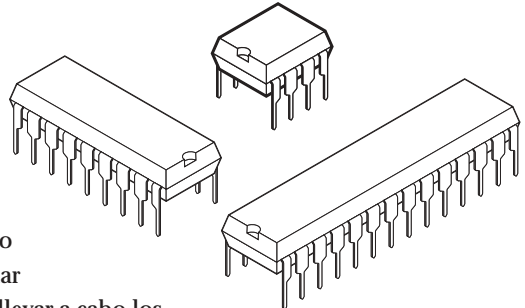


MASCOTAS ELECTRÓNICAS

¿Que es un microcontrolador?

El microcontrolador es a menudo descrito como un “ordenador en un chip”. El mismo se puede utilizar como “cerebro electrónico” para controlar productos, juguetes o máquinas.



El microcontrolador es un circuito integrado (chip) que contiene memoria (para almacenar programas), un procesador (para procesar y llevar a cabo los programas) y pines de entrada/salida (para conectar interruptores, sensores, y dispositivos de salida tales como motores).

Los microcontroladores se compran en “blanco” (vacíos) y luego se programan con un programa específico de control. Este programa es primero escrito en un ordenador y luego “descargado” en el chip del microcontrolador. Una vez programado, el microcontrolador se inserta dentro de un producto para hacer al producto más inteligente y fácil de utilizar.

Ejemplo del uso de un microcontrolador.

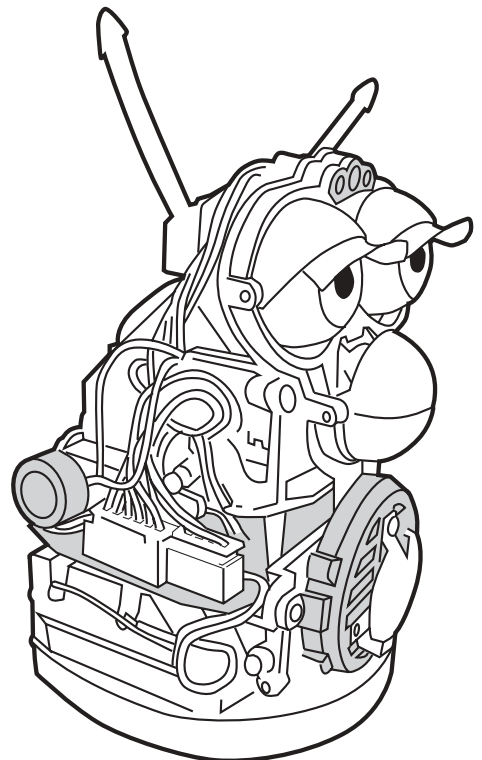
Hoy en día en la mayoría de las jugueterías se pueden conseguir juguetes “inteligentes”. Estos juguetes pueden moverse, hacer sonidos y responder al tacto o al ser movidos a un lugar oscuro.

Un ejemplo típico de uno de estos juguetes es el “Furby” fabricado por Tigre Electronics. El Furby utiliza un microcontrolador como cerebro electrónico y reacciona (al tocarlo o ponerlo en un lugar oscuro) moviéndose o emitiendo sonidos.

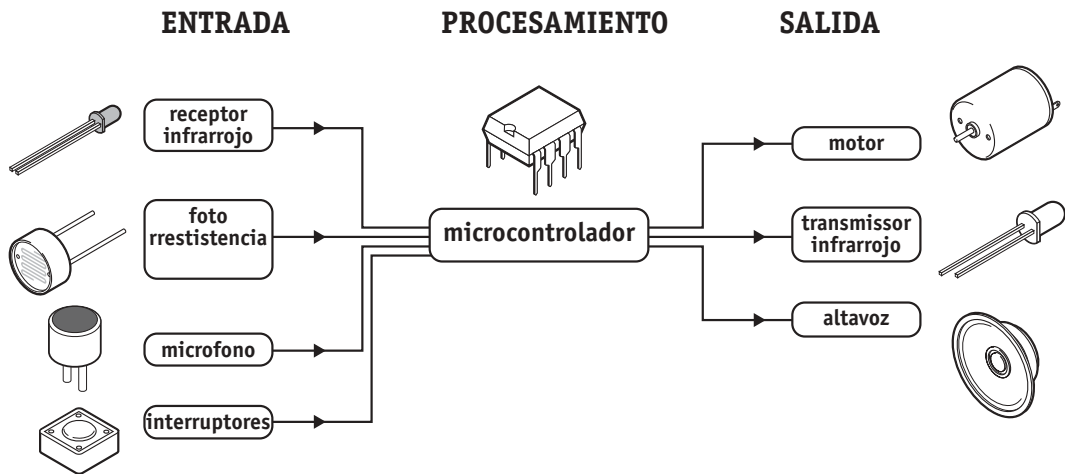
El Furby reacciona al mundo exterior gracias al uso de sensores e interruptores. Tiene un interruptor de presión en el frente y otro en la espalda, un micro-interruptor en su boca y un sensor de luz (fotorresistencia) entre sus ojos. También tiene un micrófono en un costado para detectar sonidos.

El Furby se mueve mediante el uso de un motor eléctrico. También tiene un altavoz para generar sonidos y un LED infrarrojo para enviarle señales a otros Furbys que puedan estar en las cercanías.

El microcontrolador es el “cerebro” de la criatura. Los microcontroladores son poderosos componentes electrónicos que tienen memoria y pueden programarse para encender y apagar dispositivos en una secuencia especial. Por ejemplo, el microcontrolador del Furby está programado para apagar el motor y el altavoz cuando el sensor de luz detecta que está oscuro (El Furby se va a dormir).



DIAGRAMAS DE BLOQUES



El sensor de luz, el micrófono y los interruptores proveen información al microcontrolador; por lo tanto se los conoce como “entradas”. Luego el microcontrolador “decide” como reaccionar y puede en determinados casos operar alguna de las salidas, por ejemplo hacer girar al motor o generar un sonido con el altavoz. Si hay otro Furby cerca, se pueden comunicar mediante señales infrarrojas transmitidas y recibidas por el microcontrolador

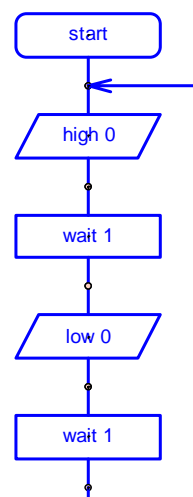
¿QUE ES EL SISTEMA PICAXE?

Los microcontroladores utilizados en dispositivos tales como el Furby pueden ser difíciles de programar, ya que generalmente utilizan un lenguaje de programación muy complejo llamado “código ensamblador” el cual puede ser muy difícil de aprender.

El sistema PICAXE permite una programación de microcontroladores mucho más sencilla. La secuencia de control puede dibujarse (y simularse) en el ordenador como un organigrama, o escribirse en un sencillo lenguaje de programación llamado BASIC. Esto hace que la utilización del microcontrolador sea una operación mucho más fácil ya que no es necesario aprender el complejo “código ensamblador”.

A continuación se muestra un ejemplo de programa BASIC con su respectivo organigrama. En este caso ambos programas hacen la misma cosa – encender y apagar una luz (conectada a la salida 0) cada segundo.

```
main:
  high 0
  wait 1
  low 0
  wait 1
  goto main
```



CONSTRUYENDO SU PROPIA MASCOTA ELECTRÓNICA

Descripción del Diseño

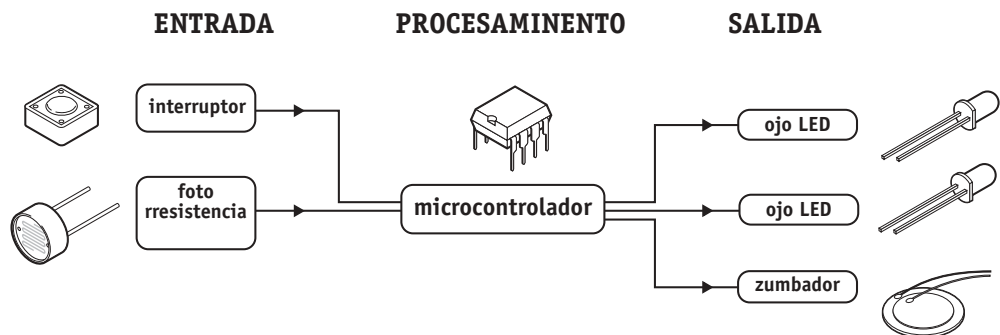
Diseñar y hacer un juguete de “ciber-mascota” electrónica. La ciber-mascota debe programarse con su propia “personalidad” de manera que reaccione en una manera única.

Especificaciones del Diseño

1. El diseño utilizará un microcontrolador PICAXE-08 como su cerebro.
2. El diseño incluirá ojos LED, un zumbador electrónico para generar sonidos y opcionalmente también utilizará un motor para generarle movimiento.
3. El diseño será capaz de reaccionar al tacto y a cambios en el nivel de luz.
4. La ciber-mascota puede diseñarse como un panel bidimensional plano o como una criatura tridimensional.

Diagrama de Bloque

El diagrama de bloque para su ciber-mascota puede verse así:



Diseñando su Ciber-mascota

Su ciber-mascota puede ser de cualquier forma o tamaño que usted escoja. Ya sea que usted quiera diseñar la “cara” de su mascota utilizando un programa editor de gráficos o dibujándola a mano; o si desea puede escanear la foto de un animal, ó diseñar un animal robot completamente nuevo.

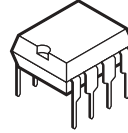
Detalles para tomar en cuenta:

Los componentes electrónicos deben montarse dentro (o debajo) de su ciber-mascota. Los LEDs y fotorresistencias deberán hacerse pasar a través de agujeros (normalmente estos dispositivos son de 5 mm de ancho, aunque hay LEDs disponibles en otros tamaños). También debe pensar cuidadosamente en la posición en donde va a colocar las baterías y donde va a conectar los cables.

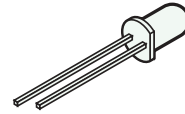
COMPONENTES ELECTRÓNICOS

A continuación se muestran los componentes principales que podría necesitar para su ciber-mascota. Las siguientes páginas describen en más detalle cada uno de estos componentes, y también proveen algunas ideas de programación que podrían serles útil posteriormente cuando esté programando su ciber-mascota con su propia personalidad.

Microcontrolador PICAXE-08



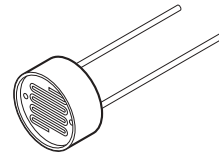
diodo emisor de luz (LED)



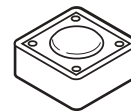
zumbador electrónico



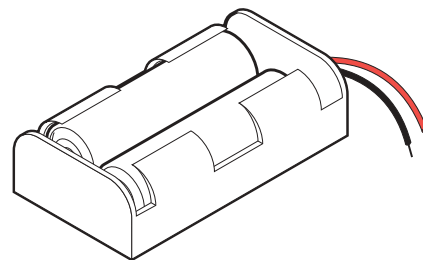
fotorresistencia



interruptor

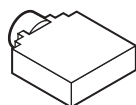


pilas eléctricas



y también necesitará

conector picaxe para descarga



resistencias

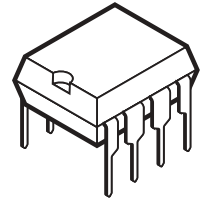


SECCIÓN 2 COMPONENTES ELECTRÓNICOS

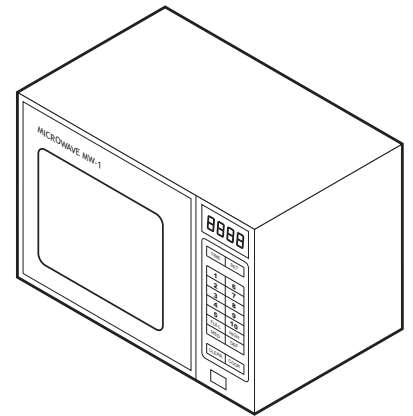
MICROCONTROLADORES

¿Que es un microcontrolador?

El microcontrolador es a menudo descrito como un “ordenador en un chip”. Es un circuito integrado que tiene memoria, unidades procesadoras y circuitos de entrada/salida en una sola unidad.



Los microcontroladores se compran en “blanco” (vacíos) y luego son programados con un programa específico de control. Este programa es primero escrito en un ordenador y luego “descargado” en el chip del microcontrolador. Una vez programado, el microcontrolador es insertado dentro de un producto para hacer al producto más inteligente y fácil de utilizar.



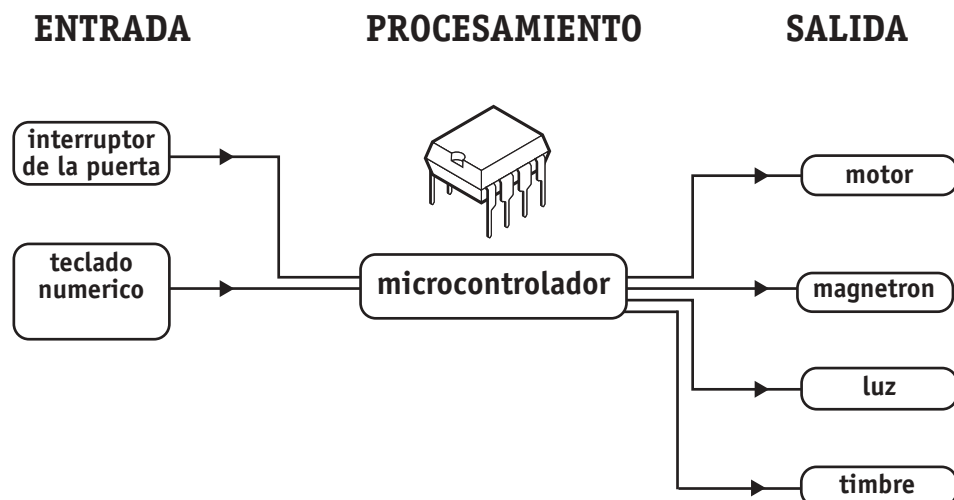
¿Donde se utilizan los microcontroladores?

Algunas de las aplicaciones que utilizan microcontroladores incluyen artefactos domésticos, sistemas de alarma, equipo médico, subsistemas de automóviles y equipo electrónico de instrumentación. Algunos automóviles modernos contienen más de treinta microcontroladores – utilizados en una amplia variedad de subsistemas desde el control del motor hasta el cierre a control remoto.

A manera de ejemplo, un horno de microondas puede utilizar un solo microcontrolador para procesar información proveniente del teclado numérico, mostrar información para el usuario en la pantalla y controlar los dispositivos de salida (motor de la mesa giratoria, luz, timbre y magnetrón).

¿Como se utilizan los microcontroladores?

Los microcontroladores se utilizan como “cerebro” en los circuitos electrónicos. Estos circuitos electrónicos se representan a menudo gráficamente como “diagramas de bloques”. Por ejemplo, para el horno de microondas descrito arriba, se podría dibujar un diagrama de bloques de la siguiente manera:

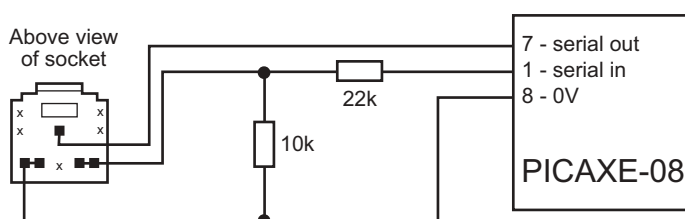


¿Cómo se escriben los programas?

Los programas se dibujan como organigramas o se escriben como listados de comandos BASIC. Esto será explicado más adelante en este libreto, en la sección de programación (sección 3).

¿Cómo se transfiere el programa al microcontrolador?

El microcontrolador PICAXE-08 se programa conectando un cable desde el puerto serie del ordenador a un conector en el circuito impreso (PCB) a un lado del microcontrolador. Este conector (el cual se parece a los conectores de audífonos utilizados en los reproductores portátiles de CD) se conecta a dos patas del microcontrolador y a la conexión de 0 V desde la batería. Esto permite que el ordenador y el microcontrolador “hablen” para permitir la descarga de un nuevo programa en la memoria del microcontrolador.



El conector y el circuito de interfase se incluyen en todo circuito impreso diseñado para utilizarse con el microcontrolador PICAXE-08. Esto permite reprogramar al microcontrolador PICAXE sin sacar el chip del circuito impreso - ¡simplemente conecte el cable cada vez que desee descargar un nuevo programa!

A menudo los diagramas de circuitos de los circuitos PICAXE no incluyen los componentes mencionados arriba para hacer más fácil la comprensión de las conexiones de entradas/salidas. Sin embargo, las dos resistencias y el conector están incluidos en todo tablero PICAXE.

Salida 0

En el sistema PICAXE-08 la pata 7 tiene dos funciones – cuando se está ejecutando un programa, la pata se denomina salida 0 y puede controlar salidas tales como LEDs y motores.

En cambio, cuando se está descargando un programa la misma pata actúa como pin de salida serie, comunicándose con el ordenador. Por lo tanto, si durante esta operación también tiene conectada a la pata una salida tal como un LED, se percatará que el mismo se encenderá y apagará continuamente mientras se descarga el programa.

Nota:

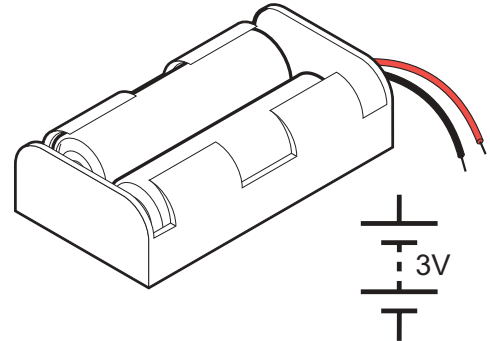
La mayor parte de los ordenadores modernos tienen dos puertos serie, usualmente denominados COM1 y COM2. El software Editor de Programación utilizado para crear los programas debe configurarse con el puerto serie correcto – seleccione Ver>Opciones>Puerto Serie para elegir el puerto serie correcto en su máquina..

Si usted está utilizando un nuevo ordenador portátil puede que este sólo tenga un conector del tipo USB. En este caso para poder utilizar el Sistema PICAXE deberá comprar un adaptador USB a serie. Estos adaptadores se pueden adquirir en la mayoría de las tiendas especializadas de ordenadores o en Internet en www.tech-supplies.co.uk (número de parte: USB010).

BATERÍAS

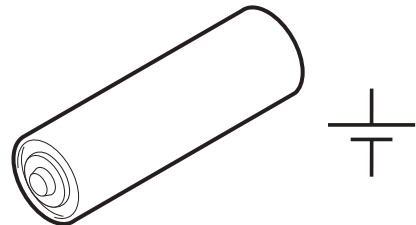
¿Qué es una batería?

Una batería es una fuente auto-contenida de energía eléctrica. Es una fuente de potencia portátil. Las baterías tienen químicos que almacenan energía. Al conectarlas a un circuito esta energía química se convierte en energía eléctrica que puede luego alimentar al circuito.



¿Qué tamaño de batería debo utilizar?

Las baterías vienen en todo clase de tipos y tamaños. La mayoría de las baterías consisten en un grupo de pilas, en donde cada pila provee cerca de 1.5V. Por lo tanto 4 pilas crean una batería de 6V y 3 pilas una de 4.5V.



Como regla general, mientras más grande es la batería, más tiempo durará (ya que contiene más químicos y por lo tanto será capaz de convertir más energía). Una batería de mayor voltaje no dura más que una batería de menor voltaje. Por lo tanto, una batería de 6V formada por 4 pilas AA dura mucho más que una batería PP3 de 9V, ya que por ser físicamente más grande contiene una mayor cantidad total de energía química. Por lo tanto, aquellos equipos que requieren mucha potencia para operar (Por ejemplo un reproductor portátil de CDs el cual tiene un motor y un láser para leer los CDs) siempre utilizarán pilas AA y no baterías PP3.

Los microcontroladores generalmente requieren entre 3 a 6V para operar, y por lo tanto es mejor utilizar una batería formada por tres a cuatro pilas AA. Nunca utilice una batería PP3 de 9V ya que la alimentación de 9V puede dañar el microcontrolador.

¿Qué tipo de batería debo utilizar?

Los distintos tipos de baterías contienen diferentes químicos. Las baterías de carbón-Zinc son las más baratas, y son adecuadas para utilizarse en muchos circuitos de microcontroladores. Las baterías alcalinas son más costosas, pero tienen una vida mucho más larga cuando se necesita alimentar dispositivos que requieren mucha corriente tales como motores. Las baterías de Litio son mucho más costosas pero tienen una larga vida, y por lo tanto se utilizan comúnmente en circuitos de ordenadores.

Otro tipo de baterías son las baterías recargables, las cuales pueden recargarse cuando se agotan. Estas están hechas usualmente de níquel y cadmio (Ni-cad) ó de hidróxido de metal cadmio (NiMH).

¡Seguridad!

Nunca haga corto-circuito en los terminales de una batería. Las baterías alcalinas y las recargables pueden suministrar corrientes muy grandes, y pueden calentarse tanto que pueden llegar a derretir la caja de baterías si se les hace corto-circuito. Siempre asegúrese de conectar la batería en el sentido correcto (rojo positivo (V+) y negro negativo (0V ó tierra)). Si las baterías se conectan al revés el microcontrolador corre peligro de calentarse y dañarse.

Utilizando los cables de las baterías

Los paquetes de baterías se conectan a menudo al circuito integrado mediante Cables de baterías. Asegúrese siempre que los cables rojo y negro estén conectados en la dirección correcta. También es de mucha utilidad pasar los cables de la batería a través de los agujeros del tablero antes de soldarlos en su lugar – esto proveerá una unión mucho más fuerte la cual será mucho menos propensa a soltarse.



Nunca conecte una batería PP3 de 9V a los cables de baterías – esto dañará al microcontrolador, ya que el mismo sólo trabaja con voltajes entre 3 y 6V.

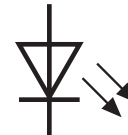
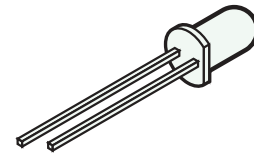
Soldadura sobre las cajas de baterías

Algunas cajas de baterías pequeñas requieren la soldadura de alambres a contactos metálicos en la caja. En este caso debe ser muy cuidadoso de no sobrecalentar los contactos metálicos. Si los contactos se calientan mucho, derretirán el plástico que los rodea y por lo tanto se caerán. Una buena manera de prevenir esto es pedirle a un amigo que sostenga los contactos metálicos con unas pinzas pequeñas. Las pinzas actuarán como un disipador de calor y ayudarán a evitar que el plástico se derrita.

DIODO EMISOR DE LUZ (LED)

¿Qué es un LED?

Un Diodo Emisor de Luz (LED) es un componente electrónico que emite luz cuando la corriente pasa a través de él. Un LED es un tipo de diodo especial. Un diodo es un componente que sólo permite el flujo de corriente en una dirección. Por lo tanto al utilizar un diodo, el mismo debe estar conectado en la dirección correcta.



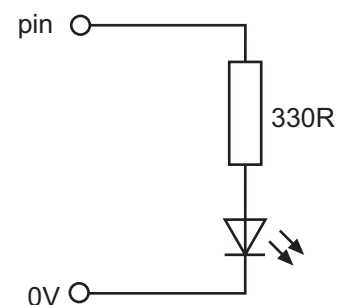
La pata positiva (ánodo) de un LED es más larga que la pata negativa (mostrada por una barra en el símbolo). La pata negativa también posee un extremo plano en la cubierta plástica del LED.

¿Para que se utilizan los LEDs?

Los LEDs se utilizan principalmente como luces indicadoras. Los LEDs rojos y verdes se utilizan comúnmente en artefactos electrónicos tales como televisores para mostrar si el televisor está encendido o si esta en el modo stand-by (en espera). Los LEDs están disponibles en una variedad de colores diferentes, incluyendo rojo, amarillo, verde y azul. Existen también LEDs ultra-brillantes, los cuales se utilizan en luces de seguridad tales como las luces intermitentes utilizadas en bicicletas. Los LEDs infrarrojos producen una luz infrarroja que no es visible al ojo humano pero que puede utilizarse en dispositivos tales como mandos a distancia de equipo de video.

Utilizando LEDs

Los LEDs sólo necesitan una pequeña cantidad de corriente para operar, esto los hace mucho más eficientes que las bombillas eléctricas (esto significa, por ejemplo, que si se tuviera una alimentación por baterías un LED alumbraría por mucho más tiempo que una bombilla eléctrica). Si se pasa demasiada corriente por un LED el mismo se puede dañar, es por esto que los LEDs normalmente se utilizan junto con una resistencia en serie, para protegerlos de corrientes excesivas.



El valor de la resistencia requerida depende del voltaje de la batería utilizada. Para una batería de 4.5V se puede utilizar una resistencia de 330R, y para una batería de 3V lo apropiado es una resistencia de 120R.

Conectando el LED a un microcontrolador

Debido a que el LED sólo requiere una pequeña cantidad de corriente para operar, el mismo se puede conectar directamente entre el pin de salida del microcontrolador y 0V (sin olvidar incluir la resistencia en serie para protección).

Probando la conexión del LED

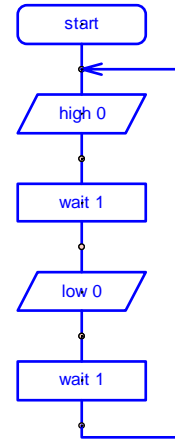
Después de conectar el LED el mismo puede probarse utilizando un simple programa tal como el siguiente:

```
main:
    high 0
    wait 1
    low 0
    wait 1
    goto main
```

Este programa debe encender y apagar el LED (conectado al pin de salida 0) cada segundo.

Si el LED no funciona verifique:

1. que el LED esté conectado en el sentido correcto
2. que se esté utilizando la resistencia correcta
3. que se esté utilizando el número de pin correcto dentro del programa
4. que todas las juntas estén bien soldadas



El siguiente programa enciende y apaga 15 veces al LED conectado al pin de salida 0 utilizando una técnica de programación BASIC llamada bucle *for...next* (esta técnica no puede utilizarse con organigramas). El número de veces que el código debe repetirse se almacena en la memoria del chip PICAXE utilizando una "variable" llamada b1 (el PICAXE tiene 14 variables nombradas de b0 a b13). Una variable es un "registro de almacenamiento de números" dentro del microcontrolador que el mismo puede utilizar para almacenar números a medida que el programa se ejecuta.

```
main:  for b1 = 1 to 15      \ iniciar un bucle for...next
        high 0            \ encender pin 0
        pause 500         \ esperar 0.5 segundos
        low 0             \ apagar pin 0
        pause 500         \ esperar 0.5 segundos
    next b1                \ fin del bucle

    end                    \ fin del programa
```

TIMBRES Y ZUMBADORES ELECTRÓNICOS

¿Qué es un zumbador electrónico?



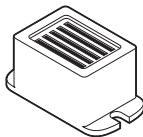
Un zumbador electrónico es un “mini-altavoz” de bajo costo que se utiliza para hacer sonidos. El sonido generado por el zumbador puede cambiarse alterando las señales electrónicas suministradas por el microcontrolador.



¿Para que se utilizan los zumbadores?

Los zumbadores se utilizan en una gran variedad de diferentes productos para dar “retroalimentación” al usuario. Un buen ejemplo de esto es una máquina expendedora, la cual emite un sonido cada vez que se presiona un botón para escoger un refresco o algo para picotear. Este sonido da retroalimentación al usuario para indicarle que se recibió la señal del botón presionado. Otros tipos de zumbadores se utilizan a menudo en tarjetas musicales de cumpleaños, para tocar una melodía cuando se abre la tarjeta.

¿Cuál es la diferencia entre un zumbador y un timbre ?



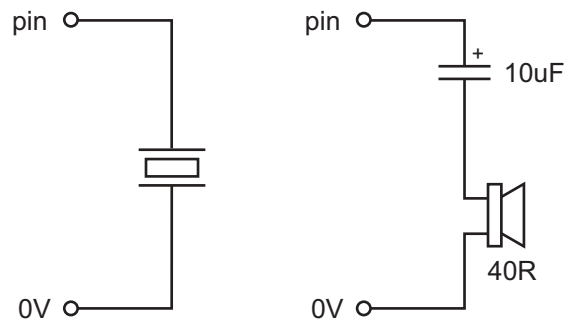
El timbre contiene un pequeño circuito electrónico el cual genera la señal electrónica necesaria para emitir un sonido. Por lo tanto cuando el timbre se conecta a una batería siempre emitirá el mismo sonido. El zumbador no tiene este circuito y por ende necesita una señal externa. Esta señal puede suministrarla un pin de salida del microcontrolador. El zumbador también requiere menos corriente para operar y por lo tanto durará más en circuitos alimentados por baterías.



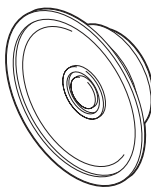
Utilizando zumbadores

La conexión de los zumbadores es muy sencilla. Simplemente conecte el cable rojo al pin de salida del microcontrolador y el cable negro a 0V (tierra).

Tome en cuenta que los zumbadores más económicos no tienen cubierta plástica exterior. En estos casos es necesario montar el zumbador sobre una sección del circuito impreso (con cinta adhesiva de doble contacto) para crear un sonido que se pueda escuchar. El circuito impreso actúa como una “caja de sonido” y amplifica el sonido emitido por el zumbador. Asegúrese de pegar la cinta adhesiva al lado correcto del zumbador (¡el lado debronce que no tiene los cables!).



Haciendo más ruido



En algunas ocasiones puede que desee emitir sonidos más fuertes. En este caso lo adecuado es utilizar un altavoz en vez de un zumbador. Al utilizar altavoces es necesario conectar un condensador (por ejemplo un condensador electrolítico de 10uf) al circuito para evitar causarle daños al microcontrolador. Recuerde que, al igual que el zumbador, los altavoces sólo operan correctamente si están montados en una “caja de sonido”.

Probando la conexión del zumbador

Después de conectar el zumbador, el mismo puede probarse utilizando un simple programa tal como el siguiente:

```
main:
    sound 2,(65,100)
    sound 2,(78,100)
    sound 2,(88,100)
    sound 2,(119,100)
    goto main
```

Este programa hará que el zumbador (conectado al pin de salida 2) haga 4 sonidos diferentes (valores 65, 78, 88, 119). Si el zumbador no funciona verifique:

1. que el valor del sonido (primer número en el paréntesis) esté entre 0 y 127
2. que se esté utilizando el número de pin correcto dentro del programa
3. que todas las juntas estén bien soldadas

Al utilizar el comando *sound*, el primer número indica el número de pin (en los proyectos el pin 2 es utilizado frecuentemente). El siguiente número es el tono, seguido por la duración. Mientras más alto es el tono, mayor será la altura tonal del sonido (tome en cuenta que algunos zumbadores no pueden emitir tonos muy altos y por lo tanto valores mayores de 127 puede que no se escuchen).

Al utilizar múltiples sonidos puede incluirlos todos en la misma línea. Por ejemplo:

```
sound 2,(65,100,78,100,88,100,119,100)
```

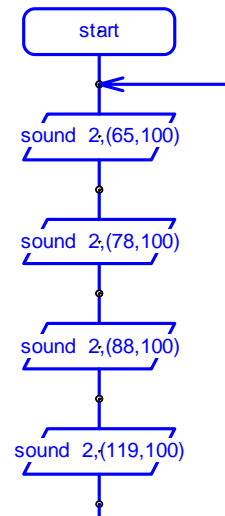
El siguiente programa BASIC utiliza un bucle *for...next* para emitir 120 sonidos diferentes, utilizando la variable *b1* para almacenar el valor (tono) del comando *sound*.

```
main:
    for b1 =1 to 120      ` iniciar un bucle for...next
        sound 2,(b1,50)  ` emitir un sonido con el tono b1
    next b1              ` siguiente b1
end
```

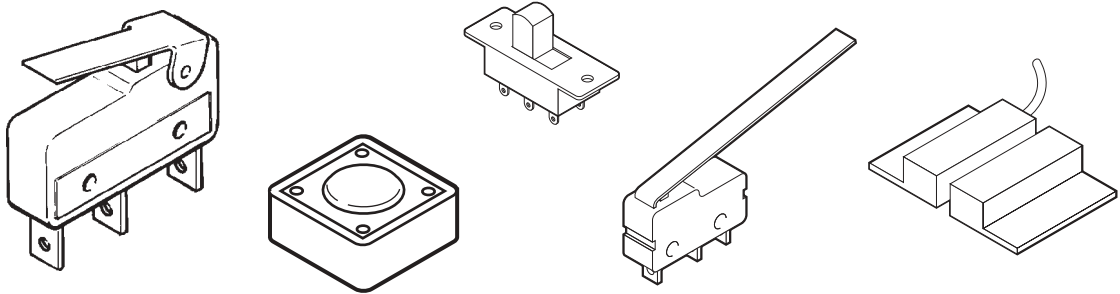
El número almacenado en la variable *b1* aumenta 1 en cada bucle (1-2-3-etc.). Por lo tanto, al utilizar la variable *b1* en la posición del tono, la nota se cambia en cada bucle.

El siguiente programa realiza la misma tarea pero en orden inverso (contando el tono en cuenta regresiva).

```
main:
    for b1 = 120 to 1 step -1 ` iniciar un bucle for...next
        sound 2,(b1,50)      ` emitir un sonido con el tono b1
    next b1                  ` siguiente b1
end
```

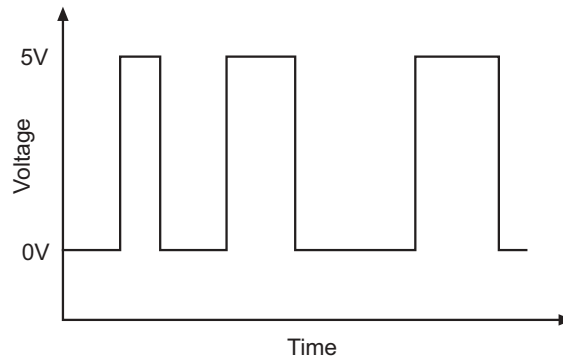


SENSORES DIGITALES (INTERRUPTORES)



¿Qué son los interruptores?

Un sensor digital es simplemente un sensor del tipo “interruptor”, el cual sólo puede estar en dos posiciones: encendido ó apagado. Si dibujáramos una gráfica de las señales de encendido-apagado del interruptor a medida que se oprime el mismo se vería así:



Los interruptores son componentes electrónicos que detectan movimiento. Existe una gran variedad de interruptores diferentes. Por ejemplo:

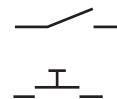
- interruptores de botón de presión que detectan cuando el botón ha sido oprimido
- micro-interruptores con palanquitas largas que detectan pequeños movimientos
- interruptores de nivel que detectan sacudidas
- interruptores de lengüeta (reed switches) que detectan el movimiento de un imán

¿Para que se utilizan los interruptores?

Los interruptores de botón de presión se utilizan comúnmente en dispositivos tales como teclados. Los micro-interruptores se utilizan en alarmas contra robos para detectar si se ha quitado la cubierta de la caja de la alarma. Los interruptores de lengüeta (reed switches) se utilizan para detectar la apertura de puertas y ventanas, y los interruptores de nivel se utilizan a menudo para detectar movimiento en dispositivos tales como juguetes, secadores y alarmas.

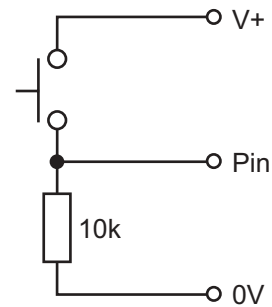
Símbolos de los Interruptores

Los símbolos para un interruptor deslizante y para un interruptor de presión se muestran a continuación:



Utilizando interruptores

Los interruptores se instalan en los circuitos junto a una resistencia como se muestra en el diagrama. El valor de la resistencia no es importante, pero a menudo se utiliza una resistencia de 10k. Cuando el interruptor está “abierto” la resistencia de 10k conecta el pin de entrada del microcontrolador a 0V, lo cual da una señal de apagado (nivel lógico 0) al pin de entrada del microcontrolador.



Cuando el interruptor se activa, el pin de entrada se conecta al terminal positivo de la batería (V+). Esto da una señal de encendido (nivel lógico 1) al microcontrolador

Probando el interruptor

Después de conectar el interruptor, el mismo puede probarse utilizando un simple programa tal como el siguiente. Este programa encenderá o apagará una salida dependiendo si el interruptor es presionado o no.

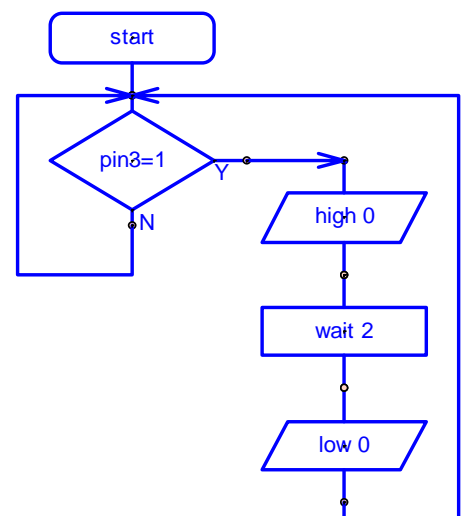
```

main:                                     ` hacer una etiqueta llamada
    if input3 is on then flash           ` saltar a flash si la entrada3
                                        ` (input3) está encendida
    goto main                           ` sino regresar al inicio

flash:                                    ` hacer una etiqueta llamada
    high 0                               ` encender salida 0
    wait 2                               ` esperar 2 segundos
    low 0                                ` apagar salida 0
    goto main                            ` regresar a inicio
    
```

En este programa las tres primeras líneas constituyen un bucle continuo. Mientras la entrada esté apagada el programa seguirá reiniciándose una y otra vez.

Si la entrada está encendida el programa salta a la etiqueta llamada “flash”. El programa luego enciende por 2 segundos la salida 0 antes de regresar al bucle principal.

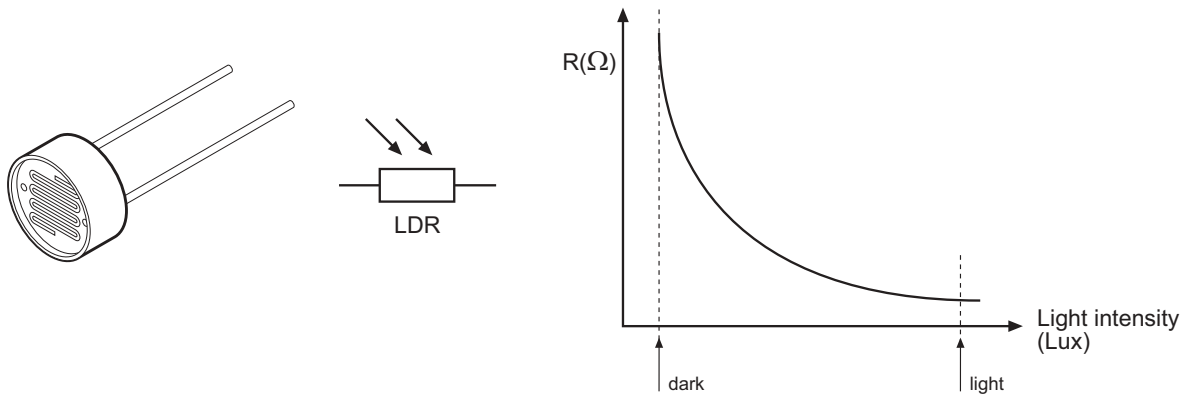


Note cuidadosamente la ortografía de la línea *if...then* – **input3** (entrada 3) es una sola palabra (sin espacios en blanco). Es permitido utilizar tanto **pin3** como **input3**, ambas significan lo mismo. Note también que después del comando *then*, únicamente está la etiqueta – en este punto no se permite ninguna otra palabra aparte de la etiqueta.

FOTORRESISTENCIAS

¿Qué es una fotorresistencia?

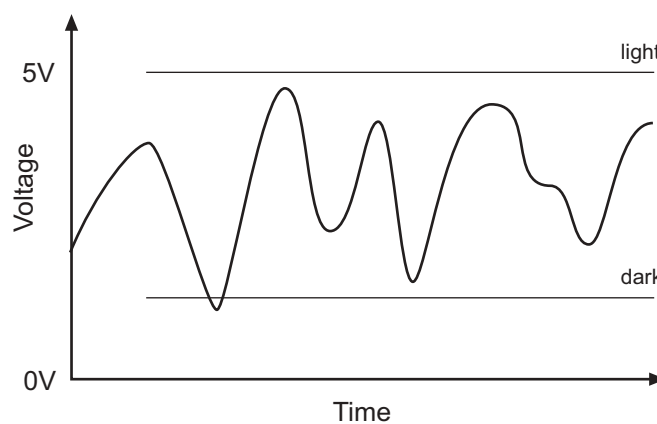
Una fotorresistencia es un tipo especial de resistencia la cual reacciona a cambios en el nivel de luz. La resistencia de la fotorresistencia cambia a medida que varía la cantidad de luz que incide sobre la “ventana” del dispositivo. Esto permite a los circuitos electrónicos medir cambios en el nivel de luz.



¿Para que se utilizan las fotorresistencias?

Las fotorresistencias se utilizan en las lámparas automáticas de las carreteras para encenderlas durante la noche y apagarlas durante el día. También se utilizan en muchas alarmas y juguetes para medir niveles de luz.

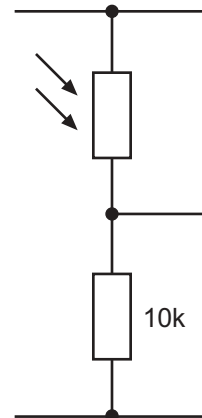
La fotorresistencia es un sensor del tipo analógico. Un sensor analógico mide una señal continua tal como luz, temperatura o posición (en vez de una señal digital de encendido/apagado como es el caso de un interruptor). El sensor analógico provee como señal un voltaje variable. Esta señal de voltaje puede representarse mediante un número entre el 0 al 255 (Por ejemplo: muy oscuro = 0, luz brillante = 255).



Utilizando fotorresistencias

Las fotorresistencias se pueden utilizar en dos maneras. La manera más simple de utilizar una fotorresistencia es como un simple interruptor de encendido/apagado (digital) – cuando el nivel de luz sube por arriba de cierto valor (llamado el valor de “umbral”), la fotorresistencia enviará una señal de encendido; cuando el nivel de luz está debajo de cierto nivel, la fotorresistencia enviará una señal de apagado.

En este caso la fotorresistencia se coloca en una configuración de divisor de voltaje con una resistencia estándar. El valor de la resistencia estándar define el “valor de umbral”. Para fotorresistencias miniatura el valor adecuado es 1 K, para fotorresistencias más grandes del tipo ORP12 una resistencia de 10 K es más adecuada. Si se desea la resistencia fija puede reemplazarse por una resistencia variable de manera que el valor de umbral se pueda ajustar para diferentes valores del nivel de luz.



Una manera más versátil de utilizar la fotorresistencia es midiendo un cierto número de valores de luz diferentes, de manera que las decisiones se puedan tomar para niveles de luz cambiantes en vez de un solo nivel de umbral fijo. Un valor cambiante se conoce como un valor “analógico”.

Para medir valores analógicos el microcontrolador debe tener un convertidor analógico/digital y además el software utilizado debe ser capaz de soportar el uso de este convertidor. La mayoría de los microcontroladores solo tienen convertidores a-d en algunos de sus pines de entrada, por lo tanto el pin de entrada se debe seleccionar cuidadosamente. En el microcontrolador de 8 patas sólo uno de los pines puede utilizarse.

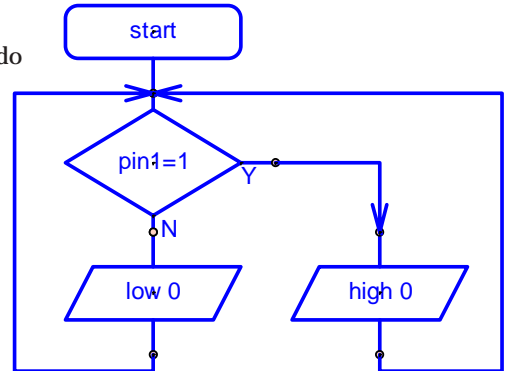
El circuito electrónico para utilizar el convertidor a-d es un circuito divisor de voltaje idéntico al mostrado arriba. Las “mediciones” analógicas se realizan dentro del microcontrolador mismo.

Probando la fotorresistencia (digital)

Luego de conectar la fotorresistencia, la misma se puede probar como un interruptor digital utilizando un simple programa tal como el siguiente:

```
main:
    if input1 is on then dohigh
    low 0
    goto main

dohigh:
    high 0
    goto main
```



Este programa encenderá y apagará la salida0 de acuerdo al nivel de luz.

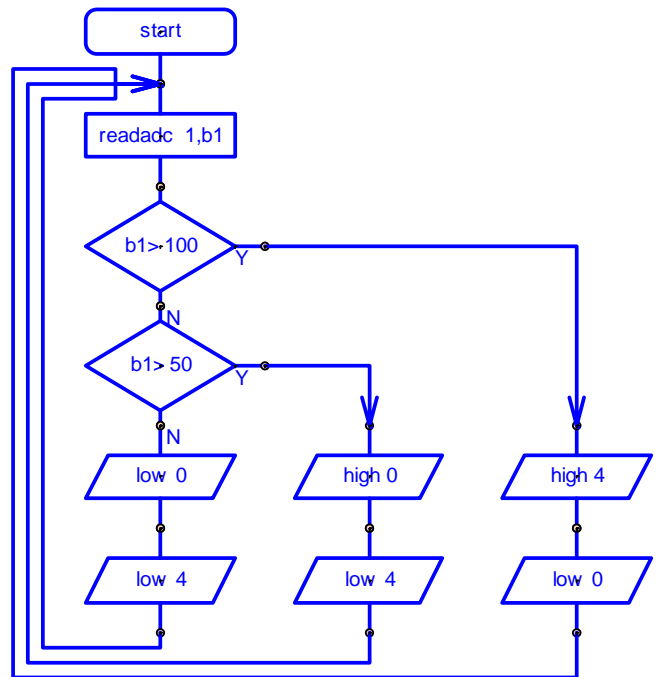
Probando la fotorresistencia (analógico)

Luego de conectar la fotorresistencia, la misma se puede probar como un sensor analógico utilizando un simple programa tal como el siguiente:

```
main:
    readadc 1,b1
    if b1 > 100 then do4
    if b1 > 50 then do0
    low 0
    low 4
    goto main

do4:
    high 4
    low 0
    goto main

do0:
    high 0
    low 4
    goto main
```



El comando "readadc" se utiliza para leer el valor analógico (un número entre 0 y 255) dentro de la variable b1. Una vez que este número está en la variable b1, el mismo se puede probar para ver si es mayor que 100 ó mayor que 50. Si es mayor que 100 la salida 4 se enciende, si está entre 50 y 100 se enciende la salida 0, y si es menor de 50 ambas salidas se apagan.

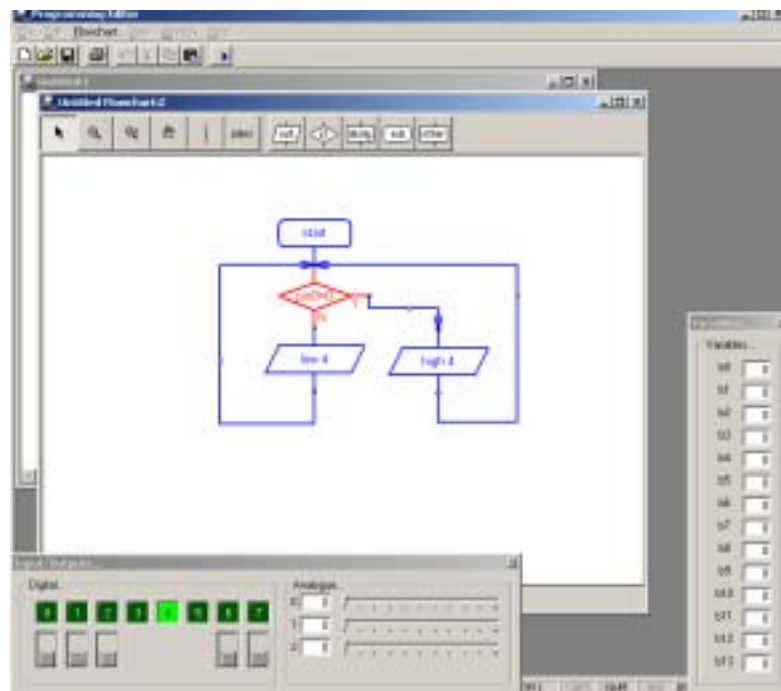
SECCIÓN 3

PROGRAMANDO – DIBUJANDO ORGANIGRAMAS

Los organigramas son una herramienta muy útil que permiten representar gráficamente (dibujar) a los programas para hacerlos más fáciles de entender. El software Editor de Programación incluye un editor de organigramas que permite dibujar organigramas en la pantalla del ordenador. Estos organigramas se pueden convertir luego en código BASIC para descargarlos en el PICAXE. Los organigramas también pueden imprimirse y exportarse como figuras para incluirlos dentro de reportes de proyectos.

Instrucciones detalladas para dibujar/descargar un organigrama:

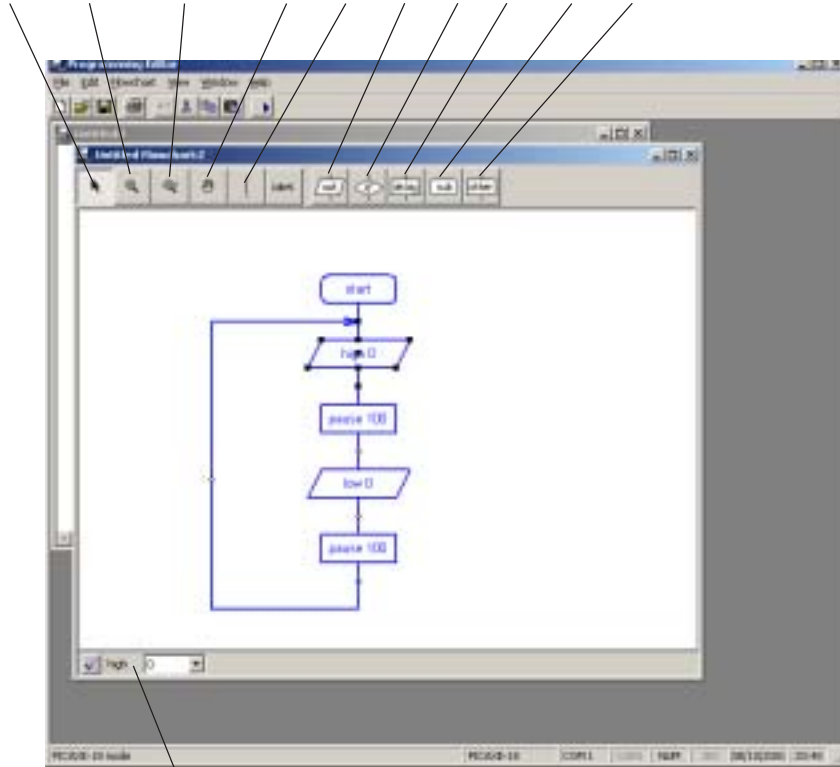
1. Conecte el cable PICAXE a uno de los puertos serie del ordenador. Recuerde tomar nota del puerto serie al cual conecta el cable (normalmente COM1 ó COM2).
2. Inicie el software “Editor de Programación”
3. En el menú desplegable seleccione Ver>Opciones para acceder a la pantalla de opciones (esta puede que aparezca automáticamente).
4. Haga clic en la lengüeta “Modo” y seleccione PICAXE-18.
5. Haga clic en la lengüeta “Puerto Serie” y seleccione el puerto serie al cual ha conectado el cable PICAXE. Haga clic en “OK”
6. Cree un nuevo organigrama haciendo clic en el menú Archivo>Nuevo Organigrama.
7. Dibuje el organigrama arrastrando los bloques requeridos a la pantalla y luego utilizando el ratón para dibujar flechas para conectar los bloques.
8. Cuando termine de dibujar el organigrama, puede convertirlo en un programa BASIC seleccionando el menú Organigrama>Convertir Organigrama a BASIC. Luego el programa BASIC puede descargarse en el PICAXE seleccionando en el menú PICAXE>Ejecutar.
9. Para imprimir o salvar el organigrama, utilice las opciones en el menú de Archivo. Para exportar el organigrama como figura, utilice el menú Archivo>Exportar. Para exportar la imagen a un documento de Word seleccione el archivo tipo EMF. Para exportar el organigrama a una página web use el archivo tipo GIF.



Pantalla del Editor de Organigramas

El Editor de Organigramas permite dibujar y simular organigramas en la pantalla. El organigrama puede luego convertirse automáticamente en un programa BASIC para descargarlo en el microcontrolador.

Selecciona Zoom Acercar/Alejar Mover Línea Salida Si Retardo Sub Ocio



barra editora

Seleccionar

Utilice este comando para seleccionar y mover bloques. Cuando se selecciona un sólo bloque, su código BASIC puede editarse en la barra editora en la parte inferior de la ventana.

Zoom

Utilice para acercar una parte del diagrama. Use el clic derecho para alejar.

Acercar/Alejar

Para acercar haga clic y mueva el ratón hacia arriba. Para alejar haga clic y mueva el ratón hacia abajo.

Mover

Utilice este comando para mover el organigrama completo alrededor de la pantalla.

Línea

Utilice este comando para dibujar líneas entre los bloques. Se pueden hacer quiebres en las líneas haciendo clic una vez. Cuando la línea está cerca de un bloque, esta se pegará al punto de conexión del mismo.

Etiqueta

Utilice este comando para añadirle etiquetas o títulos a los elementos del organigrama.

Salida/Si/Retardo/Sub/Otro

Haga clic en estos botones para ir al submenú de estos comandos y seleccionar el comando deseado.

Dibujando organigramas

Para dibujar un organigrama haga clic en uno de los botones de menús de comandos (Salida/Si/Retardo/Sub/Otro) de la barra de herramientas para ir al submenú de comandos requerido. Seleccione el comando deseado y luego haga clic en la pantalla, en el lugar donde desea situar al comando. No trate de colocar el bloque exactamente en posición en primera instancia – póngalo en la pantalla en las cercanías del área donde desea ubicarlo y luego use el comando Seleccionar para mover el bloque a la posición correcta.

Una vez que el bloque esté en posición, haga clic en él de manera que sea resaltado. El código BASIC del objeto aparecerá en la barra editora en la parte inferior de la pantalla. Edite el código si lo requiere.

Para información adicional acerca de cada comando vea el archivo de ayuda “Comandos BASIC”.

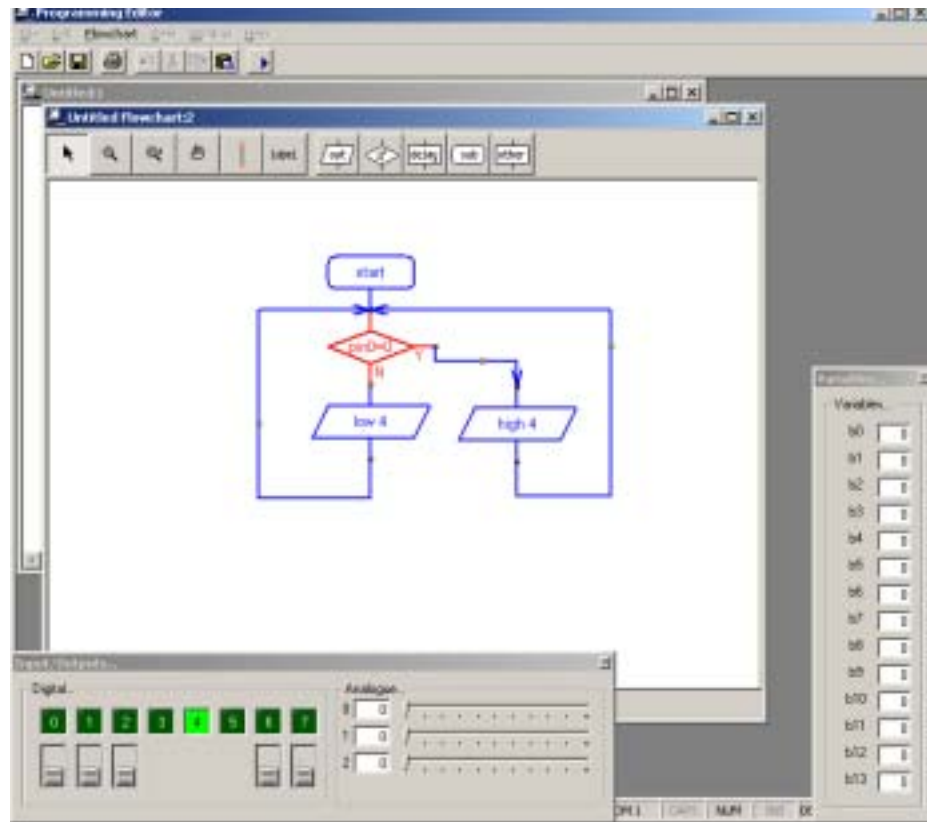
Uniando bloques

Para unir bloques, se debe acercarlos uno al otro hasta que se junten. Otra opción es dibujar líneas entre los mismos usando el comando *línea* en la barra de herramientas. Note que sólo es posible unir la parte inferior de un bloque únicamente con la parte superior de otro (no se pueden conectar líneas con líneas). Además, sólo se permite sacar una línea de la parte inferior de conexión de cada bloque.

Para hacer diagramas ordenados, se pueden agregar quiebres a las líneas haciendo clic en las mismas. Al mover una línea cerca de un punto de conexión, la misma se pegará a este; para terminar la línea haga clic una vez más y la misma quedará en posición.

Las líneas no pueden moverse. Si trata de mover una línea la misma será borrada y tendrá que crear una nueva línea.

Simulación en Pantalla



Para simular el organigrama, haga clic en “Simular” en el menú Organigrama. El programa comenzará a ejecutarse en pantalla.

A medida que el programa se ejecuta, los bloques cuyos comandos están siendo ejecutados se irán resaltando en rojo. Las ventanas de “Entradas/Salidas” y “Variables” también aparecerán mientras se ejecuta la simulación. Para cambiar los valores de las entradas haga clic en el respectivo interruptor en pantalla (mostrado debajo de el LED) ó utilice la barra deslizadora de entradas analógicas.

El tiempo de retardo entre un objeto y otro puede ser ajustado en las Opciones del Organigrama (menú Ver>Opciones>Organigrama).

Note que algunos comandos representan acciones que no pueden ser simuladas en pantalla. En estos casos el comando es simplemente ignorado al ejecutar el organigrama.

Descargando Organigramas

Los organigramas no se descargan directamente al microcontrolador. Primero el organigrama es convertido en un programa BASIC, el cual luego es descargado.

Para convertir un organigrama seleccione “Convertir” en el menú Organigrama; el programa BASIC del organigrama será creado.

Aquellos bloques que no estén conectados a los bloques “inicio” ó “sub” en el organigrama, serán ignorados al momento de hacer la conversión. La conversión se detendrá si se encuentra un bloque no conectado; por lo tanto, utilice siempre un bloque “detener” para terminar el diagrama antes de iniciar una simulación o de convertir el diagrama.

Note que es posible convertir y descargar rápidamente un organigrama presionando dos veces la tecla F5.

Utilizando Símbolos

Entradas, Salidas y Variables pueden renombrarse utilizando la “Tabla de Símbolos” del menú Organigrama. Cuando un símbolo es renombrado el nuevo nombre aparecerá en los menús desplegados en la barra editora. No deben utilizarse nombres de comandos (por ejemplo *switch* o *sound*) como símbolos ya que esto puede generar errores en el programa BASIC convertido.

Guardando y Imprimiendo Organigramas

Los organigramas pueden guardarse, imprimirse y exportarse como figuras (para insertarlos en documentos de procesadores de palabras) utilizando el menú Archivo. Los organigramas pueden también copiarse en el portapapeles de Windows (para pegarlos luego a otras aplicaciones) utilizando el menú Editar.

SECCIÓN 4

PROGRAMANDO – BASIC

La programación en BASIC es una herramienta de programación más poderosa que la utilización de organigramas. Esto se debe a que BASIC permite la utilización de más comandos - por ejemplo bucles *for...next* - los cuales no se pueden utilizar con el método gráfico de los organigramas. Sin embargo en la programación BASIC se requiere ser más preciso al escribir los programas, ya que no se permiten errores de ortografía.

El siguiente programa es una muestra de un programa BASIC el cual enciende y apaga la salida 0 cada segundo. Al descargar este programa el LED conectado a la salida 0 se encenderá y apagará cada segundo.

```
main:
    high 0
    pause 1000
    low 0
    wait 1
    goto main
```

Este programa utiliza los comandos *high* y *low* para controlar el pin de salida 0, y utiliza los comandos *pause* y *wait* para crear un tiempo de retardo. El comando *wait* opera con unidades en segundos enteros mientras que *pause* utiliza milisegundos (1000 ms = 1 segundo). Por lo tanto, en este programa ambos tiempos de retardo tienen la misma duración, sólo que están escritos de maneras distintas.

El comando *goto* hace que el programa “salte” a la etiqueta **main**: al inicio del programa. Esto significa que el programa se ejecutara continuamente una y otra vez. Note que la primera vez que se utiliza la etiqueta, la misma debe estar seguida de un símbolo de dos puntos (:). Esto indica al ordenador que la palabra es una nueva etiqueta.

Instrucciones detalladas:

1. Conecte el cable PICAXE a un puerto serie del ordenador y tome nota a cual de los puertos lo conecta (normalmente COM1 ó COM2).
2. Ejecute el Software “Programming Editor”.
3. En el menú desplegable escoja Ver>Opciones para acceder la pantalla de opciones (esta puede que aparezca automáticamente).
4. Haga clic en “Modo” y seleccione PICAXE-08.
5. Haga clic en “Puerto Serie” y seleccione el puerto serie al cual el cable PICAXE está conectado. Haga clic en “OK”.
6. Escriba el siguiente programa:

```
main:
    high 0
    pause 1000
    low 0
    wait 1
    goto main
```

(No olvide el símbolo de dos puntos (:)) directamente después de la etiqueta “main” y los espacios entre los comandos y los números)

7. Asegúrese que el circuito PICAXE esté conectado al cable serie y a las baterías, y que las baterías estén conectadas.
8. Seleccione PICAXE>Ejecutar. Una barra de descarga de programa deberá aparecer mientras el programa es descargado. Al terminar la descarga, el programa debe comenzar a ejecutarse automáticamente – el LED de la salida 0 deberá encenderse y apagarse cada segundo.

Procedimientos de funciones básicas del Software

Iconos de la barra de herramientas:



Para descargar/ejecutar un programa BASIC:

1. Verifique que el cable de descarga esté conectado tanto al PICAXE como al puerto serie del ordenador.
2. Verifique que las baterías estén conectadas al PICAXE.
3. Asegúrese que el Software “Editor de Programación” esté en el modo correcto (La palabra PICAXE-08 debe aparecer en la barra de estado en la esquina inferior izquierda de la pantalla).
4. Haga clic en **PICAXE>Ejecutar** (ó en el respectivo icono de la barra de herramientas) (ó presione la tecla F5)

Para guardar un programa/organigrama:

1. Haga clic en **Archivo- Guardar como...** (ó en el respectivo icono de la barra de herramientas)
2. Escriba el nombre bajo el cual quiere guardar el archivo
3. Haga clic en <OK>

Para abrir un programa/organigrama guardado:

1. Haga clic en **Archivo- Abrir...** (ó en el respectivo icono de la barra de herramientas)
2. Seleccione el tipo de archivo (BASIC u organigrama)
3. Seleccione el archivo deseado de la lista haciendo clic en el mismo.
4. Haga clic en <OK>

Para crear un nuevo programa BASIC:

1. Haga clic en **Archivo- Nuevo**

Para crear un nuevo organigrama:

1. Haga clic en **Archivo- Nuevo Organigrama** (ó en el respectivo icono de la barra de herramientas)

Para simular en pantalla un organigrama:

1. Haga clic en **Organigrama – Simular...** (ó en el respectivo icono de la barra de herramientas)
2. Haga clic en el organigrama para detener la simulación.

Para convertir a BASIC un organigrama:

1. Haga clic en **Organigrama – Convertir a BASIC** (ó presione la tecla F5)

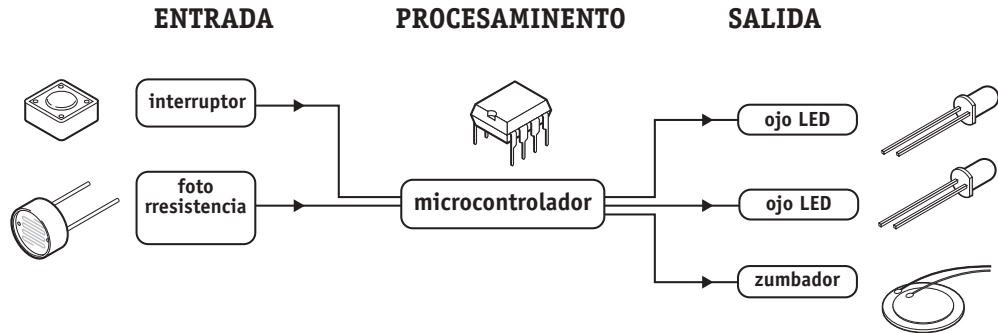
Para imprimir un programa/organigrama:

1. Haga clic en **Archivo- Imprimir...** (ó en el respectivo icono de la barra de herramientas)
2. Si desea que a cada línea del programa se le asigne un número, asegúrese de marcar la casilla “Imprimir números de línea”.
3. Haga clic en <OK>

SECCIÓN 5 – EL PCB DE LA CIBER-MASCOTA

El proyecto de ciber-mascota utiliza un microcontrolador PICAXE-08 con dos LEDs que simulan los “ojos” de las mascota y un zumbador que simula la “voz” de la mascota. El proyecto también utiliza un interruptor el cual permite a la mascota responder al “tacto”, y una fotorresistencia que permite a la mascota detectar si su alrededor esta claro u oscuro.

A continuación se muestra el diagrama de bloques electrónico del proyecto.



salida – el pin0 y el pin4 están conectados a los LEDs

salida – el pin2 está conectado al zumbador electrónico

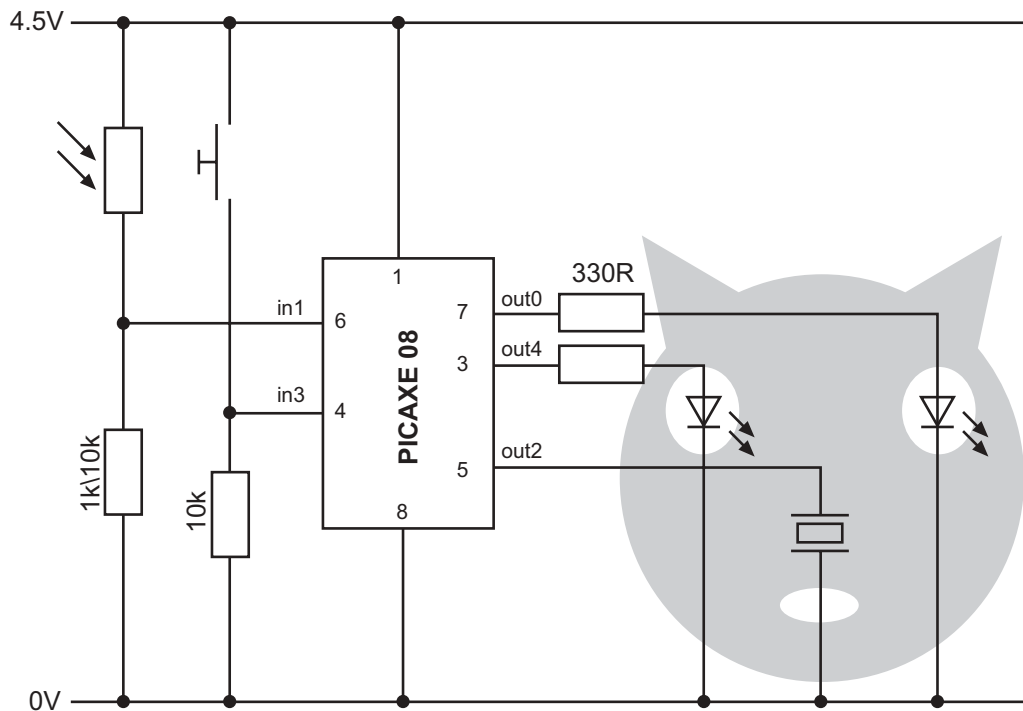
entrada - el pin1 está conectado a la fotorresistencia

entrada – el pin 3 está conectado al interruptor de botón de presión

¡Recuerde no confundir el número de pata del chip con el número de pin de entrada/salida!


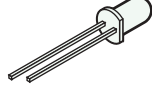
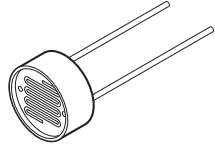

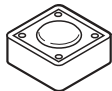

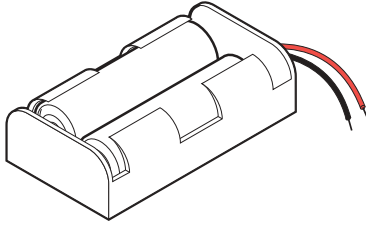
Diagrama del Circuito

A continuación se muestra el diagrama de circuito para el proyecto de la ciber-mascota :



CONSTRUYENDO EL PCB DE LA CIBER-MASCOTA

Para la construcción de este proyecto necesitará:

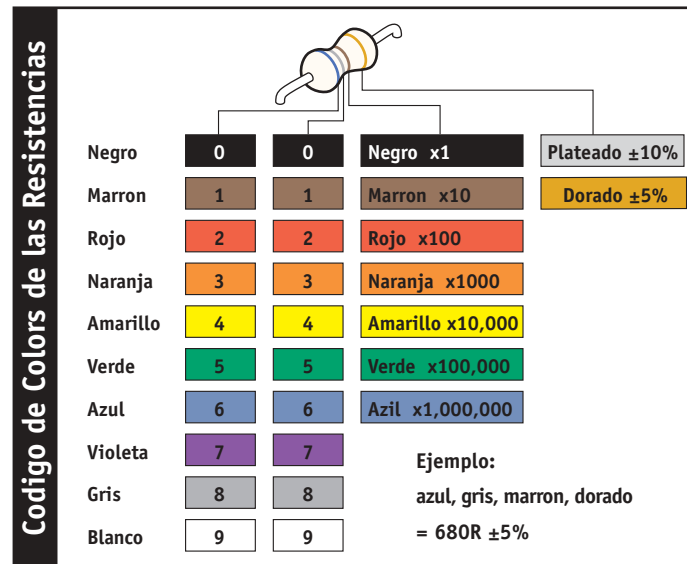
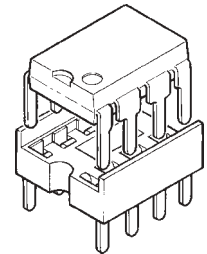
R1 y R2	resistencias de 10k (marrón negro naranja dorado)	
R3	resistencia de 22k (rojo rojo naranja dorado)	
R4 y R5	resistencias de 330R (naranja naranja marrón dorado)	
R6	resistencia de 1k /10k (marrón negro rojo/marrón dorado)	
LED1 y LED2	LEDs rojos de 5 mm	
PZ	zumbador electrónico	
LDR	fotorresistencia miniatura	
SW1	interruptor miniatura de 6 mm	
IC1	conector de 8 pines para circuito integrado	
PX	microcontrolador PICAXE-08	
CT1	conector de descarga PICAXE de 3.5 mm	
BT1	clip de la batería	
BT1	caja de baterías de 4.5 V (3 x AA)	
PCB	tablero de circuito impreso	

cables (en caso que esté conectando la fotorresistencia y los LEDs mediante cables)

Herramientas:
soldador y soldadura
alicate

Codigo de Colours de las Resistencias	Negro	0	0	Negro x1	Plateado ±10%
	Marron	1	1	Marron x10	Dorado ±5%
	Rojo	2	2	Rojo x100	
	Naranja	3	3	Naranja x1000	
	Amarillo	4	4	Amarillo x10,000	
	Verde	5	5	Verde x100,000	
	Azul	6	6	Azil x1,000,000	
	Violeta	7	7		
	Gris	8	8		
	Blanco	9	9		

Ejemplo:
azul, gris, marron, dorado
= 680R ±5%

Soldando componentes sobre el PCB

El PCB está fabricado especialmente con una película resistente a la soldadura para hacer el proceso de soldadura más sencillo. Esta película es la cubierta verde que cubre las pistas de manera que la soldadura no se pegue a las mismas. Para una construcción correcta el PCB se debe ensamblar y soldar muy cuidadosamente.

Al soldar asegúrese que la punta del soldador este caliente y limpia. Para verificar si está lo suficientemente caliente, trate de derretir un trozo de soldadura sobre la punta. La soldadura debe derretirse casi instantáneamente. Luego limpie la soldadura pasando la punta del soldador por una esponja húmeda.

Recuerde que la soldadura sólo se pega a superficies calientes. Por lo tanto nunca derrita la soldadura sobre la punta y luego trate de tirar la misma sobre la unión a soldar – esto no funcionará ya que la unión estará fría y la soldadura no se pegará.

Para soldar correctamente debe sostener en una mano el soldador y en la otra la soldadura. Por lo tanto, asegúrese que el tablero esté fijo a la mesa de manera que no se mueva (utilice una prensa ó pida alguien que lo sujete).

Pasos para soldar:

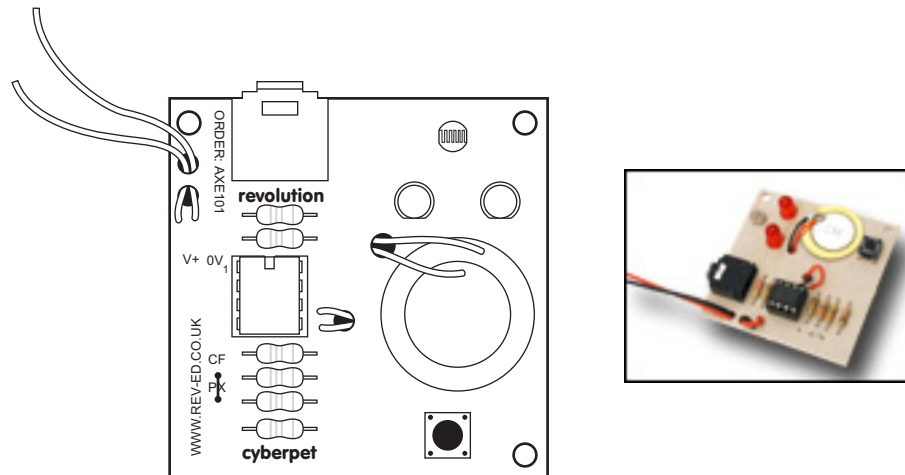
- 1) Limpie la punta de la soldadura sobre la esponja húmeda.
- 2) Presione la punta del soldador contra el terminal sobre el PCB y la pata del componente a soldar. Cuente hasta 3 para darle tiempo de calentarse a la unión.
- 3) Mantenga el soldador en posición y presione la soldadura contra la unión. Permita que se derrita suficiente soldadura para cubrir toda la unión.
- 4) Retire primero la soldadura y luego el soldador.
- 5) Permita que la soldadura se enfríe durante 5 segundos antes de mover el tablero.

¡Trucos!

- 1) Comience siempre con los componentes más pequeños, como por ejemplo resistencias. Luego continúe con componentes más grandes como el conector del circuito integrado y termine con componentes altos tales como condensadores y transistores. No intente fijar todos los componentes a la vez, fije solamente dos o tres a la vez.
- 2) Asegúrese siempre que los componentes estén bien montados (planos) sobre el tablero antes de soldarlos. Al utilizar componentes de patas largas tales como resistencias y LEDs, doble las patas de manera que el componente quede firmemente en posición antes de soldarlo.
- 3) Asegúrese que el conector estéreo de descarga PICAXE quede firme y acomodado en posición antes de soldarlo.
- 4) Asegúrese que aquellos componentes que sólo operan en un solo sentido (LEDs, diodos, transistores y condensadores) estén correctamente alineados antes de soldarlos (vea las marcas sobre el PCB).
- 5) Los alambres del zumbador son muy delgados. Tenga cuidado de no sobrecalearlos o de lo contrario puede que se derritan.
- 6) Haga pasar siempre los cables de la caja de baterías por arriba y abajo de los agujeros antes de soldarlos. Esto ayuda a hacer una unión más fuerte la cual es mucho menos propensa a soltarse.

La manera adecuada de construir su Ciber-Mascota PCB depende de la forma y tamaño de su diseño. Usted puede decidir soldar todos los componentes directamente sobre el tablero, ó puede conectar algunos de los componentes (Por ejemplo LEDs, la fotorresistencia y el interruptor) con tramos de cable de manera que pueda instalarlos cómodamente dentro de su mascota.

Estas instrucciones asumen que usted está soldando todos los componentes directamente sobre el tablero. Si utiliza cables las instrucciones son idénticas, sin embargo debe ser muy cuidadoso de colocar los cables para los LEDs en la dirección correcta.



- 1) Coloque en posición la resistencia de 1k/10k (marrón negro rojo/marrón dorado) y las dos resistencias de 10k (marrón negro marrón dorado). Dóbleles las patas para fijarlas en su posición y suelde.
- 2) Coloque en posición la resistencia de 22k (rojo rojo naranja dorado) y las dos resistencias de 330R (naranja naranja marrón dorado). Dóbleles las patas para fijarlas en su posición y suelde.
- 3) Utilizando alguno de los alambres cortados de las patas de las resistencias, haga un puente sobre las letras PX marcadas a un lado de las resistencias de 330R y luego suelde. (Ignore el agujero sobre los agujeros marcados CF)
- 4) Coloque y presione en su posición el conector estéreo de descarga PICAXE sobre el tablero y asegúrese que quede fijo (Debe escuchar un “clic”) de manera que quede nivelado sobre el tablero. Suelde los cinco contactos cuadrados metálicos (los cinco soportes plásticos redondos no tienen soldarse). No se preocupe si la soldadura une dos ó mas contactos en cualquiera de los dos lados del conector ya que supuestamente estos deben estar unidos de todas maneras.
- 5) Coloque y presione en posición el conector IC. Asegúrese que la muesca de uno de los extremos apunte hacia el conector. Dóblele las patas para fijarlo en posición y suelde.
- 6) Suelde la fotorresistencia y los dos LEDs en sus respectivas posiciones. Asegúrese que el lado plano de uno de los lados del LED esté alineado con el lado plano marcado en el PCB.
- 7) Suelde en su posición el interruptor (note que sólo encaja en un sentido). Si está utilizando cables, suelde uno de los cables en uno de los dos agujeros inferiores y el otro cable en uno de los dos agujeros superiores.
- 8) Pase los cables de la batería hacia abajo a través del agujero grande cerca de las letras AXE y luego páselos hacia arriba a través del agujero grande cerca del número 101. Luego suelde el cable negro en el agujero marcado 0V y el cable rojo en el agujero marcado V+.



- 9) Con una cinta adhesiva de doble contacto pegue el zumbador (por el lado de bronce) al PCB. Pase los cables del mismo hacia abajo por el agujero bajo LED1 y luego hacia arriba a través del agujero marcado PZ. Suelde el cable rojo en el agujero inferior y el cable negro en el agujero superior. No importa si la junta soldada del cable rojo se une con el pin5 del conector IC ya que estas se supone que deben unirse de todas maneras. Sin embargo, el cable negro NO debe tocar el pin6 del conector IC.
- 10) Revise cuidadosamente el tablero para comprobar que todas las uniones soldadas están bien hechas y que no hay ningún puente de soldadura creado accidentalmente.
- 11) Inserte el microcontrolador en el conector, asegurándose que el pin1 esté de frente al conector estéreo.

Probando su circuito

Paso 1 – Verifique las uniones soldadas.

Verifique que todas las uniones estén conectadas tanto al terminal como al cable, y que el cable este sujeto firmemente. También verifique que la soldadura no haga accidentalmente puentes entre terminales adyacentes. Esto es mucho más probable en los LEDs, la fotorresistencia y el zumbador. En el conector estéreo los terminales cuadrados a cada lado pueden unirse sin ninguna consecuencia ya que de todas formas están unidos por una pista en el tablero. Sin embargo estos no deben unirse al agujero redondo central.

Paso 2 – Verifique los componentes.

- 1) Verifique que el cable negro de la batería esté en el agujero marcado 0V y que el cable rojo esté en el agujero marcado V+.
- 2) Verifique que el chip PICAXE-08 esté insertado correctamente en el conector, con la muesca (que muestra el pin1) apuntando hacia el conector estéreo.
- 3) Verifique el lado plano del LED esté conectado al agujero correcto del PCB.
- 4) Asegúrese de no haber olvidado unir mediante un alambre los agujeros marcados PX en el extremo inferior izquierdo del tablero.
- 5) Asegúrese de pegar el lado de bronce del zumbador al tablero con cinta adhesiva de doble contacto.
- 6) Verifique que el conector esté soldado correctamente, incluyendo el terminal cuadrado central el cual a menudo es olvidado por equivocación.

Paso 3 – Conecte la batería.

Verifique que las 3 pilas AA estén colocadas correctamente dentro de la caja de baterías. Conecte la caja de baterías al cable de baterías y ponga su dedo sobre el microcontrolador PICAXE. Si comienza a calentarse desconecte la batería inmediatamente ya que debe haber algún problema – lo más seguro es que el chip o los cables de la batería estén conectados en sentido inverso.

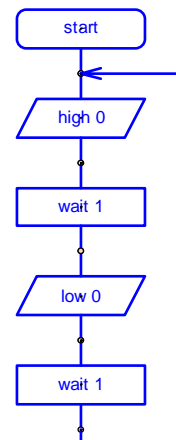
Paso 4 – Descargue un programa para probar el LED 0

Conecte el cable a su ordenador y al conector PICAXE en el PCB. Asegúrese que el enchufe del cable quede completamente dentro del conector del PCB.

Asegúrese que el software esté en el modo PICAXE-08 y que haya elegido el puerto serie correcto (para mayor información vea la sección 4 de este libretto).

Escriba y descargue el siguiente programa:

```
main:
    high 0
    wait 1
    low 0
    wait 1
    goto main
```



El LED debe titilar a medida que se descarga el programa. Al terminar la descarga el LED deberá encenderse y apagarse cada segundo. Si el LED no hace esto verifique que esté conectado correctamente y que las resistencias de 330R estén la posición correcta en el PCB.

Si el programa no se descarga verifique que la resistencia de 22k, la de 10k y el conector IC estén soldados correctamente. Utilice un voltímetro para verificar si hay 4.5V entre las patas superiores (1 y 8) del microcontrolador. Verifique que el cable esté firmemente conectado al conector y que dentro del software se haya elegido el puerto serie correcto.

Paso 5 – Pruebe el otro LED

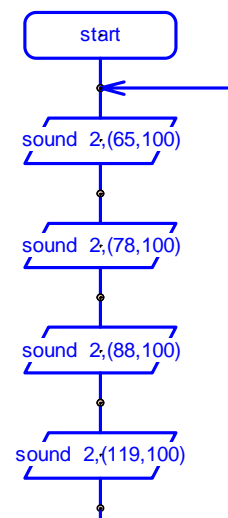
Repita el programa del paso 4, pero utilice *high 4* y *low 4* en vez de *high 0* y *low 0*. Esto probará al otro LED.

Paso 6 – Pruebe el zumbador

Escriba y descargue el siguiente programa:

```
main:
    sound 2,(65,100)
    sound 2,(78,100)
    sound 2,(88,100)
    sound 2,(119,100)
    goto main
```

El zumbador debe emitir 4 sonidos diferentes. Si no hace esto asegúrese que los alambres estén soldados correctamente, que el lado de bronce esté firmemente pegado al PCB con una cita adhesiva de doble contacto (no trabajará si esta flojo) y que los terminales sobre las letras PX estén debidamente unidos mediante un alambre soldado.



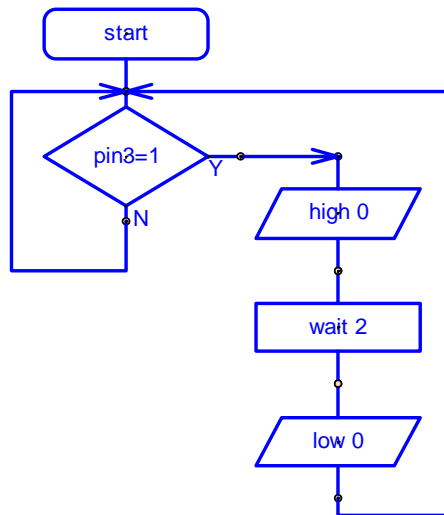
Paso 7 - Pruebe el Interruptor

Escriba y descargue el siguiente programa:

```

main:                                     ` hacer una etiqueta llamada "main"
    if input3 is on then flash ` salta a flash si la entrada está encendida
    goto main                          ` sino regresar a inicio

flash:                                    ` hacer una etiqueta llamada "flash"
    high 0                               ` encender salida 0
    wait 2                               ` esperar 2 segundos
    low 0                                 ` apagar salida 0
    goto main                            ` regresar al inicio
    
```



El LED de la salida 0 deberá encenderse cada vez que se presione el interruptor. Si no lo hace verifique que el interruptor y que las resistencias de 10k estén soldadas correctamente.

Paso 8 – Pruebe la fotorresistencia

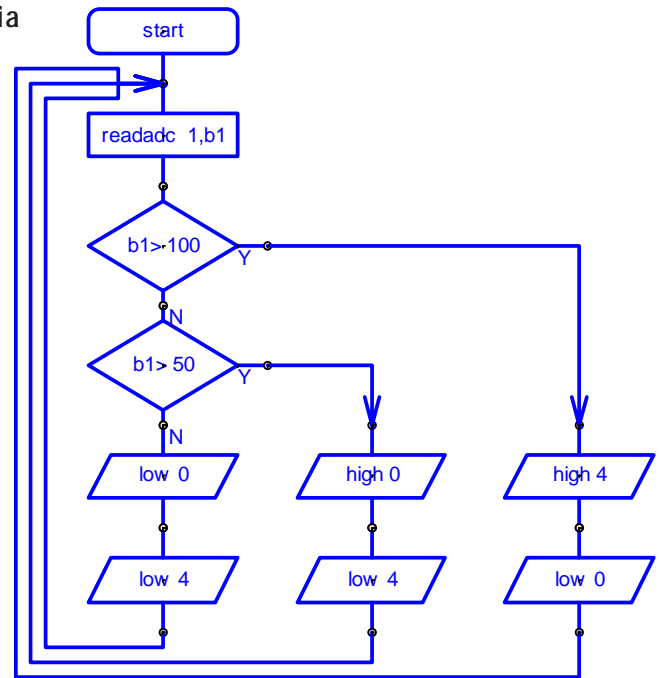
Escriba y descargue el siguiente programa:

```

main:
  readadc 1,b1
  if b1 > 100 then do4
  if b1 > 50 then do0
  low 0
  low 4
  goto main

do4:
  high 4
  low 0
  goto main

do0:
  high 0
  low 4
  goto main
    
```



Ambos LEDs deberán encenderse en patrones distintos cuando usted cubre y descubre la fotorresistencia con su mano (de manera que incidan sobre la fotorresistencia distintos niveles de luz). Si esto no ocurre verifique que la fotorresistencia y la resistencia de 1k estén soldadas correctamente.

¡Si ha ejecutado todas estas pruebas correctamente lo felicitamos ya que ha construido y ensamblado correctamente su Ciber-mascota! ¡Ahora es el momento de desarrollar y probar sus propios programas para darle personalidad a su mascota!

SECCIÓN 6 – IDEAS DE PROGRAMACIÓN

Ahora que ha ensamblado y probado su Ciber-mascota, es el momento de darle una “personalidad” desarrollando sus propios programas. Estos programas pueden hacer que la mascota reaccione de diversas maneras al toque de los interruptores de presión y a los niveles de luz.

En las siguientes páginas se incluyen dos ejemplos de programas. Estos están diseñados para darle un punto de partida para la creación de su programa. Usted puede modificarlos o comenzar a hacer un programa completamente nuevo si así lo prefiere.

¡Sea creativo!

¡Su Ciber-mascota es su creación por lo tanto dele una gran personalidad!

Programa 1 Explicación

Este programa tiene un bucle principal el cual enciende y apaga los ojos LED, y también verifica el estado del sensor de luz y del interruptor de presión. Cuando se presiona el botón del interruptor el zumbador emite un sonido. Tome en cuenta que debe mantener el interruptor presionado hasta que escuche el sonido – presionar brevemente el interruptor no funcionará.

Si la fotorresistencia se cubre la mascota se “irá a dormir” hasta que el nivel de luz suba nuevamente.

Programa 2 Explicación

Este programa es mucho más avanzado. Tiene un bucle principal el cual enciende y apaga los ojos LED atenuándolos; también verifica el estado del sensor de luz y del interruptor. Cuando se presiona el botón del interruptor, el zumbador emite un sonido mediante el uso del comando *sound*. Si se presiona el botón tres veces (contadas por una variable llamada *b3*), la mascota pone una melodía. Tome en cuenta que debe mantener el interruptor presionado hasta que escuche el sonido – presionar brevemente el interruptor no funcionará.

Si la fotorresistencia se cubre la mascota se “irá a dormir”.

Este programa utiliza una técnica llamada PWM (Pulse Width Modulation - Modulación de la anchura del impulso) para permitir la atenuación de los ojos LED, en vez de simplemente apagarlos o encenderlos inmediatamente utilizando los comandos *high* y *low*. La PWM funciona encendiendo y apagando la salida muy rápidamente, más rápido que lo que el ojo humano puede ver. Variando el tiempo que la salida está encendida (llamado *mark*) con respecto al tiempo que la misma está apagada (llamado *space*), el brillo del LED puede alterarse.

Program 1

```

\***** bucle principal ****
\este bucle hace parpadear las luces
\y verifica el estado del sensor de luz y del interruptor

main:
\Encender ambos LEDs y leer valor de luz (b1)
    high 4
    high 0
    readadc 1,b1

\si el valor de luz está debajo de 80 ir a dormir
    if b1 < 80 then bed

\si el interruptor es presionado emitir sonido
    if pin3 = 1 then purr

\hacer una pausa
    pause 500

\Apagar ambos LEDs y verificar estado del sensor nuevamente
    low 4
    low 0
    readadc 1,b1

\ si el valor de luz está debajo de 80 ir a dormir
    if b1 < 40 then bed

\si el interruptor es presionado emitir sonido
    if pin3 = 1 then purr
    pause 500
    goto main

\ ***** emitir sonido *****
purr:
    sound 2,(120,50,80,50,120,50)
    pause 200
    goto main

\ ***** rutina de ir a dormir cuando está oscuro *****
\Si está oscuro apagar ambos LEDs y esperar hasta que haya
\ luz nuevamente
bed:
    low 0
    low 4
    readadc 1,b1
    if b1 > 80 then main
    goto bed

```

Programa 2

```

` ***** bucle principal *****
main:

` hacer que el LED 0 brille progresivamente en 10 pasos
  for b2 = 1 to 10
    pwm 0, b2, 20
  next b2
  high 0

` y luego hacer que el LED 4 brille progresivamente en 10 pasos
  for b2 = 1 to 10
    pwm 4, b2, 20
  next b2
  high 4

`ya que ambos LEDs están encendidos leer el valor de luz
  readadc 1,b1

`si el valor de luz está por debajo de 80 ir a dormir
  if b1 < 80 then bed

`si el interruptor es presionado sumar 1 al valor de "pet"
  if pin3 = 1 then addpet

`primero atenuar el LED 0 en 10 pasos
  for b2 = 10 to 1 step -1
    pwm 0, b2, 20
  next b2
  low 0

`luego atenuar el LED 4 en 10 pasos
  for b2 = 10 to 1 step -1
    pwm 4, b2, 20
  next b2
  low 4

` ya que ambos LEDs están apagados leer el valor de luz
  readadc 1,b1

`si el valor de luz está debajo de 80 ir a dormir
  if b1< 80 then bed

`si el interruptor es presionado sumar 1 al valor de "pet"

  if pin3 = 1 then addpet
  goto main

` ***** fin de bucle principal *****

```

(continuación del programa en la siguiente página)

```

` ***** sumar 1 al valor de "pet" *****
addpet:
    b3 = b3 + 1
    sound 2,(120,50)
    pause 200

`verificar si el valor de "pet" es 3
`si es menor de 3 regresar al bucle
    if b3 < 3 then loop

`el valor de "pet" es 3 por lo tanto poner una melodía y reiniciar el
contador "pet"
    high 0
    high 4
    sound 2,(80,50,100,50,120,50,100,50,120,50)
    let b3 = 0
    goto main

` ***** rutina de ir a dormir cuando está oscuro *****
`Si está oscuro apagar ambos LEDs y esperar hasta que haya
` luz nuevamente

bed:
    low 0
    low 4
    readadc 1,b1
    if b1> 40 then main
    goto bed

```

AGRADECIMIENTOS

El desarrollo de este proyecto fue financiado por la "UK Offshore Oil and Gas Industry".
www.oilandgas.org.uk/education/

(c) Revolution Education Ltd 2002
www.rev-ed.co.uk

Todos los derechos reservados.
 Puede ser fotocopiado para uso no-comercial y educacional en salones de clases de escuelas y colegios únicamente.

Furby es una marca registrada de Tigre Electronics Ltd
 PICAXE es una marca registrada de Revolution Education Ltd

