

KIT 38 Sensores (Todo Micro) Recopilado de Internet – Compilado Prof: Bolaños D.J (Versión 15-3-19)

Parte 2

- Modulo color LED SMD ky-009
- Modulo receptor infrarojo ky-02
- Modulo emisor infrarojo ky-005
- Modulo interruptor de mercurio ky-017
- Modulo Magnético ky-021
- Modulo RGB KY-016
- Modulo buzzer activo KY-012
- Modulo sensor analogico de temperatura ky-013

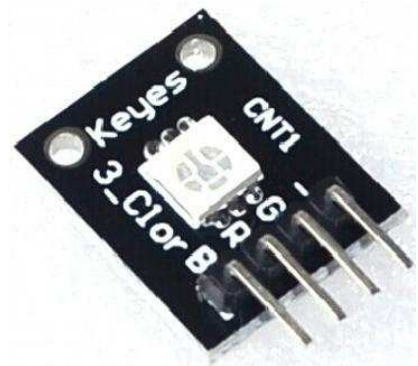
Modulo color LED SMD ky-009

El **módulo LED SMD de 3 colores**

RGB para Arduino consiste en un módulo **LED** con 3 diodos con los 3 colores primarios, rojo, verde y azul, con tres entradas y un cátodo común. Con las salidas PWM se puede controlar la intensidad de cada diodo LED.

Controlando el módulo con **Arduino** se pueden lograr efectos de iluminación sorprendentes.

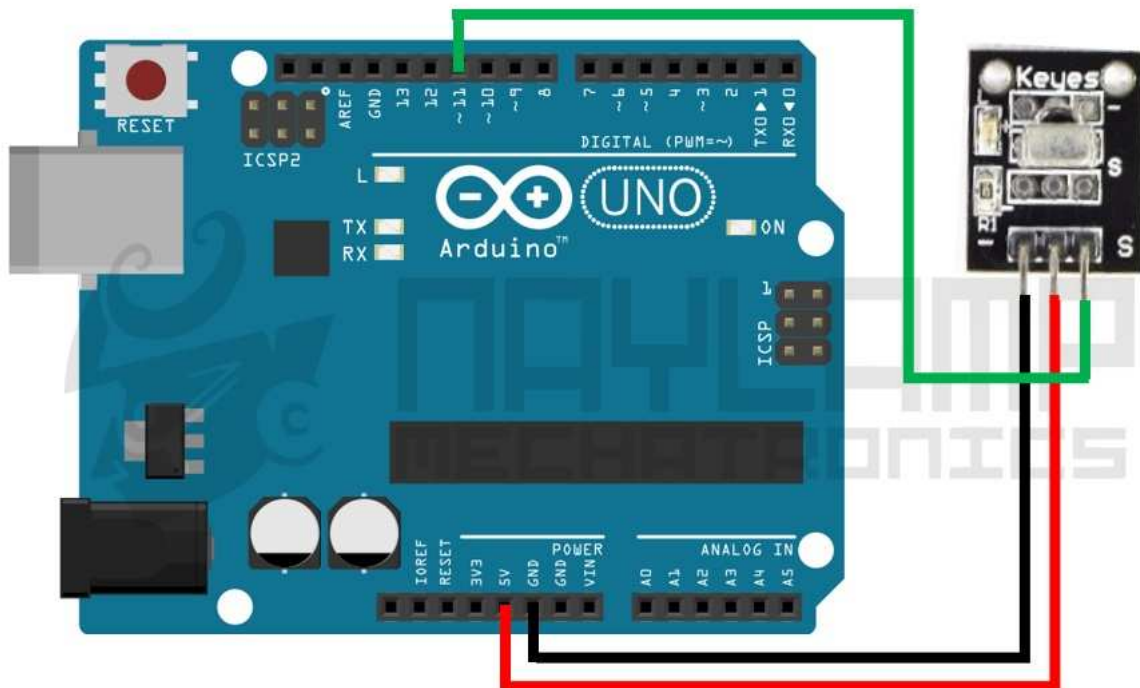
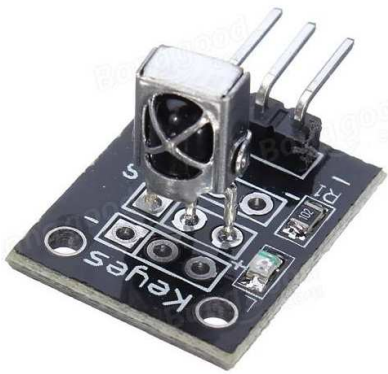
Lo hemos trabajado en tachos LED.



El **módulo LED SMD de 3 colores RGB** para **Arduino** consiste en un módulo **LED** con 3 diodos con los 3 colores primarios, rojo, verde y azul, con tres entradas y un cátodo común.

Con las salidas PWM se puede controlar la intensidad de cada diodo LED. Controlando el módulo con **Arduino** se pueden lograr efectos de iluminación sorprendentes.

Modulo receptor infrarrojo ky-02



Cargando este código que tenemos en el Tutor. Nos permite decodificar controles remotos.

```
#include <boarddefs.h>
#include <IRremote.h>
#include <IRremoteInt.h>
#include <ir_Lego_PF_BitStreamEncoder.h>

int RECV_PIN = 11;

IRrecv irrecv(RECV_PIN);

decode_results results;

void setup()
{
  Serial.begin(9600);
  irrecv.enableIRIn(); // Empezamos la recepción
  pinMode(5, OUTPUT); // configura 'pin' como salida, para visualizar
                      // llegada de desconocido
}

void dump(decode_results *results) {
  // Dumps out the decode_results structure.
  // Call this after IRrecv::decode()

  Serial.print("(");
  Serial.print(results->bits, DEC);
  Serial.print(" bits)");
}
```

```

if (results->decode_type == UNKNOWN) {
  Serial.print("Unknown encoding: ");

  digitalWrite(5,HIGH );//Pulso de llegada desconocido
  delay(100);
  digitalWrite(5,LOW );//Termina pulso

}
else if (results->decode_type == NEC) {
  Serial.print("Decoded NEC: ");

}
else if (results->decode_type == SONY) {
  Serial.print("Decoded SONY: ");
}
else if (results->decode_type == RC5) {

  Serial.print("Decoded RC5: ");
}
else if (results->decode_type == RC6) {
  Serial.print("Decoded RC6: ");
}
else if (results->decode_type == PANASONIC) {
  Serial.print("Decoded PANASONIC - Address: ");
  Serial.print(results->address, HEX);
  Serial.print(" Value: ");
}
else if (results->decode_type == LG) {
  Serial.print("Decoded LG ");
}
else if (results->decode_type == JVC) {
  Serial.print("Decoded JVC ");
}
else if (results->decode_type == AIWA_RC_T501) {
  Serial.print("Decoded AIWA RC T501 ");
}
else if (results->decode_type == WHYNTER) {
  Serial.print("Decoded Whynter ");
}
}
Serial.print(results->value, HEX);
Serial.print(" (HEX) , ");
Serial.print(results->value, BIN);
Serial.println(" (BIN)");
}
void loop() {
if (irrecv.decode(&results)) {
  dump(&results);
  irrecv.resume(); // empezamos una nueva recepción
}
delay(300);
}

```

Probamos con un control remoto Sony la función Subir Volumen e indico el siguiente código, lo usaremos con el próximo sensor:



COM4 (Arduino/Genuino Uno)

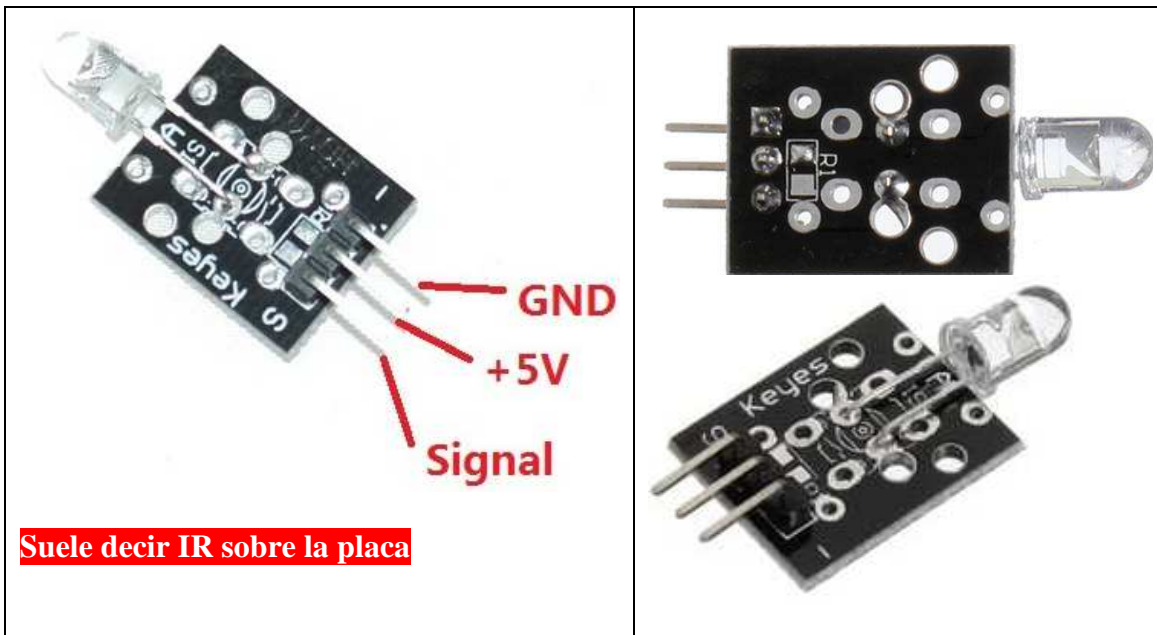
```

(12 bits)Decoded SONY: 490 (HEX) , 10010010000 (BIN)
(12 bits)Decoded SONY: 490 (HEX) , 10010010000 (BIN)
(12 bits)Decoded SONY: 490 (HEX) , 10010010000 (BIN)
(12 bits)Decoded SONY: 490 (HEX) , 10010010000 (BIN)
(12 bits)Decoded SONY: 490 (HEX) , 10010010000 (BIN)

```

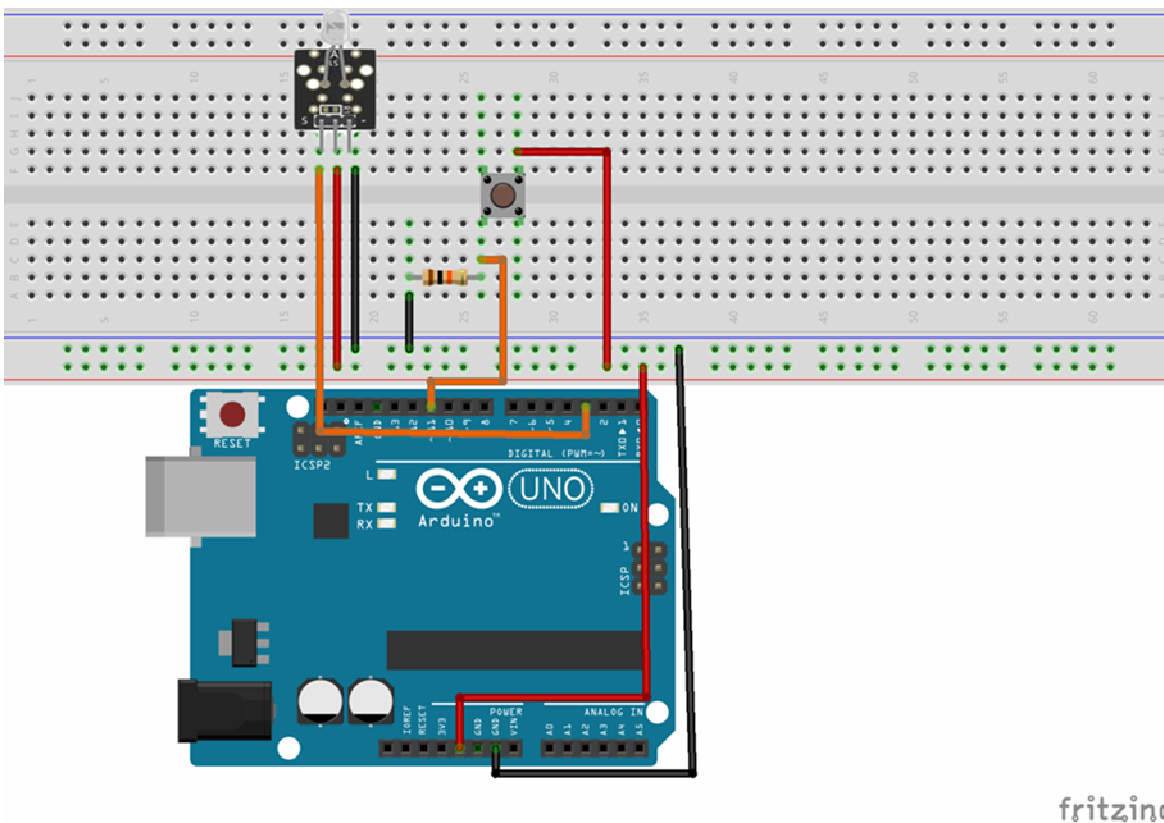
Modulo emisor infrarojo ky-005

Los sensores infrarrojos son muy cotidianos en nuestra vida hoy en día, los vemos en muchos aparatos electrónicos como televisiones, DVD's y muchos otros.



Vamos a poder encender la televisión enviándole el mismo código que el control remoto le envía a través del control, nosotros lo haremos mediante el emisor infrarrojo.

Vamos a incluir la librería `<IRremote.h>`, y ahora declaramos como emisor al IR preparándolo para enviar mensajes, declaramos el pin del pulsador, y creamos la variable del pulsador, luego configuramos al pulsador como entrada y agregamos el monitor serial para observar cuando este activo el botón, luego en el void loop, almacenamos la lectura del botón en la variable `valor_button` y agregamos condiciones, "si el botón está presionado entonces enviar el mensaje "0x20DF10EF,32" esto le envía el código del botón encendido de la TV que lo habremos obtenido previamente, y el tamaño que es de 32 bits. Agregamos un imprimir para el botón y con esto, ya podemos encender la tv.



```
//Para control remoto SONY
#include <boarddefs.h>
```

```

#include <IRremote.h>
#include <IRremoteInt.h>
#include <ir_Lego_PF_BitStreamEncoder.h>

IRsend irsend;
int pinButton = 11;

int valor_button;

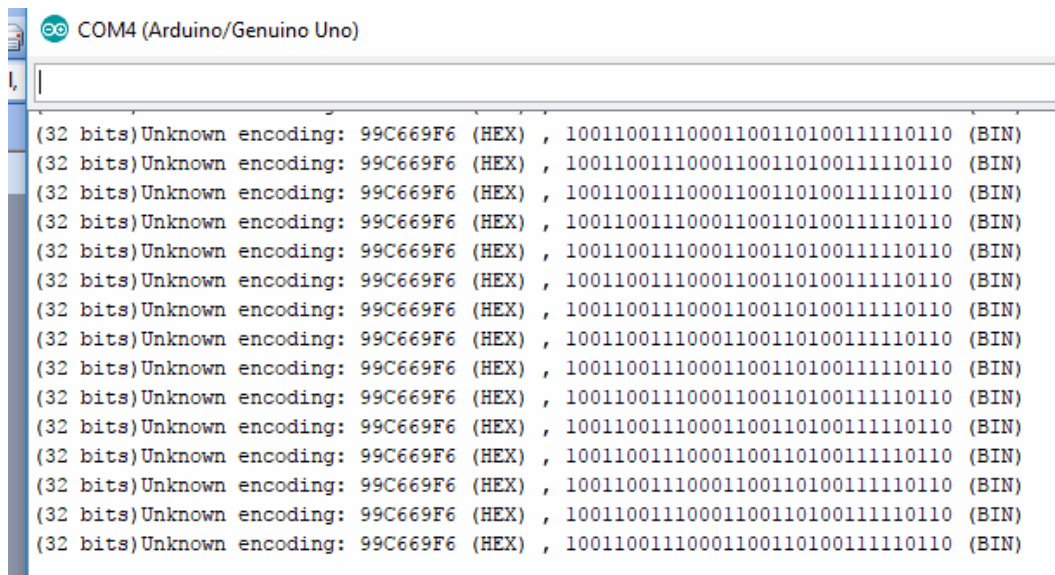
void setup()
{
  Serial.begin(9600); //solo para verificar que se esta enviando la señal
  pinMode(pinButton,INPUT); //voy a usar el pin 2 como entrada
}

void loop()
{
  valor_button = digitalRead(pinButton); //que valor tiene el pin 2
  if(valor_button == HIGH) //si esta en arriba entonces
  {
    irsend.sendNEC(0x490,12); //envia una señal (sube volumen)
    Serial.println("Enviado"); // imprime por serial que fue enviado
    delay(100); //espera 100 milisegundos
  }
}

```

El experimento funciono pero usando otro emisor infrarrojo tomado de un viejo control Sony, NO el que viene en el KIT, que no presenta a la vista y ensayo con tester un comportamiento en infrarrojo.

Armado con el emisor del Kit con el programa anterior dio en el receptor lo siguiente:



```

COM4 (Arduino/Genuino Uno)
(32 bits)Unknown encoding: 99C669F6 (HEX) , 10011001110001100110100111110110 (BIN)
(32 bits)Unknown encoding: 99C669F6 (HEX) , 10011001110001100110100111110110 (BIN)
(32 bits)Unknown encoding: 99C669F6 (HEX) , 10011001110001100110100111110110 (BIN)
(32 bits)Unknown encoding: 99C669F6 (HEX) , 10011001110001100110100111110110 (BIN)
(32 bits)Unknown encoding: 99C669F6 (HEX) , 10011001110001100110100111110110 (BIN)
(32 bits)Unknown encoding: 99C669F6 (HEX) , 10011001110001100110100111110110 (BIN)
(32 bits)Unknown encoding: 99C669F6 (HEX) , 10011001110001100110100111110110 (BIN)
(32 bits)Unknown encoding: 99C669F6 (HEX) , 10011001110001100110100111110110 (BIN)
(32 bits)Unknown encoding: 99C669F6 (HEX) , 10011001110001100110100111110110 (BIN)
(32 bits)Unknown encoding: 99C669F6 (HEX) , 10011001110001100110100111110110 (BIN)
(32 bits)Unknown encoding: 99C669F6 (HEX) , 10011001110001100110100111110110 (BIN)
(32 bits)Unknown encoding: 99C669F6 (HEX) , 10011001110001100110100111110110 (BIN)
(32 bits)Unknown encoding: 99C669F6 (HEX) , 10011001110001100110100111110110 (BIN)
(32 bits)Unknown encoding: 99C669F6 (HEX) , 10011001110001100110100111110110 (BIN)
(32 bits)Unknown encoding: 99C669F6 (HEX) , 10011001110001100110100111110110 (BIN)

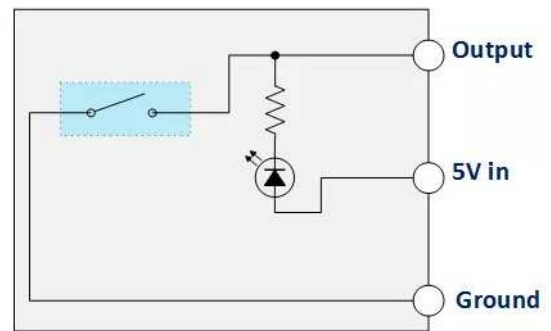
```

El tema de las diferencias será analizado en el futuro.

Modulo interruptor de mercurio ky-017

Este módulo diseñado para Arduino te permitirá detectar de forma fácil, rápida y precisa la inclinación del mismo producida en el lugar en el cual este se encuentre.

Este módulo es un conmutador de inclinación de mercurio que te permitirá detectar la inclinación de un objeto para que puedas generar una acción de control es una buena alternativa de bajo costo para reemplazar un acelerómetro.



El módulo interruptor de Mercurio es un conmutador de inclinación de mercurio que te permitirá detectar la inclinación de un objeto para que puedas generar una acción de control. Es una buena alternativa de bajo costo para reemplazar un acelerómetro.

Cuando sitúas estratégicamente este sensor, puede ser muy sensible y medir en uno o varios ejes.

La clave para usar con éxito este interruptor está en la comprensión de los dos estados de funcionamiento, usted va a tener que establecer su propia definición de lo que significa estar inclinado. Esto se debe basar en un poco de experimentación y una comprensión del circuito. El esquema adjunto está destinado a ayudar en ese entendimiento ya que cuando nos referimos a la posición inclinada, significa que el mercurio no se está conectando los contactos del interruptor. En este estado, el LED está apagado, y la salida es ALTA.

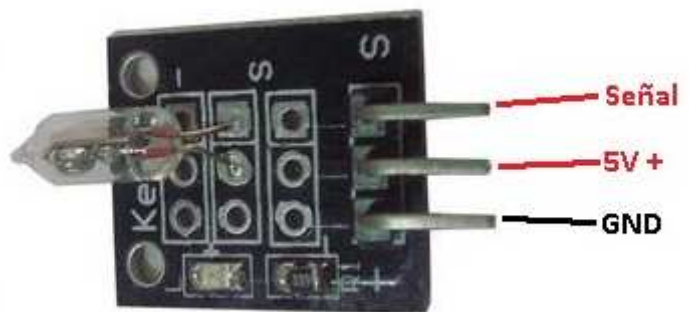
Características Técnicas:

Voltaje de funcionamiento: 5V
interruptor digital salida (0 / 1)

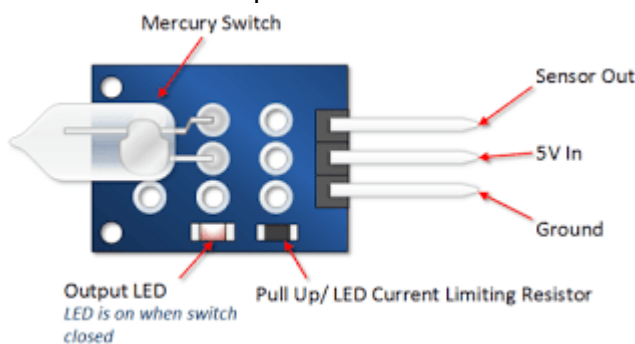
Material: PCB

Dimensiones: 2.6x1.5x0.9cm

Peso: 2g



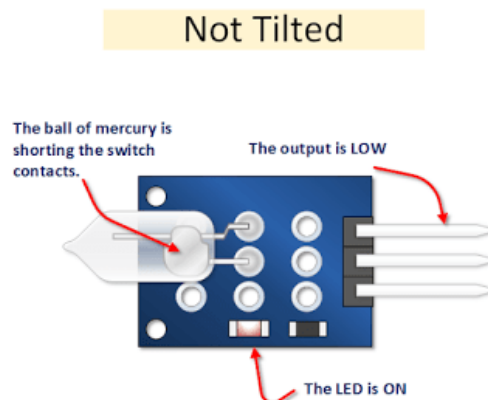
Este solamente requiere tres conexiones.



FUNCIONALIDADES - COMPRENDIENDO LOS ESTADOS DE SALIDA

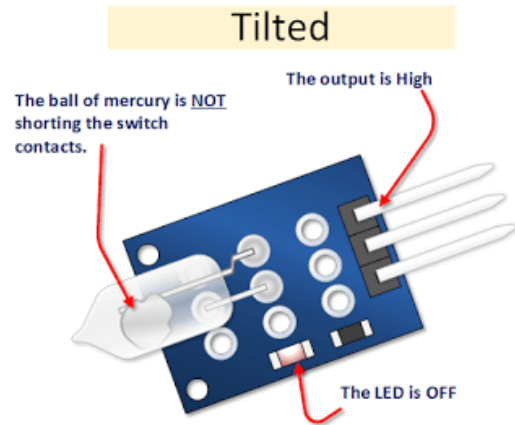
Estado No Inclinado (Not Tilted):

No inclinado significa que los contactos del interruptor se han cortocircuitados juntos por el mercurio. En este estado, la salida será baja y el LED se encenderá.



Posición Inclinada

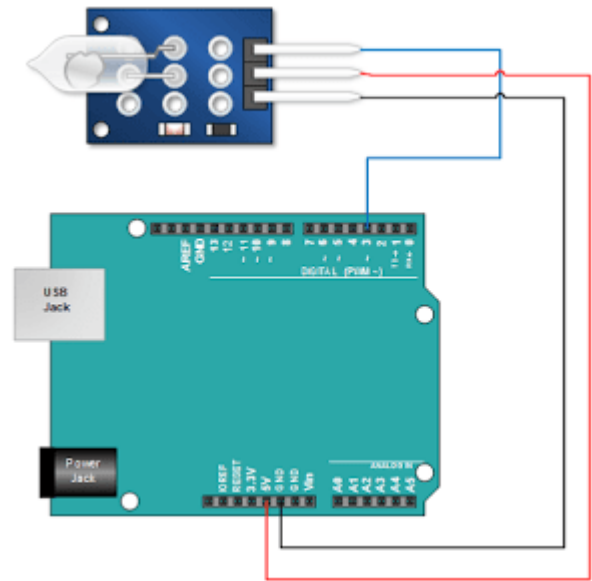
Significa que el mercurio no se está conectando los contactos del interruptor. En este estado, el LED está apagado, y la salida es ALTA.



Forma de Conexión - Implementando con Arduino

Código

El siguiente esquema detectará un cambio en los estados del interruptor de inclinación y enviar una salida a su monitor serie.



```

int tiltSwitch = 3; // Entrada de interruptor de inclinación
int tiltVal; // variable para almacenar entrada de inclinación

void setup ()
{
  Serial.begin(9600);
  pinMode (tiltSwitch, INPUT) ;// definir salida del sensor del interruptor de inclinación de mercurio
}
void loop ()
{
  tiltVal = digitalRead (tiltSwitch) ;// leer el valor del interruptor
  if (tiltVal == HIGH) // Significa que esta inclinado
  {
    Serial.println("*** - TILTED (INCLINADO) - ***");
  }

  else
  {
    Serial.println("NOT TILTED tilted (NO INCLINADO)");
  }
}
    
```

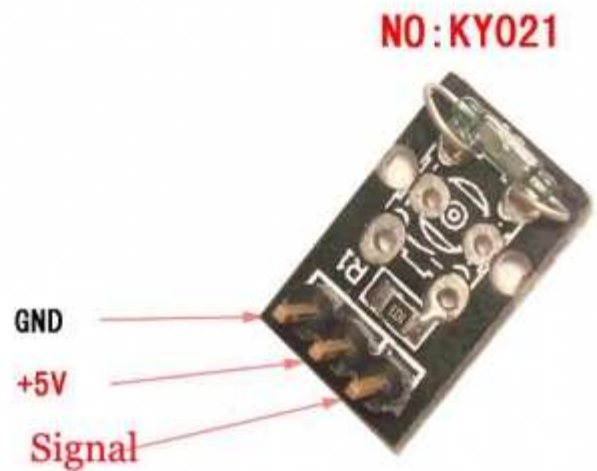
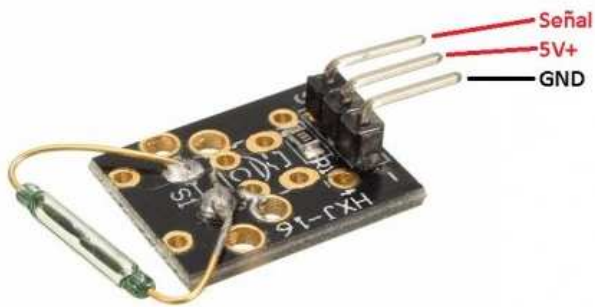
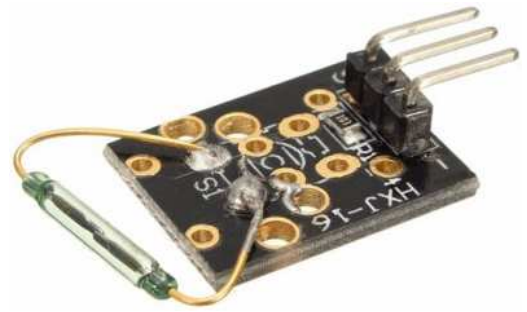
Modulo Magnetico ky-021

Características:

Voltaje de funcionamiento: 3.3V ~ 5V

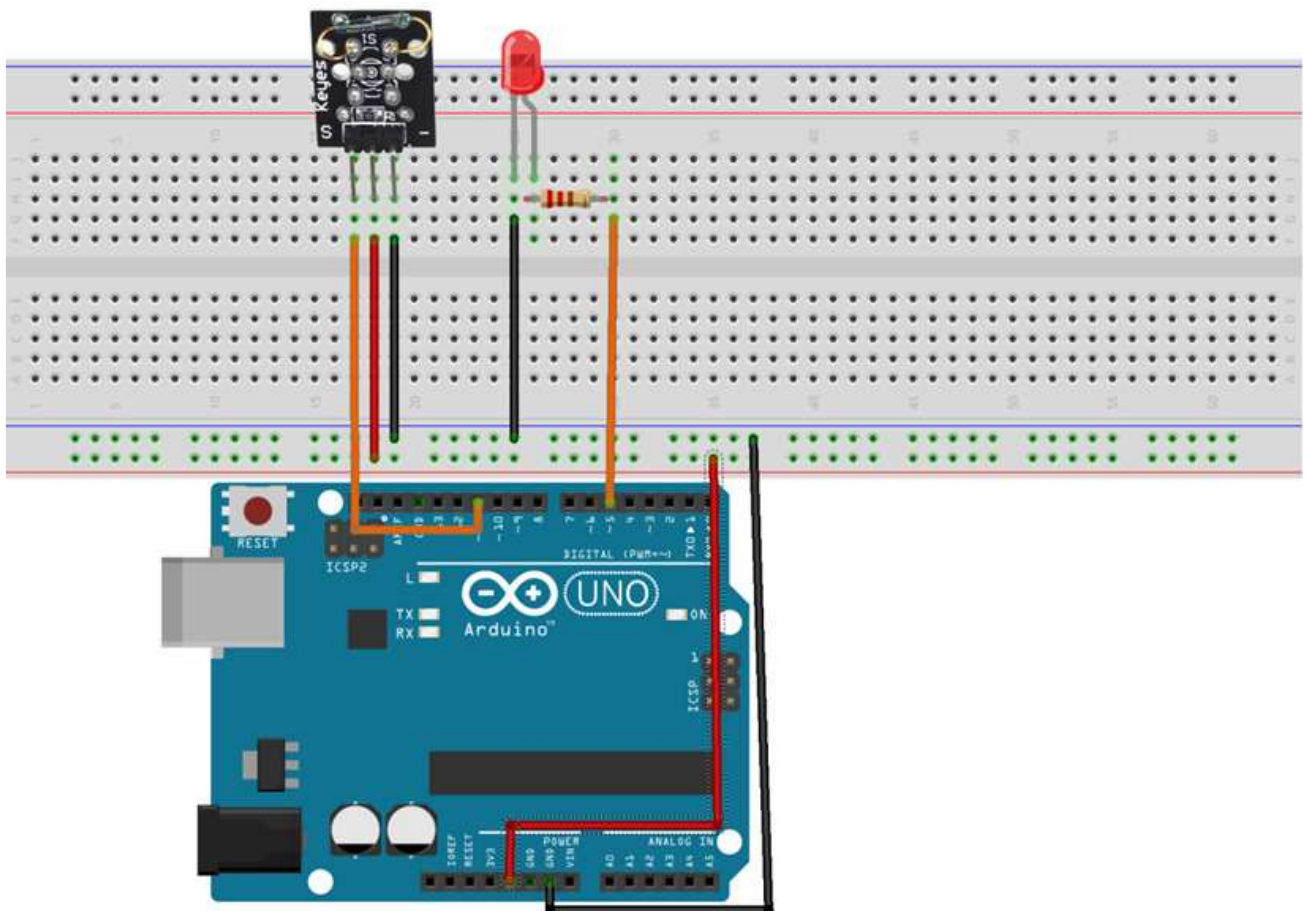
Dimensiones: 2cm x 1.5cm x 0.7cm

Peso: 1.2g



Este módulo te permitirá detectar campos magnéticos de una forma rápida, fácil y eficiente, con Mini Lámina Magnética es un sensor que se encuentra comúnmente abierto y al detectar un campo magnético se cierra, permitiendo el paso del voltaje. Este módulo incluye una resistencia de 10 kΩ y no es necesario adicionar ninguna otro resistencia.

El módulo sensor mini switch magnético KY-021 se puede utilizar como un switch magnético, ideal para proyectos con Arduino, y microcontroladores, te permitirá detectar campos magnéticos de una forma rápida, fácil y eficiente.

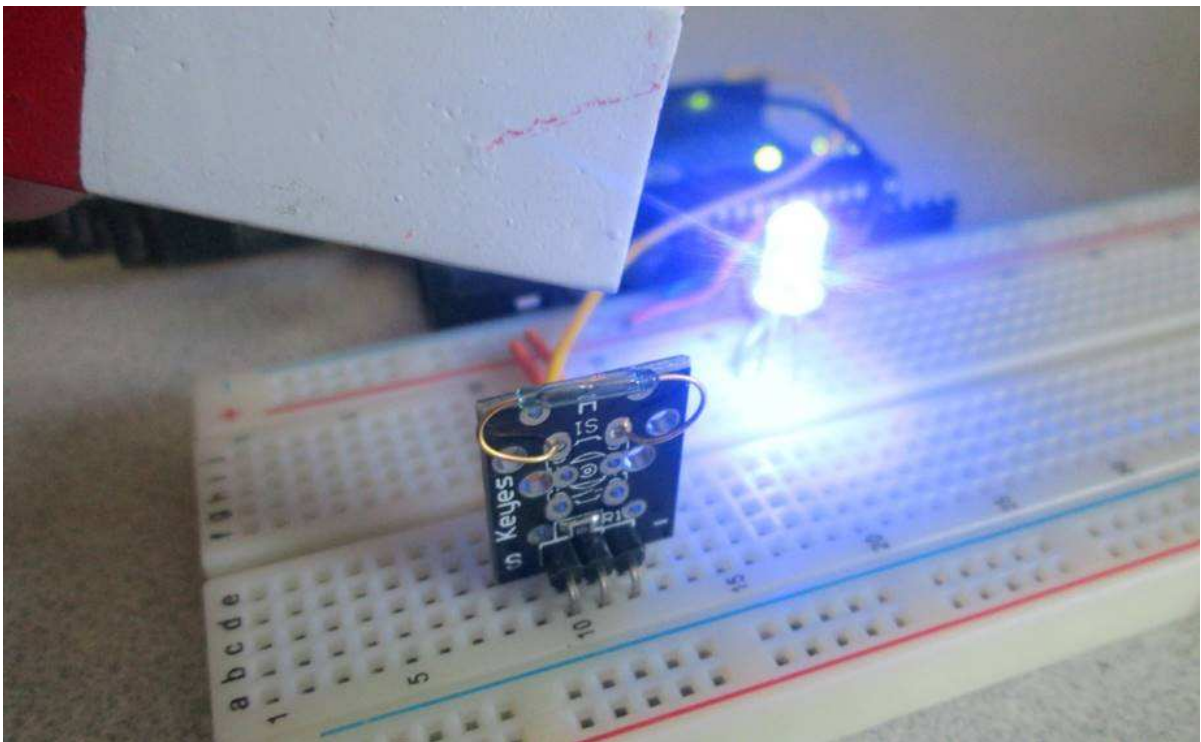



```

//Sensor magnetico KY021
int Led = 13 ;//LED en PIN13
int buttonpin = 3; //definimos la entrada digital en Arduino para el sensor
int val ;
void setup ()
{
  pinMode (Led, OUTPUT) ;
  pinMode (buttonpin, INPUT) ;
}
void loop ()
{
  val = digitalRead (buttonpin);
  if (val == HIGH)
  {
    digitalWrite (Led, LOW);
  }
  else
  {
    digitalWrite (Led, HIGH);
  }
}

```

Acercando un imán enciende LED:



Modulo Magnetico RGB KY-016

El modulo KY-016 para Arduino (5mm RGB LED), permite obtener diferentes colores mezclando los tres colores primarios.

Este módulo consta de un LED RGB de 5 mm y tres resistencias limitadoras de 150 Ω para evitar el agotamiento. El ajuste de la señal PWM en cada pin de color dará como resultado diferentes colores.

ASEGURESE QUE EL MODULO KEYES QUE ESTA UTILIZANDO CUENTE CON LAS RESISTENCIAS LIMITADORAS. DE LO CONTRARIO, DEBEN SER AGREGADAS. NO HACERLO DAÑARÍA EL MODULO.

Descripción:

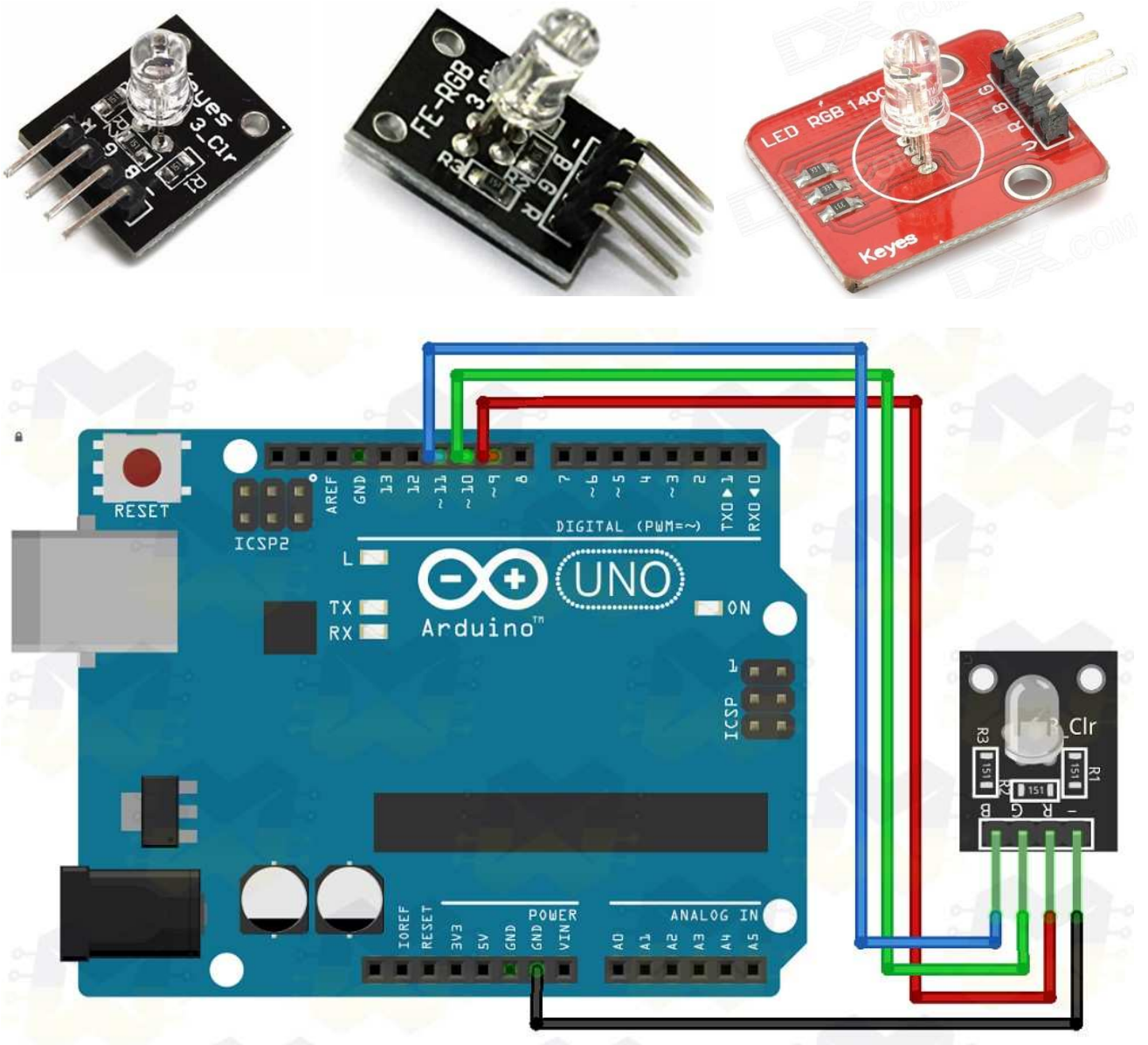
Módulo led RGB KY-016, tiene un diodo led RGB de cátodo común (O SEA LOS LED ENCIENDEN CON UNO), y 3 resistencias. **Puede usar una señal PWM en cada color para hacer muchísimas combinaciones.**

Especificaciones:

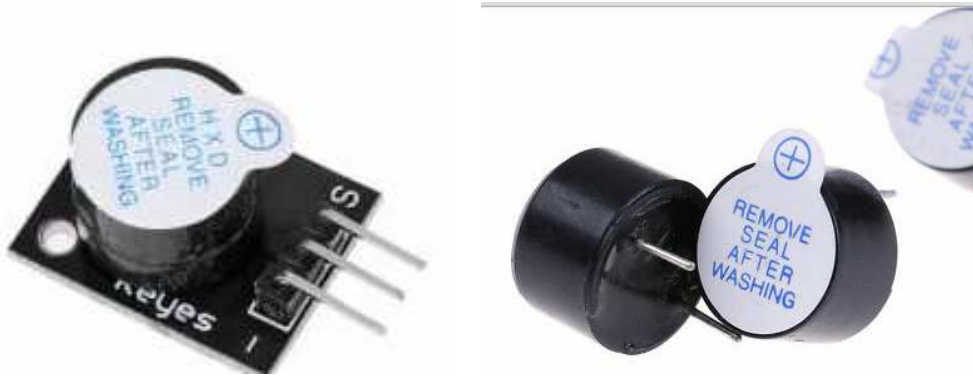
- Modelo: KY-016
- Voltaje de operación: 5 VCD
- Peso: 3 g
- Dimensiones: 24 x 16 x 15 mm

Terminales:

- R: Habilita el led rojo
- G: Habilita el led verde
- B: Habilita el led azul
- GND: GND



Modulo buzzer activo KY-012



El Módulo de zumbador activo Módulo Arduino **KY-012**, produce un sonido de un tono cuando la señal es alta.

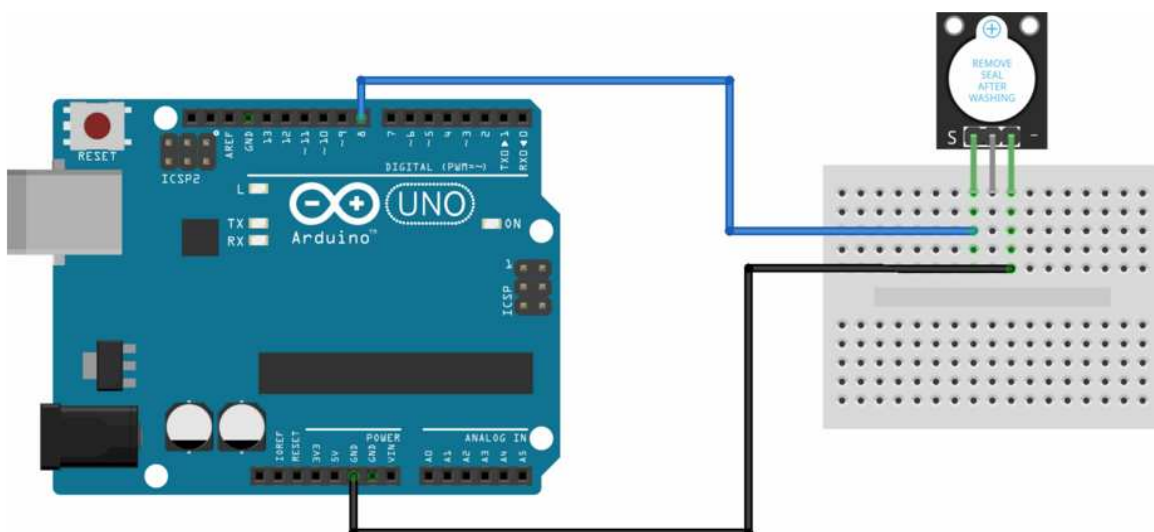
Para producir tonos diferentes, use el módulo de zumbador pasivo KY-006.

El módulo KY-012 Active Buzzer consta de un zumbador piezoeléctrico activo, que genera un sonido de aproximadamente 2,5 kHz cuando la señal es alta.

Este zumbador se utiliza comúnmente para generar alarmas sonoras en tarjetas electrónicas, computadoras, multímetros, en las series de arbolitos de navidad, etc, etc).

ESPECIFICACIONES TÉCNICAS

- Voltaje de funcionamiento 3.5V ~ 5.5V
- Corriente máxima 30mA / 5VDC
- Frecuencia de resonancia 2500Hz \pm 300Hz
- Salida de sonido mínima 85Db @ 10cm
- Temperatura de trabajo -20 °C ~ 70 °C [-4 °F ~ 158 °F]
- Temperatura de almacenamiento -30 °C ~ 105 °C [- 22 °F ~ 221 °F]
- Dimensiones 18.5mm x 15mm [0.728in x 0.591in]



Ante la facilidad de manejo, queda para el alumno realizar pruebas particulares. Seguramente ya lo hemos presentado en otros proyectos.