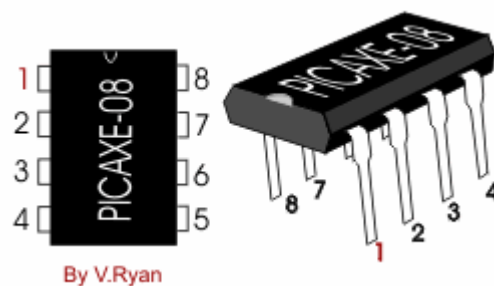


CUADERNO DE ACTIVIDADES

picaxe



Relatores:

Marco A. López Grande/José Luis Fernández Souto

Práctica 1:

Encendido y apagado de forma intermitente de un led conectado al pin 7 del PICAXE-08 (Salida: Pin0)

Código:

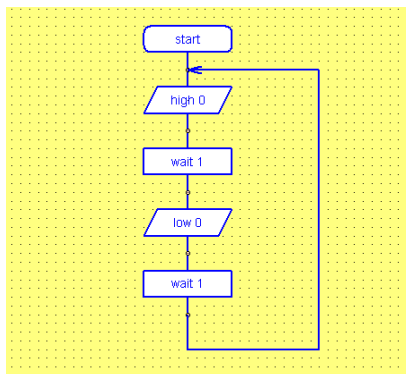
```

*****
***** PRUEBA ENCENDIDO LED D1 (pin 7, out 0) *****
*****
  
```

inicio:

high 0	'enciende LED en pin 7
wait 1	'retardo 1 segundo
low 0	'apaga LED en pin 7
wait 1	
goto inicio	'salto a inicio

Flujograma:



Práctica 2:

Encendido y apagado de forma intermitente de un led conectado al pin 3 del PICAXE-08 (Salida: Pin4)

Código:

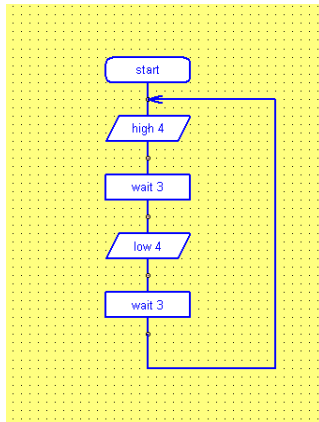
```

*****
***** PRUEBA ENCENDIDO LED D2 (pin 3, out 4) *****
*****
  
```

inicio:

high 4	'enciende LED en pin 3
wait 3	'retardo 1 segundo
low 4	'apaga LED en pin 3
wait 3	
goto inicio	'salto a inicio

Flujograma:



Práctica 3:

Encendido y apagado alternativamente de dos leds conectados al pin 3 y pin 7 del PICAXE-08 (Salida: Pin4 y Pin0).

Código:

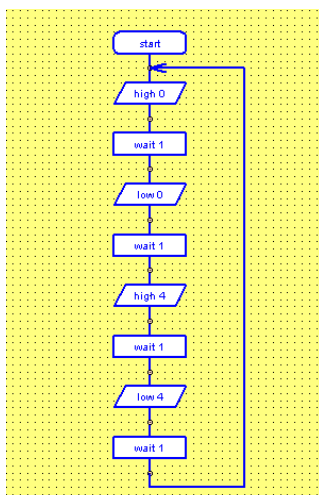
```

'*****
'***** Leds Intermitentes alternativamente (pin 7, out 0)y (pin 3, Out 4) *****
'*****
  
```

inicio:

high 0	'enciende LED en pin 7
wait 1	'retardo 1 segundo
low 0	'apaga LED en pin 7
wait 1	
high 4	'enciende LED en pin 3
wait 1	
low 4	'apaga led en pin 3
wait 1	
goto inicio	'salto a inicio

Flujograma:



Práctica 4:

Generación de 3 tonos diferentes a través del altavoz piezoeléctrico conectado al pin 5 del PICAXE-08 (Salida: Pin2).

Código:

```

*****
***** PRUEBA ALTAVOZ *****
*****

```

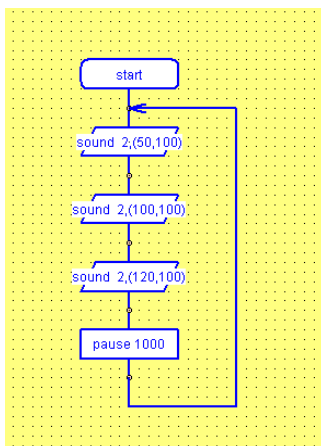
inicio:

```

sound 2, (50, 100)      'sonido en altavoz piezo del pin 5, out 2 tono ((0..127), duración ms
sound 2, (100, 100) 'otro tono
sound 2, (120, 100) 'otro tono
pause 1000              'pausa de 1000ms = 1 seg
goto inicio

```

Flujograma:



Práctica 5:

Si se activa el interruptor conectado al pin 4 (Entrada: Pin3) enciende el led conectado al pin 7 del PICAXE-08 (Salida: Pin0).

Código:

```

*****
***** PRUEBA INTERRUPTOR *****
*****

```

inicio:

```

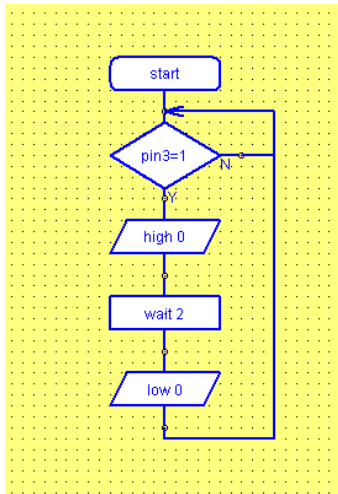
if input3 is on then led      'si la entrada 3 es "1" salta a subrutina led sino a inicio
goto inicio                  'salto a inicio

led:

high 0                       'enciende LED pin 7
wait 2                       'encendido durante un retardo de 2 segundos
low 0                        'apagado LED pin 7
goto inicio

```

Flujograma:



Práctica 6:

Tenemos conectada una LDR en el pin 6 (Entrada: Pin1) del PICAXE-08. Visualizaremos el valor que nos da el convertidor Analógico/Digital (DAC) a través del Debug de la aplicación (presenta el valor de las variables b0..b13, tanto en valor decimal como digital).

Código:

```

'*****
'***** Chequeo de umbral de luz *****
'*****
'LDR en pin 6, input1
  
```

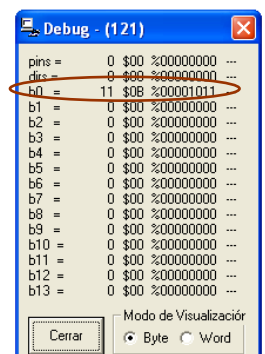
```

'*** Bucle principal ***
  
```

inicio:

readadc 1,b0	'lee la señal analógica de LDR y carga en variable b0
debug b0	'transmite el valor b0 a la pantalla del PC
pause 100	'pausa
goto inicio	'saltar a inicio

Resultado:



Al cargar el programa al PICAXE-08 se abrirá el depurador (Debug) mostrando el valor de las variables b0..b13. Deberemos fijarnos en el valor que toma la variable b0 que es donde se almacena el valor generado por el convertidor DAC, anotaremos el valor justo en el momento que la intensidad luminosa que nos interesa detectar se presente. Cada muestreo será indicado visualmente a través del Led conectado al pin 7 del PICAXE-08 (Salida: Pin0).

Práctica 7:

Tenemos conectada una LDR en el pin 6 (Entrada: Pin1) del PICAXE-08. Si la intensidad de luz es alta (>120) se iluminará el Led 1, por el contrario, si la luz que incide sobre la LDR es baja (<70) se iluminará el Led 2.

Código:

```

'*****
'***** Detector luz máx y mín *****
'*****

'LDR wn pin 6, input1
'LED D1 en pin7, out0
'LED D2 en pin3, out4

'**** Bucle principal ****

inicio:
    readadc 1,b0
    if b0 > 120 then maximo
    if b0 < 70 then minimo
    low 0
    low 4
    goto inicio

'leer señal analógica de LDR y cargar en variable b0
'si valor de variable es mayor de 120 ir a subrutina maximo
'si valor de variable es menor de 70 ir a subrutina minimo
'apagar LED D1
'apagar LED D2
'ir a inicio

'**** Acción límite máx ****

maximo:

    high 0
    low 4
    goto inicio

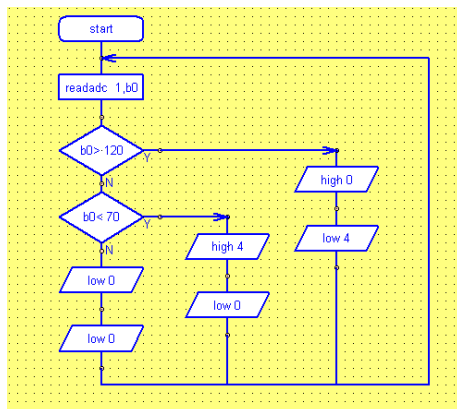
'encender LED1
'apagar LED2

'**** Acción límite mín ****

minimo:

    high 4
    low 0
    goto inicio
    
```

Flujograma:



Práctica 8:

Tenemos conectada una LDR en el pin 6 (Entrada: Pin1) del PICAXE-08. Deberemos generar un haz de luz unidireccional que deberá incidir sobre la superficie sensible de la LDR. Si la luz incide no se dispara la alarma (Led y piezo activados), por el contrario si se corta el haz de luz (atravesado por un objeto móvil) esta se disparará activando una señal acústica y luminosa (altavoz piezoeléctrico y Led).

Código:

```
*****
***** BARRERA LUMINOSA. Alarma *****
*****
```

```
'LDR wn pin 6, input1
'LED D1 en pin7, out0
'ALTAVOZ piezoeléctrico en pin5, out2
```

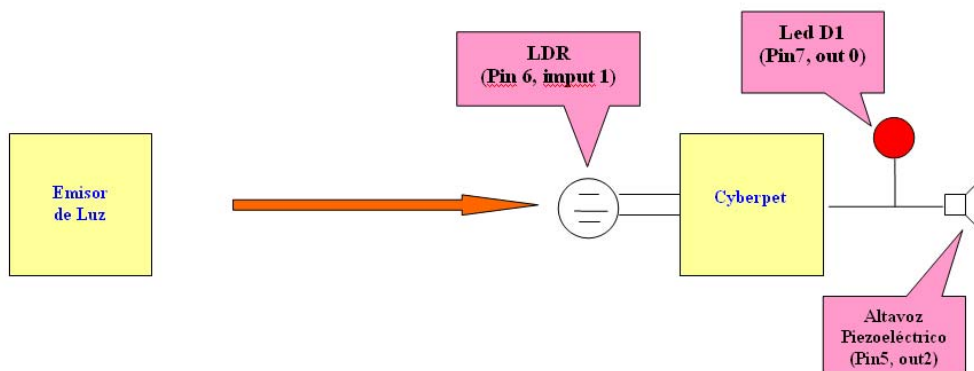
```
symbol light=b1
```

```
**** Bucle principal ****
'Este bucle detecta la presencia de intrusos y corta la barrera de luz
```

```
inicio:
    low 0
    readadc 1,light
    if light < 80 then alarma
    goto inicio
```

```
**** Activar alarma ****
'Esta rutina enciende el LED D1 y hace sonar el altavoz mientras exista corte de haz de luz
```

```
alarma:
    high 0
    sound 2,(50,300)
    readadc 1, light
    if light > 80 then inicio
    goto alarma
```



Práctica 9:

Control total de Cyberpet. Analizar el código y descubrir sus funciones a través de la interpretación del mismo.

Código:

```
*****
***** CIBERMASCOTA. Programa1 *****
*****

'LDR wn pin 6, input1
'PULSADOR en pin4, input3
'LED D1 en pin7, out0
'LED D2 en pin3, out4
'ALTAVOZ piezoeléctrico en pin5, out2

symbol light=b1

**** Bucle principal ****
'Este bucle hace parpadear los leds y verifica el estado del sensor de luz y del pulsador

inicio:

'Encender ambos led y leer valor de luz (light=b1)

    high 4
    high 0
    readadc 1,light

'Si el valor de luz está por debajo de 80 ir a dormir

    if light < 80 then bed

'Si el pulsador 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,light
    goto inicio

**** Emitir sonido ****

purr:

    sound 2, (120,50,80,50,120,50)
    pause 200
    goto inicio

**** Rutina de ir a dormir cuando oscurece ****
'Si está oscuro apagar ambos leds y esperar hasta que haya luz nuevamente

bed:

    low 0
    low 4
    readadc 1, light
    if light > 80 then inicio
    goto bed
```

Práctica 10:

Escribe un programa para controlar una luz en función de un pulsador. Si el pulsador está abierto la luz estará apagada, si el pulsador está cerrado la luz estará encendida. Al soltar el pulsador la luz permanecerá encendida durante 3 segundos más.

Solución:

```
*****
*****Práctica 10*****
*****
```

```
'PULSADOR en pin4, input3
'LED D2 en pin3, out4
```

inicio:

```
if input3 is on then encender
low 4
goto inicio
```

encender:

```
high 4
wait 3
goto inicio
```

Práctica 11:

Escribe un programa que simula el funcionamiento de un timbre. Mientras accionemos el pulsador sonará un sonido de dos tonos diferentes. Al soltar el pulsador se detiene el sonido.

Solución:

```
*****
*****Práctica 10*****
*****
```

```
'PULSADOR en pin4, input3
'ALTAVOZ piezoeléctrico en pin5, out2
```

inicio:

```
if input3 is on then sonido
low 2
goto inicio
```

sonido:

```
sound 2,(115,1,116,1,117,1,118,1,119,1,118,1,117,1,116,1,115,1)
goto inicio
```

Práctica 12:

Escribe un programa que permita controlar un faro que debe emitir dos grupos de destellos 3 y 5 destellos separados 2 segundos entre sí.

Solución:

```
*****  
***** ACTIVIDAD 7. FARO *****  
*****
```

inicio:

```
let b1=3  
gosub destellos  
wait 2  
let b1=5  
gosub destellos  
wait 2
```

goto inicio

destellos:

```
for b0=1 to b1  
  high 4  
  wait 1  
  low 4  
  wait 1  
next b0
```

return

Práctica 13:

Escribe un programa para controlar la velocidad de parpadeo de un led en función del estado de un pulsador. Si el pulsador está accionado parpadea deprisa, en caso contrario parpadea despacio.

Solución:

```
*****  
***** ACTIVIDAD 8. VELOCIDAD PARPADEO LED *****  
*****
```

inicio:

```
if input3 is on then parpadeo_rapido  
high 4  
wait 1  
low 4  
wait 1  
goto inicio
```

parpadeo_rapido:

```
high 4  
pause 100  
low 4  
pause 100  
goto inicio
```

Práctica 14:

Avisador sonoro de falta de luz. Cuando el nivel luminoso de la habitación descienda de un mínimo el zumbador debe emitir un sonido de aviso.

Solución:

```
*****  
***** ACTIVIDAD 9. ALARMA FALTA DE LUZ *****  
*****
```

inicio:

```
readadc 1, b0  
if b0 < 11 then alarma  
goto inicio
```

alarma:

```
sound 2, (50, 100)  
pause 500  
goto inicio
```

Práctica 15:

Simulador alarma: Cuando el pulsador este abierto (final de carrera ventana o puerta) y no exista luz, deberá encenderse un led y sonar el zumbador.

Solución:

```
*****  
***** ACTIVIDAD 10. SIMULADOR DE ALARMA *****  
*****
```

inicio:

```
readadc 1, b0  
if input3 is off and b0 < 11 then alarma_on  
gosub alarma_off  
goto inicio
```

alarma_on:

```
high 4  
sound 2, (50, 100)  
pause 100  
goto inicio
```

alarma_off:

```
low 4  
return
```

Práctica 16:

Escribe un programa que genere 5 tonos que se incrementen secuencialmente mediante el altavoz piezoeléctrico conectado al pin 5 del PICAXE-08.

Solución:

```
*****
***** Práctica 16 *****
*****
```

inicio:

```
for b1 = 25 to 125 step 25
    sound 2,(b1,100)
next b1
goto inicio
```

Práctica 17:

Queremos controlar un sistema “disuade-ladrones” (simulado por dos leds del cyberpet) que funcione según las siguientes especificaciones:

- Cuando comience a anochecer y el nivel de luz sea bajo, un motor (led del pin=0) se encenderá para subir la persiana del salón. Al cabo de tres segundos el motor se para (la persiana está totalmente subida).
- Cuando el nivel de luz sea un poco menor (totalmente de noche) se encienden las luces del salón (led pin4).
- Cuando amanezca la persiana subirá y la luz se apagará.

Solución:

```
*****
***** DISUADE LADRONES *****
*****
```

inicio:

```
low 4
readadc 1, b0
if b0>20 and b0<50 and b3= 0 then motor
if b0 < 20 then salon
if b0 > 100 and b3=1 then amanece
goto inicio
```

motor:

```
for b1 = 1 to 3
    high 0
    wait 1
    low 0
    if input3 is on then emergencia
next b1
```

;al asignar el valor 1 a la variable b3 sabemos que la persiana está subida

```
let b3=1
goto inicio
```

salon:

```
high 4
goto inicio
```

amanece:

```
;el parpadeo del led indica que el motor está bajando la persiana
  for b2=1 to 3
    high 0
    wait 1
    low 0
    wait 1
  next b2
  low 4
;al asignar el valor 0 a la variable b3 sabemos que la persiana está bajada
  let b3=0
  goto inicio
```

emergencia:

```
low 0
wait 5
goto inicio
```

Práctica 18:

Escribe el programa de control para un motor (led pin0) que funcione según las siguientes especificaciones:

- Si el pulsador está abierto el motor estará parado.
- Si el pulsador está cerrado Y el nivel de luz es bajo el motor funcionará ininterrumpidamente.
- Si el pulsador está cerrado Y el nivel de luz es alto el motor realizará 3 ciclos de encendido-apagado (2 segundos encendido y 1 apagado) y se detendrá.

Solución:

```
'*****
'***** CONTROL MOTOR *****
'*****
```

inicio:

```
  if input3 is off then motor_off
  readadc 1,b0
  if input3 is on and b0 = 0 then motor_on
  if input3 is on and b0 > 11 then motor_ciclo
motor_off:
  low 0
  goto inicio
motor_on:
  high 0
  goto inicio
motor_ciclo:
  for b1 = 1 to 3
    high 0
    wait 2
    low 0
    wait 1
  next b1
  goto inicio
```

Práctica 19:

Escribe el programa de control para un motor (led pin0) que funcione según las siguientes especificaciones:

- Si el pulsado está abierto el motor estará apagado.
- Si el pulsador está cerrado O el nivel de luz es bajo el motor funcionará ininterrumpidamente.

```
*****
***** CONTROL MOTOR 2 *****
*****
```

inicio:

```
readadc 1,b0
if b0 = 0 OR input3 is on then motor_on
if input3 is off then motor_off
goto inicio
```

motor_off:

```
low 0
goto inicio
```

motor_on:

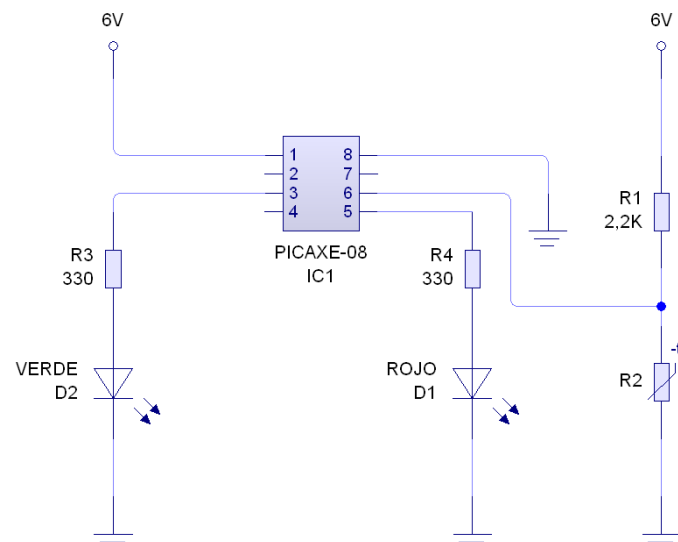
```
high 0
goto inicio
```

Práctica 20:

Diseña un termostato. Dibuja su esquema y el programa para su correcto funcionamiento.

Solución:

Esquema de conexionado:



Programa:

```

'*****
'***** SENSOR DE TEMPERATURA *****
'*****

```

```
symbol sensor=b1
```

```
**** Bucle principal ****
```

```
'Este bucle detecta la variación de temperatura
```

```
inicio:
```

```

readadc 1,sensor
debug b1
if sensor > 115 then temp_baja
if sensor < 115 then temp_alta
goto inicio

```

```
**** Activar alarma ****
```

```
'Esta rutina enciende el LED VERDE indicando temperatura por debajo del ajustado
```

```
temp_baja:
```

```

high 4          'enciende LED VERDE en patilla 3
low 2           'apaga LED ROJO en patilla 5
wait 1
goto inicio     'salto a inicio

```

```
'Esta rutina enciende el LED ROJO indicando temperatura por encima del ajustado
```

```
temp_alta:
```

```

high 2          'enciende LED ROJO en patilla 5
low 4           'apaga LED VERDE en patilla 3
wait 1
goto inicio     'salto a inicio

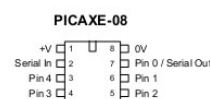
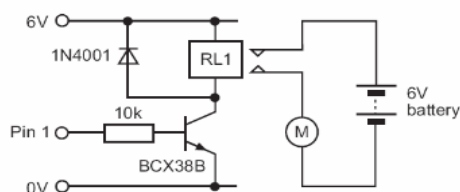
```

Práctica 21:

Control de un ventilador con PICAXE08.

Para realizar las siguientes prácticas emplearemos el picaxe 08 o 08M, una placa con pulsadores y una placa de proyectos para montar los componentes electrónicos auxiliares.

Propuesta A. El ventilador se encenderá y apagará automáticamente a intervalos de tiempo programados.



Una posibilidad para el programa de control seria:

```
input 1
symbol pulsa_on=input1

inicio:
    if pulsa_on is on then encender
    low 0
goto inicio
```

```
encender:
    high 0
    wait 3
    low 0
    wait 6
goto inicio
```

Propuesta B. Encendemos y apagamos el ventilador mediante dos pulsadores A y B. El A encenderá el ventilador de forma continua. El B lo encenderá y apagará en intervalos de tiempo de 3 segundos.

Una posibilidad para el programa de control seria:

```
input 1
input 2
symbol pulsa_A=input1
symbol pulsa_B=input2

inicio:
    if pulsa_A is on then encender
    if pulsa B is on then intervalos
    low 0
goto inicio
```

```
encender:
    high 0
goto inicio
```

```
intervalos:
    high 0
    wait 3
    low 0
    wait 3

goto inicio
```

Propuesta C. Sustituir los pulsadores por un DS18S20 que encienda automáticamente el ventilador a partir de una determinada temperatura. Para este ejemplo es necesario el uso del picaxe 08M.

```
symbol tempe=b1  
output 4
```

inicio:

```
readtemp 4,tempe  
if tempe>30 then encender  
if tempe<20 then apagar
```

goto inicio

encender:

```
high 4
```

goto inicio:

apagar:

```
low 4
```

goto inicio

Montaje Cyberpet

Diagrama de Bloque

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

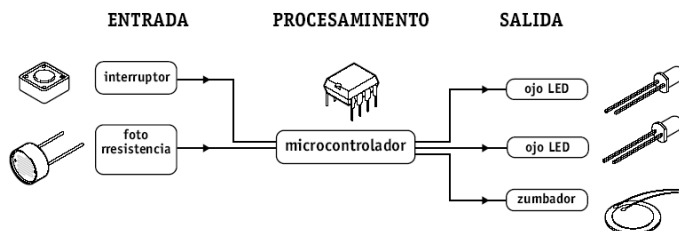
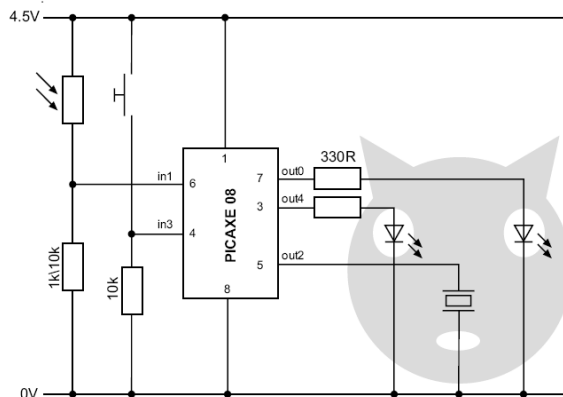


Diagrama del Circuito

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 Colores de las Resistencias			
Negro	0	0	Negro x1
Marrón	1	1	Marrón x10
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	Azul x1,000,000
Violeta	7	7	
Gris	8	8	
Blanco	9	9	

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