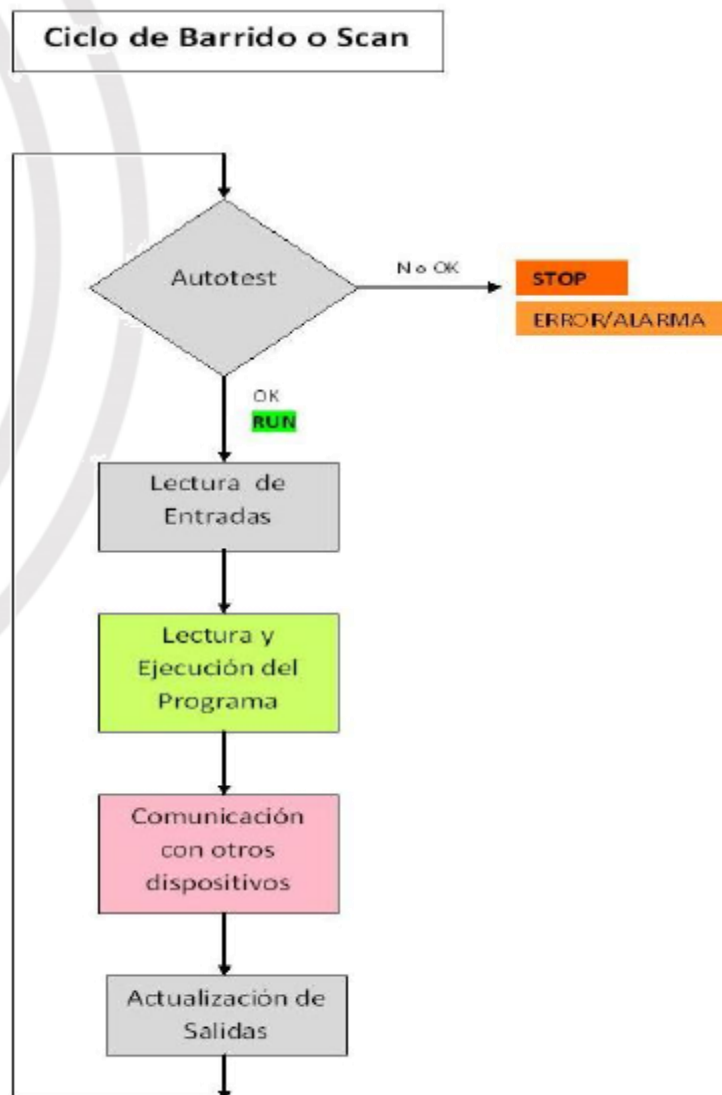


CICLO DE BARRIDO DEL PLC

Por el momento hemos dicho que al colocar al PLC en modo **RUN**, este comienza a ejecutar el programa cargado. Sin embargo, el equipo de PLC no se queda a esperar a que aparezca una señal de entrada correspondiente a un pulsador de marcha para empezar la secuencia de trabajo de una máquina. Esto no tendría sentido alguno ya que el equipo de PLC no tiene manera de reconocer qué elementos externos hay conectados a las entradas, así que no sabe a cuál señal de entrada le corresponde un pulsador de marcha, a cuál uno de parada, etc. Es por esto que el PLC al entrar en modo RUN realiza una serie de operaciones adicionales. Esta serie de operaciones conforman lo que se conoce como el **ciclo de barrido o ciclo de scan del PLC**.

El ciclo de barrido completo lo podemos ver en la siguiente imagen:



Pasemos a describir cada una de las etapas que lo componen:

PASO 1: AUTOTEST

En cuanto se da la orden de trabajar en el modo RUN, el PLC realizar una serie de **verificaciones** respecto a su **funcionamiento interno**. Nuevamente debemos mencionar que el PLC no reconoce los elementos externos conectados a las entradas y salidas así que no verifica si estos elementos funcionan correctamente. Sin embargo, internamente puede verificar cosas como:

- Que el programa tenga una sintaxis correcta. Puede suceder que el programa haya sido cargado con una sintaxis correcta pero que algún sobrevoltaje o corriente excesiva haya corrompido la memoria del programa y este haya quedado incomprensible.
- Que el sistema operativo reconozca todas las instrucciones. Puede suceder que el programa cargado requiera ejecutar alguna instrucción de temporizador, contador, etc, pero que por alguna razón esa instrucción no sea comprendida por el sistema operativo.
- Que el tiempo de ejecución del programa supere un tiempo máximo preestablecido. El PLC asume un tiempo máximo de ejecución del programa dependiendo de la capacidad máxima de programa que se pueda cargar y lo que demoran las diversas instrucciones en ejecutarse. Si el PLC comprueba que el programa cargado supera ese límite (por ejemplo, porque el programa entra en un bucle del cuál no puede salir), se dispara una señal de tiempo límite de ejecución denominada watchdog.

Si todas las verificaciones dan correctas, el PLC aprueba el modo RUN y continúa operando. Por el contrario, si alguna de las verificaciones falla el PLC detiene el ciclo de barrido, anula el modo RUN y se coloca en modo **STOP**. A su vez, informa el error encontrado mediante un código de error, como hemos visto en el apunte de los modos de funcionamiento del PLC.

PASO 2: LECTURA DE ENTRADAS

Una vez aceptado el RUN en el autotest, el PLC se encarga de leer **todas las entradas**. No solo toma la información de aquellas que tengan algún sensor o mando conectado, sino de todas las entradas, tengan algo conectado o no. Recolectada toda la información de las entradas, se dirige a la **memoria de datos de entradas** para escribir un valor 0 en aquellas entradas que no tienen señal (entradas apagadas) y un valor 1 en aquellas entradas con señal (entradas prendidas).

PASO 3: LECTURA Y EJECUCIÓN DEL PROGRAMA

Una vez leídas todas las entradas y guardada toda la información en la memoria de datos de entradas, el PLC se dispone a **leer y ejecutar el programa**. En este paso se encarga de leer línea por línea cargada en la memoria del programa y ejecutar las operaciones lógicas correspondientes para obtener los resultados. Si el PLC define en el programa que por ejemplo una salida debe prenderse, escribe un 1 en la **memoria de datos de salidas** correspondiente. Si por el contrario una salida dejó de recibir señal en las líneas de programación, escribe un 0 en la memoria de datos de salidas correspondiente.

PASO 4: COMUNICACIÓN CON OTROS DISPOSITIVOS

Una vez ejecutado todo el programa, el PLC pasa a verificar si existen **comunicaciones con otros dispositivos**. Estas pueden ser otros PLC en una red intercambiando datos, o una conexión uno a uno con una pantalla táctil, etc. En cualquier caso, si el PLC verifica por ejemplo que un dispositivo le pide el valor actuar del temporizador, se encarga de otorgarle ese dato. O si por ejemplo hay un dispositivo que le informa de un nuevo valor de seteo de un contador, el PLC se encarga de recibir ese dato y guardarlo en la memoria correspondiente. Cuando completa todas las comunicaciones o en caso de que verifique que no hay comunicaciones que atender, avanza al siguiente paso.

PASO 5: ACTUALIZACIÓN DE SALIDAS

Una vez atendidas todas las comunicaciones, el PLC continúa con la **actualización de las salidas**. En este paso el PLC usa los datos guardados en la memoria de datos de salidas y los vuelca a las salidas, actualizando **todas y cada una de ellas** sin importar si realmente se usaron en el programa o no. Evidentemente las salidas no utilizadas en el programa se quedarán abiertas ya que no hubo línea de programación donde se activen. Por otro lado, las salidas usadas en el programa se abrirán o se cerrarán dependiendo si la ejecución del programa dio como resultado abrir (señal en 0) o cerrar (señal en 1).

Una vez terminado el paso 5, el PLC **vuelve al paso 1 (autotest)** para verificar si puede proseguir con el modo RUN o debe pasar al modo STOP. En caso de continuar en RUN, repite nuevamente el ciclo de barrido.

TIEMPO DE CICLO O SCANTIME

Realizar todas las operaciones del ciclo de barrido le demora un tiempo al PLC denominado **tiempo de ciclo o scantime**. Este tiempo será la suma de los tiempos de los 5 pasos. El tiempo total de ciclo suele estar en el orden de los 4 milisegundos (dependiendo la potencia del procesador) pero no es el mismo por cada ciclo de barrido. Esto es porque algunos de los tiempos de los pasos son **fijos** pero otros son **variables**. Veamos cuáles son en cada caso.

- Tiempo 1 *autotest*, tiempo **fijo**. El autotest es una serie de operaciones de control interno preestablecidas y como tal, duran siempre el mismo tiempo.
- Tiempo 2 *lectura de entradas*, tiempo **fijo**. Hemos dicho que el PLC lee el estado de todas las entradas así que mientras no se agreguen módulos de expansión de entradas, el tiempo de lectura de todas ellas es el mismo.
- Tiempo 3 *lectura y ejecución del programa*, tiempo **variable**. La lectura y ejecución del programa dependerá de cuáles líneas de programación se activan y cuáles no. A modo de ejemplo, si un temporizador no está corriendo dado que no recibe señal, el programa no tiene que verificar cuándo es que ese temporizador cumple su tiempo (lo cual le demanda tiempo de operación al procesador, extendiendo el tiempo de ejecución del programa). Otro ejemplo pueden ser las instrucciones de salto de programa que cuando

se activan, evitan la lectura de un sector de programa, reduciendo su tiempo de ejecución.

- Tiempo 4 *comunicación con otros dispositivos*, tiempo **variable**. Como hemos dicho, puede que haya dispositivos solicitando intercambiar información con el PLC en ciertos momentos y en otras ocasiones no. Eso hará que el ciclo de barrido se demore más o menos en completarse según cuántas comunicaciones haya que atender.
- Tiempo 5 *actualización de salidas*, tiempo **fijo**. Hemos dicho que el PLC actualiza el estado de todas las salidas sin importar si se usan en el programa o no. Así que a menos que agreguemos módulos de salidas, este tiempo permanece fijo.

CONSIDERACIONES DE LAS ENTRADAS DE ALTA FRECUENCIA

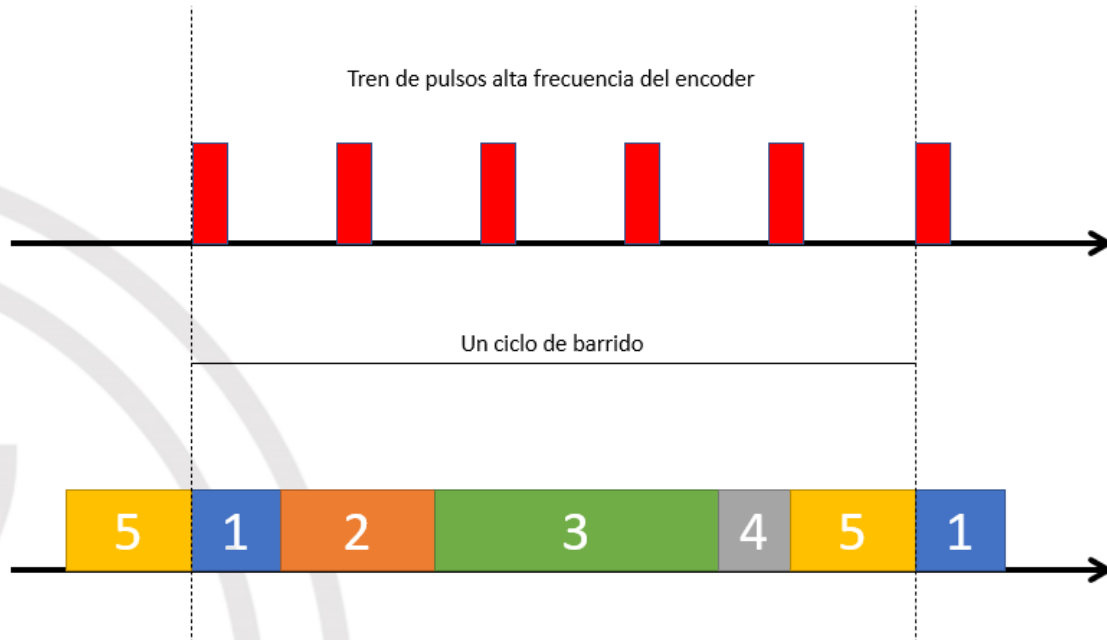
Para la mayoría de las aplicaciones que vemos en la industria, las variaciones del scan time no tienen incidencia alguna. Podemos despreocuparnos si dura más o menos tiempo por ciclo de barrido ya que los movimientos de los relés, electroválvulas y demás demoran mucho más que lo que le lleva al procesador del PLC ejecutar cada ciclo de barrido. Sin embargo, si una entrada tiene conectada una señal de **alta frecuencia**, puede que tengamos que prestar atención al tiempo de ciclo de barrido para no generar problemas.

Una entrada de alta frecuencia puede ser por ejemplo una ruedita que tiene mecanizadas unas rendijas por donde pasa la luz (dispositivo llamado encoder). Si esta ruedita gira a muy alta velocidad, habrá un tren de pulsos de señales de luz que pasan o no pasan. Si este mecanismo está conectado a la entrada de un PLC, este recibirá el tren de pulsos y deberá verificar cuándo prenden y cuándo apagan.

En la siguiente imagen tenemos representado un ciclo de barrido con los 5 pasos:



Si durante el tiempo que dura cada ciclo de barrido aparece el tren de pulsos de la entrada de alta frecuencia, el mismo puede entregar pulsos como se muestra en la siguiente figura:



El tren de pulsos del encoder entrega 5 pulsos de señal de entrada durante un ciclo de barrido del PLC. O sea que, si el ciclo de barrido del PLC es de 4 milisegundos, cada pulso de entrada aparece en un espacio de tiempo de 0,8 milisegundos

¿Qué problema puede surgir con esto?

Como se puede apreciar en la imagen, de los cinco pulsos otorgados por el encoder durante un ciclo de barrido, **solamente el segundo pulso** cayó justo en el momento en el que el PLC estaba cumpliendo con su tarea de **leer las entradas** (paso 2). Por ende, el PLC efectivamente pudo registrar ese pulso. Pero el resto de los pulsos, los otros cuatro que envió el encoder, no fueron efectivamente leídos, ya que el PLC estaba haciendo otros pasos del ciclo de barrido. Por lo tanto, de 5 pulsos del encoder, 4 se pierden y solo 1 se registra. Esto evidentemente va a hacer funcionar mal a la máquina conectada al PLC ya que no se estarían registrando todos los pulsos.

¿Cómo solucionar este problema?

Por un lado, se puede conseguir un PLC con un mayor poder de procesamiento que reduzca el tiempo del ciclo de barrido. De esta manera se puede asegurar que el PLC sea más rápido que los pulsos del encoder y siempre los encuentre. Esta solución puede ser costosa según la capacidad de procesamiento requerida.

Otra es adquirir un módulo de entradas de alta frecuencia y destinar la entrada del encoder solo a ese módulo. De esta manera, el módulo tendría su procesador encargado únicamente de recibir todos los pulsos y luego se comunicaría con el equipo principal de PLC para informar el valor de pulsos leído.

Para resumir, si el tiempo de ciclo de barrido es inferior a la duración de los pulsos de entrada, se puede asegurar que todos los pulsos de entradas van a ser leídos (gran mayoría de las aplicaciones). Por el contrario, si el tiempo de scan time es superior al tiempo de duración de las entradas (aplicaciones muy específicas como las que usan encoders) no se puede garantizar que todas las señales de entradas sean leídas.