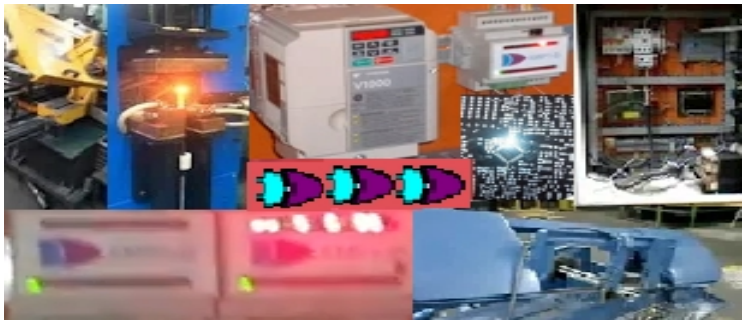
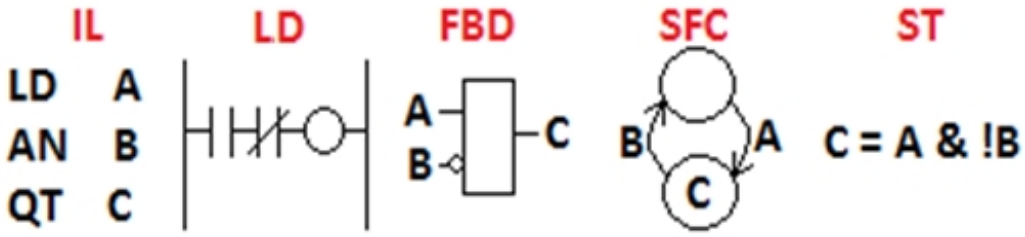
DESARROLLO DE PROGRAMAS DE CONTROL LÓGICO UTILIZANDO MÁQUINAS DE ESTADO

Conceptos fundamentales del control lógico utilizando máquinas de estado.

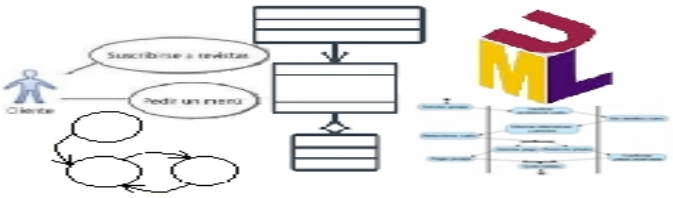
El control lógico programable de máquinas y procesos abarca un sinnúmero de aplicaciones que cubren la mayor parte de las necesidades de automatización en el sector industrial, tanto a nivel de las grandes y las más pequeñas entidades.



La escritura de programas de control lógico se desarrolló inicialmente en torno de lenguajes de contacto (LD) y en listas de instrucciones (IL). Con el desarrollo de los recursos informáticos y el incremento en el uso del control lógico, este se hizo más rico en recursos pudiendo resolver problemas más complejos. De esta forma, se hizo común el uso de bloques funcionales (FBD), diagramas de funciones secuenciales (SFC) y de texto estructurado (ST) para conformar la norma IEEE61131-3.



La existencia de los lenguajes de programación de control lógico aproxima al integrador de soluciones a métodos de diseño de aplicaciones cercanos a formas intuitivas de solución de problemas de automatización. Esto sin que deje de ser fundamental el conocimiento en principios de electricidad, de instrumentación industrial, de manejo de herramientas computacionales y de los procesos mismos.



El desarrollo de nuevas herramientas de modelado (UML) y de procesadores para el control lógico ha permitido el uso de herramientas que requieren de una curva de aprendizaje muy corta para llegar a desarrollar aplicaciones de control lógico. Este es el caso del uso del uso de máquinas de estado, las cuales son derivadas de las redes de Petri y permiten modelar aplicaciones de control lógico siguiendo unas pocas reglas simples.

Los elementos fundamentales de las máquinas de estado para el control lógico son:

Entradas: Son las variables que llevan hacia el controlador, información del estado del proceso que está siendo controlado. (Pulsadores, interruptores, sensores.)

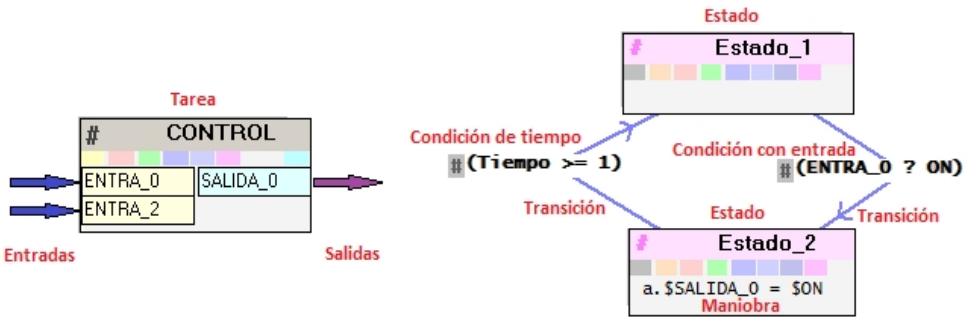
Salidas: Son las variables que llevan acciones desde el controlador hacia los actuadores. (Motores, relevos, contactores, válvulas, lámparas.)

Estado: Es un intervalo de tiempo en el cual las salidas del controlador lógico permanecen en un régimen estable. (Activadas, desactivadas, destellando, recibiendo asignando valores.)

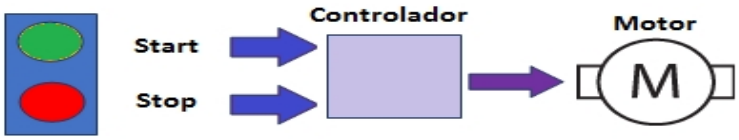
Transiciones: Cambios de estado suscitados por un eventos en las entradas o por el vencimiento de un tiempo.

Maniobras: Son las acciones que se realizan sobre las salidas dentro de los estados.

Condiciones: Son las expresiones que determinan el momento en el cual ocurre una transición. Están conformadas principalmente por evaluaciones de las entradas y el vencimiento de lapsos de tiempo.



Para ilustrar la utilización de las máquinas de estado se presenta el ejemplo típico de arranque y paro de un motor eléctrico: *En una maquina con control lógico, al pulsar el botón **Start** se pone en marcha el **Motor** y al pulsar **Stop** se detiene el motor*.

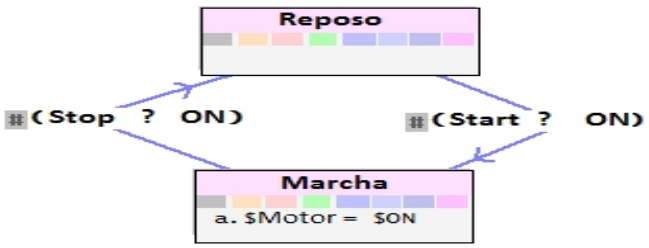


Para modelar el problema con máquinas de estado se establece un diagrama de entradas y salidas.



Start y Stop son entradas, puesto que **llevan información** sobre los deseos del operador de la maquina hacia el controlador (una pista está en que son pulsadores). **Motor es una salida**, puesto que **lleva acciones** hacia la maquina desde el controlador, haciendo energizar el motor de la máquina (es un actuador).

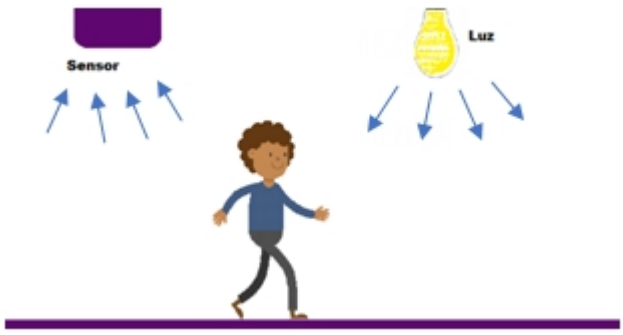
En el diagrama de entradas y salidas, la tarea corresponde a una caja negra, que pasa a ser una caja blanca al construir la secuencia, que relaciona las salidas con las entradas y el vencimiento de intervalos de tiempo.



En la primera transición, al activarse la entrada Start se pasa del estado de Reposo al estado de Marcha, en el cual estará activado el Motor. En la segunda transición, al activarse la entrada Stop se pasa del estado de Marcha al estado de Reposo, en el cual estará desactivado el Motor.

El diagrama de estados resultante muestra que el motor puede estar en dos estados (**Reposo y Marcha**). Es recomendable poner maniobras sólo en el estado en el cual el motor está activado, para disminuir la cantidad de maniobras en los estados y hacer más limpio el diagrama. Cada transición muestra el sentido de paso de un estado a otro.

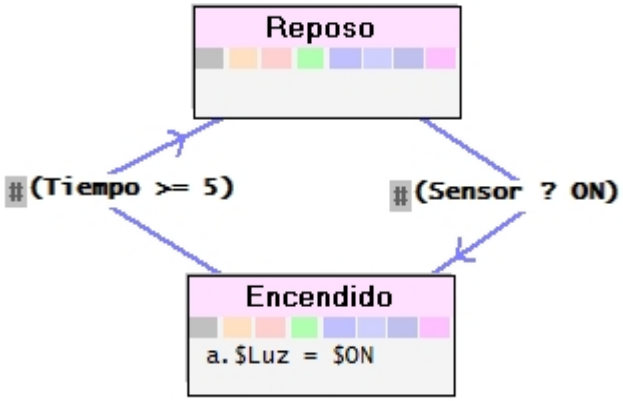
Otro ejemplo ilustrativo es el de una luz de pasillo, la cual se enciende al activarse un sensor que detecta el paso de las personas y se apaga luego de cinco (5) segundos.



El diagrama de entradas y salidas tiene la entrada **Sensor** que **lleva información** al controlador, sobre la presencia de alguna persona en el pasillo y la salida **Luz**, la cual **lleva la acción** para hacer encender una lámpara.



La secuencia tendrá una transición que depende de la entrada Sensor, representada como “Sensor ? ON”, y otra que depende del vencimiento del lapso de tiempo de cinco (5) segundos, representados como “Tiempo ≥ 5”



Los modelos de estados pueden ser traducidos a lenguaje de escalera (por el método *Set-Reset-Quit*), a texto estructurado utilizando la instrucción *Case Selector Of*, a lenguaje C con la instrucción *Switch Case*, a compuertas digitales con *mapas de Karnaugh* y *tablas de excitación* de los flip-flops, a diagramas de relevos aplicando las reglas de *DeMorgan* a los diagramas de escalera, o programado directamente en la herramienta de desarrollo MgdMod (www.tecvolucion.net/descargas).

Recomendaciones para la implementación de programas de control lógico.

Para implementar programas de control lógico utilizando máquinas de estado, se debe tener en cuenta una serie de recomendaciones, tanto de carácter tecnológico como a nivel de la construcción de los programas.

Al implementar programas de control lógico hay variables de entrada y salida, las cuales deben ser diferenciadas con claridad.

Recomendación uno. Seleccionar las *salidas* como aquellas variables que pueden estar activadas o desactivadas y *actúan* directamente *sobre el proceso*.

Motores.	Actuadores electromecánicos
Contactores.	Manejadores de actuadores
Variadores de velocidad	Controlan motores
Relevos electromecánicos y de estado sólido	Manejadores de actuadores
Válvulas neumáticas e hidráulicas	Manejadores de cilindros
Cilindros neumáticos e hidráulicos	Actuadores neumáticos e hidráulicos
Lámparas (Luces, pilotos)	Actuadores acústicos
Entradas de otros controladores	Manejadores electrónicos

Recomendación dos. Seleccionar las *entradas* como aquellas variables que *llevan información* al controlador sobre el estado del proceso.

Pulsadores normalmente abiertos y cerrados	Interfaz humano - máquina
Interruptores y selectores (N.O y N.C)	Interfaz humano - máquina
Finales de carrera (microrruptores)	Sensores ON-OFF electromecánicos
Sensores inductivos, capacitivos, ópticos	Sensores ON-OFF electrónicos
Sensores de presencia	Sensores ON-OFF electrónicos
Termostato, presostato, nivostato	Sensores ON-OFF de variables físicas
Salidas de otros controladores	Generadores electrónicos de señales

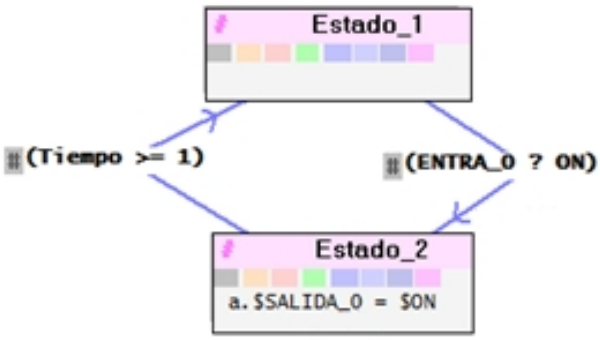
Los lapsos de tiempo no se consideran entradas, puesto que los conteos de tiempo se llevan a cabo en cada estado de las tareas al interior del controlador.

Recomendación tres. Crear *tareas* en las cuales las salidas operan en forma coordinada, unas con respecto a las otras, posiblemente bajo el control de las entradas y los lapsos de tiempo. Esto se refiere a que si una salida hace secuencias completas de activación y desactivación, independientemente a como lo hacen las demás salidas de la tarea, se debe crear una tarea a parte para la secuencia de dicha salida.



Al crear *tareas* se establecen diagramas de entradas y salidas que constituyen una primera aproximación a un *bloque funcional* (FBD) de la norma IEEE61131-3.

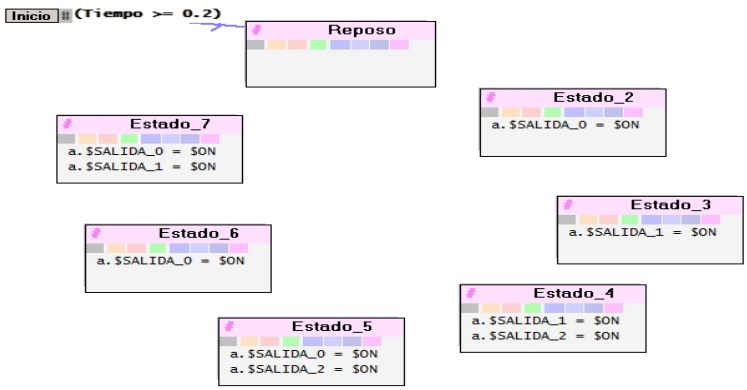
Recomendación cuatro. Crear la *secuencia* que resuelve la relación entre las salidas, las entradas y el vencimiento de lapsos de tiempo en la tarea. Las *secuencias* son próximas a los *diagramas grafcet* (SFC) de la norma IEEE61131-3.



Las salidas se activan a través de las expresiones en las *maniobras* de los *estados* y constituyen expresiones similares al texto estructurado (ST) de la norma IEEE61131-3. Las entradas se consultan en las *condiciones* de las *transiciones* que provocan el cambio de un estado a otro. Los intervalos de "Tiempo >= 1" se consultan también en las condiciones de las transiciones.

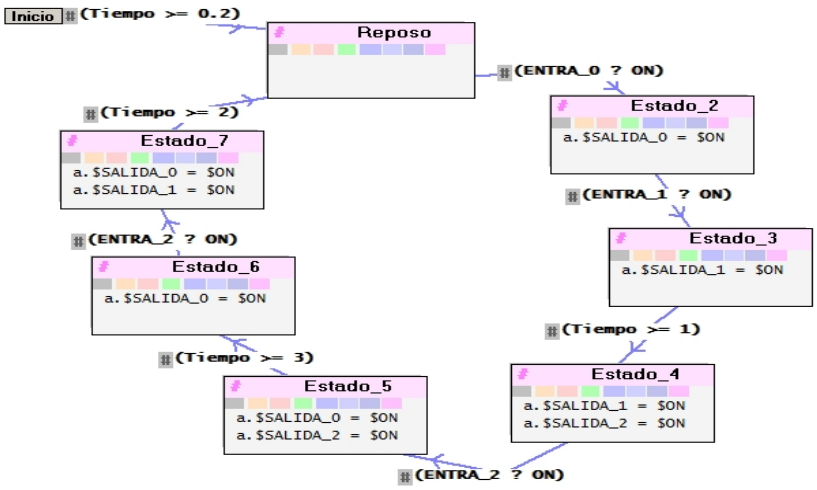
En cada estado se lleva un control de tiempo, en forma inherente. Por lo cual basta con solo comparar el tiempo con el valor del lapso que debe durar el estado.

Recomendación cinco. Para establecer la *cantidad de estados* que tiene una secuencia, se debe determinar la cantidad de combinaciones prácticas que tienen las salidas, teniendo en cuenta que dos combinaciones iguales pueden formar diferentes estados, si la condición de transición para cambiar de estado es diferente de un caso al otro. Una estrategia fundamental está en definir las etapas de ejecución que debe seguir la operación del sistema a controlar.



Recomendación seis. Hay dos *alternativas para crear la secuencia*, de las cuales la primera consiste en ubicar inicialmente todos los estados planeados y agregarle a cada uno las maniobras, activando las salidas correspondientes. Luego se agregan las transiciones que salen de cada estado, editando la condición.

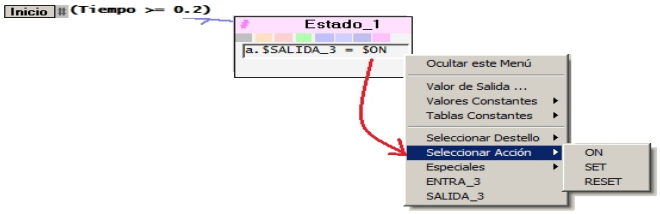
Otra alternativa, consiste en construir la secuencia agregando el estado siguiente en el orden de operación del proceso y la transición que le une con el estado predecesor. Para cada estado creado se pueden agregar las maniobras en cualquier momento, al igual que las transiciones.



Recomendación siete. El control lógico básico utiliza *salidas del tipo activado-desactivado* (ON-OFF). Al editar las maniobras se recomienda asignar a las salidas la acción "ON", la cual activa la salida física solo en los estados en los cuales aparezca.

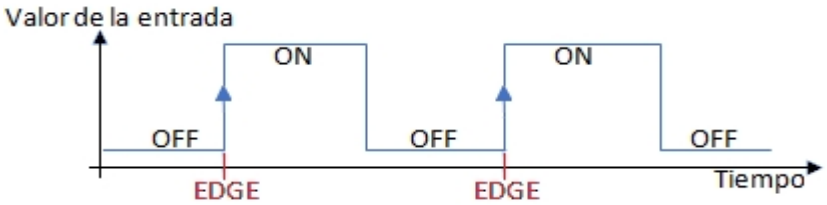
En los demás estados, la salida estará desactivada, esto brinda una operación segura y hace que la secuencia esté menos saturada de maniobras.

En caso de ser necesario conservar una salida encendida sin tener que activarla en cada estado, se utiliza la acción "SET", de este modo, la salida estará activa hasta que se llegue a un estado con una maniobra que asigne a la salida, la acción "RESET".



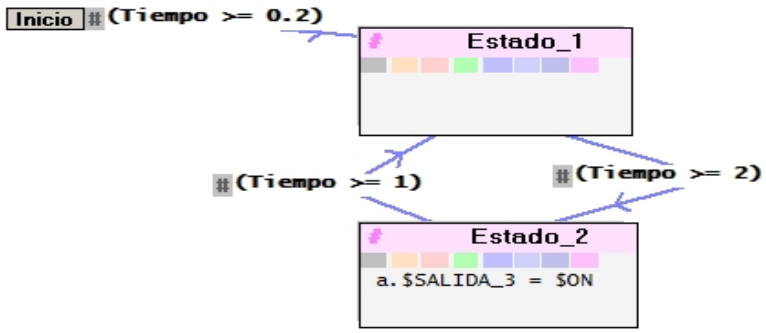
Es importante tener en cuenta que las acciones sobre las salidas físicas de los controladores lógicos operan en el momento en el cual su sistema operativo (RTOS) invoca la rutina de salidas (Al principio del siguiente ciclo de ejecución). Por lo anterior, dos maniobras dentro del mismo estado con acciones sobre la salida "SET" y "RESET", ***no*** *producen* un pulso en la salida física del controlador.

Recomendación ocho. Las *entradas* para el control lógico tienen varias *formas de consulta*. La más básica es "ON", que consiste en preguntar si la entrada está activada. La consulta con "OFF" pregunta si la estrada está desactivada. De forma similar, la consulta con "EDGE" revisa en cual momento ocurre un cambio de desactivado a activado en la entrada (el instante preciso).



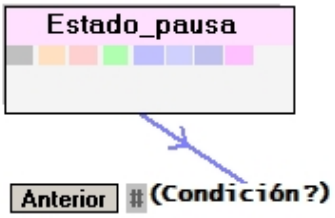
Otra forma de consultar una entrada es interpretándola como una variable continua con valores posibles entre 0 y 65535, para lo cual se le hace comparaciones de magnitud con "==, >, >=,<, <=" que utilizan la representación ANSI del lenguaje C.

Recomendación nueve. En cada estado corre un *conteo de tiempo*, el cual permite hacer transiciones por vencimiento de tiempo, utilizando la expresión “Tiempo >= n”, donde “n” es el tiempo por comparar en segundos, se compara con “>=” para hacer más segura la comparación. Aunque el reinicio del conteo se hace con cada cambio de estado, existe una función de reinicio del tiempo del estado, la cual permite reiniciar el conteo con un valor específico, tras revisar que el valor de una variable de comparación esté en un rango determinado.



La secuencia de la imagen activa la salida “SALIDA_3” durante un segundo y luego la desactiva durante dos segundos. Ambos estados utilizan el mismo contador de tiempo, el cual se reinicia durante el cambio de un estado al otro (esto permite utilizar la misma variable para llevar adelante el conteo de tiempo de todos los estados de una tarea).

Recomendación diez. Existe un recurso de *regreso al estado anterior*, el cual es uno de aquellos que potencian de forma significativa la labor de modelar una secuencia de control lógico con máquinas de estado. El recurso de regreso al estado anterior permite crear un estado de pausa o servicio, el cual se invoca desde varios estados de la secuencia, sin tener que crear un estado de pausa para cada uno de ellos. En este caso, la transición que sale del estado de pausa se crea apuntando hacia el fondo de la tarea en vez de hacia un estado en particular (hacia la etiqueta “Anterior”).



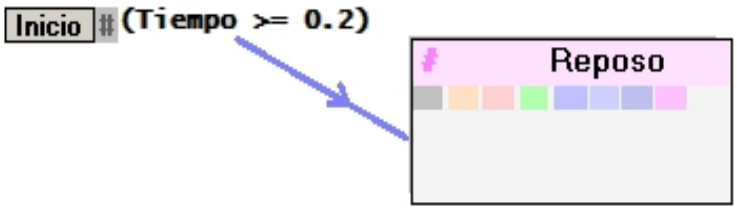
Al cumplirse la “Condición” de salida del estado, la secuencia continúa en el estado desde el cual se había llegado al “Estado_pausa”.

Recomendación once. El recurso de *cambio de estado desde cualquier estado* se utiliza cuando es necesario que la secuencia pueda ser llevada a un estado particular desde cualquier otro estado de la secuencia. En este caso no es necesario poner tantas transiciones como estados, para ello, basta con crear una transición desde el fondo de la tarea hacia el estado en cuestión (la etiqueta "Todos" significa ningún estado en particular). Esto se logra agregando a la Tarea un origen ("Agregar Origen").



Al cumplirse la "Condición", habrá un cambio de estado hacia "Estado". Si la condición en cuestión persiste, la secuencia se quedará todo el tiempo en "Estado".

Recomendación doce. En cada secuencia existe un estado de *inicio de secuencia* (generalmente "Reposo"), este tiene por defecto una transición desde el punto de "Inicio" (Reset del controlador lógico) con un intervalo de tiempo editable, durante el cual, el controlador no consulta las entradas, permitiendo que se estabilicen antes de iniciar la secuencia, lo cual evita que haya transiciones falsas al arrancar el sistema controlado.



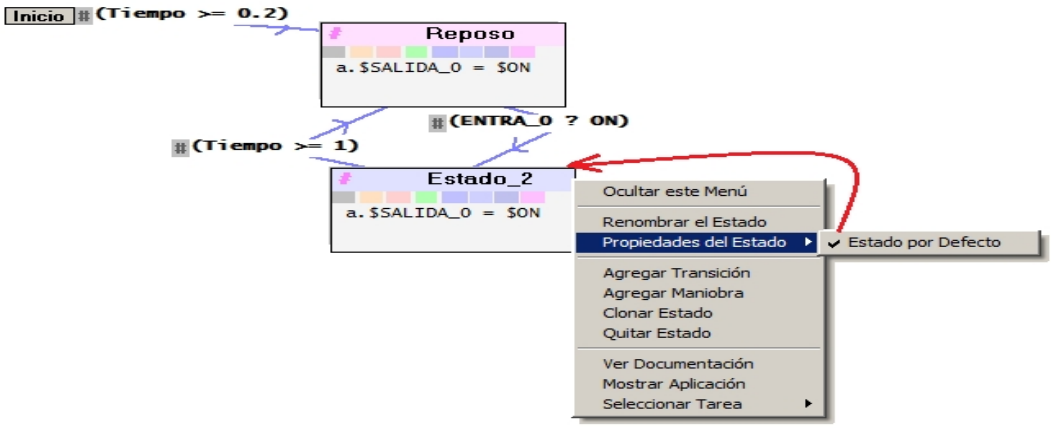
Tan pronto como se energiza el controlador lógico, el sistema operativo (RTOS) del autómata programable, reinicia las variables internas del programa y hace lo mismo con todas las secuencias de la aplicación, de las cuales, cada una estará inactiva durante un tiempo establecido durante la creación del programa, antes de pasar al estado de inicio del programa (generalmente "Reposo").

Consideraciones tecnológicas al automatizar con máquinas de estado.

La creación de soluciones de control lógico con máquinas de estado disminuye la cantidad de elementos con los cuales hay que interactuar, permitiendo que el integrador de soluciones se concentre en los aspectos propios del problema tecnológico a resolver. Los aspectos tecnológicos no se evidencian al crear y simular los programas de control lógico, en cambio son claves en el momento de llevar la aplicación de control a la máquina real, haciendo que las soluciones sean más confiables, seguras y eficientes energéticamente.

Entre estos aspectos cabe tener en cuenta una serie de casos que son resueltos mediante combinaciones específicas de estados, configuraciones de propiedades de los estados, variables especiales de los autómatas programables y mediante la utilización de funciones especiales o de estrategias de modularidad.

Uno: Estado por defecto. Existe la posibilidad de configurar un estado por defecto, al cual llegará la secuencia en caso de fallo del sistema. Aunque es una situación que probablemente nunca ocurra, el mecanismo de estado por defecto, garantiza la mayor confiabilidad del control lógico.

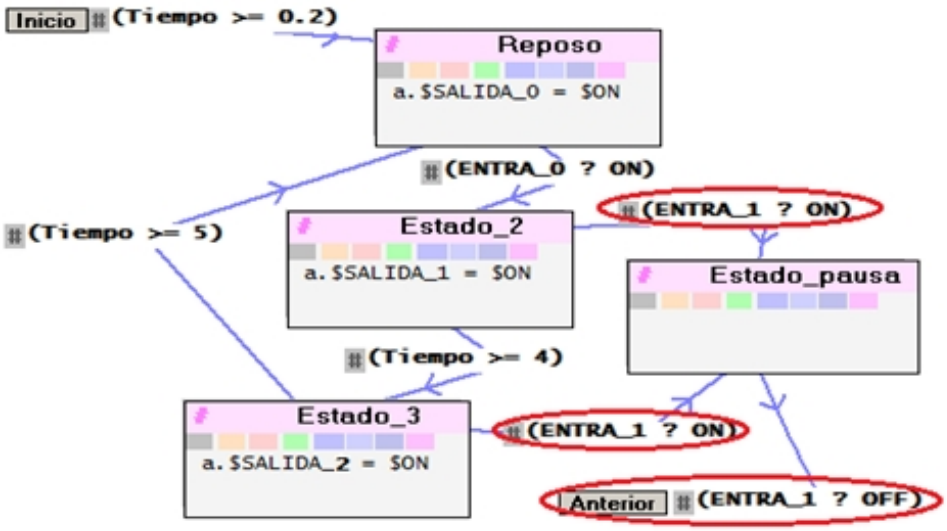


La calidad de estado por defecto se configura como propiedad de uno de los estados, pulsando click derecho sobre el nombre del estado, tras lo cual aparecerá un menú desplegable, en el cual se selecciona la opción “Propiedades del estado” y se activa o desactiva la opción “Estado por defecto”. El estado por defecto tiene el título con un color de fondo azul claro.

En caso de no configurarse ningún estado por defecto, la reposición de la secuencia (en caso de fallos) se hace hacia el “INICIO” de la secuencia.

Dos: Estado de pausa. Existe un concepto de retorno al estado anterior, el cual permite crear estados de pausa comunes a varias etapas de una secuencia. Los estados de pausa tienen una transición de salida hacia la etiqueta “ANTERIOR” con una condición específica. El estado de pausa es una especie de subrutina en la secuencia.

La ventaja de esta estrategia radica en que no es necesario tener un estado de pausa por cada etapa de la secuencia, simplificando la creación de las aplicaciones de control.

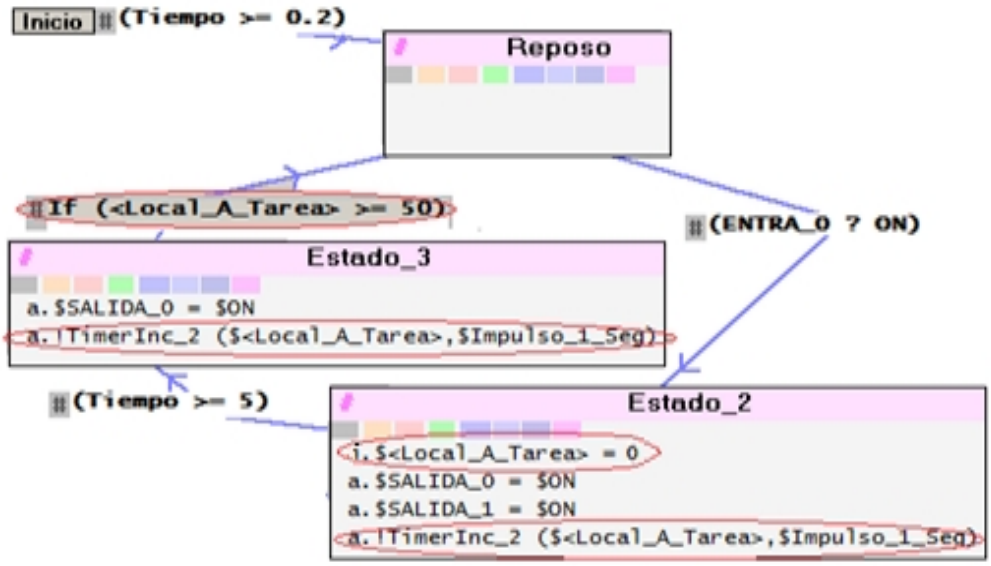


En la imagen se observa una secuencia de tres estados principales, que inicia activando la salida “SALIDA_0” (luego de 0.2 segundos de energizado el autómata programable) y al activarse la entrada “ENTRA_0”, se pasa al estado “Estado_2”, en el cual se activa la salida “SALIDA_1” (desactivándose la salida “SALIDA_0”). Luego de 4 segundos, se pasa al estado “Estado_3”, en el cual se activa la salida “SALIDA_2” (se desactiva la salida “SALIDA_1”) y luego de 5 segundos se regresa al estado “REPOSO” (se activa “SALIDA_0” y se desactiva “SALIDA_2”).

Cuando la secuencia está en el estado “Estado_2” o “Estado_3”, al activarse la entrada “ENTRA_1”, la secuencia pasa al estado “Estado_pausa”, en el cual todas las salidas están desactivadas. Al desactivarse la entrada “ENTRA_1”, la secuencia regresa al estado en el cual estaba antes de activarse la entrada y se activa la salida respectiva.

Al utilizar esta estrategia con estados que llevan conteos de tiempo, se debe utilizar un contador adicional de tiempo (y la función de conteo de tiempo), puesto que cada que hay cambios de estado, se reinicia el conteo de tiempo del estado.

Tres: Conteo de tiempo adicional. Cuando se requiere llevar un conteo de tiempo independiente del conteo inherente de los estados, se utiliza una variable local y la función de contar tiempo, lo cual permite hacer conteos del tiempo acumulado al cabo de la sucesión de varios estados, seleccionar la base de tiempo del conteo (1/50 de segundo, décimas de segundo, segundos o décimas de minuto) y hacer comparaciones de tiempo con variables ajustadas por el operario del sistema controlado.



En este caso la variable “<Local_A_Tarea>” estará contando segundos en los estados “Estado_2” y “Estado_3” para luego ser comparada con el valor 50 (segundos) para pasar al estado “Reposo”. El conteo de la variable “<Local_A_Tarea>” empieza al iniciar el estado “Estado_2”, en el cual están activadas las salidas “SALIDA_0” y “SALIDA_1”, para luego de cinco segundos pasar al estado “Estado_3”, en el cual está activada la salida “SALIDA_0”.

La variable “<Local_A_Tarea>” es una variable interna de la tarea, accesible mediante la opción “Especiales” del menú desplegable de alguna de las maniobras.

Existe otra alternativa para llevar el conteo, ubicando la función de incremento de tiempo “TimerInc_2” en un bloque de maniobras anteriores y no en los estados, lo cual simplifica el programa.

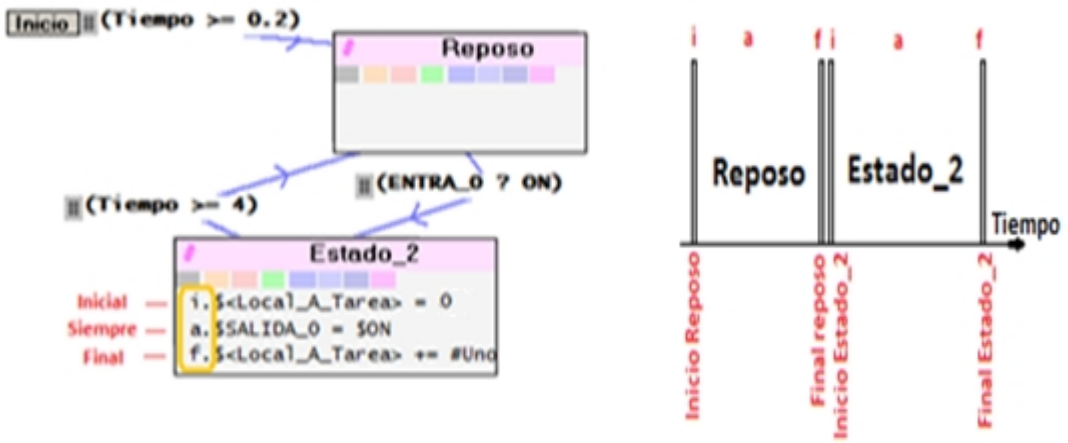
Para que el conteo en la variable “<Local_A_Tarea>” se lleve a cero al inicio del estado “Estado_2”, es necesario cambiar el contexto de la maniobra de la opción “a.” a la opción “i.”, pulsando click sobre la letra “a” ubicada al lado izquierdo de la maniobra.

Cuatro: Contexto de los estados. Las maniobras en los estados se pueden ejecutar en tres contextos diferentes. Un contexto se refiere a un momento específico dentro del estado, relativo a los instantes en los cuales ocurren las transiciones.

“ALLWAYS” = “a.”. Es el más usual (contexto por defecto), en el cual, la maniobra se ejecuta permanentemente mientras la secuencia esté en el presente estado. Las maniobras con contexto “a.” se ejecutan la cantidad de veces por segundo, que corresponde a la frecuencia de la tarea (50 veces por segundo para 50 Hz).

“INITIALIZING” = “i.”. En este contexto, la maniobra se ejecuta solamente una vez, cada que hay una transición hacia el estado de la maniobra en cuestión (el instante en el cual se inicia el estado, que es 1/50 de segundo para la tarea de 50 Hz).

“FINISHING” = “f.”. En este contexto, la maniobra se ejecuta solamente una vez, cada que hay una transición desde el estado de la maniobra en cuestión (el instante en el cual finaliza el estado, que es 1/50 de segundo para la tarea de 50 Hz).



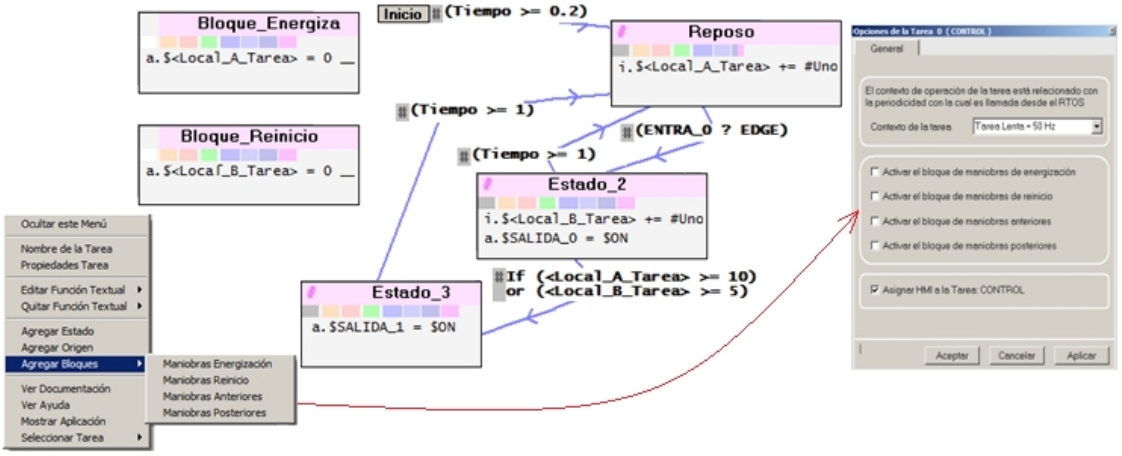
En la secuencia de la imagen, la variable “<Local_A_Tarea>” recibe el valor cero en el instante en el cual ocurre la maniobra de cambio del estado “Reposo” al estado “Estado_2”. Mientras este último estado está activo, estará activada la salida “SALIDA_0” y al ocurrir la transición hacia el estado “Reposo”, se incrementa “<Local_A_Tarea>” en una unidad.

La utilidad de la modificación del contexto de una maniobra consiste en la disminución de la cantidad de estados de una secuencia, al no tener que crear estados intermedios para realizar maniobras que operen en el preciso instante de las transiciones (por ejemplo para inicializar contadores).

Cinco: Bloques de maniobras de inicio. Cuando se requiere que algunas maniobras se ejecuten solamente en el momento en el cual se energiza el autómata programable se utilizan los bloques de maniobras de inicio, los cuales pueden ser de dos tipos:

Bloque de energizado (“Bloque_Energiza”). Es un bloque de maniobras que se ejecutan en el momento de aplicación de energía al autómata programable. En el momento de energizado del autómata programable, se inicializan todas las variables y se reconfigura el hardware.

Bloque de reinicio (“Bloque_Reinicio”). Es un bloque de maniobras que se ejecutan en el momento de “Reset” del procesador del autómata programable (energizado y reposición desde estados fallidos). En el momento de reinicio del autómata programable, se reconfigura el hardware.



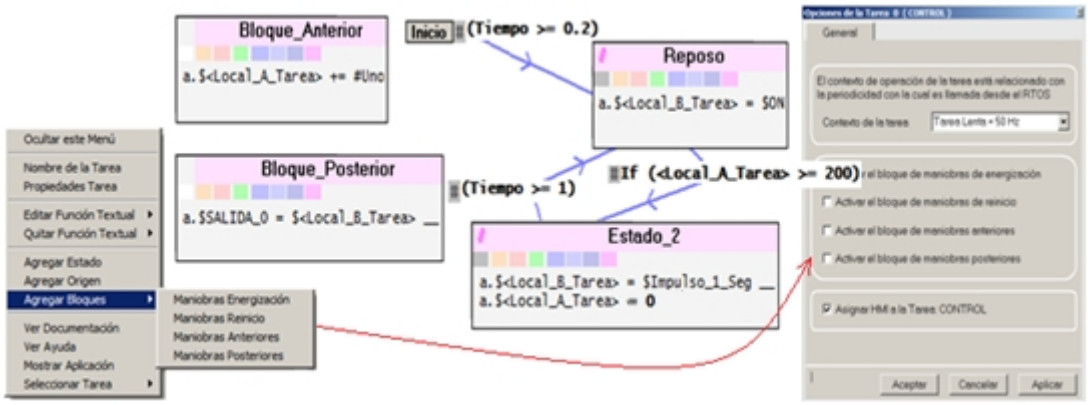
En la imagen se observa como la variable “<Local_A_Tarea>” se reinicia en cero al energizar el autómata programable y la variable “<Local_B_Tarea>” se reinicia en cero al reiniciar el autómata. Cada que la secuencia pase al estado “Reposo”, se incrementa “<Local_A_Tarea>” y cada que la secuencia pase al estado “Estado_2”, se incrementa “<Local_B_Tarea>”. Cuando hayan ocurrido 10 ciclos sin re-energizar el autómata programable o 5 sin reiniciarlo, se pasa al estado “Estado_3”, en el cual se activa la salida “SALIDA_1”

Para hacer visibles los bloque de maniobras se pulsa click derecho sobre el fondo de la tarea y se pulsa click sobre una de las opciones del submenú “Agregar bloques”, tras lo cual aparecerá el bloque de maniobras respectivo, en la parte superior izquierda de la ventana de la tarea. También se puede lograr lo mismo desde la ventana de “Propiedades Tarea”, al pulsar click derecho sobre el fondo de la tarea.

Seis: Bloques de maniobras anteriores y posteriores. Cuando se requiere que algunas maniobras se ejecuten permanentemente, independiente del estado en que esté la secuencia se utilizan los bloques de maniobras anteriores y posteriores. Las maniobras de estos bloques se ejecutan tantas veces por segundo como la frecuencia de la tarea (50 veces por segundo para la tarea de 50 Hz).

Bloque de maniobras anteriores ("Bloque_Anterior"). Es un bloque de maniobras que se ejecutan permanentemente, justo antes de la ejecución del estado activo de la secuencia.

Bloque de maniobras posteriores ("Bloque_Posterior"). Es un bloque de maniobras que se ejecutan permanentemente, justo después de la ejecución del estado activo de la secuencia.



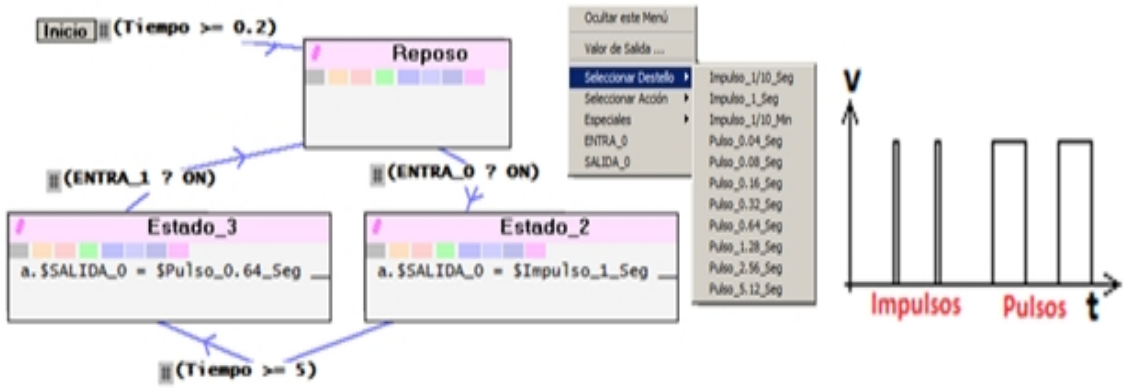
En la imagen se observa como la variable "<Local_A_Tarea>" se incrementa cada que el "RTOS" del autómata programable llama la tarea (50 veces cada segundo). La salida "SALIDA_0" se carga cada vez con el valor de la variable "<Local_B_Tarea>", la cual está activada en el estado "Reposo" y destellando cada segundo en el estado "Estado_2". Tan pronto como la variable "<Local_A_Tarea>" es mayor o igual a 200 se pasa de "Reposo" a "Estado_2" y luego de un segundo en sentido inverso, reiniciando el conteo en este último estado.

Para hacer visibles los bloques de maniobras se pulsa click derecho sobre el fondo de la tarea y se pulsa click sobre una de las opciones del submenú "Agregar bloques". También se puede lograr lo mismo desde la ventana de "Propiedades Tarea", al pulsar click derecho sobre el fondo de la tarea. En esta ventana se hace check sobre la casilla respectiva del bloque de tareas para hacerlo aparecer u ocultarlo.

Para ocultar un bloque de maniobras se debe borrar todas sus maniobras y pulsar click derecho sobre el título del bloque para seleccionar la opción "Quitar Bloque".

Siete: Salidas destellantes. Cuando se tienen casos en los cuales una salida debe destellar para informar una situación (por ejemplo balizas), se pueden montar secuencias de encendido y apagado de una salida en particular. Cuando la salida debe destellar en diferentes estados (en algunos casos a diferentes frecuencias), no es recomendable crear sub-secuencias, puesto que la secuencia se hará compleja y su edición será dificultosa.

En estos casos es conveniente utilizar las variables pulsantes del sistema operativo "RTOS" del autómata programable. Las variables pulsantes son de dos tipos, los *impulso*s (bases de tiempo con ancho de pulso de 1/50 de segundo) y los *pulsos* (con ancho de pulso del 50%) de diferentes períodos.



La secuencia inicia en el estado "Reposo", 0.2 segundos luego de haber energizado el autómata programable y al activares la entrada "ENTRA_0" pasa al estado "Estado_2", en el cual la salida "SALIDA_2" emite destellos de 1/50 de segundo, cada segundo. Luego de 5 segundos, la salida empieza a emitir destellos de 0.64 de segundo (0.32 encendido y 0.32 apagado). Al activarse la entrada "ENTRA_1" se pasa al estado "Reposo", en el cual la salida queda desactivada.

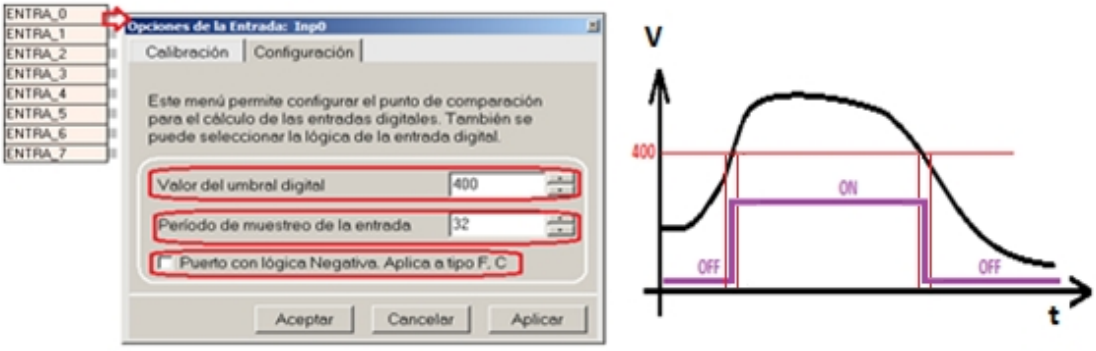
Para seleccionar un destello se pulsa click sobre el argumento (a la derecha de "=") y en el menú desplegado se selecciona una de las opciones del grupo "Seleccionar Destello".

Los impulsos (bases de tiempo) disponibles son 1/10 segundo, 1 segundo y 1/10 de minuto. Se pueden crear impulsos adicionales utilizando máquinas de estado en tareas dedicadas para tal fin.

Los pulsos disponibles son 0.04 segundos (0.02 On y 0.02 Off), 0.08 segundos (0.04 On y 0.04 Off), 0.16 segundos (0.08 On y 0.08 Off) y así sucesivamente hasta 5.12 segundos (2.56 On y 2.56 Off). Cada pulso tiene el doble del período del anterior.

Ocho: Filtrado de las entradas. Existe la posibilidad de ajustar el umbral que determina cuando una entrada ON-OFF corresponde a activado (mayor al umbral) o desactivado (menor al umbral) y de ajustar la cantidad de veces que se promedia la variable antes de calcular si está activada o desactivada. En la misma ventana se puede determinar si la lógica de la variable se invierte al ser leída (acción recomendable al utilizar entradas desde contactos normalmente cerrados).

Se accede a la ventana de ajuste pulsando click derecho sobre la entrada a configurar, para seleccionar la opción “Propiedades Entra” y luego seleccionando la pestaña “Configuración”.



Al ajustar el valor “Valor del umbral digital” entre 0 y 1023 (4095 para algunos tipos de autómatas) se establece a partir de cual valor de la entrada, se considera que está activada. Al ajustar el valor “Período de muestreo de la entrada” entre 1 y 32, se establece la cantidad de lecturas (cada 1/6250 de segundo) para sacar el promedio del valor de la entrada. Al hacer chequeo de “Puerto con lógica Negativa. Aplica …”, se establece si la lógica de la entrada se invierte al calcular su estado On u Off.

El uso de filtros aplica para eliminar componentes de alta frecuencia en las entradas (rizados) y en los casos en los cuales las entradas presentan alguna diferencia de potencial cuando están desactivadas o un valor de voltaje muy bajo al estar activadas.

Al realizar programas de control lógico es recomendable trabajar con “lógica positiva”, lo cual implica que cuando la variable esté desactivada sea igual a cero y cuando esté activada tenga un valor diferente a cero, lo cual se logra ajustando la lógica de la entrada. En tal caso las entradas normalmente cerradas se consultan con la como si fueran normalmente abiertas.

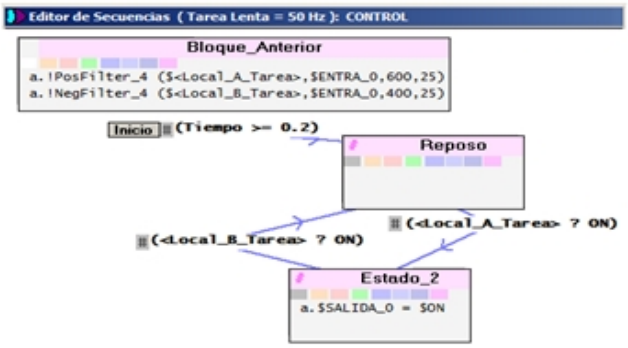
El uso de filtros no implica que las instalaciones se deban hacer defectuosas (cableado mal seleccionado y/o mal distribuido o casos de mal contacto).

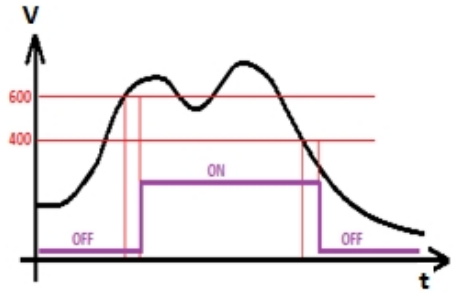
Nueve: Filtrado de entradas muy ruidosas. Existen casos en los cuales una entrada digital (ON-OFF) presenta variaciones (ruidos por inducción electro-magnética, por cableado defectuoso o malos contactos), las cuales alcanzan valores que se pueden confundir con estados de activado y desactivado durante períodos que pueden estar en el rango de décimas de segundo hasta algunos segundos.

En estos casos es recomendable utilizar funciones de filtrado de entradas digitales a partir de variables continuas (considerando que todas las entradas del autómata programable son analógicas en el rango [0…1023]).

La función “PosFilter_4” activa una variable digital (primer argumento), cuando detecta que la entrada (segundo argumento) es mayor que un valor (tercer argumento) durante una cantidad determinada de ciclos del programa (Cuarto argumento).

La función “NegFilter_4” activa una variable digital (primer argumento), cuando detecta que la entrada (segundo argumento) es menor que un valor (tercer argumento) durante una cantidad determinada de ciclos del programa (Cuarto argumento).





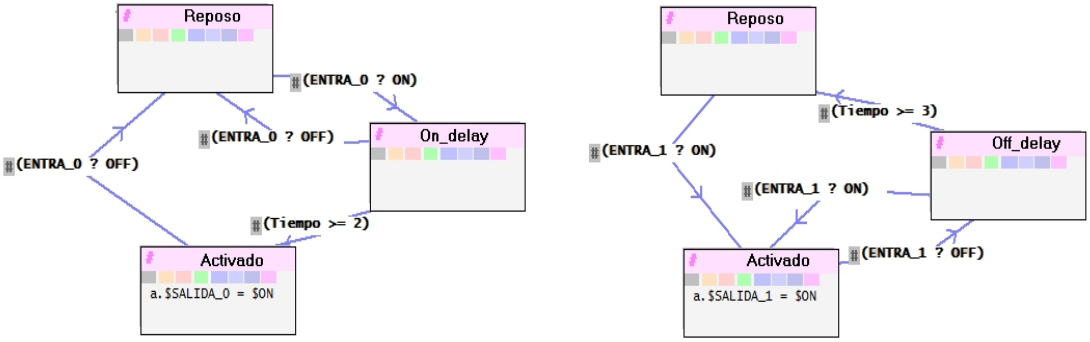
En un bloque de maniobras anteriores se combinan dos funciones de filtrado, una de las cuales activa la variable “<Local_A_Tarea>” cuando la entrada “ENTRA_0” es mayor que 600 durante 25/50 de segundo (medio segundo) y la desactiva en caso contrario. La otra función activa la variable “<Local_B_Tarea>” cuando la entrada “ENTRA_0” es menor que 400 durante medio segundo y la desactiva en caso contrario. Cuando se activa la variable “<Local_A_Tarea>”, la secuencia pasa al estado “Estado_2” y activa la salida “SALIDA_0”. Al activarse “<Local_B_Tarea>”, la secuencia regresa a “Reposo” y desactiva la salida “SALIDA_0”. Esto implica que al activarse la entrada ruidosa se activa la salida y al desactivarse la primera se desactiva la segunda.

El uso de filtros no implica que las instalaciones se deban hacer defectuosas (tendido del cableado de los sensores por las mismas tuberías del cableado de los motores).

Diez: Temporizadores On-Delay y Off-Delay. Los integradores de soluciones de control lógico que programan con lenguaje “LADDER” suelen utilizar temporizadores con *retardo al trabajo* y con *retardo al reposo*, a través de los cuales realizan las diferentes operaciones de conteo de tiempos de las aplicaciones.

Cuando se utilizan máquinas de estado como recurso de modelado de programas de control lógico, se hacen innecesarios los retardos al trabajo y al reposo. En lugar de estos se recurre a contadores de tiempo, los cuales se comparan para realizar transiciones de un estado a otro (al traducir a lenguaje LADDER se utilizarán solamente retardos al trabajo o simples contadores de tiempo).

Lo anterior implica que la lógica no se establece diferenciando sin un retardo se hace al trabajo o al reposo, sino para hacer un cambio de estado. Lo que define si el retardo es al trabajo o al reposo es la acción de activar o desactivar variables en los estados (inclusive considerado que hay estados en los cuales se activan varias variables y se desactivan otras).



La imagen muestra la implementación de un retardo al trabajo (“On delay”) utilizando una secuencia y un retardo al reposo (“Off delay”) utilizando otra. La implementación se hace con fines ilustrativos antes que con fines prácticos.

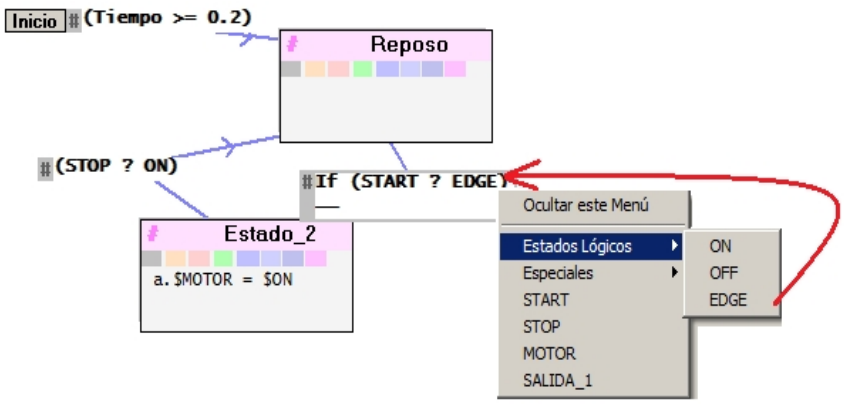
En la primera secuencia (de retardo al trabajo), al activarse la entrada “ENTRA_0” durante dos segundos continuos, se activará la salida “SALIDA_0”. Si en cualquier momento se desactiva la entrada, se desactivará la salida y si la entrada se desactiva antes de pasados los dos segundos, la salida nunca se activará.

En la segunda secuencia (de retardo al reposo), al activarse la entrada “ENTRA_1”, se activará la salida “SALIDA_1” inmediatamente. Al desactivar la entrada durante tres segundos se desactivará la salida. Si al desactivar la entrada, se vuelve a activar antes de los tres segundos, la salida nunca se desactivará.

Once: Activación-desactivación. Cuando se desarrolla una aplicación con acciones de activación y desactivación (Start-Stop), es recomendable que la consulta de la entrada que da la orden de activación "START" se haga con la opción "EDGE" y la consulta de la entrada que da la orden de desactivación "STOP" se haga con la opción "ON" si el pulsador es normalmente abierto "N.O.", o la opción "OFF" si es normalmente cerrado "N.C.".

Lo anterior con el fin de que no haya arranques falsos en caso de que se quede pegado el pulsador "START" y para que el pulsador "STOP" detenga el proceso siempre que esté activado.

En caso de tener un pulsador normalmente cerrado "N.C." y la configuración de la entrada tenga negada la lógica, se utiliza la entrada como si el pulsador de fuera normalmente abierto "N.O."



La secuencia se inicia en el estado "Reposo", luego de 0.2 segundos de energizado el autómata. Al detectarse un flanco ascendente en la entrada "START", se pasa al estado "Estado_2", en el cual se activa la salida "MOTOR" y al detectarse una activación de la entrada "STOP" se regresa al estado "Reposo" y se desactiva la salida.

Para seleccionar si la entrada se revisa por nivel o por flanco, se pulsa click sobre el argumento (luego de "?") de la expresión en la condición de la transición, para elegir una de las opciones (ON, OFF o EDGE) en el sub-menú "Estados Lógicos".

Las opciones del grupo "Estados Lógicos" son:

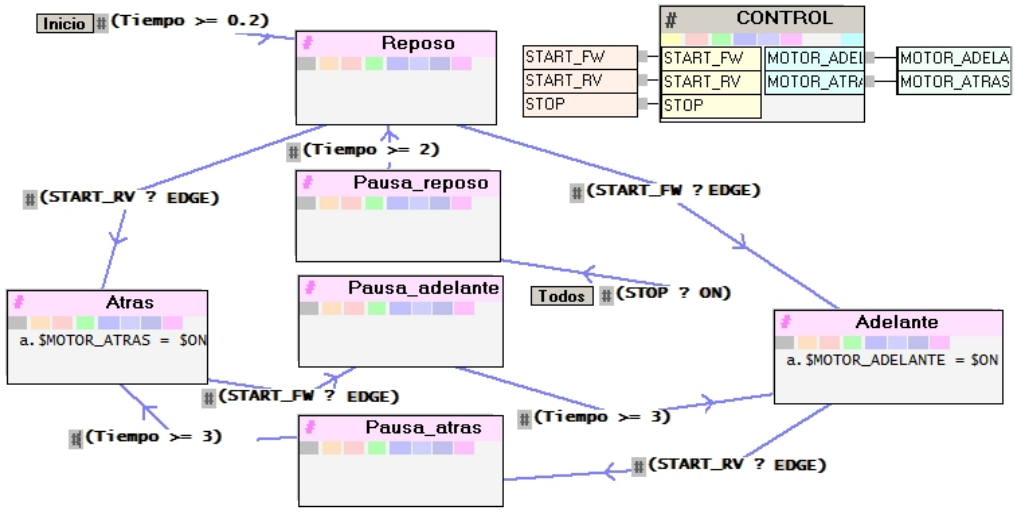
ON: Detecta que la variable está activada, o sea que la variable es diferente de cero.

OFF: Detecta que la variable está desactivada, o sea que la variable es igual a cero.

EDGE: Detecta que la variable ha cambiado de desactivada a activada.

Doce: Inversión de giro de motores eléctricos. Al realizar una inversión de giro en un motor eléctrico con determinada inercia mecánica y con peligro de destrucción del mecanismo al hacer cambios del sentido de giro en forma instantánea, es recomendable agregar estados de retardo para evitar que haya operaciones peligrosas para la máquina.

Las inversiones de giro en los motores eléctricos pueden además, generar picos de corriente (motor en contra corriente), suficientemente altos para disparar la protección del motor (El doble de la corriente de arranque). Cuando los autómatas programables comandan el motor a través de variadores de velocidad, no es necesario adicionar retardos, ya que los variadores pueden aplicar rampas de arranque y parada.



Un accionamiento típico de inversión de giro tiene un pulsador de arranque hacia adelante (“START_FW”), otro de arranque hacia atrás (“START_RV”) y otro de paro (“STOP”). También tiene una salida de energizado adelante (“MOTOR_ADELANTE”) y otra de energizado hacia atrás (“MOTOR_ATRAS”).

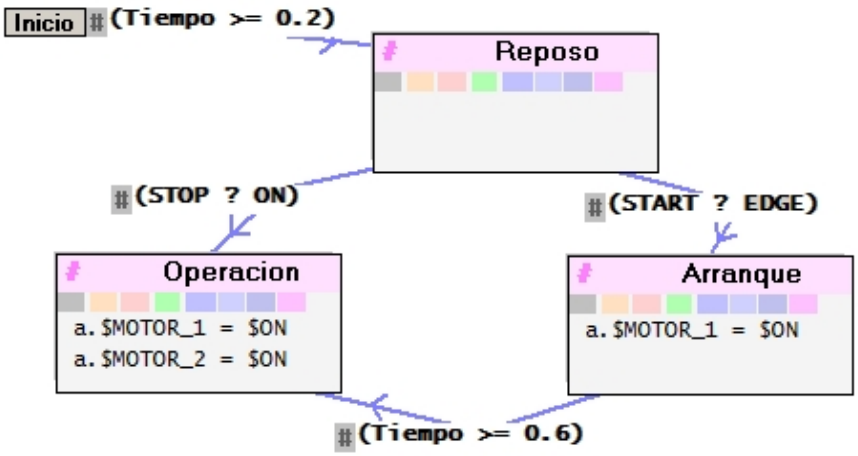
La secuencia arranca en el estado “Reposo” y al activarse la entrada “START_FW” se pasa al estado “Adelante”, en el cual se activa la salida “MOTOR_ADELANTE”. Al pulsar “START_RV” se espera un tiempo antes de pasar a “Atrás”, con lo cual el motor empieza a girar en sentido contrario. Cuando la secuencia está en el estado “Atrás”, también hay un retardo para pasar al estado “Adelante” luego de pulsar “START_FW”.

La transición desde el origen “TODOS”, al pulsar “STOP” lleva la secuencia al retardo de 2 segundos para pasar a “Reposo” desde cualquiera de los estados de la secuencia.

Para determinar el tiempo de los retardos de la secuencia se puede realizar una maniobra de arranque y paro del motor para hacer la respectiva medición de tiempo.

Trece: Arranque simultaneo de motores. En los casos en los cuales es necesario arrancar más de un motor a la vez y no se quiere generar picos altos de corriente en la red eléctrica, se debe recurrir a la adición de estados en la secuencia, con el fin de evitar la simultaneidad de los arranques.

El arranque simultaneo de motores eléctricos influye negativamente en la calidad de la energía eléctrica y dificulta la selección de los sistemas de protección (los totalizadores pueden no proteger un motor operando solo).



La secuencia arranca en el estado “Reposo”, luego de 0.2 segundos de haberse energizado el autómata programable. Al haber un flanco en la entrada “START” se pasa al estado “Arranque”, en el cual se activa el primer motor “MOTOR_1”. Luego de 0.6 segundos (tiempo en el cual debe haber bajado la corriente de arranque del primer motor), se pasa al estado “Operación”, en el cual estarán activados los dos motores. Al activarse la entrada “STOP” la secuencia pasa al estado “Reposo”, en el cual estarán desactivados los dos motores.

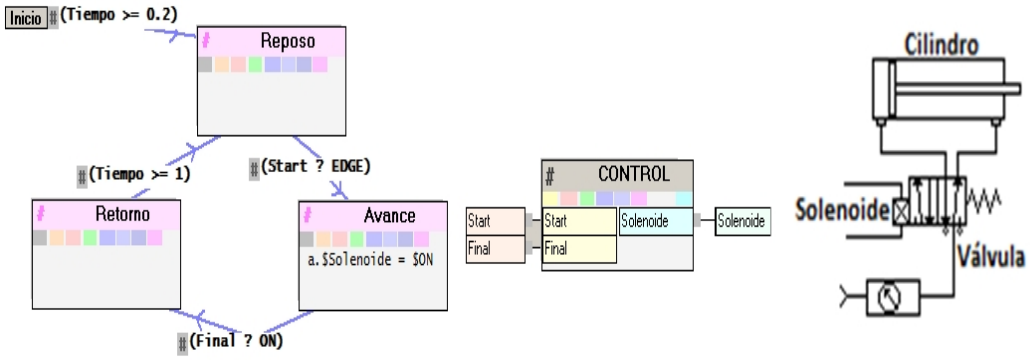
En casos en los cuales también hay picos de corriente significativos al des energizar los motores, se debe agregar un estado intermedio para desactivar una salida y pasado un corto período de tiempo, des energizar la otra.

El orden de arranque de los motores está relacionado con la secuencia productiva, así un motor de salida del proceso debe arrancar primero que un motor de entrada, para evitar acumulaciones de material en mitad del proceso.

El concepto también aplica a la activación y desactivación de válvulas hidráulicas, neumáticas o de gases, para evitar picos de corriente o caídas de voltaje que pueden inducir falsas órdenes hacia las entradas del autómata programable.

Catorce: Válvulas neumáticas de un solenoide. Las válvulas electro-neumáticas se utilizan para dirigir y distribuir el aire comprimido en los circuitos neumáticos usados en diferentes tipos de máquinas. Un número significativo de actuadores neumáticos, se controlan mediante válvulas de una única solenoide.

Para el caso se utilizan válvulas del tipo 5/2, lo cual significa cinco vías y dos posiciones. Entre las cinco vías hay una de entrada de aire, dos de salida (generalmente a través de exhostos) y dos de conexión a los acoples del actuador (de salida y de retorno). De las dos posiciones, una permite que la presión de aire se dirija del acople de salida al de retorno (solenoide desactivada) y la otra permite que sea del acople de retorno al de salida (solenoide activada). En tal caso el cilindro transmitirá la fuerza del aire en ambos sentidos, por lo que se dice que tal cilindro neumático es de doble efecto.



En la imagen se aprecia en el extremo derecho, el diagrama del circuito neumático, en el cual se destaca la unidad de mantenimiento, la válvula con su solenoide y su muelle de retorno, así como el cilindro neumático.

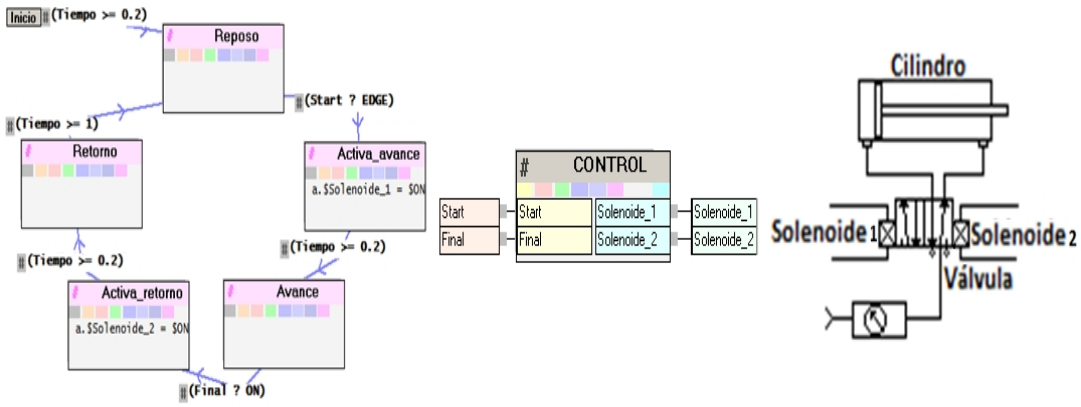
Para controlar un actuador neumático del tipo 5/2, de un solenoide, se utiliza una salida directa del autómata programable. Si el solenoide opera a 12 ó 24 VDC, se puede conectar directamente al autómata programable. Para válvulas con solenoide de 24, 110 ó 220 VAC, se deben utilizar relevos electro-mecánicos o relevos de estado sólido entre la salida del autómata y el solenoide.

Cuando la secuencia está en el estado “Reposo” y se activa la entrada “Start” (por flanco), se pasa al estado “Avance”, en el cual se activa la salida “Solenoide”, que hace que el émbolo del cilindro salga. Cuando el émbolo del cilindro hidráulico llega al sensor de final de carrera, se activa la entrada “Final”, tras lo cual se pasa al estado “Retorno”, durante el cual se espera 1 segundo a que el cilindro regrese a su posición inicial y la secuencia pase al estado “Reposo”.

Quince: Válvulas neumáticas biestables. Existen válvulas neumáticas de doble solenoide, de las cuales una hace que la presión se dirija del acople de salida hacia el de retorno y la otra hace que la presión se dirija en sentido contrario.

La utilidad de estas válvulas consiste en que conservan la posición del cilindro neumático en los casos en los cuales el sistema se des energiza, lo cual hace que sean útiles en un número importantes de aplicaciones.

Las válvulas de doble solenoide de dos posiciones, cambian de una posición a la otra al aplicar pulsos cortos en el respectivo solenoide. Al aplicar un pulso de 200 mS en el solenoide 1, la válvula pasa a la posición en la cual la presión se dirige del acople de salida al acople de retorno. Al aplicar un pulso de 200 mS en el solenoide 2, la válvula pasa a la posición en la cual la presión se dirige del acople de retorno al de salida.



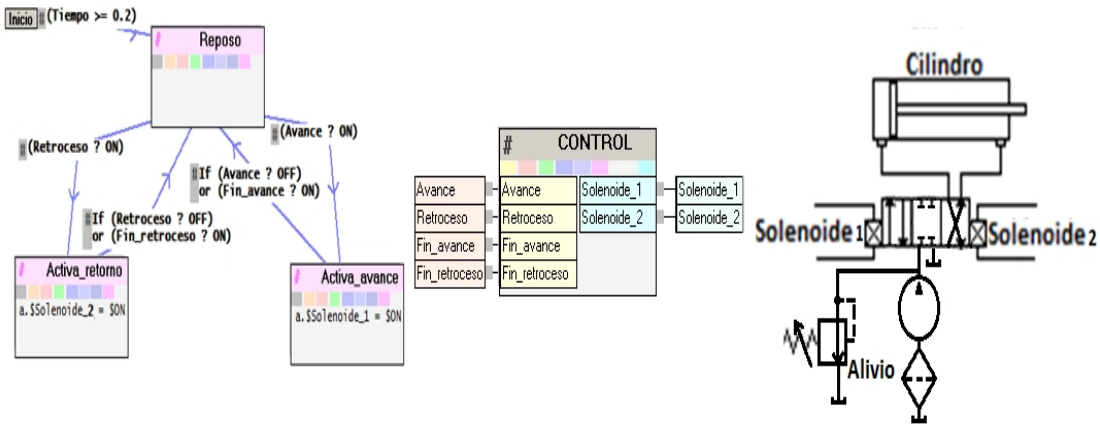
En la imagen se aprecia en el extremo derecho, el diagrama del circuito neumático, en el cual se destaca la unidad de mantenimiento, la válvula con sus dos solenoides, así como el cilindro neumático.

Para controlar la posición del cilindro neumático se utilizan dos salidas del autómata programable, directamente o a través de relevos, de acuerdo al voltaje de los solenoides.

La secuencia inicia en el estado “Reposo” y al activarse la entrada “Start” (por flanco), se pasa al estado “Activa_avance”, en el cual se activa la salida “Solenoide_1” durante 0.2 segundos y luego se desactiva de nuevo, lo cual hace que el émbolo del cilindro salga. Cuando el émbolo del cilindro hidráulico llega al sensor de final de carrera, se activa la entrada “Final”, tras lo cual se pasa al estado “Activa_retorno”, en el cual se activa la salida “Solenoide_2” durante 0.2 segundos y luego se desactiva de nuevo para pasar al estado “retorno”, durante el cual se espera 1 segundo a que el cilindro regrese a su posición inicial y la secuencia pase al estado “Reposo”.

Dieciséis: Válvulas hidráulicas de trabajo. Las válvulas electro-hidráulicas se utilizan para controlar el flujo de fluidos en los circuitos hidráulicos usados en diferentes tipos de sistemas. Un número significativo de actuadores hidráulicos, se controlan mediante válvulas de dos solenoide del tipo 5/3 (5 vías, 3 posiciones) ó 4/3 (4 vías, 3 posiciones).

Las válvulas electro-hidráulicas del tipo 5/3, tienen cinco vías, entre la cuales hay una de entrada de aceite desde la bomba, dos de retorno al tanque y dos de conexión a los acoples del actuador (de salida y de retorno). De las tres posiciones, una permite que el flujo de aceite se dirija del acople de salida al de retorno, la otra permite que sea del acople de retorno al de salida y la tercera hace que el flujo pase directo al tanque (centro abierto) o esté bloqueado (centro cerrado). Las válvulas del tipo 4/3 tienen sólo una salida de retorno al tanque. Los cilindros hidráulicos son generalmente de doble efecto.

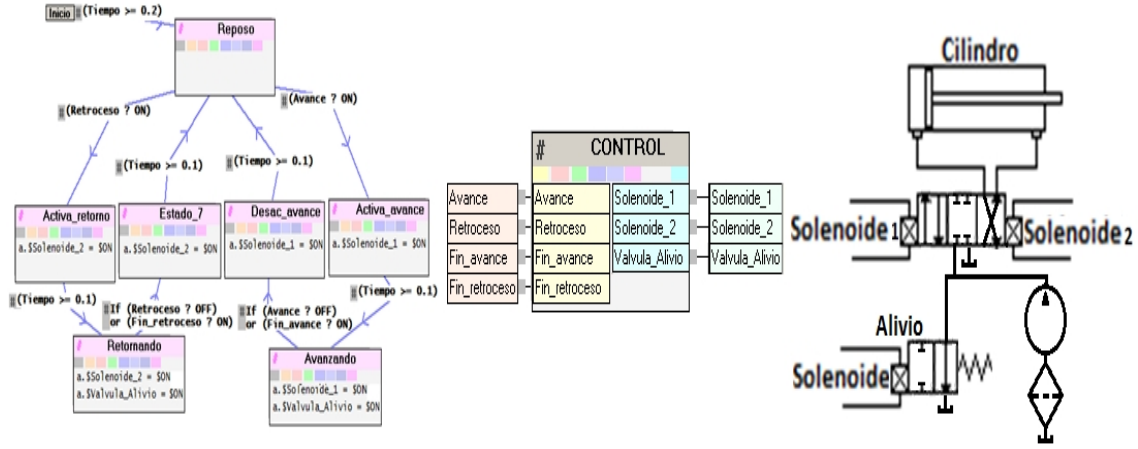


En la imagen se aprecia en el extremo derecho, el diagrama del circuito hidráulico, en el cual se destacan la bomba, la válvula de centro cerrado con sus dos solenoides, así como el cilindro hidráulico y la válvula de alivio por exceso de presión.

Para controlar la posición del cilindro hidráulico se utilizan dos salidas del autómata programable, directamente o a través de relevos, de acuerdo al voltaje de los solenoides.

La secuencia inicia en el estado “Reposo” y al activarse la entrada “Avance”, se pasa al estado “Activa_avance”, en el cual se activa la salida “Solenoide_1”, lo cual hace que el émbolo del cilindro salga. Cuando el émbolo del cilindro hidráulico llega al sensor de final de carrera o se desactiva “Avance”, se pasa al estado “Reposo”. Al activarse la entrada “Retroceso”, se pasa al estado “Activa_retorno”, en el cual se activa la salida “Solenoide_2”, tras lo cual el émbolo del cilindro retrocede y al llegar al sensor de inicio de carrera o al desactivar “Retroceso”, se pasa al estado “Reposo” (Los actuadores hidráulicos deben tener los sensores de inicio y final de carrera).

Diecisiete: Válvulas hidráulicas de alivio. Cuando se utiliza un grupo válvulas hidráulicas, las cuales deben ser de centro cerrado, es importante utilizar válvulas de alivio, para aquellos momentos en los cuales las válvulas de trabajo están desactivadas. Estas válvulas pueden operar por acción de la presión, la cual vence los muelles de la válvula de alivio al estar desactivados los solenoides de las válvulas de trabajo. Otras pueden operar controladas por el autómata programable, de tal manera que por defecto dirigen el flujo hidráulico desde la bomba hacia el tanque y bloquean el flujo al ser activadas, para que este pase a las válvulas de trabajo.



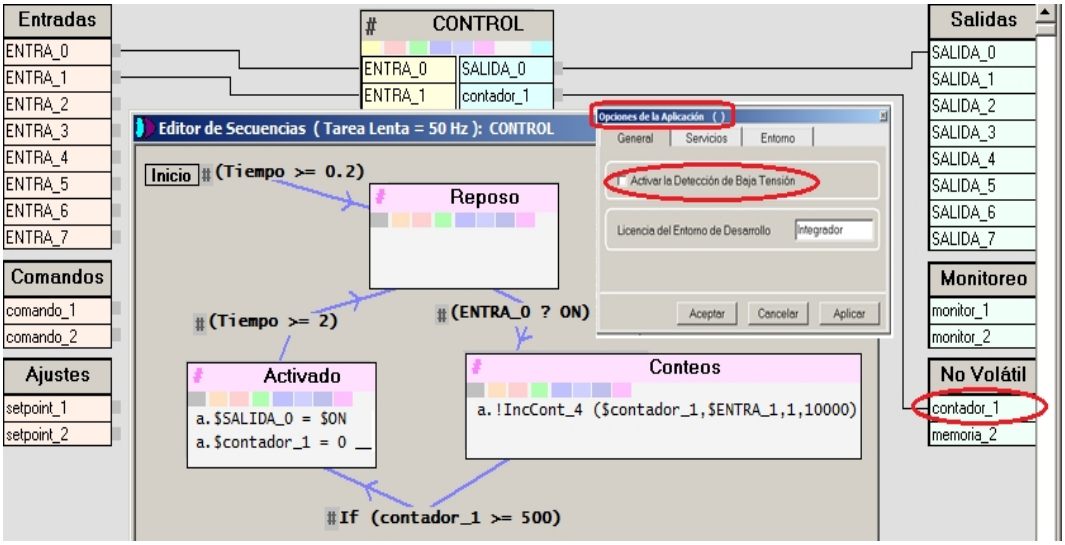
En la imagen se aprecia en el extremo derecho, el diagrama del circuito hidráulico, en el cual se destacan la bomba, la válvula de centro cerrado con sus dos solenoides, así como el cilindro hidráulico y la válvula de alivio con un solenoide.

Para controlar la posición del cilindro hidráulico se utilizan tres salidas del autómata programable, directamente o a través de relevos, de acuerdo al voltaje de los solenoides.

La secuencia inicia en el estado “Reposo” y al activarse la entrada “Avance”, se pasa al estado “Activa_avance”, en el cual se activa la salida “Solenoide_1” y luego de 0.1 segundos se activa también “Valvula_alivio”, lo cual hace que el émbolo del cilindro salga y al llegar al sensor de final de carrera o se desactivarse “Avance”, se pasa al estado “Reposo”, desactivando primero “Valvula_alivio” y luego de 0.1 segundos “Solenoide_1”. Al activarse la entrada “Retroceso”, se pasa al estado “Activa_retorno”, en el cual se activa la salida “Solenoide_2” ” y luego de 0.1 segundos se activa también “Valvula_alivio”, tras lo cual el émbolo del cilindro retrocede y al llegar al sensor de inicio de carrera o al desactivar “Retroceso”, se pasa al estado “Reposo”, desactivando primero “Valvula_alivio” y luego de 0.1 segundos “Solenoide_2” (Los actuadores hidráulicos deben tener los sensores de inicio y final de carrera).

Dieciocho: Operación No-volátil. En los casos en los cuales se desea conservar el estado de un proceso entre sucesivos ciclos de apagado y encendido, se utilizan las variables no-volátiles para guardar y recuperar la información que se desea conservar.

Este recurso es especialmente útil cuando se llevan conteos tanto de eventos como de tiempo y cuando se tienen variables que guardan el status de operación de una secuencia, para ser recuperado tan pronto como se energiza el autómata programable.



Para que el autómata programable almacene la variables N0-volátiles, es necesario hacer check en la casilla “Activar la Detección de Baja Tensión” en la pestaña “General” de la ventana de “Opciones de la aplicación”, la cual se abre pulsando click derecho sobre el fondo de la ventana de aplicación y seleccionando la opción “Propiedades Aplicación”.

Cuando está operando el servicio de detección de baja tensión, el sistema operativo (RTOS) almacena las variables No-volátiles en el momento en que se des energiza el autómata programable. Al energizarse el autómata, el RTOS carga las variables No-volátiles con el contenido almacenado previamente, en la memoria flash.

La secuencia inicia en el estado “Reposo” y al activarse la entrada “ENTRA_0”, se pasa al estado “Conteos”, en el cual se incrementa la variable No-volátil “Contador_1” de a uno, hasta el valor 10000 cada que hay una activación de la entrada “ENTRA_1”. Tan pronto como “Contador_1” sea mayor o igual a 500, se pasa al estado “Activado”, en el cual se activa la salida “SALIDA_0” y se borra “Contador_1” (se hace igual a cero). Luego de 2 segundos se regresa al estado “Reposo”. Si en cualquier momento se des energiza el autómata programable, no se perderá el valor de “Contador_1”.