



UD 4.- CONTROL POR ORDENADOR

- 4.1. SISTEMAS AUTOMÁTICOS
- 4.2. PROGRAMACIÓN BÁSICA EN MSWLogo
- 4.3. CONTROL MEDIANTE ORDENADOR. LA CONTROLADORA ENCONOR
- 4.4. EJEMPLOS DE CIRCUITOS

1. SISTEMAS AUTOMÁTICOS

Un dispositivo es **automático** cuando es capaz de realizar una tarea determinada sin la intervención del hombre. Para que una máquina funcione automáticamente, es necesario disponer de un **sistema de control**, es decir, de un conjunto de dispositivos de distinta naturaleza (mecánicos, hidráulicos, neumáticos, eléctricos, informáticos, etc.) que realicen las acciones necesarias en cada momento.

Los sistemas de control automático tienen tres partes diferenciadas:

- Las entradas o sensores.- son los dispositivos que detectan el estado en el que se encuentra el sistema.
- La unidad de control.- decide la acción que han de realizar las salidas con la información recibida de las entradas.
- Las salidas o actuadores.- son los dispositivos que ejecutan las acciones que decide la unidad de control.

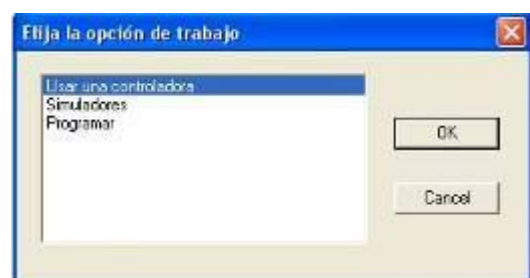
Para realizar programas de control automático, vamos a utilizar el ordenador con unidad de control, pero conectaremos las entradas y las salidas a través de una tarjeta controladora. Para manejar esta tarjeta controladora, se utilizan distintos programas informáticos y uno de los más empleados es el MSWLogo, que estudiaremos a continuación.

2. PROGRAMACIÓN BÁSICA EN MSWLogo

El **MSWLogo** es uno de los programas informáticos más utilizados debido a su sencillez, al idioma (las primitivas están en castellano) y a la inmediatez de los resultados.

2.1. Instalación del programa y entorno de trabajo

Puedes encontrar el programa en el blog www.tecnosefarad.com, en la categoría "MSWLOGO". Su instalación es muy rápida y sólo tienes que seguir los pasos indicados. Una vez que empieces a trabajar, tendrás que tener en cuenta si vas a utilizar una tarjeta controladora, un simulador o simplemente vas a programar.





En la pantalla, aparecerán dos ventanas:

- La de trabajo (similar a la de la imagen), en la cual escribiremos las instrucciones y podremos realizar sobre ellas diversas funciones sobre los programas.
- La de MSWLogo, en la cual podremos ver los resultados de los programas realizados.



2.2. Las primitivas y los procedimientos

Las **primitivas** son los órdenes propios del lenguaje y cada una de ellas ejecuta una acción dentro del programa. A continuación, puedes ver las primitivas básicas más utilizadas, su abreviatura y la orden que ejecutan.

Primitiva	Abrev.	Orden
Avanza <i>numero</i>	av	El cursor avanza en la pantalla el número de píxeles indicados.
Retrocede <i>numero</i>	re	El cursor retrocede en la pantalla el número de píxeles indicados.
Giraderecha <i>numero</i>	gd	El cursor gira hacia la derecha los grados indicados.
Giraizquierda <i>numero</i>	Gi	El cursor gira hacia la izquierda los grados indicados.
Repite <i>numero</i> [..]		Repite la orden entre corchetes el número de veces indicado.
Borratexto	bt	Borra el texto escrito en pantalla.
borrapantalla	bp	Borra la pantalla.
Subelapiz	sl	Sube el cursor para que no dibuje en pantalla en las siguientes instrucciones.
bajalapiz	bl	Baja el cursor para que dibuje en pantalla las siguientes instrucciones.
Ponpos [x y]		Sitúa el cursor en la posición indicada en con las coordenadas.
Ponrumbos <i>numero</i>		Orienta el cursor el número de grados indicado.
Poncolorlapiz [n n n]	Poncl	Pone el color indicado al lápiz.
Poncolorrelleno [n n n]	poncr	Pone el color indicado en el relleno.
Poncolorpapel [n n n]	poncp	Pone el color indicado al papel.
Rellena	re	Colorea la zona encerrada en el lugar donde está el cursor.
Espera n		El programa espera durante n centésimas de segundo.
leepalabra	lp	Lee la palabra que escribimos en el teclado.
leecar	lc	Lee el carácter que escribimos en el teclado.
Escribe [..]	es	Escribe en la pantalla de texto la frase que va entre corchetes.
alto		Provoca el fin del procedimiento que se está ejecutando.

Con el objetivo de hacer una programación más ordenada, éstas se agrupan en **procedimientos**. Ejecutar el procedimiento significa ejecutar todas las instrucciones que lo componen. Los procedimientos comienzan por la primitiva **para**, seguida del nombre del procedimiento, y terminan por la primitiva **fin**.

Después de escribir los órdenes de un procedimiento en la ventana de edición, hay que guardarlo. Si ahora escribimos el nombre del procedimiento, se ejecutarán los órdenes contenidos en él.



2.3. Programación estructurada

Cuando los programas son largos, es conveniente realizar una **programación estructurada**, es decir, dividir el programa en varios procedimientos secundarios. La estructura, por tanto, de un programa en MSWLogo contendrá los procedimientos secundarios en primer lugar y después el procedimiento principal.

Las ventajas de una programación estructurada son:

- ✓ Más sencillo de programar.
- ✓ Más fácil de analizar los posibles errores que se hayan cometido.
- ✓ Los procedimientos secundarios pueden utilizarse en el mismo u otros programas, haciendo éstos más cortos.

2.4. Las variables

Una **variable** en programación es una expresión que puede tomar diferentes valores (de texto o numérico) en el transcurso de la ejecución de un procedimiento. Toda variable debe tener un nombre, que puede ser cualquiera mientras no sea el de una primitiva.

Cuando se trabaja con variables, hay que distinguir entre la forma de nombrarla y la forma de declarar su contenido:

- Se nombra escribiendo delante del nombre comillas ("variable).
- Se le asigna valor escribiendo delante del valor dos puntos (:variable).

Con las variables se utilizan las siguientes primitivas:

Primitiva	Abrev.	Orden
Haz "variable valor		Asigna el valor determinado a la variable.
Escribe :variable	es	Escribe en la pantalla el valor que tiene la variable.

2.5. Las condicionales

Como en la mayoría de los programas, MSWLogo dispone de la **primitiva condicional SI**, cuyo formato es:

Si condición [acción1] [acción2]

El programa realizará la acción1 si la condición se cumple y la acción2 si no es así. La orden acción2 se puede omitir y, en ese caso, el programa seguirá su ejecución si la condición no se cumple.

Quando se requieran condicionales compuestas, se pueden utilizar los **operadores lógicos O e Y**.

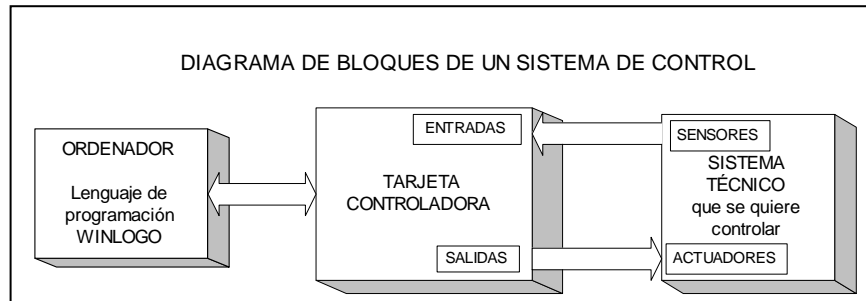
2.6. La recursividad

La **recursividad** es una llamada de un procedimiento a sí mismo, de forma que cuando la ejecución llega a la línea donde está escrito el nombre del procedimiento éste vuelve a ejecutarse. Para interrumpir un programa de forma voluntaria se utiliza la tecla ESC y, si es por programa, la primitiva ALTO.



3. CONTROL POR ORDENADOR

Para realizar el control de circuitos automáticos por ordenador, necesitamos una **tarjeta controladora** que permita comunicar el ordenador con los elementos del circuito.



En este caso, vamos a utilizar la controladora de ENCONOR PLUS (BSP), cuya imagen puedes ver a continuación. El equipo proporciona los elementos de entrada y salida (E/S) necesarios para la experimentación y simulación con diversos sistemas.



Está diseñada para poder accionar y controlar elementos actuadores como bombillas, motores y relés, y capturar datos de interruptores, pulsadores y otros sensores de luz, de temperatura, posición, etc.

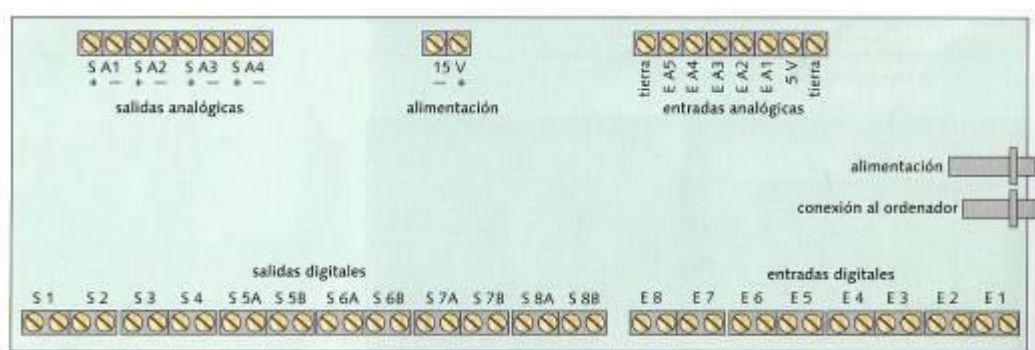
3.1. Descripción de la controladora

En las técnicas de control, los conceptos de ENTRADA y SALIDA son fundamentales.

- Se considera **entrada** cualquier tipo de señal, que es captada y procesada por el ordenador, ya sea, analógica (señales que pueden adquirir múltiples valores) o digital (señales que pueden adquirir únicamente el valor 0 ó 1).
- Se considera **salida** cualquier tipo de señal que el ordenador envía al exterior.

La tarjeta controladora dispone de:

- ✓ 8 SALIDAS DIGITALES INDIVIDUALES ó 4 SALIDAS PARA MOTORES con control del sentido de giro.
- ✓ 4 SALIDAS ANALÓGICAS, que suministran una tensión variable entre 1'6 y 10 v.
- ✓ 8 ENTRADAS DIGITALES.
- ✓ 5 ENTRADAS ANALÓGICAS.





Hay una serie de primitivas especiales que se utilizan cuando se trabaja con la controladora.

Primitiva	Orden
conecta n	Activa la salida n (entre 1 y 8).
apaga n	Desactiva la salida n (entre 1 y 8).
segundos $numero$	Indica los segundos que estará esperando el programa en la ejecución de la primitiva anterior.
voltaje n $valor$	Activa la salida n (entre 1 y 8) y establece el $valor$ de tensión que habrá en ella (entre 1,6 y 10).
Entrada n	Comprueba el valor de la entrada n (entre 1 y 8) y devuelve FALSO, si el sensor digital está abierto, y VERDADERO, si está cerrado.
Leeanalógica n	Comprueba el valor de la entrada analógica n (entre 1 y 5).
Esperaon n	El programa se para hasta que se active la entrada n .
Esperaoff n	El programa se para hasta que se desactive la entrada n .

3.2. Ejemplo circuito automático

A continuación vamos a realizar la **apertura automática de una puerta mediante sonido**.

- ✓ En primer lugar, elegimos los componentes que vamos a emplear para realizar el circuito.
 - Dos finales de carrera para indicar la posición de la puerta.
 - Un sensor de sonido para captar los ruidos.
 - Un motor para abrir y cerrar la puerta.
- ✓ En segundo lugar, establecemos las conexiones de la controladora.
 - Conectamos los finales de carrera y el sensor de sonido a las entradas digitales.
 - Conectamos el motor en una salida digital.
- ✓ Después, realizamos el programa que controlará el circuito con las siguientes condiciones: al detectar un ruido, se abre la puerta activándose el motor hasta que el final de carrera de puerta abierta es accionado. En ese momento, se para el motor durante 5 segundos y después se cierra la puerta.
- ✓ Por último, conectamos los elementos y cargamos el programa en la controladora.

