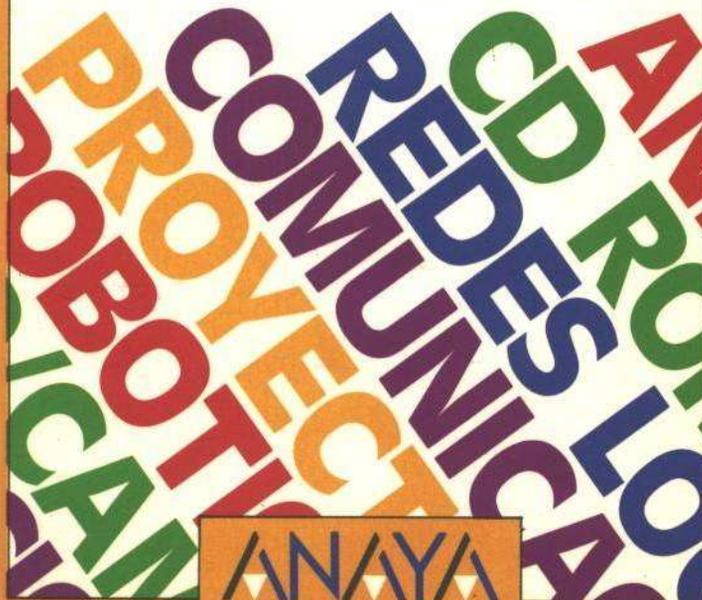


MICROMANUALES

**Proyectos de  
periféricos para  
Amstrad y MSX**

Owen Bishop



**ANAYA**  
MULTIMEDIA

**Proyectos  
de periféricos  
para Amstrad y MSX**

# Proyectos de periféricos para Amstrad y MSX

Owen Bishop



Título de la obra original:  
EASY ADD ON PROJECTS FOR  
AMSTRAD CPC464, 664, 6128 AND MSX

Traducción: Pedro García  
Diseño de cubierta: Narcís Fernández

Reservados todos los derechos. Ni la totalidad ni parte de este libro puede reproducirse o transmitirse por ningún procedimiento electrónico o mecánico, incluyendo fotocopia, grabación magnética o cualquier almacenamiento de información y sistema de recuperación, sin permiso escrito de Ediciones Anaya Multimedia, S. A.

© 1986 BERNARD BABANI (Publishing) LTD  
Originally published by Bernard Babani (Publishing) Ltd  
London, England

© EDICIONES ANAYA MULTIMEDIA, S. A., 1987  
Villafranca, 22. 28028 Madrid  
Depósito legal: M. 10.342-1987  
ISBN: 84-7614-129-7  
Printed in Spain  
Imprime: Anzos, S. A. - Fuenlabrada (Madrid)

Introducción .....	7
1. Fuentes de alimentación.....	17
2. Digitalizador de dibujos.....	25
3. Teclado reducido.....	43
4. Controlador de maquetas.....	51
5. Pitador.....	63
6. Controlador de luces intermitentes.....	73
7. Controlador analógico de modelos.....	79
8. Actuador magnético.....	87
9. Cuentavueeltas por barrera luminosa.....	97
10. Fotoflash.....	107
11. Controlador de juegos.....	117
12. Detector de lluvia.....	129
13. Veleta.....	135
14. Anemómetro.....	145
15. Termómetro.....	155
16. Barómetro.....	161
17. Medidor solar.....	175
18. Selector de datos.....	185
Apéndice: Relación de los patillajes .....	191

# Introducción

## La conexión a microordenadores

Este libro describe cómo construir circuitos electrónicos que le permitirán conectarse a su microordenador Amstrad CPC 464, 664, 6128 o MSX. Todos los circuitos han sido probados en un Amstrad CPC 464, que es totalmente compatible con los modelos 664 y 6128, y en un Toshiba HX-10. Los circuitos para MSX deben funcionar en cualquier otra máquina que se ciña a las especificaciones MSX. Todos los proyectos son sencillos y baratos, y requieren sólo unos pocos transistores o circuitos integrados. Los integrados utilizados son de los más baratos del mercado. El cableado se ha reducido al máximo, cosa inusual en este tipo de circuitos. El resultado de todas estas simplificaciones hace que los proyectos desarrollados en este libro puedan construirse fácilmente, incluso por un principiante. Una vez contruidos, resultan fáciles de operar. Los programas de control que necesitan no son complejos. Cada proyecto incluye uno o dos programas simples que pueden resultarle de orientación. Los lectores expertos podrán desarrollar programas más complejos, pero los

principiantes pueden empezar con estos programas cortos y quizá, más adelante, añadirles posibilidades extra.

## Niveles lógicos

Los ordenadores tratan los números y otros tipos de información como series de ceros y unos. Los ceros y unos, o bits (dígitos binarios), se representan en los ordenadores mediante niveles de voltaje:

- 0 está representado por 0V, y se le denomina nivel bajo.
- 1 está representado por 5V, y se le denomina nivel alto.

Los números y otros tipos de información, incluidas las instrucciones, se representan en los ordenadores haciendo tomar a diferentes cables (o líneas) uno de estos dos voltajes (0V o 5V). En la práctica, los voltajes no tienen por qué ser exactamente 0V o 5V. Cualquier voltaje inferior a 0.8V equivale a 0V (nivel bajo), y cualquier voltaje superior a 2.4V equivale a 5V (nivel alto). Los voltajes comprendidos entre 0.8 y 2.4V no resultan identificables por el ordenador. Esto se menciona para los casos en que, por ejemplo, se encuentre un nivel de 3.7V en una línea donde se debe encontrar un nivel alto. No debemos preocuparnos. 3.7V resulta totalmente aceptable como nivel alto.

## Entradas

Muchos de los proyectos de este libro envían datos (información) de algún tipo al ordenador. Llamaremos a esto entrada, ya que los datos circulan hacia

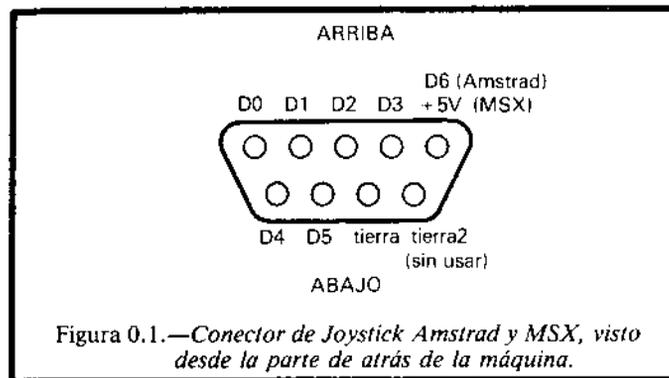


Figura 0.1.—Conector de Joystick Amstrad y MSX, visto desde la parte de atrás de la máquina.

adentro del ordenador. Tanto el MSX como los Amstrad tienen conectores para joystick que pueden utilizarse como puertos de entrada para nuestros circuitos. Estos puertos son prácticamente idénticos para ambos ordenadores. Esto es una ventaja, ya que los sistemas de conexión deben ser los mismos para ambos ordenadores. La mayoría de los proyectos pueden ser utilizados tanto en un Amstrad como en un MSX sin ninguna modificación. Las conexiones se deben realizar utilizando un conector macho del tipo D-9. Existen diferencias entre los BASIC que utilizan ambas máquinas, las cuales hacen necesario el uso de sentencias diferentes para la recepción de datos. Estas diferencias se explicarán más adelante. El patillaje del puerto de joystick se muestra en la figura 0.1. Todos los proyectos deben conectarse siempre al terminal marcado GND (tierra o 0V). Además, se conectará a uno o más de los terminales de datos. El Amstrad tiene siete terminales de datos, mientras que el MSX sólo tiene seis. La mayoría de los proyectos sólo requieren ser conectados a un terminal de datos, así que resulta posible conectar hasta seis o siete dispositivos distintos. El MSX tiene dos puertos de joystick, lo que hace posible añadir otros seis dispo-

Tabla 0.1.—Valores obtenidos en la recepción de datos.

Estado de las entradas de datos	Sentencia utilizada			
	MSX			Amstrad JOY(0)
	STICK(1)	STRIG(1)	STRIG(3)	
Todas altas	0	0	0	0
D0 baja	1	0	0	1
D1 baja	5	0	0	2
D2 baja	7	0	0	4
D3 baja	3	0	0	8
D4 baja	0	-1	0	16
D5 baja	0	0	-1	32
D6 baja	n.a.	n.a.	n.a.	64

\* Las líneas que no especifican como bajas se encuentran en nivel alto.

sitivos adicionales. La señal enviada al terminal de datos desde el circuito es binaria. Como se explicó en el epígrafe anterior, esto significa que los voltajes serán de 0 o 5V (nivel bajo o nivel alto). El programa utiliza una sentencia especial para identificar el nivel de señal de cada uno de los terminales de entrada de datos. En los MSX, una de las sentencias que se pueden utilizar es:

$$X = \text{STICK}(1)$$

La tabla 0.1 indica los valores que se obtienen.

Fíjese que las patillas deben estar normalmente a nivel alto para dar a X el valor 0. Esto incluye el caso en el que las patillas están sin conectar, ya que una patilla sin conectar se puede considerar que está a nivel alto. Cuando se le aplica un nivel bajo a una o más patillas se obtiene un valor distinto de 0. En otras palabras, la entrada se activa en nivel bajo. La tabla 0.1 también indica que se utilizan instrucciones distintas para leer las entradas de las patillas D4 o D5:

$$X = \text{STRIG}(1) \quad \text{o} \quad X = \text{STRIG}(3)$$

Estas patillas están normalmente conectadas a los gatillos de los *joysticks*. Para la lectura de los datos de los dispositivos conectados al puerto del segundo *joystick*, en los ordenadores MSX, las instrucciones correspondientes son:

$$\begin{aligned} X &= \text{STICK}(2) && \text{(para D0 a D3)} \\ X &= \text{STRIG}(2) && \text{(para D4)} \\ X &= \text{STRIG}(4) && \text{(para D5)} \end{aligned}$$

La sentencia ON STRIG y sus asociados permiten interrumpir un programa y traspasar el control a subrutinas, siempre que un determinado dato pase a nivel bajo. El ordenador puede estar totalmente bajo el control de un circuito externo. Esta es una potente posibilidad del BASIC MSX. Las funciones STICK(1) y STICK(2) permiten detectar si determinados pares de patillas están a nivel bajo al mismo tiempo. El par D0/D2 produce un 8, D0/D3 produce un 2, D1/D2 produce un 6, y D1/D3 produce un 4. Fíjese que no se pueden utilizar juntos determinados pares, ya que, si ambos toman el nivel bajo a la vez, el resultado será un 0, igual que si ninguno de ellos lo tomara. Los pares son D0/D1 y D2/D3. Si algún circuito utiliza más de una línea de datos, o si se conecta más de un circuito en un MSX, no se podrán utilizar a la vez las líneas D0 y D1 o D2 y D3. De todas formas, esta deficiencia se compensa con el hecho de que los MSX tienen dos puertos para *joystick*. Los ordenadores Amstrad utilizan otra función distinta y diferentes valores:

$$X = \text{JOY}(0)$$

Los valores dados se pueden ver en la columna de la derecha de la tabla 0.1. Como los MSX, la entrada es

Tabla 0.2.—Estado de las líneas de datos correspondiente a los valores utilizados con OUT y JOY.

Valor	OUT							JOY							
	D7	D6	D5	D4	D3	D2	D1	D0	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1
1	0	0	0	0	0	0	0	1	1	1	1	1	1	1	0
2	0	0	0	0	0	0	1	0	1	1	1	1	1	0	1
3	0	0	0	0	0	0	1	1	1	1	1	1	1	0	0
4	0	0	0	0	0	1	0	0	1	1	1	1	0	1	1
5	0	0	0	0	0	1	0	1	1	1	1	1	0	1	0
6	0	0	0	0	0	1	1	0	1	1	1	1	0	0	1
7	0	0	0	0	0	1	1	1	1	1	1	1	0	0	0
8	0	0	0	0	1	0	0	0	1	1	1	0	1	1	1
16	0	0	0	1	0	0	0	0	1	1	0	1	1	1	1
32	0	0	1	0	0	0	0	0	1	0	1	1	1	1	1
64	0	1	0	0	0	0	0	0	0	1	1	1	1	1	1
127	0	1	1	1	1	1	1	1	0	1	1	1	1	1	1
128	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
255	1	1	1	1	1	1	1	1	n.a.						

1 = alto = 5V

0 = bajo = 0V

activa en nivel bajo; una entrada en nivel alto es equivalente a una no entrada. En un Amstrad, es posible detectar cualquier combinación de entradas a nivel bajo. El valor obtenido de JOY(0) es igual a la suma de los valores de las líneas individuales. Por ejemplo, si D2, D3, D5 están bajas (el resto se encuentra a nivel alto), el valor devuelto por JOY(0) será 44. La tabla 0.2. contiene los números en decimal y su equivalente en binario. Ayudará a descubrir qué entradas se encuentran bajas a partir del valor obtenido por JOY(0). A lo largo de este libro daremos ejemplos de programas para el control de circuitos. No hay espacio suficiente para dar los programas de MSX y Amstrad. Estos comentarios y las tablas 0.1 y 0.2 deben resultar suficientes para adaptar todos los programas a sus necesidades.

## Salida

Cuando un ordenador debe controlar un circuito externo como, por ejemplo, encender una bombilla o activar un relé, debe enviar datos (una orden) al circuito. A este proceso le llamamos *salida*. Los ordenadores Amstrad y MSX son equivalentes en el hecho de que ambos poseen puertos para el control de impresoras. El puerto de impresora nos proporciona suficientes líneas de salida para controlar los proyectos de este libro. Podríamos, en su lugar, utilizar los puertos de expansión, pero resulta más complicado. Además, los interfaces resultarían más caros. Dado que estas máquinas ya tienen incorporados los interfaces necesarios en los puertos de impresora, podremos conectar directamente nuestros circuitos a ellos.

También seremos capaces de controlarlos fácilmente utilizando sencillas instrucciones BASIC. El puerto de impresora MSX se ilustra en la figura 0.2.

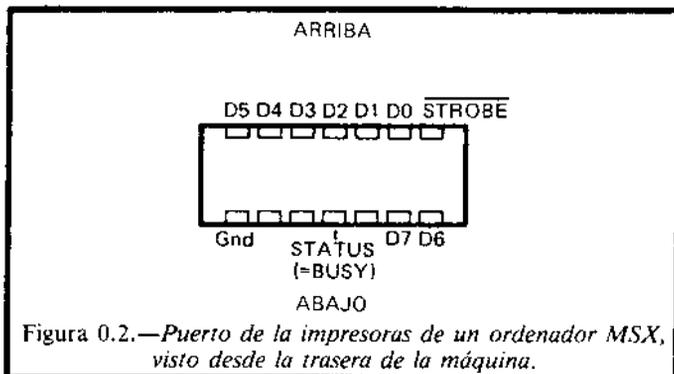


Figura 0.2.—Puerto de la impresoras de un ordenador MSX, visto desde la trasera de la máquina.

El puerto tiene ocho líneas de datos, las cuales todas son salidas. Hay una única entrada, BUSY (ocupado), que se utiliza normalmente por la impresora para indicar al ordenador que está ocupada escribiendo y no puede recibir más datos. Los niveles de voltaje de las patillas son controlables desde el ordenador. Con un ordenador MSX utilizaremos instrucciones del tipo:

OUT 145,6

El número 145 es la dirección del puerto de la impresora en la lista de entrada/salida. El valor 6 determina qué patilla o patillas pasarán a nivel alto. Para determinar estas patillas para un número dado, escriba el número en binario, por ejemplo:

6 en decimal equivale a 00000110 en binario

Los dígitos de este número indican el estado de cada una de las ocho patillas de D7 a D0:

D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	1	1	0

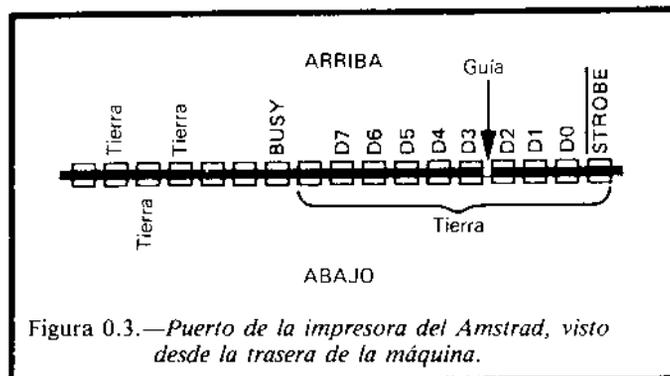


Figura 0.3.—Puerto de la impresora del Amstrad, visto desde la trasera de la máquina.

Utilizando esta instrucción, las líneas D2 y D1 pasarán a nivel alto, mientras que el resto permanecerá en nivel bajo. Los valores utilizados con el OUT deben estar comprendidos entre 0 y 255, permitiendo esto cualquier combinación de salidas. La tabla 0.2 muestra qué salidas pasan a nivel alto cuando se utiliza un valor determinado en la sentencia OUT. Los circuitos pueden utilizar la línea BUSY para hacer esperar al ordenador mientras que se realiza una determinada acción. Si el programa MSX contiene una sentencia como:

LPRINT "X"

(no importa lo que haya que imprimir) el ordenador espera hasta que el circuito externo aplica un nivel bajo en la línea BUSY, antes de continuar con el programa. En el Amstrad, la sentencia correspondiente es:

PRINT#8,"X"

El control de las líneas de salida en el Amstrad (figura 0.3) se realiza con la misma instrucción que en

el MSX. La única excepción consiste en que la dirección del puerto de la impresora es 61184 en lugar de 145. Por tanto, la sentencia correspondiente al Amstrad será:

OUT 61184,6

En la descripción de los proyectos aparecerán diversos ejemplos de la utilización de estas sentencias.

### ***Importante: Conexión de tierra***

Todos los proyectos realizan conexiones de una o más líneas de datos. Depende de cada proyecto qué líneas hay que conectar, pero, de todas formas, en todos los proyectos siempre debe haber una conexión entre el terminal de tierra de los puertos de *joystick* o impresora y la línea de tierra (0V) del circuito del proyecto. Además, siempre debe existir una conexión entre la línea de 0 voltios del proyecto y la línea de 0 voltios de la fuente de alimentación.

# Fuentes de alimentación

Todos los proyectos de este libro requieren una fuente de alimentación. Algunos requieren solamente 5 voltios para los circuitos lógicos y otros 9 ó 12 voltios para la activación de relés. A pesar de que es posible obtener los 5 voltios a partir del ordenador, no resulta recomendable. La potencia requerida puede resultar demasiado grande y sobrecargar la fuente de alimentación del ordenador. Una fuente de alimentación adecuada para estos proyectos debe proporcionar 5 voltios de corriente continua regulada y proporcionar una intensidad de 1A. A la vez, debe proporcionar 9 ó 12V (no es necesario que estén regulados) con una corriente de hasta 2A. Este tipo de fuentes se puede comprar ya construida o alternativamente se puede construir como se indica en esta sección. Este tipo de fuentes proporcionan suficiente potencia como para operar simultáneamente con varios circuitos.

Otra solución consiste en construir cualesquiera de las fuentes de alimentación que se publican en las revistas de electrónica. Suele ser posible obtener números atrasados de los editores. Estas revistas también suelen proporcionar las placas de circuito impreso necesarias, e incluso los kits completos para el montaje. La fuente de alimentación aquí descrita tie-

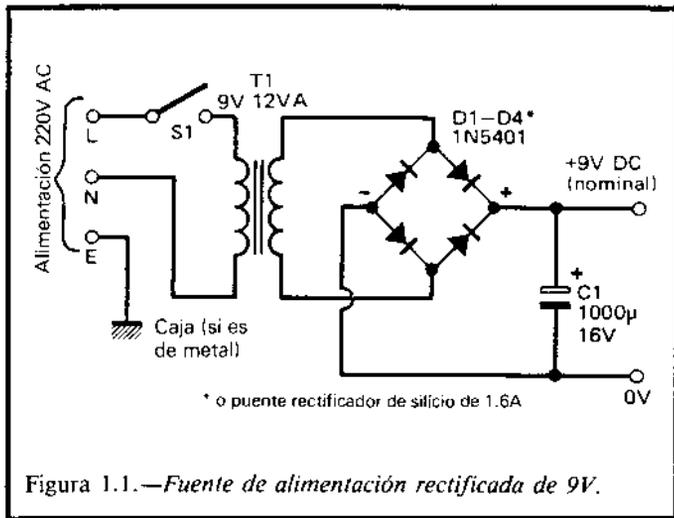


Figura 1.1.—Fuente de alimentación rectificadora de 9V.

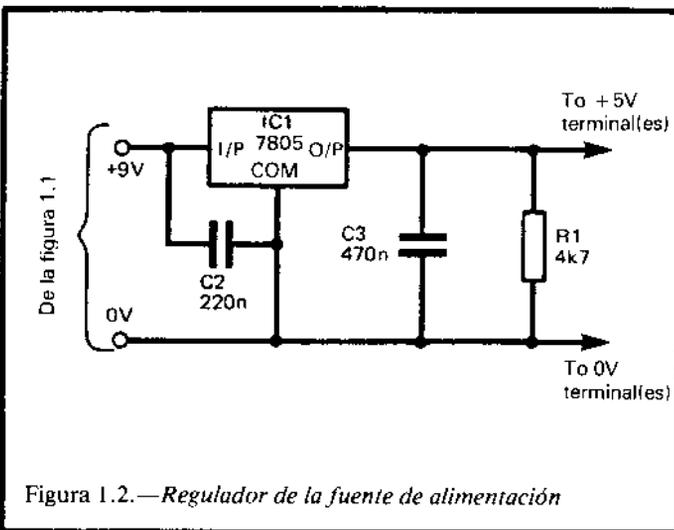


Figura 1.2.—Regulador de la fuente de alimentación

ne dos partes: el transformador/rectificador (Fig. 1.1) y el regulador de voltaje (Fig. 1.2). El transformador/rectificador produce un voltaje de corriente continua sin regular de 9V, capaz para operar con relés. Es capaz de proporcionar 1A de corriente máxima. Sirve para proporcionar potencia al circuito regulador. El circuito regulador produce un voltaje estable de 5V, idóneo para alimentar circuitos lógicos. Si ya posee una fuente de alimentación de 9V, tal como la que acompaña al ZX Spectrum o al ZX81, podrá utilizarla, ya que proporciona suficiente corriente para el regulador y los circuitos. Si esto ocurre, sólo le será necesario construir la unidad reguladora. También puede utilizar un alimentador de 12V como los empleados para alimentar *cassettes* y radios portátiles. Normalmente, son capaces de proporcionar menos potencia (quizá 300 mA), pero pueden resultar suficientes si sólo se quieren alimentar unos pocos circuitos y el regulador de 5V. Los detalles de la construcción de las unidades se ilustran en las figuras 1.1 y 1.2, y su comprensión queda como trabajo para el lector. Si es su primer intento de construir una fuente de alimentación y se le plantean dudas, consulte a un amigo experto antes de empezar. Los circuitos deben alojarse en una caja, de tal manera que no queden al aire cables con voltajes peligrosos. En la sección de transformación/rectificación, el rectificador (D1-D4) y el condensador (C1) podrán soldarse directamente a los terminales del transformador. También podrá usarse un circuito impreso como el de la figura 1.3. El transformador debe atornillarse fuertemente a la caja o pilar, de tal manera que no se suelte en el caso de que la caja se golpee accidentalmente. Tenga cuidado con las polaridades de los diodos y del condensador. Si el condensador se conecta con la polaridad cambiada puede estallar a los pocos segundos de la conexión. El patillaje de los diodos o del puente rectificador (Fig. 1.4) se da en el apéndice. El

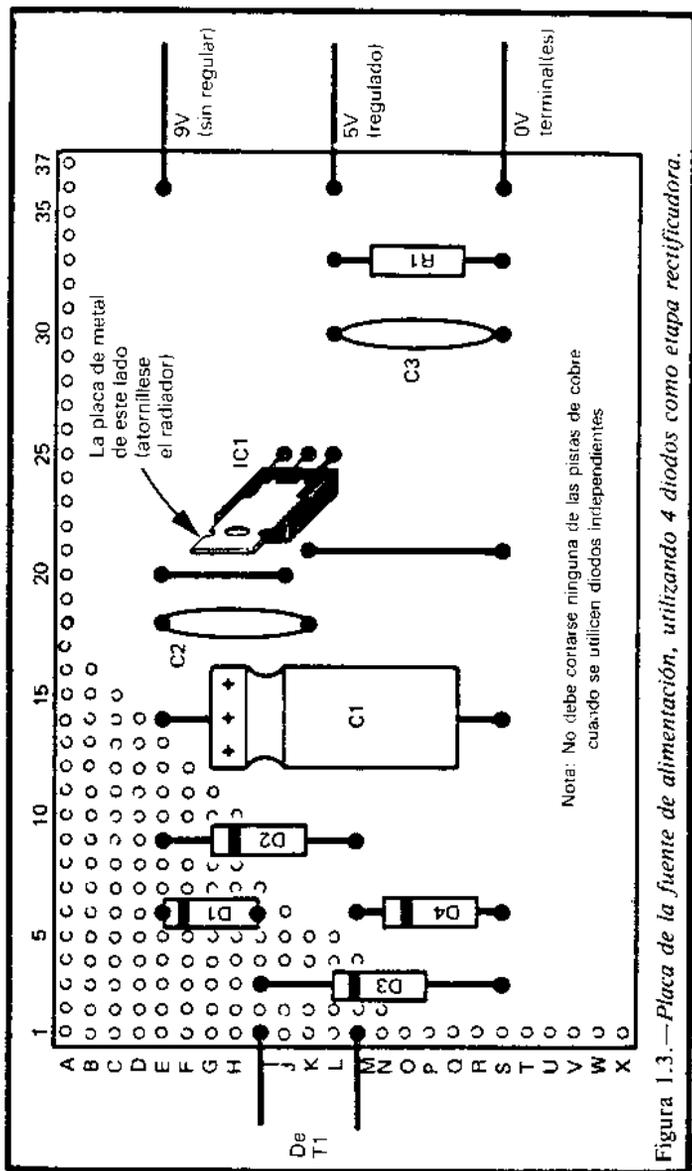


Figura 1.3.—Placa de la fuente de alimentación, utilizando 4 diodos como etapa rectificadora.

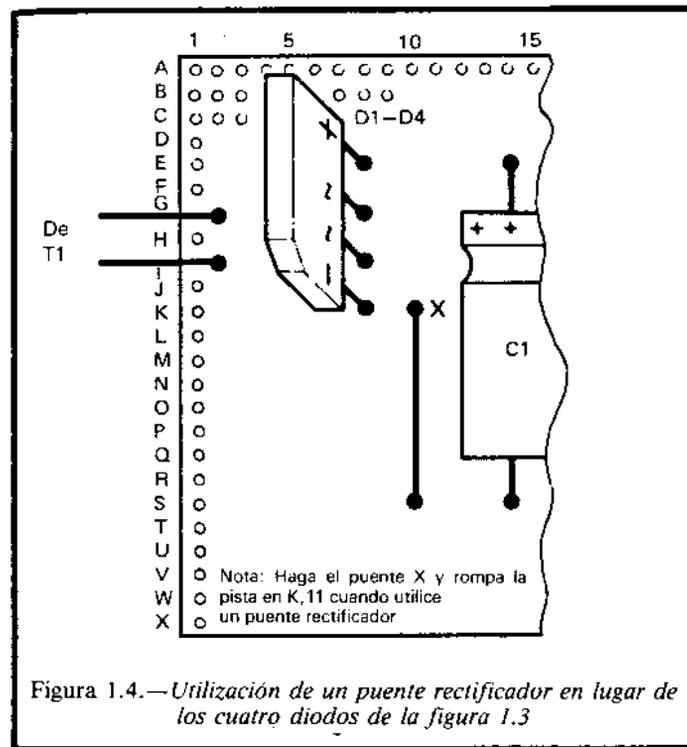


Figura 1.4.—Utilización de un puente rectificador en lugar de los cuatro diodos de la figura 1.3

circuito regulador se ensambla en una placa de circuito impreso como se ve en la figura 1.3. El patillaje del circuito integrado regulador también se puede ver en el apéndice.

El regulador debe tener un disipador de calor. La unidad requiere que los cables de alimentación pasen a través de un agujero practicado en la caja (utilice un pasacables para protegerlo). Si la caja es de metal, debe ser puesta a tierra.

La unidad también requiere tres conectores o terminales para la alimentación de bajo voltaje. Corresponderán a 0, 5 ó 9V. Si pretende alimentar

simultáneamente más de un circuito puede considerar la idea de instalar varios terminales para cada voltaje. Cuando el montaje esté acabado compruebe que no haya cables al descubierto u otras partes de metal en contacto, o con facilidad de entrar en contacto con el resto del circuito o con la caja, si ésta es de metal. Inspeccione las soldaduras para ver si se han realizado correctamente (el estaño ha fluido suavemente a lo largo de las partes soldadas). Inspeccione la parte inferior del circuito impreso utilizando una lupa para comprobar que no hay pequeños (o grandes) puentes de estaño que cortocircuiten pistas adyacentes. Utilice un polímetro para comprobar que cuando el circuito está apagado, y cuando está encendido, no hay cortocircuitos entre las líneas de alimentación. A continuación, cierre la caja y compruebe que al poner la tapa no entran en contacto partes del circuito. Cuando la unidad esté conectada, mida el voltaje que hay entre los terminales de 0 y 9V. La lectura debe ser aproximadamente de 9V, aunque quizá sea un par de voltios superior debido a que no hay carga en el circuito. Mida el voltaje entre los terminales de 0 y 5V. El polímetro debe marcar 5V casi exactamente. Si la unidad no funciona correctamente, apáguela y desenchúfela de la red. Abrala y examínala de nuevo, buscando conexiones incorrectas, soldaduras frías o puentes. Cierre la caja de nuevo antes de volver a enchufarla y conectarla.

## Componentes

### Resistencias

R1 4K7 carbón, 0.25W, 5%

### Condensadores

C1 1000 $\mu$ , 16 ó 25V electrolítico.  
C2 220n poliéster.  
C3 470n poliéster.

### Semiconductores

D1-D4 Diodos rectificadores 1N5401, 3A o puente rectificador de 1.6A. Por ejemplo, W005.  
IC1 Regulador 5V 7805.

### Varios

T1 Transformador 220/9V, 12V.  
S1 Interruptor una posición, un circuito.

Caja de plástico o metal.  
Placa de circuito impreso.  
Terminales 1mm.  
Tuercas y tornillos.  
Tres terminales o hembrillas para conexiones de salida.  
Cable.

## 2

# Digitalizador de dibujos

Utilizando este dispositivo podrá digitalizar un dibujo o fotografía y hacerlo aparecer en la pantalla de su ordenador. El escaner, o dispositivo de barrido, es de baja resolución, así es que trabaja mejor cuando los dibujos tienen grandes zonas oscuras y altos contrastes. Se perderán los detalles, pero suponiendo que se elija un dibujo adecuado, resultará fascinante verlo aparecer en pantalla bloque a bloque. El escaner se mueve a mano, así que será lento y dependerá su precisión de los movimientos realizados. A pesar de esto, este proyecto utiliza los principios básicos de digitalización de los potentes y caros equipos profesionales. Puede resultar de gran interés la utilización de un digitalizador simple.

## Funcionamiento

La idea fundamental de este dispositivo es la de dividir un dibujo en un número determinado de elementos o *pixels*. El dibujo consiste en un grupo de filas y columnas de *pixels* (véase la figura 2.1). El escaner se mueve (en este caso es movido) a lo largo de cada fila de *pixels*. Mide su luminosidad y la transmite al micro. La luminosidad es algo que pue-

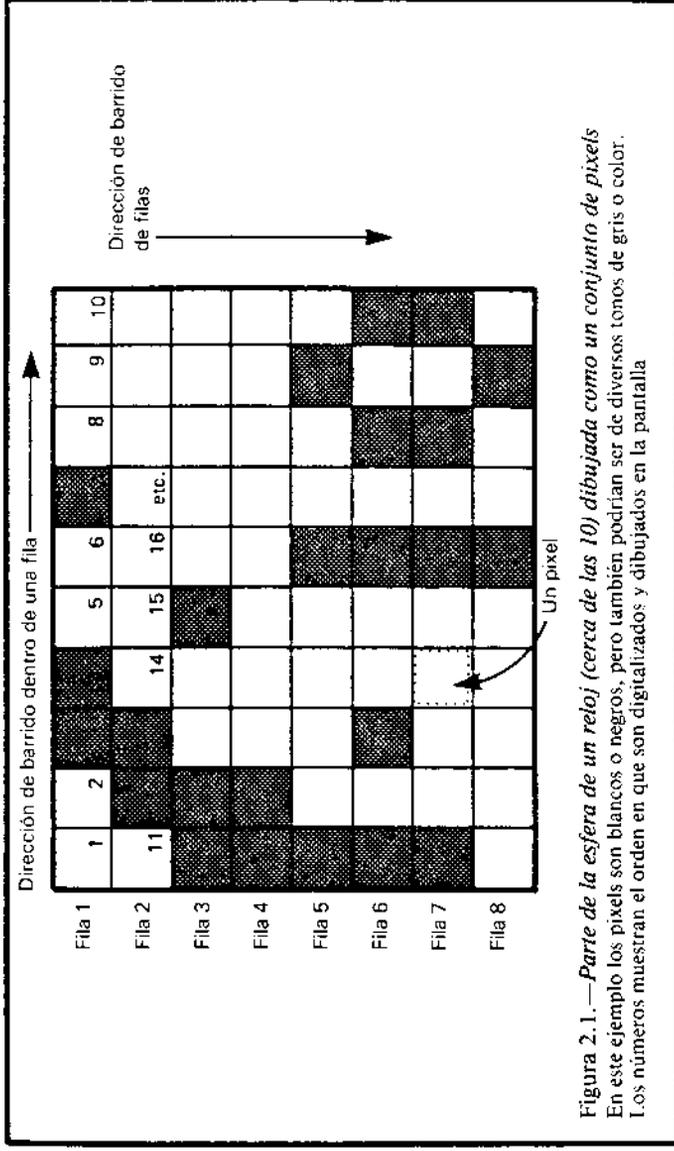


Figura 2.1.—Parte de la esfera de un reloj (cerca de las 10) dibujada como un conjunto de pixels. En este ejemplo los pixels son blancos o negros, pero también podrían ser de diversos tonos de gris o color. Los números muestran el orden en que son digitalizados y dibujados en la pantalla.

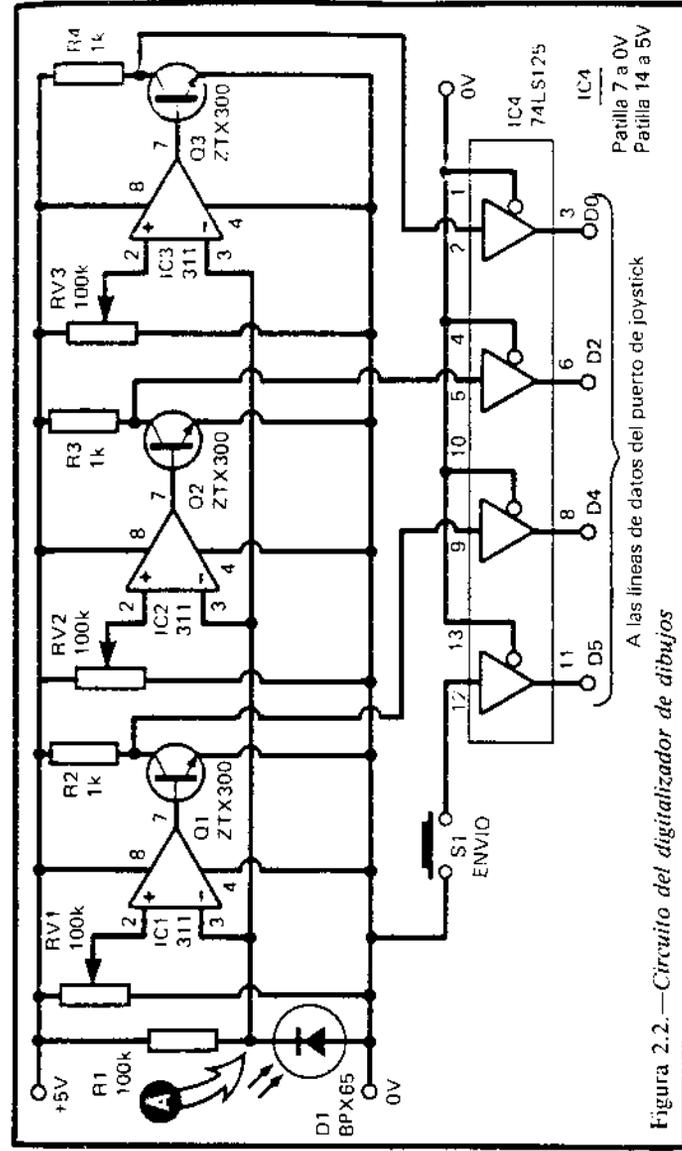


Figura 2.2.—Circuito del digitalizador de dibujos.

de variar entre el blanco más brillante y el más profundo de los negros. Existe un ilimitado número de tonos de grises. Si el dibujo es en color, los diferentes colores pueden ser considerados también como tonos de gris. El ordenador no puede aceptar un número ilimitado de tonos. Para mantener la simplicidad del circuito, el escaner se ha diseñado para que reconozca sólo cuatro tonos. El más claro es el blanco; el más oscuro el negro, y hay dos tonos de gris: gris claro y gris oscuro. Cuando el escaner mide la luminosidad media de un *pixel* transmite al micro el valor más aproximado de los cuatro tonos. El micro dibuja cada *pixel* en pantalla según va recibiendo la información. El blanco se dibuja como un bloque compacto brillante en la pantalla; el negro como un espacio, y los grises como diferentes tramas de puntos. El circuito (Fig. 2.2) está basado en un fotodiodo (D1) que detecta la luz reflejada por el dibujo.

La luz la produce una bombilla de linterna que se monta, junto con el fotodiodo, en el escaner (Fig. 2.3). Cuando la luz incide en D1, se produce una pequeña corriente a través de R1 y D1. Cuanta más luz incida en D1, mayor será la corriente. Cuanto mayor sea la corriente, mayor será la diferencia de potencial entre las bornas de R1. Cuanto mayor sea esta diferencia, menor será el voltaje en el punto A. Esto ocurre porque la borna superior de R1 está siempre en un voltaje fijo de 5V, así que cualquier incremento en la diferencia de potencial hará que el voltaje en la otra borna decaiga. El voltaje en el punto A se mide con tres comparadores (IC1, IC2, IC3). Estos comparan el voltaje de su entrada inversora (-) con el voltaje de su entrada no inversora (+). En este circuito, las entradas inversoras están todas conectadas al punto A, mientras que las entradas no inversoras están unidas cada una al conector central de unos potenciómetros (RV1, RV2, RV3). Cada potenciómetro está dispuesto de tal

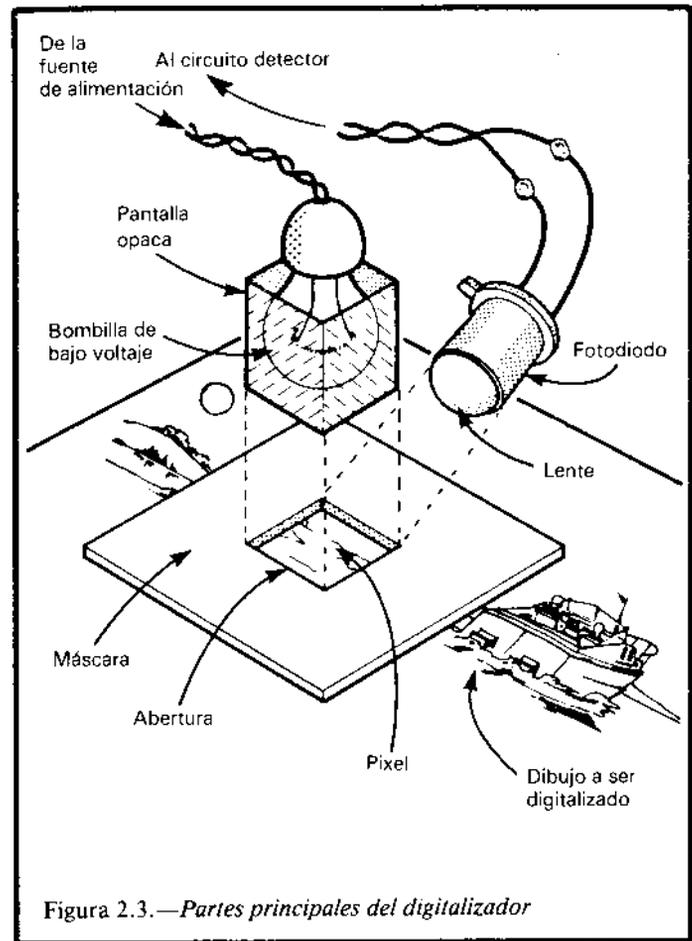


Figura 2.3.—Partes principales del digitalizador

manera que el voltaje en su conector central varíe entre 0 y 5V. Los voltajes obtenidos en cada conector son distintos. En los comparadores, si el voltaje de la entrada inversora (punto A) es superior que el voltaje de la entrada inversora (potenciómetro), el voltaje de salida alcanzará casi los 5V. Si el voltaje de A es

superior al voltaje del potenciómetro, la salida caerá prácticamente a 0. Los potenciómetros se fijan de tal manera que cuando D1 recibe la luz reflejada de un papel blanco, el voltaje en A es inferior a cualquiera de los voltajes obtenidos de los potenciómetros. Las salidas de los tres comparadores están a nivel bajo.

Si una muy pequeña cantidad de luz (la reflejada por un papel negro) alcanza D1, el voltaje de A deberá ser mayor que cualquiera de los voltajes de los potenciómetros, de tal manera que todas las salidas estén a nivel alto. Para valores intermedios (tonos de gris o colores), uno o más de los comparadores debe estar alto mientras los otros estén bajos. La salida de cada comparador alimenta un transistor. Cuando el comparador tiene una salida alta, el transistor conduce. El voltaje de su colector cae a 0V. El transistor invierte la salida del comparador. Este voltaje se alimenta al *buffer* IC4. Cuando el *buffer* se activa (véase el siguiente epígrafe) y se le alimenta un nivel bajo, este mismo nivel bajo pasa al *bus* de datos. Es decir, cuando el comparador tiene un nivel bajo, aparecerá un nivel alto en el *bus* de datos. Las diferentes posibles salidas del circuito se alistan en la tabla 2.1. La parte superior de la tabla muestra qué ocurre cuando no se pulsa el botón de transmisión (S1, figura 2.2). La patilla D5 está a nivel alto, mientras que las otras tres patillas pueden tener cualquier otro nivel. Con un Amstrad, una entrada alta en la patilla D5 hacen que JOY(0) devuelva el valor 0. Si alguna de las otras patillas está baja, el valor será mayor que 0. Como se ve en la tabla, el valor será siempre menor que 32 cuando el botón de transmisión no está pulsado. Con los ordenadores MSX utilizaremos STRIG(3) para detectar cuándo se ha pulsado el botón de transmisión. Cuando no se ha pulsado D5, está alta. El estado de las otras líneas no afecta a STRIG(3), que devuelve el valor 0. Cuando está pulsado el botón de transmisión, STRIG(3)

Tabla 2.1.—Salida de datos del digitalizador

Botón de envío	Sombra	Líneas de datos				Valor leído		
		5	4	2	0	JOY (0)	STICK (1)	STRIG (3)
No pulsado	Ninguna	1	X	X	X	X	X	0
Pulsado	Blanco	0	1	1	1	0	0	-1
	Gris claro	0	0	1	1	48	0	-1
	Gris oscuro	0	0	0	1	52	7	-1
	Negro	0	0	0	0	53	8	-1

0 = 'bajo'      1 = 'alto'      X = 'bajo' o 'alto'

devuelve -1. En el Amstrad, JOY(0) devuelve 32. Este valor se puede incrementar dependiendo de las salidas proporcionadas por los comparadores, como se muestra en la tabla. Si se examina la tabla, se podrá ver que cada combinación diferente de salidas da un valor diferente para JOY(0), o una combinación diferente con STICK(1) y STRIG(1). La forma en la que el programa interpreta estos valores se explicará más adelante.

## Construcción

El primer elemento a realizar es el escaner (Figura 2.3). Existen diversas formas de realizarlo, y ésta dependerá de los materiales disponibles y de su habilidad para trabajarlos. Puede construirlo en una pequeña caja de plástico o metal, realizando las particiones y agujeros que se muestran en la figura 2.3.

Quizá decida construir el escaner en un bloque de madera y perforar los diversos agujeros requeridos. Los elementos esenciales del escaner son:

1. Debe tener una superficie plana que esté en contacto con el dibujo.
2. Debe tener una abertura cuadrada de 5 mm. en la base que defina el área del *pixel*.
3. Una pequeña bombilla debe hacer incidir su luz a través de esta abertura.
4. Debe ser posible acercar o alejar la bombilla, de tal manera que resulte ajustable la cantidad de luz que llegue al dibujo. La lámpara debe estar firmemente sujeta a su soporte, de tal manera que no se deslice.
5. El fotodiodo debe fijarse en una posición tal que reciba la luz reflejada por el dibujo, pero no la luz directa de la bombilla.

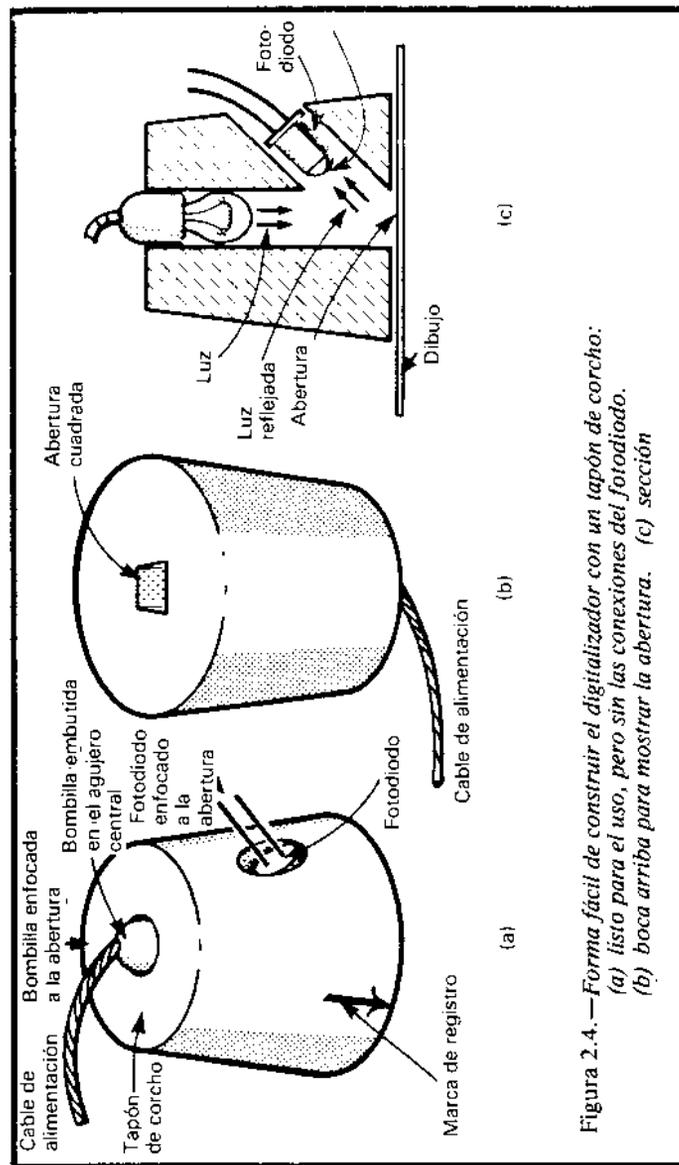


Figura 2.4. — Forma fácil de construir el digitalizador con un tapón de corcho:  
 (a) listo para el uso, pero sin las conexiones del fotodiodo.  
 (b) boca arriba para mostrar la abertura. (c) sección

6. Resulta conveniente que todas las superficies del escaner se pinten de negro mate para evitar reflexiones indeseables.

Nuestro prototipo se realizó en corcho (Fig. 2.4). Se le practicó un agujero con forma cuadrada en la base y se introdujo una pequeña lámpara incandescente que encajaba perfectamente en la parte superior del agujero y que podía desplazarse arriba y abajo. Se realizó un agujero inclinado en el lateral del corcho que acababa en la abertura de la base. Después de soldar los terminales del fotodiodo y aislarse con cinta, se introdujo en el agujero inclinado.

Antes de situar la lámpara y el fotodiodo en sus posiciones finales, se pintaron los interiores de los agujeros con un rotulador negro. Se cubrió todo el corcho con cinta aislante negra, excepto la base, con el fin de evitar filtraciones de luz. Cuando se diseñe el circuito impreso, recuerde que hay que añadir una fuente de alimentación para la bombilla del escaner.

Es posible utilizar bombillas de 3.5V alimentadas por la fuente regulada de 5V. Esta alimentación sobrecarga la lámpara, que proporciona mucha más luz pero que se fundirá después de un relativamente corto período de tiempo. Si la fuente de alimentación proporciona 12V sin regular, puede utilizarse para alimentar bombillas de 12V, 0.3A. Si se desea utilizar bombillas de 6V, se puede hacer uniendo a ésta una resistencia en serie de 10 ohmios y un vatio, de carbono. Conecte primero R1, D1 y uno de los comparadores (por ejemplo, IC1 y RV1). Compruebe el circuito con un voltímetro conectado en el punto A.

Si la iluminación de la habitación es baja, el voltaje de A debe estar cercano a 5V, pero cuando D1 se acerca a una lámpara, el voltaje debe caer prácticamente a 0. Si el voltaje no cambia, es posible que haya conectado D1 con la polaridad cambiada. Fije RV1 aproximadamente en el centro de su recorrido; el

voltaje en el conector central y en la patilla 2 de IC1 debe ser de aproximadamente 2.5V. Mueva D1 hacia la lámpara. La salida de IC1 (patilla 7) está cercana a 5V, pero repentinamente caerá a 0 cuando D1 se acerque a la lámpara. Conecte R2 y Q1. Compruebe que el voltaje en el colector Q1 (donde Q1 se conecta a R2) pasa de 0 a 5V cuando D1 se acerca a la lámpara. Repita la secuencia anterior de pruebas con los otros dos comparadores a medida que los ensambla. A continuación, conecte el *buffer* (IC4). Puede comprobar que las salidas cambian según lo esperado cuando D1 se mueve hacia la lámpara (véase la tabla 2.1).

Ajuste RV1, RV2 y RV3 de tal manera que las salidas cambien de nivel en orden cuando D1 se acerca y se aleja de la lámpara.

## Comprobación final

Antes de conectar el ordenador al circuito es necesario comprobarlo a fondo. Utilice un polímetro para comprobar que no hay cortocircuitos entre líneas. Resulta fácil que un hilo de cobre o una gota de estaño formen un puente. También compruebe que no están cortocircuitadas las dos líneas de alimentación. También resulta interesante comprobar que cada línea de datos une realmente el conector con la patilla correspondiente del circuito integrado en el otro extremo del cable. Una soldadura fría que interrumpa una línea puede causar los problemas suficientes para que el circuito no funcione. Si ha pasado todas las pruebas anteriores, el digitalizador debe encontrarse en las condiciones necesarias para la conexión con el ordenador. Conecte la alimentación del ordenador; si nota algo raro, apáguelo inmediatamente y repita las comprobaciones.

## Programa de prueba

A continuación, se da un pequeño programa para Amstrad que comprueba el correcto funcionamiento del digitalizador:

```
10 FOR J=1 TO 100
20 PRINT JOY(0);
30 PRINT "";
40 NEXT J
50 FOR K=1 TO 500:NEXT K
60 CLS
70 GOTO 10
```

En el MSX, la línea 20 debe ser:

```
20 PRINT STICK(1); STRIG(1); STRIG(3)
```

Cuando todo esté correcto, ejecute el programa. El programa lee cien veces la salida del digitalizador y escribe los resultados en pantalla. Después de un corto retardo, borra la pantalla y realiza 100 nuevas lecturas. La tabla 2.1 muestra los resultados esperados. Sitúe el escaner encima de un papel blanco y pulse el botón de transmisión. La lectura debe ser "32" para el Amstrad y "0 0 —1" para el MSX. Haga pruebas sobre papel negro y diversos tonos de grises. Puede variar los potenciómetros para ajustar los grises al nivel deseado. Las únicas lecturas posibles cuando el botón está pulsado son 32, 48, 52 y 53 (o sus equivalentes en MSX), dependiendo de la luminosidad del *pixel*. Cuando el botón no está pulsado, se obtiene 0, 16, 20 ó 21. Si obtiene otros valores, es que algo falla. Si obtiene números indeseados, escriba los valores obtenidos en binario junto a los números esperados, también en binario. Comparándolos, podrá descubrir qué línea está bloqueada. Una vez localizada, revise de nuevo la línea buscando puentes, interrupciones o cortocircuitos. Los circuitos lógicos

nunca suelen fallar. Han sufrido un riguroso control en el proceso de fabricación, así que si ha comprado los integrados nuevos y de un distribuidor fiable lo más probable es que si el digitalizador falla sea debido a un defecto de construcción.

## Programación

El programa siguiente muestra cómo puede utilizarse el Amstrad para leer datos del escaner y representarlos en pantalla:

```
10 MODE 1:J=0:CLS
20 INK 0,0:INK 1,13:INK 2,26
30 WHILE J<920:SCAN=0
40 WHILE SCAN<32:SCAN=JOY(0)
50 FOR K=1 TO 50:NEXT:WEND
60 IF SCAN =32 THEN PEN 2:PRINT CHR$(143);
70 IF SCAN =48 THEN PEN 1:PRINT CHR$(143);
80 IF SCAN =52 THEN PEN 1:PRINT CHR$(207);
90 IF SCAN =53 THEN PRINT " ";
100 J=J+1
110 WHILE JOY(0)>31: FOR K=1 TO 50: NEXT:
    GOTO 110
120 WEND
130 WEND
140 END
```

El escaner debe moverse a lo largo del dibujo como muestra la figura 2.5. Fije una regla sobre dos escalas situadas a los lados del dibujo. Mueva el escaner de izquierda a derecha, paso a paso, a lo largo de la regla. Pulse el botón de transmisión cada vez que sitúe el escaner en una nueva posición. Al acabar una línea, la regla debe bajarse una posición y volver a situar el escaner a la izquierda. A continuación, se digitaliza la siguiente línea. Examinaremos el programa en detalle para ilustrar algunos puntos sobre

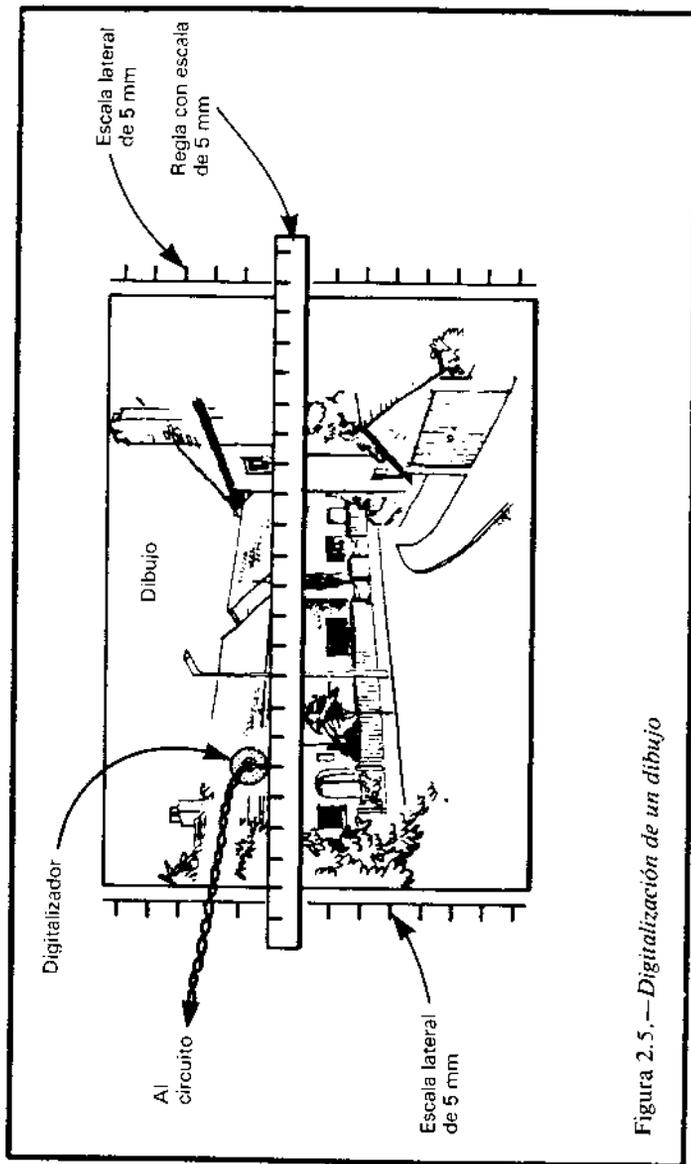


Figura 2.5.— Digitalización de un dibujo

la programación de dispositivos externos. En la línea 30 comienza el bucle principal del programa, encargado de dibujar cada *pixel* en su correcta intensidad. El bucle se repite 920 veces para formar un dibujo de 40 columnas y 23 filas. La línea 40 hace que el ordenador espere hasta que se pulse el botón de transmisión. Cuando se pulsa el botón, se le asigna un valor válido a SCAN y se dibuja un cuadrado de la intensidad luminosa requerida. En función del valor de SCAN (obtenido de la función JOY(0), véase la tabla 2.1), se dibuja un cuadrado blanco brillante (CHR\$(143)), un cuadrado blanco, una trama gris (CHR\$(207)) o un espacio (negro). A continuación, se incrementa J, que es la variable que lleva la cuenta de los *pixels* dibujados hasta el momento. En la línea 110, el ordenador espera a que se deje de presionar el botón. Si no se hiciera esto, se dibujarían docenas de cuadrados antes de que uno fuera capaz de colocar el dedo en el botón y mover el escaner a la siguiente posición. Una vez que se suelta el botón, y si J no ha alcanzado el valor de 920, el bucle se repite. El ordenador espera de nuevo en la línea 40 mientras se mueve el escaner. Cuando se pulsa el botón, lee la luminosidad del área y dibuja el siguiente cuadrado. Cuando se escriben programas de este tipo resulta esencial seguir las instrucciones exactamente. En concreto, debe hacerse esperar al ordenador hasta que se ha hecho lo necesario, antes de permitirle continuar. A continuación, el programa correspondiente para MSX:

```

10 SCREEN 2: COLOR 14,1: CLS
20 OPEN "GRP:" AS 1
30 PRESET(8,8): J=8: K=8
40 IF STRIG(3)=0 THEN GOTO 40
50 SCANA=STICK(1): SCANB=STRIG(1)
60 IF SCANB=0 THEN COLOR 15:
   PRINT#1,CHR$(219)

```

```

70 IF SCANA=0 AND SCANB=1 THEN
COLOR 14: PRINT#1,CHR$(219)
80 IF SCANA=7 THEN COLOR 14:
PRINT#1,CHR$(215)
90 IF SCANA=8 THEN PRINT#1," "
100 J=J+8: IF J=248 THEN J=8: K=K+8
110 PRESET(J,K)
120 IF K=184 THEN GOTO 120
130 IF STRIG(3)=--1 THEN GOTO 130
140 GOTO 40

```

El programa del AMSTRAD trabaja en 23 filas y 40 columnas. Suponiendo que el escaner se mueve 5 mm cada vez, el dibujo más grande que puede ser digitalizado mide 200 mm por 115 mm. Si desea digitalizar dibujos mayores, agrande la abertura del escaner. La versión MSX lee 22 filas y 30 columnas, permitiendo dibujos de hasta 150 por 110 mm.

Hay otras posibilidades de utilización de la salida del escaner. Resulta fácil alterar los programas para que produzcan una imagen negativa. Puede ampliar el programa para que produzca copias en la impresora. Una vez que el dibujo está almacenado en la memoria del ordenador resulta fácil procesarlo de diversas maneras. En algunas aplicaciones puede resultar interesante calcular la proporción entre zonas claras y zonas oscuras, o el número de *pixels* blancos o negros. El ordenador puede localizar los bordes de las zonas de luz y sombra y generar una imagen con éstos, que tendrá la apariencia de un dibujo de línea en lugar de una imagen con tonos continuos. Puede escribir programas para crear imágenes en color basadas en el dibujo original.

## Componentes

*Resistencias* (carbón, 0.25W, tolerancia 5%)

R1 100K.  
R2-R4 1K (3 unidades).

*Potenciómetros*

RV1-RV3 100K miniatura horizontales (tres unidades).

*Semiconductores*

D1 BPX65 o fotodiodo similar.  
Q1-Q3 ZTX300 o transistor NPN de propósito general similar (3 unidades)

*Circuitos integrados*

IC1-IC3 Comparador 311 (3 unidades).  
IC4 74LS125 *buffer* cuádruple triestado.

*Varios*

S1 Pulsador.

Placa de circuito impreso.  
Zócalos de 8 patillas (3 unidades).  
Zócalo de 14 patillas.  
Conector para el puerto de *joystick*.  
Bombilla de bajo voltaje (3.5 a 12V), 0.3A  
(resulta altamente recomendable las bombillas que tienen una pequeña lente, ya que focalizan

la luz en el papel; se encuentran disponibles en cualquier tienda de electricidad).  
Casquillo para la bombilla.  
Materiales para la construcción del escaner.  
Cable.

### 3

## Teclado reducido

A pesar de que es normal controlar el movimiento del cursor, o apuntar y disparar las armas láser contra los marcianos, utilizando las teclas estándar del ordenador, resulta muy útil tener un teclado aparte diseñado especialmente para este propósito. Este teclado reducido tiene cinco teclas, que se pueden utilizar de múltiples formas. Por ejemplo, marcarlas con flechas en cuatro sentidos y utilizarlas para mover el cursor o cualquier otra cosa por la pantalla. Las teclas se pueden disponer según la figura 3.1, de tal manera que resulte fácil identificarlas y encontrar la tecla adecuada. Otra forma de marcarlas puede ser utilizando los cuatro puntos cardinales: N, S, E y O. La quinta tecla es de control y puede tener múltiples funciones, dependiendo de cómo se programe. En un juego se puede utilizar para disparar, o quizá para detener el movimiento. En un programa de dibujo se puede utilizar para activar o desactivar el trazo. El teclado también se puede utilizar como controlador especializado como, por ejemplo, en maquetas de trenes o coches de escalextric (véase el proyecto 4), con las teclas marcadas marcha, deprisa, despacio y marcha atrás. La quinta tecla puede ser parada de emergencia. Se puede utilizar un segundo teclado para controlar los semáforos y cambios de vía. En

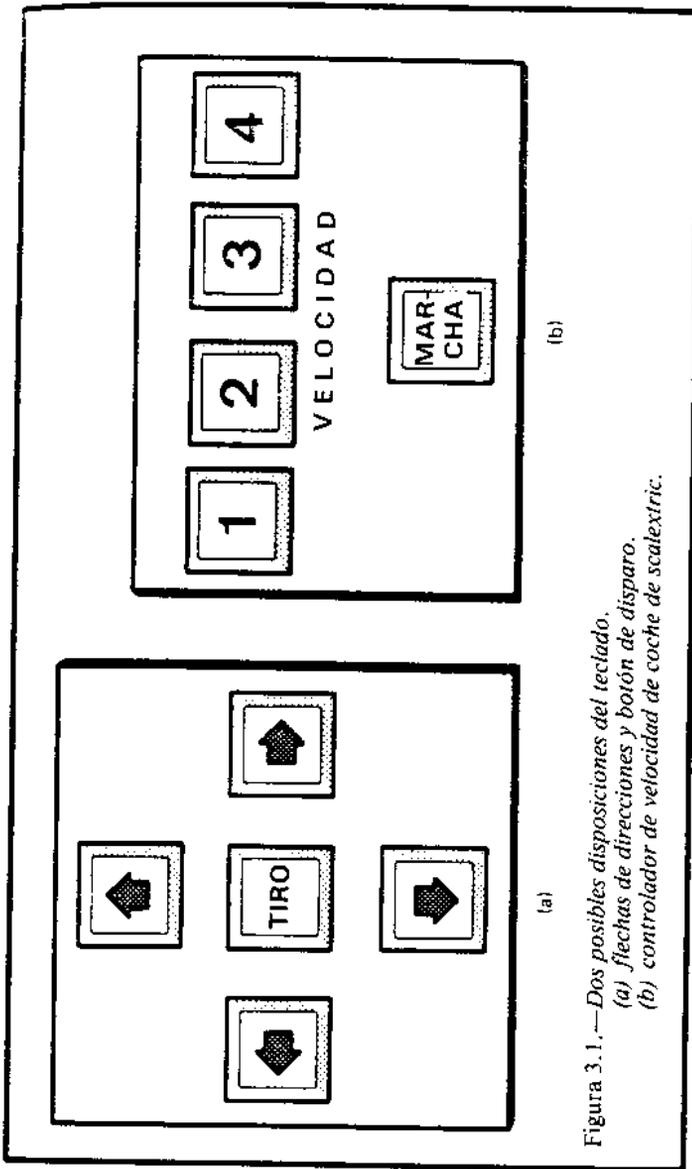


Figura 3.1.—Dos posibles disposiciones del teclado.  
 (a) flechas de direcciones y botón de disparo.  
 (b) controlador de velocidad de coche de scalextric.

muchos juegos para dos jugadores se gana agilidad si ambos poseen un teclado. Resulta especialmente útil cuando un jugador no debe saber el movimiento que realiza el otro, y viceversa. También se puede programar el ordenador para que lea uno u otro teclado, dependiendo del turno de juego. El teclado también se puede utilizar como controlador remoto. El micro se sitúa en una habitación y el teclado en la adyacente, mandándose las señales a lo largo de un cable de seis hilos.

## Funcionamiento

Cada tecla está conectada a una de las entradas de una puerta XOR (o exclusivo, figura 3.2). Una puerta XOR tiene dos entradas. Cuando ambas entradas tienen el mismo nivel (ambas altas o ambas bajas), la salida toma nivel bajo. La salida toma nivel alto sólo cuando las entradas son diferentes. El circuito integrado que se utiliza pertenece a la familia TTL. Es una característica de esta familia el que cuando una entrada está desconectada se considera situada a nivel alto. Por tanto, cuando no se pulsa una tecla, la entrada correspondiente actúa como si tuviera un nivel alto. Cuando se pulsa una tecla, la entrada correspondiente pasa a nivel bajo. El efecto de pulsar varias teclas se muestra en la tabla 3.1

## Construcción

Es posible utilizar cualquier tipo de pulsador, pero sin duda los mejores son los especialmente diseñados para teclado. Habitualmente, el mecanismo del pulsador se vende separado de la tecla. Las teclas se encajan en los pulsadores y las hay de varios tipos. Algunas son de colores brillantes y otras blancas con

Tabla 3.1.—Salidas del teclado

Teclas pulsadas 5 4 3 2 1					Lineas de datos 5 4 2 0				Valor			
									JOY (0)	STICK (1)	STRIG (1)	STRIG (3)
-	-	-	-	-	0	0	0	0	53	8	-1	-1
-	-	-	-	+	0	0	0	1	52	7	-1	-1
-	-	-	+	-	0	0	1	0	49	1	-1	-1
-	-	+	-	-	0	1	0	0	37	8	0	-1
-	+	-	-	-	1	0	0	0	21	8	-1	0
+	-	-	-	-	1	1	1	1	0	0	0	0
+	-	-	-	+	1	1	1	0	1	1	0	0
+	-	-	+	-	1	1	0	1	4	7	0	0
+	-	+	-	-	1	0	1	1	16	0	-1	0
+	+	-	-	-	0	1	1	1	32	0	0	-1

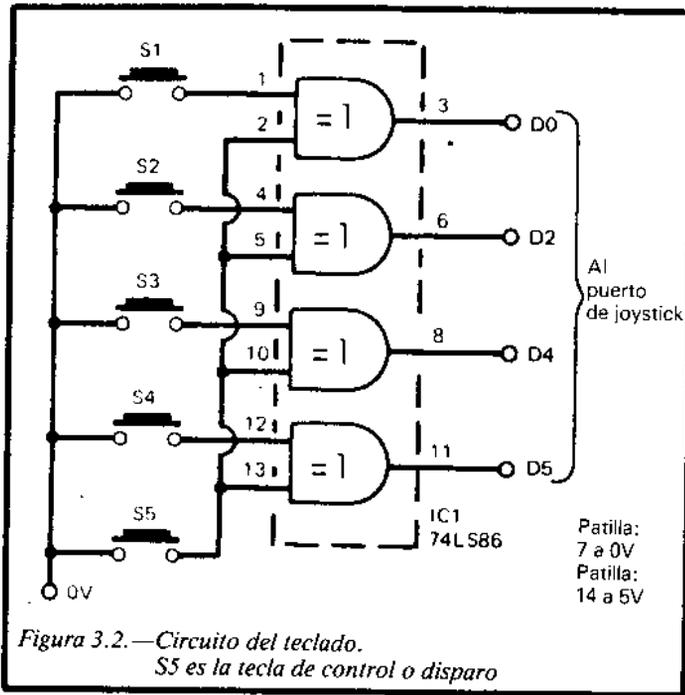
- = Tecla no pulsada  
+ = Tecla pulsada

0 = 'alto'  
1 = 'bajo'

circuito impreso que quepa en la caja. Hacen falta cinco hilos entre el teclado y el decodificador: cuatro para datos y uno para tierra. Estos pueden tener un metro aproximadamente, de tal manera que permitan operar el teclado con facilidad.

## Prueba

Quando esté acabada la construcción, compruebe el circuito para asegurarse que no existen puentes entre las líneas de datos. También compruebe que no hay cortos entre las líneas de alimentación. Conecte el circuito a la alimentación de 5V. A continuación, pulse los botones uno a uno mientras mide los voltajes en las salidas de los *buffers*. La tabla 3.1 da las salidas esperadas. Llegado este momento, puede



cubiertas transparentes. En estas últimas los caracteres se rotulan en la parte superior y a continuación se tapan con la cubierta de plástico para protegerlos. También se puede rotular el símbolo o la letra en un pequeño cuadrado de cartón e insertarlo entre la tecla y la cubierta. El proyecto tiene un acabado mejor si se monta en una pequeña caja de plástico. Algunas cajas tienen el lado superior con una ligera inclinación, lo que las hace ideales para su uso como teclado. Alternativamente, se puede utilizar cualquier caja de plástico de las empleadas para empaquetar alimentos o cualquier otro objeto. La construcción del circuito no presenta problemas. El integrado puede acomodarse fácilmente sobre cualquier pequeño trozo de

conectarse el teclado al ordenador. Una vez hecho esto conecte el ordenador. Si cualquier cosa marcha mal apáguelo inmediatamente y repita la comprobación del circuito. Seguidamente ejecute un programa de prueba como el que aparece en la página 36.

La columna de la derecha de la tabla 3.1 muestra qué números deben aparecer en pantalla. Si no aparecen los valores esperados, lea la página 37 para encontrar el fallo. Una vez que ha sido probado el teclado y se sabe que funciona, podrá utilizarse en los programas propios.

El modo exacto en que esto se haga depende del tipo de programa. El principio es simple: debe leerse el teclado y, en función de los valores obtenidos, transferir el control a diversas partes del programa. El programa de la página 37 ilustra dos puntos importantes sobre la lectura de teclas. Un ordenador trabaja lo suficientemente rápido como para leer cientos de veces la misma tecla en el corto intervalo de tiempo que dura una única pulsación. El programa de la página 37 se controla pulsando el botón de transmisión del digitalizador. En la línea 40 se espera hasta que se detecta que ha sido pulsado el botón. El retardo en el bucle se hace para evitar los rebotes en las teclas. Cuando se pulsa una tecla muy pocas veces pasa de *off* (desconectado) a *on* (conectado). Normalmente se producen una serie de rebotes *on-off-on-off-on-off* antes de que definitivamente se estabilice en *on*.

El retardo permite realizar la lectura sobre una señal estable. La parte principal del programa procesa la lectura obtenida y, a continuación, comprueba que el botón no sigue pulsado antes de tomar una nueva lectura. A continuación, salta al comienzo del programa y espera de nuevo que se pulse alguna tecla. Será necesario utilizar rutinas de este estilo en su programa si pretende interpretar series de pulsaciones como series de comandos. Desde luego, si lo que se quiere es que se realice una acción mientras que una

tecla está pulsada, lo único que habrá que hacer será leer el teclado repetidas veces, sin pausas, hasta que se detecte un cambio de entrada.

## Componentes

### *Circuitos integrados*

IC1      74LS86 cuádruple puerta XOR.

### *Varios*

S1-S5    teclas (5 unidades).

Zócalo de 14 patillas.

Conector para el puerto de *joystick*.

Caja para el teclado.

Cable.

# 4

## Controlador de maquetas

El controlar una maqueta de trenes con un ordenador le añade gran realismo. La locomotora puede ser parada, arrancada o puesta en marcha atrás, y su velocidad puede variarse. El proyecto también puede utilizarse para controlar otro tipo de juguetes controlados eléctricamente, como los escalextric. Si es lo suficientemente hábil como para construir un robot, este proyecto le proporciona los medios adecuados para controlarlo. De hecho, cualquier maqueta o juguete que se mueva mediante motores de corriente continua y bajo voltaje, o que esté activado por electroimanes, debe ser controlado mediante este interfaz. Una maqueta de trenes puede usarse para controlar los cambios de vía o para conectar farolas. También es ideal para el control de semáforos. La mayoría de los circuitos que se utilizan para controlar la velocidad de un motor requieren fuentes de alimentación de un voltaje superior al necesario para hacer que el motor funcione a máxima velocidad. Este circuito utiliza relés para la conmutación; por tanto, sigue siendo posible la utilización de los mismos transformadores y fuentes de alimentación que ya se utilizaban en la maqueta. No resulta necesario obtener o construir una fuente de alimentación especial. Los relés no producen caídas de voltaje, así que

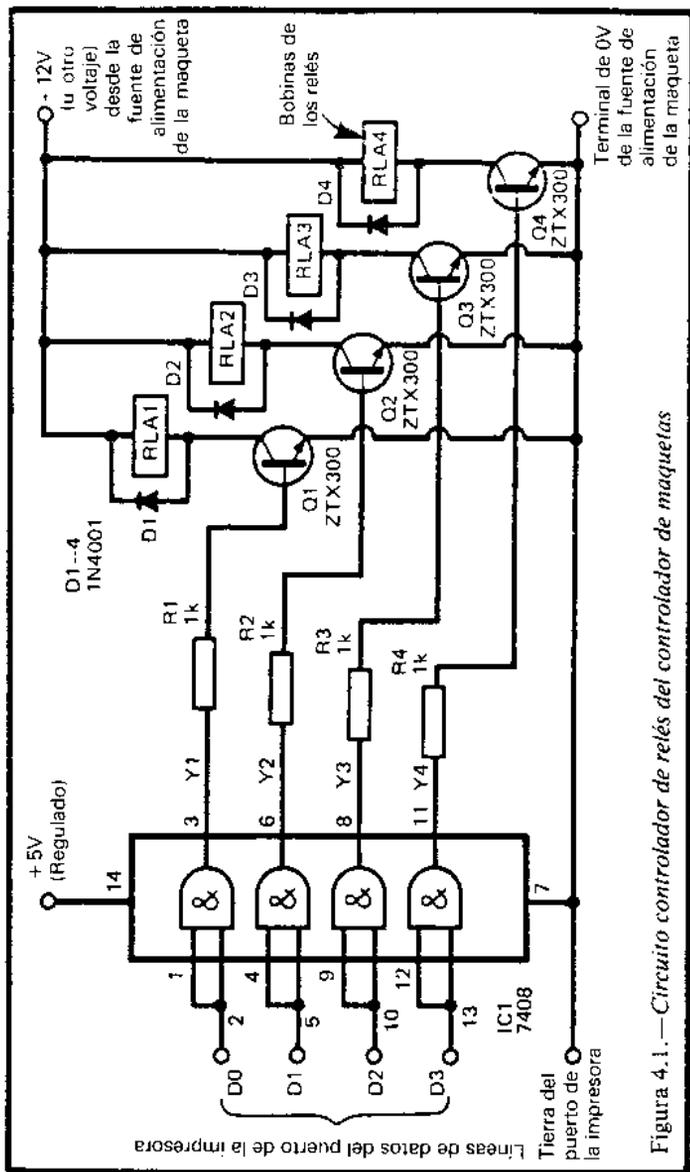


Figura 4.1.—Circuito controlador de relés del controlador de maquetas

los motores pueden funcionar a velocidad máxima cuando se requiere. Una característica de este proyecto es que no resulta necesario construirlo completo. Se puede empezar sólo con un relé y ampliarlo más adelante. A medida que crezca su maqueta, podrá construir una segunda o tercera versión del proyecto. Quizá se utilice uno para controlar la locomotora y otro para los cambios de vía y los semáforos. Véase, en el proyecto 7, un controlador alternativo.

## Funcionamiento

La figura 4.1 muestra la sección controladora de relés. El integrado IC1 contiene cuatro puertas AND que funcionan como *buffers* entre el micro y el controlador. Ambas entradas de cada puerta se conectan a una línea de datos del puerto de la impresora. La salida de cada puerta AND proporciona una corriente de base a un transistor. Si la salida está a nivel bajo no se produce corriente de base. Si está a nivel alto, la corriente de base hace que el transistor conduzca. Esto hace que se produzca una corriente de colector que activa la bobina del relé. Lo que ocurre acto seguido depende de lo que se conecte al relé. La figura 4.2 muestra un circuito de relés típico para el control de una locomotora. Fíjese que la fuente de alimentación para las bobinas del relé y para la locomotora es la misma. Esta debe ser de corriente continua y de un voltaje no superior a 25V, ya que dañaría los transistores. El voltaje normal en las maquetas debe ser de 12V o inferior, así que seguramente no se planteará ningún tipo de problema.

El relé 3 (RLA3) es un simple interruptor *on-off*. Está cableado de tal forma que el circuito está cerrado (la locomotora funciona) cuando el transistor conduce. La corriente que circula por las vías puede pasar a través de dos resistencias (R5, R6), pero una

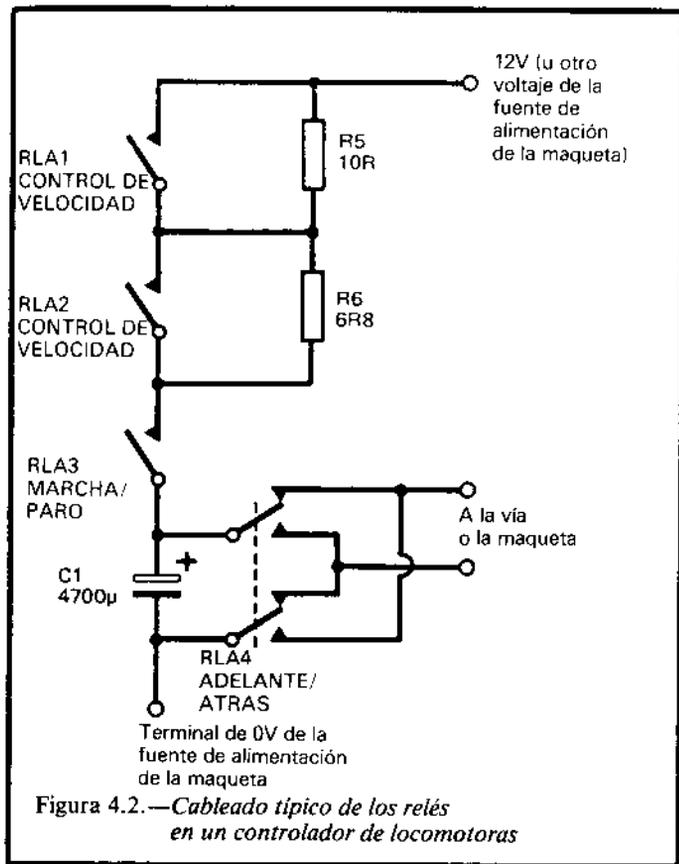


Figura 4.2.—Cableado típico de los relés en un controlador de locomotoras

de ellas, o ambas, se deben cortocircuitar cerrando los relés 1 y 2. Estos están cableados de tal manera que estén abiertos cuando los transistores no conduzcan. Cuando los relés 1 y 2 están abiertos, la corriente de las vías pasa a través de R5 y R6, que valen 16.8 ohmios y que están en serie por el motor de la locomotora. Esto hace que la locomotora se mueva despacio. Los valores que se muestran en la figu-

ra 4.2 son los utilizados en la maqueta concreta en la que se probó el circuito. Seguramente los valores necesarios en su circuito serán distintos, y deberá hallarlos por su cuenta mediante pruebas. Cuando se cierra uno de los relés, la resistencia en serie con el motor se convierte en 10 o en 6.8 ohmios. Estas resistencias reducidas proporcionan dos velocidades altas distintas. Finalmente, si se cierran ambos relés se reduce la resistencia a 0 y la locomotora alcanza su velocidad máxima. El relé 4 proporciona un cambio de sentido. El relé tiene dos pares de contactos que cambian simultáneamente. Como resultado, la corriente fluye en un sentido o en el contrario, haciendo que la locomotora pueda cambiar el sentido de marcha. La tabla 4.1 muestra las salidas necesarias para producir las diferentes velocidades y sentidos.

## Construcción

El circuito integrado ocupa muy poco espacio y permite que quede en la placa el suficiente como para montar los relés R5 y R6. C1 también ocupa gran cantidad de espacio. Resulta esencial instalar este condensador porque la locomotora produce chispas a medida que cruza las uniones entre tramos de vía, o si encuentra zonas donde los raíles no son lisos o se encuentran ligeramente corroídos. Estas chispas generan picos de voltaje que pueden transmitirse a través del circuito y alcanzar el ordenador. No son suficiente como para dañarlo, pero pueden producir cambios en la memoria y producir extraños errores que parecen no tener relación con el programa. Incluso si son muy fuertes o muy frecuentes, pueden llegar a colgar el ordenador. El condensador elimina este tipo de interferencias. C1 debe situarse tan cerca como sea posible de los dos terminales que alimentan las vías. También debe situarse lo más cerca posible

del relé que controla el cambio de sentido (véase la figura 4.2). Otro elemento de protección son los diodos. Cuando se desconectan los relés se produce una gran corriente inversa que puede dañar los transistores. Los diodos sirven para conducir esta corriente, poniendo así a salvo a los transistores. Los diodos deben conectarse lo más cerca posible de los terminales de las bobinas de los relés. El controlador se conecta al micro mediante cinco líneas: cuatro para datos y una para tierra. Como se dijo antes, no es necesario construir el circuito completo de una sola vez. Si usted quiere realizar una prueba a pequeña escala, conecte simplemente IC1, R1, Q1, D1 y RLA1. También será necesario incluir a C1. Este montaje permite experimentar con un único relé, que puede servir para controlar la marcha de la locomotora. El control de velocidad y de sentido puede añadirse posteriormente. En ese momento quizá decida utilizar un único relé para control de velocidad (lo que le proporcionará dos velocidades: media y máxima), quedando así libre un relé que se puede utilizar para controlar cambios de vía o semáforos. La idea principal consiste en que puede empezar con un sistema pequeño, pero construirlo en una placa suficientemente grande que le permita futuras expansiones. Como se dijo antes, los valores ideales para R5 y R6 dependen de las características particulares del motor de la locomotora. La mejor forma de hallarlas consiste en conectar resistencias en serie con la locomotora antes de soldarlas definitivamente al circuito. La corriente que pasa por el motor puede ser de 1 amperio o más, así que resulta esencial que las resistencias puedan soportar una potencia de al menos 2.5W. Resulta fácil encontrar a precios asequibles resistencias bobinadas de valor fijo. También es posible encontrar resistencias variables bobinadas, pero son relativamente más caras y, por tanto, resulta mejor comprar varias resistencias fijas y experimentar

Tabla 4.1—Valores de comando del controlador

Dirección de movimiento	Velocidad	Salida de las puertas				Valores utilizados en el OUT
		Y4	Y3	Y2	Y1	
Adelante	STOP	0	0	X	X	0 a 3
	Lenta	0	1	0	0	4
	Media	0	1	1	0	6
	Rápida	0	1	0	1	5
	Máxima	0	1	1	1	7
Atrás	STOP	1	0	X	X	8 a 11
	Lenta	1	1	0	0	12
	Media	1	1	1	0	14
	Rápida	1	1	0	1	13
	Máxima	1	1	1	1	15

0 = 'bajo'

1 = 'alto'

X = 'bajo' o 'alto'

con ellas. Si se sigue el esquema de dos resistencias y dos relés de la figura 4.2, primero encuentre la resistencia que permita a la locomotora funcionar a su velocidad mínima. Esta velocidad debe ser suficiente para que el motor no se detenga cuando la locomotora cruce las uniones entre vías. También debe permitir que una locomotora parada arranque arrastrando un número de vagones lo suficientemente grande. Una vez que se ha establecido correctamente este valor (que es la suma total de R5 y R6), divida este valor en dos partes, una mayor que la otra. En la figura 4.2 el valor total es 16R8 ohmios, que se ha dividido en 10R y 6R8. Estos son los valores comerciales más cercanos a los necesarios. Si se necesita un valor que no es comercial, es posible unir dos o más resistencias en serie; por ejemplo, se puede construir una resistencia de 7R5 uniendo 1R, 1R, 2R2 y 3R3 en serie. A menudo resulta más fácil poner dos resistencias más grandes en paralelo. En este caso, utilice dos resistencias de 15R en paralelo. Cuando

se conectan en paralelo dos resistencias que son aproximadamente del mismo valor, comparten la corriente, por lo que se pueden utilizar resistencias de menos potencia, tales como las de 1 ó 2W de carbón. Antes de conectar el controlador al micro compruébelo para asegurarse que no haya cortocircuitos. Los 12V de alimentación de la maqueta no deben conectarse directamente a ninguna línea que entre o salga del ordenador, pero la línea de 0V debe conectarse a la línea de 0V del proyecto. Para comprobar el funcionamiento del circuito conecte el controlador al micro y la fuente de alimentación. Encienda el ordenador. Si el ordenador no funciona como es habitual, desconéctelo inmediatamente y revise de nuevo el controlador. A continuación, se da un útil programa de pruebas:

```
10 FOR J=0 TO 15
20 OUT 145,J
30 PRINT J
40 FOR K=1 TO 500: NEXT
50 NEXT
60 GOTO 10
```

El programa anterior es para un ordenador MSX. Para el Amstrad sustituya la línea 20 por:

```
20 OUT 61184,J
```

El programa envía una serie de valores al puerto de la impresora. En la pantalla se puede ver el último valor enviado. La tabla 4.1 muestra qué relés se activan en cada caso. Al principio (valor 0) ninguno está activado. A continuación, para el valor 1, se activa el relé 1 (Y0 pasa a nivel alto). La secuencia continúa con el relé 2 (valor 2), relés 1 y 2 (valor 3), relé 3 (valor 4), relés 1 y 4 (valor 5), etc. Si algún relé no funciona cuando debiera compruebe en la

línea del puerto de la impresora al controlador, las conexiones a la puerta AND y el circuito del transistor asociado. Si funcionan todos los relés, pero en una secuencia errónea, seguramente habrá cruzado los cables.

## Programación

La programación del controlador es extremadamente fácil, simplemente utiliza en comando OUT apropiado con uno de los valores de la tabla 4.1. Hay un punto a tener en cuenta cuando se arranca la locomotora (o cualquier otro motor). Algunas fuentes de alimentación tienen un automatismo que las desconectan cuando resultan sobrecargadas. Si intenta acelerar la locomotora demasiado rápidamente, arrancándola a velocidad máxima (relés 1, 2 y 3 conectados), es posible que el exceso de corriente sobrecargue la fuente de alimentación. Si su fuente de alimentación es de este tipo, arranque la locomotora a velocidad mínima. Una vez que ha iniciado el movimiento puede pasarse a velocidades superiores sin miedo de producir un corte de corriente. Por el contrario, algunos tipos de motores necesitan un exceso de corriente para empezar a funcionar, después de lo cual ya pueden bajar a velocidades menores. Los coches de escalextric suelen ser de este tipo. En este caso, el mejor procedimiento consiste en arrancar con los relés 2 y 3 cerrados, permitiendo el paso máximo de corriente. Acto seguido, genere una orden que abra ambos relés. El micro puede programarse para que produzca estas dos órdenes en rápida sucesión, de tal manera que se produzca una alta aceleración, pero en un tiempo tan pequeño que no sea perceptible, y el coche aparentemente acelere suavemente desde 0. En maquetas de trenes u otros modelos, especialmente robots, puede ser interesante escribir un programa que

controle el modelo desde el teclado del ordenador siguiendo la pulsación de determinadas teclas. A medida que se pulsan las teclas el ordenador almacena en memoria la secuencia y la duración de las pulsaciones, transmitiendo al modelo, acto seguido, la correspondiente secuencia de órdenes. Este tipo de instrucción es la utilizada con robots industriales. Puede ser muy interesante que el ordenador conozca la situación del modelo. En una maqueta de trenes, por ejemplo, es útil conocer exactamente qué zona de la vía se ha alcanzado, cuándo se está aproximando el tren a una estación o cuándo el final de un tren ha entrado totalmente en una vía muerta. Una forma de situar los trenes es disponer un conjunto de rayos de luz que son interrumpidos por el tren. El sensor descrito en el proyecto 9 resulta ideal para este propósito. Se puede situar uno o más en posiciones estratégicas del sistema de vías y hacer posible gran cantidad de maniobras automáticas. Un programa sofisticado puede permitir al operador escribir los nombres de las estaciones de salida y de destino, y el ordenador se encargará de trazar la ruta, fijar los cambios de vía y semáforos, y conducir el tren de una estación a otra sin necesidad de ningún control manual.

## Componentes

### *Resistencias*

R1-R4	1K de carbón, 0.25W, tolerancia 5%.
R5, R6	Resistencias bobinadas, 2.5W, valores adecuados (véase la página 57).

### *Condensador*

C1	4700 $\mu$ , electrolítico, voltaje superior al de la fuente de alimentación del modelo.
----	--

### *Semiconductores*

D1-D4	1N4001 (4 unidades).
Q1-Q4	ZTX300 o transistor NPN similar.

### *Circuito integrado*

IC1	7408 puerta AND cuádruple.
-----	----------------------------

### *Varios*

RLA1-RLA3	Relés miniatura; bobinas adecuadas al voltaje de la fuente de alimentación del modelo, un circuito una posición; contactos adecuados para el voltaje de la fuente de alimentación del modelo y un mínimo de 2A (3 unidades).
RLA4	Relé miniatura de las mismas especificaciones del anterior, pero de un circuito y dos posiciones.

Circuito impreso.  
Zócalo para IC de 14 patillas.  
Cable.

# 5

## Pitador

Un dispositivo que puede admitir un corto pitido cuando se activa puede tener múltiples aplicaciones en juegos, programas y en el hogar. A pesar de que el Amstrad y en MSX ya tienen incorporado un altavoz que se puede programar para pitar, el altavoz está firmemente fijado en el interior de la caja del micro y no puede ser situado en otro lugar. Con este proyecto, el ordenador puede estar en una dirección y el pitador en otra; por ejemplo, situando el pitador en la cocina se puede utilizar el ordenador como "temporizador de huevos duros" pitando al final del tiempo necesario para cada miembro de la familia.

### Funcionamiento

La nota que produce el pitador se genera en un oscilador construido a partir de una única puerta NAND (IC2, figura 5.1). Esta puerta tiene entradas de tipo disparador Schmitt, que hacen que la salida varíe bruscamente cuando la entrada alcanza un determinado valor. Las entradas de la puerta están unidas (excepto una, de la cual se hablará más adelante), de tal manera que actúa como un inversor. Cuando

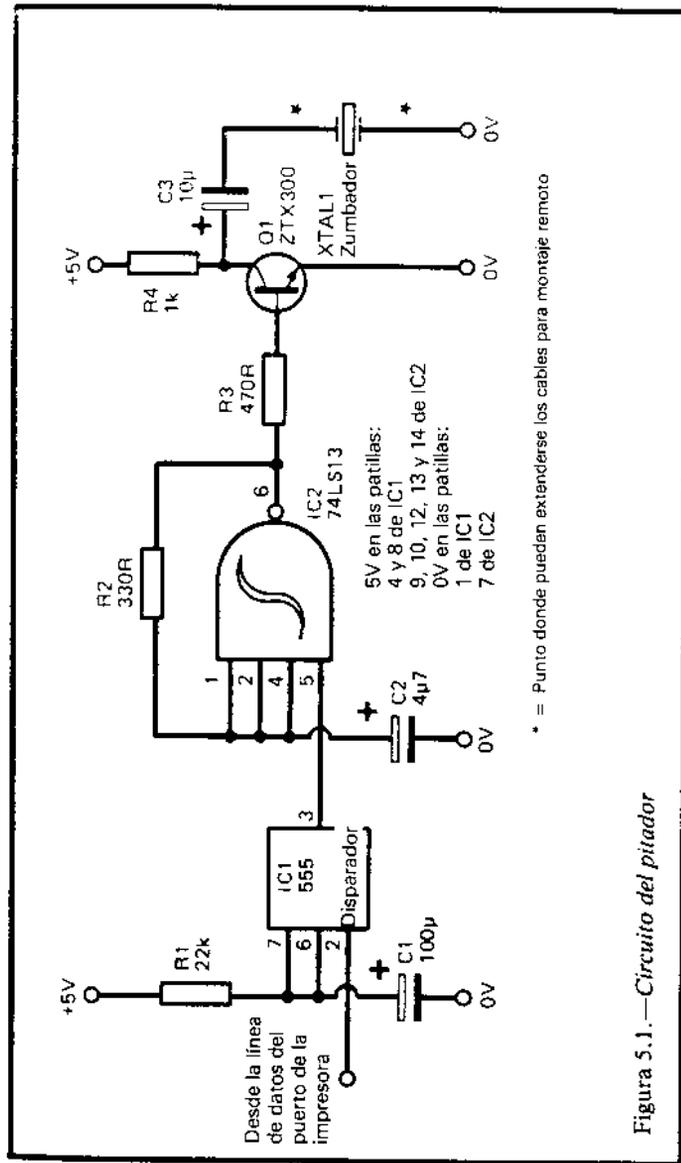


Figura 5.1.—Circuito del pitador

las entradas están altas la salida es baja, y cuando las entradas están bajas la salida es alta. Suponga que la entrada es baja en un principio; por tanto, la salida será alta. La corriente que fluye a través de R2 carga gradualmente el condensador C2. Cuando el voltaje en C2 alcanza un determinado nivel, se considera que la entrada alcanza un valor alto. La salida pasa, consecuentemente, a nivel bajo. Acto seguido, la corriente comienza a fluir de C2 hacia la salida, que está a 0V. El condensador gradualmente se descarga y el voltaje entre sus bornas cae. Cuando cae por debajo de un cierto nivel se considera que la entrada de las puertas está a nivel bajo y, por tanto, la salida pasa a nivel alto otra vez. Siguiendo con este ciclo, la salida pasa continuamente de nivel alto a bajo, y viceversa. La frecuencia a la que oscila el circuito depende de los valores de C2 y R2. Para que el circuito funcione correctamente, el valor de R2 debe estar entre 330 y 470 ohmios, pero el condensador podrá variar razonablemente para conseguir una nota deseada. Con los valores de la figura 5.1 la nota es aproximadamente de 500 Hz. La duración del pitido se controla con el temporizador 555 (IC1). Está cableado de tal manera que se produzca un único pulso alto (en su salida, patilla 3), siempre que un pulso bajo corto alcance su disparador (patilla 2). El pulso de entrada puede ser muy corto y en este circuito utilizaremos un pulso generado por cualquier línea de datos del puerto de la impresora. La longitud del pulso de salida puede ser cualquiera que queramos, dentro de unos márgenes. Su longitud depende de los valores de C1 y R1. La ecuación para calcular la duración es:

$$t = 1.1 RC$$

Siendo t el tiempo en segundos, R la resistencia R1 en ohmios y C la capacidad de C1 en faradios. Con

los valores dados en la figura 5.1, la duración es aproximadamente de 2.4 segundos.

La salida del temporizador está normalmente a nivel bajo, lo que hace que una de las entradas de la puerta esté a nivel bajo, lo que evita que la puerta cambie de estado. No puede oscilar y, por tanto, no se oye ningún sonido. Cuando se activa el temporizador su salida pasa a nivel alto por 2.4 segundos, durante los cuales la puerta NAND oscila y produce el pitido. El sonido lo genera un zumbador piezoeléctrico. Este componente es una delgada lámina de material cristalino que vibra cuando es atravesado por una señal pulsante. Es muy similar a un micrófono de cristal o a la cabeza cerámica de un tocadiscos, pero funcionando al contrario. La salida del oscilador es insuficiente para hacer que el cristal vibre, así que utilizamos el transistor Q1. También es posible sustituir el zumbador por un pequeño altavoz.

## Construcción

El circuito puede montarse en una pequeña caja de plástico, suficiente para albergar la placa de circuito impreso. Sólo necesita conectarse al puerto de la impresora mediante dos cables: tierra y una línea de datos. Si se pretende hacer funcionar el pitador a distancia del ordenador, es mejor situar la parte principal del circuito cerca del ordenador, y el cristal (o altavoz) al otro extremo de un largo par de cables. El volumen obtenido del zumbador será mayor si se monta sobre una superficie firme (pero no demasiado). La superficie actúa como resonador, ayudando a transferir la energía del cristal al aire circundante. La mayoría de los cristales vienen montados en una carcasa de metal ligero con lengüetas (figura 5.2). Estas lengüetas se pasan a través de agujeros realizados en

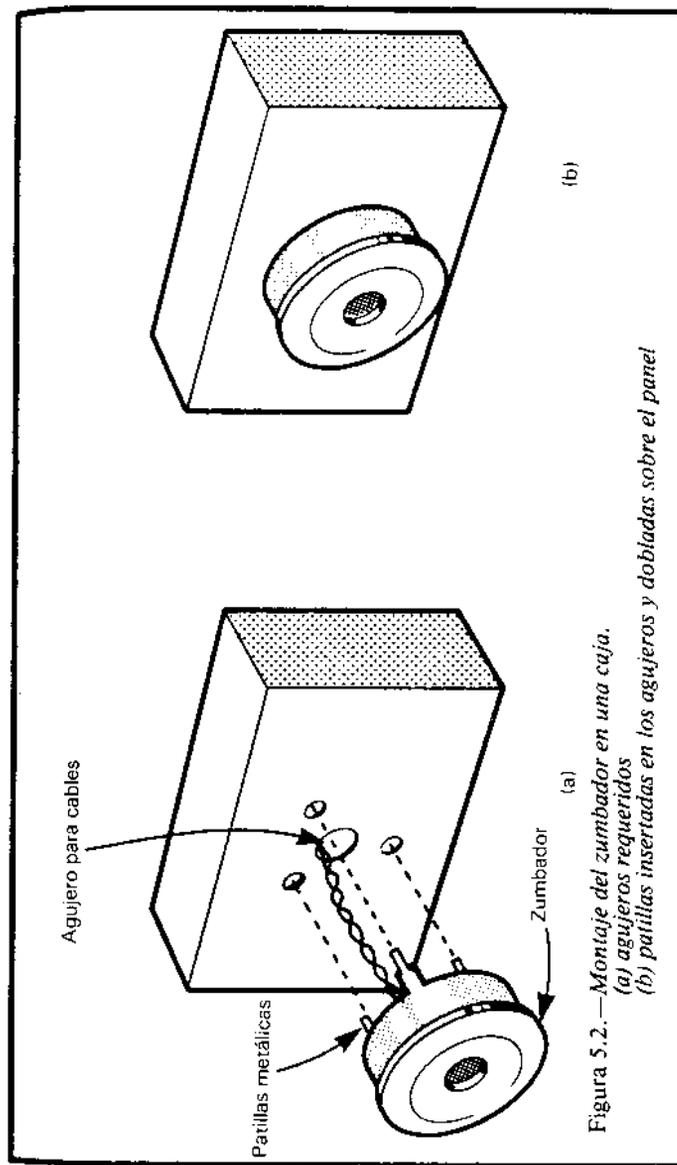


Figura 5.2.— Montaje del zumbador en una caja.  
(a) agujeros requeridos  
(b) patillas insertadas en los agujeros y dobladas sobre el panel

la caja de plástico que contiene el circuito impreso (u otra caja, si el cristal se va a situar en otra habitación). A continuación, se doblan las lengüetas de tal manera que el zumbador quede firmemente unido a la pared de la caja. Si en lugar de un zumbador piezo-eléctrico se utiliza un altavoz, móntese en el interior de la caja y realícense algunos agujeros, de tal manera que el sonido pueda salir al exterior. Se puede utilizar una vieja radio para este circuito, siempre que el altavoz funcione correctamente. Aparte de los puntos antes mencionados, la construcción del circuito es muy sencilla. El circuito se puede probar simplemente conectándolo a la alimentación de 5V. Habitualmente pita tan pronto como se conecta. Conectando temporalmente la patilla 2 de IC1 a la línea de 0V producirá un nuevo pitido. Si esto no es así, compruebe cuidadosamente el cableado. Si el tono no es el deseado, sustituya el condensador C2. Cuanto más grande sea el condensador, menor será la frecuencia. Si la longitud del tono no es correcta, varíe C1 o R1. Cambiar uno o ambos de estos componentes a valores mayores hace que el tono se prolongue. Es posible tener el tono sonando unas decenas de minutos. Cuando el circuito esté funcionando correctamente, conéctelo al ordenador. Para hacer que el circuito funcione es necesario que la línea a la que esté conectado esté a nivel alto, pase por un instante a nivel bajo y, de nuevo, vuelva a nivel alto. Si permanece en nivel bajo, el pitido continuará indefinidamente. Un programa de prueba puede ser:

```
10 OUT 145,0
20 OUT 145,255
```

Utilice la dirección 61184 para la versión Amstrad. Este programa pasa las líneas de datos a nivel bajo y de nuevo a nivel alto. Ejecute el programa una vez antes de conectar el pitador al ordenador para que

todas las líneas pasen a nivel alto. A continuación, conecte el pitador y enciéndalo. Pitará una vez, pero acto seguido se mantendrá en silencio. Ejecute el programa. Una señal baja corta es suficiente para dispararlo, haciéndolo sonar el tiempo que determine el valor de C1. Cuando deje de pitar, ejecute de nuevo el programa y volverá a sonar.

## Programación

Comience cada programa con una instrucción que pase todas las líneas de datos a nivel alto (línea 20). Siempre que quiera hacer sonar el pitador utilice el par de líneas de programa que se mostraron en el epígrafe anterior. Si el pitador debe sonar varias veces, puede ser conveniente realizar una subrutina con estas dos líneas. Es importante resaltar que los valores del programa afectan a todas las líneas de datos. Si el pitador es el único periférico conectado al puerto de la impresora, no existe ningún problema. Pero si usted tiene otros periféricos, deberá ingeniárselas para mandar los pulsos única y exclusivamente a la línea que controla el pitador. Si se modificara una línea que controla otro dispositivo, seguramente se producirían efectos indeseados. Para evitar errores de este estilo, el programa debe mantener en memoria el valor que se está enviando al puerto de la impresora. La forma más fácil es tener una variable PUERTO que se modifique cada vez que se da una instrucción OUT. Por tanto, se deberá empezar el programa con una sentencia del tipo :

```
10 PUERTO=255:OUT 145,PUERTO
```

Más adelante, en el programa, quizá se quiera que la línea D2 tome el nivel bajo para controlar una ma-

queta. Dado que la línea D2 corresponde al valor 4 (véase la tabla 0.2), la línea requerida es:

200 PUERTO=251: OUT 145,PUERTO

Obteniéndose 251 como resultado de restar 4 a 255. Esta sentencia baja el nivel de D2 mientras mantiene el resto en alto. Supongamos que el pitador está conectado en la línea D5 (valor 32). Normalmente, para bajar esta línea hubiéramos utilizado la instrucción:

300 PUERTO=223: OUT 145,PUERTO

Obteniéndose 223 como resultado de restar 32 a 255. Esta instrucción disparará el pitador como se requiere, pero hará que la línea D2 pase de nuevo a nivel alto, siendo esto un efecto indeseado. La sentencia correcta sería:

300 PUERTO=219: OUT 145,PUERTO

Obteniéndose 219 como resultado de restar 32 y 4 a 255. Esta sentencia es perfectamente satisfactoria si se sabe que D2 está en nivel bajo, pero en muchos programas no será posible predecir el nivel de la línea 2. Esta es la auténtica razón de utilizar la variable PUERTO. Esta variable mantiene un registro del estado de las líneas. Hacemos uso de esta información utilizando su valor actual para calcular su nuevo valor. Para los cálculos se utiliza el operador lógico AND:

300 PUERTO=PUERTO AND 223: OUT 145,PUERTO

Recuerde que el pitador está conectado a la línea D5, cuyo valor es 32, y que 223 se obtiene de restar 32 a

255. Si enviamos este nuevo valor de PUERTO al puerto de la impresora, D5 pasará a nivel bajo, pero sin cambiarse los valores de las otras líneas de datos; así se disparará el pitador sin afectar a las otras direcciones. Todo lo que tenemos que hacer después de esto es devolver D5 de nuevo a nivel alto sin alterar las otras líneas:

310 PUERTO=PUERTO+32: OUT 145, PUERTO

El par de líneas anteriores se pueden utilizar en cualquier parte del programa o en una subrutina que dispare el pitador sin afectar a otros dispositivos conectados al puerto. Hemos dado una extensa explicación de este sistema porque es el método general de controlar un dispositivo sin afectar a los demás. A pesar de que los párrafos siguientes no tienen nada que ver con el pitador, los incluimos aquí para que la explicación sea más completa. La línea 310 anterior vale para pasar D5 a nivel alto sólo cuando sabemos que se encuentra a nivel bajo. En el ejemplo anterior sabemos que esto es así porque la línea 300 se ha encargado de que ocurra. Supongamos que queremos hacer que una línea, de la cual no conocemos el estado, pase a nivel alto. Para ello, sumaremos 32 al valor de PUERTO, sólo si éste indica que D5 se encuentra en nivel bajo. Para este propósito utilizaremos el operador O lógico (OR):

400 PUERTO=PUERTO OR 32: OUT 145, PUERTO

Esta sentencia sirve para que D5 pase a nivel alto, independientemente de su estado anterior. Deja las demás líneas inalteradas. Esta sentencia puede reemplazar perfectamente la línea 310 antes descrita.

## Componentes

## 6

*Resistencias* (carbón, 0.25W, tolerancia 5%)

R1	22K
R2	330R
R3	470R
R4	1K

*Condensadores* (electrolíticos)

C1	100 $\mu$
C2	4 $\mu$ 7
C3	10 $\mu$

*Semiconductor*

Q1	ZTX300 o transistor NPN similar.
----	----------------------------------

*Circuitos integrados*

IC1	Temporizador 555.
IC2	74LS13 doble puerta NAND y disparador Schmitt.

*Varios*

XTAL1	Zumbador piezoeléctrico o altavoz de 3 a 15 ohmios.
-------	---

Placa de circuito impreso.

Zócalo de 8 patillas.

Zócalo de 14 patillas.

Espadines de 1 mm. (5 unidades).

Conector para el puerto de la impresora.

Cable.

# Controlador de luces intermitentes

Este circuito sirve para que una o dos luces (quizá más) despidan destellos siempre que el micro ordene. Tiene cantidad de aplicaciones en modelismo, desde los ojos intermitentes de un robot hasta los semáforos de una maqueta de trenes. Resulta útil en juegos y en aplicaciones más serias en el hogar como control de luces de alerta u otros indicadores. Las luces son diodos LED que se pueden conseguir de varias formas y colores rojos, naranjas, amarillos o verdes. También existe la posibilidad de utilizar diodos infrarrojos, con aplicaciones en control remoto. También es posible modificar el circuito para controlar pequeñas bombillas de tungsteno.

## Funcionamiento

El circuito básico consiste en un *flip-flop* J-K. Los diodos se conectan a las dos salidas, Q y  $\bar{Q}$  (figura 6.1). Estas salidas tienen estados opuestos, de tal forma que cuando un diodo está encendido el otro permanece pagado. Este funcionamiento puede ser útil para un par de luces de alerta. En una maqueta de trenes, situando los diodos uno junto a otro, se pue-

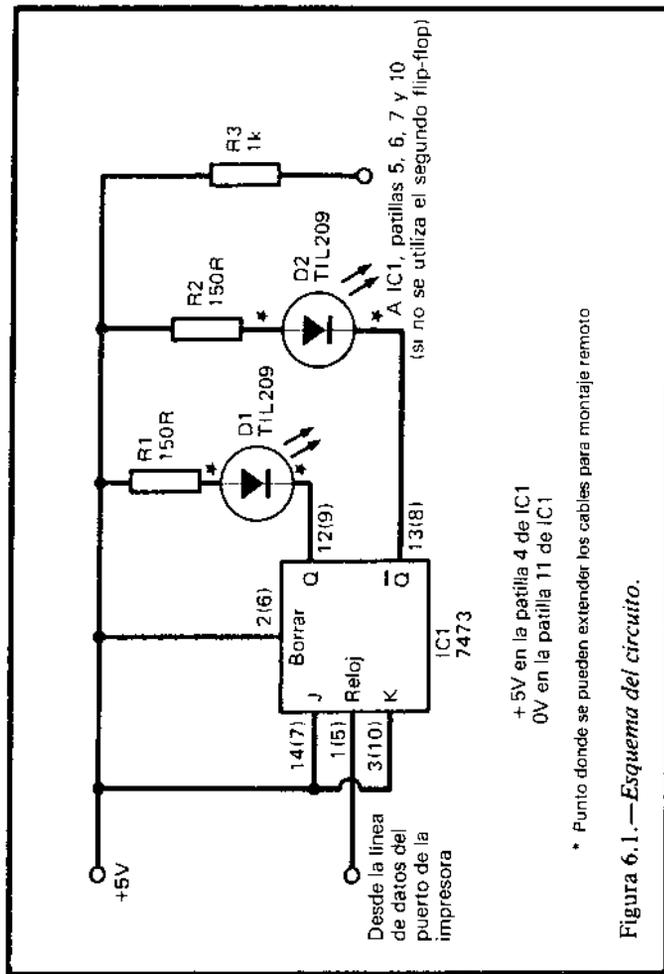


Figura 6.1.—Esquema del circuito.

de imitar al típico par de luces intermitentes que se encuentran en las carreteras cerca de los pasos a nivel. Si sólo se quiere que una luz se encienda y se apague, entonces bastará con conectar un único diodo. Las entradas J y K del *flip-flop* están conectadas

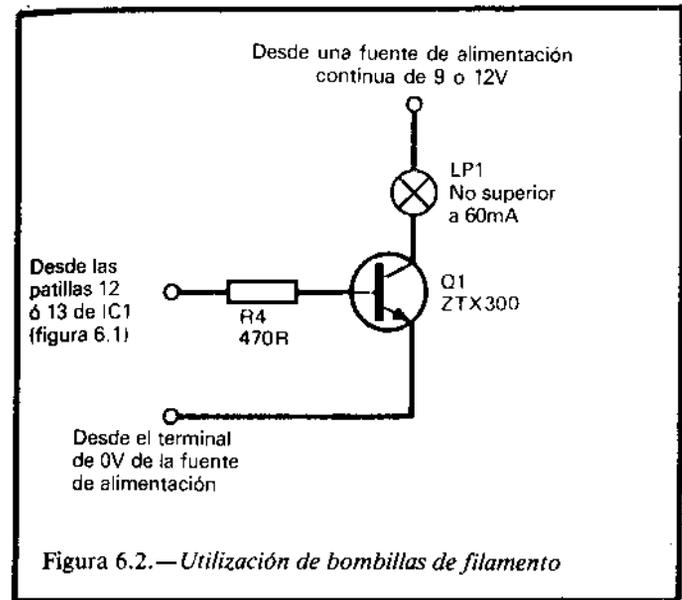


Figura 6.2.—Utilización de bombillas de filamento

a 5V, de tal manera que éste cambie de estado cada vez que le llegue un pulso a la entrada del reloj. A esto se le llama una acción de vasculamiento. Si un diodo está pagado y debe conectarse, el micro hace que la salida de datos que controla el dispositivo pase, por un instante, a nivel bajo. El diodo continuará encendido hasta que de nuevo la salida de datos pase a nivel bajo. Téngase en cuenta que los diodos lucen cuando la salida del *flip-flop* se encuentra a nivel bajo. Es posible realizar múltiples modificaciones al circuito. La salida del *flip-flop* resulta insuficiente para controlar una bombilla de incandescencia, pero añadiendo un transistor resulta factible. En este caso, la bombilla luce cuando la salida del *flip-flop* pasa a nivel alto. El integrado contiene dos *flip-flop*, de los cuales sólo se utiliza uno. Las patillas de conexión del segundo se muestran entre paréntesis en la figura 6.1.

Ambas entradas de reloj pueden conectarse a la misma línea de datos, controlándose así dos pares de luces. Alternativamente, se pueden utilizar dos líneas de datos distintas, controlándose así los *flip-flop* por separado.

## Construcción

El circuito ocupa tan poco espacio que, cuando se utiliza en un modelo, puede ocultarse en el interior. En el caso contrario, necesita una pequeña caja de plástico. El cableado no presenta problemas. Si se pretende situar las luces a cierta distancia del ordenador (por ejemplo en otra habitación o en un modelo móvil) es preferible mantener el circuito cercano al ordenador y prolongar un par de cables hasta los diodos. Los asteriscos de la figura 6.1 muestran dónde debe realizarse esta prolongación. Cuando haya montado el circuito asegúrese de que no existen puentes ni cortocircuitos entre los cables que lo conectan al micro. A continuación, enchúfelo en el puerto de la impresora y conecte la alimentación. Uno de los diodos se encenderá, pero resulta imposible predecir cuál de ellos. El siguiente programa de prueba para Amstrad hace que las bombillas realicen 10 intermitencias:

```
10 OUT 61184,255
20 FOR J=1 TO 20
30 OUT 61184,0: OUT 61184,255
40 FOR K=1 TO 500: NEXT
50 NEXT
```

## Programación

El principal punto a recordar, a la hora de programar, es que para que las luces cambien de estado es necesario hacer pasar el nivel de la línea de salida de alto a bajo. Un cambio de bajo a alto no tiene efecto alguno. Cada vez que las luces deban cambiar será necesario una sentencia doble "bajo-alto", del tipo de la línea 30 del programa anterior. Si tiene otros dispositivos conectados al puerto de la impresora que deban operar independientemente, necesitará hacer unas ligeras modificaciones al programa. Este tema se discutió a fondo en el proyecto 5, el pitador. Desde luego, se pueden realizar programas más complejos. Cuando se utilizan los diodos como indicadores, el ritmo de la intermitencia puede variar de acuerdo con la información que se intenta suministrar. Un interesante proyecto de programación podría consistir en diseñar un emisor de código morse. Resultaría un aparato práctico. El usuario escribiría un mensaje en el teclado y, cuando se pulsara "enter", el micro convertiría éste a código morse; a continuación, transmitiría el mensaje a través de los diodos.

## Componentes

### *Resistencias*

R1, R2 150R (2 unidades).  
R3 1K.

### *Semiconductores*

D1, D2 TIL209 u otro diodo LED similar (dos unidades).

- IC1 7473 doble *flip-flop* J-K con borrado (este proyecto requiere circuitos TTL estándar, no siendo, por tanto, útiles, los circuitos Schottky de baja potencia, serie LS).

*Varios*

Placa de circuito impreso.  
Zócalo de 14 patillas.  
Conector para el puerto de la impresora .

# 7

## Controlador analógico de modelos

Este proyecto puede utilizarse de igual manera que el número 4, pero incorpora múltiples mejoras. El proyecto 4 controla interruptores (los relés). Es ideal para conectar motores, luces, solenoides, sirenas, radios u otros dispositivos eléctricos con dos estados, encendido o apagado. También puede utilizarse para conectar o desconectar resistencias de un circuito (figura 4.2), produciendo un voltaje variable que controla la velocidad de una locomotora, un coche de escalextric o un brazo robot. El circuito funciona bien, pero su principal inconveniente consiste en que sólo se puede conseguir un número limitado de velocidades. Puede resultar un trabajo difícil elegir los valores adecuados de las resistencias que permitan funcionar al modelo a las velocidades deseadas bajo cargas diversas.

Mientras que el proyecto 4 sólo permite producir unas pocas velocidades distintas, este proyecto permite que la velocidad varíe suavemente dentro de un rango determinado. Esto es lo que quiere decir, en el título, la palabra analógico. Es un proyecto ideal para utilizar con robots en los que las velocidades del motor deben controlarse con gran precisión. Si se utiliza este proyecto para control de velocidad, junto con el proyecto 4 para la conexión de encendido o la

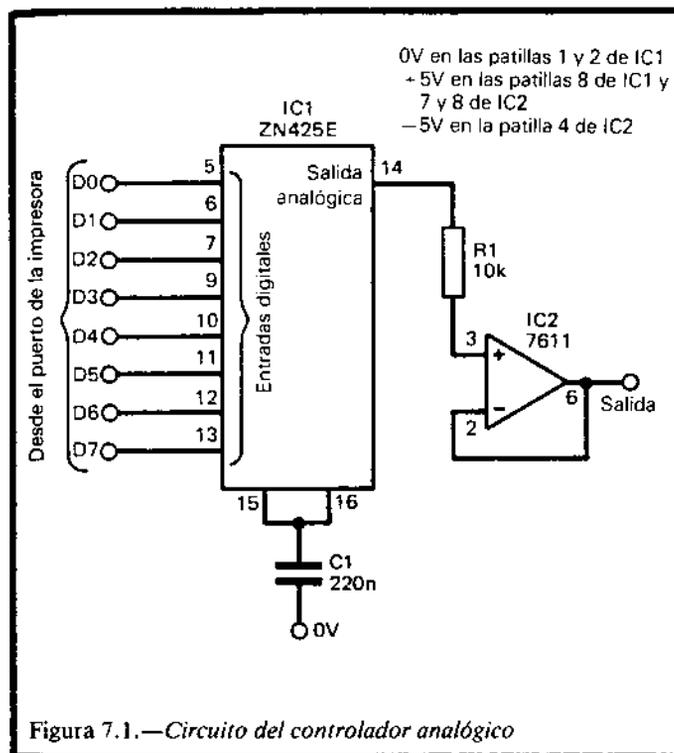
inversión de polaridad, tendremos un versátil sistema para control de muchos tipos de modelos y maquetas.

## Funcionamiento

El circuito está diseñado en función del convertidor digital/analógico ZN425E de 8 bits (Fig. 7.1). Utilizando el puerto de la impresora, podremos enviar cualquier valor en el rango de 0 a 255 al convertidor. Cuanto mayor sea el valor enviado, mayor será el voltaje que aparezca en la patilla (patilla 14). Para un valor 0 (todas las líneas bajas), el voltaje de salida será de 0V; para un valor de 255 (todas las líneas altas), el voltaje será de 1.27V. Para valores dentro de este rango, el voltaje es proporcional al valor enviado al integrado. Dijimos antes que la salida varía suavemente en el rango dado ( 0V-1.27V ), pero no es estrictamente cierto. En realidad, el voltaje varía en 256 pasos diferenciados. Pero, dado que la diferencia entre pasos es sólo de  $1.27/256 = 0.005V$ , se puede decir que el voltaje varía suavemente a todos los efectos.

La corriente que puede circular por la patilla 14 es muy limitada. Aunque sea muy pequeña la carga aplicada a dicha patilla, el voltaje cae considerablemente. Para proporcionar una corriente superior (hasta 50 mA) debemos utilizar un amplificador operacional (IC2). La figura 7.1 muestra dicho amplificador conectado como seguidor de voltaje de ganancia unidad. El voltaje de salida del amplificador (patilla 6) es exactamente igual al voltaje de entrada (patilla 3). La diferencia estriba en que puede proporcionar hasta 50 mA sin que se produzca caída del voltaje.

IC2 es un amplificador operacional que requiere una fuente de alimentación simétrica de +5V y -5V.



La corriente requerida de la fuente de -5V es tan pequeña que resulta innecesario construir un alimentador independiente. En su lugar, utilizaremos un convertidor de voltaje (IC3, figura 7.2). Se alimenta de la fuente de +5V y produce -5V en el condensador C3.

Un voltaje de +1.27V y una corriente de 50 mA resulta insuficiente para la mayoría de los modelos, así que necesitaremos una forma de utilizar voltajes y corrientes superiores. La figura 7.3 muestra una forma simple de realizarlo. La salida de IC2 se alimenta a un transistor de potencia (Q1). Utilizando el

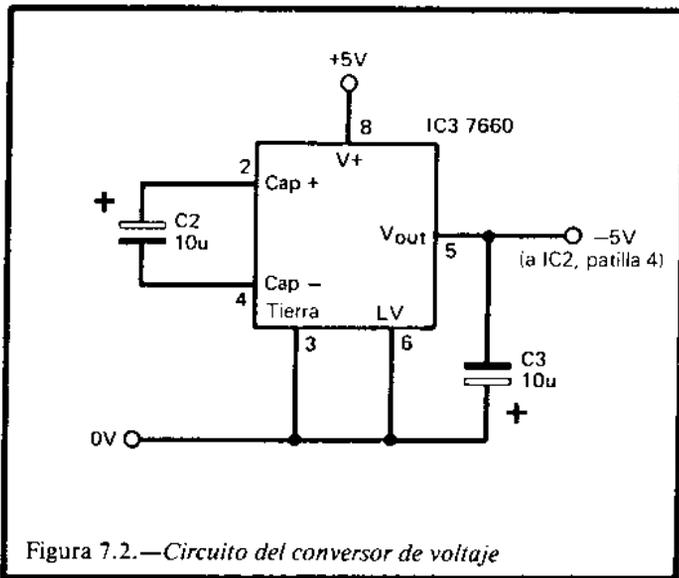


Figura 7.2.—Circuito del convertor de voltaje

transistor recomendado de la figura, el voltaje externo puede ser de hasta 60V, y la corriente de 15A, suponiendo que se monte sobre un radiador adecuado.

Un punto interesante de la figura 7.3 es la conexión entre el emisor de Q1 y la entrada inversora de IC2 (punto A). El propósito de esta conexión es proporcionar una retroalimentación negativa que establezca la alimentación del motor. Esta conexión permite que el motor funcione a velocidad constante, aunque varíe las cargas mecánicas. También permite que el motor funcione sin problemas a velocidades lentas.

Cuando el motor funciona produce una FCEM (fuerza contraelectromotriz) que se opone a la FEM (fuerza electromotriz) que se le aplica. La magnitud de la FCEM es proporcional a la velocidad de giro del motor. Cuanto más rápido gira, mayor resulta la FCEM. Esta es la razón por la que debe de conectarse inmediatamente la alimentación de un motor, si

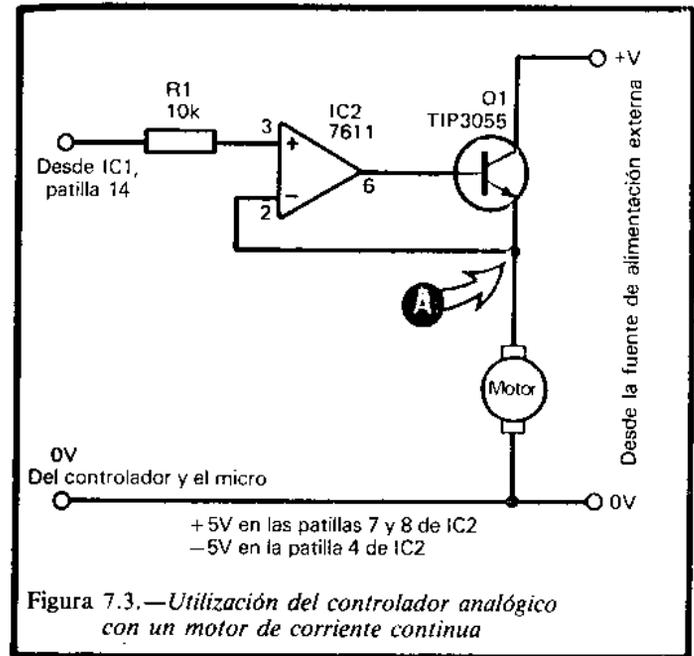


Figura 7.3.—Utilización del controlador analógico con un motor de corriente continua

éste se para por un exceso de carga. Si un motor deja de girar, su FCEM cae a 0 y la FEM, sin contra-restrar, producirá que una gran corriente circule a través de la bobina del motor, probablemente quemándola. Esto no puede ocurrir en el circuito de la figura 7.3. El mecanismo de retroalimentación funciona de la siguiente manera: supongamos que el motor funciona a una velocidad moderada, soportando una carga mecánica que varía con el tiempo; cuando la carga se incrementa, la velocidad del motor empieza a bajar y, por tanto, también la FCEM. La corriente que atraviesa el motor tiende a incrementarse, produciendo una subida de potencial en el punto A. Este efecto reduce el voltaje de entrada en IC2 y, por tanto, la corriente de base de Q1, produ-

ciendo que la corriente del colector descienda. La corriente suministrada al motor decrece a medida que el potencial en el emisor de Q1 empieza a caer. Por tanto, la tendencia a aumentar la corriente que atraviesa el motor queda contrarrestada por la tendencia de ésta a disminuir. De igual manera, si la carga se reduce y la corriente a través del motor tiende a decrecer, el efecto de retroalimentación hará que ésta crezca de nuevo. En otras palabras, la corriente a través del motor y, por tanto, su velocidad, se mantienen constantes, aunque varíen las cargas mecánicas.

## Construcción

Los integrados requieren sólo una pequeña placa de circuito impreso. Tenga cuidado de conectar correctamente la polaridad de C3 cuando lo suelde. Asegúrese de que no hay equivocaciones en las conexiones entre IC1 y el puerto de la impresora. Si su circuito incluye Q1, normalmente se montará en un radiador del tamaño adecuado. Si los requerimientos de potencia son inferiores a los del TIP3055, puede ser suficiente utilizar en su lugar un transistor de potencia media. Por ejemplo, el BD131 puede operar con voltajes de hasta 45V y proporcionar corrientes de hasta 3A. Si sólo se requieren voltajes de hasta 25V y corrientes de 0.5A, se puede utilizar el transistor ZTX300. La figura 7.3 muestra un único transistor controlado por la salida de IC1. Si se quiere controlar varios motores, es posible alimentar varios circuitos similares a los de la figura 7.3 con la salida de un único IC1.

Asegúrese de que no existen cortocircuitos entre las líneas de datos o las de alimentación. Conecte el circuito al ordenador y encienda la alimentación. Compruebe que el voltaje en la patilla 4 de IC2 es de

—5V respecto a la línea de tierra. Conecte un voltímetro entre la línea de 0V y la patilla 6 de IC2.

Envíe una serie de valores al puerto de la impresora, utilizando un programa similar a éste:

```
10 FOR J=0 TO 255 STEP 25
20 OUT 145,J
30 FOR K=1 TO 2000:NEXT
40 NEXT
```

Utilice la instrucción OUT 61184,J si está programando un Amstrad. Cuando ejecute este programa, el voltímetro debe mostrar una serie de voltajes crecientes que comienzan en 0V, y que aumenta en pasos de aproximadamente 0.125V. Para comprobar el funcionamiento de Q1, utilice una fuente de alimentación externa de +6V y conecte una bombilla de 6V en lugar del motor. Ejecute el siguiente programa:

```
10 FOR J=0 TO 255
20 OUT 145,J
30 NEXT
40 FOR J=255 TO 0 STEP -1
50 OUT 145,J
60 NEXT
70 GOTO 10
```

Este programa hace que la bombilla varíe su luminosidad de manera uniforme repetidas veces. En algunas zonas del ciclo el voltaje puede ser demasiado bajo como para hacer que la bombilla luzca.

## Programación

Los programas de prueba de la sección anterior indican claramente cómo programar este proyecto. Para hacer que un motor funcione a una velocidad con-

creta, resulta necesario experimentar para encontrar el valor apropiado que se ha de enviar al puerto de la impresora.

## Componentes

*Resistencias* (carbón, 0.25W, tolerancia 5%)

R1 10k

*Condensadores*

C1 220n poliéster.

C2, C3 10 $\mu$  electrolítico, 10V.

*Semiconductores*

Q1 TIP3055 transistor de potencia NPN (o inferior).

*Circuitos integrados*

IC1 ZN425E conversor digital-analógico.

IC2 7611 amplificador operacional.

IC3 7660 conversor de voltaje.

*Varios*

Placa de circuito impreso.

Radiador.

Zócalos de 8 patillas (dos unidades).

Zócalo de 16 patillas.

Conector para el puerto de la impresora.

Espadines de 1 mm. (13 unidades).

Cable.

# 8

## Actuador magnético

Este circuito produce una acción inmediata. Debe desactivarse manualmente y permanece desactivado hasta que el ordenador lo dispara de nuevo. La acción generada por cada disparo depende exactamente de la aplicación a la que se destine. Puede utilizarse para abrir el cerrojo de una puerta y, por ejemplo, permitir que el gato salga de madrugada. O quizá, para abrir una puerta una vez teclada una clave de seguridad. Se puede utilizar para volcar una determinada cantidad de comida en el acuario y solucionar el problema de alimentar a los peces el sábado por la tarde cuando usted está fuera. Si está fuera durante el día y el termómetro del proyecto 15 se utiliza para controlar la temperatura del invernadero, el actuador magnético puede activarse cuando la temperatura suba demasiado, abriendo las ventanas o la puerta, o permitiendo que se desenrollen las persianas.

## Funcionamiento

El funcionamiento del actuador depende de una bobina o solenoide. En el eje de la bobina se encuentra una pieza de metal, o núcleo, que puede desplazarse. Cuando un pulso de corriente fluye a través



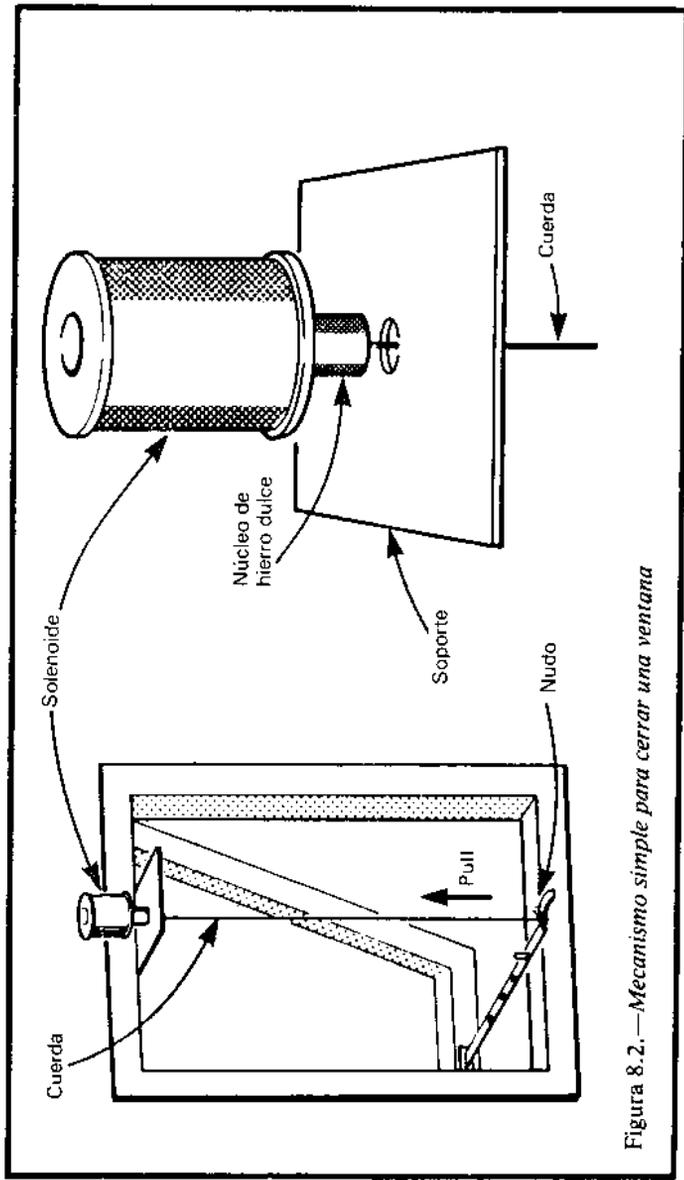


Figura 8.2.—Mecanismo simple para cerrar una ventana

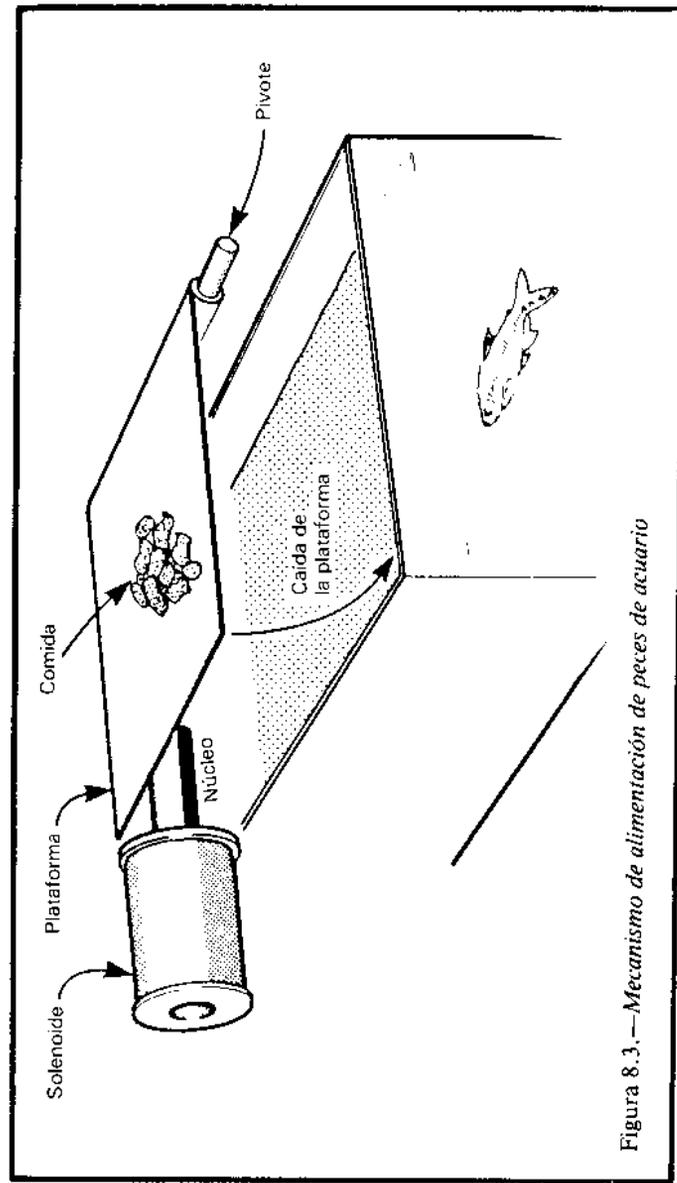


Figura 8.3.—Mecanismo de alimentación de peces de acuario

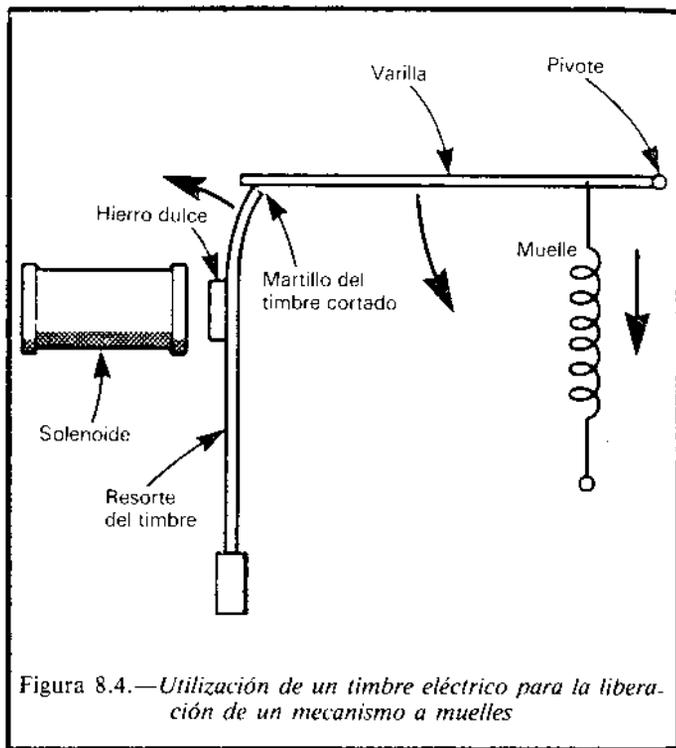


Figura 8.4.—Utilización de un timbre eléctrico para la liberación de un mecanismo a muelles

Una vez que se han descargado los condensadores, no ocurre nada nuevo hasta que el *flip-flop* se desactiva y los condensadores se recargan.

La parte mecánica del proyecto se deja al ingenio del lector y depende mucho de la aplicación que se le quiera dar, la fuerza del solenoide utilizado y la habilidad del lector para trabajos mecánicos. La figura 8.2 muestra cómo se puede utilizar el proyecto para cerrar una ventana. La figura 8.3 muestra cómo se puede descargar una plataforma con alimento en un acuario.

En lugar de un solenoide con un núcleo que se desplaza, el lector quizá prefiera utilizar un solenoide

con un núcleo fijo que atraiga una armadura metálica. Para estos propósitos se puede utilizar un zumbador o un timbre viejo. La figura 8.4 muestra cómo el movimiento de una armadura puede liberar un mecanismo e iniciar un proceso que requiera más fuerza que la pueda producir el solenoide.

## Construcción

Se requieren dos cables para conectar el circuito al micro: la línea de tierra y un cable conectado a una línea de datos del puerto de la impresora. El circuito debe conectarse al micro mediante cables lo más cortos posible. Si el solenoide debe situarse a distancia del micro, los cables deben prolongarse a partir de los puntos marcados por asteriscos (\*) en la figura 8.1. La mejor situación para el circuito es en una caja, cerca del ordenador, en la que se montarán los dos pulsadores y el LED. Si se están utilizando baterías para cargar los condensadores, la caja deberá ser lo suficientemente grande como para contenerlas.

Es mejor diseñar y construir la parte mecánica del actuador antes que la parte electrónica. Es esencial asegurarse de que el solenoide que se pretende utilizar desarrolla la fuerza necesaria para actuar sobre el mecanismo. Se pueden conectar juntos los condensadores en una placa de prueba o mediante cables con pinzas de cocodrilo. A continuación, cargarlos con una fuente de alimentación o con baterías y, seguidamente, hacer una prueba. Si la fuerza resulta insuficiente hay varios métodos para incrementarla:

1. Empezar el movimiento con el núcleo situado más adentro de la bobina (como es lógico, también se reduce la longitud del movimiento).
2. Utilizar un solenoide con un bobinado mayor.
3. Utilizar más condensadores en serie con C1-C3.

4. Cargar los condensadores a un voltaje mayor (suponiendo que el solenoide lo admita).

También es interesante comprobar el acabado de la parte mecánica. Por ejemplo, comprobar que las partes deslizantes lo hagan suavemente, sin fricciones innecesarias. Debido al peso de un mecanismo, una fuerza perpendicular aplicada al núcleo puede hacer que éste se deslice con dificultad.

Una vez montado debe comprobarse que el circuito no tenga puentes y que esté correctamente conectado al ordenador. Una vez conectado, tecléese el siguiente programa:

```
10 OUT 145,255
20 OUT 145,0
```

Utilícese la dirección 61184 si se está programando sobre un Amstrad.

Pulse S2 para desactivar el *flip-flop*; el LED debe encenderse. A continuación, pulse S1 para recargar los condensadores. Mantenga el botón pulsado durante unos dos segundos. Disponga el mecanismo de tal manera que quede listo para ser activado. Cuando todo esté en orden ejecute el programa. El solenoide se activa, el mecanismo opera y el LED se apaga.

## Programación

Las líneas esenciales de programación se han dado en el apartado anterior. El programa concreto dependerá de la aplicación a la cual se destine el proyecto.

## Componentes

*Resistencias* (carbón, 0.25W, tolerancia 20%)

R1 2M2.  
R2 180R.

*Condensadores*

C1-C3 4700 $\mu$  electrolítico (3 unidades).  
C4 100n poliéster.

*Semiconductores*

D1 TIL209 o LED similar.  
D2 1N4001.  
THY1 C106 o tiristor similar.

*Circuito integrado*

IC1 74LS00 cuádruple puerta NAND de dos entradas.

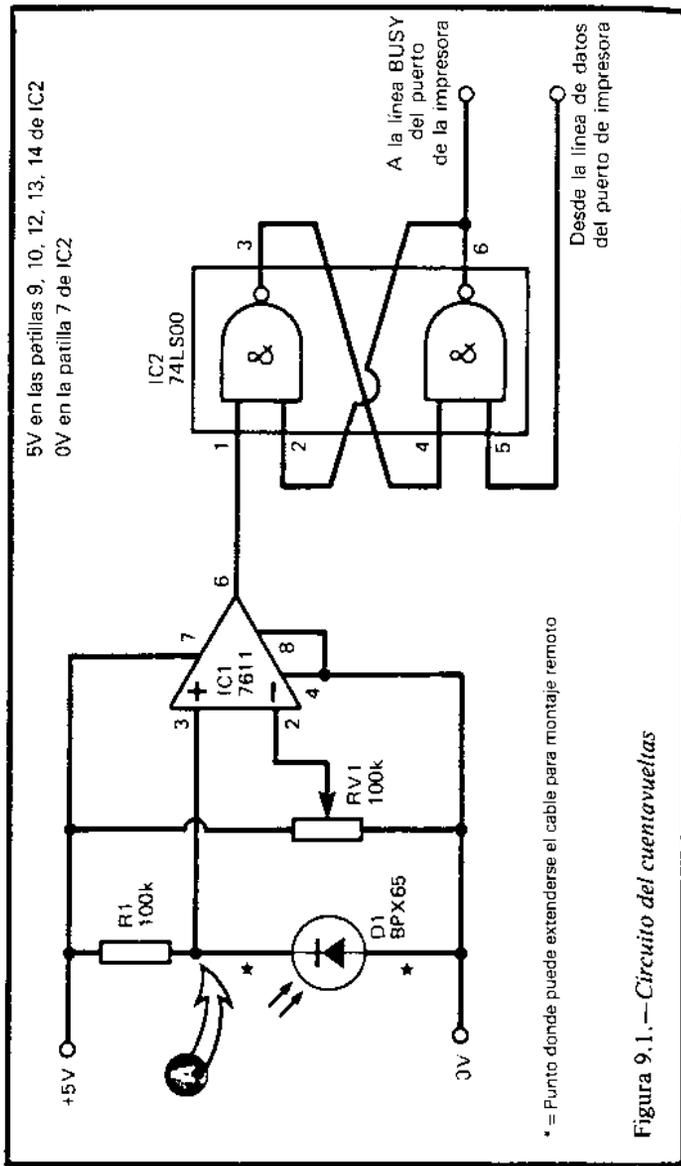
*Varios*

L1 Solenoide con núcleo móvil de hierro dulce que opere a 9V ó 12V (alternativamente, un timbre, un zumbador o un relé).  
S1, S2 Pulsadores (dos unidades).

Circuito impreso.  
Zócalo de 14 patillas.  
Conector para el puerto de la impresora.  
Espadines de 1mm. (9 unidades).  
Cable.

## Cuentavueltas por barrera luminosa

A pesar de que este proyecto se llama cuentavueltas, puede tener muchas otras aplicaciones. En su versión original, es un cuentavueltas para coches de escala, y detecta cuándo un coche pasa a través de un punto dado en una pista. Dependiendo de cómo se programa, el micro puede mantener un registro de las vueltas efectuadas, del tiempo invertido o de ambas cosas a la vez. A partir de esta información puede calcular y mostrar la velocidad media por vuelta y la velocidad media de la carrera. Dado que el dispositivo funciona por la interrupción de un rayo de luz, no importa qué objeto es el que lo interrumpe. Se puede utilizar igualmente bien en carreras de bicicletas o caballos, y en muchos tipos de competiciones atléticas. De igual manera, tampoco tiene por qué contar siempre el número de veces que el mismo objeto atraviesa el rayo; también puede contar cuántos objetos distintos lo atraviesen. Puede contar el número de personas que entra en una habitación, los coches que entran en un aparcamiento o los objetos que mueve una cinta transportadora. Funciona tan rápido que puede contar objetos (o cualquier otra causa de interrupción del rayo) que, por su velocidad, no pueda discernir el ojo humano. Otro tipo de aplicaciones son aquellas en las que queremos que el



ordenador espere a que ocurra una única interrupción del rayo. Un ejemplo típico de estas aplicaciones es la detección de intrusos. Un rayo de luz dispuesto a lo largo de un corredor es interrumpido por el intruso. El micro lo detecta y hace sonar la alarma. También puede programarse para detectar la puesta de sol y encender el alumbrado exterior.

## Funcionamiento

La luz es detectada por un fotodiodo (D1, figura 9.1). Este recibe la luz de una fuente situada a distancia. Puede ser necesario utilizar una bombilla especial que focalice un rayo bien definido a través de la pista o corredor, pero no siempre es esencial. Si se coloca un flexo con una bombilla de 60W en un lado del pasillo y el fotodiodo en el otro, apuntando a la bombilla, la sombra de una persona que pase entre ellos es suficiente para disparar el sensor. A menor escala (por ejemplo, con los coches de escalextric o las maquetas de trenes), una bombilla de 60 mA, situada a 10 cm. o poco más, es perfectamente suficiente como fuente de luz. El único punto importante a tener en cuenta, además, es que otras bombillas cercanas no iluminen el fotodiodo cuando la luz proveniente de la bombilla-fuente es supuestamente interrumpida.

Los fotodiodos funcionan bien con luz de bombillas de filamento y son relativamente insensibles a la luz de tubos fluorescentes. Funcionan bien con la luz del día, pero dado que la intensidad de la luz del día varía, dependiendo de la nubosidad, no resulta una fuente válida.

El fotodiodo se conecta de tal manera que quede polarizado en inverso. La corriente inversa crece proporcionalmente a la cantidad de luz que influye en el fotodiodo. A medida que la corriente crece, la

diferencia de potencial en el diodo crece, así que el potencial en el punto A también crece. El amplificador operacional IC1 compara el potencial en el punto A con el potencial estándar generado por el potenciómetro RV1. Ajustaremos el potencial estándar de tal manera que sea ligeramente inferior al potencial en A cuando el rayo de luz no está cortado. La pequeña diferencia de potencial es amplificada y la salida de IC1 crece bruscamente a 5V (alto). Cuando se rompe el rayo, la corriente a través de D1 decrece, así como el potencial en D1. Esto hace que el potencial de A caiga también hasta quedar por debajo del nivel de referencia generado por RV1. La diferencia de potencial puede seguir siendo pequeña, pero ahora la diferencia tiene voltaje opuesto. Consecuentemente, la salida de IC1 cae rápidamente a 0V (bajo). Se puede ajustar RV1 en función de la luminosidad y la distancia a la que esté situada la fuente de luz.

La salida de IC1 alimenta un *flip-flop* (IC2). El *flip-flop* se desactiva mediante un pulso bajo en la línea de datos del puerto de la impresora. Al decir "un pulso bajo" queremos indicar que la línea de datos se encuentra en nivel alto, cae brevemente a nivel bajo y pasa de nuevo a nivel alto. La forma en que esto se hace se explica más adelante. Cuando el *flip-flop* está desactivado, la salida de la patilla 6 de IC2 está alta; esto hace que la entrada BUSY del puerto de la impresora esté en nivel alto. Si el ordenador está programado para utilizar la instrucción LPRINT (o PRINT#8 si se trata de un Amstrand), la señal BUSY le hace esperar. El ordenador no hace nada hasta que se interrumpe el rayo de luz. Cuando esto ocurre, aunque sólo sea por unos milisegundos, el *flip-flop* se dispara y cambia de estado. Su salida pasa a nivel bajo, lo que hace que el ordenador continúe con el programa. La primera acción es registrar el tiempo. A continuación, desactiva el *flip-flop* y espera de nuevo a que se interrumpa el rayo.

## Construcción

El circuito se sitúa en una pequeña caja con el potenciómetro montado. El fotodiodo puede ocultarse dentro de la caja, haciendo un orificio que permita entrar luz. Esta disposición permite apantallar la luz que llega de otras fuentes. En determinadas aplicaciones puede ser más conveniente tener la fuente de luz y el fotodiodo a cierta distancia del micro, quizá en otra habitación. Si esto ocurre, el circuito debe estar cerca del micro, y los cables que conducen al fotodiodo deben prolongarse desde los puntos marcados con asterisco (\*) en la figura 9.1.

El circuito requiere tres conexiones con el puerto de la impresora: la tierra, la línea de BUSY y una de las salidas de datos. A efectos de programación, supondremos que está conectada la salida de datos D0.

No existen dificultades en el montaje del circuito. Cuando esté acabado, conecte la fuente de alimentación (5 ó 6V) y mida con un voltímetro en el punto A. Apunte con el fotodiodo hacia una fuente de luz cercana. Cuando coloque su mano entre la bombilla y el fotodiodo, el voltaje en A debe crecer. A continuación, conecte el voltímetro en la salida de IC1. Ajuste RV1 hasta determinar el punto exacto en el que la salida pasa a 5V. La interrupción del rayo debe hacer que pase bruscamente a 0V. Finalmente, conecte el voltímetro a la salida del *flip-flop* (patilla 6, IC2). Conecte brevemente la patilla 5 a 0V para desactivarlo. Su salida pasará a bajo (cerca de 0V). Cuando el rayo se interrumpe, la salida pasa a alto (2.5V o más). El circuito está ahora listo para ser probado con el micro. A continuación, se da un programa de prueba para el Amstrand:

```
10 OUT 61184,0:OUT 61184,1
20 PRINT#8," ESPERA"
```

```

30 T1=TIME
40 FOR J=1 TO 500: NEXT
50 OUT 61184,0: OUT 61184,1
60 PRINT#8, "ESPERA"
70 T2=TIME
80 PRINT "EL TIEMPO FUE DE "+(T2—
    T1)/300;" SEGUNDOS"

```

En máquinas MSX utilice 145 en lugar de 61184, y la instrucción LPRINT en lugar de PRINT#8. En la línea 80 el divisor debe ser 50, en lugar de 300.

La línea 10 produce un pulso bajo para desactivar el *flip-flop*. La línea 20 intenta que la impresora (que no existe) escriba "ESPERA". Esta es una acción en falso y el mensaje nunca se escribirá. Hemos utilizado la palabra "ESPERA" para recordarnos que esta sentencia hace esperar al ordenador hasta que se interrumpe el rayo. La acción es similar al cronómetro oficial de una carrera esperando a que se produzca la salida (o, más tarde, la llegada).

Tan pronto como el rayo se interrumpe, el programa continúa. En ese momento, el tiempo se registra en T1. Hay un pequeño retardo para permitir que se recupere el rayo. Esto permite que un coche de escalextric atraviese el rayo por completo. Resulta esencial este retardo porque, si no, no es posible desactivar el *flip-flop* cuando el rayo permanece todavía interrumpido. La longitud del retardo debe ajustarse en otros programas. Debe ser suficientemente larga como para que se recupere el rayo, pero no tanto como para que el coche dé una vuelta completa. Tan pronto como acaba el retardo, el micro se pone a esperar de nuevo a que se vuelva a interrumpir el rayo. Tan pronto como ocurre, el segundo tiempo se registra en T2. La línea 80 calcula la diferencia entre ambas interrupciones y las convierte en segundos.

Conecte el cuentavueeltas al micro, encienda el circuito y la bombilla, asegurándose de que la luz de

ésta incida en el sensor. Ejecute el programa. El programa no debe hacer nada. Si quiere seguir paso a paso lo que el programa hace (o no hace) escriba TRON antes de ejecutarlo. Se verá que el programa avanza hasta la línea 20 y, a continuación, espera. Interrumpa el rayo por un instante. Espere unos segundos (o más) e interrúmpalo de nuevo. Cuando se produzca la segunda interrupción, el ordenador escribirá la diferencia en tiempo que las separa, y terminará el programa.

## Programación

Los puntos principales de la programación han sido tratados en la descripción del programa de prueba. Ese programa muestra cómo medir un único intervalo. El siguiente, servirá para medir 10:

```

90 FOR J=1 TO 500:NEXT
100 VUELTA=VUELTA+1
110 TOTAL=TOTAL+T2—T1
120 IF VUELTA<10 THEN GOTO 10
130 CLS: PRINT "FIN DE LA CARRERA"
140 PRINT "VUELTA MEDIA REALIZADA
    EN ";TOTAL/3000;" SEGUNDOS"

```

Después de 10 vueltas, la pantalla se borra y se escribe el tiempo medio de las vueltas. Quizá se pueda añadir una línea que calcule la velocidad media. Si dentro del bucle se incluye una línea que calcule el tiempo de cada vuelta, se puede encontrar también la vuelta más rápida y escribir cuál fue.

Resulta esencial que los bucles de retardo de las líneas 40 y 90 sean de la longitud adecuada. Los valores hasta los que cuenta J deben ajustarse según la longitud y la velocidad de los coches. Otra alternativa consiste en utilizar una entrada de datos del puerto

de *joystick* para detectar cuándo se ha desactivado el *flip-flop*. Si en lugar de conectar el *flip-flop* a la salida BUSY, conectáramos la otra salida (patilla 3) a la línea D0 del puerto de *joystick*, podríamos saber el estado del *flip-flop* utilizando STICK(1) o JOY(0). La forma de hacerlo se muestra en el siguiente programa:

```
10 VUELTA=0
20 OUT 61184,0: OUT 61184,1
30 IF JOY(0) AND 1=0 THEN GOTO 20
40 VUELTA=VUELTA+1: CLS: PRINT
   VUELTA
50 OUT 61184,0: OUT 61184,1
60 IF JOY(0) AND 1=0 THEN GOTO 50
70 GOTO 30
```

Fíjese en el uso de la función AND, que determina el estado de la línea D0 sin tener en cuenta el estado de las otras líneas. En el programa anterior, la línea 20 desactiva el *flip-flop*, por si se encontraba activado cuando se produjo la conexión. La línea 30 espera a que el *flip-flop* se active, es decir, que la salida de la patilla 3 pase a nivel bajo. Esto indica que un coche (o cualquier otro objeto) ha interrumpido el rayo. El programa pasa a la línea 40, que incrementa y escribe la variable VUELTA. En la línea 50, el micro desactiva el *flip-flop* y espera hasta que se produzca una nueva interrupción. Dado que los micros son más rápidos que la mayoría de los objetos, es muy probable que, llegado este estado, el rayo se encuentre todavía interrumpido. Si esto es así, el *flip-flop* no se activará y el micro permanecerá en un bucle (líneas 50 y 60), e intentará activar el *flip-flop* continuamente hasta que lo consiga. Cuando esto ocurre es porque el coche ha salido del rayo, el *flip-flop* se ha desactivado y la línea 60 detecta qué ha ocurrido. El ordenador pasa entonces a la lí-

nea 70, y de nuevo a la 30 para esperar una nueva interrupción.

Los programas de medición de intervalos necesitan dos rutinas similares a las líneas 30, 50 y 60, para detectar la interrupción y desactivar el sensor. Cuando se produce la primera interrupción se registra el tiempo utilizando el reloj interno. Lo mismo se hace con la segunda interrupción. La diferencia es el intervalo. El programa anterior puede contar y medir los intervalos de un único coche. Si se desea tener dos coches controlados por dos dispositivos distintos, es esencial utilizar el puerto de *joystick* para leer el estado de los *flip-flops* de una manera similar a la antes descrita. Los *flip-flops* se desactivan por una salida de datos en el puerto de la impresora, como ya se dijo. Se pueden conectar a D0 y D1. Las salidas de los *flip-flops* irán a dos entradas de datos del puerto de *joystick*. Se pueden utilizar las líneas D0 y D1, pero si se utiliza un MSX será mejor utilizar D4 y D5. El programa será más complejo que el antes listado, ya que el ordenador tiene que mantener el control de dos coches simultáneamente. Si se utiliza un MSX, pueden utilizarse con buenos resultados las instrucciones ON STICK y ON STRIG, que permiten que el ordenador pase el control a diferentes subrutinas en función del rayo que se haya interrumpido.

## Componentes

### Resistencias

R1 100K carbón, 0.25V.  
RV1 100K potenciómetro.

### Semiconductor

D1 BPX65 o cualquier otro fotodiodo.

- IC1 7611 amplificador operacional CMOS.
- IC2 74LS00 cuádruple puerta NAND de dos entradas.

*Varios*

- Placa de circuito impreso y caja.
- Botón para RV1.
- Zócalo de 8 patillas.
- Zócalo de 14 patillas.
- Conector para el puerto de la impresora (opcional para el puerto de *joystick*).
- Espadines de 1 mm. (9 unidades).
- Cable.
- Bombilla.

# 10

## Fotoflash

En realidad, este proyecto tiene dos secciones que se pueden utilizar juntas o cada una de ellas por separado. Una sección permite que el micro dispare un flash fotográfico y la otra es un detector de sonidos. Las dos juntas se pueden utilizar para disparar un flash cuando se detecta un ruido. Se puede aplicar a la fotografía de alta velocidad. La cámara se apunta hacia un globo hinchado (o incluso una botella de cristal). Se apagan las luces y se abre el obturador. A continuación, se pincha el globo o se golpea la botella con un martillo, y el ruido resultante dispara el flash. El resultado es una fotografía del globo estallando o de la botella rompiéndose. Desde luego, el sensor de ruidos puede conectarse directamente al circuito disparador del flash, pero conectándolos a través de un micro se puede programar el tiempo de retardo entre el sonido y el disparo, permitiéndose así obtener fotografías a diferentes intervalos de tiempo, después del impacto inicial. Teniendo más de un flash, el micro puede dispararlos en sucesión, produciéndose una impresión múltiple.

Otra aplicación puede ser la fotografía nocturna de animales. Un ruido realizado por un animal dispara el flash. El circuito de detección de ruidos puede utilizarse por sí solo de muchas maneras. Es particu-

larmente sensible a ruidos bruscos (palmas y chasquidos de dedos) y a los de tono agudo. Esta parte del proyecto puede utilizarse, junto con el proyecto cuatro, para controlar maquetas de trenes y coches de escalextric, de tal manera que arranquen o se paren mediante silbidos, palmadas o incluso órdenes verbales. El proyecto ocho, actuador magnético, también puede ser controlado acústicamente a través de un micro. La sección de fotoflash también puede utilizarse por sí sola, quizá para proporcionar retardos cuando uno quiera aparecer en una fotografía de grupo. El proyecto nueve también puede ser útil junto con el fotoflash. En lugar de disparar el flash por sonido se puede disparar por interrupción de un rayo de luz.

## Funcionamiento

La sección de flash se controla mediante una salida de datos que se conecta a un *flip-flop* (Fig. 10.1). El *flip-flop* se desactiva pulsando S1, pasando su salida a nivel bajo. Cuando el dispositivo recibe un pulso bajo de la salida de datos, se activa. Su salida pasa a nivel alto, haciendo que el transistor Q1 conduzca y se active el relé RLA1, cerrándose así sus contactos. A estos contactos se encuentran conectados los terminales de disparo del flash, lo que hace que éste se dispare. Hay que tener en cuenta que la potencia que utiliza el flash proviene de la cámara y no del circuito. Los contactos del relé realizan la misma función que los contactos internos de la cámara encargados del disparo de flash. Una vez que el flash se ha disparado, el *flip-flop* debe desactivarse manualmente antes del siguiente destello. Es posible prescindir de S1 y conectar la patilla 5 de IC2 directamente a otra salida de datos del ordenador. Esto permitiría desactivar el *flip-flop* de una manera automática. Si se está

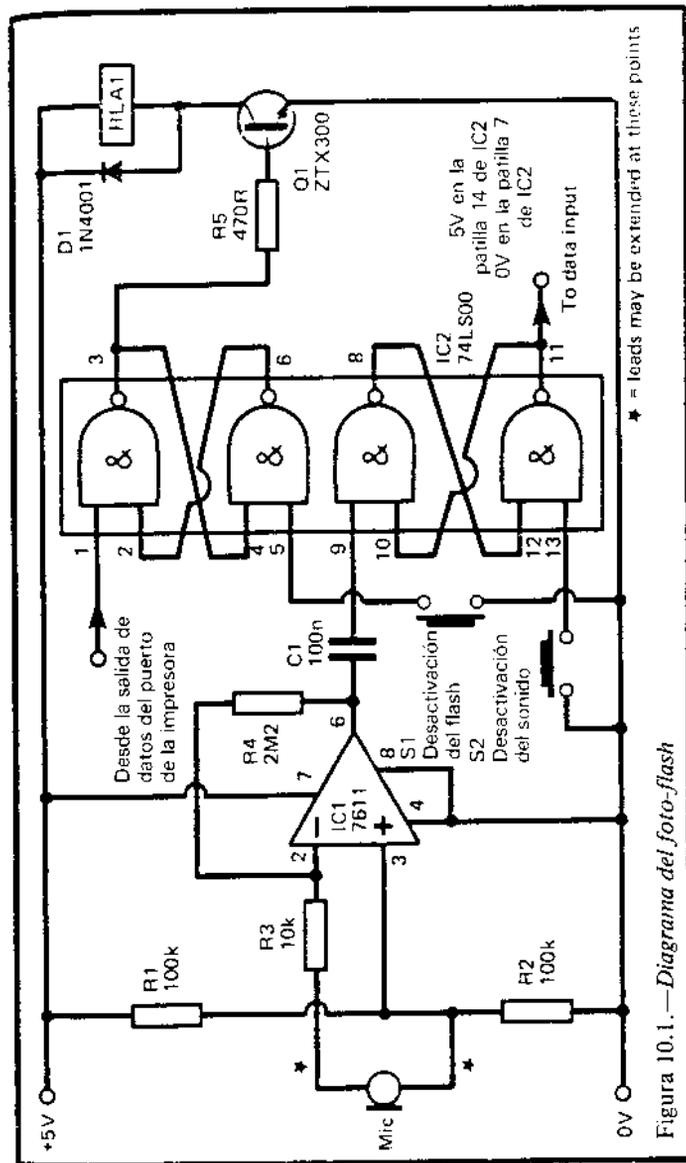


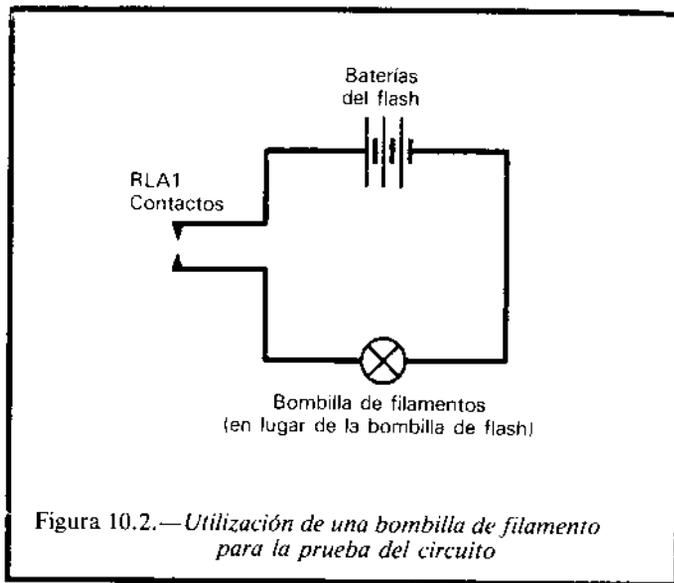
Figura 10.1.—Diagrama del foto-flash

utilizando un flash de lámparas de magnesio no tiene mucho sentido, pero puede resultar imprescindible si se utiliza un flash electrónico y se desea realizar exposiciones múltiples, mediante varios disparos separados por cortos intervalos de tiempo. El sensor de ruidos toma la señal de un micrófono de cristal y la amplifica mediante el operacional IC1. En ausencia de ruido, la salida del amplificador es del orden de 2.5V. Cuando se detecta algún ruido, la salida oscila y toma valores superiores e inferiores. Las bajadas de voltaje actúan como los pulsos de nivel bajo necesarios para activar el *flip-flop*. Su salida, que se encuentra habitualmente a nivel alto, pasa a nivel bajo. Este cambio puede leerse mediante el micro. Fíjese que el *flip-flop* permanece bajo una vez que ha sido activado, así que no importa que el micro no esté leyendo los datos en el momento que el ruido ocurre. El *flip-flop* se desactiva pulsando S2. En la aplicación original de este proyecto puede valer la desactivación manual, pero en alguna otra aplicación puede ser conveniente que la desactivación la realice el micro. Esto se puede hacer reemplazando S2 por una conexión entre la patilla 13 de IC2 y una salida de datos. Esto permite que el micro realice la desactivación y el circuito esté inmediatamente dispuesto para la detección de un nuevo ruido. El dispositivo necesita un mínimo de tres líneas para la conexión con el micro: la tierra, una línea de entrada y una de salida. La entrada de datos se puede realizar por cualquier línea del puerto de *joystick*. Se necesitan más líneas si se quiere desactivar alguno de los *flip-flop* automáticamente. El circuito puede situarse en una caja, fuera de la cual se ha de colocar el micrófono. Los dos pulsadores pueden situarse en el panel de la caja. Los terminales del relé se cablearán a un conector situado también en el panel de la caja. A este conector se deberá enchufar exteriormente el cable de disparo del flash.

Como se explicó antes, el circuito consta de dos partes y no es necesaria la construcción de ambas si sólo se está interesado en una de ellas. Si quiere operar con varios flashes es fácil añadir un segundo integrado 74LS00, con el que se pueden construir más *flip-flops*.

Cuando esté acabado el montaje y el proyecto esté revisado, puede realizarse la primera prueba. Conéctelo a una alimentación de 5 ó 6V. Si alternativamente se cursa S1 y se toca con un cable de tierra (0V) la patilla 1 de IC2, debe oírse al relé conectarse y desconectarse. A continuación, conecte un voltímetro a la salida de datos (patilla 11 de IC2), pulse S2 y compruebe que la salida pasa a nivel bajo. Haga un ruido cerca del micrófono y verifique que la salida pasa a nivel alto (2.5V o más). El circuito responde mejor a sonidos agudos o palmadas. Pruebe a "chascar" sus dedos a unos 50 cm del micrófono. Incluso con un leve ruido el circuito debe dispararse. Si falla, golpee ligeramente el micrófono. Si sigue fallando, es evidente que el circuito no funciona. El fallo puede ser debido a una falta de nivel de salida del micrófono. Para solucionarlo, mejore la amplificación sustituyendo la resistencia R4 por una de mayor valor, quizá 3M3 ó 4M7.

El circuito puede ahora probarse a través del micro. Conéctelo al puerto de la impresora y encienda el micro. Escriba OUT 145,255 (o OUT 61184,255) y pulse RETURN. Esto hace que todas las líneas pasen a nivel alto. Pulse ambos botones de desactivación. Si tiene un flash electrónico puede utilizarlo durante la prueba. Si lo que utiliza son bombillas de flash desechables, la prueba le puede salir un poco cara. En su lugar utilice una bombilla normal de bajo voltaje como se muestra en la figura 10.2. En el Amstrad, la orden de destello es OUT 61184,0: OUT 61184,255. En el MSX deberá utilizarse la dirección 145.



## Programación

El control del circuito es relativamente simple y ya se han descrito varios ejemplos de sus usos posibles. A continuación, se da un programa que dispara el flash un corto período de tiempo después de la detección del ruido. Funciona en el Amstrad:

```

10 OUT 61184,255
20 IF JOY(0) AND 1=1 THEN GOTO 20
30 IF JOY(0) AND 1=0 THEN GOTO 30
40 T=TIME+30
50 WHILE T>TIME: WEND
60 OUT 61184,0: OUT 61184,255
70 PRINT "FOTOGRAFIA REALIZADA"

```

Este programa supone que la salida del *flip-flop* se conecta a la línea D0 del puerto de *joystick*. La salida de datos puede ser cualquiera del puerto de la impresora. La línea 10 fija todas las salidas del puerto de la impresora a nivel alto, de tal manera que el *flip-flop* del flash pueda ser desactivado. La línea 20 espera hasta que se ha pulsado el botón de desactivación del sensor de ruidos, es decir, cuando la línea de datos pasa a nivel alto. La línea 30 espera a que pase de nuevo a nivel bajo, siendo ésta la indicación de que se ha detectado un sonido.

Las líneas 40 a 50 retardan el tiempo deseado el disparo del flash. Con un valor de 30, el retardo es aproximadamente de 0.1 segundos. Alterando este valor se pueden conseguir retardos de distinta duración. Utilizando un ordenador MSX, las líneas 40 y 50 deberán ser:

```

40 T=TIME+5
50 IF T>TIME THEN GOTO 50

```

Tan pronto como se haya realizado el retardo, la línea 60 hace que la salida de datos del puerto de la impresora tome el nivel bajo y, a continuación, el alto. Este breve pulso bajo activa el *flip-flop* y dispara el flash. A continuación, se escribe en la pantalla el mensaje: "FOTOGRAFIA REALIZADA".

Otro refinamiento especialmente útil, si está utilizando el dispositivo para fotografiar animales que se pueden acercar a la cámara, es insertar una larga pausa antes de que el programa se haga operativo para poder alejarse. En este caso, sería una buena idea que el ordenador tuviera el sensor desactivado, de tal manera que los ruidos producidos durante la marcha no lo disparasen.

Si está utilizando el ordenador para desactivar el sensor, después de que se ha detectado un sonido, el programa debe comprobar que, efectivamente, la des-

activación se ha llevado a cabo. Si se intenta una desactivación mientras el ruido todavía continúa, ésta será infructuosa (véase la página 81).

Conector para el puerto de la impresora.  
Circuito impreso.  
Espadines de 1mm. (8 unidades).  
Cable .

## Componentes

*Resistencias* (carbono 0.25W, tolerancia 5 %)

R1, R2 100K (dos unidades).  
R3 10K.  
R4 2M2.  
R5 470R.

*Condensador*

C1 100n poliéster

*Semiconductores*

Q1 ZTX300 o transistor NPN similar.  
D1 1N4001.

*Circuitos integrados*

IC1 7611 amplificador operacional CMOS.  
IC2 74LS00 cuádruple puerta NAND de dos entradas.

*Varios*

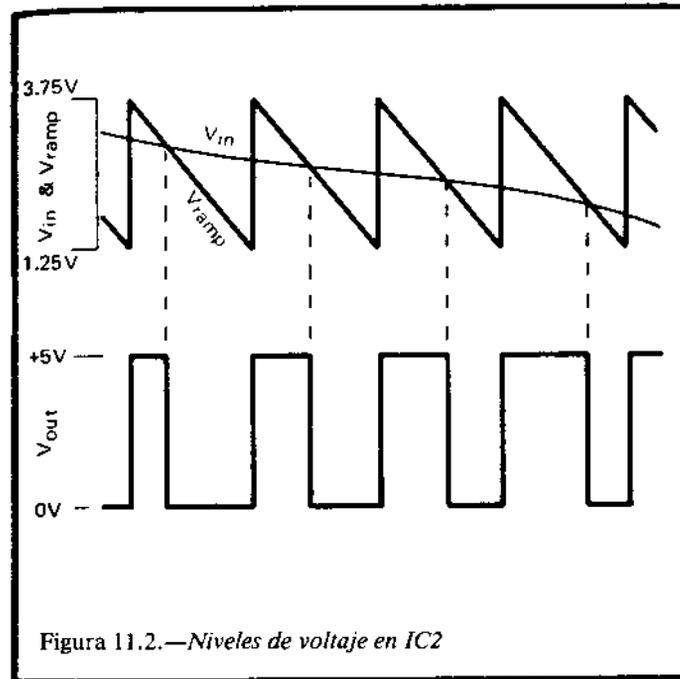
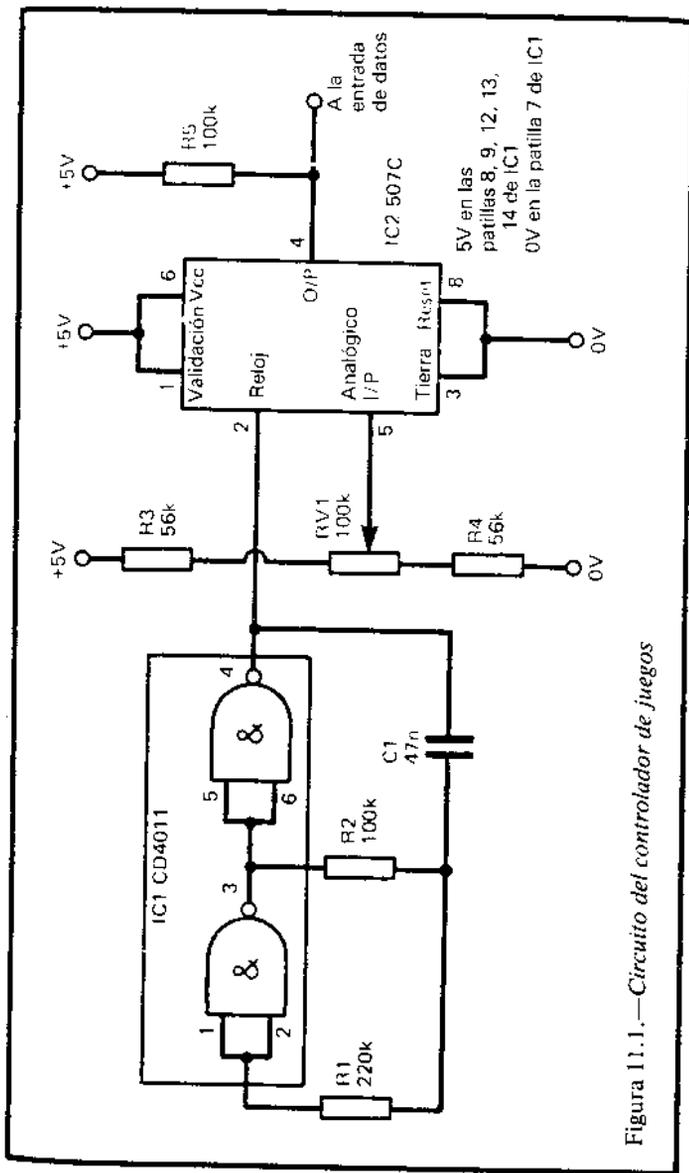
RLA1 Relé de un circuito, una posición, 6V.  
S1, S2 Pulsadores (dos unidades).  
MIC Micrófono de cristal.  
Zócalo de 8 patillas.  
Zócalo de 14 patillas.  
Conector para flash.

# Controlador de juegos

Un controlador de juegos añade una nueva faceta a la informática recreativa. Consiste en una pequeña caja con un botón de control. Moviendo el botón en uno o en otro sentido, se puede mover un "murciélago" por la pantalla, apuntar un "cañón láser", conducir un "coche de carreras" o tomar parte activa de mil y una formas en cualquier juego. Se pueden construir dos o más circuitos y operarlos a voluntad. Otra posibilidad es albergar el proyecto en la misma caja que el teclado de 5 teclas del proyecto 3.

## Funcionamiento

El botón de control altera la posición de una resistencia variable (RV1, figura 11.1). Este está conectado entre dos resistencias de 56K y la cadena de resistencias está conectada entre 0V y 5V. A medida que el contacto móvil de RV1 se mueve de un extremo a otro de su recorrido, el voltaje en él varía entre 1.25 y 0.75V, más o menos. El voltaje varía suavemente en este rango; por ello, decimos que es un valor analógico. Los micros no son capaces de aceptar entradas de este estilo. Un micro entiende sólo dos



tipos de entrada, alta y baja. Así que IC2 convierte el voltaje analógico, que varía suavemente, en uno que se alterna entre alto y bajo. Este integrado funciona con una señal de reloj construida con dos puertas NAND de IC1. Los pulsos de reloj producen un voltaje ( $V_{rampa}$ ) dentro de IC2 que decrece en forma de rampa de 3.75 a 1.25V. Esto realmente ocurre en una serie de 128 pasos, pero, a todos los efectos prácticos, es una rampa continua, como se muestra en la figura 11.2. Siempre que el voltaje analógico de la entrada es mayor que  $V_{rampa}$ , la salida de IC1 es alta. En caso contrario, es baja. El resultado es que tenemos una onda de una frecuencia fija (128 avo) de la frecuencia del reloj, pero que tiene una relación

variable entre el tiempo que se encuentra en nivel bajo y el que se encuentra en nivel alto.

A medida que la zona alta aumenta de tamaño, la zona baja disminuye, y viceversa. El micro está programado para calcular cuánto tiempo se encuentra en alto y cuánto en bajo, y, a partir de esta información, puede calcular el nivel de voltaje de la entrada analógica. Llamaremos a IC2 un conversor de voltaje a tiempo. No es un auténtico conversor analógico digital, ya que el tiempo, como el voltaje, es una magnitud analógica. Pero el tiempo puede representarse por niveles altos y bajos, y, en esta forma, puede ser entendido y medido por el ordenador.

## Construcción

El dispositivo necesita dos líneas para conectarse al ordenador: la tierra y una línea de entrada de datos del puerto de *joystick*. El circuito puede montarse en una pequeña caja de plástico, preferiblemente con la tapa inclinada. La resistencia variable se monta en la tapa y el resto del circuito se ensambla en una pequeña placa de circuito impreso contenida en el interior de la caja. Si tiene un osciloscopio, puede comprobar la salida de reloj y la salida de IC2 después del ensamblado. El reloj tiene una frecuencia de aproximadamente 1KHz. La exactitud no es importante. La salida de IC2 tiene una frecuencia aproximada de 8 Hz. A medida que se altera la posición de RV1, la relación alto-bajo varía aproximadamente de 1:20 a 20:1. Si la salida está continuamente a nivel bajo, cuando RV1 está en un extremo de su recorrido, es que el voltaje de entrada a caído por debajo de los 200 mV, que es el mínimo requerido. Esto puede ocurrir si las resistencias no se ciñen a los valores especificados. Para compensarlo, incremente R4. Si la salida de IC2 permanece alta en el otro extremo del

recorrido, es que el voltaje analógico excede todos los valores de Vrampa. Incremente el valor de R3. Sin un osciloscopio, las salidas pueden detectarse conectando un auricular de cristal (por ejemplo, de una radio) a la salida y a la línea de 0V, con un condensador de 100nF, aproximadamente, en serie. Se debe oír una nota alta cuando se conecte a IC1 y un zumbido de baja frecuencia cuando se conecte a IC2. Una vez realizada la construcción, compruebe los cortocircuitos entre las líneas y conéctelo al micro. A continuación, enchufe el controlador en el decodificador. Para controlar su salida se necesita un programa que lea las entradas repetidas veces y las escriba. El programa para el Amstrad es :

```
10 PRINT JOY(0)
20 GOTO 10
```

Cuando este programa funciona, aparecen en pantalla una serie de valores que cambian regularmente entre 1 y 0. La proporción entre los unos y los ceros varía en función de la disposición de RV1. Si sólo se obtienen unos o ceros es que los voltajes se encuentran fuera de rango y deben modificarse los valores de R3 y R4, como se indicó antes.

## Programación

El uso más frecuente de este proyecto es mover objetos por la pantalla. Hay dos formas de acceder a la salida del circuito. Un método consiste en medir el tiempo durante el que la salida es alta. Este es el método de tiempos. Para hacer esto con exactitud, es necesario que el micro funcione rápido, por lo que resulta esencial trabajar en lenguaje máquina. Trataremos este método más adelante. El otro método consiste en leer la entrada un número fijo de veces y

contar cuántas de estas lecturas son unos. Este es el método de muestreo. Los resultados obtenidos son proporcionales directamente a la relación alto-bajo, pero resulta necesario tomar al menos 100 lecturas para obtener unos resultados razonablemente fiables.

Las lecturas deben extenderse sobre varios períodos, con el fin de obtener un muestreo adecuado de los tiempos estudiados. Esto significa que el método es bastante lento. El siguiente programa BASIC para Amstrad utiliza el método de muestreo:

```

10 X=3
20 FOR J=1 TO 100
30 IF JOY(0)>0 THEN X=X+1
40 NEXT
50 X= X/3
60 CLS
70 LOCATE X,3: PRINT ""
80 GOTO 10

```

Este programa permite que el controlador se conecte a cualquier entrada de datos, pero presupone que no hay ningún otro circuito conectado al puerto. Si se pretende conectar otros circuitos, sustituya la siguiente línea:

```
30 IF JOY(0) AND 2=2 THEN X=X+1
```

Esta línea detecta la señal del controlador cuando se conecta a la línea D1. El programa toma 100 muestras en una rápida sucesión. El número de veces que la línea se encuentra alta varía de 5 a 95, dependiendo del controlador. Este es un número demasiado grande para ser utilizado directamente y, como en todo suceso, aparece un cierto error de muestreo. Este aparece principalmente porque no empezamos a muestrear exactamente en el mismo punto de la secuencia de salida en cada ocasión que se ejecuta el programa.

Dividiendo el resultado por 3 (línea 50) obtendremos un valor adecuado para el uso de LOCATE, y al mismo tiempo reduciremos los errores. Cuando se ejecuta el programa, el asterisco (\*) se mueve a lo largo de la línea superior de la pantalla cuando se gira el botón del controlador. Este programa es muy simple y vale únicamente para iniciarse en el uso del controlador. El siguiente paso es incorporar versiones del programa a sus propios programas de juegos. El programa anterior es lento. Si se pretende acelerarlo tomando menos muestras, el muestreo no resulta fiable y el asterisco se mueve de un lado para otro como un tenista esperando a recibir un servicio rápido. Quizá esto añada realismo al juego. Si se quiere un programa todavía más rápido y que funcione, resulta necesario un programa en lenguaje máquina que se base en el método de temporización.

11 00 00	LD DE,0	;pone DE a 0
CD 24 BB	CALLA&BB24	;llamada a rutina <i>firmware</i> para lectura del puerto
A7	AND A	;da valor al indicador cero
28 FA	JRZ A	;salta a A si entrada a nivel alto
CD 24 BB B:	CALL &BB24	;lee el puerto
A7	AND A	;da valor al indicador cero
20 FA	JRNZ B	;salta a B si entrada a nivel bajo
I3	C: INC DE	;cuenta
CD 24 BB	CALL &BB24	;lee el puerto
A7	AND A	;da valor al indicador cero
28 F9	JRZ C	;salta a C si entrada a nivel alto
ED 53 00 A0	LD (&A000),DE	;almacena el contador
C9	RET	;devuelve el control al BASIC

El programa mide la longitud del período bajo realizando un conteo en el par de registros DE del microprocesador. Primero se ponen a cero estos registros. A continuación, se lee la entrada. La subrutina *firmware* (del sistema operativo de la máquina) devuelve en el acumulador del microprocesador los mismos valores que JOY(0), véase la tabla 0.2. A

continuación, se realiza una operación Y lógica (AND) del acumulador consigo mismo, con el fin de que el indicador 0 se active o desactive en función del contenido del acumulador. El siguiente paso del programa es saltar a la etiqueta A si el indicador está activado, esto es, si la entrada está a nivel alto. Por tanto, el microprocesador esperará, metido en un bucle, hasta que la entrada pase a nivel bajo. Si la entrada está a nivel bajo cuando el programa empieza a ejecutarse, el control pasará directamente a la segunda etapa del programa. La segunda etapa es igual que la primera, excepto que ahora espera hasta que la entrada pasa a nivel alto, saltando a B si esto es falso. Los dos bucles evitan que el microprocesador empiece a contar antes de que la entrada pase de bajo a alto, independientemente del estado en que se encuentre cuando el programa empiece a ejecutarse. El programa permanece en el tercer bucle mientras que la entrada permanece en nivel alto, pero con la diferencia de que el registro DE se incrementa en cada vuelta. El microprocesador devuelve el control al BASIC cuando la entrada pasa definitivamente a nivel bajo. Cuando esto ocurre, los registros DE contienen un valor que depende de la longitud del período alto. Justo antes de devolver el control al BASIC, el programa sitúa el valor de este par de registros en una posición memoria. Veamos el programa para el Amstrad que hace uso de la rutina en código máquina:

```

10 MEMORY 40959
20 FOR J=40962 TO 40988:READ X: POKE
  J,X: NEXT
30 CALL 40962
40 PRINT PEEK(40960)+256*PEEK(40961)
50 GOTO 30
200 DATA 17,0,0,205,36,187,167,40,250,205,
  36,187,167,32,250,19,205,36,187,167,
  40,249,237,83,0,160,201

```

La línea 200 contiene el código máquina en formato decimal. Cuando el programa ejecuta la línea 10 reserva la memoria a partir de la posición 40960 (&A000) y hasta el final del espacio direccional. La línea 20 lee los datos y escribe el código máquina en las posiciones 40962 a 40988. Las direcciones 40960 y 40961 se reservan para almacenar el conteo al final de la rutina. Escriba este programa y sávelo antes de ejecutarlo. Cuando está en ejecución, la tecla escape resulta inefectiva, así que será necesario apagar el ordenador para parar la máquina. Desde luego, si se omite la línea 50 simplemente contará una vez y se parará, pero resulta mejor dejar que el programa se repita de tal manera que se puedan investigar los efectos producidos por la variación de RV1. Ejecute el programa. No ocurrirá nada, ya que el programa está esperando que el voltaje en una línea de datos siga la secuencia bajo-alto-bajo. Si no hay nada conectado, las entradas permanecen a nivel alto y el programa espera en el primer bucle. Conecte el controlador al puerto de joystick (en cualquier línea) y conecte su alimentación. Inmediatamente verá una sucesión de números que aparecen en la pantalla. Se genera un número por cada período en el que la señal está alta. Gire RV1 de un extremo a otro de su recorrido. Cuando el terminal deslizante se haya en el extremo positivo (Vin es alto), los números serán pequeños (100-200). La figura 11.2 explica la causa. A medida que se gira el potenciómetro hacia su terminal negativo (reduciendo Vin), el número crece hasta un valor de 10000, aproximadamente. Quizá, situando el potenciómetro en uno de sus extremos, es posible que el ordenador deje de contar. Esto, sin duda, es debido a que la salida que produce IC2 permanece continuamente en nivel alto o bajo. Si esto ocurre, ajuste los valores de R3 o R4 como se explicó en el epígrafe de la construcción. El ritmo con el que aparecen los números en pantalla depende de la

velocidad del reloj (IC1). Los números del orden de varios miles dan una indicación precisa de la posición de RV1, pero para nuestros propósitos también serán aceptables números menores. Se puede incrementar la frecuencia del reloj reduciendo los valores de R2 o de C1. Esto hará que el tiempo de conversión disminuya y que el circuito dé una indicación más rápida, aunque menos precisa. Los ordenadores MSX necesitan una rutina en código máquina distinta:

```

11 00 00      LD DE,0           ;pone a cero los registros DE
DB A2      A: IN A,(162)       ;carga A desde el puerto 2
CB 47      BIT 0,A            ;busca D0
28 FA      JRZ A              ;salta a A si la entrada es baja
DB A2      B: IN A,(162)       ;carga A desde el puerto 2
CB 47      BIT 0,A            ;verifica D0
20 FA      JRNZ B             ;salta a B si la entrada es alta
13         C: INC DE           ;cuenta
DB A2      IN A,(162)         ;carga A desde el puerto 2
CB 47      BIT 0,A            ;verifica D0
28 F9      JRZ C              ;salta a C si la entrada es baja
ED 53 00 F3 LD (&F300),DE     ;almacena el resultado
C9         RET                ;devuelve el control al BASIC

```

El controlador debe conectarse a la línea D0 del puerto de *joystick* número 2 (no al puerto número 1). La entrada del puerto es 255 cuando D0 es alta o 254 cuando D0 es baja. La instrucción BIT determina el estado del bit menos significativo, que es el que muestra el estado de la línea D0. Si se desea conectar el controlador a cualquier otra línea, ésta debe probarse modificando la instrucción BIT para que lea el bit adecuado. El programa para utilizar la rutina es el siguiente:

```

10 CLEAR 200,62208: DEF USR0=62210
20 FOR J=62210 TO 62236: READ X:
   POKE J,X:NEXT
30 X=USR0(0)
40 PRINT PEEK(62208)+256*PEEK(62209)

```

```

50 GOTO 30
200 DATA 17,0,0,219,162,203,71,40,250,219,
        162,203,71,32,250,19,219,162,203,71,
        40,249,237, 83,0,243,201

```

El método para utilizar este programa es exactamente el mismo que el programa anterior dedicado al Amstrad. Recuerde que debe salvar el programa antes de ejecutarlo.

## Componentes

*Resistencias* (carbón, 025W tolerancia 5%)

R1            220K.  
R2, R5        100K (dos unidades).  
R3, R4        56K (dos unidades).  
RV1           100K potenciómetro líneal.

*Condensador*

C1            47n poliéster.

*Circuitos integrados*

IC1           CD4011 cuádruple puerta NAND de  
              dos entradas, CMOS.  
IC2           507C conversor de voltaje a tiempo.

*Varios*

Caja para el montaje del potenciómetro.  
Botón para RV1.  
Placa de circuito impreso.  
Espadines de 1 mm. (6 unidades).  
Zócalo de 8 patillas.  
Zócalo de 14 patillas.  
Conector para el puerto de *joystick*.

## Detector de lluvia

Los restantes proyectos de este libro forman la serie dedicada a la estación meteorológica. Pueden todos ellos conectarse juntos al ordenador y registrar una serie de variables meteorológicas. Son proyectos sencillos y no instrumentos de precisión, pero proporcionan información suficiente para permitir predecir el tiempo. A pesar de que todos estos proyectos tienen aplicación meteorológica, muchos de ellos pueden tener otras aplicaciones. El lector ambicioso puede desear conectar a la vez los seis proyectos. El proyecto 13 requiere tres entradas de datos, pero los restantes sólo una. La tabla 12.1 sugiere un sistema de conexión. Este sistema no proporciona línea para el registrador solar, que se puede conectar en lugar de alguno de los otros proyectos. De todas formas, utilizando el selector de datos (proyecto 18) es posible tener los siete proyectos, o incluso más, conectados simultáneamente. En los ordenadores MSX, la veleta se conecta a las líneas de datos correspondientes a STICK(1), STRIG(0) y STRIG(2). Sólo se pueden utilizar cuatro líneas simultáneamente en un puerto de *joystick* MSX, pero utilizando los dos puertos podremos lograr las ocho líneas necesarias. Si se quieren tener otros proyectos funcionando a la vez, utilice el selector de datos. El detector de lluvia

Tabla 12.1.—*Conexión de la estaca meteorológica a los puentes de Joystick*

Proyecto	Amstrad	MSX
12 Detector de lluvia	D0	Stick 1; D0
13 Veleta	D1–D3	Stick 1; D2,D4,D5
14 Anemómetro	D4	Stick 2; D0
15 Termómetro	D5	Stick 2; D2
16 Barómetro	D6	Stick 2; D4
17 Medidor solar	—	Stick 2; D5

es el proyecto más simple de todo el libro. No sólo le sirve para mantener un registro del número de días lluviosos o el número y duración de los chaparrones, sino que también tiene otros usos en el hogar. En los días de colada es útil para detectar el comienzo de la lluvia. Si el pitador (proyecto 5) está también conectado y situado en la cocina (o al lado del televisor), el micro puede avisar de que conviene recoger la ropa de la cuerda antes que se moje más. El proyecto también detecta cuándo el agua sube más allá de un determinado nivel. Junto con el micro, puede detectar cuándo se llena el baño o cuándo se seca un pozo.

## Funcionamiento

La corriente de base del transistor Q1 (Fig. 12.1) pasa a través de la sonda. Esta consiste en tiras de un material conductor situadas unas junto a las otras. La distancia de separación entre las tiras debe ser suficientemente pequeña para que las gotas de lluvia puedan producir un puente entre ellas. Una única gota de lluvia puede producir suficiente conducción para hacer que el transistor empiece a conducir también. El resultado de esto es que el nivel de la entrada de datos cae de alto a bajo. La figura 12.2 muestra una forma

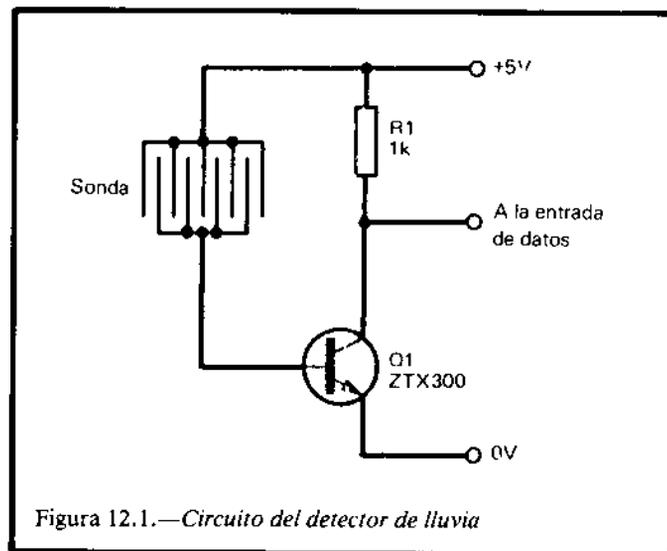


Figura 12.1.—*Circuito del detector de lluvia*

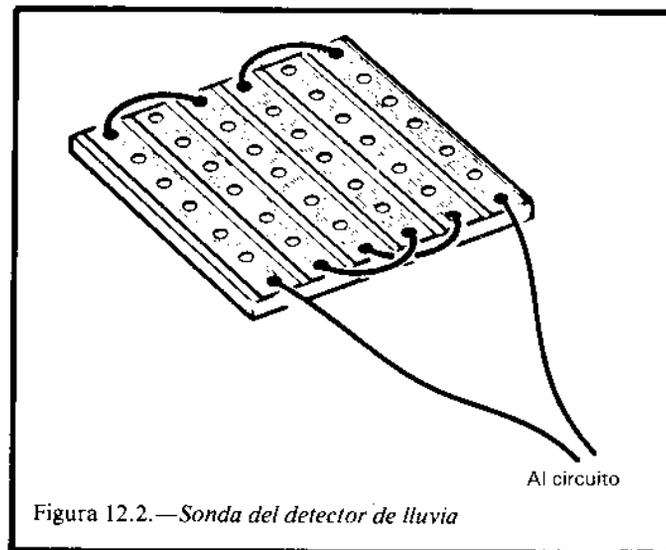


Figura 12.2.—*Sonda del detector de lluvia*

fácil de construir una sonda con un trozo de placa de circuito impreso.

## Construcción

Hacen falta dos pequeñas placas de circuito impreso: una para el circuito y otra para la sonda. De hecho, es posible construir todo el dispositivo en una única placa, tapando la zona de los componentes y dejando al aire la zona correspondiente a la sonda. Este circuito produce una salida continua, de tal manera que se pueda conectar al ordenador con un cable largo. Compruebe el circuito conectando un voltímetro a su salida y verificando que se producen 5V. Cuando deje caer una gota de agua en la sonda, la salida caerá hasta un voltaje próximo a los 0V. El mismo efecto se puede obtener tocando la sonda con los dedos humedecidos, así que este dispositivo también puede tener aplicaciones como interruptor sensible al tacto. La sonda debe situarse al aire libre, de tal manera que no quede protegida de la lluvia por las edificaciones y los árboles. Es mejor disponerla ligeramente inclinada, de tal manera que cuando deje de llover el agua resbale y se seque pronto.

## Programación

Lo único que se necesita es un programa que lea la línea de entrada en intervalos regulares de tiempo. Si llueve, JOY(0) devolverá un 1; en caso contrario, un 0. Con un MSX utilice STICK o STRIG.

## Componentes

### *Resistencia*

R1 1K

### *Semiconductor*

Q1 ZTX300 o cualquier otro transistor NPN.

### *Varios*

Placa para la sonda y el circuito.  
Conector para el puerto de *joystick*.  
Cable.

## Veleta

Conocer la dirección del viento es fundamental para la predicción meteorológica. Este proyecto lee la dirección del viento y pasa la información al micro. También es interesante como ejemplo de cómo el micro puede ser informado de una posición que se mide como un ángulo.

### Funcionamiento

La posición de la veleta es detectada por tres resistencias dependientes de la luz (LDR, *light dependent resistor*). La figura 13.1 muestra el circuito que controla una de ellas. Las LDR reciben la luz de una bombilla que pasa a través del disco codificador (figuras 13.2 y 13.3). Este disco se fija a la base de la veleta. El disco es transparente, pero tiene dibujado una serie de sectores en negro. En cualquier posición de la veleta (y, por tanto, del disco), cada una de las LDR puede estar iluminada o no. Cuando cambia la dirección del viento el disco gira. Algunas LDR se iluminan y otras se cubren. Un examen más detallado del disco (Fig. 13.2) muestra que la forma de los sectores NO representa la secuencia binaria. Leyendo en el sentido de las agujas del reloj y convirtiendo a

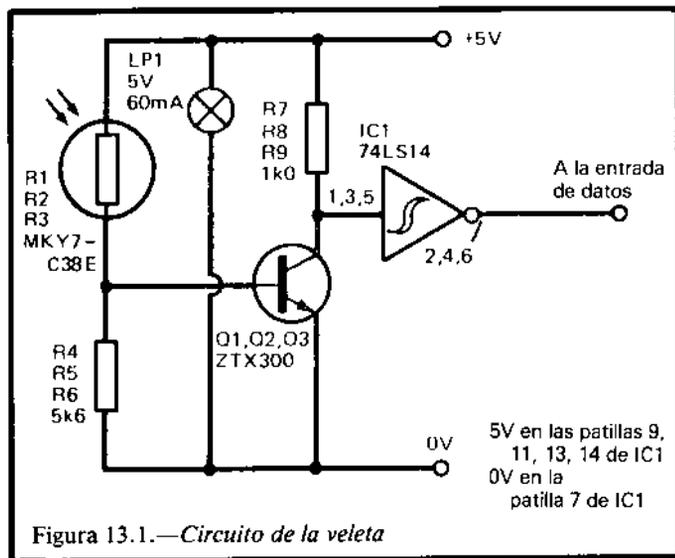


Figura 13.1.—Circuito de la veleta

decimal, los números obtenidos son: 0, 1, 3, 2, 6, 7, 5, 4. Esto es lo que se conoce como código Gray. La principal característica de esta secuencia es que cuando pasamos de un número al siguiente (o retornamos a uno anterior) sólo cambia un dígito binario. Para ver la importancia de esto veamos qué ocurre si empezamos la secuencia 0, 1, 2, etc. El primer cambio sería:

	0	0	0
cambio a	0	0	1

Este cambio no crea problemas y se encuentra en la secuencia Gray, pero el segundo sí produce complicaciones:

	0	0	1
cambio a	0	1	0

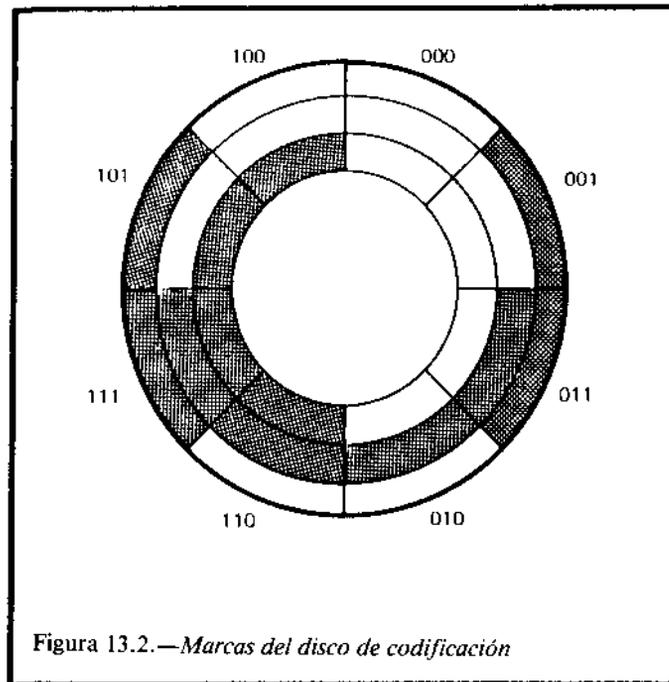


Figura 13.2.—Marcas del disco de codificación

No habría ningún problema si ambos dígitos cambiasen en el mismo instante, pero en un equipo como el nuestro es muy improbable que las LDR estén alineadas tan precisamente que ambas respondan de forma idéntica. No cambiarán ambas en el mismo instante. Si el dígito intermedio cambia antes que el de la derecha obtendremos el estado intermedio 0 1 1. Obtendríamos el mismo estado si cambiara primero el dígito derecho. En lugar de obtener 0, 1, 2, ..., cuando el disco rota obtendríamos 0, 1, 3, 2. El 3 aparecerá en un breve intervalo de tiempo, pero será suficiente para que el micro haga una lectura y obtenga un resultado falso. El código Gray evita este problema, ya que sólo cambia un dígito cada vez.

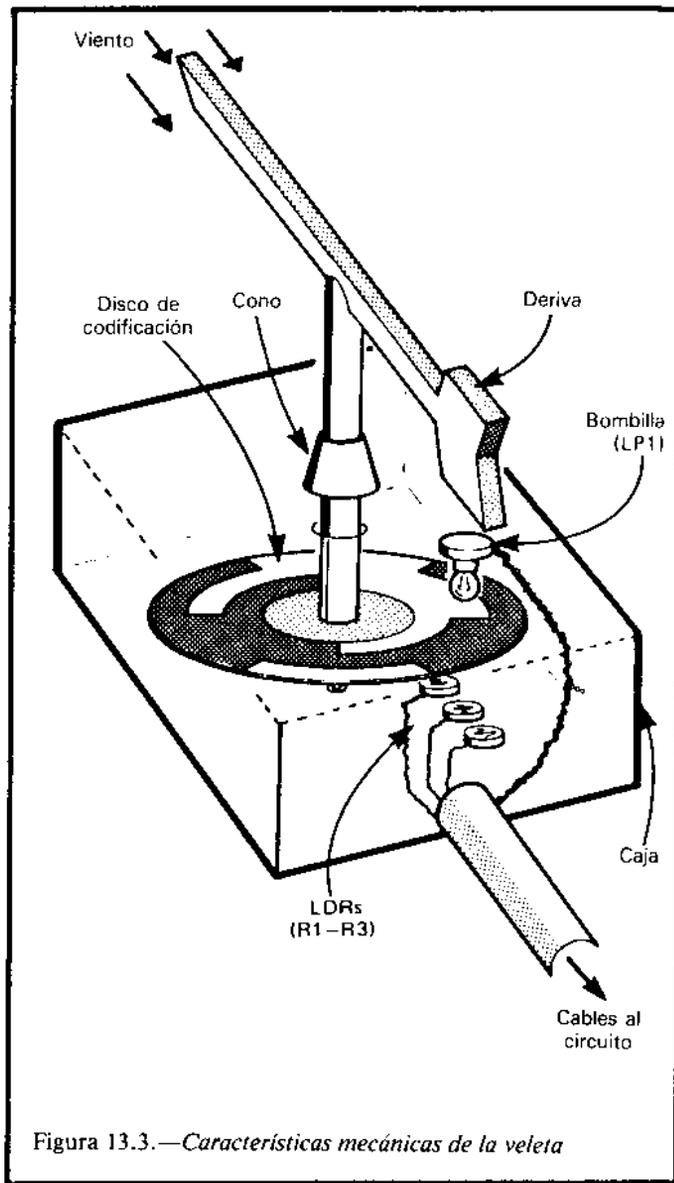


Figura 13.3.—Características mecánicas de la veleta

## Construcción de la veleta

La figura 13.3 muestra las características generales de una veleta típica. La forma exacta de construcción dependerá de los materiales y herramientas disponibles, así como de la habilidad manual del constructor. A veces es posible adaptar veletas ya construidas, que se pueden obtener fácilmente en tiendas de material educativo. Suelen estar hechas de plástico y, aunque son menos resistentes que las destinadas a usos serios, son más fáciles de adaptar. El disco se hace de plástico transparente. Se puede utilizar una lámina de acetato que se puede pintar con un rotulador negro. También pueden hacerse los sectores recortando cinta aislante y pegándola en el disco. Para que las lecturas indiquen a los principales puntos cardinales (NE, E, SE, etc.), las LDR deben situarse en el centro de cada sector cuando la veleta está alineada con los puntos cardinales.

Para mayor claridad, la figura 13.3 muestra el disco algunos centímetros por encima de las LDR, pero los resultados son mejores si el espacio es menor y el disco prácticamente las toca. La bombilla debe montarse unos centímetros por encima del disco, de tal manera que la luz se extienda uniformemente por las tres LDR. Se deben tender cables que unan las LDR con la placa principal del circuito, que debe de estar situada en una pequeña caja cerca del micro.

La figura 13.3 muestra una forma de evitar que el agua de lluvia se cuele dentro del mecanismo. El cono la deflecta hacia la superficie de una caja opaca. No es esencial que el agujero por el que pasa el eje de la veleta sea absolutamente estanco a la luz, ya que una anilla puede evitar que la luz se filtre, alcanzando a la LDR.

## Construcción del circuito

El dispositivo requiere cuatro líneas de conexión con el micro: la tierra y tres líneas de datos. La veleta requiere cinco cables: la alimentación de 5V, tres cables para las LDR y una masa. Se recomienda utilizar LDR pequeñas. Pueden utilizarse tipos mayores, si no están disponibles las pequeñas, pero esto requerirá que el disco sea también más grande. Si las LDR no son del tipo especificado y tienen una resistencia mayor o menor, puede ser necesario sustituir las resistencias R4 a R6 por otras de un valor distinto.

Es mejor construir primero un único circuito y probarlo. Así es fácil comprobar que la resistencia utilizada es la correcta antes de proceder a la construcción de lo demás. Cuando todo esté acabado gire el disco despacio para medir la salida de cada puerta con un voltímetro. La figura 13.2 muestra los resultados esperados. 0 representa nivel bajo (menos de 0.8V) y 1 representa nivel alto (más de 2V). Suponiendo que todo es correcto, pruebe todos los cables que van al micro para asegurarse de que no hay cortocircuitos.

## Programación

Todo lo que se requiere es leer la entrada de datos. En el Amstrad se utilizará una sentencia del tipo:

```
10 X=JOY(0) AND 14
```

Esta instrucción supone que la veleta está conectada a las líneas D1 a D3, como se recomienda en la tabla 12.1. La sentencia devuelve ocho posibles valores para X, que son los números pares comprendidos entre 0 y 14, ambos inclusive. Los valores correspondientes a las posiciones de la figura 13.2 son:

Posición	Valor
000	0
001	2
011	6
010	4
110	12
111	14
101	10
100	8

La situación es ligeramente más compleja en un MSX. La sentencia utilizada debe ser:

```
10 X=STICK(1) AND 7: Y=STRIG(1):  
Z=STRIG(3)
```

Los valores obtenidos son:

Posición	X	Y	Z
000	7	-1	-1
001	0	-1	-1
011	0	0	-1
010	7	0	-1
110	7	0	0
111	0	0	0
101	0	-1	0
100	7	-1	0

Estos valores pueden utilizarse con una línea del estilo:

```
50 IF X=7 AND Y=0 AND Z=-1 THEN  
DIRECCION=4
```

Se puede utilizar la dirección del viento para predecir el tiempo. La predicción dependerá mucho de la localidad. También puede depender de los datos me-

teorológicos obtenidos por otros proyectos. Además de para leer la dirección del viento, también se puede utilizar la veleta para detectar los cambios del viento que se producen en un período largo de tiempo. Si el viento cambia de dirección siguiendo un giro similar al de las agujas del reloj, es posible que se acerquen bajas presiones y, por tanto, nubes y lluvia. Otro aspecto interesante es la estabilidad de la dirección del viento. Si al realizar lecturas de la veleta descubrimos que la dirección del viento cambia a cada minuto, podremos tomarlo como pista para una predicción del tiempo. Existen en el mercado numerosos libros de meteorología que le ayudarán a realizar sus pronósticos. También es posible programar el micro para que lo realice.

### *Varios*

LP1      lámpara de filamento 5V 60 mA.

Casquillo para LP1.

Zócalo de 14 patillas.

Conector para el puerto *joystick*.

Placa de circuito impreso.

Espadines de 1 mm. (14 unidades).

## Componentes

*Resistencias* (carbón, 0.25V, tolerancia 5 excepto R1-R3):

R1-R3    MKY7C38E o LDR similar (preferiblemente pequeña; 3 unidades).

R4-R6    5K6 (véase el texto; 3 unidades).

R7-R9    1K (3 unidades).

*Semiconductores*

Q1-13    ZTX300 o transistor NPN similar (3 unidades).

*Circuito integrado*

IC1      74LS14 disparador Schmitt h́exuple.

# Anemómetro

El anemómetro mide la velocidad del viento. Se compone de un conjunto de 3 ó 4 semiesferas situadas en la parte superior de un eje (Fig. 14.1), de tal manera que giren cuando las da el viento. Se supone que las semiesferas viajan a una velocidad similar a la del viento. Así que, conociendo el radio del montaje y la velocidad de rotación, se puede estimar la velocidad del viento. De hecho, este proyecto es en realidad un tacómetro, esto es, un medidor de velocidad de rotación. El circuito puede adaptarse para medir la velocidad de giro de cualquier otro mecanismo.

## Funcionamiento

La figura 14.1 muestra una rueda en la base del eje del anemómetro. Esta gira a la misma velocidad que las semiesferas. La rueda está hecha de un material no magnético, tal como plástico o madera. Sobre ella se encuentra un pequeño imán. El imán está dispuesto de tal manera que sobresalga uno de sus polos. Cuando la rueda gira, el imán pasa por delante de un integrado. Este integrado es un sensor de efecto Hall. El efecto Hall se produce cuando se sitúa un semiconductor en un campo magnético. Aquellos que

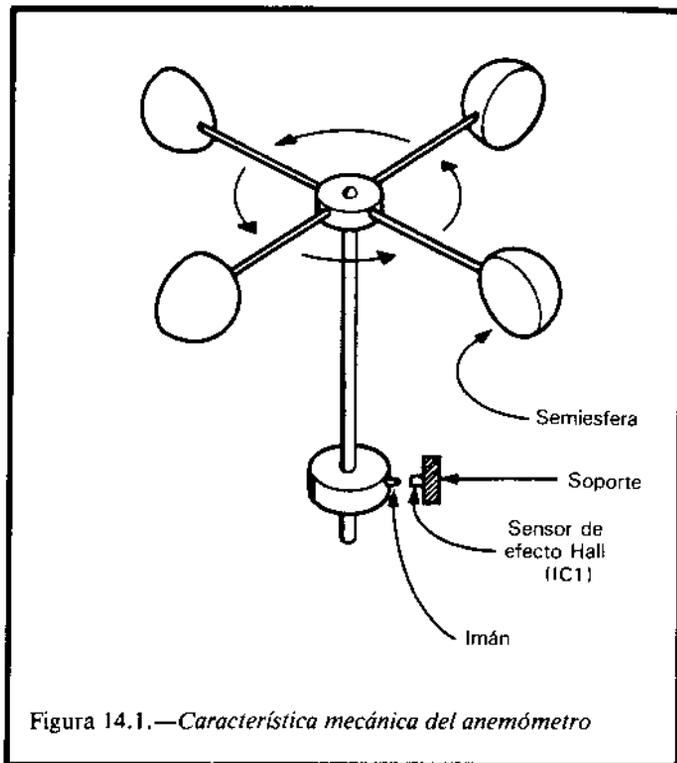


Figura 14.1.—Característica mecánica del anemómetro

recuerden haber estudiado en las clases de Física las reglas de la mano izquierda, sabrán que hay una interacción entre la corriente y el campo magnético. En la figura 14.3 se hace pasar una corriente a través de una oblea de silicio semiconductor. Los electrones en movimiento se deflexan hacia un lado en presencia de un campo magnético. Esta es detectada y amplificada por los circuitos internos del integrado. Las salidas de IC1 están aproximadamente a 2V, habitualmente. Cuan-

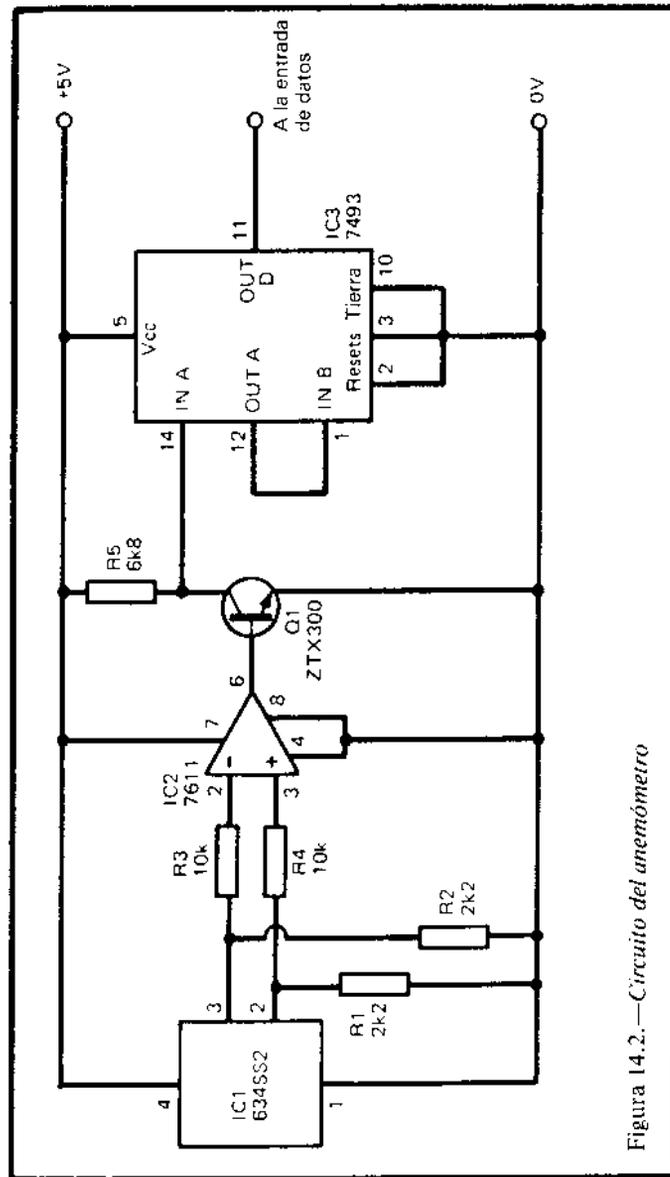


Figura 14.2.—Circuito del anemómetro

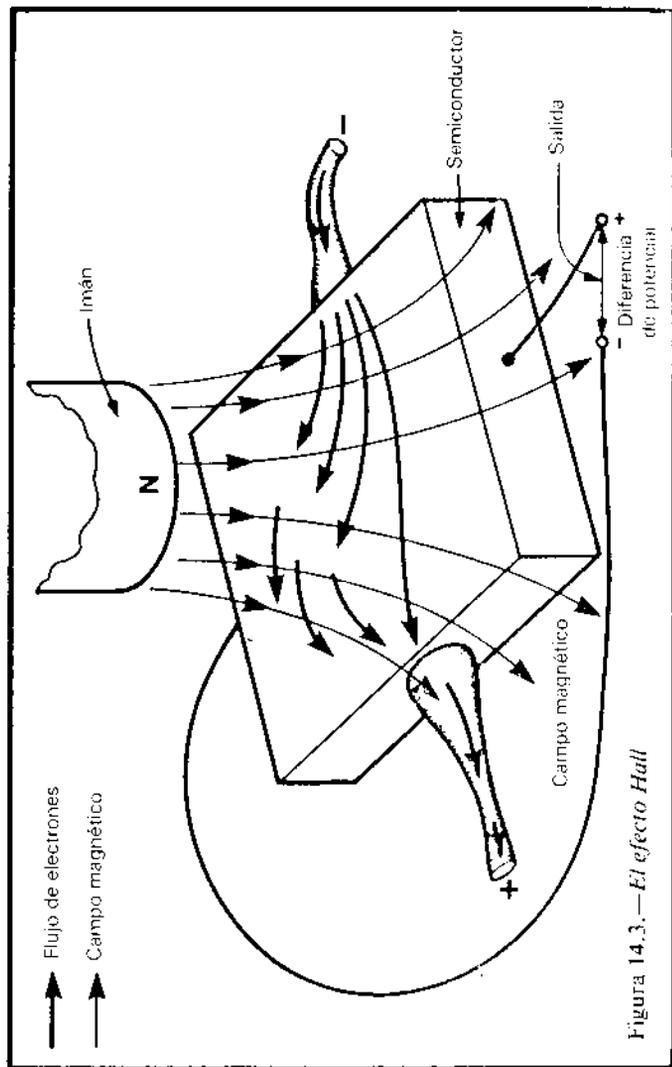


Figura 14.3.—El efecto Hall

do se acerca un imán, el voltaje de una de las salidas aumenta ligeramente y el de la otra, cae. El amplificador operacional IC2 compara estos voltajes. Cuando son iguales (o casi) su salida es baja. Cuando los voltajes son distintos (es decir, cuando el imán está cerca), la salida de IC2 prácticamente es de 5V. Esto hace que Q1 conduzca. El voltaje en la entrada del contador (IC3) cae haciendo que este actúe. Por tanto, el contador se incrementa cada vez que el imán pasa cerca de IC1. IC3 está cableado como un divisor por 16, por lo que su salida D cambia a un dieciseisavo de la velocidad de giro del anemómetro. Estos cambios son medidos por el micro que, a partir de ellos, calcula la velocidad del viento.

## Construcción

Al contrario que la mayoría de los interfaces de este libro, el grueso del circuito está situado mejor cerca del sensor. La salida de IC3 varía de espacio, lo que la hace adecuada para su transmisión por cables relativamente largos hasta la entrada del micro. El circuito requiere dos cables para conectarse con el micro: la tierra y un cable para la entrada de datos. El circuito debe situarse en una caja a prueba de agua. Puede ser conveniente montar IC1 en el exterior de la caja. Los terminales del integrado deben cubrirse con cera o silicona para eliminar la posibilidad de cortocircuitos. El anemómetro es fácil de construir. Las semiesferas pueden hacerse con medias pelotas de ping-pong y los radios con cable rígido o cualquier otra varilla. Es fundamental que el montaje pueda girar libremente. El diámetro debe ser de aproximadamente 30 cm, o quizá un poco mayor. La rueda sobre la que está situada el imán puede cortarse en madera, plástico o corcho. Un imán adecuado se puede comprar junto con el sensor de efecto Hall.

Tenga cuidado al montar el sensor y el imán, de tal manera que no se toquen cuando el anemómetro gira, particularmente con vientos altos. Por otra parte, el imán debe pasar a menos de 1 mm del centro del integrado para generar una diferencia de potencial suficiente. Construya el circuito y, a continuación, aliméntelo con 5V para probarlo. Conecte un voltímetro a una de las salidas de IC1. Gire el anemómetro despacio y compruebe si el voltaje crece o decrece cuando el imán se acerca al integrado. Coloque el voltímetro en la salida de IC2. La salida debe ser de 0V, pero debe pasar 5V siempre que el imán pase cerca de IC1. Si esto no ocurre, la polaridad del imán está cambiada; desmóntelo de la rueda y vuelva a colocarlo dándole la vuelta. Conecte el voltímetro a la salida D de IC3. Debe pasar de alto a bajo, y de nuevo a alto una vez cada 16 rotaciones del anemómetro. La salida D cambia a baja velocidad, de tal manera que el cable entre el interfaz y el decodificador puede ser de varios metros de longitud, lo cual permite montar el dispositivo lo más alto posible, de tal manera que quede expuesto al viento.

## Programación

Resulta necesario un programa que mida la longitud de los pulsos que llegan desde D. El programa en lenguaje máquina de la página 123 puede valer. De todas formas, los pulsos son lentos y un programa en BASIC puede resultar suficiente. A continuación, se da un programa para el Amstrand que funciona, suponiendo que el anemómetro está conectado a la entrada de datos D4:

```
10 IF JOY(0) AND 16=0 THEN 10
20 IF JOY(0) AND 16=16 THEN 20
```

```
30 COMIENZO=TIME
40 IF JOY(0) AND 16=0 THEN 40
50 IF JOY(0) AND 16=16 THEN 50
60 FINAL=TIME
70 R=(FINAL-COMIENZO)/18.75
```

Este programa espera a que la entrada pase a nivel bajo (JOY(0) es distinto de 0) y, a continuación, de nuevo a alto, antes de empezar a contar. Así se asegura que si la entrada se encuentra a nivel bajo cuando el programa empieza a ejecutarse, la temporización no comenzará hasta el siguiente pulso bajo. La temporización continúa hasta que la entrada pasa de nivel bajo a alto de nuevo. La expresión COMIENZO—FINAL da la duración de un pulso que es dieciséis veces la duración de una revolución del anemómetro. Dividiéndola entre 18.5 obtenemos la velocidad de rotación en revoluciones por segundo. En el MSX, el anemómetro debe conectarse a la línea D0 del puerto 2. Las sentencias correspondientes tienen la forma:

```
10 IF STICK(2) AND 1=1
```

Las líneas 20, 40 y 60 deben modificarse similarmente. La línea de cálculos se convierte en:

```
70 R=(FINAL-COMIENZO)/3.125
```

Suponiendo que las semiesferas se mueven a la misma velocidad que el viento, se puede calcular la velocidad de este mediante la forma:

$$S=2*\pi r*R*3600/100000$$

Siendo S la velocidad del tiempo en kilómetros por hora, R la velocidad de rotación en revoluciones por segundo, y r el radio del anemómetro en centímetros.

$\pi$  tiene el valor de 3.1416, así que la ecuación simplificada queda como:

$$S=0.226*r*R$$

Cualquiera de estas ecuaciones puede usarse en un programa para calcular la velocidad del viento dada la velocidad de rotación. Por ejemplo, si el interfaz produce un conteo de 1 por segundo, quiere decir que la velocidad de rotación (R) es de 16 revoluciones por segundo. Si el radio del anemómetro es de 15 cm, la velocidad del viento será:

$$S=0.226*15*16=54 \text{ Km/h}$$

El circuito del anemómetro puede soportar velocidades de hasta el doble de las del ejemplo, por lo que puede registrar vientos de 100 Km/h o más.

## Componentes

*Resistencias* (carbón, 0.25W, tolerancia 5%):

R1, R2    2K2 (2 unidades).  
R3, R4    10K (2 unidades).  
R5        6K8

*Semiconductor*

Q1        ZTX300 o transistor NPN similar.

*Circuitos integrados*

IC1       634SS2 sensor de efecto Hall (junto con su imán).  
IC2       7611 amplificador operacional CMOS.  
IC3       7493 contador/divisor binario de 4 bits.

*Varios*

Placa de circuito impreso.  
Zócalos de 8 patillas (2 unidades).  
Zócalo de 14 patillas.  
Espadines de 1mm. (7 unidades).  
Conector para el puerto de *joystick* .  
Materiales para la construcción del anemómetro.  
Cable.

# Termómetro

Aunque este proyecto forma parte de la estación meteorológica, puede tener muchas aplicaciones independientes. Puede utilizarse para medir la temperatura de habitaciones o la de soluciones fotográficas. Puede utilizarse para detectar excesos de temperatura como parte de un sistema doméstico de alarmas contra incendios.

## Funcionamiento

El circuito funciona de la misma forma que el controlador de juegos (proyecto 11). En lugar de un potenciómetro, en este caso utilizamos otro tipo de resistencia variable (Fig. 15.1): un termistor. Un termistor es una resistencia que varía con la temperatura. El termistor utilizado en este circuito tiene un coeficiente negativo de temperatura, lo cual significa que, en el rango de funcionamiento, la resistencia decrece cuando la temperatura crece. Este tipo de resistencias son llamadas también NTC. Un incremento de temperatura hace que la resistencia del termistor baje, bajando a su vez el potencial del punto A y, por tanto, el voltaje analógico de entrada del circuito IC2. Como se explicó en el proyecto 11, la forma de onda

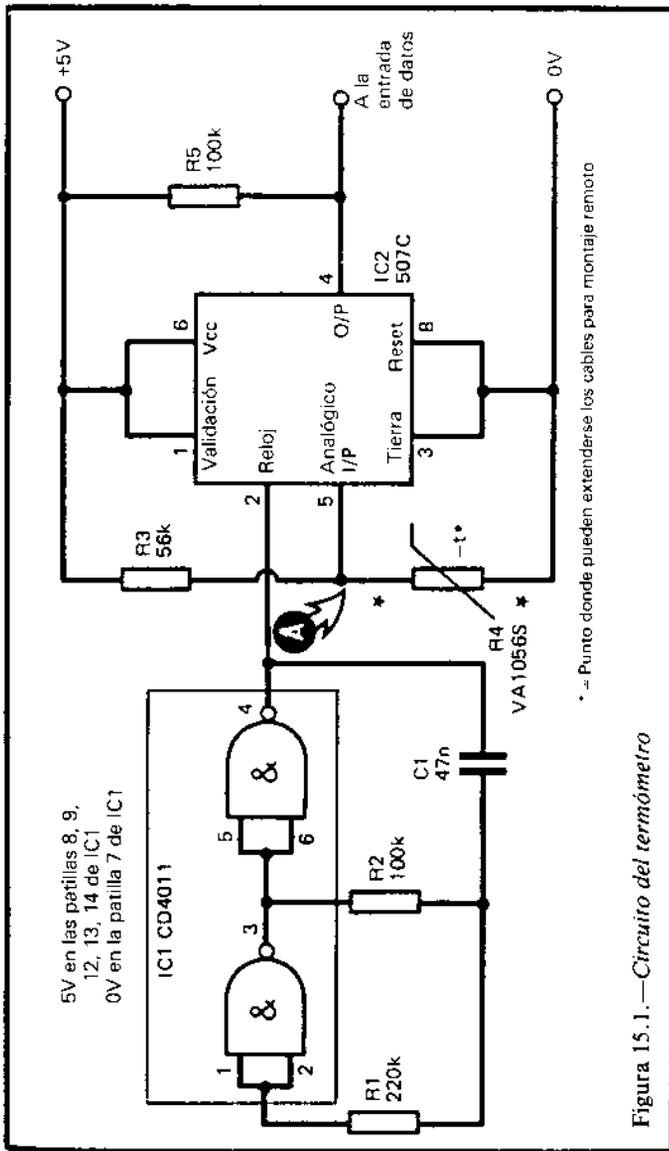


Figura 15.1.—Circuito del termómetro

de salida del integrado cambia (Fig. 11.2), y esto puede ser medido por el micro.

## Construcción

Normalmente, el termistor se sitúa en el exterior. Para utilizarlo para registrar temperaturas ambiente, el mejor sitio es dentro de una caja de observación meteorológica. Esta evita que el sol incida directamente sobre el termistor, pero permite que el aire circule libremente a su alrededor. Si no tiene una caja de este estilo, móntelo en un lugar protegido de la luz solar directa y donde la circulación del aire sea buena. Debe estar separado varios centímetros de cualquier superficie, tal como una pared o una valla, ya que, si no, en realidad se medirá la temperatura de ésta en lugar de la del aire. Si está situado al aire libre, debe protegerse de la lluvia para evitar que se produzcan cortocircuitos que falsearían las medidas. Un método para impermeabilizarlo es sumergirlo en cera fundida para recubrirlo. Otro método es utilizar un pegamento de resina de EPOXY (Araldite, o Imedio banda verde) para el recubrimiento.

El resto del circuito se monta en una caja pequeña y se sitúa cerca del micro. Se necesitan dos cables para realizar la conexión: la tierra y la línea de datos.

Cuando el circuito está acabado, puede probarse de igual forma que se hizo con el controlador de juegos. La prueba se puede hacer sumergiendo el termistor en agua caliente o en cubitos de hielo. En tales extremos, la salida de IC2 nunca debe ser continuamente alta o baja. Si esto ocurre, cambie R3 por una resistencia distinta, o sitúe una resistencia adicional entre el termistor R4 y la línea de 0V. La solución a tomar depende de las características del termistor que se esté utilizando y del rango de temperaturas en el que se quiera usar.

## Calibración y programación

La forma más simple de programación es leer la salida del circuito muchas veces y contar en cuántas de ellas se obtiene el nivel alto. Este es el método de muestreo. Se puede utilizar un programa muy similar al del proyecto 11. La única diferencia consiste en que, en este caso, podremos tomar más muestras y lograr mejores resultados. La línea 20 puede modificarse para tomar, por ejemplo, quinientas muestras en lugar de sólo cien. El tiempo adicional requerido no resulta importante en esta aplicación. Las líneas 50 a 70 no son aplicables en este caso, y se pueden sustituir por otras que dibujen en la pantalla un termómetro o, simplemente, una línea del tipo:

50 PRINT X

Habiendo obtenido el valor de X, que varía proporcionalmente con la temperatura, el siguiente paso es calibrar el termómetro. Sitúe el termistor en un vaso de agua que contenga cubitos de hielo. Agítelo de vez en cuando, durante al menos cinco minutos, para que el termistor tome la temperatura del agua, que será muy cercana a 0 °C.

Ejecute el programa varias veces y calcule una media de los valores obtenidos para X. El resultado debe estar comprendido entre 20 y 50. Si cuando realice las medidas ve que éstas decrecen progresivamente, espere un poco, ya que el termistor no se ha enfriado todavía por completo. Si los cubitos se funden, añada más. A continuación, repita el procedimiento con un vaso de agua templada, aproximadamente 30 °C. Elija una temperatura un poco mayor a la cota superior del rango en el que se pretendan tomar las medidas. Será necesario utilizar un termómetro para saber la temperatura exacta del agua. Las lecturas, en este caso deben ser superiores a 300.

Si la diferencia entre unas lecturas y otras es muy pequeña, no será posible obtener resultados precisos. Si esto ocurre, debe remplazarse R3 o situar una resistencia adicional, como antes se mencionó.

Supongamos que se obtiene una lectura X1 a 0 °C y X2 a T °C. La fórmula para convertir la lectura X a temperaturas es:

$$t = (X - X1) * \frac{T}{X2 - X1}$$

Por ejemplo, si la cuenta es 40 a 0 °C y 420 a 30 °C, entonces una cuenta de 50 equivale a una temperatura de:

$$t = (150 - 40) * \frac{30}{420 - 40} = 90 * 30 / 380 = 7.1^{\circ}\text{C}$$

Además de programarse para tomar lecturas en momentos dados, el micro puede utilizarse para tomar lecturas en intervalos fijos y registrarlas en memoria. Pueden entonces utilizarse para hacer tablas diarias. Si las lecturas se hacen con la suficiente frecuencia, es fácil programar el micro para dar las máximas y las mínimas diarias.

## Componentes

*Resistencias* (carbón, 25W, tolerancia 5%):

R1        220K.  
R2, R5    100K (2 unidades).  
R3        56K.

R4 VA1056S o cualquier termistor con coeficiente de temperatura negativo, NTC, con una resistencia aproximada de 47K a 25 °C.

# 16

## Barómetro

### Condensador

C1 47n poliéster

### Circuitos integrados

IC1 CD4011 cuádruple puerta NAND de dos entradas, CMOS.  
IC2 507C conversor de voltaje a tiempo.

### Varios

Caja para el circuito.  
Placa de circuito impreso.  
Espadines de 1mm. (5 unidades).  
Zócalo de 8 patillas.  
Zócalo de 14 patillas.  
Conector del puerto del *joystick*.  
Cable.

Un barómetro mide la presión atmosférica. Resulta esencial para la predicción del tiempo. La aproximación de borrascas y anticiclones está precedida por una caída o subida de presión. Esto nos permite presuponer el tiempo que va a hacer. La velocidad del cambio de presión nos indica cuándo debemos esperar el cambio de tiempo. Además, los cambios rápidos de presión indican fuertes vientos, mientras que los cambios lentos están asociados con el tiempo tranquilo.

Los equipos destinados a medir la presión atmosférica electrónicamente suelen ser muy caros. Este proyecto es barato y, aunque no da una lectura muy precisa, indica los cambios, que es lo realmente importante. El conocer exactamente la presión no resulta de mucha importancia, por lo que este aparato resulta útil para la predicción del tiempo.

## Funcionamiento

El barómetro está basado en una idea muy simple que a menudo aparece en los libros de divulgación científica (Fig. 16.1). El tubo invertido contiene aire. A medida que la presión atmosférica crece, el aire

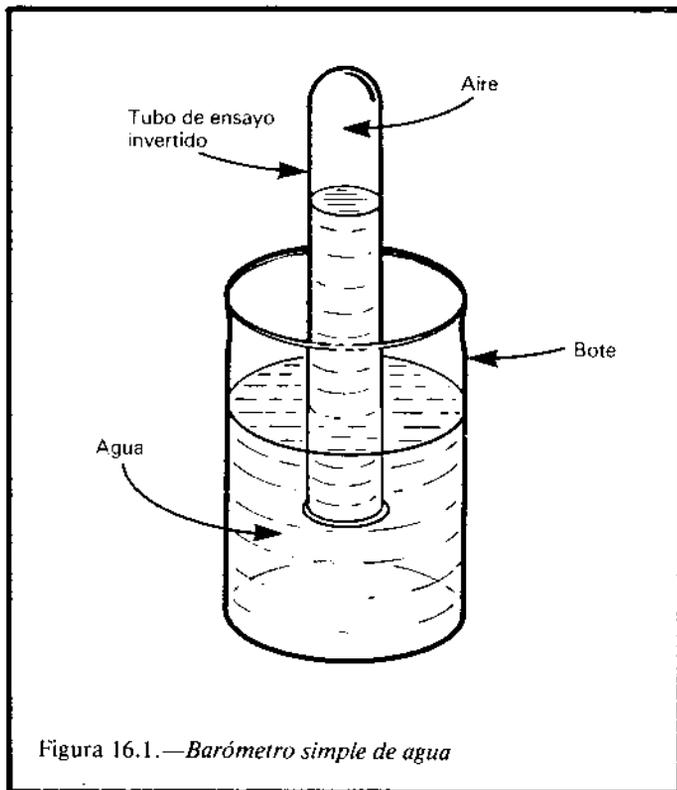


Figura 16.1.—Barómetro simple de agua

que hay dentro del tubo se comprime y la columna de agua crece. Cuando la presión cae, el aire se expande y el agua baja. El valor de la presión se obtiene midiendo el nivel del agua. Desgraciadamente, este aparato tiene un pequeño fallo: el aire se expande cuando se calienta. Si la temperatura crece, el nivel del agua bajará, aunque la presión atmosférica permanezca constante. Es posible saber cuánto variará el volumen del aire en función de la temperatura, pero resulta demasiado complicado para aplicarlo al baró-

metro simple de la figura 16.1. La solución consiste en conectar el barómetro a un ordenador y que éste realice el trabajo. Para compensar los cambios de temperatura, el micro debe conocerla en la región cercana al barómetro. Si el termómetro (proyecto 15) opera al mismo tiempo, nos será fácil conocer la temperatura en todo momento. Hay una ecuación que relaciona el volumen, la temperatura y la presión:

$$\frac{\text{Presión} * \text{Volumen}}{\text{Temperatura}} = \text{Constante}$$

La presión y el volumen pueden medirse en cualquier unidad (siempre las mismas), mientras que la temperatura debe medirse en grados Kelvin. Para pasar a grados Kelvin cualquier temperatura en grados centígrados basta con sumarle 273.

Si medimos la presión, el volumen y la temperatura del aire dentro del barómetro, en dos ocasiones distintas, se puede afirmar que:

$$\frac{\text{Presión1} * \text{Volumen1}}{\text{Temperatura1}} = \frac{\text{Presión2} * \text{Volumen2}}{\text{Temperatura2}}$$

O despejando:

$$\text{Presión2} = \text{Presión1} * \frac{\text{Volumen1}}{\text{Volumen2}} * \frac{\text{Temperatura2}}{\text{Temperatura1}}$$

Utilizaremos un barómetro ordinario para medir presión1 cuando tomemos por primera vez una medida con nuestro barómetro casero. Al mismo tiempo, mediremos temperatura1 y volumen1. Un tiempo después mediremos volumen2 y temperatura2 y calcularemos presión2. El proyecto 15 mide la tempera-

tura, mientras que este proyecto mide el volumen. A continuación, el micro calcula la presión.

En este proyecto, el aire se encierra en un tubo de cristal o de plástico (Fig. 16.2). La parte superior del tubo tiene una válvula y la parte inferior se sumerge en un vaso que contenga agua. El agua estará teñida con tinta china. Los mayores cambios de presión y temperatura hacen que el agua se mueva en un rango de aproximadamente 2 cm.

En la región en que el agua se va a mover, el tubo se cubre con cinta aislante negra, dejando dos ranuras libres en lados opuestos del tubo. La luz de una pequeña bombilla pasa a través de una de las ranuras, cruza el tubo y la otra ranura, e incide en una resistencia dependiente de la luz (LDR). Dado que el color del agua es negro, absorberá una cantidad de luz dependiendo de su nivel. Por tanto, los cambios en el nivel del agua producirán cambios en la resistencia. En el prototipo construido de este proyecto, la LDR tenía 940 ohmios cuando el agua se encontraba en su nivel más bajo, y se incrementaba hasta 1700 ohmios cuando el agua interrumpía totalmente el paso de la luz. Por las mismas razones dadas en la página 135, los cambios de resistencia en la LDR afectan al voltaje en el punto A. Este voltaje variable se alimenta a IC2, que es un conversor de voltaje a tiempo. El funcionamiento se explica en la página 117. En resumen, se puede decir que si la presión crece, el volumen decrece, el voltaje en A crece y la forma de onda generada por IC2 ocupa proporcionalmente más tiempo en estado alto. Si el micro se programa para medir este tiempo, el número obtenido será mayor.

Es importante tener en cuenta que la respuesta del dispositivo no es lineal. Esto quiere decir que a incrementos iguales de presión no se producen incrementos proporcionales del valor calculado por el micro. De todas formas, el número obtenido puede utilizarse para detectar si la presión crece o decrece en un

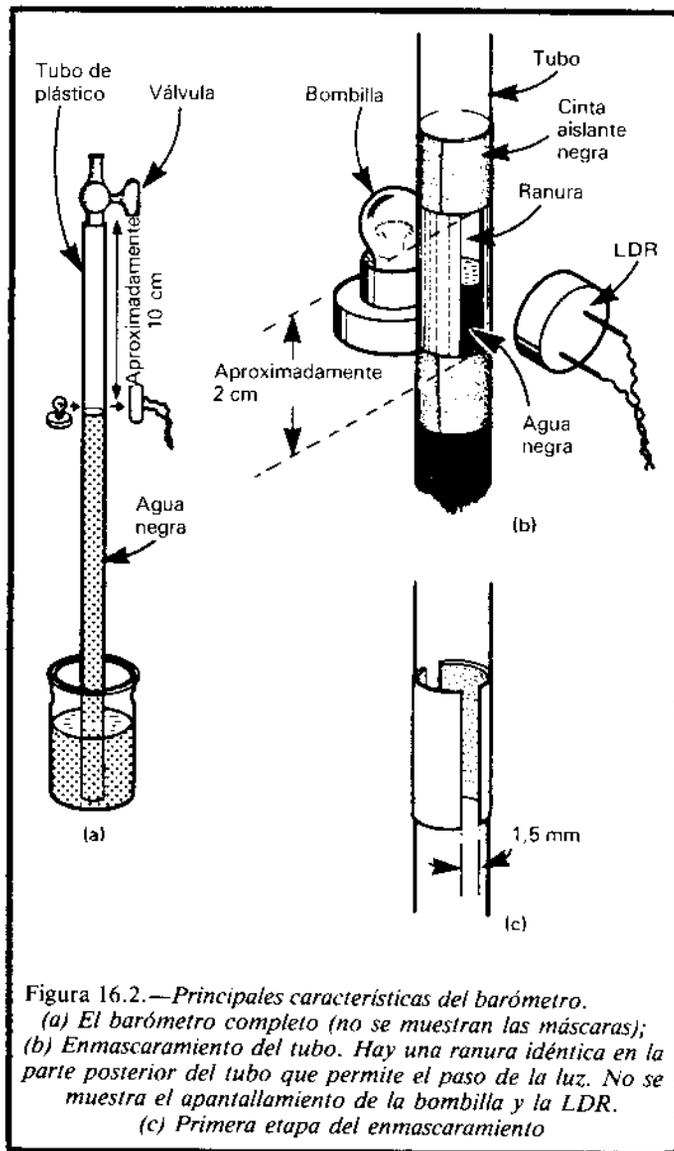


Figura 16.2.—Principales características del barómetro.  
 (a) El barómetro completo (no se muestran las máscaras);  
 (b) Enmascaramiento del tubo. Hay una ranura idéntica en la parte posterior del tubo que permite el paso de la luz. No se muestra el apantallamiento de la bombilla y la LDR.  
 (c) Primera etapa del enmascaramiento



El circuito debe situarse razonablemente cerca del micro. En la práctica, puede pensarse en situar todo el aparato en el interior, ya que la presión atmosférica es prácticamente la misma que al aire libre. Si se sitúa en una habitación que tenga una temperatura constante, no hay necesidad de aplicar correcciones a los cálculos por efectos de la temperatura, y la programación se simplifica.

Cuando la construcción esté acabada compruebe que no hay cortocircuitos entre las líneas. A continuación, enchufe el barómetro al micro y conecte la alimentación. Dado que el dispositivo utiliza el mismo conversor que el proyecto 11, puede utilizarse un programa similar para probarlo en un Amstrand:

```

10 X=0
20 FOR J=1 TO 1000
30 IF JOY(0)>0 THEN X=X+1
40 NEXT J
50 PRINT X;" ";
60 GOTO 10

```

Ejecute el programa con el tubo vacío. El resultado obtenido varía mucho según el tipo de LDR y otras características de construcción, pero debe ser razonablemente bajo, entre 200 y 500. Aunque la lectura no varíe más que en 2 ó 3 unidades, deberá tomarse una media sobre aproximadamente 20 lecturas para obtener el resultado mínimo de la escala.

A continuación, mezcle tres partes de agua con una de tinta china negra. Añada una gota de detergente. Sitúe el agua negra en el bote. Abra la válvula del tubo y absorba lentamente. Tenga cuidado, porque puede llegar a tragar el agua sin darse cuenta. Consiga que el agua alcance un nivel superior a la zona de medida, de tal manera que la ranura quede totalmente bloqueada. Ejecute el programa de nuevo. Se obtendrán una serie de valores superiores a los obtenidos

con el tubo vacío en 50 ó 60 unidades. Si el incremento es menor, reemplace R3 por una resistencia de menor valor. Cuando haya alcanzado valores satisfactorios, realice 10 medidas y obtenga la media para lograr el valor definitivo del máximo de la escala.

Abra la válvula ligeramente y permita que el agua baje hasta alcanzar aproximadamente la mitad de la ranura. Deje el tubo en reposo durante aproximadamente un minuto para que caiga el agua retenida en las paredes interiores.

Si el barómetro se va a mantener en una temperatura constante, no hacen falta más preparativos para empezar con la programación. Ejecute el programa una docena de veces y tome la media para calcular la lectura inicial. A la vez, mida la temperatura y la presión (utilizando otro barómetro). De esta forma se obtienen los datos necesarios para la programación final (Fig. 16.4):

MAX	Máximo de escala	} Valores obtenidos del programa de prueba.
MIN	Mínimo de escala	
INIC	Valor inicial	
P	Presión l.	
T	Temperatura en grados Kelvin (centígrados + 273).	
LT	Longitud del tubo.	
LR	Longitud de la ranura.	

## Programación

Lo primero que hay que calcular es el valor que utilizaremos en el programa para representar la posición inicial del nivel del agua, medido desde la parte superior del tubo. Esto es lo que llamaremos volumen1. Es el volumen con el que se compararán las medidas posteriores. Utilizando las abreviaturas listadas al final de la sección anterior, la fórmula queda:

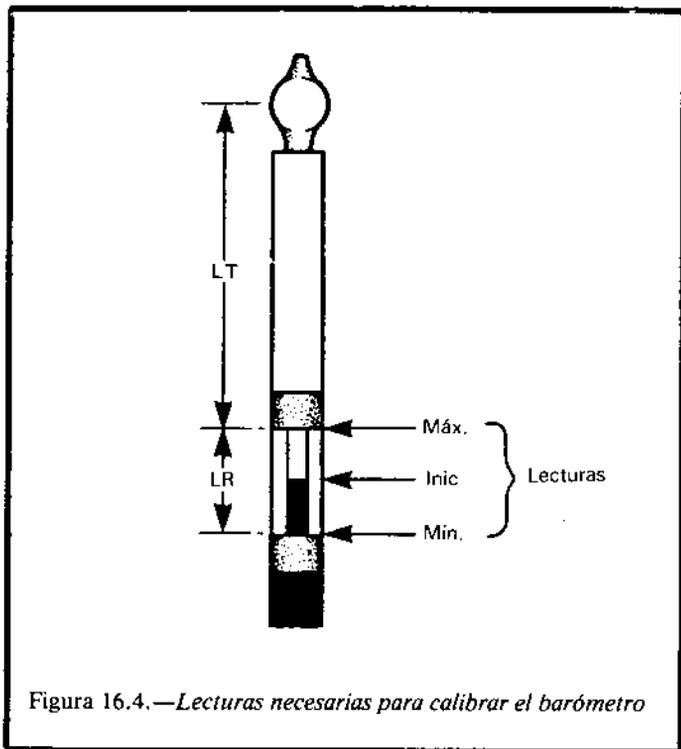


Figura 16.4.—Lecturas necesarias para calibrar el barómetro

$$\text{Volumen1} = \text{LT} + \frac{(\text{MAX}-\text{INIC}) * \text{LR}}{\text{MAX}-\text{MIN}}$$

Por ejemplo, si el tubo tiene una longitud de 100 mm y la ranura 20 mm, MAX es 270, MIN es 200 e INIC es 250, calcularemos:

$$\text{Volumen1} = 100 + \frac{(270-250) * 20}{270-200} = 106$$

Este valor se utilizará en todos los cálculos posteriores hasta que, quizá, se dé cuenta que el agua se ha evaporado y desee reemplazarla, en cuyo caso se empezará de nuevo, con nuevas medidas y un nuevo valor para volumen1.

La misma fórmula se utiliza para calcular el volumen en otras situaciones, con la única diferencia que se utiliza el valor real de la cuenta obtenida, en lugar de INIC. Por ejemplo, si la lectura (X) es 230, el ordenador calcula el nuevo volumen como 111, utilizando la línea:

$$60 \text{ NUEVO} = 100 + (270 - X) * 20 / 70$$

Ahora tenemos un nuevo volumen que se utiliza para calcular una nueva presión. Realmente, si no se está interesado en hacer correcciones por causa de la temperatura, no resulta necesario seguir adelante. Puede utilizar el valor del nuevo volumen directamente. Si la presión está creciendo, este valor decrecerá y, por el contrario, si la presión cae, este valor crece. Se puede descubrir por experiencia qué velocidad de cambio es necesaria para presuponer un cambio de tiempo.

Si se desea convertir el nuevo volumen en una presión, la línea de programa necesaria es:

$$70 \text{ PRESION} = 106 * P / \text{NUEVO}$$

Donde P es la presión inicial. La figura 106 hace referencia al volumen1 como el calculado anteriormente. Con la línea anterior puede escribirse el valor de la presión y sus unidades serán las mismas que las que utiliza el barómetro comercial que se usó para la calibración.

Si se quiere tener en cuenta la temperatura, es necesario programar el micro para que realice la medida, utilizando el proyecto 15. Supongamos que se ha

realizado y que la variable TEMP contiene el valor en grados centígrados. El valor de la presión sin convertir está contenido en la variable PRESION, calculada como se describió antes. La línea para la conversión es:

$$80 \text{ PRESION}=\text{PRESION}*\text{T}/(\text{TEMP}+273)$$

Siendo T la temperatura inicial. Si se desea una precisión extra, debe tenerse en cuenta que el vapor de agua que se produce dentro del tubo también ejerce su propia presión, que depende de la temperatura. Una corrección aproximada para este fenómeno, en el rango de utilización del barómetro, se puede incluir mediante la línea:

$$80 \text{ PRESION}=\text{PRESION}*\text{T}/(\text{TEMP}+273)-(\text{TEMP}+10)/27$$

Esta línea dará una lectura de presión tan exacta como el barómetro pueda generar. Compruebe los valores obtenidos confrontándolos con un barómetro comercial, o con un informe meteorológico reciente con cierta perioricidad, ya que determinados parámetros del barómetro pueden variar y quizá sea necesario modificar algunas constantes del programa. Por esto, debe considerarse este proyecto como experimental, y sólo válido para experimentar cambios de presión a corto plazo.

## Componentes

*Resistencias* (carbono, 0.25W, tolerancia 5%):

R1 220K.  
R2 100K.  
R3 820R.

R4 MKY7C38E resistencia dependiente de la luz.  
R5 100K.

### *Condensador*

C1 47n poliéster

### *Circuitos integrados*

IC1 CD4011 cuádruple puerta NAND de dos entradas, CMOS.  
IC2 507C conversor de voltaje a tiempo.

### *Varios*

LP1 Bombilla de filamento de 5V, 60 mA.

Circuito impreso.

Zócalo de 8 patillas.

Zócalo de 14 patillas.

Conector para el puerto de *joystick*.

Espadines de 1 mm (5 unidades).

Tubo de plástico de unos 20 cm. de longitud y 5 milímetros de diámetro (tubo de oxigenación de acuarios).

Válvula para el tubo.

Bote de plástico o cristal (como los utilizados para guardar medicamentos).

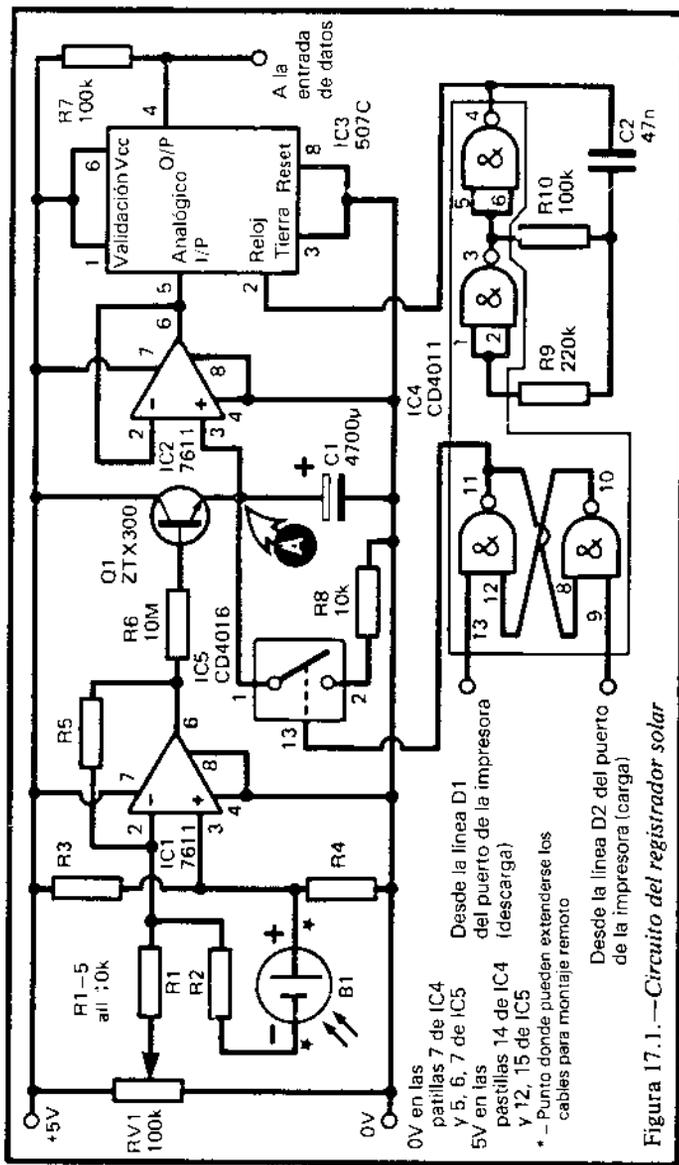
Otros materiales necesarios para la construcción del barómetro.

## Medidor solar

Acabaremos esta serie dedicada a la estación meteorológica con un proyecto que mide la cantidad de energía solar recibida en un período de tiempo. Es un circuito integrador, lo que significa que en lugar de informar de la energía que está llegando en cada momento, informa de la energía recibida desde la última lectura. Por tanto, el micro sólo tendrá que realizar lecturas de tiempo en tiempo. Las nubes que pasen por delante del Sol o la sombra que produzcan los árboles mientras que el micro está ocupado, se tendrán en cuenta cuando se realice de nuevo una lectura.

### Funcionamiento

Este es uno de los proyectos más complicados, pero algunas de sus características le resultarán familiares si ha construido alguno de los anteriores. La luz del sol incide en una célula fotovoltaica (V1, figura 17.1). Esta es una célula de silicio de las que, a menudo, se llaman células solares. Produce un voltaje de aproximadamente 0.45V, cuando está completamente iluminada, y voltajes menores si la iluminación es parcial. Este voltaje se alimenta a un am-



plificador operacional (IC1), que está configurado como un sumador. Su función es sumar el voltaje de la fotocélula con el del divisor de potencial RV1. La razón de esto se explicará más tarde. La salida de IC1 va al transistor Q1. Este actúa como un dispositivo de corriente constante que carga el condensador C1. Debemos hacer hincapié en el hecho de que, para cualquier corriente de base, la corriente de colector queda relativamente inalterada por la diferencia de potencial entre el conector y el emisor. A medida que C1 se carga, el potencial del punto A crece. El resultado es que la diferencia de potencial entre el colector y el emisor de Q1 decrece. De todas maneras, hasta que la diferencia de potencial no caiga muy por debajo de los niveles permitidos, la reducción de la diferencia de potencial no reducirá la corriente que fluya hacia C1. La cantidad de corriente depende de la corriente de base de Q1 que tiene su origen en IC1 y atraviesa R6. El valor de R6 es alto para mantener esta corriente baja. Hasta que el potencial de salida IC1 no exceda de 0.6V, no se producirá corriente de base y Q1 estará al corte. Debemos asegurarnos que tan pronto como se produzca una diferencia de potencial en B1 empiece a fluir corriente hacia Q1. Esta es la razón del sumador. La salida de IC1 es la suma de dos entradas. RV1 está dispuesta para que produzca una entrada de 0.6V, de tal manera que cualquier entrada adicional debida a B1 actúe directamente sobre Q1.

En la realidad, el sumador es inversor, así que aplicaremos 1.9V desde RV1, esto es 0.6V menos que la mitad del voltaje que hay en la patilla 3 de IC1, y la célula se conecta con su terminal negativo a R2. El resultado es una salida positiva hacia Q1.

En un período de varias decenas de minutos, o quizá una hora, C1 llega a cargarse a una velocidad que depende de la cantidad de luz que incide en la célula. La diferencia de potencial en C1 crece deprisa

cuando el Sol es fuerte, más despacio cuando está nublado o al final del día, y no crece durante la noche. La diferencia de potencial se alimenta a otro amplificador operacional, IC2, y está configurado como un seguidor de voltaje de ganancia unidad. Este se utiliza como un *buffer* entre C1 y el conversor de voltaje a tiempo, IC2. La entrada de IC2 tiene una impedancia de aproximadamente 100K, que permitiría una corriente de fuga de C1 de al menos la misma magnitud que la que llega de Q1. Pero la impedancia de entrada de IC2, que es un integrado de tecnología CMOS, es extremadamente alta (aproximadamente 1 teraohmio o, lo que es lo mismo,  $1 \cdot 10^{12}$  ohmios). Las pérdidas por esta entrada son virtualmente inexistentes. Como se explicó en el proyecto 11, IC3 convierte el voltaje en una forma de onda. La forma de ésta depende del nivel del voltaje.

Para que IC3 funcione correctamente, el voltaje de entrada debe estar en el rango de 1.25 a 3.75V. Durante el período de medidas, la diferencia de potencial debe empezar un poco por encima de 1.25V y no debe exceder de 3.75V.

El micro tiene la tarea de ajustar la diferencia de potencial en C1 a un nivel ligeramente superior a 1.25V cada vez que se inicie un período de medidas. Descarga C1 por la resistencia R8, que está conectada con un interruptor CMOS (IC5). Esto está controlado por un *flip-flop* compuesto por dos puertas de IC4. Las otras dos puertas del integrado se utilizan para construir el reloj de IC3 (véase la página 121). El *flip-flop* está controlado por dos salidas de datos del puerto de la impresora. Un pulso bajo en una de estas líneas activa el *flip-flop*, que cierra el interruptor CMOS y comienza a descargar C1. El micro comprueba la diferencia de potencial leyendo la entrada de datos. Cuando la diferencia de potencial ha alcanzado un nivel adecuadamente bajo, el micro envía un pulso bajo por la otra línea de datos. Esta acción

desactiva el *flip-flop* que abre el interruptor CMOS, y comienza de nuevo a cargar C1. Se requiere un nivel alto en la patilla 13 de IC5 para cerrar el interruptor.

La longitud de cada período de medida se fija de tal manera que incluso con un sol brillante continuo, la diferencia de potencial en C1 nunca exceda de 3.75V.

## Construcción

El proyecto requiere cuatro líneas de conexión con el micro: una tierra, una línea de entrada de datos y dos del puerto de la impresora. Si la célula solar (B1) debe montarse lejos del micro, los cables deben prolongarse a partir de los puntos marcados con asteriscos (\*) en la figura 17.1. La célula solar no tiene por qué ser grande; de hecho, es mejor una pequeña, ya que produce menos corriente y permite períodos más largos entre medidas. Suele ser posible comprar pequeños fragmentos de célula solar a bajo precio en las tiendas especializadas. Debe recordarse que las células solares se rompen fácilmente, salvo que estén montadas en una caja de protección. Deben montarse de tal manera que queden protegidas del tiempo y los daños físicos, pero de tal forma que reciban la luz solar por completo durante la mayor parte del día. La célula debe situarse horizontalmente, mirando hacia arriba. La figura 17.2 muestra una forma de protegerla. Si, más adelante, encuentra que el condensador se carga muy rápido, podrá situarse una o dos capas de papel blanco sobre la célula, de tal manera que se reduzca la cantidad de luz que la alcanza. Recuerde que debe eliminar cualquier objeto que cubra la célula, especialmente hojas en otoño.

Conecte primero IC1 y Q1 a sus componentes asociados. En esta etapa no conecte IC5. El circuito puede probarse alimentándolo con 5 ó 6V. Conecte un voltímetro entre las bornas de C1 (punto A). Tape

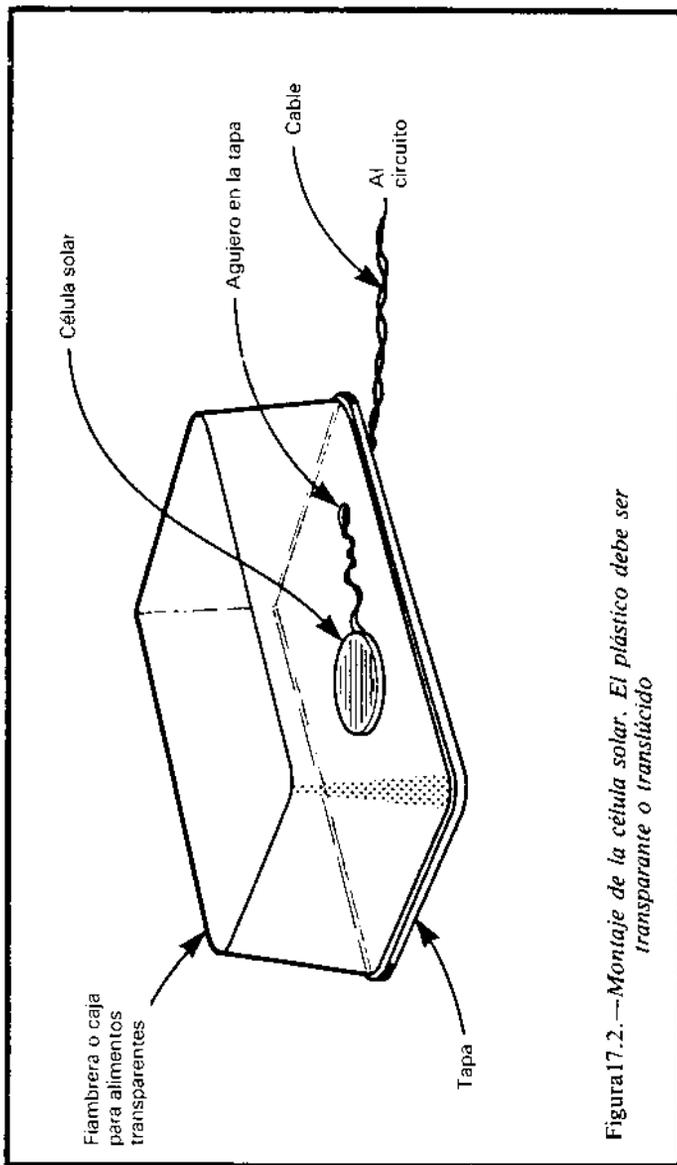


Figura 17.2.—Montaje de la célula solar. El plástico debe ser transparente o translúcido

B1 de tal manera que no le alcance la luz. Varíe la posición de RV1 varias veces. En algunas posiciones verá que la diferencia de potencial en C1 crece lentamente. Debe tardar varios minutos en incrementarse 0.1V. Ajuste RV1 de tal manera que justamente la diferencia de potencial no crezca. Este es el punto en el que suministra aproximadamente 1.V. Necesitará esperar unos diez minutos para asegurarse de que, efectivamente, la diferencia de potencial no crece. Descubra B1. La aguja del voltímetro debe empezar a moverse, mostrando un incremento en la diferencia de potencial. El ritmo de crecimiento depende de la cantidad de luz. Permita que la luz del Sol incida en la célula o mantenga una bombilla a pocos centímetros de ella y el ritmo de crecimiento aumentará perceptiblemente.

A continuación, añada IC2, IC3 e IC4. Si se prueba, como antes se indicó, con un voltímetro conectado a la salida de IC2 (patilla 6) se deberá obtener el mismo resultado. Con un osciloscopio conectado en la salida de IC3, se deberá ver una forma de onda similar a la de la figura 11.2. La proporción del tiempo durante el que la señal se mantiene en nivel alto decrece gradualmente a medida que C1 se carga.

Por último, conecte el *flip-flop* IC4 y el interruptor CMOS IC5. El *flip-flop* puede controlarse conectando sus entradas temporalmente a 5V y, a continuación, conectando una de ellas brevemente a 0V. Cuando el *flip-flop* cambia de estado, el interruptor se abre o se cierra. Cuando el interruptor está cerrado, C1 se descarga.

El condensador se descarga mucho más rápido de lo que se carga, pero aun así tarda algunos segundos. No debe descargarse muy rápidamente, o el micro será incapaz de detener la descarga en el punto adecuado.

Antes de conectar el dispositivo al micro, compruebe los cables para asegurarse de que no hay cor-

tocircuitos. Enchufe el dispositivo en el micro y conecte la alimentación. Permita que C1 se cargue hasta que la diferencia de potencial esté entre 1.25 y 3.75V, según la medida del voltímetro.

Utilice un programa como el de la página 122 para leer la entrada del conversor de voltaje a tiempo. Cubra la célula y fije RV1, de tal manera que la lectura obtenida por el micro no decrezca en un período largo de tiempo, aproximadamente diez minutos. Desde luego, los resultados individuales pueden variar en 2 ó 3 unidades, pero la media debe mantenerse. El punto fijado debe ser tal que cualquier ligero movimiento de RV1 hacia la línea de 0V produzca que el condensador empiece a cargarse, y que el número obtenido por el micro empiece a caer. A continuación, descubra la célula. La lectura debe bajar inmediatamente, y su velocidad de caída debe variar según la cantidad de luz que incida en la célula.

Descargue C1 hasta que la diferencia de potencial sea un poco mayor que 1.25V. Puede hacerlo conectando una resistencia de 10K entre sus bornas. Cuando el condensador esté descargado hasta el nivel correcto, ejecute el programa de nuevo varias veces. La lectura debe ser la cota superior del rango. Este valor quedará determinado como nivel de descarga. Suponiendo que el valor de nivel de descarga sea 420, el siguiente programa controlará el proceso:

```
10 OUT 61184,253: OUT 61184,255
20 COMIENZO=TIME: WHILE
    TIME<COMIENZO+50:WEND
30 OUT 61184,251: OUT 61184,255
40 X=0
50 FOR J=1 TO 500
60 IF JOY(0) AND 64=0 THEN X=X+1
70 NEXT J
80 IF X<420 THEN 10
90 ...resto del programa
```

Este programa funciona en un Amstrad y supone que la salida se conecta a la línea D6 del puerto de joystick y que el *flip-flop* está conectado al puerto de la impresora como indica la figura 17.1. En la línea 10, se produce un pulso bajo en la línea D1 del puerto de la impresora, mientras que las líneas restantes (incluyendo la D2) permanecen en nivel alto. Esto inicia la descarga de C1. Después de una décima de segundo (línea 20), la línea 30 genera un pulso bajo en la línea D2, sin afectar a las otras líneas (D1 inclusive).

Las líneas 40 a 80 leen el voltaje y comprueban si ha alcanzado el nivel de descarga. Si no lo ha hecho, el proceso se repite. Cuando se alcanza el nivel 0, el programa salta a la línea 90. A continuación, el micro pasa a otras rutinas, mientras que el condensador se carga de nuevo al ritmo correspondiente en función de la luz recibida. Estas rutinas deben incluir algún tipo de subrutina temporizadora, de tal manera que, por ejemplo, 30 minutos más tarde, el micro vuelva a leer el valor mostrado por el medidor solar. Este es proporcional a la cantidad de luz que ha incidido sobre la fotocélula durante el período de medida. Este valor puede escribirse o almacenarse en memoria para ser sumado a los demás valores obtenidos a lo largo del día.

## Componentes

*Resistencias* (carbón, 0.25W, tolerancia 5%, salvo indicación en contra):

R1-R5	10K tolerancia 1% (5 unidades).
R6	10M.
R7, R10	100K (2 unidades).
R8	10K.
R9	220K.
RV1	100K, potenciómetro ajustable.

*Condensadores*

C1	4700 $\mu$ electrolítico
C2	47n poliéster

*Semiconductor*

Q1	ZTX300 o transistor NPN similar.
----	----------------------------------

*Circuitos integrados*

IC1,IC2	7611 amplificadores operacionales CMOS (2 unidades).
IC3	507C conversor de voltaje a tiempo.
IC4	CD4011 cuádruple puerta NAND de dos entradas, CMOS.
IC5	CD4016 cuádruple interruptor analógico CMOS.

*Varios*

B1	Célula fotovoltaica de silicio (célula solar).
----	--

Zócalos de 8 patillas (3 unidades).  
 Zócalos de 14 patillas (2 unidades).  
 Conectores para los puertos de *joystick* e impresora  
 Placa de circuito impreso.  
 Espadines de 1 mm. (7 unidades).  
 Materiales para el montaje de la célula.

## Selector de datos

Este proyecto permite compartir cuatro líneas de datos entre dos dispositivos. Está diseñado especialmente para el uso con los proyectos de la estación meteorológica, pero puede usarse para cualquier otro proyecto que envíe datos al ordenador.

### Funcionamiento

El integrado utilizado en este proyecto contiene cuatro selectores de datos (Fig. 18.1). Cada selector de datos tiene dos entradas (A y B) y una salida (Y). El integrado tiene una entrada de selección que controla los cuatro selectores de datos. Cuando la entrada de selección está baja, el estado de la salida Y de cada selector de datos es el mismo que el estado de su entrada A. El estado de la entrada B se ignora. Por otra parte, cuando la entrada de selección está alta, el estado de la salida muestra la entrada B, y se ignora la entrada A. Esta es una forma de conectar ocho líneas de datos de periféricos a cuatro entradas del micro. Podemos leer los datos de un conjunto de cuatro líneas o del otro, simplemente variando el nivel de la entrada de selección. Esto es particularmente conveniente en el MSX, ya que no es

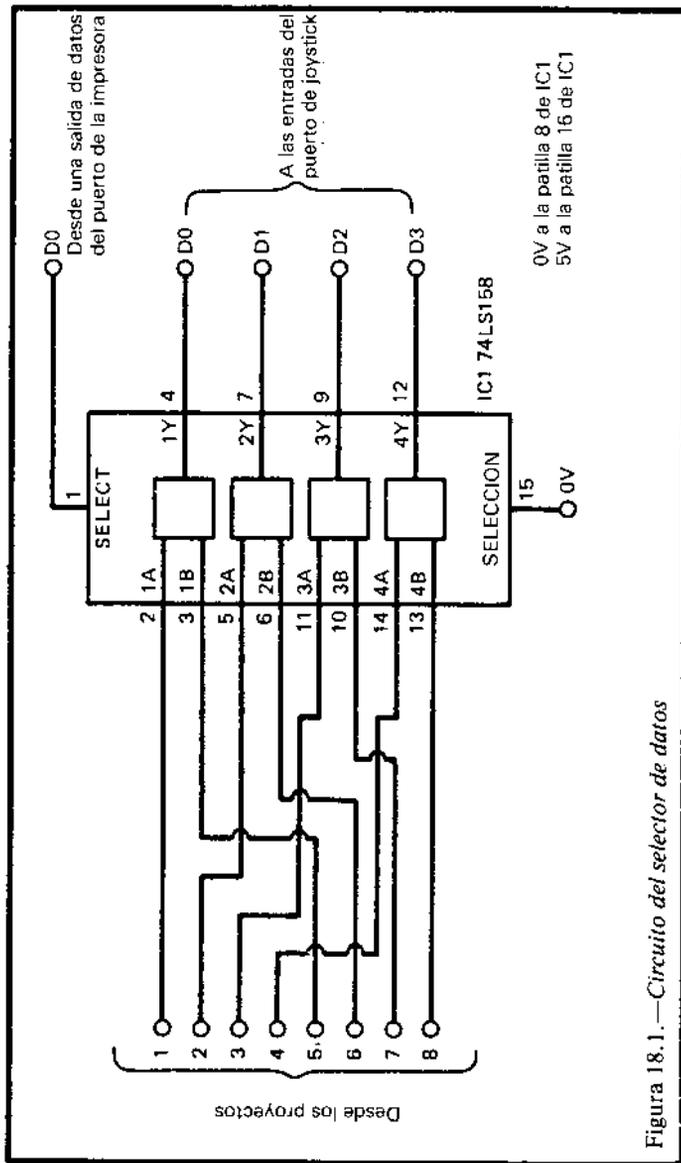


Figura 18.1.—Circuito del selector de datos

posible utilizar D0 al mismo tiempo que D1, o D2 al mismo tiempo que D3. Podemos utilizar sólo cuatro líneas de cada puerto (por ejemplo, D0, D2, D4 y D5). Utilizando el selector de datos podemos tener hasta ocho líneas, o incluso 16 si situamos un selector de datos en cada puerto de *joystick*. La entrada de selección se controla mediante una salida del puerto de la impresora.

## Construcción

Es extremadamente simple, aunque, debido al gran número de líneas, debe tenerse especial cuidado en que no se crucen los cables. El proyecto necesita seis conexiones con el micro: la tierra, cuatro cables al puerto de *joystick* y uno a la impresora. Cuando el dispositivo esté construido, compruébelo cuidadosamente antes de conectarlo al micro, para asegurarse de que no hay cortocircuitos y que las líneas están conectadas correctamente.

## Programación

Los programas escritos para usar con este dispositivo son los mismos que se describieron en proyectos anteriores, con la salvedad de que debe controlarse el nivel de la línea de selección. Supongamos que el selector se controla con la salida D0. Empiece el programa con la línea:

10 OUT 61184,256

Si está utilizando un MSX, la línea deberá ser:

10 OUT 145,254

Esto hace que la línea D0 del puerto de la impresora pase a nivel bajo y que la entrada de selección de IC5 también lo haga. El puerto de *joystick* está ahora comunicándose con el grupo de periféricos A (1-4 en la figura 18.1). Cualquier lectura hecha utilizando JOY(0), STICK (1), STRIG (1) o STRIG (3) mostrará el estado de estos cuatro periféricos. Para leer los datos del grupo B (5-8 en la figura 18.1), utilice OUT 61184,255 o OUT 145,255 para hacer pasar la entrada de selección de IC1 a nivel alto. Para devolver el control al grupo A, repita la sentencia dada arriba en la línea 10. Estas sentencias suponen que no hay otros dispositivos conectados en el puerto de la impresora. Si los hubiera (por ejemplo, el medidor solar), el programa deberá alterar el estado de D0 sin afectar la salida de las otras líneas del puerto de la impresora. Como se sugirió en el proyecto 5, la forma más simple es utilizar una variable PUERTO para registrar el estado de las líneas. Supongamos que PUERTO mantiene el valor actual de las líneas de salida. La sentencia requerida para seleccionar el grupo A es:

```
10 PUERTO=PUERTO AND 254: OUT
   145,PUERTO
```

Para seleccionar el grupo B, utilice:

```
10 PUERTO=PUERTO OR 1: OUT
   145,PUERTO
```

## Componentes

### *Circuitos integrados*

IC1 74LS158 cuádruple selector de datos de dos a una línea.

### *Varios*

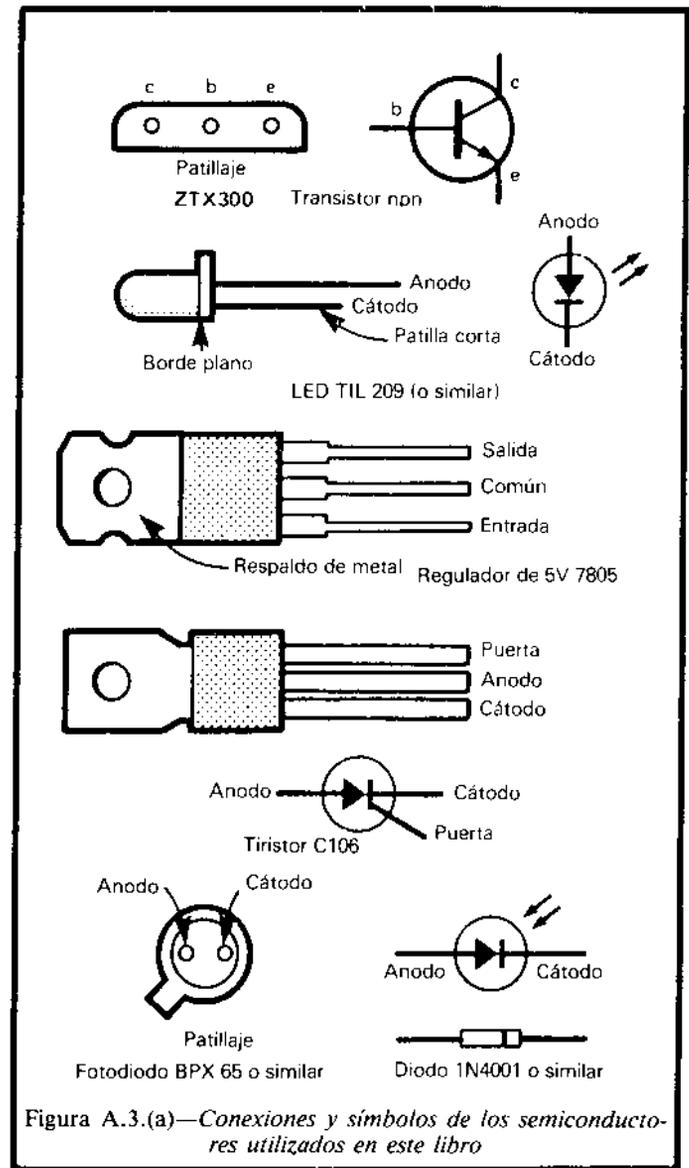
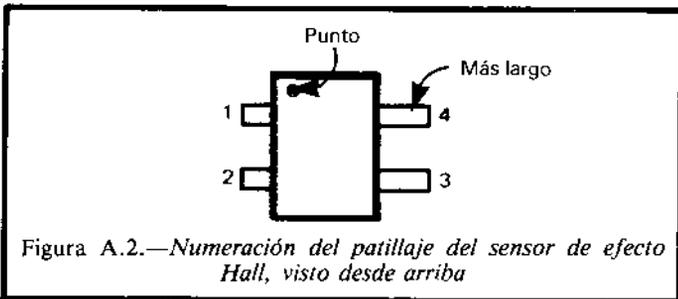
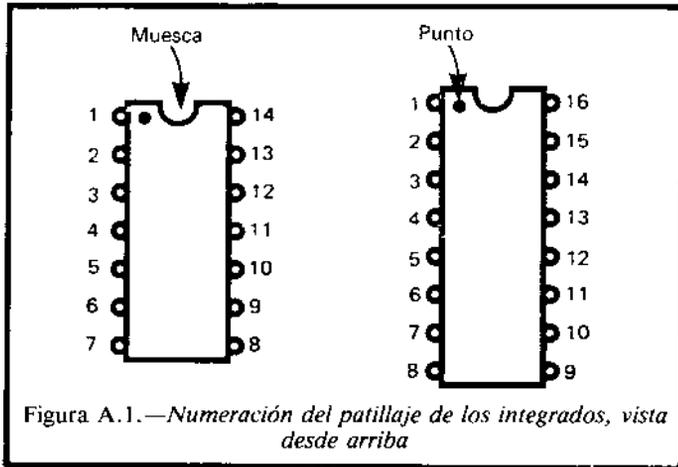
Placa de circuito impreso.

Conectores para los puertos de *joystick* y de impresora.

Espadines de 1mm. (16 unidades).

Zócalo de 16 patillas.

# Relación de los patillajes



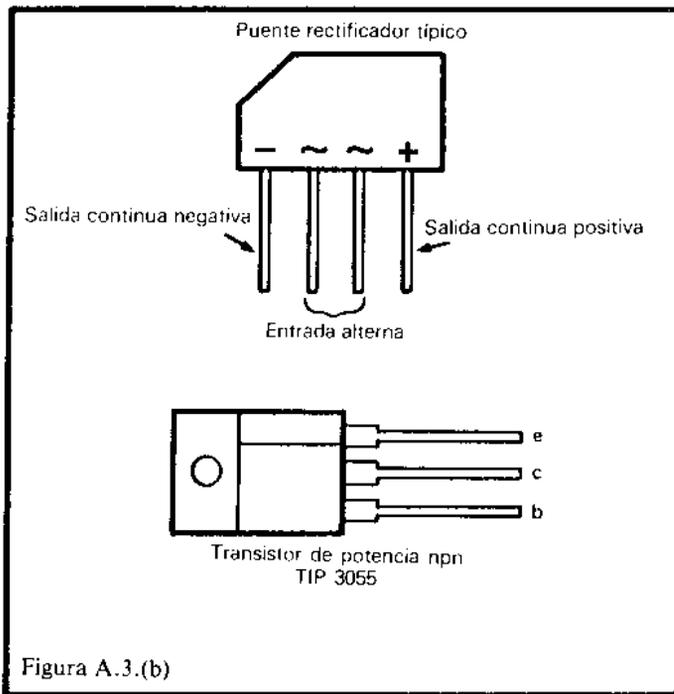


Figura A.3.(b)

# MICROMANUALES

Los MICROMANUALES son libros de referencia condensados que le ofrecen información práctica sobre todas las áreas de la microinformática: programación, aplicaciones, uso de programas, proyectos, etcétera.

## *Otros títulos de la colección*

- dBASE III
- Introducción a los sistemas operativos
- Proyectos de música con microordenadores
- SYMPHONY
- Introducción a las comunicaciones
- Programación del 8088/8086
- Proyectos *hardware* para Amstrad y MSX
- Técnicas y proyectos de interfaces
- FRAMEWORK
- Amstrad CPC 646/664/6128.  
Manual de referencia avanzado
- Tratamiento de textos con el Amstrad  
PCW 8256/8512
- LOTUS 1, 2, 3

