gráficos de la paleta de programación y los iremos situando en el entorno grafico de programación. El grafcet queda de la siguiente manera:



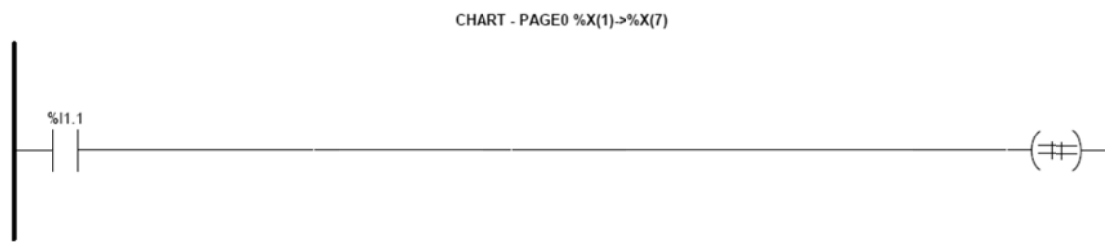
Programamos las transiciones: Las transiciones se programan poniendo la condición de cambio de etapa, seguida de la bobina de transición (\Rightarrow)

Por ejemplo, la transición desde la etapa 0 a las etapas 1 y 2, se programa de la siguiente forma:

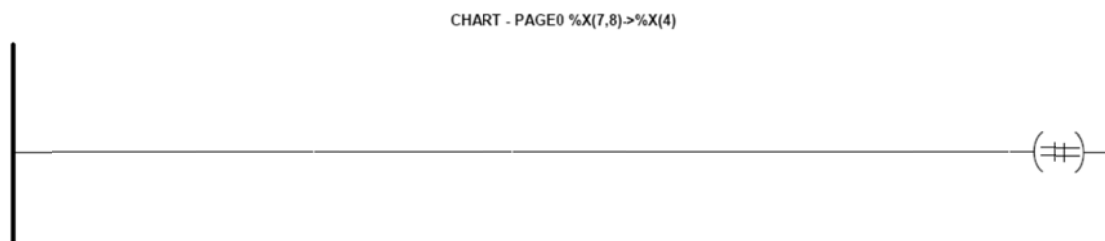


Hacemos lo mismo con el **resto de transiciones**:

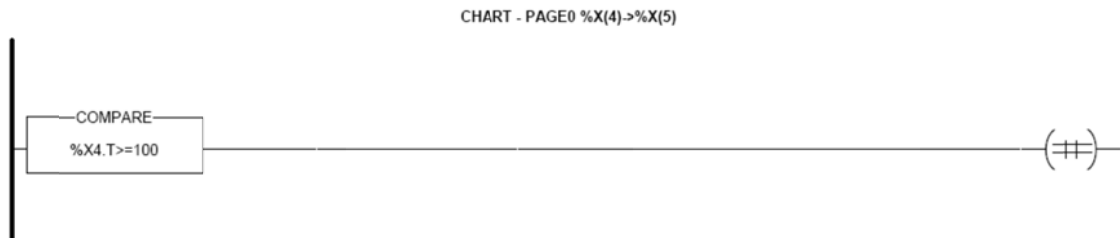
Etapa 1 a Etapa 7:



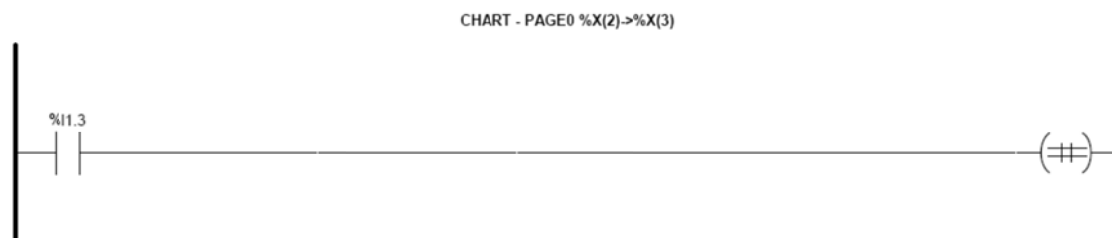
Etapa 1 a Etapa 7:



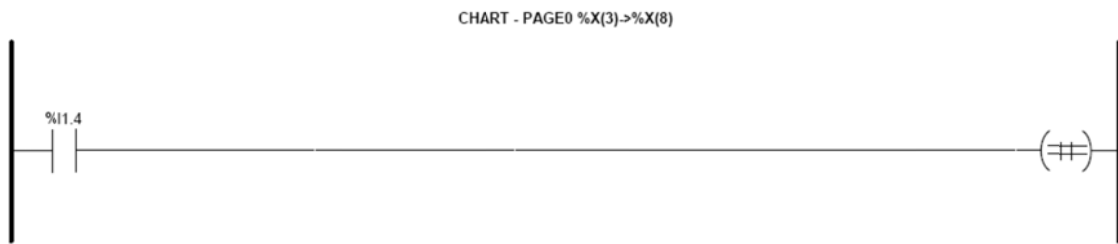
Etapa 4 a Etapa 5: Esta transición debe realizarse cuando la etapa 4 esté activa durante 10 sg. Para ello utilizaremos la palabra que almacena el tiempo que dicha etapa permanece activa (%Xi.T). Hay que tener en cuenta que el tiempo base de estas palabras es de 100 ms.



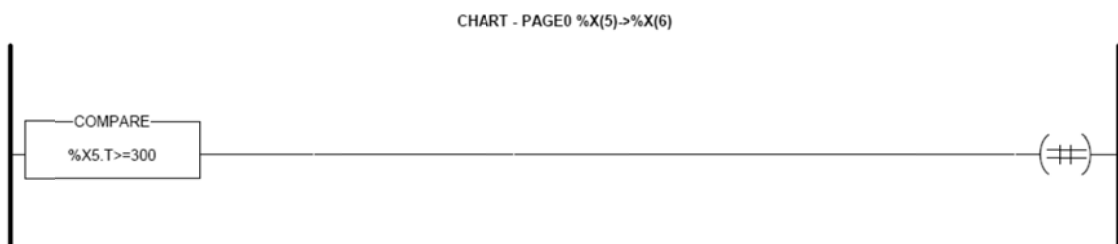
Etapa 2 a Etapa 3:



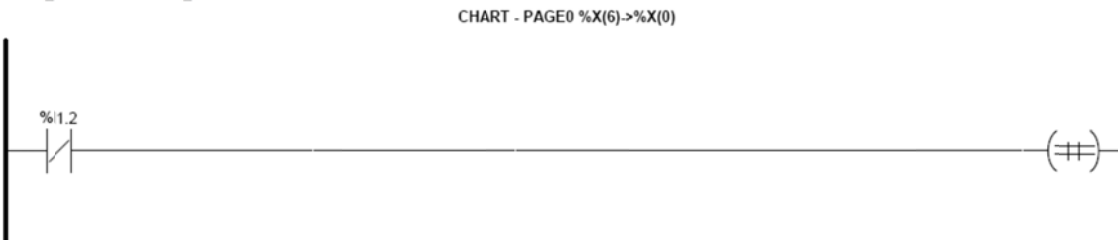
Etapa 3 a Etapa 8:



Etapa 5 a Etapa 6: Esta transición debe realizarse cuando la etapa 5 esté activa durante 30 sg. Para ello utilizaremos la palabra que almacena el tiempo que dicha etapa permanece activa (%Xi.T). Hay que tener en cuenta que el tiempo base de estas palabras es de 100 ms.



Etapa 6 a Etapa 0:

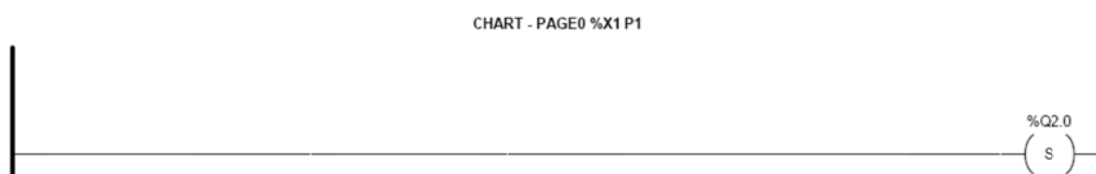


Programación de las etapas:

Cada una de las etapas se puede programar al activarse, en continuo, o en la desactivación de la etapa. En nuestro caso programaremos la activación de la salida (la poemos a una), en la acción a la activación (s) y las desactivamos en la desactivación de la etapa (R) .

Por ejemplo, en la etapa uno, debemos programar las acciones de activación y desactivación de la salida..., como se muestra en las figuras:

Etapa 1: Activación



Etapa 1: Desactivación

CHART - PAGE0 %X1 P0



Etapa 4: Activación

CHART - PAGE0 %X4 P1



Etapa 4: Desactivación

CHART - PAGE0 %X4 P0



Etapa 2: Activación

CHART - PAGE0 %X2 P1



Etapa 2: Desactivación

CHART - PAGE0 %X2 P0



Etapa 3: Activación

CHART - PAGE0 %X3 P1



Etapa 3: Desactivación

CHART - PAGE0 %X3 P0



Etapa 5: Activación

CHART - PAGE0 %X5 P1



Etapa 5: Desactivación

CHART - PAGE0 %X5 P0



Etapa 6: Activación

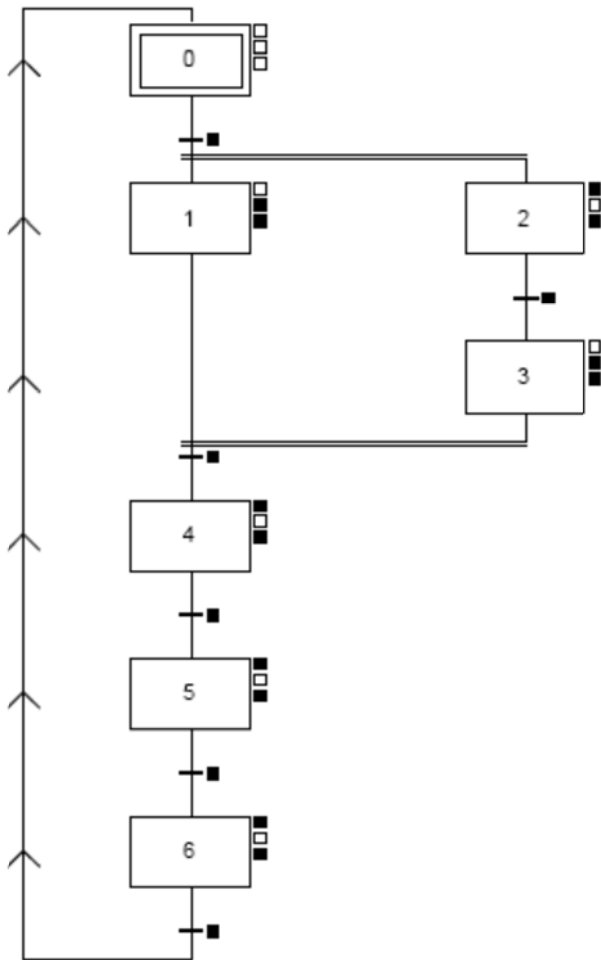
CHART - PAGE0 %X6 P1



Etapa 6: Desactivación

CHART - PAGE0 %X6 P0





Programamos las transiciones: Las transiciones se programan poniendo la condición de cambio de etapa, seguida de la bobina de transición (\Rightarrow)

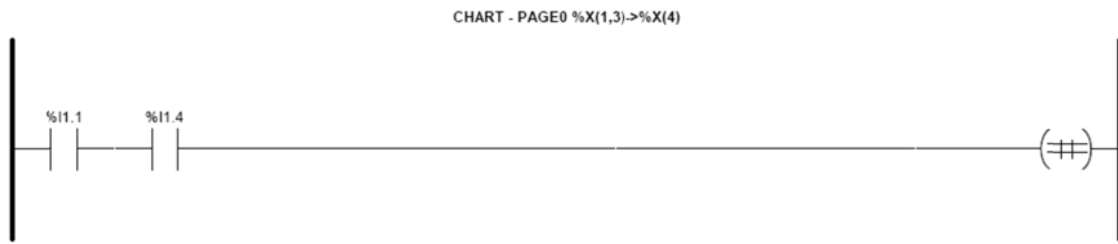
Siguiendo el mismo esquema que en la solución anterior:

Etapa 0 a Etapa 1 y Etapa 2:

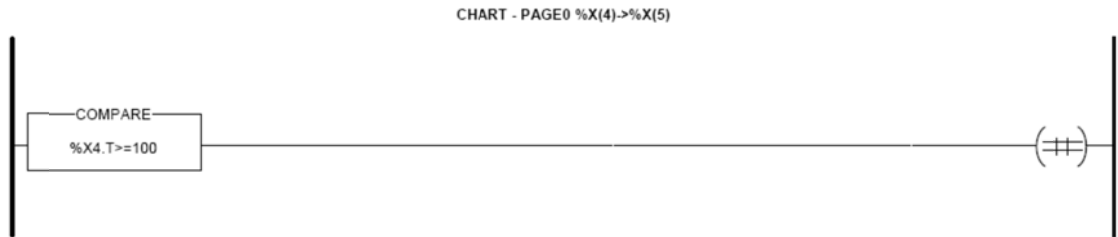
CHART - PAGE0 %X(0)->%X(1,2)



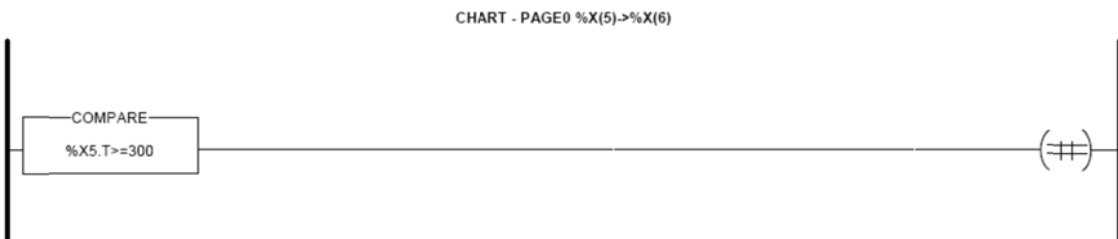
Etapa 0 y 3 a Etapa 4:



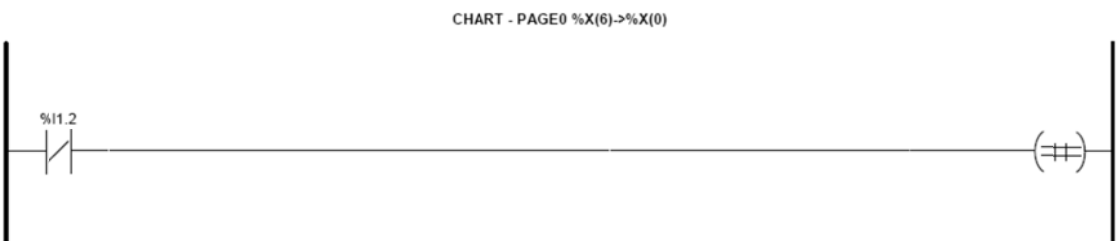
Etapa 4 a Etapa 5:



Etapa 5 a Etapa 6:



Etapa 6 a Etapa 0:



Etapa 2 a Etapa 3:

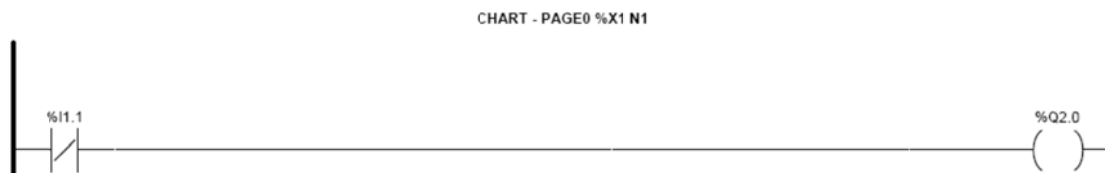


Programación de las etapas:

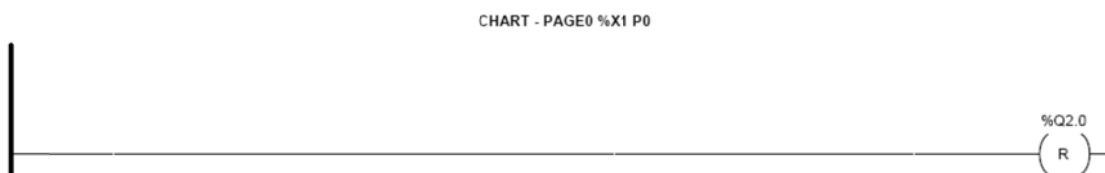
De la misma manera que en la solución anterior, cada una de las etapas se puede programar al activarse, en continuo, o en la desactivación de la etapa.

En este caso, la diferencia está en que en las etapas 1 y 3, tal y como aparece en el graficet, están programadas al desactivarse y en continua. En el caso de la etapa 1, la programación en continua se hace ya que la salida se puede activar o desactivar durante esa etapa, dependiendo del valor de una entrada en concreto.

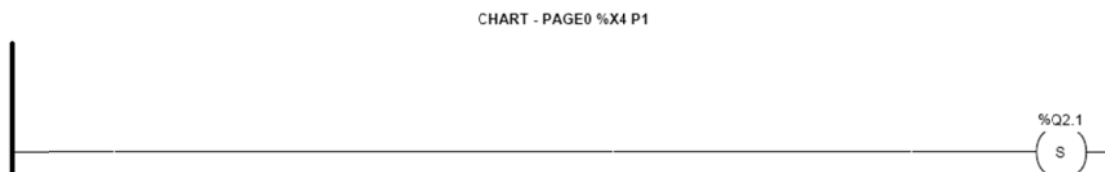
Etapa 1: Continua



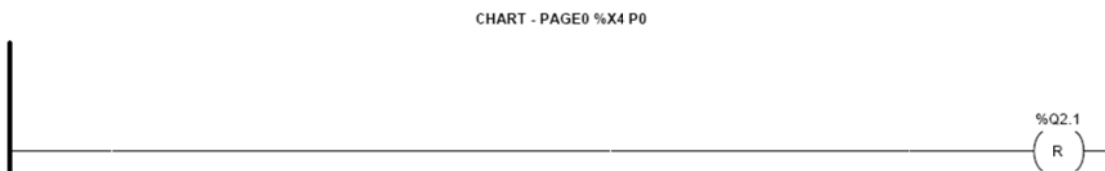
Etapa 1: Desactivación



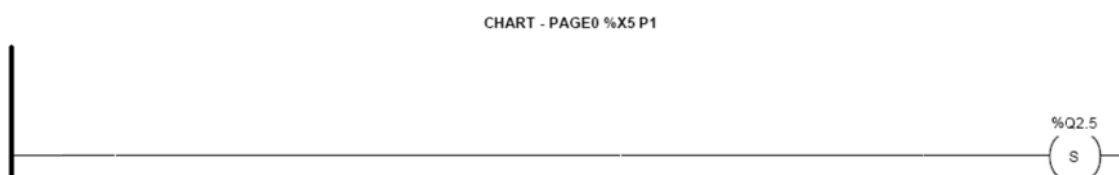
Etapa 4: Activación



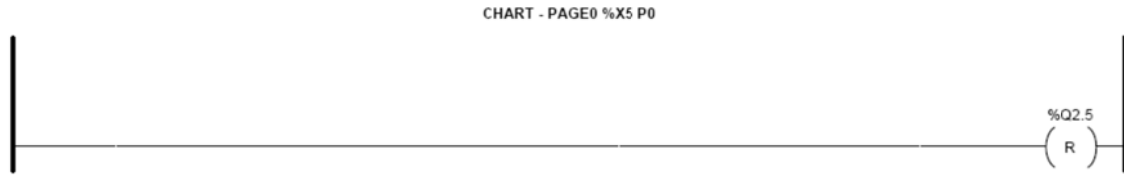
Etapa 4: Desactivación



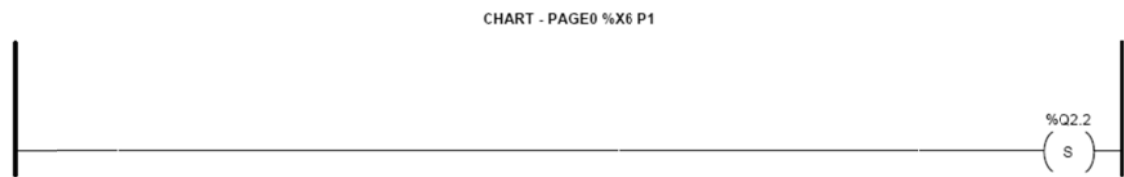
Etapa 5: Activación



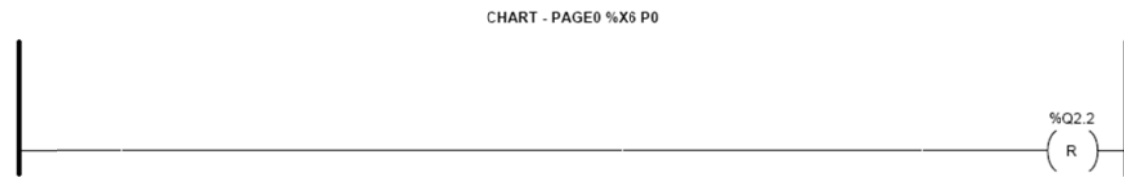
Etapa 5: Desactivación



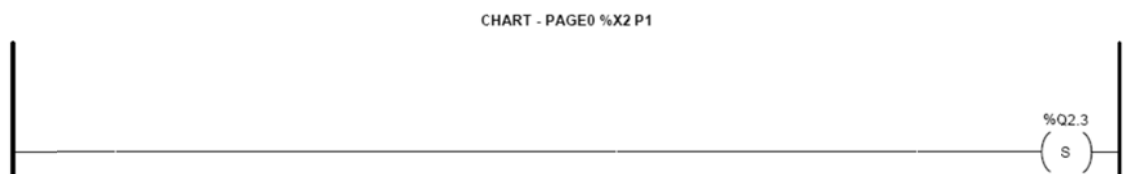
Etapa 6: Activación



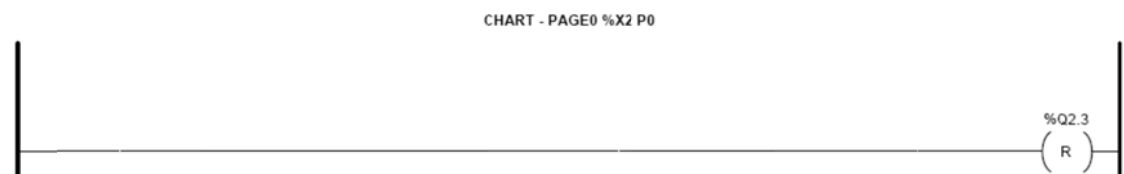
Etapa 6: Desactivación



Etapa 2: Activación



Etapa 2: Desactivación



Etapa 3: Continua.

En esta etapa, se programa en continua ya que el valor de la salida puede cambiar durante el tiempo que la etapa esté activa.



Etapa 3: Desactivación

