GUIA DE MANEJO DE VARIABLES CONTINUAS CON EL AUTÓMATA PROGRAMABLE AMP1-E

INTRODUCCIÓN

Las variables continuas, en contraposición al concepto de variable discreta, son aquellas que pueden tomar todos los valores dentro de un intervalo determinado (en los autómatas Amp1, los valores enteros entre 0 y 65535). Las variables continuas son el recurso fundamental de operación de los autómatas programables del modelo Amp1, por lo que no es necesario declarar variables para desarrollar programas.

Esta guía orienta a los integradores de soluciones de automatización, en la utilización de las variables continuas, las cuales están ubicadas en la memoria RAM y en la memoria FLASH para servir como medio de almacenamiento (variables internas y especiales) y de interfaz hacia los diferentes periféricos del autómata programable (Entradas, salidas, comunicaciones y eventos).

En la guía se presentan las precauciones básicas para el correcto aprovechamiento, los recursos de trabajo con variables continuas, los tipos de variables continuas, su conexión, manipulación, escalamiento, filtrado y utilización en transiciones y maniobras para construir diferentes aplicaciones de monitoreo y control.

PRECAUCIONES Y ACCIONES INICIALES

Conecte el autómata programable a voltajes menores que 30 VDC. Revise que la fuente de alimentación esté adecuadamente aterrizada.

Alimente los circuitos de entradas continuas desde fuentes de alimentación que no inyecten ruidos que alteren las señales de salida de los sensores.

Seleccione los cables de las entradas al Amp1 de acuerdo al tipo de señal conducida, para evitar alteraciones de las mediciones por interferencia electromagnética.

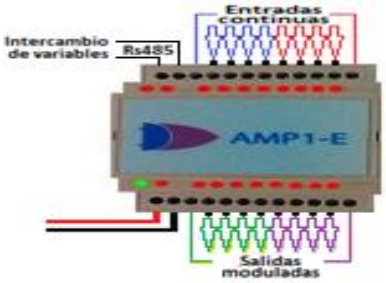
No conecte las entradas a voltajes mayores a 30 voltios de corriente directa, para entradas tipo U (universal), ni mayores que 250 voltios de corriente alterna para entradas tipo H (tensión > 50 V).

En caso de alimentar los sensores de manera directa desde una fuente de alimentación externa, verifique que sea de 12 ó 24 VDC +/- 5%. En este caso asegúrese de unir las tierras de las entradas del autómata programable y de la fuente de alimentación de los sensores.

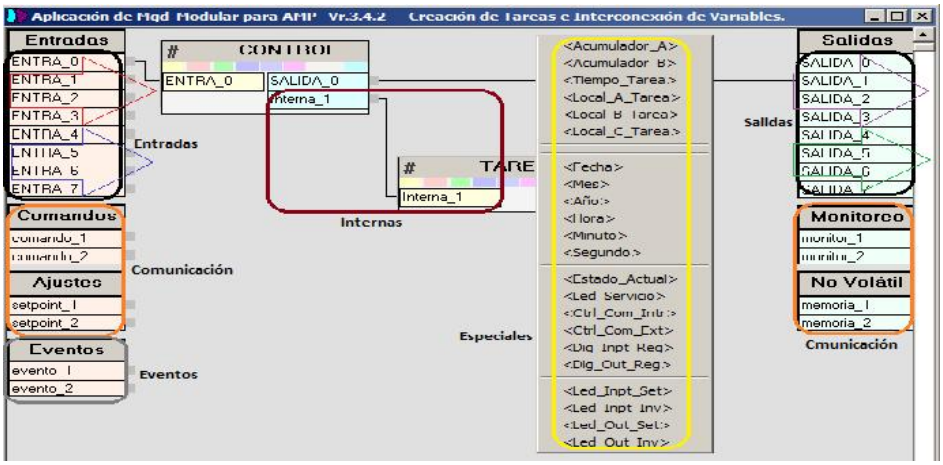
En caso de no disponer del software Mgdmod, descárguelo desde www.tecvolucion.net/descargas e instálelo según las instrucciones presentadas en el documento “Instalación del software Mgdmod” el cual se descarga desde www.tecvolucion.net/soporte.

RECURSOS PARA EL MANEJO DE LAS VARIABLES CONTINUAS

El conjunto de recursos para el manejo de las variables continuas se proyecta desde la interfaz del autómata programable con su entorno. Las entradas del autómata programable son variables analógicas en el rango 1 a 10 voltios, las cuales pueden recibir trenes de pulsos hasta de 3000 Hz. Las salidas pueden ser moduladas en el rango 0 a 1000 (pwm, tren de ondas, servo, stepper, brushless, ángulo de fase). Las comunicaciones intercambian variables en el rango 0 a 65535 (input register, holding register).



Al interior del autómata programable, las variables continuas están ubicada en la memoria RAM (entradas, salidas, internas, especiales, comandos, monitoreo, no volátil, eventos) y en la memoria FLASH (Ajustes, respaldo de no volátil).



Las variables continuas del autómata programable son enteros sin signo en el rango 0 a 65535, lo cual permite la interconexión de las variables de interfaz de los diferentes periféricos con las entradas y salidas de las tareas y la interconexión de las variables de las tareas entre sí.

Al interior de las tareas, las variables continuas pueden ser consultadas en las condiciones de las transiciones, tanto por su valor continuo, como por su comportamiento booleano. En las maniobras (de los estados o bloques de maniobras) pueden ser cargadas con valores constantes o valores provenientes de otras variables (valores continuos o valores booleanos).

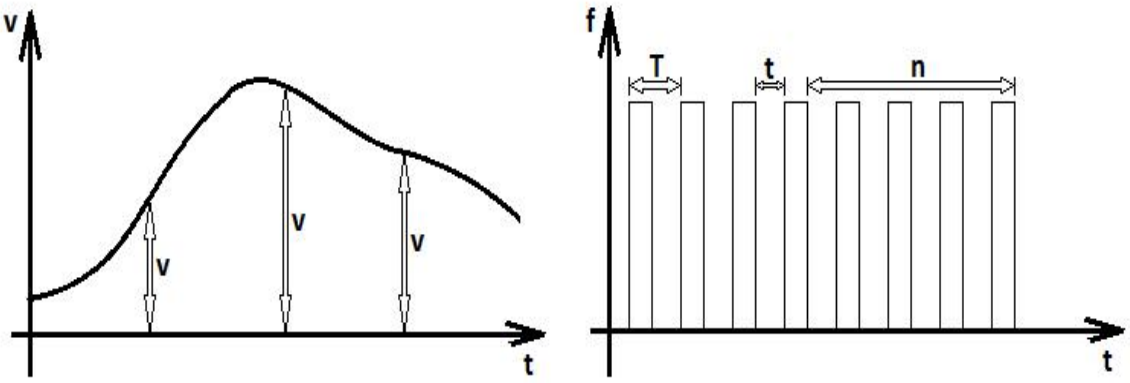
El comportamiento booleano de una variable en el autómata programable se refiere al estado desactivado (falso) cuando su valor es cero y activo (verdadero) cuando es diferente de cero.

TIPOS DE ENTRADAS CONTINUAS

Las variables continuas, en el autómata programable, son de dos tipos fundamentales:

Variables analógicas, las cuales están en el rango 0 a 1023 (0 a 4095 para Amp1-G y Amp1-P) y corresponden al resultado de la conversión análogo-digital en las entradas del autómata programable (V).

Variables frecuenciales, las cuales están en el rango 0 a 1000 y corresponden a los valores del período (T), el ancho de pulso (t), la cantidad de pulsos (n) o la frecuencia (f) de una entrada o salida del autómata programable.



Al desarrollar aplicaciones con el autómata programable, ambos tipos de variables continuas se manipulan de la misma forma, pudiendo consultar su valor, escribir en ellas, escalarlas, filtrarlas, hacer operaciones entre ellas, generar modulaciones en las salidas a partir de su valor, almacenarlas en memoria flash e intercambiarlas con agentes externos a través de los buses de comunicación.

Cuando la variable continua representa una magnitud física que corresponde a un número con cifras decimales, se debe utilizar numeración de punto fijo. Esto es, se debe trabajar en décimas, centésimas o milésimas de la magnitud física (el autómata programable sólo opera con enteros).

Cuando la variable continua representa una magnitud física que corresponde a un número negativo, se debe desplazar el rango de numeración, tomando como valor inicial el valor negativo de mayor magnitud (cero corresponde al valor negativo con mayor magnitud). Esta estrategia aplica sólo a operaciones de asignación, de suma y resta.

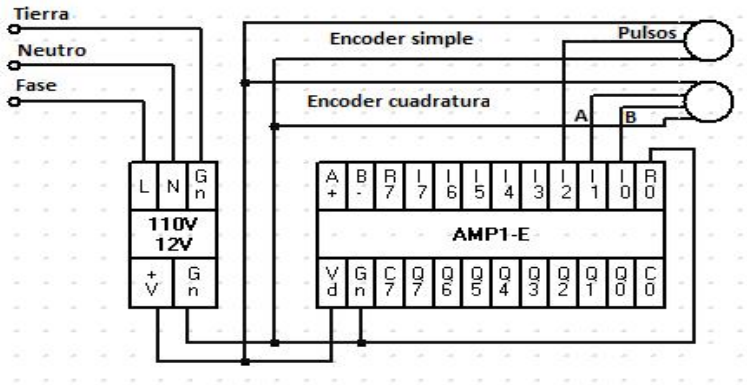
Cuando se intercambian variables entre tareas, es importante considerar que una tarea rápida puede ver todos los cambios realizados en una variable por parte de una tarea más lenta, pero una tarea lenta no ve todos los cambios realizados en una variable por una tarea más rápida.

Para destacar las particularidades de interconexión y la utilización de algunas funciones específicas, se estudia por separado la conexión y la manipulación de las variables analógicas y frecuenciales.

CONEXIÓN DE VARIABLES FRECUENCIALES AL AUTÓMATA PROGRAMABLE

Las variables frecuenciales se pueden generar al interior del autómata programable o ser recibidas desde dispositivos externos como pulsadores, generadores/sensores de pulsos o encoder.

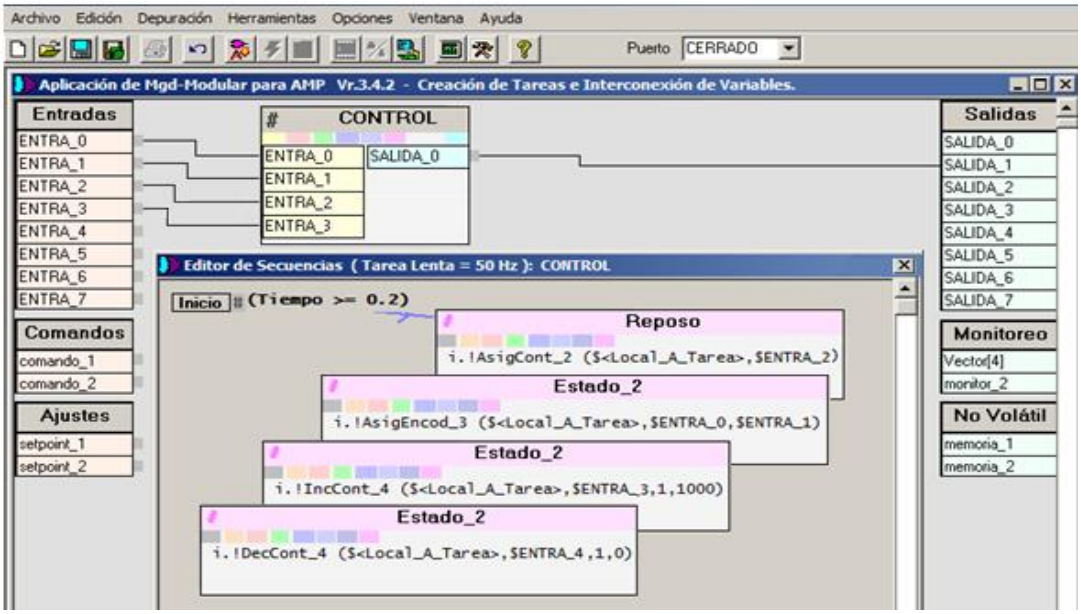
La imagen siguiente muestra la forma de conectar algunos de esos dispositivos al Amp1. Se debe revisar la placa del dispositivo para evitar aplicar voltajes que lo puedan dañar.



Ambos dispositivos están conectados a 12 VDC. El sensor de pulsos entrega pulsos positivos (o "push-pull") por la entrada "I2" y el encoder hace lo mismo por las entradas "I0" e "I1".

MANIPULACIÓN DE VARIABLES FRECUENCIALES

Las variables continuas de origen frecuencial se obtienen mediante la utilización de funciones de configuración de los puertos de entrada y de funciones de conteo de eventos internos o externos.



Las funciones utilizadas para la generación de valores de origen frecuencial se deben ubicar en tareas cuya frecuencia de ejecución sea mayor a la frecuencia de la señal frecuencial detectada. Estas funciones se describen a continuación.

AsigCont_2 (Acumulador_A>, $<Acumulador_A>): Asignar a la Entrada Contador Encoder a ARG1 con pulsos en ARG2. Realiza un conteo ascendente en el contador ARG1 incrementándolo cada que hay flancos ascendentes en la entrada ARG2. Se utiliza para contar eventos en los puertos de entrada y en otras variables del autómata programable.

ARG1 = Variable contador, ARG2 = Entrada de pulsos.
if(Edge(ARG2)) ARG1 += 1.

AsigEncod_3 ($<Acumulador_A>, $<Acumulador_A>, $<Acumulador_A>): Asignar a la Entrada Contador Encoder a ARG1 (ARG2=A, ARG3=B). Asignar Contador Encoder a dos puertos del autómata programable. Asignar a las entradas "A" y "B" en cuadratura, la función de contador ascendente descendente. Se utiliza para realizar conteos absolutos con Encoder de cuadratura.

ARG1 = Variable de conteo, ARG2 = Puerto entrada A, ARG3 = Puerto entrada B.
if(ARG2->ARG3) ARG1++, if(ARG2<-ARG3) ARG1- -.

Las variables de salida de las dos funciones anteriores se utilizan en las aplicaciones como variables continuas para ser utilizadas tanto en maniobras como en condiciones de transición.

Las entradas a las cuales han sido asignadas las configuraciones de contador o encoder, pueden ser escaladas y filtradas como si fueran variables analógicas.

IncCont_4 ($<Acumulador_A>, $Impulso_1_Seg, 1, 1000): Contador Incremental ARG1 += ARG3 hasta ARG4 al haber flanco en ARG2. Incrementa la variable ARG1en el valor ARG3 al haber flanco positivo en la variable ARG2, hasta el valor ARG4. Se usa para contar eventos incrementalmente.

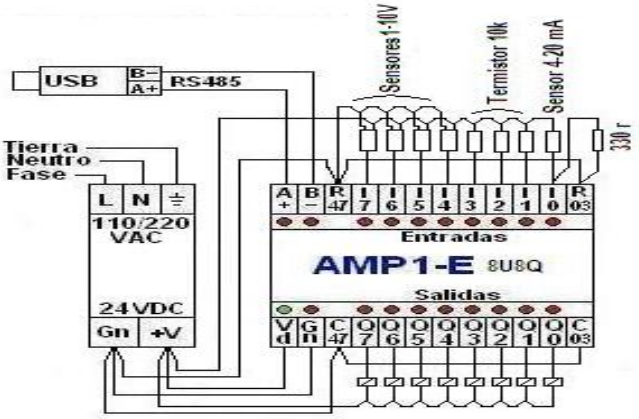
ARG1 = Variable contador, ARG2 = Variable pulsos, ARG3 = Valor de incremento, ARG4 = Máximo valor de conteo.
If ((Edge(ARG2)) && (ARG1<ARG4)) ARG1 += ARG3.

DecCont_4 ($<Acumulador_A>, $Impulso_1_Seg, 1, 0): Contador Decremental ARG1 -= ARG3 hasta ARG4, al haber flanco en ARG2. Decrementa la variable ARG1 en el valor ARG3 al haber flanco positivo en la variable ARG2, hasta el valor ARG4. Se usa para contar eventos decrementalmente.

ARG1 = Variable contador, ARG2 = Variable pulsos, ARG3 = Valor de decremento, ARG4 = Mínimo valor de conteo.
If ((Edge(ARG2)) && (ARG1>ARG4)) ARG1 -= ARG3.

CONEXIÓN DE VARIABLES ANALÓGICAS AL AUTÓMATA PROGRAMABLE

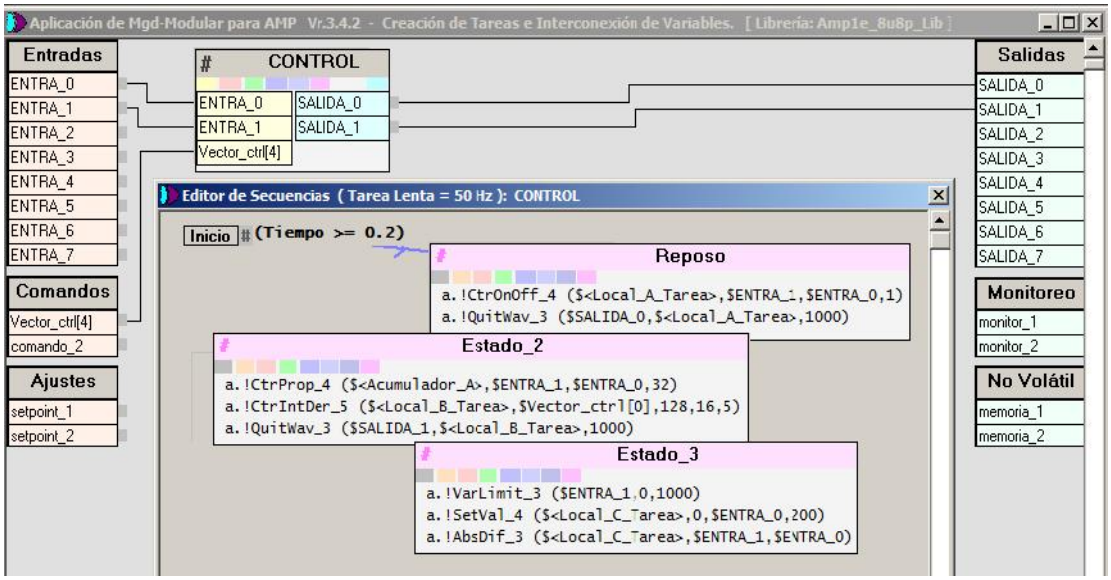
Las variables analógicas se obtienen mediante las entradas del autómata programable, las cuales son opto-aisladas con una zona de lineal entre 1 y 10 voltios. La imagen siguiente muestra la forma de conexión de sensores de tres terminales (1 – 10 voltios), de termistores de 10k, de transmisores de corriente (4 – 20 mA). La fuente de alimentación de las entradas puede ser la misma del Amp1, siempre y cuando no esté contaminada con interferencias electromagnéticas.



Es fundamental que la tierra de la fuente esté conectada al circuito de tierra de la acometida. Las entradas del autómata programable se comportan como la carga de un circuito eléctrico.

MANIPULACIÓN DE VARIABLES ANALÓGICAS

Las variables continuas de origen analógico se obtienen de las entradas del autómata programable y pueden ser interconectadas hacia y entre las tareas, para ser utilizadas en las condiciones de las transiciones y en las maniobras.



Las variables continuas (tanto de origen analógico como de origen frecuencial) se utilizan en las expresiones de las transiciones para hacer comparaciones de magnitud y en las maniobras (de los estados y/o bloques de maniobras) para la escritura de expresiones y funciones, las cuales estructuran la lógica de los programas de aplicación.

Algunas funciones importantes en la manipulación de las variables continuas son las de control ON-OFF, de control proporcional-integral-derivativo, las cuales entregan como resultado valores en variables que pueden ser utilizadas como argumento en las funciones de salida hacia los puertos del autómata programable, para generar diferentes tipos de modulación de la potencia aplicada a los actuadores. Las funciones de control y de salida se explican en las secciones de "UTILIZACIÓN DE VARIABLES CONTINUAS EN FUNCIONES DE CONTROL" y "UTILIZACIÓN DE VARIABLES CONTINUAS EN FUNCIONES DE SALIDA" respectivamente.

Hay otro grupo de funciones que realizan diferentes tipos de manipulación de las variables continuas. Las funciones de este tipo deben ser ubicadas en puntos estratégicos del programa (antes de las maniobras que han de utilizar su resultado). Estas funciones son:

VarLimit_3 ($<Acumulador_A>, 0, 1000): Limitador de variable en un rango de valores. Limita la variable ARG1 entre los valores ARG2 y ARG3. Compara la variable ARG1, si es menor que el valor ARG2 se hace igual a ARG2, si es mayor que ARG3 se hace igual a ARG3. Se usa para limitar el valor de una variable en un rango de valores.

ARG1 = Variable a limitar, ARG2 = Mínimo valor de la variable, ARG3 = Máximo valor de la variable.
ARG1 = [ARG2 ... ARG3].

Cuando una variable ha sido limitada a un valor, este se conserva en la variable hasta la siguiente vez que el sistema operativo actualice la variable. Por ello la función de limitación de variable debe estar en tareas que se ejecuten con la misma frecuencia de la actualización de la variable por parte del sistema operativo del autómata programable (generalmente a 50 Hz).

SetVal_4 ($<Acumulador_A>, 0, $<Acumulador_A>, 200): Asignación condicional de valor a variable. Asigna el valor ARG2 a la variable ARG1 cuando ARG3 >= ARG4. Asigna el valor en ARG2 a la variable ARG1 cuando la variable ARG3 es mayor o igual que el valor ARG4. Se usa para reiniciar contadores de eventos IncCont y DecCont.

ARG1 = Variable a modificar, ARG2 = Valor a asignar, ARG3 = Variable de entrada, ARG4 = Valor de comparación.
If (ARG3 >= ARG4) ARG1 = ARG2.

AbsDif_3 ($<Acumulador_A>, $<Acumulador_A>, $<Acumulador_A>): Valor absoluto de la diferencia entre dos valores. Carga en ARG1 el valor absoluto de la diferencia entre ARG2 y ARG3. Carga en la variable ARG1 el valor absoluto de la diferencia entre las variables ARG2 y ARG3. Se usa para determinar si dos variables son similares.

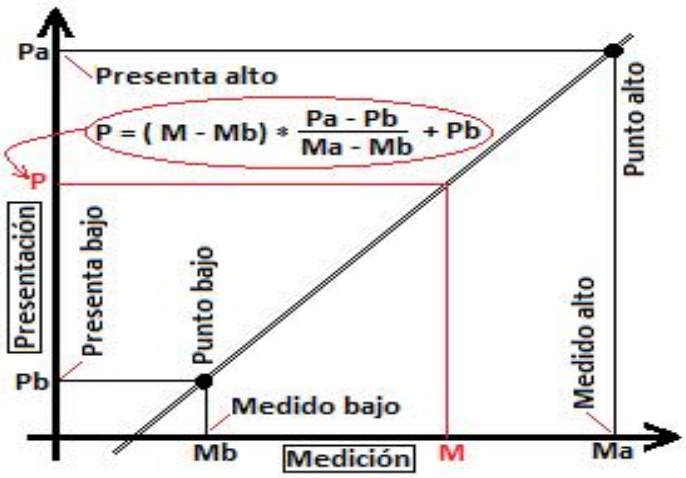
ARG1 = Variable de salida, ARG2 = Variable minuendo, ARG3 = Variable sustraendo.
ARG1 = Abs(ARG2 - ARG3).

ESCALAMIENTO DE VARIABLES CONTINUAS

Las variables continuas adquiridas a través de las entradas del autómata programable (tanto de origen frecuencial como de origen analógico) pueden ser escaladas desde unidades de conteo o de conversión análogo-digital a unidades de ingeniería (voltaje, temperatura, presión, frecuencia, nivel, longitud, peso, etc).

El principio del escalamiento de variables continuas de entrada, consiste en la aplicación de la ecuación de la recta que no pasa por el origen, para obtener (en cada ciclo de programa) el valor de la entrada en unidades de ingeniería. Para tal fin es necesario que el valor medido tenga una relación lineal con la variable física que representa y que estén programados cuatro valores de escalamiento que son:

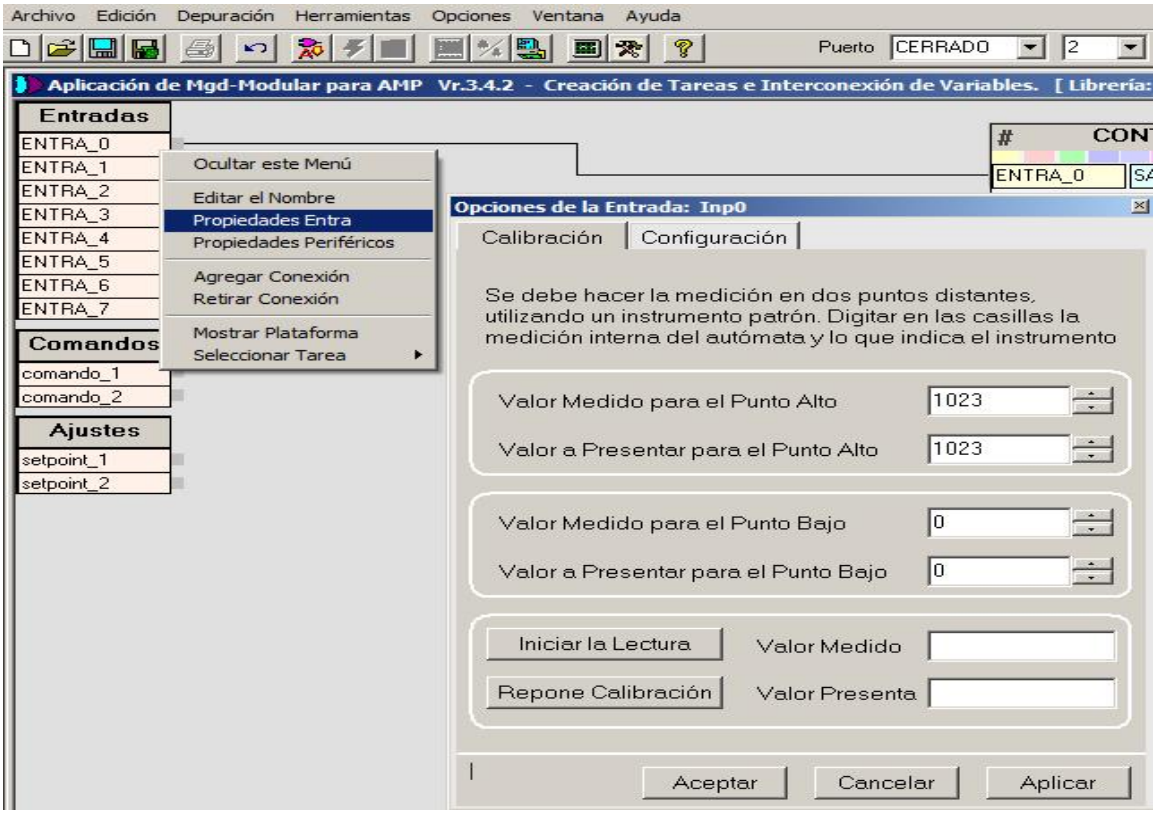
Pb = Presenta bajo. Es el menor valor útil que se puede obtener de la variable física.
Mb = Medición bajo. Es el valor que se mide en el Amp1 al conectar un sensor midiendo “Pb”.
Pa = Presenta alto. Es el mayor valor útil que se puede obtener de la variable física.
Ma = Medición alto. Es el valor que se mide en el Amp1 al conectar un sensor midiendo “Pa”.



El escalamiento lo realiza continuamente el sistema operativo del autómata programable, leyendo la variable de entrada y calculando la variable escalada a unidades de ingeniería. Las variables que se utilizan para conectar a las tareas y luego dentro de estas, son el resultado del proceso de escalamiento por parte del sistema operativo del autómata programable (variable a presentar).

Al abrir un nuevo programa, las variables de entrada están configuradas con un escalamiento de uno a uno. Es decir, el valor de la variable escalada es el mismo de la variable medida.

Para realizar la operación de escalamiento de una entrada, se debe pulsar click derecho sobre la casilla de la variable de entrada y seleccionar la opción “Propiedades entra”, luego de los cual se abrirá la ventana de configuración de las entradas.



La ventana de calibración de cada entrada, permite ver el valor medido por la entrada del autómata programable, tanto en unidades de conversión análogo-digital o conteo (“Valor Medido”), como en unidades de ingeniería (“Valor Presenta”), luego de pulsar el botón “Iniciar la Lectura”, estando el autómata encendido, conectado al PC y con el puerto abierto.

Para realizar el escalamiento se debe contar con un instrumento patrón que muestre el valor de la variable física, en las unidades de ingeniería que se desee utilizar en el programa de aplicación.

Se debe llevar la variable física al menor valor posible y que esté dentro del rango útil de medición (rango de trabajo). El valor de la casilla “Valor Medido” se copia a la casilla ”Valor Medido para el Punto Bajo”, el valor de la variable física (presentado por un instrumento patrón) se escribe en la casilla ”Valor a presentar para el Punto Bajo”. Luego de esto se pulsa el botón “Aplicar”.

Se debe llevar la variable física al mayor valor posible y que esté dentro del rango útil de medición (rango de trabajo). El valor de la casilla “Valor Medido” se copia a la casilla ”Valor Medido para el Punto Alto”, el valor de la variable física (presentado por un instrumento patrón) se escribe en la casilla ”Valor a presentar para el Punto Alto”. Luego de esto se pulsa el botón “Terminar Lectura” y para finalizar, el botón “Aceptar”.

Luego se descarga el programa al autómata y se abre de nuevo la ventana de configuración de la entrada en cuestión, se pulsa el botón “Iniciar la Lectura”. Se debe observar que lo que muestra la ventana “Valor Presenta” coincide o es cercano a lo que presenta el instrumento patrón.

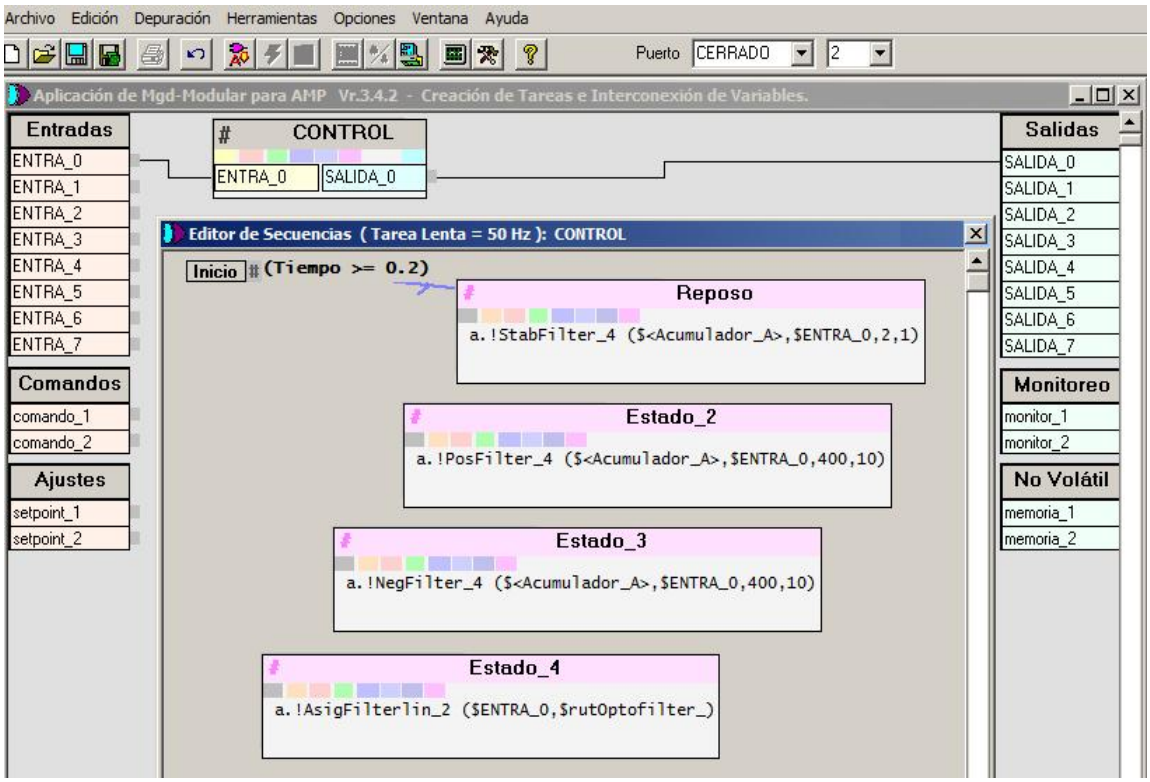
FILTRADO DE VARIABLES CONTINUAS

La utilidad fundamental de la medición de variables continuas, está en la posibilidad de realizar acciones de control y/o de hacer adquisición de datos (monitoreo). Por ello es fundamental que la variable devuelva valores lo más próximos a las magnitudes reales que están siendo medidas. Esto tiene que ver, no sólo con la linealidad del sistema de medición y con el correcto escalamiento de la variable, sino también con la ausencia de señales de interferencia que alteren los valores medidos.

En los casos en los cuales por una u otra causa se tienen mediciones acompañadas de perturbaciones, es necesario utilizar funciones de filtraje que eliminen las interferencias que alteran la medición. Esta situación se evidencia al revisar la variable de entrada y observar que el valor de la medición fluctúa de forma no esperada (se puede revisar en la casilla “Valor Presenta” de la ventana de propiedades de la entrada, luego de pulsar el botón “Iniciar la Lectura”).

El sistema operativo del autómata programable hace un filtrado consistente en la realización de un número determinado de mediciones (32) por cada entrada antes de extraer un promedio, lo cual filtra la fluctuación de la medición. En la pestaña “Configuración” de la ventana de “Propiedades Entra” se puede seleccionar la cantidad de promedios que se realiza por cada entrada del autómata programable (1…32).

Existen perturbaciones de diferentes tipos que no son contrarrestadas por el filtro por promedios de las entradas. Para estos casos existe un grupo de funciones de filtraje de las entradas y de variables internas, las cuales pueden ser utilizadas en cualquiera de las maniobras de las tareas.



De los filtro de variable continua, hay tres que operan en las tareas para obtener nuevas variables con los valores filtrados (StabFilter, PosFilter y NegFilter) y uno que configura la entrada para que el sistema operativo realice una linealización de la variable de entrada (AsigFilterlin) antes del escalamiento, útil para aumentar la precisión y para las mediciones con termistores.

StabFilter_4 ($<Acumulador_A>, $<Acumulador_A>, 2, 1): Filtro de estabilización de variables. Filtro de estabilización de entrada ARG1 (ARG2=resultado, ARG3=umbral, ARG4=ciclos). Filtro de estabilización de la variable ARG1 hacia ARG2 cuando el cambio de ARG1 es menor que ARG3 al cabo de ARG4 ciclos de la Tarea. La variable ARG2 conserva su valor en caso contrario. Se usa para estabilizar variables usadas en TabFind

ARG1 = Variable salida, ARG2 = Entrada filtrar, ARG3 = Umbral compara, ARG4 = Cantidad ciclos.
If (Abs(ARG1(i)-ARG1(i - ARG4))<ARG3) ARG2 = ARG1.

PosFilter_4 ($<Acumulador_A>, $<Acumulador_A>, 400, 10): Filtro positivo de variable continua. Filtrar entrada ARG2 hacia ARG1 (ARG3=Umbral Superior, ARG4=Ciclos Retardo). Filtra la entrada ARG2 hacia la variable ARG1, activándola cuando la entrada sea mayor que el umbral superior durante los ciclos de retardo ARG4. Se utiliza para filtrar en amplitud y en frecuencia entradas digitales ruidosas. El umbral está en unidades de ADC.

ARG1 = Variable resulta, ARG2 = Variable entrada, ARG3 = Umbral superior, ARG4 Ciclos retardo.
if((ARG2 > ARG3) && (Tiempo > ARG4)) ARG1=1.

NegFilter_4 ($<Acumulador_A>, $<Acumulador_A>, 400, 10): Filtro negativo de entrada analógica. Filtrar entrada ARG2 hacia ARG1 (ARG3=Umbral Inferior, ARG4=Ciclos Retardo). Filtra la entrada ARG2 hacia la variable ARG1, activándola cuando la entrada sea menor que el umbral inferior durante los ciclos de retardo ARG4. Se utiliza para filtrar en amplitud y en frecuencia entradas digitales ruidosas. El umbral está en unidades de ADC.

ARG1 = Variable resulta, ARG2 = Variable entrada, ARG3 = Umbral inferior, ARG4 Ciclos retardo.
if((ARG2 < ARG3) && (Tiempo > ARG4)) ARG1=1.

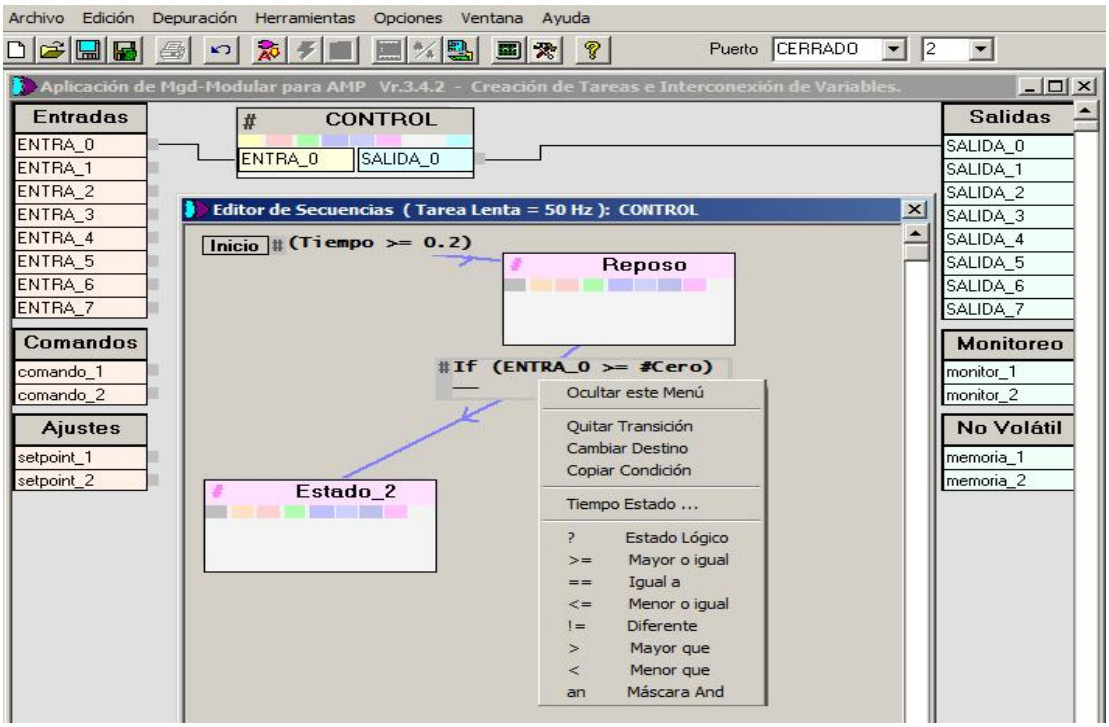
Las funciones “PosFilter” y “NegFilter” se pueden usar en cascada para conformar un filtro con comparación por encima y por debajo y operación simétrica o asimétrica en el tiempo.

AsigFilterlin_2 ($Port_Reg, $rutOptofilter): Asignar Filtro de linelización a puerto de entrada. Asignar a la Entrada Filtro de linelización ARG2[Longitud] al Puerto ARG1. Asigna un Filtro de linelización a través de la consulta de la tabla ARG2[Longitud], al puerto de entrada ARG1. Se utiliza para filtrar las entradas optoaisladas y las entradas con termistores y sensores no-lineales.

ARG1 = Puerto de entrada, ARG2 = Tabla de filtrado, Longitud = Cantidad de datos en la tabla.
Variable filtrada=Tabla[Entrada / 32].

UTILIZACIÓN DE VARIABLES CONTINUAS EN LAS TRANSICIONES

Una estructura fundamental para la construcción de la lógica de los programas de aplicación, son las **Transiciones**, las cuales implican el paso de un estado del proceso a otro. Cada transición tiene una condición, la cual consta de un conjunto de expresiones de comparación lógica y por magnitud (de variables continuas), unidas por conectivos lógicos (“an” y “or”).



Cada expresión de la condición de una transición revisa una variable continua y la compara en forma booleana (“?”) con un estado lógico (“On”, “Off”, “Edge”) o en magnitud (“>=”, “==”, “<=”, “!=”, “>”, “<”) con otra variable o constante. También se puede hacer una máscara (“an”) con otra variable o constante para obtener una condición verdadera si el resultado es diferente de cero.

En las transiciones, la variable “Tiempo Estado” (“Tiempo de la Tarea”) es una variable especial que compara horas, minutos, segundos y fracciones de segundo. Al ser tomada como variable continua, su valor será la cantidad de ciclos de programa transcurridos desde el inicio del estado.

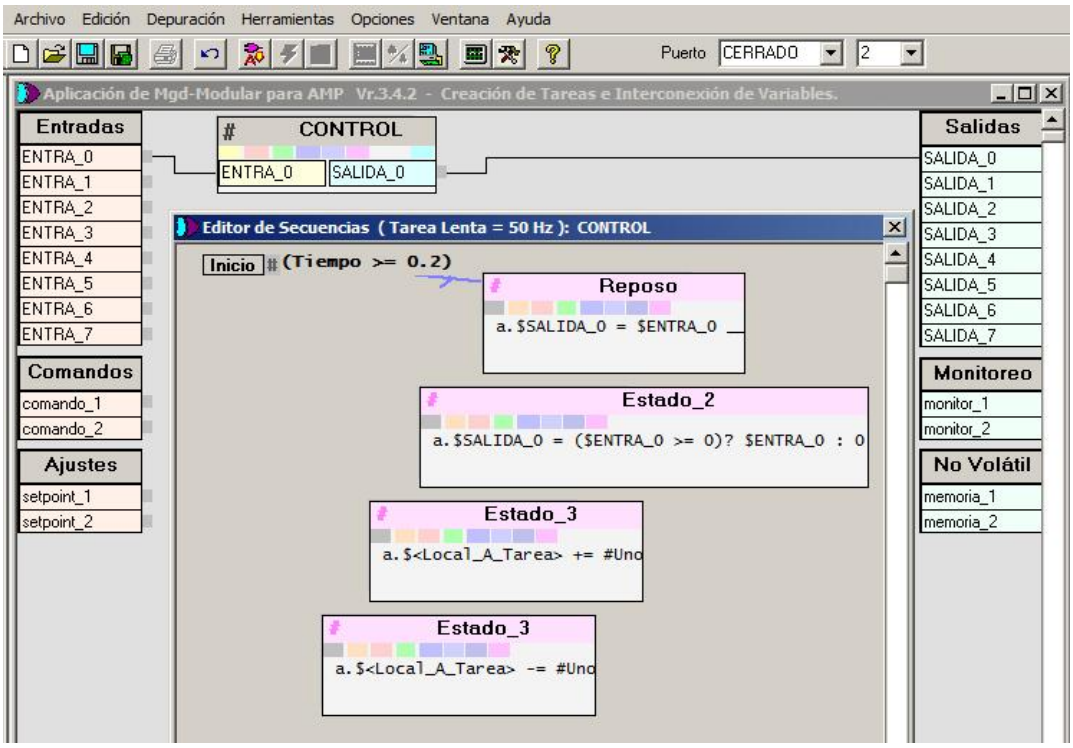
Las variables continuas son privadas para cada tarea (una tarea no puede ver las variables de otra), a menos que haya una conexión en la ventana de “Aplicación” que las comparta. La excepción a esta regla está en las variables del sistema (destellos y variables especiales diferentes de los acumuladores y variables de la tarea), que son variables públicas. También son públicas las constantes (valores y tablas).

Las expresiones de las condiciones de las transiciones pueden consultar las variables continuas pero no pueden alterarlas.

UTILIZACIÓN DE VARIABLES CONTINUAS EN LAS MANIOBRAS

La otra estructura fundamental para el desarrollo de programas de control lógico y continuo, son las **Maniobras**, las cuales se ubican en los estados y/o bloques de maniobras de las diferentes tareas. Cada una de las maniobras está constituida por una única expresión que puede ser de asignación simple (“var = oper”), de comparación (“var = (oper > oper)? oper : oper”), de operaciones lógicas y aritméticas de uno (“var += oper”) y dos operandos (var = oper + oper), de ejecución condicionada de varias maniobras (“if(oper > oper){”… “}”), de ejecución repetitiva un número exacto de veces (“for(oper = 0; < oper; + oper)” / “for(oper = K; > oper; - oper)” … “}”) y de ejecución de una función del sistema operativo o de usuario (“¡Function(oper, oper, oper)”).

La expresiones de las maniobras tiene un una letra inicial a la izquierda (antes del punto), la cual indica si la maniobra se ejecuta permanentemente mientras el estado esté activo (“a.”), al ocurrir una transición de activación (llegada) del estado (“i.”) o al ocurrir una transición de desactivación (salida) del estado (“f.”).



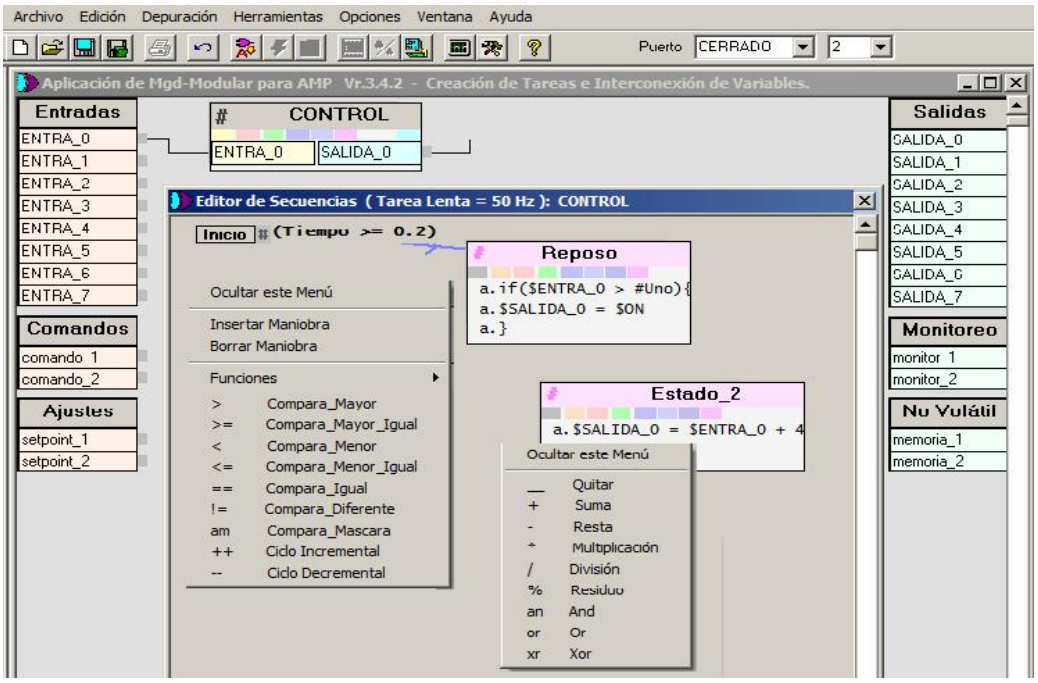
La imagen anterior muestra una maniobra de asignación de la entrada “ENTRA_0” a la salida “SALIDA_0” (la raya a la derecha permite agregar signos para construir operaciones de dos operandos).

La segunda maniobra asigna el valor de la variable “ENTRA_0” a la variable “SALIDA_0” cuando el valor de la variable “ENTRA_0” es mayor o igual a cero, y asigna “0” en caso contrario.

La tercera y la cuarta maniobras realizan la operación de incremento / decremento de la variable “<Local_A_Tarea>” en una unidad.

A cada estado o grupo de maniobras se le puede agregar un número indeterminado de maniobras, a través de las cuales se construyen las acciones que realizará el autómata programable, permitiendo construir lógicas complejas a partir de expresiones simples.

La imagen siguiente muestra dos casos especiales de construcción de estructuras de programación a través de la extensión y la combinación de las maniobras.



La maniobra del estado “Estado_2” Asigna el valor de la variable “ENTRA_0” más la constante “4” a la variable “SALIDA_0”. Esta expresión se construye pulsando click sobre la raya que hay a la derecha de la expresión de asignación simple (vínculo de extensión), tras lo cual aparece el menú con los signos “+, -, *, /, %, an, or, xr” o la opción quitar para regresar a una expresión simple.

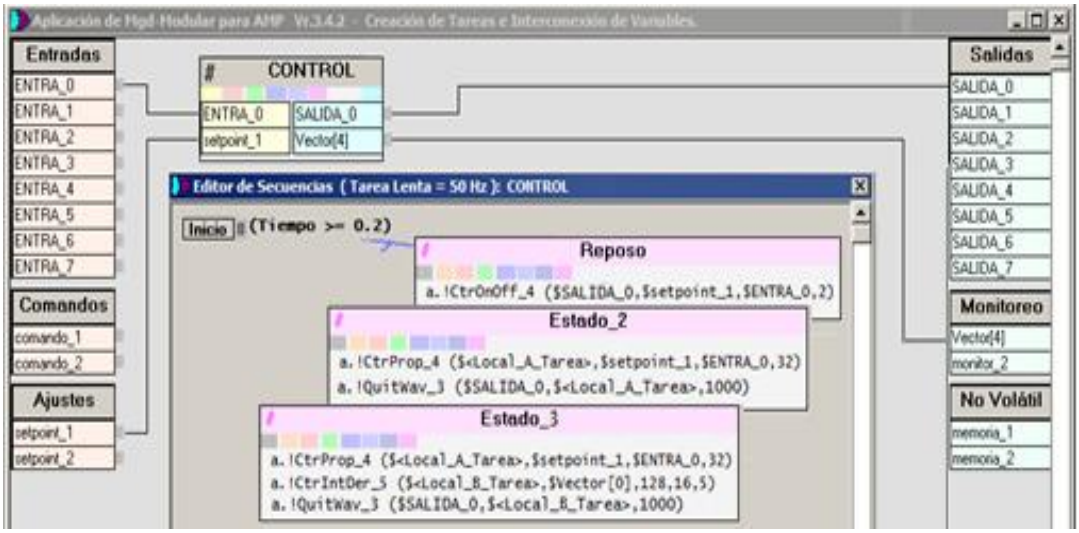
Las maniobras del estado “Reposo” conforman una combinación para activar la salida “SALIDA_0” en caso de que la variable “ENTRA_0” sea mayor que “1”. Esta combinación se construye agregando una maniobra y pegando cualquier variable disponible en la maniobra (se obtiene una asignación simple de activación). Luego sobre el signo “=” se hace click y aparece un menú con las opciones “>, >=, <, <=, ==, ¡=, an”, de las cuales se debe seleccionar una (“>”). Tras esto aparecen las maniobras “if(var > 0){” y “}”, para pulsar click sobre la segunda y seleccionar la opción “Insertar Maniobra”. Aparecerá una maniobra en medio de las llaves (“{}”), en la cual se pegará la variable “SALIDA_0”. Finalmente se pega la variable “ENTRA_0” a la primera maniobra.

Al seleccionar las opciones “++ Ciclo Incremental” o “-- Ciclo Decremental” aparecerá la estructura de programación “for”, la cual se edita de forma similar a la estructura “if”, teniendo en cuenta que el primer operando indica la variable de conteo de ciclos, el segundo indica el valor inicial de conteo, el tercero indica el último valor para realizar ciclos de ejecución y el último indica el valor de incremento / decremento de la variable por cada iteración del ciclo.

UTILIZACIÓN DE VARIABLES CONTINUAS EN FUNCIONES DE CONTROL

Nota: Para estudiar esta sección es recomendable conocer la teoría de control automático.

El control continuo de variables es una disciplina, que a pesar de ser clásica, abre posibilidades de solución a un número apreciable de problemas de control. Los autómatas programables Amp1 disponen de una serie de funciones de control de variable continua (control On-Off, control proporcional, control PID y la combinación de estos con máquinas de estado).



La imagen anterior muestra el estado “Reposo” con una maniobra de control On-Off, que obra sobre la salida “SALIDA_0”, a partir de la comparación del setpoint “Setpoint_1” (ubicado en memoria flash) con la variable de entrada “ENTRA_0”, con una banda muerta de 2 unidades.

Un control On-Off también puede ser implementado a través de dos estados (uno de ellos activa la salida), cuyas transiciones realicen las comparaciones entre el setpoint y la entrada.

El estado “Estado_2” tiene una maniobra con un control proporcional que tiene salida hacia la variable “<Local_A_Tarea>”, setpoint en la variable de FLASH “Setpoint_1”, medición en la variable “ENTRA_0” y una ganancia proporcional de 32 / 16 (2 unidades). Adicionalmente tiene otra maniobra de modulación de la acción de control “<Local_A_Tarea>”, hacia la salida “SALIDA_0” con rango de 0 a 1000 (1250 / 1000 Hz que corresponde a un período de 800 mS).

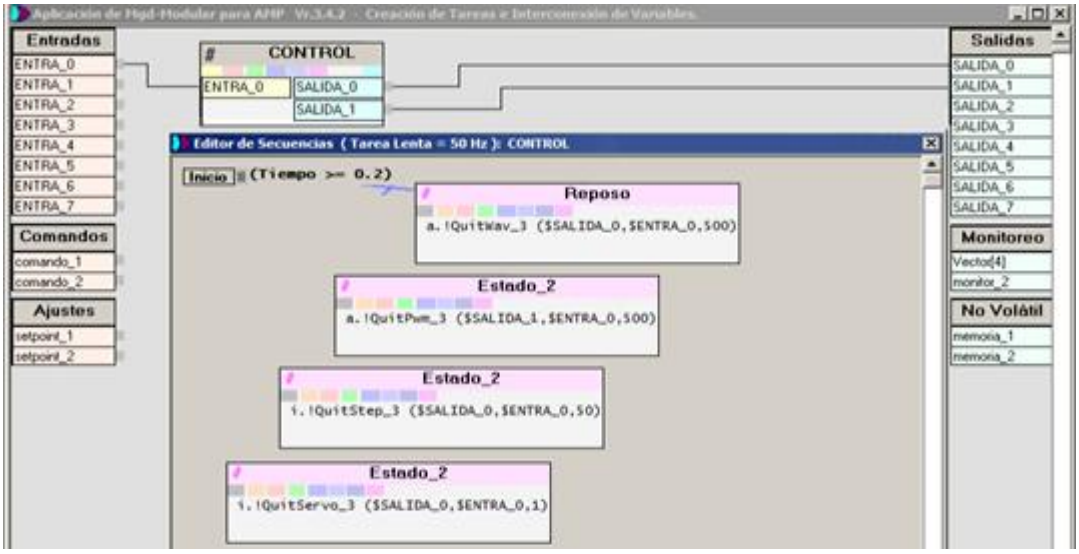
El estado “Estado_3” tiene una maniobra con un control proporcional como en el caso anterior, con la adición de una maniobra integral-derivativa, la cual entrega la acción de control PID en la variable “<Local_B_Tarea>” y utiliza un vector de RAM de cuatro posiciones “Vector[4]”, para la realización de cálculos internos. Tiene una constante integral de 128 / 256, una constante derivativa de 16 / 16 y un tiempo integral de 5 ciclos de la tarea. Así mismo, la acción de control “<Local_B_Tarea>” es modulada hacia la salida “SALIDA_0” en una tercera maniobra del estado.

Las variables de setpoint ubicadas en memoria FLASH pueden ser ajustadas en campo, utilizando una interfaz Hmi Lcd 2x16 o una interfaz Hmi tipo OP conectada via Rs485 con protocolo modbus.

UTILIZACIÓN DE VARIABLES CONTINUAS EN FUNCIONES DE SALIDA

Para complementar las funciones de control, se dispone de funciones de salida, las cuales envían órdenes al sistema operativo para generar deferentes tipos de modulaciones en las salidas del autómata programable. Estas funciones sólo son aplicables a Amp1 con salidas por semiconductor.

Las funciones permiten enviar salidas por tren de ondas (QuitWav), por Pwm (QuitPwm sólo por las salidas Q1 y Q5), hacia motores de paso (QuitStep) y hacia servomotores (QuitServo).



En la imagen anterior se observa un primer estado “Reposo” con una maniobra de configuración de la salida “SALIDA_0” para enviar trenes de onda con período de (500 / 1250) Segundos y con ancho de pulso determinado por “ENTRA_0” en un rango entre 0 y 500 (0 es apagado y 500 o mayor es encendido). Los trenes de onda son Pwm lentos que se envían a relevos de estado sólido con cruce por cero para el control de calor en resistencias eléctricas.

El segundo estado tiene una maniobra de configuración de la salida “SALIDA_0” para enviar modulación Pwm con período de (500 / 1250000) Segundos y con ancho de pulso determinado por “ENTRA_0” en un rango entre 0 y 500 (0 es apagado y 500 o mayor es encendido). Para los autómatas Amp1-E, la modulación por ancho de pulso sólo opera por las salidas “Q1” y “Q5”.

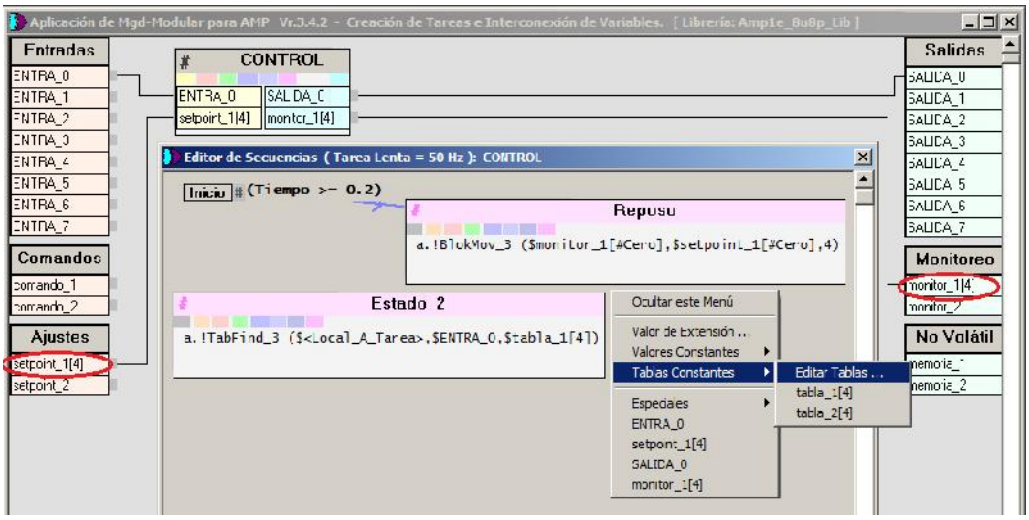
El tercer estado tiene una maniobra con una orden (al iniciar el estado) para que se envíen pulsos a un motor de paso con driver conectado a la salida “SALIDA_0” (“Q0”). La cantidad de pulsos a enviar está determinada por la variable “ENTRA_0”, a una frecuencia de (6250 / 50) pulsos por segundo. Consultar el documento de salidas de los autómatas programables para ver más detalles.

El cuarto estado tiene una maniobra con una orden (al iniciar el estado) para que se envíe modulación Pwm a un servo motor conectado a la salida “SALIDA_0” (“Q0”). La posición a alcanzar por el servo motor, está determinada por la variable “ENTRA_0”, en un rango entre 0 (ángulo 0) y 1000 (mayor ángulo del servo motor), con polaridad “1” (positiva), a una frecuencia de 50 Hz con ancho de pulso entre 1 y 2 mS. Consultar el documento de salidas de los autómatas programables.

ARREGLOS DE VARIABLES CONTINUAS

Las variables continuas del tipo “Comandos”, “Ajustes”, “Monitoreo” y “No Volátil” pueden ser presentadas como vectores (conjuntos de variables continuas que comparten el mismo nombre y pueden ser accedidas para lectura y escritura a través de un índice). Un vector de variables continuas tiene la representación “Variable[indice]” para ser leída o escrita, y la representación “Variable[longitud]” para mostrar cuantos ítems la constituyen.

Para crear un vector se pulsa click derecho sobre una de las variables antes mencionadas y se selecciona la opción “Presentar como arreglo”, a lo cual aparece un diálogo preguntando la longitud de arreglo, luego de lo cual queda establecido el arreglo. Los valores iniciales de los arreglos tipo “Ajustes” pueden ser editados pulsando click derecho sobre el vector y click sobre la opción “Editar Tabla arreglo”. Los ítems de un vector se numeran de cero en adelante.



La imagen anterior muestra la conexión de vectores de “Ajustes” y de “Monitoreo” a una tarea y su utilización en diferentes funciones. También muestra las tablas que son vectores de constantes.

La función “BlokMov” del primer estado “Reposo” permite mover 4 (cuatro) ítems del vector “setpoint_1”, empezando desde el ítem cero, hacia el vector “monitor_1”, a partir de su ítem cero. Esta función es útil en operaciones de carga de conjuntos de setpoints y configuraciones en aplicaciones con selección de programa.

La función “TabFind” del estado “Estado_2” busca en la tabla “Tabla_1”, con 4 (cuatro) ítems, el índice del ítem más cercano por exceso al valor de la variable “ENTRA_0”. El resultado de la función es puesto en la variable “<Local_A_Tarea>” para ser utilizado como índice para la consulta de otras tablas o vectores. La función es útil en aplicaciones que utilizan selectores con resistencias entrando por una sola entrada, para determinar la posición del selector.

La combinación de las dos funciones permite crear aplicaciones con selección de recetas de operación, las cuales son recibidas por el bus de comunicaciones hacia las variables “Ajustes” desde SCADA o desde Hmi tipo OP, vía Modbus RTU.