

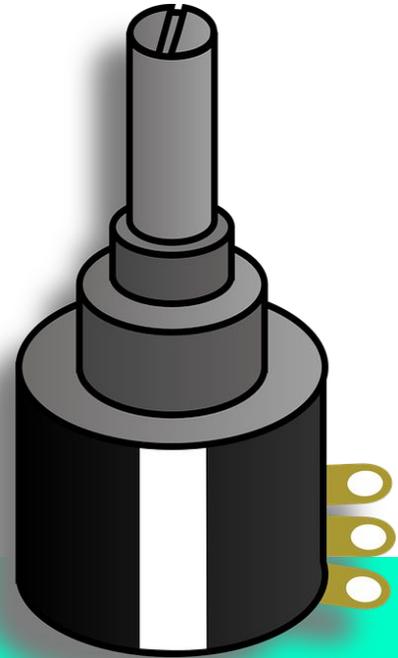


# VISUALINO

(APUNTES Y PROYECTOS. PARTE II)

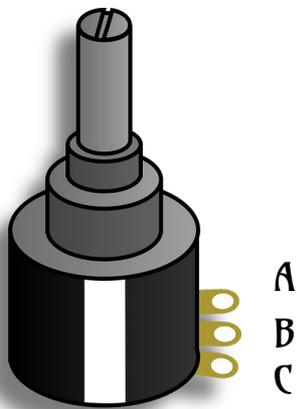
por Aurelio Gallardo Rodríguez BY – SA – NC

# POTENCIÓMETRO

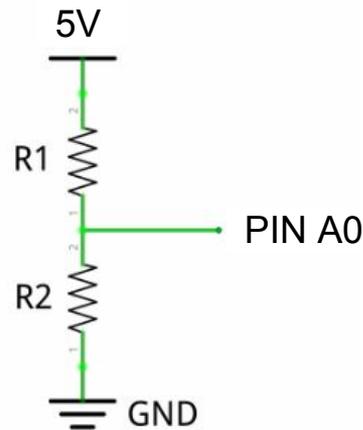


# ¿QUÉ ES UN POTENCIÓMETRO? CONEXIÓN A ARDUINO

Un potenciómetro es una **resistencia variable**. Si por ejemplo, su valor es de  $100\text{K}\Omega$ , quiere decir que mediante un elemento móvil (como un deslizador o vástago rotatorio), puedo poner entre uno de sus extremos y el pin central un valor menor que  $100\text{K}\Omega$ . Por ejemplo, supongamos que entre las patas A y B tengo  $10\text{K}\Omega$ . Entre las patas B y C tendré  $90\text{K}\Omega$ .



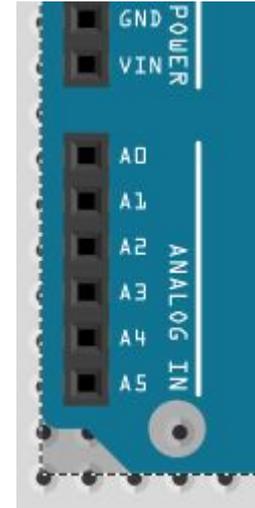
Con los potenciómetros se diseñan fácilmente **divisores de tensiones**. No entraremos en analizarlos con profundidad. Simplemente decir que si conecto en A la tensión del ARDUINO (5V) y en C el pin de tierra (GND) en la pata B central obtendré valores de tensión entre 0 y 5 Volt. simplemente accionando el elemento móvil.



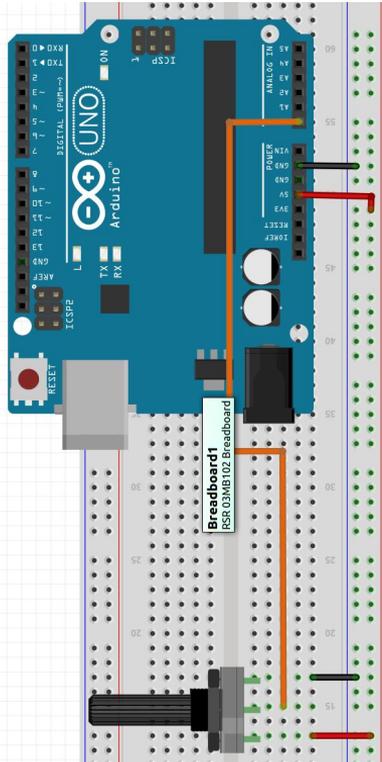
# ¿CÓMO LEE ARDUINO LAS ENTRADAS VARIABLES?

Cuando por ejemplo diseñamos los botones, el pin que recibía la señal de ARDUINO podía estar en **ALTO** o **BAJO**, es decir activado (1) o desactivado (0). El botón definía uno de los pines de ARDUINO como **entrada digital**. Los leds también se encendían en ALTO (1) o se apagaban en BAJO (0); eran **salidas digitales**.

Pero ¿qué ocurre ahora con el potenciómetro? En principio podremos obtener con él valores intermedios. No sólo tengo 0 y/o 5 Volts. sino todos los valores entre esos números: 1V, 2V, 3.4V, etc. Claro, ahora no puedo usar entradas digitales, porque las entradas digitales sólo se definen para dos valores y tengo muchos más. Ahora necesito otro tipo de entradas, las **entradas analógicas** en ARDUINO: **A0, A1, A2, A3, A4 y A5**. Estas entradas se usan, en general, para leer potenciómetros y señales de sensores (algunos).



# ¿CÓMO LEE ARDUINO LAS ENTRADAS VARIABLES?



Aparentemente no ocurre nada. Pero si en el programa pulsas el botón MONITOR y giras el potenciómetro, verás que obtienes una lista de números entre 0 y 1023.

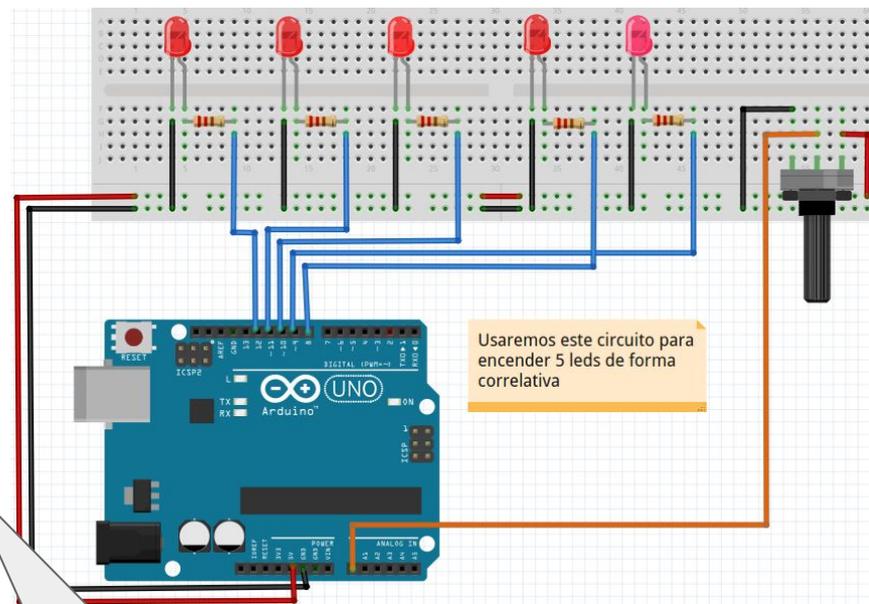
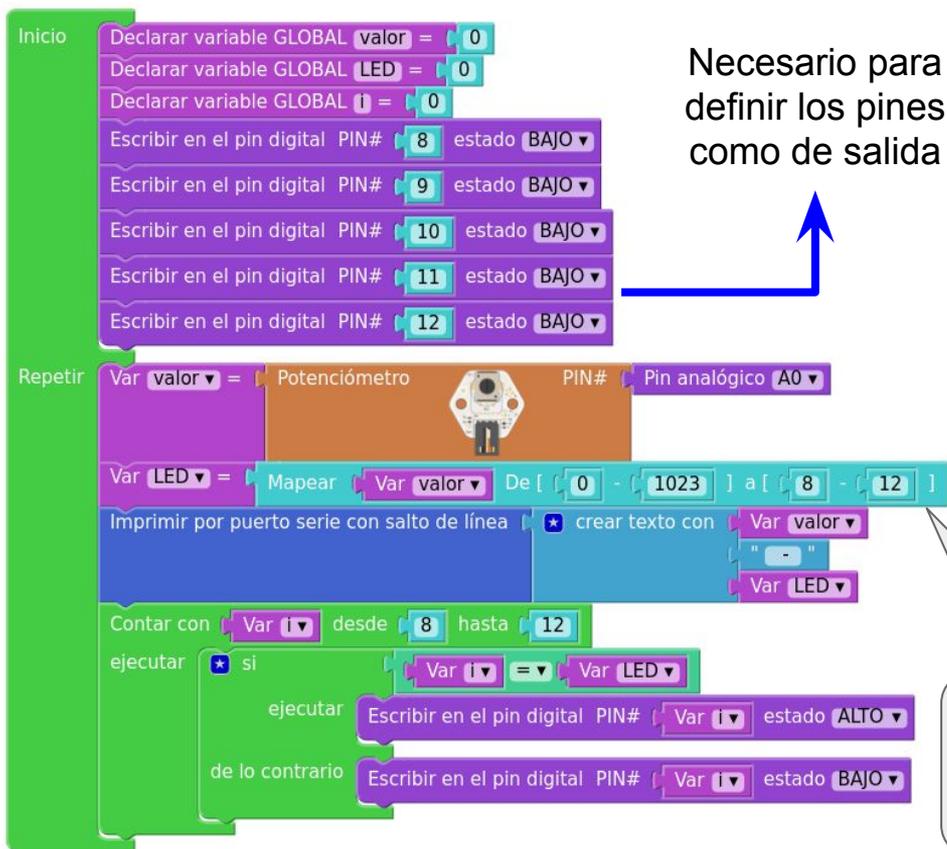
# ¿CÓMO LEE ARDUINO LAS ENTRADAS VARIABLES?

0 Voltios	→	0
1 Voltio	→	205
2 Voltios	→	409
.		.
2,5 Voltios	→	511
.		.
.		.
3,5 Voltios	→	716
4 Voltios		818
.		.
4,5 Voltios	→	921
5 Voltios	→	1023

## Fórmula

$$numero = \frac{1023}{5} \cdot V$$

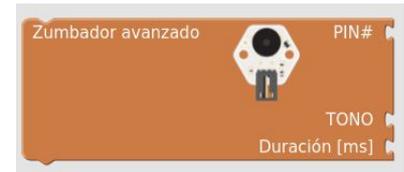
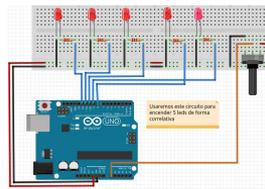
# VARIOS LEDS QUE ENCIENDEN CORRELATIVAMENTE



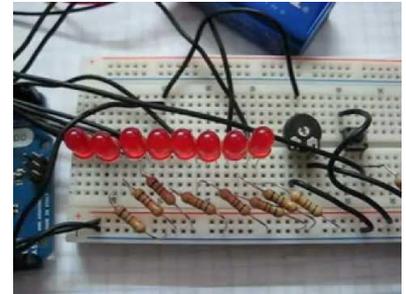
**MAPEAR:** forma fácil de hacer proporciones y reglas de tres.

# VARIOS LEDS QUE ENCIENDEN CORRELATIVAMENTE

1. ¿Qué modificación hay que hacer para que los LEDs avancen correlativamente PERO no se vayan apagando?
2. Tal como está hecho el programa, siempre se queda uno encendido. ¿Qué tengo que hacer para que empiecen desde “todos apagados”?
3. Propuesta: utilizando el bloque “zumbador avanzado” (zum bloqs) intenta que suene un tono diferente musical en función del led que se esté iluminando.
4. Ampliación: intenta hacer las luces del coche fantástico.



[https://youtu.be/Gt\\_JPdOwH7s](https://youtu.be/Gt_JPdOwH7s)



# SENSORES DE LUZ

(LDR - LIGHT DEPENDENT RESISTOR)

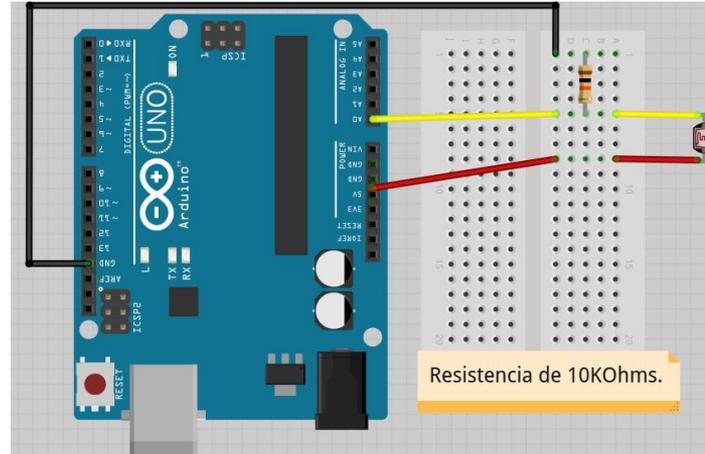
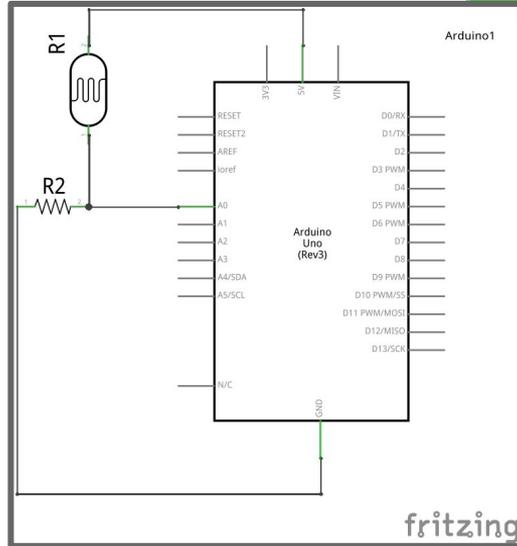


# RESISTENCIAS FOTODEPENDIENTES

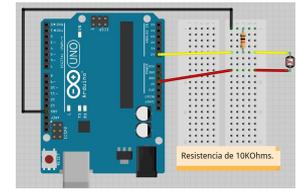
Las LDR son dispositivos que cambian su valor de resistencia cuando inciden en ellas más o menos luz. Para poder usarlas haz lo siguiente:

1. Con un ohmímetro, mide el valor de la resistencia a oscuras, medidas en un cajón o tapándolas completamente con algo. Anota el valor  $R_{\text{oscuras}}$  (si no puedes medirla porque es demasiado alta coge el valor  $1000\text{K}\Omega$  o  $1\text{M}\Omega$ )
2. Anota el valor cuando incide unas condiciones buenas de luz (iluminándolas con una linterna o similar). Ese es el valor  $R_{\text{iluminada}}$ .
3. Escoge un valor intermedio. Busca una resistencia de ese valor y ponla en serie con ella para fabricar un divisor de tensiones.
4. Usa el programa de comunicación serie sencillo que usaste con el potenciómetro para obtener valores con la LDR tapada y sin tapar. Averigua un valor intermedio.

# RESISTENCIAS FOTODEPENDIENTES



# ALARMA DE CAJÓN

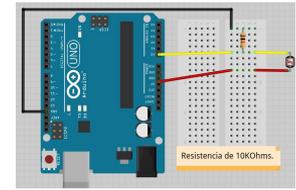


Imagina que en un cajón (o en una caja fuerte) tienes algo muy valioso escondido. Estás preocupado/a de que nadie te robe lo que tienes. Intenta hacer una alarma con un zumbador que suene cuando alguien abra el cajón.

**Fundamento:** Al iluminar la LDR cambia el valor que detecta ARDUINO. Detecta ese cambio para activar el zumbador.

**Ampliación.** Modifica la alarma de cajón para que haga lo contrario: que se encienda un LED cuando esté a oscuras. Sería un circuito que permite encender una farola cuando cae la noche o hay circunstancias de oscuridad (días muy nublados).

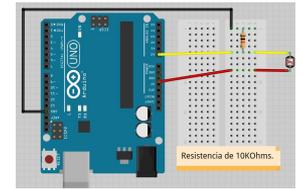
# ALARMA DE CAJÓN



Para la otra alarma (la de farola) hay dos posibilidades:

O hacer un nuevo programa, que “suene” o “encienda un LED” cuando se baje de cierto grado de luz o se puede dar la vuelta al LDR y a la resistencia (cambiar valor de resistencia).

# IDEAS CON LDRS



1. Describe cómo podrías hacer un interruptor con un LDR.
2. Inventa un juego con dos LDRs.

# SENSORES DE TEMPERATURA

(NTC-PTC)



# RESISTENCIAS TERMODEPENDIENTES

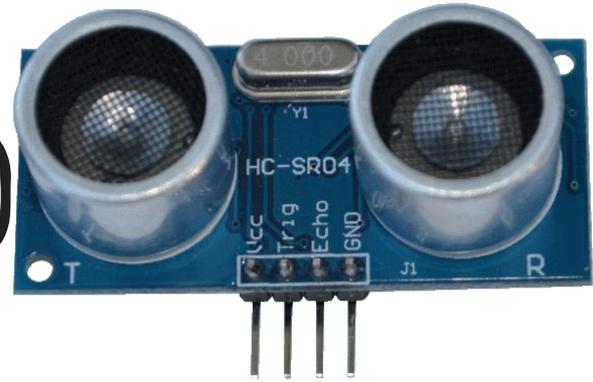
Las NTC-PTC son resistencias cuyos valores dependen de la temperatura. Las NTC descienden su resistencia al aumentar la temperatura y las PTC al contrario, aumentan su resistencia cuando aumenta la temperatura.

En todo caso, hay que estudiarlas ANTES de poder usarlas (calibración). Sin embargo, para ciertas aplicaciones basta con probarlas experimentalmente. Por ejemplo, poner una NTC en serie con una resistencia de  $10K\Omega$  y comprobar los valores a temperatura ambiente y calentándola al apretarla con los dedos. Hacer un programa que encienda un LED cuando pase de cierto valor. Comprobar cómo la temperatura es un parámetro que no cambia tan rápidamente como la cantidad de luz.

# RESISTENCIAS TERMODEPENDIENTES



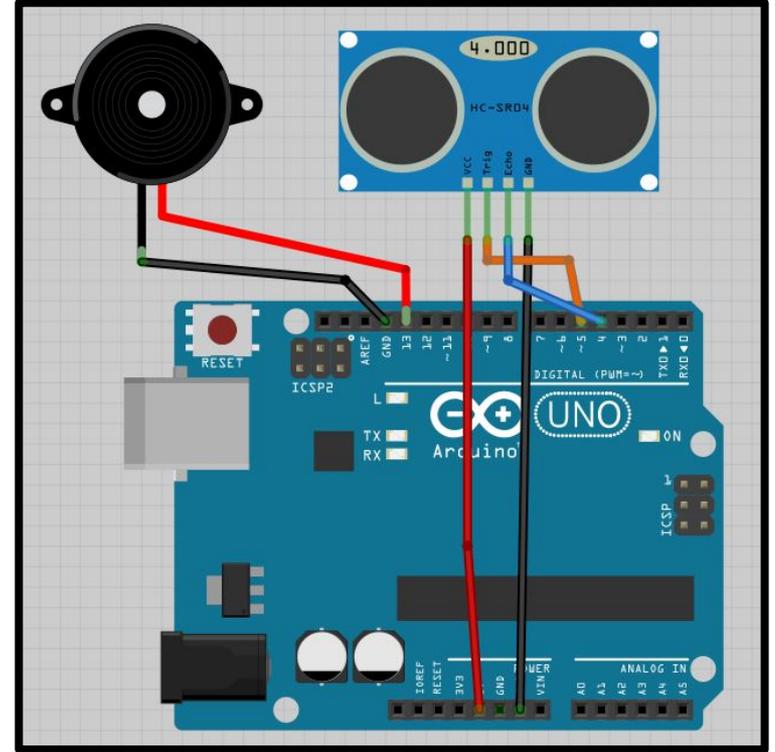
# SENSOR DE ULTRASONID



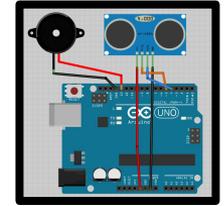
# SENSOR DE ULTRASONIDOS

El sensor de ultrasonidos tiene cuatro conexiones. Tensión de alimentación o Vcc que hay que conectar a 5V, tierra o GND, ECHO y TRIGGER. La combinación de los dos últimos nos da una distancia a un obstáculo que se encuentre enfrente de nosotros.

Hacer un programa que “pite” cuando se acerque un obstáculo a menos de 30cm de nosotros.



# SENSOR DE ULTRASONIDOS



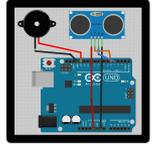
Activa el sensor de distancias

Reconoce la distancia de corte

Toca un tono y enciende el LED

Apaga el LED si está fuera del rango

# PÉNDULO CON ARDUINO



El estudio del péndulo, en Físicas, ha tenido siempre mucha trascendencia. A Galileo se le atribuye originariamente las primeras fórmulas del mismo.

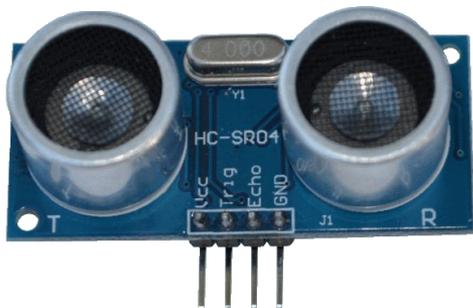
Cuando hacemos oscilar un péndulo levemente, podemos observar que el tiempo que tarda en oscilar no depende de la masa del péndulo, sino de su longitud y del valor de la gravedad. Se demuestra que para oscilaciones pequeñas, se cumple:

Siendo  $l$  la longitud de la cuerda (en metros),  $g=9.8 \text{ m/s}^2$  y  $\pi=3.1416$

$$T \approx 2\pi\sqrt{\frac{l}{g}}$$

**Experimento:** monta un péndulo de longitud conocida y calcula su período. Hazlo oscilar delante de un sensor de ultrasonidos e inventa un programa que calcule dicho período. Comprueba la exactitud de ambos valores.

# EL PROBLEMA DEL SENSOR SRO4

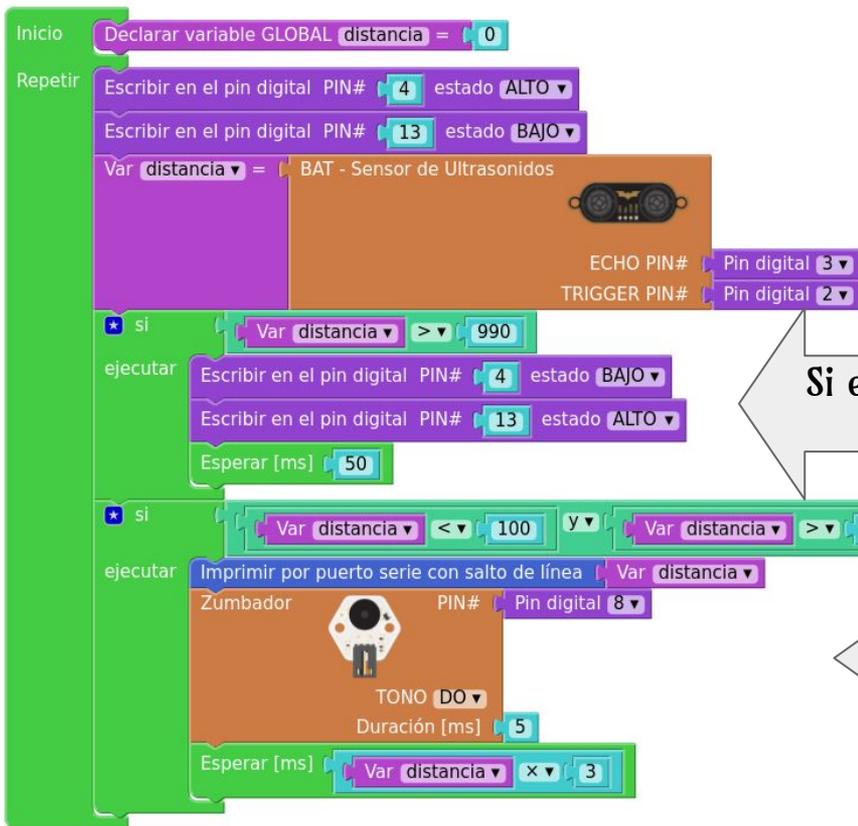


El sensor de ultrasonidos SRO4 tiene un problema (que al parecer no tiene su hermano el SR05) y es que cuando mide una “distancia infinita” (fuera de rango) se queda “colgado”. El dispositivo, una vez disparado (TRIGGER), espera un pulso por el pin ECHO. Si no tiene o no funciona un timeout que lo saque de la espera, se quedará bloqueado al no recibir el eco.

En foros de internet se encuentran dos soluciones (por ahora). La primera es asequible con VISUALINO; la segunda sólo con el IDE de ARDUINO.

- Apagar/encender el sensor. Vamos, reinicializarlo (no el programa, el sensor). Se puede conseguir alimentando el sensor (Vcc) a través de un transistor que se active por otro pin de ARDUINO que cuando detecte un bloqueo lo apague momentáneamente.
- Al detectar un bloqueo, realizar la siguiente secuencia en ARDUINO: definir el pin ECHO como salida, ponerlo en estado bajo, hacer un pequeño delay (empezar con 100ms) y volverlo a definir como entrada. Eso debe sacarlo de su bloqueo.

# PROBLEMA EJEMPLO CON SRO4



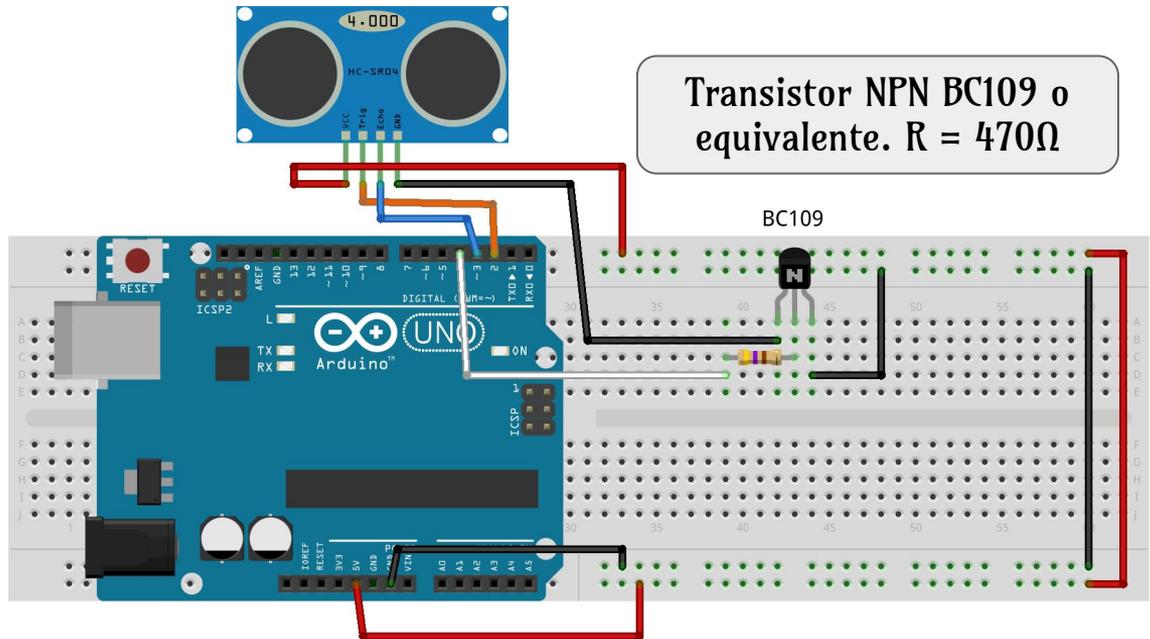
Activa el 4: Encender SRO4 // Apaga LED 13

Almacena distancia detectada por el sensor

Si es "infinita", mayor de 990, pin 4 bajo (apagar sensor).  
Enciende LED y espera un poco (50ms)

En un rango definido (5,100), imprime distancia por puerto serie y toca un tono variable según la distancia.

# PROBLEMA EJEMPLO CON SR04



fritzing



Aurelio Gallardo Rodríguez



# Y ASÍ SE HACE EL PÉNDULO

- Montar un péndulo. Simplemente algo que pese colgado de una cuerda. Cuanto más largo podamos montarlo, mejor deben salir los resultados.
- Dirigir el sensor hacia una superficie de manera que siempre mantenga una lectura constante de distancia. El péndulo debe estar montado de forma que su oscilación corte el haz del sensor.
- El programa debe registrar datos en serie. Esos datos, recogidos en el monitor, deben pasarse a una hoja de cálculo, en una columna. En la hoja de cálculo, añadir una columna temporal que empiece en 0 y de paso sea el tiempo (delay) espera que pusimos en el programa (en mi caso 20ms). Representar gráficamente los resultados. Deducir el período en función de los valles o picos de la gráfica y compararlos con los de la fórmula.
- **CONSEJOS:** péndulos largos, oscilaciones pequeñas. Intentar que el péndulo no gire sobre sí mismo (péndulo cónico).

