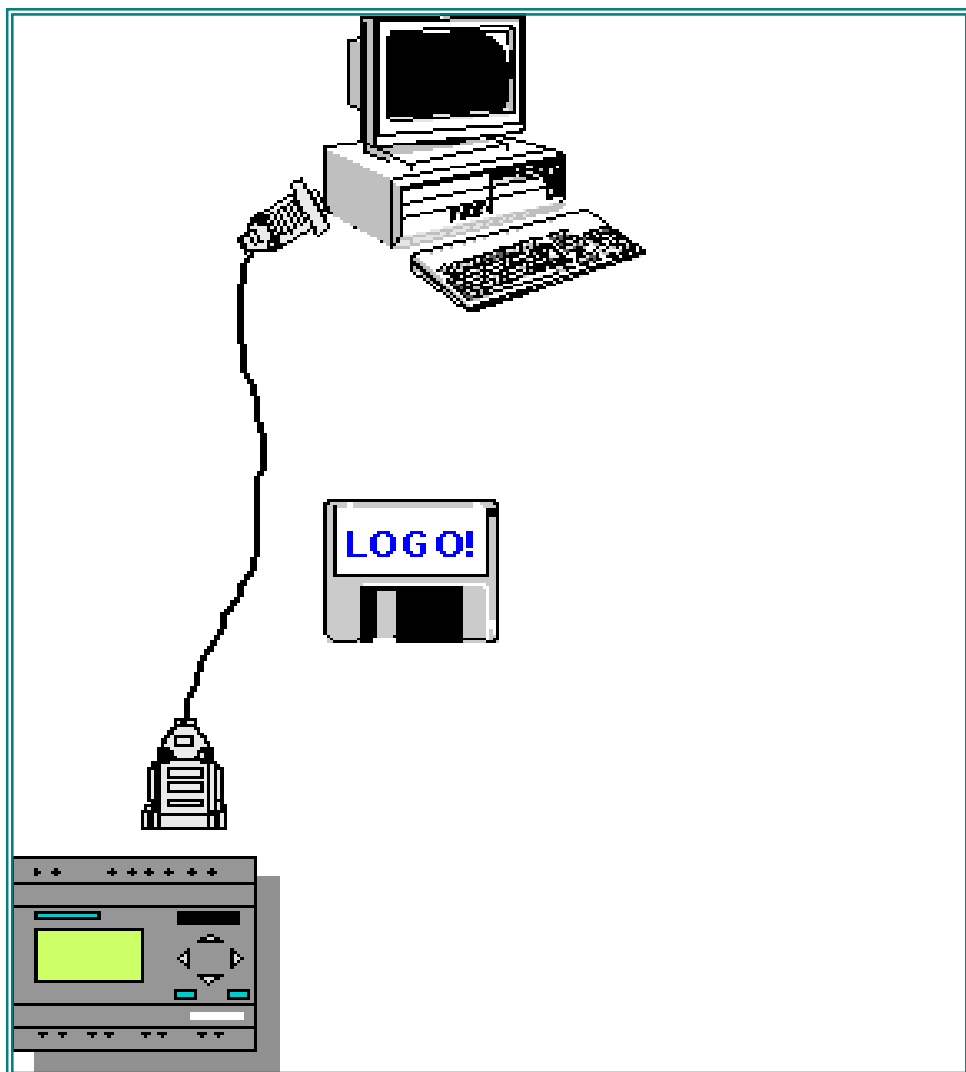


CAPÍTULO III

INTERACCIÓN ENTRE UN LENGUAJE DE ALTO NIVEL Y PLC



3.1. PERIFÉRICOS DE COMUNICACIÓN, CARGADORES DE PROGRAMA Y UTILIZACIÓN DE LA PC PARA ELABORAR Y TRANSFERIR LAS INSTRUCCIONES

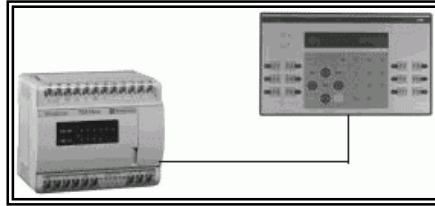


Figura 3.1. - Conexión de un visualizador a un autómata o PLC

La comunicación más usada entre el PLC y sus periféricos (terminales, consolas, teclados, impresoras) es la del tipo serial asincrónico. Este modo de comunicación permite el intercambio de caracteres alfanuméricos (generalmente en código ASCII) compuesto de una secuencia de bits transmitidos uno detrás del otro. La velocidad de transmisión se expresa en baudios (bits /seg.). Las interfaces se rigen por normas estándar como RS 232C.

Los periféricos no intervienen directamente en el funcionamiento del autómata, pero sin embargo facilitan la labor del operario. [WWW. 11]

Los más utilizados son:

- ✍ Impresoras
- ✍ Grabadoras a cassettes.
- ✍ Cartuchos de memoria EEPROM.
- ✍ Visualizadores y paneles de operación.

El autómata programable (PLC), en la mayoría de los casos, puede ser ampliable. Las ampliaciones abarcan un gran abanico de posibilidades, que van desde las redes internas (LAN, etc.), módulos auxiliares de E/S, memoria adicional, hasta la conexión con otros autómatas del mismo modelo. [WWW. 12]

Cada fabricante facilita las posibilidades de ampliación de sus modelos, los cuales pueden variar incluso entre modelos de la misma serie.

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

- ✍ Unidad de programación
- ✍ Consola de programación
- ✍ PC

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.

Permite la visualización de bloques completos de programa, con identificación simbólica de las variables y menús de ayuda “on line”, los programas se confeccionan sin conexión entre la unidad de programación y el autómata, y una vez finalizados se transfieren a módulos de memoria independientes enchufables después al autómata, o se mantienen en

la memoria del terminal de programación para su transferencia directa a la memoria del autómata, posteriormente.



Figura 3.2. - Terminal de programación portátil

Consola de programación

Las consolas de programación son pequeños dispositivos de bolsillo que permiten la programación, ajuste y diagnóstico del autómata, con un visualizador (<<display>>) apto para observar una (o pocas) líneas del programa, y un teclado alfanumérico.

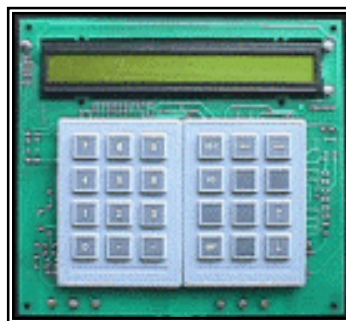


Figura 3.3. - Terminal usuario UNI-USERS que permite la comunicación entre el usuario y una PC o PLC

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.

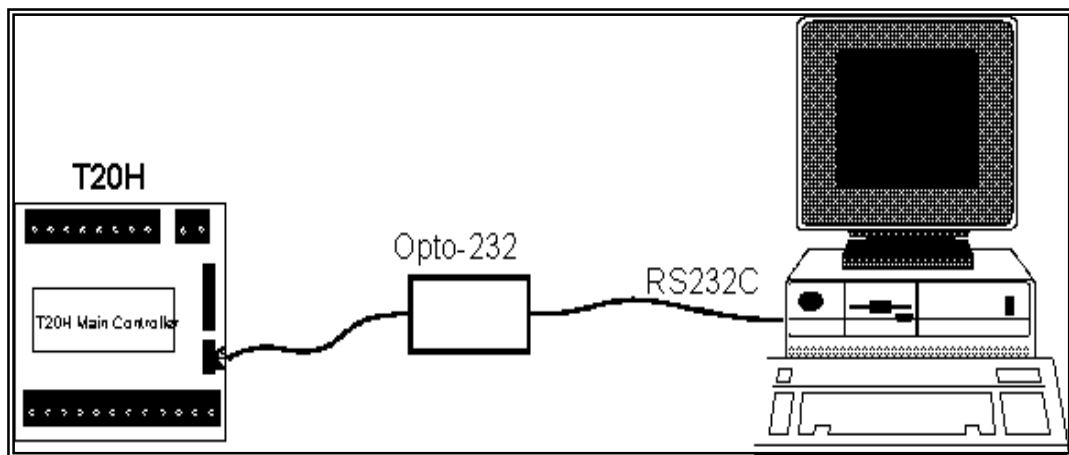




Figura 3.4. -Conexión a una PC el PLC SCM T20H

EQUIPOS DE PROGRAMACIÓN DEL PLC

	MODELO	DESCRIPCIÓN
UNIDAD DE PROGRAMACIÓN PORTATIL	C1	<ul style="list-style-type: none">▪ Visualización 4 líneas de 20 caracteres.▪ Guía para el operador▪ Salvaguarda interna del programa▪ Lenguaje literal, esquemas de relés y Grafcet.▪ Alimentación tomada desde el PLC.
		
UNIDAD DE PROGRAMACIÓN PORTATIL	C2	<ul style="list-style-type: none">▪ Visualización 4 líneas de 20 caracteres.▪ Guía para el operador▪ Salvaguarda y almacenamiento de programas en tarjetas de memoria RAM▪ Lenguaje literal, esquemas de relés y Grafcet.▪ Alimentación automática.▪ Puerto de salida para impresora serie
		

EQUIPOS DE PROGRAMACIÓN DEL PLC

	MODELO	DESCRIPCIÓN
CONSOLA DE PROGRAMACIÓN O TERMINAL	<p><u>Omron NT2S</u></p> 	<ul style="list-style-type: none"> ▪ Display de 2 líneas por 16 caracteres cada una. ▪ Admite la totalidad de protocolos de comunicación para todos los PLC's disponibles ▪ Tiene protección para trabajar en ambientes severos.
CONSOLA DE PROGRAMACIÓN O TERMINAL	<p>TERMINAL DE PROGRAMACIÓN 0054</p> 	<ul style="list-style-type: none"> • Display alfanumérico de 4 líneas por 20 caracteres. • No requiere alimentación auxiliar ni baterías. • Programación guiada por menús. • Reducidas dimensiones y peso. • Utiliza el mismo puerto del MODEM o PC. • Almacena los parámetros para su transferencia a otros controladores. • Teclas robustas y con fuerte sensación táctil. • Interfaz RS485.

EQUIPOS DE PROGRAMACIÓN DEL PLC

	MODELO	DESCRIPCIÓN
PROGRAMACIÓN DESDE PC	<p><u>SOFTWARE RPX C3 BAJO WINDOWS</u></p>  <p>The image shows a software box for 'RPX C3' with a CD-ROM and a USB cable. The box is black with white and red text.</p>	<ul style="list-style-type: none">▪ Lenguaje español▪ Sencillo, versátil, potente, profesional.▪ El software RPX C3 permite programar y documentar proyectos con PLC.▪ Permite crear, renombrar, suprimir, archivar, restituir.
PROGRAMACIÓN DESDE PC	<p><u>OMRON CX-Programmer 3.9</u></p>  <p>The image shows a software box and a CD-ROM for 'Omron CX-Programmer 3.9'. The box is green and blue with the Omron logo.</p>	<ul style="list-style-type: none">▪ Potente herramienta de programación para PLC.▪ Entorno eficaz de configuración, desarrollo y depuración de aplicaciones.▪ Optimización de tiempo en la fase de desarrollo y depuración de proyectos de automatización.

3.2. METODOLOGÍA DE PROGRAMACIÓN PARA LOS PLC

Existen distintos métodos más o menos standard de programación de PLC.

Uno de estos métodos, es la utilización de *códigos de operación (opcode)* en la forma de listado que le indica al PLC la secuencia exacta de operaciones a realizar, estos símbolos corresponden directamente a instrucciones de máquina.

Habitualmente estas operaciones son del tipo: "examine el estado de la entrada n", "active la salida m"; codificadas con siglas conocidas con el nombre de MNEMONICOS, del tipo LOD N, OUT M, etc, es decir, que un método para mejorar el manejo de las instrucciones para el programador consiste en asignar un nombre a cada instrucción, por ejemplo a la instrucción de decrementar se le nombra con "DEC" y a la de cargar un dato se nombra como "LD", a los nombres de las instrucciones se les conoce como "MNEMÓNICOS", se escogen de tal manera que orienten en forma fácil al programador.

Es más fácil recordar que INC A es el mnemónico de la instrucción para incrementar en uno el contenido del registro, que 3Ch es el código operacional correspondiente a la misma instrucción, cada fabricante de microprocesadores proporciona una serie de mnemónicos para el conjunto de instrucciones de cada microprocesador.

El lenguaje en mnemónico es similar al lenguaje ensamblador, es decir el lenguaje ensamblador es la primera abstracción del lenguaje de máquina, consistente en asociar a los opcodes palabras claves que faciliten su uso por parte del programador.

Las instrucciones en lenguaje ensamblador se dividen en campos de la siguiente forma:

ETIQUETA: Código-Operación Operandos # Comentarios

- El campo del código de operación, es el único que nunca puede estar vacío, su contenido puede ser un mnemónico o una directiva del ensamblador, también llamada pseudoinstrucción.
- El campo operandos, puede contener una dirección.
- El campo comentarios y etiquetas, son opcionales. El programador los utiliza para hacer el programa más fácil de leer y de utilizar.

La separación entre campos usualmente se realiza:

- Utilizando dos punto (:) entre la etiqueta y el código de operación.
- Utilizando comas (,) entre operandos.
- Utilizando 3 antes de una línea de comentarios .

Etiquetas

El campo de etiquetas, es el primero de una instrucción de ensamblador, puede ser un blanco. Si existe, dicho campo no debe sobrepasar de los ocho caracteres. El ensamblador define la etiqueta como el valor de la dirección de memoria en la cual se ubicará el primer byte del código objeto generado por esa instrucción. Después de una etiqueta siempre debe haber una instrucción.

La ventaja que se obtiene de la utilización de etiquetas, es que son fáciles de recordar y pueden cambiarse, o corregirse para introducir código.

Hay que tener cuidado al utilizar etiquetas y evitar que sean iguales a los códigos de operación, no utilizar caracteres especiales en las mismas, no empezarlas con números, y sobre todo darles un significado acorde con la función que realizan.

Códigos de operación

En este campo podemos tener o bien un mnemónico, una instrucción del PLC para el cual se escribe el programa, o una directiva, una instrucción especial del programa ensamblador utilizado.

La principal tarea del ensamblador es la traducción de los mnemónicos, en sus códigos binarios equivalentes, los cuales serán entendidos por el PLC. Para ello el programa ensamblador de una tabla con todos los posibles mnemónicos del micro, es decir, conoce todo el conjunto de instrucciones del micro y sus equivalentes en código máquina.

Algunas instrucciones del lenguaje ensamblador, no son traducidas directamente en instrucciones en código máquina. Estas instrucciones son las directivas. Una de sus funciones es indicarle al programa ensamblador cómo desea realizar el proceso de ensamblado, otra función es la de reservar áreas de memoria, definir zonas de almacenamiento, etc.

Operando y direcciones

Los números que se especifican en los operandos son tratados en notación decimal a menos que se le anteponga: Ox. Los nombres de etiquetas que aparezcan como operandos en instrucciones, con tratadas como direcciones.

Comentarios

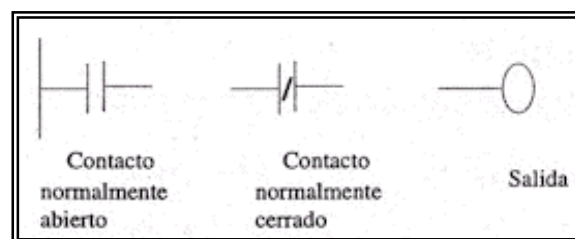
Los ensambladores, permiten la colocación de comentarios en el programa fuente. No tiene efecto sobre el código objeto, pero son útiles para leer, comprender y documentar un programa. Los comentarios deben utilizarse con precaución, deben describir la tarea a realizar, al mismo tiempo ser concisos, claros y concretos. Se les antepone el símbolo #.

Otro método consiste en la utilización de ***símbolos gráficos*** que representan determinadas operaciones básicas del PLC. La principal ventaja de este sistema es que está estandarizado y que no depende de la marca de PLC que se está programando. Además, existen programas para computadora personal que permiten construir los programas de PLC de forma gráfica, por manipulación de estos símbolos.

Finalmente, el método de programación LADDER, que dada su sencillez y similitud con un diagrama eléctrico es el más difundido.

El lenguaje de contactos es un lenguaje gráfico, derivado del lenguaje de relés, que mediante símbolos representa contactos, solenoides, etc. Su principal ventaja es que los símbolos básicos están normalizados y son empleados por todos los fabricantes. [WWW.13]

Los símbolos básicos son:



Los elementos básicos que configuran la función se representan entre dos líneas verticales que simbolizan las líneas de alimentación.

Para las líneas de función más complejas como temporizadores, registros de desplazamiento, etc, se emplea el formato de bloques. Estos no están formalizados, aunque guardan una gran similitud entre sí para distintos fabricantes y resultan mucho

más expresivos que si se utiliza para el mismo fin el lenguaje en lista de instrucciones o mnemónico.

Un programa en esquema de contactos, la constituyen una serie de ramas de contactos.

Una rama está compuesta de una serie de contactos, conectados en serie o en paralelo que dan origen a una salida que bien puede ser una bobina o una función especial.

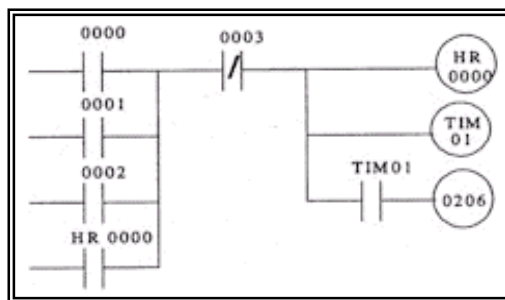


Figura 3.5. - Composición de una rama

El flujo de la señal va de izquierda a derecha y de arriba abajo.

A una rama de circuito en esquema de contactos le corresponde una secuencia de instrucciones en forma mnemónica.

Una bobina no puede venir conectada directamente de la barra de inicio. En tal caso es necesario interponer un contacto siempre cerrado.

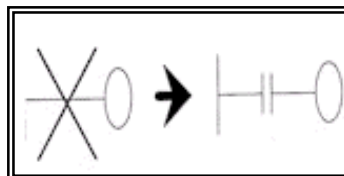


Figura 3.6. - Conexión de una bobina

A la derecha de una bobina no es posible programar ningún contacto.

El número de contactos posibles en serie o en paralelo es prácticamente ilimitado, es posible colocar en paralelo 2 o más bobinas.

Finalmente, el método de **programación LADDER**, que dada su sencillez y similitud con un diagrama eléctrico es el más difundido.

Hay 2 tipos de lenguajes en la programación del PLC:

1. - Lenguaje de Bajo Nivel o Básico
2. - Lenguaje de Alto Nivel

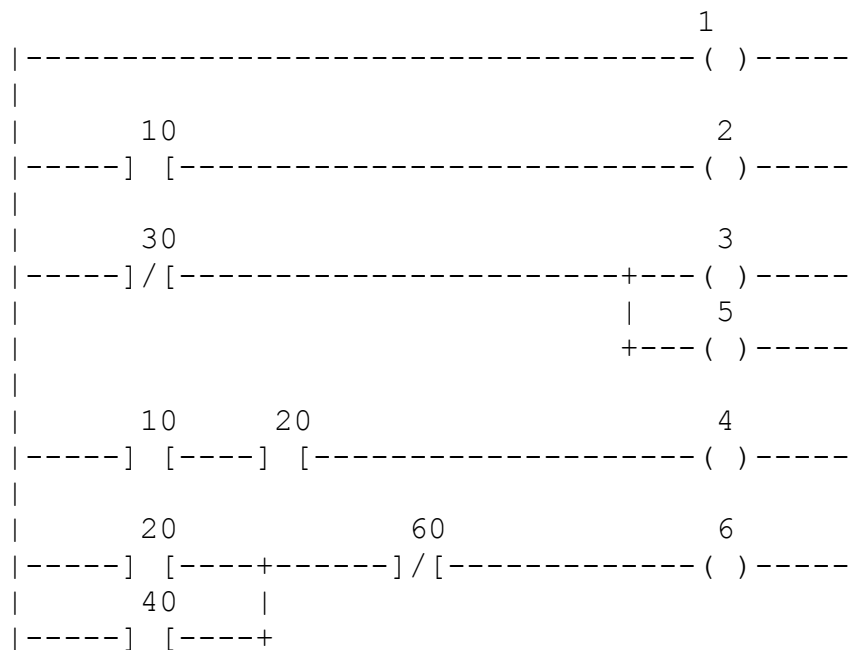
Lenguaje De Bajo Nivel

- Diagrama Ladder de Escalera
- Lenguaje Booleano

Diagrama Ladder De Escalera

El diagrama ladder de escalera o de contactos está constituido por varias líneas horizontales que contienen símbolos gráficos de prueba (“Contactos”) y de acción (“Bobinas”), que representan la secuencia lógica de operaciones que debe realizar el PLC. [www.14]

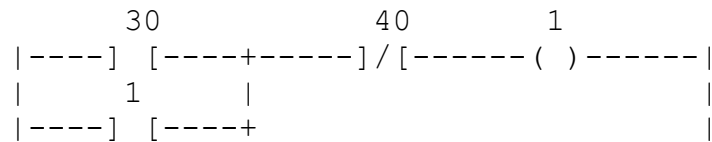
El diagrama Ladder, es el lenguaje de programación de PLC más difundido, es una metodología de programación gráfica. Un ejemplo de este diagrama se muestra enseguida:



Se debe considerar primero el diagrama como circuito eléctrico. Aquí la línea vertical a la izquierda representa un conductor con tensión, y la línea vertical a la derecha representa tierra o masa. Los corchetes --] [-- representan contactos normalmente abiertos y los símbolos --]/[-- representan contactos normalmente cerrados. Los paréntesis --()-- representan cargas, por lo general bobinas de relés, lámparas indicadoras, etc. Con esta simbología, la salida 1, en el primer renglón o RUNG, está permanentemente conectada, ya que esta alimentada entre tensión y masa independientemente de cualquier contacto. En el segundo rung, la carga 2 se activará solo cuando el contacto 10 esté cerrado (lo que para un contacto normalmente abierto como 10 significa que debe estar actuado), ya que solo así podrá circular corriente. En el tercer rung, las cargas 3 y 5 (conectadas en paralelo) se activarán cuando el contacto 30 no esté actuado (30 es un contacto normalmente cerrado, no debe estar actuado si queremos que por él circule corriente). El cuarto rung muestra la conexión en serie de dos contactos (los llamados 10 y 20) para que solo se conecte la carga cuando ambos se encuentren cerrados. En el último rung, la carga 6 se activará si el contacto 60 no está actuado y si además están cerrados los contactos 20 ó 40 (o ambos). Cuando las cargas son

bobinas de relés, sus contactos reciben el nombre de la carga.

Ejemplo.

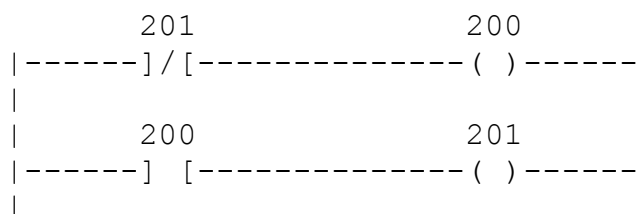


La carga 1 se activará cuando estén actuados los contactos 30 y/o 1, y cuando el contacto 40 esté sin actuar.

Supongamos que la carga no se encuentra activada, por lo que el contacto 1 está abierto y el contacto 40 esta cerrado. En estas condiciones se actúa momentáneamente el contacto 30, por lo que la carga 1 queda energizada y el contacto auxiliar 1 cerrado. Ahora, si se libera el contacto 30, sigue existiendo un camino para la corriente, por lo que una vez conectada la carga solo se la puede desconectar actuando el contacto 40.

Este es el esquema eléctrico de un sistema de encendido y apagado con un pulsador de arranque y otro de parada.


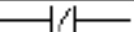
Los montajes de este tipo pueden presentar comportamientos inesperados. Obsérvese el diagrama siguiente:


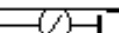
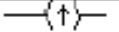
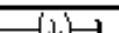
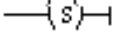
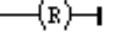


Si suponemos que inicialmente ninguna de las salidas está activada, ya que acabamos de alimentar al sistema, el contacto 201 permite el paso de corriente y se activa 200. Esto permite que se active la salida 201, el segundo rung. Al suceder esto, se interrumpe la alimentación de la bobina 200, con la consecuente liberación de su contacto auxiliar y la desconexión de la bobina 201. A partir de este momento el ciclo se repite indefinidamente.

Con este tipo de diagramas se describe normalmente la operación eléctrica de distinto tipo de máquinas, y puede utilizarse para sintetizar un sistema de control y, con las herramientas de software adecuadas, realizar la programación del PLC.

Los contactos y bobinas básicas que se utilizan son:

Tipo de Contacto	Visualización	Pasa corriente cuando...
Normalmente abierto		La referencia está ON
Normalmente cerrado		La referencia está OFF

Tipo de Bobina	Visualización	¿Corriente a la bobina?	Resultado
Normalmente abierta		Si	Referencia ON
		No	Referencia OFF
Inversa		Si	Referencia OFF
		No	Referencia ON
Transición Positiva		No → Si	Ref. ON durante un barrido
Transición Negativa		Si → No	Ref. OFF durante un barrido
Set		Si	Ref. ON hasta que se apague con "R"
		No	Sigue igual
Reset		Si	Ref. OFF hasta que se active con "S"
		No	Sigue igual

Cada fabricante ha puesto en el mercado su propia simbología, pero casi todos tienden a una similar a la indicada respetando normas internacionales.

Lenguaje Booleano

Se basa en los principios del álgebra booleana, se puede programar introduciendo operadores booleanos y generalmente se ingresa como mnemónicos en la programación AND, OR, NOT, NAND, NOR, etc, que son símbolos comúnmente usando para proveer un lenguaje de programación muy similar al utilizado en la programación de microprocesadores y con ello hacer más rápido un programa o, más compacto.

- OR función suma lógica.
- AND función producto lógico.
- LOD leer variable inicial.
- OUT enviar resultado a salida.
- OR LOD coloca bloque en paralelo.
- AND LOD coloca bloque en serie.

En una operación normal el autómata utiliza algunas otras instrucciones del lenguaje booleano que le permite manejar elementos de común automatización y que son las siguientes instrucciones secuenciales:

- TIM definir un temporizador.
- CNT definir un contador.
- SET activar una variable binaria (unidad de memoria).
- RST desactivar una variable binaria .

También existen otros tipos de instrucciones como las siguientes:

- DD sumar.
- BB restar.
- MUL multiplicar.
- DIV dividir.
- CMP comparar variables digitales.
- FR rotaciones de bits (variables de binarias).
- HIFT rotaciones de palabras (variables digitales).
- MOV transferencias de datos.
- CD/BIN conversiones de códigos numéricos, etc.
- END fin de programa.
- JMP salto de bloque de programa.
- MCS habilitación de bloque de programa.
- MCS habilitación de bloque de programa.
- JMPSUB salto a subrutina, etc.

Al lenguaje restante que no puede llamarse ya booleano después de ampliarlo con estas extensiones se le denomina de *lista de instrucciones* (“Instruction List”).

Algunos fabricantes amplían las capacidades de programación de sus autómatas de gama baja con estas instrucciones avanzadas que serían de más lógica aplicación en autómatas de superiores prestaciones. [www. 15]

Generalmente este tipo de lenguaje de programación se halla relacionado con los programadores manuales de los PLC.

Lenguaje De Alto Nivel

- Diagrama de Bloques Funcionales
- Lenguaje tipo Computadora

Diagrama De Funciones – Bloques

El diagrama de funciones (Function Block Diagram o FBD) es lenguaje gráfico que permite programar elementos que aparecen como bloques para ser cableados entre sí de forma análoga al esquema de un circuito.

El diagrama de bloques funcionales (FBD) es muy común en aplicaciones que implican flujo de información o datos entre componentes de control. Las funciones y bloques funcionales aparecen como circuitos integrados y requiere manejo de tablas de verdad y simplificación de funciones booleanas para su empleo, también se usa para construir procedimientos complejos a partir de una librería de funciones. Las librerías estándar, tales como matemática o control, pueden ser combinadas con librerías de funciones personalizadas, tales como la llamada del MODEM, controladores PID, etc.

Así como las funciones ADD, SQUARE ROOT, SIN, COS, GREATER THAN, etc, se puede crear funciones como la siguiente:

```
FUNCION SIMPLE_FUN : REAL  
VAR_INPUT  
    A, B : REAL;  
    C : REAL := 1.0;  
END_VAR  
SIMPLE_FUN := A*B/C;  
END FUNCTION
```

Una vez definida la función se podrá utilizar una y otra vez, en el mismo programa, en diferentes programas o en cualquier otro proyecto.

Lo mismo es válido para los bloques funcionales, es decir que también se pueden crear bloques funcionales propios y añadirlos a la librería de bloques funcionales. Todos los bloques funcionales son altamente reutilizables en el mismo programa, nuevos programas o distintos proyectos. Esta reutilizabilidad incrementa la eficiencia, y reduce el número de errores.

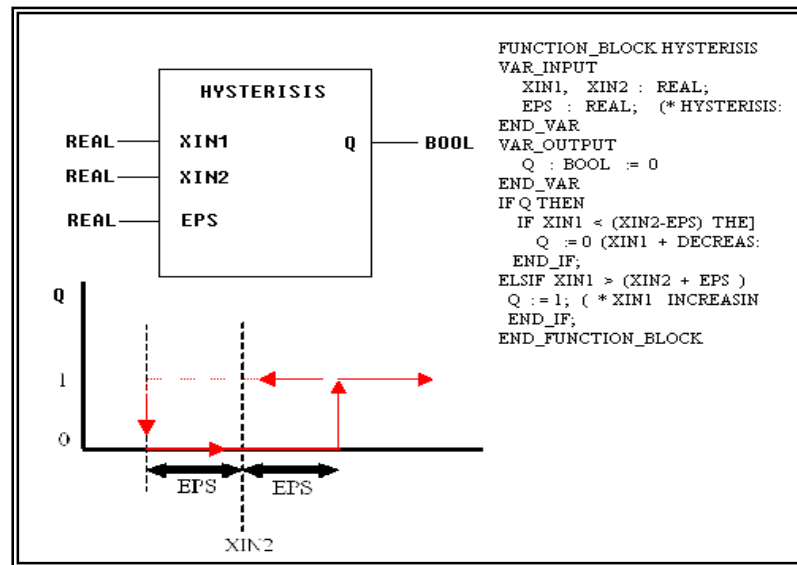


Figura 3.7. - Bloque Funcional

El Bloque Funcional arriba mostrado, (en el lado izquierdo), está representado aquí en el lenguaje de programación de Diagramas de Bloques Funcionales. El Bloque Funcional tiene el nombre de Histéresis, Tiene tres entradas a la izquierda, denominadas XIN1, XIN2 y EPS, todas del tipo de variable real. Tiene una salida, a la derecha, llamada Q, del tipo variable booleana (BOOL).

Internamente, el bloque funcional contiene un código de programa, como el mostrado en el lado derecho. La primera parte configura los datos de la estructura, la segunda parte con el algoritmo, usa las entradas, realiza los cálculos, y modifica las salidas. El algoritmo está oculto para el usuario del Bloque Funcional, quien sólo ve la

funcionalidad del Bloque como se muestra en la izquierda. Esto crea un diferente nivel de acceso, mostrando la encapsulación.

Lenguaje Tipo Computadora

Utilizan instrucciones dadas en Inglés, usualmente muy similares a lenguajes de programación conocidos como por ejemplo: el BASIC y el C de amplia difusión en computadoras personales, para hacer cada vez más amigable la programación.

Este es el más usado en la programación del PLC.

Las instrucciones son líneas de texto que utilizan palabras o símbolos reservados (SET, AND, FOR, etc.). Las operaciones se definen por los símbolos matemáticos habituales (+, *, <, etc.), y se dispone de funciones trigonométricas, logarítmicas y de manipulación de variables complejas (Cos, pi, real, img, etc.).

Sin embargo, lo que distingue realmente estos lenguajes avanzados de las listas de instrucciones ampliadas son las tres características siguientes:

- Son lenguajes estructurados, donde es posible la programación por bloques o “procedimientos”, con definición de variables locales o globales.
- Incluyen estructuras de cálculo repetitivo y condicional tales como:
 - FOR . . . TO
 - REPEAT . . . UNTIL X
 - WHILE X . . .
 - IF . . . THEN . . . ELSE
- Disponen de instrucciones de manipulación de cadenas de caracteres, muy útiles en aplicaciones de gestión, estadística, etc.

Un programa escrito en alto nivel necesita para su edición de una unidad de programación avanzada o de un software de desarrollo de programas que corra sobre PC.

Adicionalmente, es frecuente que el empleo de estos lenguajes estructurados obligue además a utilizar no solo una unidad de programación tipo PC, sino incluso una CPU especial en el autómata (coprocesador), capaz de interpretar y ejecutar las nuevas instrucciones.

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. Esta ventaja, que evita pasar a algún otro modelo o transcribir este inicial a otro lenguaje, supone sin duda un importante ahorro de tiempo en la puesta en marcha de aplicación.

Metodología para programar un PLC

La metodología a seguir para la programación de un controlador lógico programable consta de los siguientes pasos: [LIB. 1]

1 . Estructurar el proceso en secciones independientes entre sí.

2 . Describir las funciones de cada proceso, para esto se debe tomar en cuenta

lo siguiente:

- Entradas y Salidas
- Describir el funcionamiento
- Condiciones que deben cumplirse para la operación de los actuadores (electro válvulas, motores, accionadores, sensores, etc).

- 3** . Diseñar los circuitos de seguridad para la instalación de los cables de interconexión. En este caso es necesario considerar condiciones de operación normal y de falla, tanto en condiciones de cortocircuito, sobrecarga o sobrevoltaje, tomando en cuenta la seguridad del equipo y del personal que operará el sistema.

- 4** . Distribuir adecuadamente las estaciones de control del operador, considerando la disposición mecánica de los CPU's, de los módulos de ampliación y lógicamente de los equipos que se controlarán.

- 5** . Elaborar una lista de nombres asociados a las entradas, salidas y de todos los elementos que intervienen en el sistema.

3.3. ANÁLISIS Y EVALUACIÓN DE LAS SEÑALES DE LOS SENSORES EXTERNOS

Los sensores son elementos físicos que pertenecen a un tipo de dispositivo llamado *transductor*. Los transductores son elementos capaces de transformar una variable física en otra diferente, de tal forma los sensores son un tipo concreto de transductores que se caracterizan porque son usados para medir la variable transformada.

La magnitud física que es empleada por los sensores como resultado suele ser la tensión eléctrica, debido a la facilidad del trabajo con ella.

Desde el punto de vista de la forma de la variable de salida, los sensores se clasifican en dos grupos: *analógicos*, en los que la señal de salida es una señal continua, analógica; y *digitales*, que transforman la variable medida en una señal digital, a modo de pulsos o bits.

En la actualidad los sensores más empleados son los digitales, debido sobre todo a la compatibilidad de su uso con los ordenadores.

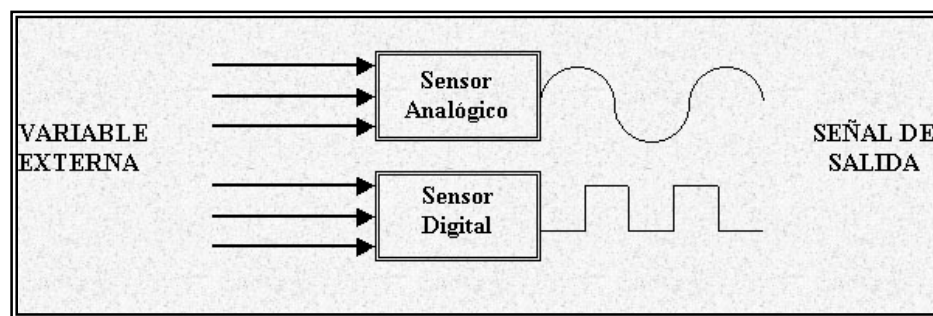


Figura 3.8. -Señales de sensores analógicos y digitales

Señales Analógicas: El nivel o la amplitud de las señales analógicas puede variar de forma continua con el tiempo. Es decir, la amplitud de la señal puede adoptar un valor cualquiera, dentro de las posibilidades que ofrece un rango prefijado por unos valores máximo y mínimo.

En la siguiente figura se muestra una señal muy común, denominada senoide u onda senoidal, a causa de que su forma ondulada queda descrita por la función matemática correspondiente al seno. Es lo que se denomina una señal periódica, ya que la forma se repite cada T segundos. El intervalo de tiempo T , en segundos, es lo que se llama el periodo de la señal. Si se calcula el inverso del período T , se obtiene entonces la frecuencia de la señal. De forma análoga, si se conoce la frecuencia de la señal, calculando su inverso puede obtenerse el período. así pues:

$$f = 1 / T \quad \text{y} \quad T = 1 / f$$

T = PERIODO f = FRECUENCIA

La frecuencia indica cuantas veces se repite la señal por cada segundo, es decir, cuantos ciclos hay por segundo (cps). La unidad de frecuencia es el hercio (Hz), y un hercio equivale a un cps. No todas las señales continuas son periódicas, ni todas las señales periódicas son sinusoides, pero todas pueden describirse en términos de sinusoides. [WWW. 16]

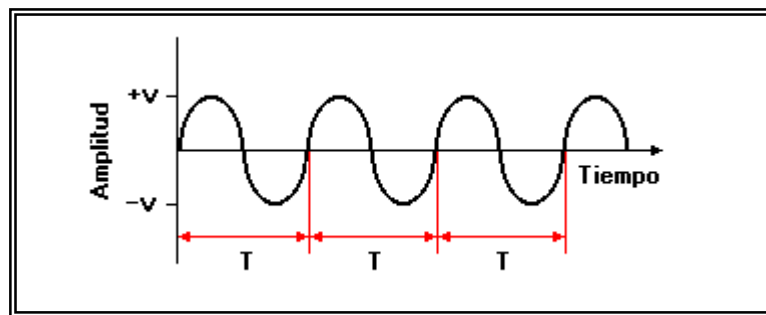


Figura 3.9. - Señal senoide u onda senoidal

La mayoría de los sensores generan señales de tipo analógico. La frecuencia de la señal proporcionada por el sensor puede cubrir una banda de frecuencias que van desde un hercio hasta miles de hercios, aunque, lo normal es que la frecuencia de dicha señal sea de pocos hercios.

Señales Digitales: Las señales digitales que se utilizan en los sistemas de control adoptan uno de dos posibles niveles, es decir, son señales binarias. Se puede representar este tipo de señales de forma numérica, asignando a uno de los niveles el valor "1", y al otro el valor "0", los dos valores permitidos en el sistema binario de numeración (base 2).

Los circuitos digitales funcionan con niveles de tensión, pero los niveles concretos que representan al 0 y al 1 en un sistema pueden ser diferentes de las tensiones que lo hacen en otro sistema distinto. Por ejemplo, un sistema puede utilizar un nivel de +0.4 voltios para representar al 0, y +2.4 voltios para el 1, Sin embargo, otro sistema diferente puede hacer uso de -0.4 voltios para representar al 0, y de +0.4 voltios para el 1.

Al igual que las señales analógicas, las señales digitales tienen un periodo y una frecuencia, pero el periodo en este caso se mide de manera diferente. Tal como se puede ver en la siguiente figura, el periodo T es el tiempo durante el cual la señal se encuentra en el nivel correspondiente a un dígito binario (bit). Este tiempo varía de un sistema a otro, pero, para un sistema concreto, debe encontrarse dentro de los límites especificados. La frecuencia es la inversa del periodo ($1 / T$), y se mide en bits por segundo (bps). Dado que un bit únicamente puede representar dos posibles estados, tales como ON/OFF, SI/NO, o A/B.

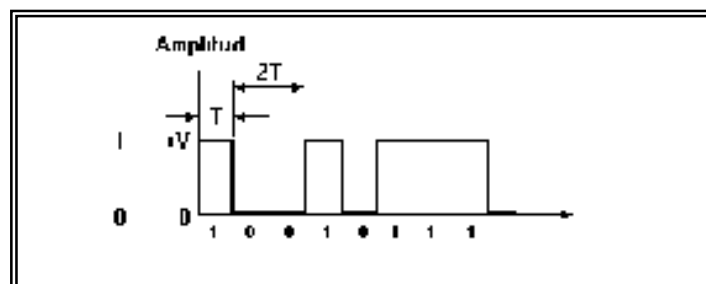


Figura 3.10.- Señal digital

Conversión analógica/digital

Los ordenadores digitales requieren que sus entradas sean señales digitales en forma binaria, es decir, que únicamente pueden adoptar dos niveles de tensión, y generan asimismo señales de salida digitales. Desafortunadamente, la mayor parte de los sistemas que se van a controlar se comportan como sistemas analógicos, y muchos de los sensores generan señales de entradas analógicas, el mismo tipo de señales que necesitan los

dispositivos de actuación. En consecuencia, es necesario el uso de convertidores analógicos-digitales (A/D) y digital-analógicos(D/A), entre el ordenador digital y el sistema controlado.

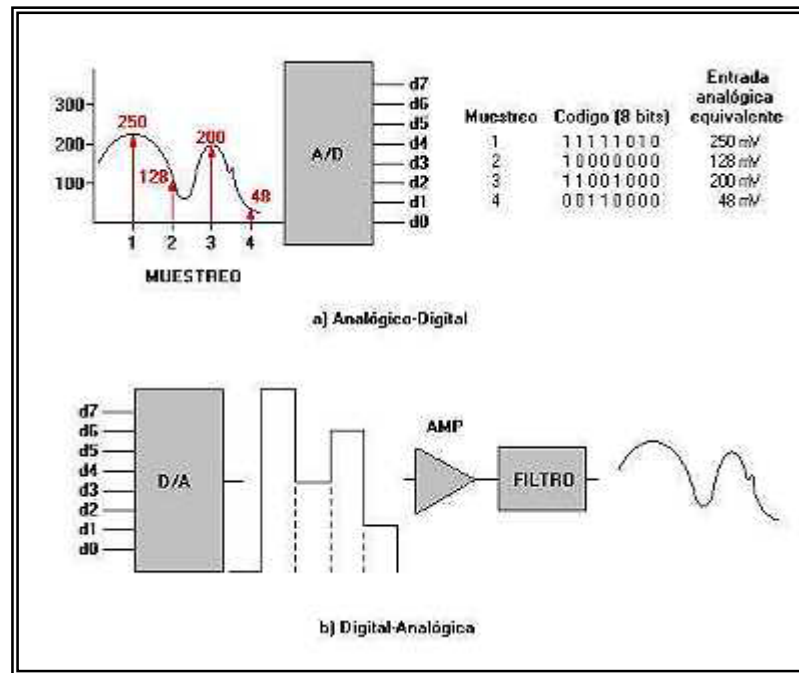

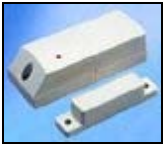
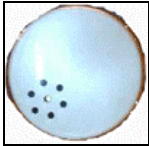


Figura 3.11.- Conversión de señal analógica/digital y digital/analogica

Un convertidor A/D funciona de la siguiente manera: la entrada es una señal que varía de forma continua, y que se muestra en determinados instantes, según la frecuencia de muestreo especificada. El valor del voltaje muestreado se convierte a un código digital (de 8, 12 ó 16 bits según el sistema) de forma que la combinación de ceros y unos que la representa es única. Todos los bits que constituyen el código salen al mismo tiempo (en paralelo) del convertidor, cada vez que se muestrea una entrada, y el código representa el valor de la señal de entrada en el instante de muestreo. Los códigos pueden enviarse a otros circuitos digitales del sistema en forma también paralela (cada bit por una línea separada).


Conversión digital / analógica: la función de un convertidor digital/analógico (DAC, de "Digital to Analogic Converter") es la de aceptar en su entrada un número entero, codificado en binario con "n" bits, y producir en su salida una magnitud analógica cuyo valor esté relacionado unívocamente con el número aplicado a su entrada. Esta magnitud puede ser una tensión, una corriente o una ganancia, dependiendo del dispositivo(s) de actuación que se dispongan.

SENSORES			
http://www.domotica.net/Sensores_y_Centralitas.htm			
http://www.superinventos.com/seguridadX10.htm			
PRODUCTO	DESCRIPCIÓN	SEÑAL = ALIMENTACIÓN	COSTO
Escape De Gas 	Detecta un escape de gas natural en el hogar.	Alimentación AC/230V Tensión de salida DC9 -12V	71.68
Detector De Apertura De Puertas Y Ventanas 	Detecta la apertura de puertas o ventanas. Sirve como complemento del detector de presencia para incrementar la seguridad del sistema	Alimentación señal RF Transmisión RF 433.92 Mhz Salida 1 Mw	88.66
Escape De Agua 	Es un detector de inundación, detecta hasta la más mínima fuga, funciona con pila de 9V o 12 Vcc	Alimentación 3Vcc	44.00

SENSORES

http://www.domotica.net/Sensores_y_Centralitas.htm

<http://www.superinventos.com/seguridadX10.htm>

SENSOR	DESCRIPCIÓN	SEÑAL = ALIMENTACIÓN	COSTO
Sirenas De Alarma Interna Y Externa 	Sirven para generar avisos acústicos interna o externamente a la casa en caso de alarma.	Alimentación 230V. 50 Hz	99.50
Sensor De Presencia 	Este sensor de presencia se activa al detectar cualquier movimiento enviando una señal a un receptor de señales de Radio Frecuencia.	Alimentación Señal RF Frecuencia de RF 433.92 Mh Salida 1Mw	76.86