

3 Ciclo de funcionamiento del Autómata

Se puede considerar a los autómatas programables como máquinas secuenciales que ejecutan de forma correlativa las instrucciones indicadas en el programa de usuario que se encuentra almacenado en su memoria, generando unas señales u ordenes de mando a partir de las señales de entrada leídas de la planta.

Una vez detectados los cambios en las señales, el autómata reacciona, dependiendo del programa, hasta obtener las órdenes de salida necesarias.

Esta secuencia se ejecuta de forma continua para poder conseguir el control actualizado del proceso.

La secuencia de operación del autómata puede dividirse en tres fases principales. Estas son:

- Lectura de señales desde la interfaz de entradas.
- Escritura de señales en la interfaz de salidas.
- Procesado del programa para obtener señales de control.

Las dos primeras, lectura y escritura, suelen designarse también como volcado de las señales de interfaz de entrada o salida a la memoria imagen.

Para la optimización de los tiempos de acceso a las interfaces E/S, debe realizarse la lectura y la escritura de señales, de manera simultánea para todas las entradas y salidas implicadas, con el objetivo de que las entradas leídas se almacenen en una memoria temporal o memoria imagen de entradas, a la que acude la CPU mientras dura la ejecución del programa, consiguiendo que los resultados o señales se vayan almacenando en otra memoria temporal o memoria de imagen de salidas, según se vayan obteniendo.

Una vez concluida la ejecución, los resultados se colocan en la interfaz de salida de una sola vez.

Para garantizar la seguridad en el funcionamiento, el autómata programable realiza una serie de acciones comunes como son los chequeos de memoria de la CPU, comprobación del reloj de guarda o "**watchdog**", etc.

También, y de forma esporádica, el autómata establece comunicación con periféricos exteriores, por ejemplo para volcado de datos en impresora, comunicación con otros ordenadores o autómatas o la facilitar la conexión con la unidad de programación, entre otras acciones.

Toda esta serie de acciones se van repitiendo de forma periódica, definiendo así un ciclo de operación que necesita de un cierto tiempo para poder ser ejecutado. Este tiempo será determinante cuando el autómata pretenda controlar procesos rápidos, con señales de duración muy corta o una alta frecuencia de conmutación.

Nos disponemos a ver en este punto el ciclo de operación del autómata y las distintas comprobaciones que se realizan durante su funcionamiento, con una valoración cualitativa de los tiempos que emplea en realizar cada operación.

También veremos el concepto de respuesta del autómata en tiempo real y las posibles soluciones que ofrecen los fabricantes para controlar los procesos rápidos.

1. Modos de operación

Un autómata que se encuentre bajo tensión puede mantenerse en uno de los estados de funcionamiento (modo de operación) que citamos a continuación:

- **RUN:** el autómata ejecuta normalmente el programa de usuario contenido en su memoria. Las salidas evolucionan tomando distintos estados en función del estado de las entradas y las órdenes del programa. En este estado los temporizadores y contadores programados operan normalmente.
- **STOP:** la ejecución del programa se detiene por orden del usuario. En ese momento las salidas pasan a estado OFF, las posiciones internas (relés, registros, contadores y temporizadores) mantienen su estado en memoria interna. Cuando se pasa otra vez a RUN, todas las posiciones internas, excepto las mantenidas, o protegidas contra pérdidas de tensión, pasan a estado inicial.

El modo STOP es normalmente utilizado para servicios de mantenimiento o diagnóstico, al congelar el funcionamiento del autómata sin pérdida de la información contenida en su interior, que puede, en ese momento, ser visualizada desde la unidad de programación.

- **ERROR:** en este caso, el autómata detiene la ejecución debido a un error de funcionamiento, desactiva todas las salidas. El autómata queda bloqueado hasta que el error es subsanado. El autómata puede salir de este modo bien por "reset" de puesta en tensión, bien por orden desde la CPU o bien con comando enviado desde la unidad de programación.

En cuanto al modo de operación del autómata hay que considerar que dicho modo de operación o estado de funcionamiento puede ser controlado por medio de conmutadores situados en la misma CPU, o bien, desde la unidad de programación, con el envío de los comandos adecuados.

Estas dos posibilidades pueden encontrarse simultáneamente en autómatas incluidos en las gamas medias y altas, mientras que los modelos compactos de gama baja suelen incluir, únicamente, la segunda.

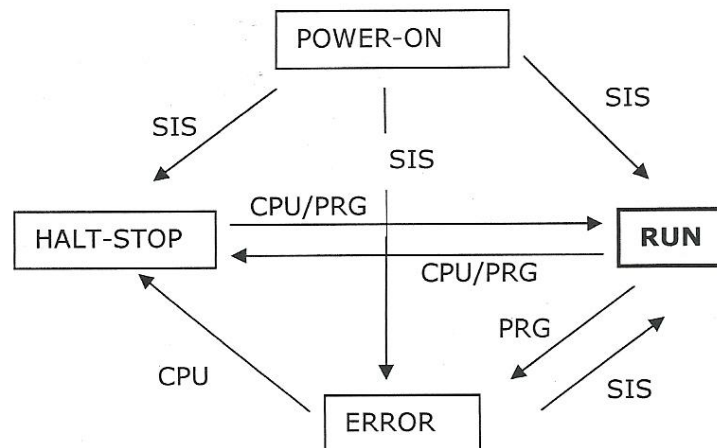
Algunos modelos de autómatas compactos no poseen conmutadores para el cambio de modo, aunque pueden sustituirlos por interruptores externos que se leen por medio de puntos de la interfaz de entradas, asignados para esta función mediante los parámetros de configuración.

Una vez realizada la puesta en tensión, el autómata pasa a los modos RUN o HALT/STOP, dependiendo del modelo y de la configuración del mismo. De cualquier forma, las conmutaciones RUN/STOP pueden ser forzadas por el usuario desde la unidad de programación o desde los interruptores previstos a tal efecto sobre la CPU.

Si cuando en el intento de puesta en marcha o paso a estado RUN, el autómata detecta algún mal funcionamiento sobre el aparato (alimentación, conexiones, etc) o sobre el programa (sintaxis), no se mantiene el estado RUN y la CPU cae en ERROR.

La siguiente figura muestra la relación entre los distintos modos, indicando los cambios entre ellos con los siguientes códigos:

- **SIS:** cambio de modo ordenado mediante el sistema monitor.
- **CPU:** cambio de modo ordenado por usuario desde conmutadores de la CPU.
- **PRG:** cambio de modo ordenado por usuario por comandos desde la unidad de programación.



El autómata puede disponer de una función de "reset" que, una vez activada desde la unidad de programación, borra todas las posiciones internas, incluso las mantenidas, dejando al autómata totalmente reinicializado para que pueda ejecutar el programa.

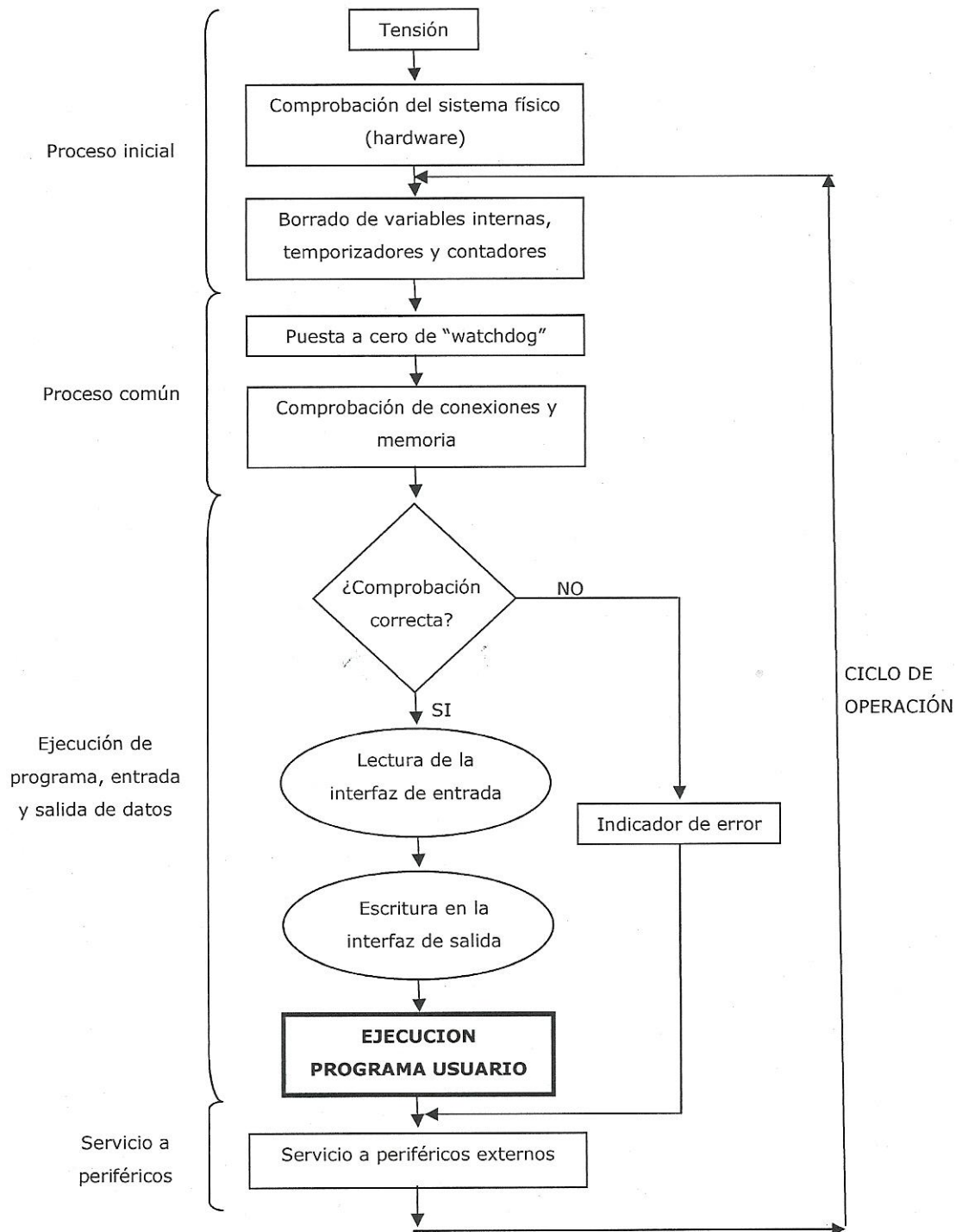
Habitualmente, y puesto que en el modo RUN no se puede modificar el programa de usuario (únicamente es posible en autómatas con programación ON-LINE), el aparato debe estar en modo HALT/STOP cuando se desea introducir o modificar un programa de aplicación.

La confirmación o comprobación del modo de funcionamiento se puede realizar de manera sencilla sobre unos LED incorporados en la CPU, por observación de su estado apagado, encendido o parpadeante, posibilitando así las labores de mantenimiento y puesta a punto en caso de un posible error.

2. Ciclo de funcionamiento

Con la salvedad del proceso inicial que sigue a un "reset", el funcionamiento del autómata es de tipo **secuencial y cíclico**, o lo que es lo mismo, las operaciones tienen lugar una tras otra, y se van repitiendo de forma continua mientras el autómata se encuentre bajo tensión.

La figura muestra de forma esquemática la secuencia de operaciones que el autómata ejecuta. Sobre esta figura se define el denominado ciclo de operación con aquellas que se repiten interminablemente.



Como se indica en la figura, previamente a la entrada en el ciclo de operación el autómata realiza una serie de acciones comunes que tratan de inicializar los estados del mismo y de chequear el hardware.

Estas rutinas de chequeo, incluidas en el monitor ROM, analizan y comprueban:

- El bus de conexión de las unidades de E / S.
- El nivel de la batería, si es que existe.
- La conexión de las memorias internas del sistema.
- El módulo de memoria exterior, si existe.

Si se encuentra algún error en chequeo, se activa el LED de error, quedando registrado el código del mismo. No obstante, el funcionamiento puede quedar interrumpido en ese punto, según sea la gravedad del error.

Una vez comprobadas las conexiones, se inicializan las variables internas:

- Se ponen en OFF las posiciones de memoria interna (excepto las mantenidas o protegidas contra pérdida de tensión).
- Se borran todas las posiciones de memoria imagen E / S.
- Son borrados todos los contadores y temporizadores (excepto los mantenidos o protegidos contra pérdidas de tensión).

Una vez transcurrida la secuencia de inicialización y siempre que no hayan aparecido errores, el autómata entra en el ciclo de operación que está constituido por aquellas operaciones que se ejecutan continuamente. Este ciclo puede considerarse dividido en tres bloques, como muestra la figura. Es decir:

- Bloque 1. Proceso común.
- Bloque 2. Ejecución del programa de usuario.
- Bloque 3. Servicio a periféricos.

En el primer bloque (proceso común), se comprueba el reloj de guarda y realizan los chequeos cíclicos de conexiones y de memoria de programa, protegiendo al sistema contra los siguientes tipos de errores:

- Errores de hardware (conexiones I/O, ausencia de memoria de programa, etc.).
- Errores de sintaxis (programa imposible de ejecutar).

El "watchdog" o reloj de guarda es un temporizador interno no accesible por el usuario que fija el tiempo máximo de ejecución de un ciclo de operaciones. Si

este temporizador alcanza el valor prefijado que puede ser de entre 0,1 y 0,5 s. según modelos, el autómata pasa al estado STOP y se ilumina el indicador de error.

Las posibles causas para la activación del reloj de guarda pueden ser:

- Existencia de algún error de sintaxis en el programa, de manera que nunca se alcanza la instrucción END.
- Bloqueo de la comunicación con periféricos externos.
- Avería en el funcionamiento de la CPU, etc.

A continuación citamos algunos puntos que se comprueba en el chequeo cíclico de conexiones:

- Niveles de las tensiones de alimentación.
- Buses de conexión con las interfaces E/S.
- Estado de la batería, si es que existe.

En cuanto al chequeo de la memoria de programa se comprueban la integridad de la misma y los posibles errores de sintaxis y gramática:

- Mantenimiento de los datos, comprobados con el "checksum".
- Existencia de la instrucción END de fin de programa.
- Estructuras de salto y anidamiento de bloques correctas.
- Códigos de instrucciones correctas, etc.

Normalmente este proceso común no supera 1 ó 2 milisegundos de tiempo.

Con respecto al segundo bloque de operaciones (ejecución del programa de usuario), se consultan y actualizan los estados de las entradas y las salidas, elaborándose las órdenes de mando a partir de ellos, por ejecución secuencial de las instrucciones del programa.

El tiempo de ejecución de este bloque de operaciones es la suma del tiempo de acceso a interfaces de E/S y el tiempo de escrutación del programa.

Estos tiempos dependen a su vez del número y ubicación de las interfaces E/S y de la longitud del programa y tipo de CPU que lo procesa, respectivamente.

El tiempo de acceso a las interfaces depende de sí éstas se encuentran cableadas como locales o como remotas, es decir; conectadas a la CPU por el bus

interno, o por medio de procesadores de comunicaciones, o del número de entradas / salidas instaladas.

Cuando hablamos de un autómata compacto o modular con interfaces locales, este tiempo es del orden de 5 ms., aunque puede variar significativamente de uno a otro modelo.

El tiempo de ejecución del programa depende del tipo de instrucciones que se procesan, siendo mayor en caso de instrucciones aritméticas o de manipulación de datos, y del modelo de CPU que ejecuta, reduciéndose en el caso de que el autómata utilice procesadores a medida o personalizados ("custom") en vez de microprocesadores estándar.

La tabla muestra los tiempos de ejecución de instrucciones (en microsegundos), de la gama de autómatas OMRON. Los tiempos más bajo corresponden a la ejecución de las instrucciones básicas (AND, OR, etc.), y lo más largos a los de manipulación de datos, contadores y temporizadores.

FAMILIA DE AUTÓMATAS OMRON Y TIEMPOS DE EJECUCIÓN			
GAMA DE AUTÓMATAS	AUTÓMATA	NÚMERO DE PUNTOS E/S	TIEMPOS DE EJECUCIÓN (µs/instr.)
ALTA	C 2000 H C 1000 H C 200 H	2048 1680	0,4 a 2,4 0,75 a 2,25
MEDIA	C 500 C 120	512 256	3 a 83 3 a 83
BAJA	Serie K Serie P C 20	100 a 148 36 a 120 84	4 a 95 4 a 95 4 a 80

El tercer bloque de operaciones dentro de un ciclo de ejecución, el servicio a periféricos externos, solamente es atendido si queda pendiente o en suspenso algún intercambio con el exterior.

La comunicación entre los periféricos y el autómata se realiza, bien directamente por medio de un conector situado sobre la CPU, o bien a través de procesadores de comunicación descargando a la unidad de control de esta tarea.

Cuando la comunicación se realiza a través de un conector incorporado queda reservada casi totalmente a la unidad de programación, aunque existen

productos de mercado (terminales de operador y visualizadores, por ejemplo) directamente conectables a esta línea.

Para realizar la conexión con otros periféricos se ejecuta, habitualmente a través de procesadores de comunicación que, a su vez, permiten el enlace con ordenadores, impresoras, visualizadores de plasma o siete segmentos, lectores de código de barra, otros autómatas, etc.

Después de quedar establecida la comunicación, la CPU dedica un tiempo limitado entre 1 y 2 ms., en reconocer al intercambio de datos. Si este tiempo no fuera suficiente, el servicio sufre una interrupción hasta el siguiente ciclo.

Esta interrupción del servicio puede llegar a ser apreciable en procesos de monitorización de estados, una vez que la unidad de programación u otro periférico determinado pretende visualizar la evolución de las variables en el autómata, ofreciendo como resultado la aparición de un retardo entre el estado de los LED indicadores de señal en las interfaces de E/S y la información visualizada sobre la unidad externa. Para el resto de servicios, conexión con impresora, con monitores o visualizadores industriales, etc., los retardos son inapreciables.

3. Chequeos del sistema

Como se ha visto en el punto anterior, el autómata incorpora rutinas de autochequeo, en su monitor ROM que le permiten diagnosticar el estado del programa y del hardware conectado.

Estas rutinas pueden ser:

- **Iniciales**, que corren, únicamente, tras la puesta o reanudación de la tensión.
- **Cíclicas**, que se repiten de forma continua mientras el autómata está ejecutando el programa.

Las primeras se encargan de comprobar las conexiones físicas del sistema y la identificación de la configuración existente.

Las rutinas cíclicas se encargan de comprobar la integridad del programa y de las conexiones E/S utilizadas en el mismo:

Si una vez realizados los chequeos o comprobaciones pertinentes se detecta algún problema, el autómata acusa el error, memoriza el código identificador correspondiente y de forma eventual, señala la situación iluminando el LED de error y algún otro identificador si lo tiene incorporado.

Realizando el estudio de los códigos de error y/o combinando el estado de los diferentes pilotos, es posible analizar los defectos y corregirlos con celeridad.

La identificación completa del error se realiza visualizando sobre la unidad de programación el código del mismo (lectura del registro que almacena el tipo de error) y utilizando una tabla cruzada de códigos y nomenclatura de errores.

En la documentación del autómata se suele incluir un capítulo de diagnóstico y corrección de errores que permite, en primer lugar, la identificación del mismo y proporciona, además, posibles soluciones para su corrección.

Los errores se pueden clasificar en dos categorías: fatales y no fatales.

Los fallos o errores fatales provocan la detención del autómata (STOP), y exigen la reinicialización de la máquina por reset o puesta en tensión.

En cuanto a los errores no fatales quedan recogidos y señalizados en el autómata pero no provocan la parada de la máquina.

Es posible encontrar en algunos autómatas un "histórico" de errores, es decir un conjunto de registros internos que van almacenando los errores producidos con sus códigos de identificación correspondientes e, incluso los datos de fecha y hora en los que se producen, si es que la CPU tiene incorporado un reloj de tiempo real.

Los contenidos de esta tabla de registro se van actualizando conforme aparecen nuevos errores, siempre que éstos no provoquen la parada del autómata.

A continuación se van a enumerar los errores que son detectados de forma más habitual:

- Fallos de alimentación.
- Desbordamiento del reloj de guarda.
- Ausencia del módulo de memoria en autómatas sin RAM de programa.
- Destrucción y pérdida del programa o parte de él.
- Fallos de conexión con las interfaces de entrada/salida.
- Valores de preselección de temporizadores y contadores fuera de rango.

- Fallo de conexión con la unidad de programación o con otros autómatas que se encuentre dentro de una determinada red.
- Direccionamiento incorrecto de las interfaces E/S o de variables internas.
- Imposibilidad de carga del programa (demasiado largo, o incorrectamente identificado), etc.

Las unidades de programación pueden disponer, además de las rutinas de autochequeo incorporadas en el monitor del autómata, de unos comandos de test que tienen como misión la comprobación sistemática de todo el sistema. Es decir: entradas/salidas, estado de la CPU y de las memorias, conexiones del bus, conexiones con la unidad de programación, teclados y visualizadores de ésta, entre otras. La función de test, si existe, siempre es ordenada por el usuario, a diferencia de las rutinas del autochequeo anteriores, de activación automática.

En la actualidad todos los fabricantes disponen de software de programación para la simulación del funcionamiento del programa de usuario en el PC. De esta manera se pueden depurar los posibles errores de programa antes de volcarlos al autómata.

4. Tiempo de ejecución y control en tiempo real

El tiempo total que el autómata emplea para realizar un ciclo de operación se denomina tiempo de ejecución de ciclo o **tiempo de ciclo** ("Scan Time"). Este tiempo depende directamente de:

- El número de entradas/salidas involucradas.
- La longitud del programa de usuario.
- El número y tipo de periféricos conectados al autómata.

El tiempo total del ciclo es, por tanto, la suma de los tiempos empleados en realizar las distintas operaciones del ciclo, es decir:

- Actualización de E/S.
- Autodiagnósticos.
- Ejecución y/o escrutación del programa.
- Servicios a periféricos.

Los tiempos empleados en chequeos y comprobaciones cíclicas (autodiagnóstico) son del orden de 1 y 2 ms., al igual que los consumidos en la atención a periféricos externos.

Los tiempos de ejecución de instrucciones se miden en unidades de microsegundos (μ s), dando por resultado un tiempo de escrutación del programa variable en función del número y tipo de instrucciones contenidas.

Hay que tener presente que el tiempo de escrutación es uno de los parámetros característico del autómata y que se expresa, habitualmente en microsegundos por cada mil instrucciones (ms/K).

Es lógico, por tanto, deducir que el tiempo resultante dependerá del tipo de instrucciones elegidas, por lo que el fabricante suele indicar las utilizadas como base de cálculo (aritméticas, booleanas, de temporización, etc.) y las proporciones entre ellas dentro del programa muestra.

Los tiempos típicos oscilan entre los 5 y los 15 ms/K instrucciones, aunque la aplicación de nuevas tecnologías y diseños a la CPU están reduciendo, considerablemente los tiempos a valores menores de 1 ms/K instrucciones.

En referencia a los tiempos de acceso al exterior para lectura/escritura de datos, éstos varían entre 1 y 5 ms, dependiendo del número de E/S involucradas, de la CPU del autómata, de si éste es compacto o modular, si se utilizan o no entradas/salidas remotas, etc.

En un autómata estándar de gama media, para una aplicación que necesite aproximadamente unas 1000 instrucciones, el tiempo de ciclo total puede, por lo tanto, estimarse del orden de 20 milisegundos.

El tiempo de ciclo es, sin duda, uno de los factores determinantes en el diseño del sistema de control cuando en éste aparezcan unas constantes de tiempo muy bajas (sistemas de respuesta rápida), pero no siendo el único a considerar, ya que también la electrónica de las interfaces incorpora retardos adicionales.

La lectura/escritura de señales de planta, realizada a través de las interfaces de entrada/salida, se encuentra afectada por los retardos intrínsecos de estos circuitos que, al incluir filtros que tratan de evitar la entrada de ruido eléctrico en el autómata, introducen también retardos en las conmutaciones de la señal, limitando la frecuencia máxima de éstas a valores inferiores a 100 Hz.

La tabla presenta los rangos de retardo máximo típicos, en milisegundos, para distintas interfaces de entrada y salida en las conmutaciones de encendido a apagado y viceversa.

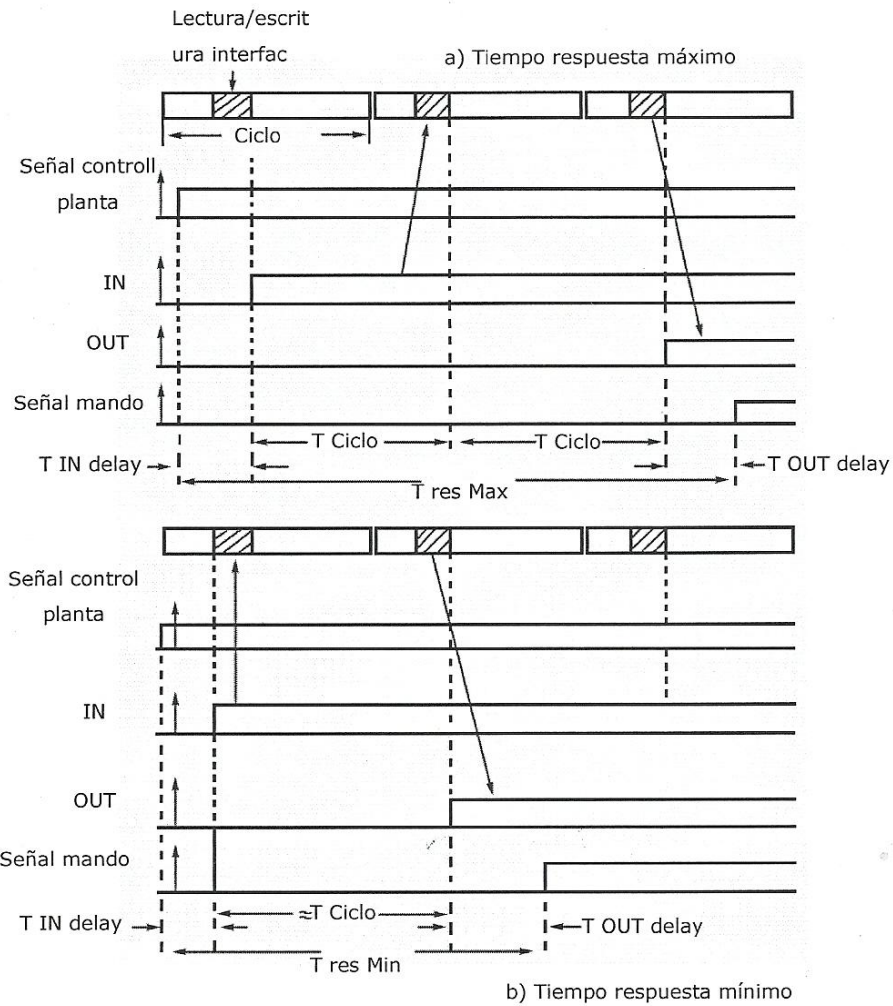
RETARDOS DE CONMUTACIÓN TÍPICOS DE INTERFACES DE AUTÓMATAS (ms)					
INTERFAZ CONMUTACIÓN	ENTRADA ($T_{INP\ delay}$)		SALIDA ($T_{OUT\ delay}$)		
	AC	DC	RELÉ	TRANSISTOR	TRIAC
OFF - ON	15 - 30	5 - 10	5 - 15	0,5 - 2	1 - 10
ON - OFF	15 - 30	5 - 10	5 - 20	1 - 2	11 - 12

Como se observa, las entradas en DC son más rápidas que en AC, por la mayor necesidad de filtrado de las tensiones de red, mientras que las salidas estáticas son, además, más rápidas que las salidas por relé. Se puede comprobar el largo tiempo de conmutación ON-OFF de la salida por triac, por el retardo que introduce el apagado natural (a corriente cero) del dispositivo.

Si el proceso utiliza además señales analógicas, es preciso, entonces, considerar también los tiempos de conversión A/D y D/A, lo que, debido al procedimiento de conversión por doble rampa normalmente utilizado, obliga a considerar retardos aún mayores, de 15 ms.

Dada una señal de mando (salida) función de una o varias señales de planta (entradas), se conoce como **tiempo de respuesta** al que transcurre desde que un cambio en una de aquellas señales de planta es acusado por la señal de mando. Este tiempo depende directamente de los retardos de conmutación y adaptación de señal en las interfaces de entrada y salida, $T_{INPUN\ delay}$ y $T_{OUTPUT\ delay}$ y del tiempo de ciclo del autómata.

Aunque el tiempo de respuesta es variable en función del instante en que cambia la entrada respecto al ciclo de operación, se mueve siempre entre dos valores límite máximo y mínimo, según se puede apreciar en el esquema.



Estos valores límites son:

- **Valor mínimo** (la señal se encuentra disponible en la interfaz justo antes de la lectura de entradas).

$$T_{RES min} = T_{INPUT delay} + T_{CICLO} + T_{OUTPUT delay}$$

- **Valor máximo** (la señal está disponible en la interfaz inmediatamente después de la lectura de entradas).

$$T_{RES max} = T_{INPUT delay} + 2 T_{CICLO} + T_{OUTPUT delay}$$

El tiempo de retardo en las interfaces de entradas y salidas depende del tipo de conmutación, ON/OFF o OFF/ON, así que para el cálculo del tiempo de respuesta máximo se tendrá que tomar los mayores posibles que indique el fabricante.

Hay que tener presente que los valores de tiempo de respuesta resultante, y de su inversa, la frecuencia de operación, pueden resultar despreciables frente a las constantes de tiempo de los sistemas electromecánicos, es decir: relés, solenoides,

electroválvulas, etc, a los que el autómata sustituía o controlaba en sus primeras aplicaciones de mercado, pero no lo son tanto comparados con los tiempos excesivamente cortos que intervienen en la lectura y control de los elementos electrónicos que son típicos en los procesos de posicionamiento de motores (lecturas de pulsos de encoders), detección de móviles de alta velocidad, respuesta a alarmas, etc.

Cuando se trata de controlar uno de estos procesos, relativamente rápidos, puede darse la circunstancia de que la respuesta del autómata sea demasiado lenta, con lo que el control no actuará como se espera.

El autómata tiene la facultad de controlar en tiempo real a un determinado proceso, siempre que sus tiempos de respuesta o retardo resulten muy pequeños frente a los tiempos de reacción del mismo.

Un ejemplo clarificador es que, utilizando un autómata con interfaz de entrada DC y salida por relé, con un tiempo de ciclo de ejecución de 10 ms, y despreciando los tiempos de autochequeo y actualización E/S, se obtiene un tiempo de respuesta máximo de:

$$- T_{RES \text{ máx}} \approx 10 \text{ ms} + 2 \times 10 \text{ ms} + 10 \text{ ms} = 40 \text{ ms}$$

A este tiempo le corresponde una frecuencia máxima de conmutación que el autómata pueda leer de:

$$- f = 1/T_{RES \text{ máx}} = 1/40 \text{ ms} = 25 \text{ Hz}$$

Si el proceso a controlar precisa de un tiempo de respuesta en alguna variable inferior a 40 ms o si generan o manipulan frecuencias superiores a 25 Hz, el autómata, visto en el ejemplo anterior, no podrá tener el control de dicho proceso sin riesgo de provocar oscilaciones, pudiendo saturar y bloquear al mismo.

Con el objetivo de mantener la potencialidad de los autómatas, incluso en las aplicaciones que exigen una mayor rapidez, los fabricantes incluyen en sus distintos modelos, elementos o funciones de proceso rápido, los cuales se verán en el siguiente apartado.

5. Elementos de proceso rápido

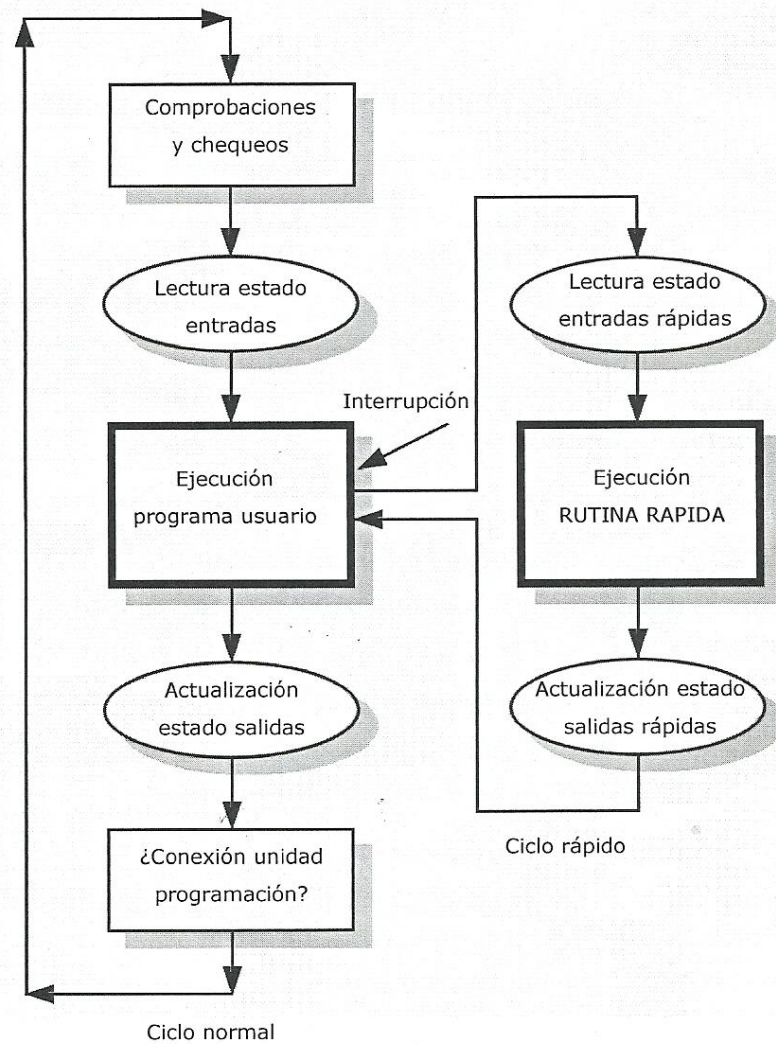
Existen tres tipos de actividades o circunstancias que exigen rápidas respuestas del autómata y que obligan a buscar distintas soluciones en el autómata:

- Ejecución de subrutinas o programas a intervalos menores de los que permite el tiempo de ciclo general del autómata como procesado de alarmas, por ejemplo. Para conseguir esta reducción del tiempo de respuesta para una señal determinada, puede ordenarse la ejecución del bloque de programa que la controla de forma independiente al resto del ciclo, pudiéndose iniciar ésta por los siguientes métodos:

Mediante un ciclo periódico, prefijado por el usuario (temporización interna) por medio del cual se activa una rutina rápida que asegura la reducción del tiempo de ciclo para las señales E/S manipuladas dentro de la rutina.

Mediante una señal de interrupción exterior (asíncrona) para conseguir respuestas rápidas, lo que hace que el autómata responda con rapidez frente a estímulos externos.

El programa principal continua en el punto en que fue interrumpido tras la ejecución del bloque de tratamiento rápido. Esto significa que, si bien se reducen los tiempos de respuesta para las entradas/salidas procesadas en la rutina, se incrementan los correspondientes al resto de señales, en un valor igual al del tiempo de ciclo de la rutina rápida por el número de interrupciones al programa normal.



Algunos autómatas, sobre todo los de gama media y alta, permiten la programación de varios bloques de proceso específico, cada uno de ellos con su propia retícula de tiempo: una vez alcanzado uno de estos tiempos, se activa una interrupción y si la tarea que se encontraba en ejecución era de un período mayor o menor prioridad, se interrumpe para ejecutar la tarea asociada a la interrupción activada, transcurrida la cual, el proceso se reanuda en el punto donde sufrió la interrupción.

Esta multitarea o estructura de programación llamada **programación estructurada** permite definir programas diferentes, para ejecutar sobre el procesador principal o sobre los auxiliares. Estos programas son:

- Programa básico.
- Programa rápido.

- Programa lento.
- Rutinas de emergencia, entre otros.

La totalidad de estos programas pueden intercambiar datos y acceder a las mismas entradas, salidas y variables internas, resultando una mayor simplicidad y facilidad de programación y supervisión del funcionamiento.

Una estructura como la descrita permite además la priorización total de las tareas de control, lo que implica que en la práctica el usuario puede elegir los tiempos de respuesta que desea para cada una de las señales de su proceso.

- Lectura de impulsos de entrada a alta frecuencia como por ejemplo los procedentes de escoders. La pérdida de información que supone la aparición de varios impulsos dentro de un mismo ciclo de ejecución, cuando es leído sólo el que coincida con el instante de lectura, puede evitarse si se dispone de un contador rápido que, aprovechando un hardware específico con entradas débilmente filtradas y circuito propio, sea capaz de leer señales de alta frecuencia.
- La detección de señales de muy corta duración, menor que el tiempo de ciclo como por ejemplo, para detección de elementos móviles a alta velocidad. Para este supuesto podría emplearse cualquiera de las dos soluciones anteriores, aunque algunos autómatas que no las incluyen, en particular las de gamas bajas, pueden tener en cambio una o varias entradas detectoras de flancos, capaces de captar y mantener en un relé interno, y hasta que produzca la lectura por parte de la CPU, señales individuales de tan corta duración como 1ms (menores de un "scan"), de manera que el autómata llega a tener constancia de ellas.

RECUERDA

- Un autómata que se encuentre bajo tensión puede mantenerse en uno de los estados de funcionamiento siguientes: RUN, STOP y ERROR.
- Con la salvedad del proceso inicial que sigue a un "reset", el funcionamiento del autómata es de tipo secuencial y cíclico, o lo que es lo mismo, las operaciones tienen lugar una tras otra, y se van repitiendo de forma continua mientras el autómata se encuentre bajo tensión.
- Las rutinas de chequeo analizan y comprueban:
 - El bus de conexión de las unidades E/S.
 - El nivel de la batería.
 - La conexión de las memorias internas del sistema.
 - El módulo de memoria exterior.
- El autómata incorpora rutinas de autochequeo, en su monitor ROM que le permiten diagnosticar el estado del programa y del hardware conectado. Estas rutinas pueden ser iniciales y cíclicas.
- El tiempo total que el autómata emplea para realizar un ciclo de operación se denomina tiempo de ejecución de ciclo o tiempo de ciclo ("Scan Time"). Este tiempo depende directamente de:
 - El número de entradas/salidas involucradas.
 - La longitud del programa de usuario.
 - El número y tipo de periféricos conectados al autómata.
- Existen tres tipos de actividades o circunstancias que exigen rápidas respuestas del autómata y que obligan a buscar distintas soluciones en el autómata:
 - Ejecución de subrutinas.
 - Lectura de impulsos de entrada a alta frecuencia.
 - Detección de señales de muy corta duración.