

INTRODUCCIÓN TEÓRICA

Este documento pretende ser una introducción al control industrial y como evolución más actual de éste se introducirán los autómatas programables (PLC), que son elementos fundamentales en la cotidianidad de la automatización industrial.

Posteriormente se introducirá su programación y en concreto el lenguaje de contactos y un método de ayuda al diseño de programas para PLC como es el método GRAFCET.

1.1. INTRODUCCIÓN AL CONTROL INDUSTRIAL

El concepto de control es muy amplio, abarcando desde un simple interruptor que gobierna el encendido de una bombilla o el grifo que regula el paso de agua en una tubería, hasta el más complejo ordenador de proceso o el piloto automático de un avión. Podríamos definir el control como la manipulación indirecta de las magnitudes de un sistema denominado planta a través de otro sistema llamado sistema de control.

Los primeros sistemas de control se desarrollaron con la revolución industrial de finales del siglo XIX y principios del siglo XX. Al principio, se basaron casi exclusivamente en componentes mecánicos y electromecánicos, básicamente engranajes, palancas, relés y pequeños motores, pero a partir de los años cincuenta empezaron a emplearse los semiconductores, que permitían el diseño de sistemas de menor tamaño y consumo, más rápidos y con menor desgaste.

En la década de los setenta, la complejidad y las prestaciones de los sistemas de control se incrementaron gracias al empleo de circuitos integrados y en particular los de tipo programable (sistemas basados en microprocesadores).

Al tiempo que se desarrollaban los circuitos integrados lo hacían también los ordenadores digitales, si bien su empleo en la industria quedaba restringido al control de procesos muy complejos, debido a su elevado coste, necesidad de personal especializado para su instalación y manejo y a la poca facilidad de interconexión (interfaz) con el proceso, donde se manejan habitualmente tensiones y corrientes para las cuales no está preparado el ordenador.

La demanda en la industria de un sistema económico, robusto, flexible, fácilmente modificable y preparado para tratar con las altas tensiones y corrientes comunes en el ámbito industrial, hizo que se desarrollasen los autómatas programables industriales, abreviadamente API en la literatura castellana o PLC en la literatura anglosajona.

Los primeros autómatas pretendían, básicamente, sustituir a los sistemas convencionales con relés o circuitos lógicos, con las ventajas evidentes que suponía tener un hardware estándar. Por ello nacieron con prestaciones muy similares a las que ofrecían dichas tecnologías convencionales y sus lenguajes de programación eran muy próximos a los esquemáticos empleados en las mismas.

Estas limitaciones eran aconsejadas sólo por razones de mercado y no respondían a limitaciones tecnológicas de aquel momento, ya que las posibilidades que realmente podían ofrecer eran mucho mayores.

Los autómatas actuales han mejorado sus prestaciones respecto a los primeros en muchos aspectos, pero fundamentalmente a base de incorporar un juego de instrucciones más potente, mejorar la velocidad de respuesta y dotar al autómata de capacidad de comunicación. Los juegos de instrucciones incluyen actualmente, aparte de las operaciones lógicas con bits, temporizadores y contadores, otra serie de operaciones lógicas con palabras, operaciones aritméticas, tratamiento de señales

analógicas, funciones de comunicación y una serie de funciones de control no disponibles en la tecnología clásica de relés. Todo ello ha potenciado su aplicación masiva al control industrial.

En definitiva, podríamos decir que los grandes autómatas actuales se acercan cada vez más a las prestaciones de un pequeño ordenador, siendo algunos incluso programables en lenguajes típicamente informáticos como el BASIC.

Sin embargo, la principal virtud del autómata sigue siendo su robustez y facilidad de interconexión al proceso y la tendencia actual no es precisamente la de acercarlo más a las prestaciones de los ordenadores en cuanto a su capacidad de cálculo, sino dotarlo de funciones específicas de control y de canales de comunicación para que puedan conectarse entre sí y a los propios ordenadores. El resultado de esta integración es la red de autómatas conectada a ordenador, capaz de ofrecer las prestaciones y ventajas de ambos sistemas al integrar en un solo sistema todas las funciones de producción asistida por ordenador (CIM).

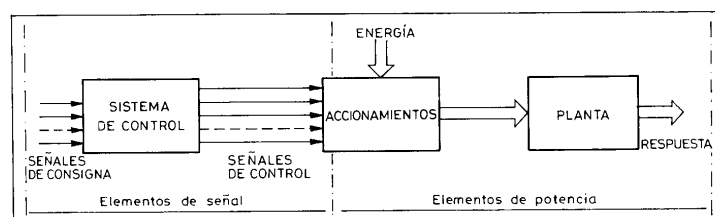
La disponibilidad de estos nuevos elementos y funciones en el campo del control industrial obliga a replantearse la configuración y los propios métodos de diseño de los automatismos.

1.1.1. SISTEMAS DE CONTROL

El objetivo de un sistema de control es el de gobernar la respuesta de una planta, sin que el operador intervenga directamente sobre sus salidas. Dicho operador manipula únicamente las magnitudes de consigna y el sistema de control se encarga de gobernar dichas salidas a través de los accionamientos.

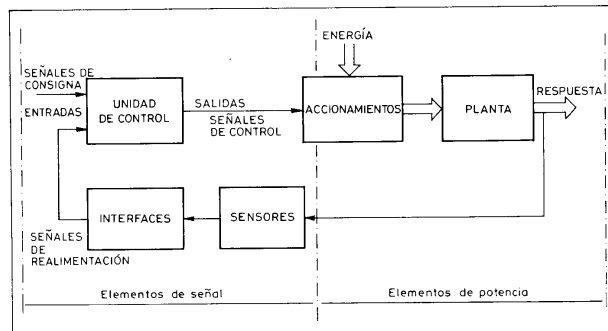
El sistema de control opera, en general, con magnitudes de baja potencia, señales, y gobierna unos accionamientos que son los que realmente modulan la potencia entregada a la planta según sea necesaria.

El conjunto de sistema de control y accionamientos se limitaría a ser un convertidor amplificador de potencia que ejecuta las órdenes dadas a través de las magnitudes de consigna. Este tipo de sistemas de control se denomina en lazo abierto, por el hecho de que no recibe ningún tipo de información del comportamiento de la planta.



Sistema de control en lazo abierto

Lo habitual, sin embargo, es que el sistema de control se encargue de la toma de ciertas decisiones ante determinados comportamientos de la planta, hablándose entonces de sistemas automáticos de control. Para ello se requiere la existencia de unos sensores que detecten el comportamiento de dicha planta y de unas interfaces para adaptar las señales de los sensores a las entradas del sistema de control. Este tipo de sistemas se denominan en lazo cerrado, ya que su diagrama muestra claramente una estructura con una cadena directa y un retorno o realimentación, formando un lazo de control.



Sistema de control en lazo cerrado

Así pues, en el caso más general, Podremos dividir el sistema de control en los siguientes bloques:

- Unidad de control.
- Accionamientos.
- Sensores.
- Interfaces.

El papel del autómat programable dentro del sistema de control es el de unidad de control, aunque suele incluir también, total o parcialmente, las interfaces con las señales de proceso.

Al conjunto de señales de consigna y de realimentación que entran a la unidad de control se les llama entradas y al conjunto de señales obtenidas, salidas.

1.1.2. AUTOMATISMOS ANALOGICOS Y DIGITALES

Según la naturaleza de las señales que intervienen en el proceso, los sistemas de control pueden dividirse en analógicos, digitales o híbridos de ambos.

Ciertos sensores o aparatos pueden requerir entradas o salidas digitales de nuestro sistema de control como es el caso de los sensores todo/nada. Por otra parte puede ser necesario actuar sobre salidas analógicas para actuar sobre servosistemas externos o para ciertos indicadores o tener entradas de este tipo que nos permitan medir la señal generada por un sensor de temperatura o presión.

Todo esto nos lleva a que los sistemas actuales suelen ser híbridos entre uno analógico y otro digital.

Además, para poder utilizar esta información de tipo analógico nuestro sistema de control deberá tener conversores A/D y D/A.

1.1.3. AUTOMATISMOS CABLEADOS Y PROGRAMABLES

Una de las claves del éxito de los autómatas programables frente a los equipos de relés, o incluso frente a equipos construidos a base de circuitos integrados, ha sido la posibilidad de realizar funciones muy diversas con un mismo equipo (hardware estándar) y cambiando únicamente un programa (software).

Atendiendo a este criterio podemos clasificar los sistemas de control en dos grandes grupos:

- Sistemas cableados (poco adaptables).
- Sistemas programables (muy adaptables).

Los primeros realizan una función de control fija, que depende de los componentes que lo forman y de la forma en que se han interconectado. Por tanto, la única forma de alterar la función de control es modificando sus componentes o la forma de interconectarlos.

Los sistemas programables, en cambio, pueden realizar distintas funciones de control sin alterar su configuración física, sino sólo cambiando el programa de control.

El atributo «programable» hay que interpretarlo como «programable por el usuario», con lo cual éste obtiene los beneficios de un equipo multifunción con un hardware fijo. La base sigue siendo un equipo con microprocesador, al cual se ha incorporado un programa intérprete, capaz de alterar la función de transferencia salida/entrada en razón de un programa de usuario.

CARACTERÍSTICA	SISTEMA CABLEADO	PLC
Flexibilidad de adaptación al proceso	Baja	Alta
Hardware estándar para distintas aplicaciones	No	Sí
Posibilidades de ampliación	Bajas	Altas
Interconexiones y cableado exterior	Mucho	Poco
Tiempo de desarrollo del proyecto	Largo	Corto
Posibilidades de modificación	Difícil	Fácil
Mantenimiento	Difícil	Fácil
Herramientas para prueba	No	Sí
Stocks de mantenimiento	Medios	Bajos
Modificaciones 'on line'	No	Sí
Coste para pequeñas series	Alto	Bajo
Estructuración en bloques independientes	Difícil	Fácil

Comparación de sistemas cableados y sistemas programables

1.1.4. EL AUTOMATA PROGRAMABLE EN EL CONTROL INDUSTRIAL

El autómata programable es la unidad de control, incluyendo total o parcialmente las interfaces con las señales de proceso. Por otro lado, se trata de un sistema con un hardware estándar, con capacidad de conexión directa a las señales de campo (niveles de tensión y corriente industriales, transductores y periféricos electrónicos) y programable por el usuario.

Al conjunto de señales de consigna y de realimentación que entran en el autómata se les denomina genéricamente entradas y al conjunto de señales de control obtenidas, salidas, pudiendo ser ambas analógicas o digitales.

El concepto de hardware estándar que venimos indicando para el autómata se complementa con el de modularidad, entendiendo como tal el hecho de que este hardware está fragmentado en partes interconectables que permiten configurar un sistema a la medida de las necesidades.

Así pues, encontramos autómatas compactos que incluyen una unidad de control y un mínimo de entradas y salidas y luego tienen previstas una serie de unidades de expansión que les permiten llegar hasta 128 ó 256 entradas/salidas. Para aplicaciones más complejas se dispone de autómatas montados en rack con posibilidad hasta unas 2000 entradas/salidas controladas por una única unidad central (CPU).

Existe también la posibilidad, en autómatas grandes, de elección entre varios tipos de CPU, adaptados a la tarea que deba realizarse o incluso de múltiples CPU trabajando en paralelo en tareas distintas.

Así, las posibilidades de elección, tanto en capacidad de proceso como en número de entradas/salidas, son muy amplias y esto permite afirmar que se dispone siempre de un hardware estándar adaptado a cualquier necesidad.

Esta adaptabilidad ha derivado hacia el concepto de inteligencia distribuida, gracias a las comunicaciones entre autómatas y a las redes autómata/ordenador. Esta técnica sustituye el gran autómata, con muchas entradas/salidas controladas por una única CPU, por varios autómatas, con un número menor de E/S, conectados en red y controlando cada punto o sección de una planta bajo el control de una CPU central.

1.1.5. CONTROL POR ORDENADOR

Algunos procesos complejos requieren sistemas de control con una gran capacidad de cálculo, conexión a estaciones gráficas, múltiples canales de comunicación, facilidad de adaptación, capacidad de multiproceso, etc. Para ellos se han venido utilizando mini ordenadores a los que se han adaptado interfaces específicas para la planta a controlar.

Actualmente esta solución no está descartada, pero resulta económicamente cara y poco estándar, sobre todo por el hecho de que el ordenador no suele disponer de interfaces adecuadas para recoger y enviar las señales de planta.

En general, la elección de un sistema de control con PC o PLC depende de diversos factores. Por una parte, lo que se busca con un sistema de control es que pueda responder en tiempo real a un evento en la planta y para esto están especialmente diseñados los PLC's. En este contexto, tiempo real significa que responda en un tiempo especificado en el 100% de los casos. Además los PLC's están diseñados para ser utilizados en ambientes industriales exigentes. Por otro lado, los PC's están mejor preparados para realizar grandes cálculos y operaciones con bases de datos o ficheros además de ser un medio generalmente barato y altamente estándar en componentes y software.

Hay que considerar, además, que la frontera entre un autómata de gama alta y un ordenador es cada vez más difusa, ya que dichos autómatas incorporan funciones de cálculo potentes, capacidad de programación en alto nivel, herramientas de gestión de la producción, etc., y, por otro lado, permiten fácilmente comunicarse entre sí o con un ordenador central.

Así pues, los PLC's han sido diseñados específicamente para el control en ambientes duros y donde se necesite alta fiabilidad. Estos factores, unidos a su sencillez de uso y alto rendimiento los convierten en la opción prioritaria en el control de sistemas donde un PC podría estar simplemente "sobrecargado".

Un inconveniente para la adopción de los PC's en control es lo cerrado de las soluciones que dan las marcas de PLC's que no tienen un standard en las extensiones hardware para control en tiempo real. Cada empresa da su propia solución. Hay que tener en cuenta que las tarjetas de expansión para PC no dejan de ser PLC's, con lo cual volvemos a trabajar de un modo u otro con los PLC's.

Así pues, la tendencia actual en el control de procesos complejos es utilizar los autómatas en red o como periféricos de un ordenador, con lo cual se combinan la potencia de cálculo del ordenador y la facilidad de interfaces estándar que ofrece el autómata.

El sistema de control resultante de esta combinación ofrece las siguientes prestaciones:

- Sistema programable con una gran potencia de cálculo.
- Gran cantidad de software estándar para manipulación de datos y gestión de la producción.

- Interfaces estándar de ordenador para estaciones gráficas, utilizadas para monitorizar el proceso.
- Control descentralizado con inteligencia distribuida, sin interrumpir todo el proceso cuando hay fallos del control central.
- Sistemas de comunicación estándar LAN o WAN.
- Facilidad de interfaz con la planta.
- Mantenimiento fácil por secciones.
- Disponibilidad de herramientas de test y mantenimiento.
- Posibilidad de visualizar el proceso en tiempo real.
- Programación fácil al nivel de secciones.
- Flexibilidad para realizar cambios.

2. INTRODUCCIÓN A LOS AUTÓMATAS PROGRAMABLES

2.1. HISTORIA DE LOS AUTÓMATAS PROGRAMABLES

Los PLC's se introdujeron por primera vez en la industria en la década de 1960. La razón principal fue la necesidad de eliminar el alto coste que se producía al reemplazar el complejo sistema de control basado en relés y contactores. Bedford Associates propuso algo denominado Controlador Digital Modular (MODICON, MOdular DIGital CONtroller) a un gran fabricante de coches. Otras compañías propusieron esquemas basados en ordenador, uno de los cuales estaba basado en el PDP-8. El MODICON 084 resultó ser el primer PLC del mundo en ser producido comercialmente.

El problema de los relés era que cuando los requerimientos de producción cambiaban también lo hacía el sistema de control. Esto comenzó a resultar bastante caro cuando los cambios fueron frecuentes. Dado que los relés son dispositivos mecánicos y poseen una vida limitada se requería un mantenimiento planificado. Por otra parte, a veces se debían realizar conexiones entre cientos o miles de relés, lo que implicaba un enorme esfuerzo de diseño y mantenimiento.

Los nuevos controladores debían ser fácilmente programables por ingenieros de planta o personal de mantenimiento. El tiempo de vida debía ser largo y los cambios en el programa tenían que realizarse de forma sencilla. Finalmente se imponía que trabajaran sin problemas en entornos industriales adversos. La solución fue el empleo de una técnica de programación familiar y reemplazar los relés mecánicos por relés de estado sólido.

A mediados de los 70 las tecnologías dominantes de los PLC eran máquinas de estado secuenciales y CPU basadas en desplazamiento de bit. Los AMD 2901 y 2903 fueron muy populares. Los microprocesadores convencionales proveyeron la potencia necesaria para resolver de forma rápida y completa la lógica de los pequeños PLC's. Por cada modelo de microprocesador había un modelo de PLC basado en el mismo. No obstante, el 2903 fue de los más utilizados.

La capacidad de comunicación comenzó a aparecer en 1973 aproximadamente. El primer sistema fue el bus Modicon (Modbus). El PLC podía ahora dialogar con otros PLC y ubicarse alejado de las máquinas que controlaba. También se introdujeron en el mundo analógico al poder enviar y recibir señales de tensión. Desafortunadamente, la falta de un estándar acompañado de un continuo cambio tecnológico ha hecho que la

comunicación de PLC sea un cúmulo inconexo de sistemas físicos y protocolos incompatibles entre sí.

En los años 80 se produjo un intento de estandarización de las comunicaciones con el protocolo MAP (Manufacturing Automation Protocol) de General Motor's. También se redujeron las dimensiones del PLC y se pasó a programar con lenguajes simbólicos a través de ordenadores personales en vez de los clásicos terminales de programación.

Los años 90 mostraron una gradual reducción en el número de nuevos protocolos de comunicación, y en la modernización de las capas físicas de los protocolos más populares que sobrevivieron a los 80. El estándar IEC 1131-3 intenta unificar el sistema de programación de todos los PLC en un único estándar internacional. Ahora disponemos de PLC's que pueden ser programados en diagramas de bloques, lista de instrucciones, C y texto estructurado al mismo tiempo.

Los PC están comenzando a reemplazar al PLC en algunas aplicaciones, incluso la compañía que introdujo el Modicon 084 ha cambiado al control basado en PC.

2.2. DEFINICIÓN DE AUTÓMATA PROGRAMABLE

Se trata de un sistema con un hardware estándar, con capacidad de conexión directa a las señales de campo (niveles de tensión y corriente industriales, transductores y periféricos electrónicos) y programable por el usuario que hace de unidad de control incluyendo total o parcialmente las interfaces con las señales de proceso.

Otra definición algo más simple entiende por controlador lógico programable (PLC), o autómata programable, a toda máquina electrónica diseñada para controlar en tiempo real y en medio industrial procesos secuenciales.

2.3. CAMPOS DE APLICACIÓN

Un autómata programable suele emplearse en procesos industriales que tengan una o varias de las siguientes necesidades:

- Espacio reducido.
- Utilización en ambientes exigentes o agresivos.
- Procesos de producción periódicamente cambiantes.
- Procesos secuenciales.
- Maquinaria de procesos variables.
- Instalaciones de procesos complejos y amplios.
- Chequeo de programación centralizada de las partes del proceso.

Aplicaciones generales:

- Maniobra de máquinas.
- Maniobra de instalaciones.
- Señalización y control.

Esto se refiere a los autómatas programables industriales, dejando de lado los pequeños autómatas para uso más personal (que se pueden emplear, incluso, para automatizar procesos en el hogar, como la puerta de un cochera o las luces de la casa).

2.4. VENTAJAS E INCONVENIENTES DE LOS AUTÓMATAS

Entre las ventajas de los autómatas frente a los sistemas cableados podemos citar:

- Menor tiempo de elaboración de proyectos.

- Posibilidad de añadir modificaciones sin coste añadido en otros componentes.
- Mínimo espacio de ocupación.
- Menor costo de mano de obra.
- Mantenimiento económico.
- Posibilidad de gobernar varias máquinas con el mismo autómeta.
- Menor tiempo de puesta en funcionamiento.
- Si el autómeta queda pequeño para el proceso industrial puede seguir siendo de utilidad en otras máquinas o sistemas de producción.

Y entre los inconvenientes:

- Adiestramiento de técnicos.
- Coste.

A día de hoy, los inconvenientes se han minimizado, ya que la formación previa del personal suele incluir la automatización como una de sus asignaturas. En cuanto al costo tampoco hay problema, ya que hay autómetas para todas las necesidades y a precios ajustados (tenemos desde pequeños autómetas por unos 100€ hasta PLC's que alcanzan cifras exorbitantes).

2.5. ESTRUCTURA EXTERNA

En cuanto a su estructura, todos los autómetas programables se clasifican en:

- Compactos: en un solo bloque están todos los elementos.
- Modulares: separan por unidades las distintas partes operativas.

Y, en este caso, se distingue entre:

- Estructura americana: separa las E/S del resto del autómeta.
- Estructura europea: cada módulo es una función (fuente de alimentación, CPU, E/S, etc.).

Exteriormente nos encontraremos con cajas que contienen una de estas estructuras, las cuales poseen indicadores y conectores en función del modelo y fabricante.

Para el caso de una estructura modular se dispone de la posibilidad de fijar los distintos módulos en raíles (o *racks*) normalizados, para que el conjunto sea compacto y resistente.

Los micro-autómetas pueden servirse sin caja, en formato kit, ya que su empleo no es determinado y se suele incluir dentro de un conjunto más grande de control o dentro de la misma maquinaria que se debe controlar.

2.6. ESTRUCTURA INTERNA

Los elementos esenciales, que todo autómeta programable posee como mínimo, son:

- Sección de entradas: se trata de líneas de entrada, las cuales pueden ser de tipo digital o analógico. En ambos casos tenemos unos rangos de tensión característicos, los cuales se encuentran en las hojas de características del fabricante. A estas líneas conectaremos los sensores.
- Sección de salidas: son una serie de líneas de salida, que también pueden ser de carácter digital o analógico. A estas líneas conectaremos los actuadores.
- Unidad central de proceso (CPU): se encarga de procesar el programa de usuario que le introduciremos. Para ello disponemos de diversas zonas de memoria, registros, e instrucciones de programa. Adicionalmente, en determinados modelos

más avanzados, podemos disponer de funciones ya integradas en la CPU, como reguladores PID, control de posición, etc.

- Memoria: Es la encargada de contener el programa de usuario y de trabajo. En los autómatas está separada en diversas áreas según su función o datos que debe contener. Las imágenes de salidas y entradas se hacen en memoria.

Tanto las entradas como las salidas están aisladas de la CPU según el tipo de autómata que utilicemos. Normalmente se suelen emplear optoacopladores en las entradas y relés/optoacopladores en las salidas.

Aparte de estos elementos podemos disponer de los siguientes:

- Unidad de alimentación (algunas CPU la llevan incluida).
- Unidad o consola de programación: que nos permitirá introducir, modificar y supervisar el programa de usuario.
- Dispositivos periféricos: como nuevas unidades de E/S, más memoria, unidades de comunicación en red, etc.
- Interfaces: facilitan la comunicación del autómata mediante enlace serie con otros dispositivos (como un PC).

En los siguientes apartados se comentará la estructura de cada elemento.

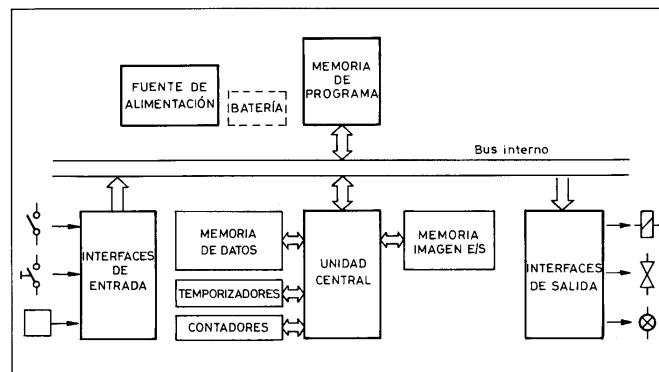


Diagrama de bloques de un autómata programable

2.6.1. MEMORIA

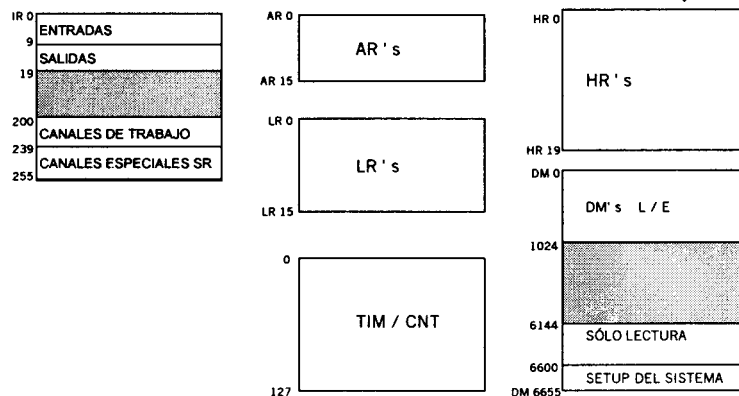
Vamos a disponer de un área de memoria, la cual emplearemos para diversas funciones:

- Memoria del programa de usuario: aquí introduciremos el programa que el autómata va a ejecutar cíclicamente.
- Memoria interna: contiene datos intermedios de los cálculos realizados así como variables internas y una imagen de las entradas y salidas.
- Memoria de datos: se suele subdividir en zonas según el tipo de datos (como temporizadores, contadores, etc.).
- Memoria del sistema: aquí se encuentra el programa en código máquina que monitoriza el sistema (programa del sistema o firmware). Este programa es ejecutado directamente por el microprocesador/microcontrolador que posea el autómata.
- Memoria auxiliar: se trata de memoria externa que empleamos para almacenar el programa de usuario, y en ciertos casos parte de la memoria de la tabla de datos.

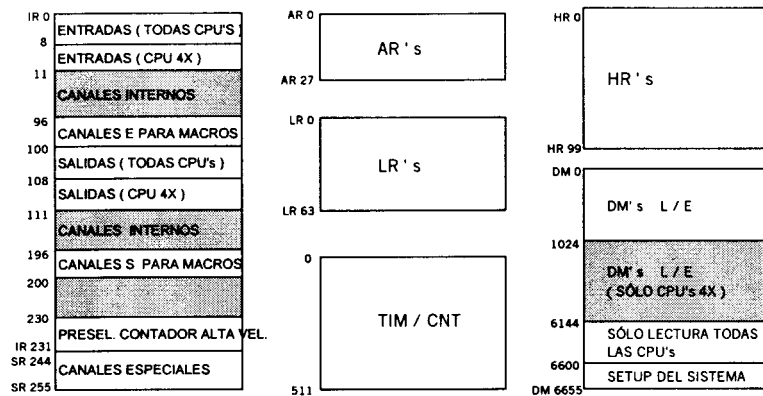
La memoria puede ser accesible bit a bit o en palabras de 8 o 16 bits.

Cada autómata divide su memoria de esta forma genérica, haciendo subdivisiones específicas según el modelo y fabricante como, por ejemplo, la zona de memoria de

datos de 16 bits (DM) o la zona de memoria no volátil accesible a nivel de bit (HR) en los autómatas de OMRON.



Mapa de Memoria de CPM1 de OMRON



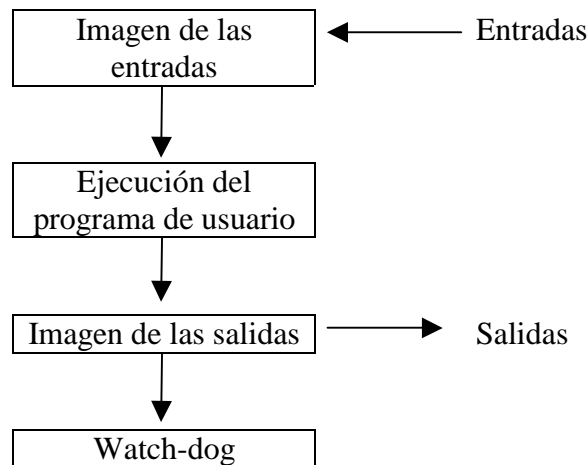
Mapa de memoria de CQM1 de OMRON

2.6.2. CPU

La CPU es el corazón del autómata programable. Es la encargada de ejecutar el programa de usuario mediante el programa del sistema (es decir, el programa de usuario es interpretado por el programa del sistema). Sus funciones son:

- Vigilar que el tiempo de ejecución del programa de usuario no excede un determinado tiempo máximo (tiempo de ciclo máximo). A esta función se le suele denominar Watchdog.
- Ejecutar el programa de usuario.
- Actualizar los contadores y temporizadores programados.
- Crear una imagen de las entradas, ya que el programa de usuario no accede directamente a dichas entradas.
- Renovar el estado de las salidas en función de la imagen de las mismas obtenida al final del ciclo de ejecución del programa de usuario.
- Chequeo del sistema.

Para ello el autómata va a poseer un ciclo de trabajo, que ejecutará de forma continua:



La CPU ejecuta el programa de usuario secuencialmente y casi siempre de manera interpretada con decodificación de las instrucciones cada vez que son ejecutadas.

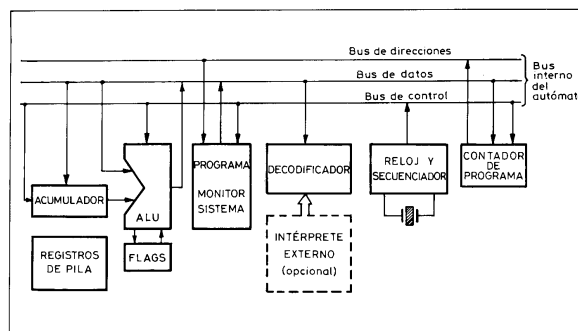


Diagrama de bloques de la Unidad Central de Proceso CPU

2.6.3. UNIDADES DE E/S

Generalmente vamos a disponer de dos tipos de E/S:

- Digital.
- Analógica.

Las E/S digitales se basan en el principio de todo o nada. Estas E/S se manejan a nivel de bit dentro del programa de usuario. Pueden ser de tipo relé o de tipo diodo.

Las E/S analógicas pueden poseer cualquier valor dentro de un rango determinado especificado por el fabricante. Se basan en convertores A/D y D/A aislados de la CPU (ópticamente o por etapa de potencia). Estas señales se manejan a nivel de byte o palabra (8/16 bits) dentro del programa de usuario.

Las E/S son leídas y escritas dependiendo del modelo y del fabricante, es decir pueden estar incluidas sus imágenes dentro del área de memoria o ser manejadas a través de instrucciones específicas de E/S.

2.6.4. BUS INTERNO

Se conoce como bus interno al conjunto de líneas y conexiones que permiten la unión eléctrica entre la unidad de control, las memorias y las interfaces de entrada y salida. Un bus se compone de un conjunto de líneas utilizadas para intercambiar datos u

órdenes (por ejemplo el contenido de celdas de memoria o las instrucciones de la unidad de control). Permite minimizar el número de conexiones entre subsistemas y el acceso de los dispositivos al mismo es controlado por la unidad de control.

Los tres buses característicos de un sistema digital son:

- bus de datos, por el que tienen lugar las transferencias de datos del sistema,
- bus de direcciones, a través del cual se direccionan la memoria y el resto de los periféricos, y,
- bus de control, constituido por todas las conexiones destinadas a gobernar los intercambios de información, se reunifican en el autómata en uno sólo, que recibe el nombre de bus interno.

El número de líneas del interno depende de cada fabricante.

Se considera también como bus del autómata cualquier conexión entre bloques o módulos que no necesite de procesadores específicos de comunicaciones en sus extremos, como, por ejemplo, el cable de conexión entre el autómata y una unidad externa de expansión de E/S.

2.7. INTERFACES

Todo autómata, salvo casos excepcionales, posee la virtud de poder comunicarse con otros dispositivos (como un PC o un MODEM).

Lo normal es que posea una E/S serie del tipo RS-232 / RS-422.

A través de esta línea se pueden manejar todas las características internas del autómata, incluida la programación del mismo, y suele emplearse para monitorización del proceso en otro lugar separado.

2.8. EQUIPOS O UNIDADES DE PROGRAMACIÓN

El autómata debe disponer de alguna forma de programación, la cual se suele realizar empleando alguno de los siguientes elementos:

- Unidad de programación: suele ser en forma de calculadora. Es la forma más simple de programar el autómata, y se suele reservar para pequeñas modificaciones del programa o la lectura de datos en el lugar de colocación del autómata.
- Consola de programación: es un terminal a modo de ordenador que proporciona una forma más cómoda de realizar el programa de usuario y observar parámetros internos del autómata. Su alto coste y la ubicuidad del ordenador portátil han relegado su uso.
- PC: es el modo más potente y empleado en la actualidad. Permite programar desde un ordenador personal estándar, con todo lo que ello supone: herramientas más potentes, posibilidad de almacenamiento en soporte magnético, impresión, transferencia de datos, monitorización mediante software SCADA, etc.

Para cada caso el fabricante proporciona lo necesario, bien el equipo o el software/cables adecuados. Cada equipo, dependiendo del modelo y fabricante, puede poseer una conexión a uno o varios de los elementos anteriores. En el caso de los micro-PLC se escoge la programación por PC o por unidad de programación integrada en la propia CPU.

2.9. DISPOSITIVOS PERIFÉRICOS

El autómata programable, en la mayoría de los casos, puede ser ampliable. Las ampliaciones abarcan un gran abanico de posibilidades, que van desde las redes internas (LAN, buses de campo como Profibus, CAN-Bus, etc.), módulos auxiliares de E/S, memoria adicional... hasta la conexión con otros autómatas del mismo modelo. Cada fabricante facilita las posibilidades de ampliación de sus modelos, los cuales pueden variar incluso entre modelos de la misma serie.

2.10. FUNCIONAMIENTO DE UN AUTÓMATA

Los autómatas programables son máquinas de funcionamiento secuencial que ejecutan las instrucciones de programa que se les introduce, una detrás de otra y continuamente mientras el autómata está funcionando. Utilizan en la ejecución del programa las señales de entrada al PLC y generan unas señales de salida para el control de la planta.

El programa y una serie acciones comunes de funcionamiento del autómata como por ejemplo vigilar que el tiempo de ejecución del programa de usuario no excede un determinado tiempo máximo (tiempo de ciclo máximo), actualizar los contadores y temporizadores programados, etc., se ejecutan de manera periódica y cíclica en lo que se suele denominar el ciclo de 'scan' o ciclo de operación.

La secuencia de operación suele ser la siguiente:

- Lectura de señales de entrada desde interfaces de entrada.
- Escritura de señales de salida a interfaces de salida.
- Procesado del programa.

Las señales de entrada a través de los interfaces de entrada son copiadas a una memoria intermedia de manera que el programa que se ejecuta no accede nunca directamente a las entradas. Se crea una imagen de la entrada.

Lo mismo se hace con las salidas pero esta vez la CPU escribirá la memoria con las salidas (imagen de la salida) en la salida o interfaz de salida.

Estas dos acciones se ejecutan de una sola vez para ahorrar tiempo y ser más eficiente.

Además, un autómata tiene varios modos de funcionamiento como son:

RUN: El autómata ejecuta el programa de usuario y hace funcionar contadores y temporizadores evolucionando normalmente.

STOP: El autómata está parado y listo para ser programado o para labores de mantenimiento.

ERROR: Ante una situación de mal funcionamiento del aparato, éste se detiene y bloquea a la espera de atención por el programador o salida del error que causó la detención.

Algunos autómatas tienen otro modo denominado PROGRAM. La diferencia entre RUN y PROGRAM es que en el primero ningún dispositivo de programación puede forzar o modificar ninguna posición de memoria en el PLC y en el segundo sí.

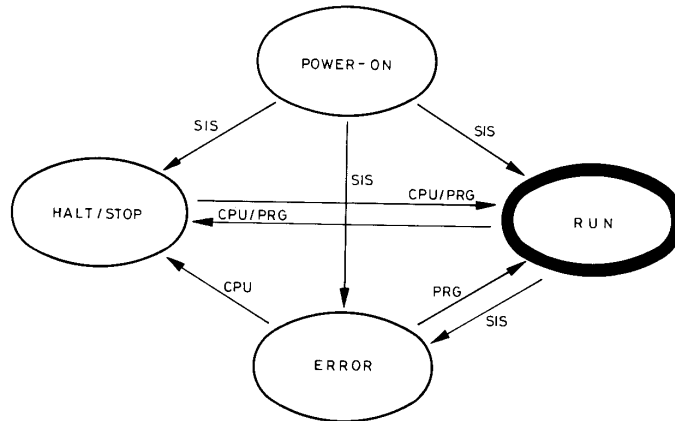
El paso entre estos modos de funcionamiento se hace desde consolas o botones. El modo actual suele estar indicado mediante LEDs en la carcasa del PLC.

Existe un área de memoria no volátil en la que puede configurarse el modo de arranque del autómata. Puede hacerlo en modo STOP o RUN.

En cuanto al modo de funcionamiento RUN, cuando se arranca un PLC, se ejecutan una serie de comprobaciones del hardware y se hace un borrado de ciertas partes de la memoria del PLC antes de pasar al ciclo que se repite continuamente durante el funcionamiento del aparato.

Esta primera fase suele ser muy rápida (menos de 1 segundo).

Luego, la fase del ciclo de operación se divide en tres partes como es la de procesos comunes, la de ejecución del programa de usuario y la de atención a periféricos.



Modos de operación de un autómata

Fase de procesos comunes:

En esta fase se comprueba periódicamente que el programa de usuario no tarde más de un tiempo en ejecutarse (watchdog), que no existan errores de conexiones y que no existan errores de sintaxis del programa.

El watchdog es un reloj interno no accesible por el usuario (salvo para ser inicializado o programar su tiempo de cuenta) que genera una interrupción al transcurrir el tiempo establecido si no es re-inicializado. Por tanto, comprueba que el programa alcanza la instrucción END antes de un tiempo dado, y que no se ha producido una detención por un bucle infinito u otras causas.

La comprobación de conexiones comprueba los niveles de tensión en las conexiones y en la pila, si existe, y en los buses.

En la comprobación del programa se revisa si el mismo está correctamente escrito y si se mantienen los datos.

Fase de ejecución de programa:

En esta fase se atienden las entradas leyendo su contenido y se actualizan las salidas escribiendo los nuevos valores al finalizar la fase.

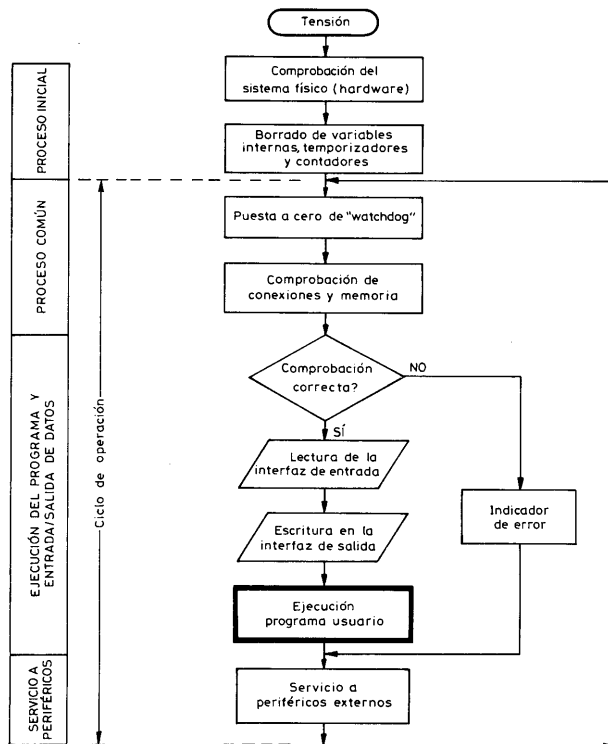
También se ejecuta el programa guardado en su memoria.

El tiempo que tarde el plc en ejecutarlo depende del procesador utilizado, de la longitud del programa, del tipo de instrucciones usadas en el programa y del número y ubicación de las interfases de E/S. Este tiempo suele ser del orden de 5ms para un autómata compacto con interfaces locales. No cuesta de ejecutar lo mismo instrucciones AND, OR, que otras de movimiento de datos o de transmisión a través de un puerto.

Fase de atención a periféricos:

Esta fase sólo se atiende si existen peticiones de intercambio de información pendientes con los periféricos conectados a la CPU o a procesadores auxiliares a éste.

El autómata contiene rutinas cíclicas de auto chequeo y de arranque en ROM con las que comprueban el programa de usuario y el propio hardware y en caso de detectar algún problema, registrarlo y, si es grave, detener la ejecución e informar del mismo mediante leds o un número de error en un registro.



Ciclo de SCAN típico de un autómata

Resumiendo, el tiempo de ciclo total será la suma de los distintos tiempos antes comentados como son la atención a periféricos, actualización de E/S, ejecución del programa y el autodiagnóstico.

Los tiempos típicos de ejecución de chequeos y comprobaciones suelen ser de 1ms o 2ms y los tiempos de ejecución de instrucciones se miden en μ s.

Estos tiempos son críticos cuando se utilicen sistemas que necesiten una alta velocidad aunque no sólo de ellos depende el tiempo de reacción del equipo. Las entradas y salidas aportan un retardo importante debido a los filtros que a menudo utilizan para eliminar ruidos y que limitan las frecuencias máximas a no más de 100 Hz

		Autómata	Número de puntos de E/S	Tiempo de ejecución (us/instr.)
GAMA DE AUTÓMATAS	ALTA	C 2000 H	2048	0,4 a 2,4
		C 1000 H		
		C 200 H		
	MEDIA	C 500	512	3 a 83
		C 120	256	3 a 83
	BAJA	Serie K	100 a 148	4 a 95
Serie P		36 a 120	4 a 95	
C 20		84	4 a 80	

Familia de autómatas OMRON y tiempos de ejecución

	ENTRADA (Tin)		SALIDA (Tout)		
	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

Retardos de conmutación típicos de interfaces de autómatas (ms)

Una manera en que se solucionan en parte las necesidades de alta velocidad es utilizando elementos de entrada para procesos rápidos como contadores rápidos, entradas con filtrado débil, etc.

2.11. CONFIGURACIONES DE UN SISTEMA CON AUTÓMATA

Con el paso del tiempo, los autómatas han pasado de ser lentos, compactos y con un reducido número de instrucciones a ser capaces de ejecutar todo tipo de instrucciones, ser modulares y capaces de comunicarse con gran rapidez con otros elementos tanto analógicos como digitales.

Se entiende por configuración de un autómata la selección de la estructura física que adopta la unidad de control, la memoria y las entradas y salidas especialmente para adaptarlo a unos requerimientos determinados.

Así, podemos decir que existen básicamente dos configuraciones para la unidad de control:

- Unidad de control compacta (control centralizado).
- Unidad de control modular (control distribuido).

En el primer caso una sola CPU controla varios módulos de E/S que son gestionados desde una única CPU.

En el segundo caso varias CPU controlan sus respectivos módulos de E/S y a su vez están conectados a una CPU maestra que gestiona el sistema.

Otra clasificación respecto a la configuración de las entradas/salidas sería de E/S distribuidas o centralizadas siendo la primera configuración cuando el autómata se comunica con ellas a través de otros procesadores al efecto y la segunda configuración cuando están conectadas con un bus interno directamente al cpu.

Respecto a los procesadores utilizados, han ido incrementando su potencia y han pasado de manejar palabras de 4 y 8 bits a 16 ó 32 bits y por consiguiente han ido aumentando su complejidad y su capacidad de cálculo. Se utilizaron y se siguen utilizando procesadores comerciales de propósito general como los de Intel o Motorola por su reducido coste aunque a costa de reducir la velocidad de ejecución.

Pero el tamaño de palabra de los procesadores comerciales no es muy adecuado para manejo de datos de tipo bit con lo que, con el tiempo, el procesador único se dividió en un microprocesador de palabras y en otro de bit.

Los microprocesadores de bit son los utilizados en operaciones booleanas entre variables de 1 bit y los microprocesadores de palabra operan con operaciones de 8 y más bits.

Los procesadores de bit son dispositivos con tecnología ASIC (Application Specific Integrated Circuit) sobre circuitos de puertas (Gate Array) fabricados casi a medida (custom) a las especificaciones del fabricante del autómata y a las instrucciones particulares con tiempos de ejecución de 1µs/instrucción aproximadamente.

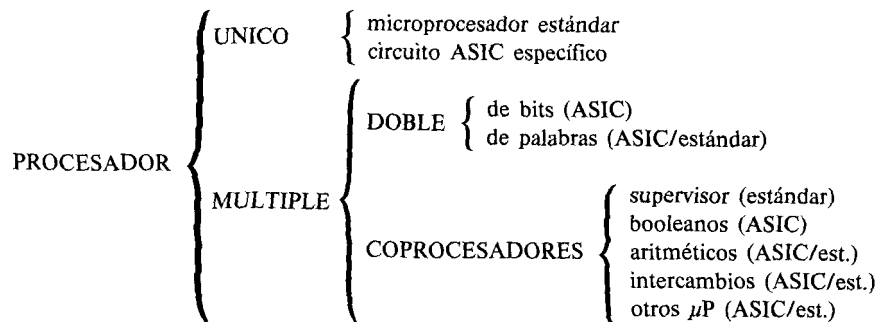
Para los procesadores de palabras se utilizan procesadores comerciales adaptados al lenguaje del autómata con un intérprete o con un circuito ASIC. Los autómatas pequeños llevan un único procesador mixto bit/palabra mientras que los de gama alta subdividen incluso los procesadores de palabra especializándolos en comunicaciones, cálculo u otros procesos. Por supuesto, esta es la solución idónea para optimizar los tiempos de respuesta del sistema pero resulta más cara.

Se ha hablado de la estructura del procesador, veamos ahora la unidad de control.

La unidad de control necesita cada vez ser más compleja y desarrollar más tareas más diversas. Entonces se puede optar por dos soluciones, usar un único procesador todo lo complejo que sea necesario o varios que se repartan el trabajo según el tipo. Esta última solución conlleva multiproceso (varios procesadores trabajan en paralelo ejecutando diversas tareas).

Según se conecten estos procesadores, tenemos tres tipos de arquitecturas:

- multiprocesadores centrales.
- Procesador central y procesadores periféricos.
- Procesadores en red.



Clasificación de procesadores para las CPU de autómatas

Los procesadores en red son procesadores independientes y autónomos que trabajan de manera coordinada conectados a través de una red local con un bus industrial.

Los multiprocesadores centrales son una agrupación de unidades centrales (CPU) coordinadas por una de ellas y que cada una ejecuta parte del programa conectados a un bus interno de modo que funcionan como uno solo. Aquí el coordinador es fundamental pues controla y gestiona toda la información que pasa por él, distribuye las tareas y el uso del bus entre otras tareas. Una unidad de control así, es un sistema a la carta fácilmente ampliable y programable, que reduce los tiempos de respuesta frente a señales rápidas de la planta puesto que puede distribuir las tareas críticas y no críticas entre sus procesadores, especializados y dedicados en cada una de ellas y que comparten el mismo mapa de E/S.

Los procesadores centrales con procesadores periféricos, son utilizados debido a la necesidad de realizar tareas específicas, repetitivas y que requieren gran potencia de cálculo y atención por parte de la unidad de control. El módulo especializado en cuestión, se encarga, con tan solo los parámetros necesarios, de controlar el proceso y de realizar todos los cálculos necesarios independientemente de la unidad central y con instrucciones mejor adaptadas a sus necesidades que las más generales del autómatas.

Con una unidad de este tipo, se libera al procesador central y se atienden mejor determinados procesos.

Según su complejidad, se pueden dividir en:

- Interfaces inteligentes de E/S.
- Procesadores periféricos inteligentes.

Los interfaces inteligentes de E/S son módulos de entradas y salidas con cierta capacidad de proceso aunque muy limitada.

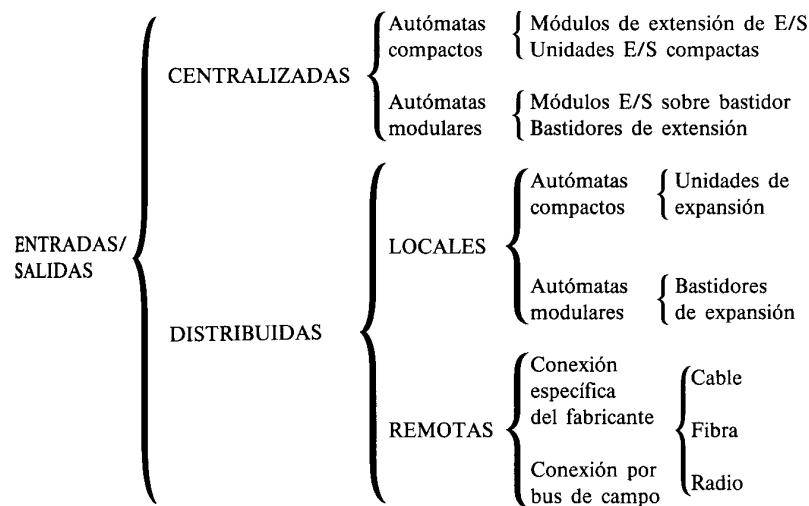
Los procesadores periféricos inteligentes son procesadores auxiliares con sus propias E/S especializados en algún tipo de control específico y con capacidad de cálculo y

programas de tratamiento específico. Se comunican con la CPU para el intercambio de información reducida.

Otra arquitectura para aumentar la seguridad y fiabilidad en un proceso es la de varias CPU redundantes que se supervisan y controlan para que, en el caso de que falle alguna, la otra tome el control.

Respecto a las configuraciones de entradas y salidas, existen dos tipos de distribuciones:

- Sistema de E/S distribuidas, y
- Sistema de E/S centralizadas.



Clasificación del sistema de E/S

Según la aplicación en la que se va a montar el PLC, puede ser necesaria una disposición concreta de los módulos E/S. Con el tiempo y especialmente en autómatas de gama alta los autómatas se han ido modularizando para adaptarse a diversas necesidades. Tenemos entonces módulos de E/S conectables al PLC directamente o por bus.

Las E/S centralizadas se unen a la CPU directamente por el bus interno, están cerca del autómata, intercambian datos con la CPU en paralelo y con frecuencia no llevan fuente de alimentación propia. La ampliación máxima de E/S depende del tipo y modelo del autómata.

Las E/S distribuidas se utilizan cuando el autómata no está físicamente cerca de los puntos de entrada o salida a captar. Dado que cablear largas distancias no es recomendable por coste y riesgos de error, se distribuyen módulos de E/S por la planta que recogen las señales y las envían al autómata, en modo serie, a través de un bus.

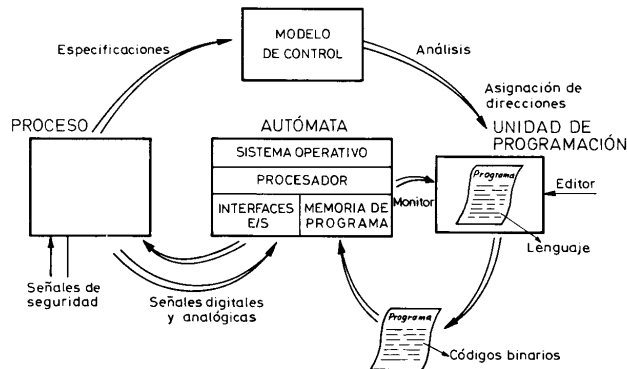
2.12. INTRODUCCIÓN A LA PROGRAMACIÓN DE AUTÓMATAS

Hasta ahora se ha tratado el hardware de los autómatas y ahora nos introduciremos en la parte software.

Se va a hacer un repaso de las distintas representaciones de los sistemas de control que luego se codificarán y de los lenguajes de programación de autómatas.

El software asociado a un autómata consta de dos partes. Por una parte tenemos el sistema operativo (firmware) residente en el autómata que ejecuta las órdenes del programa de usuario y vigila el correcto funcionamiento del equipo y por otra parte tenemos el software de edición y depuración de programas que permite escribir e

introducir el programa en el autómatas o en un soporte físico adecuado por parte del programador desde un dispositivo como un PC o una consola de programación. Tenemos entonces una unidad de programación sobre la que se escribe un programa en un determinado lenguaje y que luego se conecta al PLC para traspasarle la información. Posteriormente, el firmware del PLC interpretará el código y ejecutará el programa.



Para realizar un programa comprensible por el PLC, se ejecutan diversos pasos.

Una vez se sabe lo que se quiere hacer, en qué orden y los medios de que se dispone (E/S, memoria, etc.) se ha de realizar una representación del sistema de control con un modelo que indique las funciones que se van a realizar mediante una representación que puede ser algebraica o gráfica.

Después se asignan las salidas y entradas a los distintos elementos de nuestro modelo y se codifica el modelo anterior en un lenguaje entendible por el PLC. Por último, el código se transfiere al PLC.

Las representaciones de los sistemas de control son una manera de expresar mediante un lenguaje lo que hace exactamente el proceso que queremos estudiar o lo que queremos que haga. Se puede expresar de diversas maneras como por ejemplo con lenguaje usual (descripción literal) pero puede que no nos permita ser muy exactos en las descripciones debido a la complejidad de los sistemas y de la necesidad de dejar perfectamente especificadas las condiciones de trabajo, lo que nos obliga a utilizar útiles simbólicos de representación que nos permitan ser más precisos y dar toda la información necesaria.

Estas representaciones deben de ser comunes a los que la utilicen y coherentes en las reglas que utilicen (sintaxis).

Formas de representación:

- Proposicional: descripciones literales.
- Algebraica: funciones booleanas y aritméticas.
- Gráfica: esquemas, diagramas, grafcet.

Descripciones literales: Esta manera de describir el proceso mediante el lenguaje corriente, no será exacta en las especificaciones aunque explique correctamente el proceso a nivel general. Así, le costará expresar relaciones entre variables, etc.

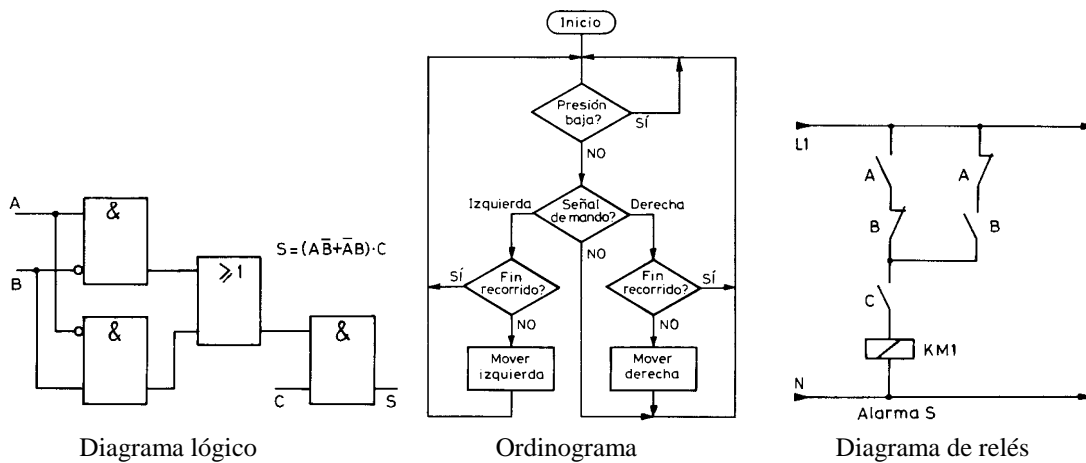
Funciones algebraicas: Esta representación utilizaría funciones booleanas y aritméticas para establecer condiciones entre variables de uno o varios bits. Aunque este sistema es exacto, adolece de no poder expresarse en función del tiempo.

Esquema de relés: Es una representación gráfica de la tarea de automatización mediante símbolos de contactos abierto/cerrado. Con este modelo pueden expresarse sistemas

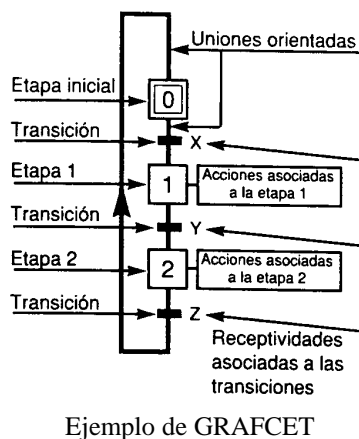
sencillos puesto que es un tanto deficiente en representación de funciones secuenciales complejas como son los secuenciadores, los registros de memoria, etc., y, sobre todo las señales digitales de varios bits.

Diagramas lógicos: Utiliza símbolos que representan las funciones lógicas and, or, etc., o biestables, registros, contadores, etc. Este modo de representación es útil pero está alejada de muchos usuarios finales del autómat, quedando circunscrito a la representación del sistema en alto nivel, a efectos de documentaciones y especificaciones.

Ordinogramas: Este modelo refleja con especial claridad la evolución y toma de decisiones del proceso aunque no es tan claro si se intentan representar variables y señales que intervienen y las relaciones entre ellas a no ser que se utilicen expresiones algebraicas lo que va en contra de la claridad del gráfico del modelo.



Representación GRAFCET: Es una evolución del diagrama de flujos que representa la sucesión de los distintos estados, separados por transiciones, por los que pasa el sistema. Comienza con una etapa inicial y evoluciona por el resto según las condiciones de las transiciones. En cada etapa se representan las distintas acciones a realizar en ella y que se pueden representar aparte con cualquiera de las representaciones anteriores (por ejemplo con diagrama de relés). Esta representación se tratará posteriormente con más amplitud por su importancia.



Ejemplo de GRAFCET

Una vez vistos los distintos tipos de representaciones del sistema que queremos automatizar, se verán los distintos lenguajes que se pueden utilizar para codificar lo expresado con un GRAFCET, un diagrama de relés o cualquiera de las otras representaciones.

Los lenguajes de programación vuelven a ser, como las distintas maneras de representación anteriores, algebraicos y gráficos y, aunque sean distintos, intentan con unas reglas sintácticas definidas, construir un programa coherente y comprensible por el autómeta.

Todos los lenguajes poseen instrucciones con una estructura similar:

Campo de Operación	Operando(s)
--------------------	-------------

El Campo de Operación indica qué hacer como el tipo de operación, etc., mientras que el Operando dice con quién hay que realizarlo como por ejemplo una dirección de memoria (DM150.13) o un valor numérico (136).

Los lenguajes son lo suficientemente completos y complejos como para realizar cualquier tipo de operación desde las más sencillas como las booleanas hasta las más complejas como movimiento de datos, rotaciones de bits o manipulación de ficheros.

Además del lenguaje propiamente dicho puede ser necesario pasarle al autómeta ciertos parámetros para configurar su entorno como por ejemplo si está conectado a red o cómo actuará en caso de fallo de tensión o al volver a arrancar.

Lenguaje de Lista de instrucciones: Este lenguaje de mnemónicos utiliza básicamente las funciones del álgebra de Boole aunque como resultan insuficientes para todas las tareas que puede realizar el autómeta, se amplían con un juego de instrucciones mayor como son de contadores (CNT), temporizadores (TIM), manipulación de datos (comparar datos CMP, mover datos MOV), gestión de programa (final de programa END, salto a otro bloque JMP) o aritméticas (suma ADD, multiplicación MUL).

Lenguaje de Diagramas de contactos: Es un lenguaje gráfico que expresa relaciones entre señales binarias como una sucesión de contactos en serie y en paralelo. Estos contactos con dos estados (abierto/cerrado) permiten su combinación formando todo tipo de funciones lógicas equivalentes directa e inmediatamente a funciones en lenguaje mnemónico.

Esta manera de representación nació como representación del diagrama de relés de uso común en automatización y contiene contactos y bobinas a modo de relés de salida o internos además de otras funciones que visualmente a modo de ‘caja’ realizan operaciones de lo más diverso, desde contadores a funciones de comunicaciones.

Al igual que el lenguaje de Lista de instrucciones, éste ha tenido que ampliarse con otras instrucciones mediante bloques funcionales complejos que permiten la manipulación de datos, gestión del programa, funciones aritméticas y otras muchas operaciones.

Este lenguaje, muy utilizado por fabricantes norteamericanos y japoneses.

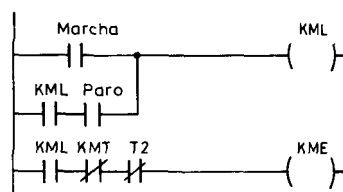
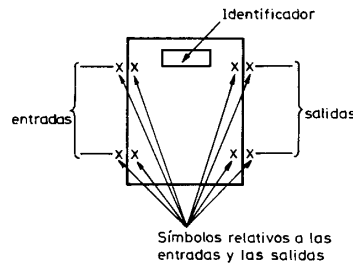


Diagrama de contactos

Lenguaje de Plano de funciones: Este método utiliza los símbolos contenidos en las normas DIN 40700 y DIN 40719.

Estos símbolos corresponden a funciones lógicas, contadores, temporizadores, multiplexores, sumadores, etc., pero al igual que les pasaba a los otros lenguajes, no abarca todas las posibles funciones que puede realizar el autómata. Por eso, se utiliza en diagramas lógicos sencillos con variables booleanas y bloques de tipo contador, temporizador y poco más. Debido a su baja difusión y a la no normalización de las distintas funciones del autómata, cada fabricante ha ampliado este lenguaje con las extensiones al lenguaje de Diagrama de contactos.



Función lógica de Plano de funciones

Intérprete de GRAFCET: Algunos autómatas pueden ser programados mediante GRAFCET.

Lenguajes de alto nivel: Con CPU cada vez más rápidas, más potentes y de mayor capacidad de tratamiento, los autómatas de gamas altas invaden aplicaciones hasta hace bien poco reservadas a los miniordenadores industriales.

Para estas aplicaciones, los lenguajes tradicionales de Lista de instrucciones o Diagrama de contactos resultan ya insuficientes.

Por esta razón, los fabricantes han desarrollado lenguajes de programación próximos a la informática tradicional, con sentencias literales de alto nivel.

Sin embargo, lo más importante es que son lenguajes estructurados, donde es posible la programación por bloques o “procedimientos”, con definición de variables locales o globales, que incluyen estructuras de cálculo repetitivo y condicional como FOR ... TO, REPEAT ... UNTIL X, WHILE X ... y que disponen de instrucciones de manipulación de cadenas de caracteres, muy útiles en aplicaciones de gestión, estadística, etc.

Se pueden encontrar intérpretes o compiladores de C, PASCAL, FORTRAN, BASIC, etc., lo que permite resolver tareas de cálculo científico en alta resolución, clasificaciones de datos, estadísticas, etc., con total facilidad, y con acceso además a módulos y subrutinas específicos ya escritos en estos lenguajes y de uso general en aplicaciones informáticas.

Los lenguajes de alto nivel son posibilidades adicionales al alcance del programador, que puede, si así lo desea, utilizar sólo las formas básicas de contactos/bloques o lista de instrucciones para escribir sus aplicaciones: en otras palabras, los lenguajes avanzados nunca constituyen el lenguaje básico de un autómata o familia de autómatas, papel que queda reservado a la lista de instrucciones o al diagrama de contactos.

Una ventaja adicional del programa en alto nivel es que a él se puede transcribir, casi literalmente, el diagrama de flujos que constituye la primera aproximación a la representación del sistema de control.

2.13. EL GRAFCET

2.13.1. RESUMEN HISTÓRICO

El GRAFCET nació como resultado de los trabajos de la AFCET (Association Francaise pour la Cybernétique Économique et Technique), iniciados en la década de los setenta. En principio se pretendía satisfacer la necesidad de disponer de un método de descripción de procesos, con total independencia de la tecnología, mediante un gráfico funcional que pudiera ser interpretado por no especialistas en automatización. El gráfico funcional permite unificar la forma de descripción del proceso para técnicos de distintos campos, desde el ingeniero de organización o de producción, que define las necesidades del automatismo, pasando por el de diseño, que debe implementar el sistema de control y los accionamientos, hasta el técnico de mantenimiento, que debe cuidar de su funcionalidad o introducir modificaciones en la fase de explotación.

A partir de 1977 y gracias a la colaboración entre AFCET y ADEPA (Agence pour le Développement de la Productique Appliquée) se crearon una serie de útiles metodológicos, entre los que destaca el GEMIU (Guide d'Étude des Modes de Marche et Arrêt), para apoyar el GRAFCET como método no sólo descriptivo, sino como herramienta de diseño.

En 1982 el trabajo fue recogido por un grupo de trabajo de AFNOR, organismo encargado de la normalización en Francia, culminado con la publicación de la Norma NF CO3-1904. Esta norma fue también adoptada por la Comisión Electrotécnica Internacional (IEC) en 1988, con el número IEC-848 y título “Établissement des diagrammes fonctionnels pour systèmes de commande”.

En la actualidad, diversos autómatas programables incorporan algunas instrucciones de programación que permiten introducir directamente el grafo de GRAFCET. En otros casos se dispone de software capaz de compilar un grafo GRAFCET al lenguaje de la máquina, permitiendo en ambos casos una gran flexibilidad y rapidez de diseño, con ventajas sustanciales en las fases de verificación, explotación o eventual modificación del automatismo. A pesar de ello no debe confundirse el GRAFCET con un lenguaje de programación.

El gráfico funcional, complementado con los métodos del álgebra de Boole, permite ir más allá de la simple descripción e interpretación gráfica de un proceso y se ha convertido en una potente herramienta de diseño de sistemas lógicos, con unas reglas bastante simples.

2.13.2. INTRODUCCIÓN AL DISEÑO

Al ir a diseñar con GRAFCET, debemos tener en cuenta la filosofía de este tipo de diseño para poder aplicar correctamente las reglas que rigen este método.

Así se deberá caracterizar el sistema en función de lo que va a hacer y no en función de lo que está constituido. Así será independiente de la tecnología utilizada.

Se deberá dividir el proceso estudiado en macroetapas que representen operaciones distintas y a su vez deberíamos subdividir estas macroetapas en etapas elementales que correspondan a las acciones más simples posibles que se realizan en nuestro sistema.

A continuación se establecerá un gráfico de evolución que indique la secuencia de etapas y las condiciones para pasar de una a otra. Se establece para cada etapa lo que acontece en ella y las relaciones entre salidas y entradas. Finalmente se utiliza para cada etapa un biestable que maneje una variable denominada variable de estado.

A continuación se exponen los principales elementos y reglas utilizados para realizar un GRAFCET.

El modelo GRAFCET está basado en los siguientes elementos y reglas:

2.13.3. ELEMENTOS GRÁFICOS.

Estos son los elementos principales que constituyen los símbolos a partir de los cuales se dibuja el gráfico funcional.

- 1.- Las etapas, que representan cada uno de los estados del sistema. La relación de entradas y salidas dentro de una etapa es puramente combinacional. El símbolo empleado para representar una etapa es un cuadrado con un número o símbolo en su interior que la identifica.
Se denominan etapas iniciales aquéllas en que se posiciona el sistema al iniciarse el proceso por primera vez. Las etapas iniciales se representan por un cuadrado con doble línea.
2. Las líneas de evolución, que unen entre sí las etapas. Las líneas siempre orientadas de arriba hacia abajo, a menos que se represente una flecha en sentido contrario.
3. Las transiciones, que representan las condiciones lógicas necesarias para que finalice una etapa y se inicie la etapa o etapas inmediatamente consecutivas. Gráficamente se representan las transiciones por una línea cruzada sobre las líneas de evolución.
4. Los reenvíos son símbolos en forma de flecha que indican la procedencia o destino de las líneas de evolución. Las flechas de reenvío permiten fraccionar un gráfico o evitan dibujar líneas de evolución con excesivos cruces.
5. Dos líneas de evolución que se crucen debe interpretarse, en principio, que no están unidas.
6. Cuando se recorre el gráfico de evolución, por cualquier camino posible, deben alternarse siempre una etapa y una transición. La regla básica de sintaxis del GRAFCET es que entre dos etapas debe existir una y sólo una condición de transición.

Es posible representar qué etapas están activas y cuáles están inactivas. El simbolismo utilizado para ello consiste en marcar con un punto las etapas activas.

2.13.4. MENSAJES DE INTERPRETACIÓN.

Estos mensajes pueden ser textos, símbolos o ecuaciones lógicas asociados a las etapas o transiciones para indicar la actividad desarrollada o las relaciones entre variables del sistema que deben cumplirse. Pueden distinguirse dos tipos de mensajes:

1. Mensajes de acción asociados a cada etapa. Indican cuál es la actividad a desarrollar en dicha etapa cuando esté activa bien sea en forma de texto o en forma de ecuaciones lógicas que indiquen la relación salidas/entradas.
2. Mensajes de receptividad asociados a cada transición. Estos mensajes indican las condiciones lógicas necesarias y suficientes para pasar de cada etapa a la consecutiva o consecutivas.

2.13.5. REGLAS DE EVOLUCIÓN.

Estas son las principales reglas que permiten definir e interpretar de forma unívoca el comportamiento dinámico del sistema.

1. Cada etapa tiene asociada una variable de estado X_i de tipo bit.
2. Se distinguen dos posibles estados de una etapa: activa o inactiva.
3. Durante la evolución normal del proceso, una etapa no inicial se activará cuando esté activada la etapa anterior y se cumplan las condiciones de transición entre ambas.
4. Cualquier etapa se desactiva cuando se cumplan las condiciones de transición a la siguiente o siguientes y dicha transición se haya efectuado.
5. Una transición puede encontrarse en una de las cuatro situaciones siguientes:
 - No validada: La etapa o etapas inmediatamente anteriores o siguientes no están activas.
 - Validada: La etapa o etapas inmediatamente anteriores están activas, pero no se cumple la condición lógica de transición.
 - Franqueable: La etapa o etapas inmediatamente anteriores están activas y se cumple la condición lógica de transición.
 - Franqueada: Se ha activado la etapa o etapas inmediatamente siguientes y se han desactivado la etapa o etapas inmediatamente anteriores.
6. Sólo se podrá franquear una transición si ésta está previamente validada.
7. Toda transición franqueable será inmediatamente franqueada.
8. Si hay varias transiciones franqueables simultáneamente, serán franqueadas simultáneamente.
9. El franqueo de una transición implica automáticamente la desactivación de todas las etapas inmediatamente anteriores.
10. Si una etapa debe ser simultáneamente activada y desactivada, dicha etapa permanecerá activada.
11. El gráfico de evolución debe ser siempre cerrado, sin dejar ningún camino abierto.

En el GRAFCET existen dos estructuras como son la divergencia en Y y en O que permiten que un GRAFCET evolucione por diversos caminos a la vez o bajo determinadas condiciones.