

Kit de Sensores y Actuadores Compatibles con Arduino

Número de parte: KIT-SEN-ACT



Descripción:

Este paquete contiene 26 sensores y 11 actuadores compatibles con tarjetas Arduino con los cuales podrás probar el funcionamiento de tu Arduino y explotarlo al máximo. Es justo lo que necesitas para comenzar tus proyectos.

El kit 37 en 1 contiene los siguientes sensores:

- 1 x módulo zumbador pasivo KY-006
- 1 x módulo LED bicolor KY-011
- 1 x módulo sensor de golpe KY-031
- 1 x módulo interruptor de vibración KY-002
- 1 x módulo de foto-resistencia KY-018
- 1 x módulo interruptor tipo boton KY-004
- 1 x módulo interruptor de inclinación KY-020
- 1 x módulo LED SMD tricolor KY-009
- 1 x módulo sensor de emisión infrarroja KY-005
- 1 x módulo LED tricolor KY-016
- 1 x módulo óptico de mercurio KY-017
- 1 x módulo LED bicolor 3mm KY-029
- 1 x módulo zumbador activo KY-012
- 1 x módulo sensor de temperatura KY-013
- 1 x módulo flash automático de 7 colores KY-034
- 1 x módulo de contacto con mini lamina magnética KY-021
- 1 x módulo sensor magnético Hall KY-003
- 1 x módulo sensor receptor infrarrojo KY-022
- 1 x modulo sensor magnético clase Bihor KY-035
- 1 x módulo taza mágica de luz KY-027
- 1 x módulo codificador rotatorio KY-040
- 1 x módulo sensor fotoeléctrico tipo U KY-010
- 1 x módulo detector de latido KY-039
- 1 x módulo sensor de reflexión KY-025
- 1 x módulo sensor de evitación de obstáculos KY-032
- 1 x módulo sensor de rastreo KY-033
- 1 x módulo micrófono con sensor de sonido KY-038
- 1 x módulo sensor láser KY-008
- 1 x módulo con relevador KY-019
- 1 x módulo sensor de temperatura KY-001
- 1 x módulo sensor de temperatura digital KY-028
- 1 x modulo sensor de campo magnético lineal KY-024
- 1 x módulo sensor de llama KY-026
- 1 x módulo sensor de micrófono sensible KY-037
- 1 x módulo sensor de temperatura y humedad KY-015
- 1 x módulo joystick de eje XY-KY-023
- 1 x módulo sensor sonido metálico KY-036
- 1 x caja

A continuación se mostrara información de cada uno de los sensores, actuadores y códigos ejemplo para su uso.

Para consultar la página oficial del fabricante sigue el siguiente link:

<http://en.keyes-robot.com/productshow.aspx?id=152>

Arduino KY-001: Modulo Sensor de Temperatura



Los Arduino tienen un gran número de usos, uno de ellos es la medición de la temperatura, en este caso usaremos un sensor DS18B20, el cual está integrado en este módulo y además es uno de los más usados en el mundo.

Con este módulo podrás medir la temperatura de lo que desees, por ejemplo una habitación, el coche y muchas opciones más.

Características:

- Rango de Voltaje: 3.0 V ~ 5.5 V
- Rango de Temperatura: -55°C ~ +125°C
- Rango de Temperatura: 67°F ~ 257°F
- Rango de Precisión: $\pm 0.5^{\circ}\text{C}$

Ejemplo:

Material Necesario:

- Arduino x1
- Modulo Sensor de Temperatura DS18B20 x 1
- Cable USB x 1
- Descargar e instalar la librería [OneWire library](#)

Código:

```
#include <OneWire.h>

// DS18B20/DS18S20 Temperature chip i/o
OneWire ds(10); // on pin 10

void setup(void) {
  // initialize inputs/outputs
  // start serial port
  Serial.begin(9600);
}

void loop(void) {

  //For conversion of raw data to C
  int HighByte, LowByte, TReading, SignBit, Tc_100, Whole, Fract;

  byte i;
  byte present = 0;
  byte data[12];
  byte addr[8];

  if ( !ds.search(addr) ) {
    Serial.print("No more addresses.\n");
    ds.reset_search();
    return;
  }

  Serial.print("R=");
  for( i = 0; i < 8; i++) {
    Serial.print(addr[i], HEX);
    Serial.print(" ");
  }

  if ( OneWire::crc8( addr, 7) != addr[7] ) {
    Serial.print("CRC is not valid!\n");
    return;
  }

  if ( addr[0] == 0x10) {
    Serial.print("Device is a DS18S20 family device.\n");
  }
  else if ( addr[0] == 0x28) {
    Serial.print("Device is a DS18B20 family device.\n");
  }
  else {
    Serial.print("Device family is not recognized: 0x");
    Serial.println(addr[0],HEX);
    return;
  }

  ds.reset();
  ds.select(addr);
  ds.write(0x44,1); // start conversion, with parasite power
on at the end

  delay(1000); // maybe 750ms is enough, maybe not
  // we might do a ds.depower() here, but the reset will take care of
it.
```

```

present = ds.reset();
ds.select(addr);
ds.write(0xBE);          // Read Scratchpad

Serial.print("P=");
Serial.print(present,HEX);
Serial.print(" ");
for ( i = 0; i < 9; i++) {          // we need 9 bytes
  data[i] = ds.read();
  Serial.print(data[i], HEX);
  Serial.print(" ");
}
Serial.print(" CRC=");
Serial.print( OneWire::crc8( data, 8), HEX);
Serial.println();

//Conversion of raw data to C
LowByte = data[0];
HighByte = data[1];
TReading = (HighByte << 8) + LowByte;
SignBit = TReading & 0x8000; // test most sig bit
if (SignBit) // negative
{
  TReading = (TReading ^ 0xffff) + 1; // 2's comp
}
Tc_100 = (6 * TReading) + TReading / 4;    // multiply by (100 *
0.0625) or 6.25

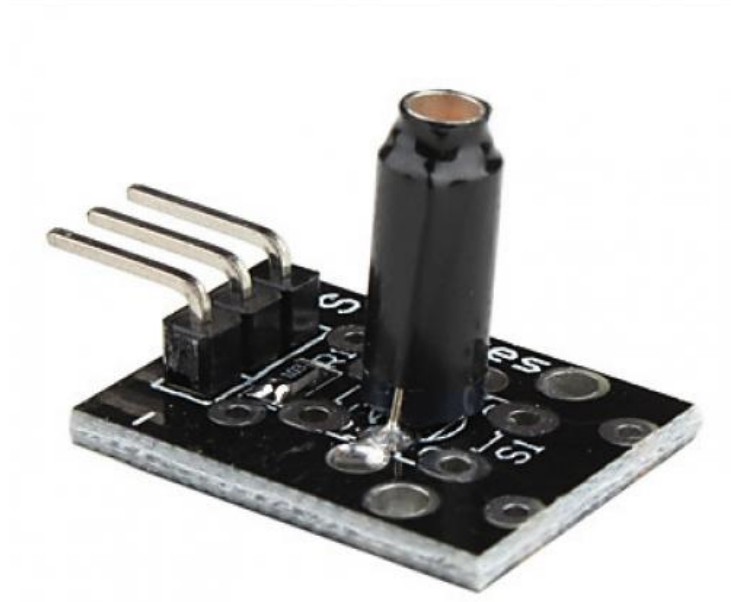
Whole = Tc_100 / 100; // separate off the whole and fractional
portions
Fract = Tc_100 % 100;

if (SignBit) // If its negative
{
  Serial.print("-");
}
Serial.print(Whole);
Serial.print(".");
if (Fract < 10)
{
  Serial.print("0");
}
Serial.print(Fract);

Serial.print("\n");
//End conversion to C
}

```

Arduino KY-002: Modulo Sensor de Vibración

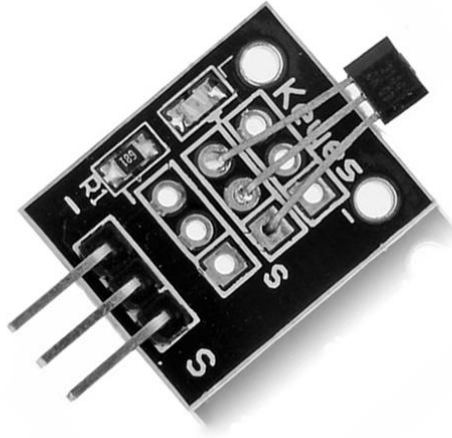


Este módulo sensor de vibraciones funciona como un Switch, al detectar una vibración cierra un circuito lo que genera que el usuario pueda detectar los movimientos. El modulo debe conectarse el PIN 10 del Arduino. Cuando detecta una vibración se enciende el LED conectado al PIN 13.

Ejemplo:

```
int Led = 13 ;// define LED Interface
int Shock = 10; // define the vibration sensor interface
int val; // define numeric variables val
void setup ()
{
  pinMode (Led, OUTPUT) ; // define LED as output interface
  pinMode (Shock, INPUT) ; // output interface defines vibration
  sensor
}
void loop ()
{
  val = digitalRead (Shock) ; // read digital interface is assigned a
  value of 3 val
  if (val == HIGH) // When the shock sensor detects a signal, LED
  flashes
  {
    digitalWrite (Led, LOW);
  }
  else
  {
    digitalWrite (Led, HIGH);
  }
}
```

Arduino KY-003: Modulo Sensor de Campo Magnético

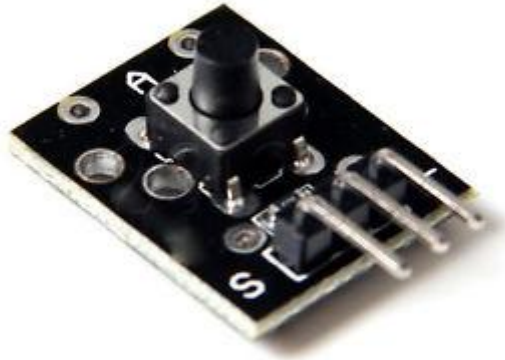


Este sensor mide un campo magnético. Si existe la presencia de un campo magnético se creara una señal en alto. En este ejemplo, el LED de la placa Arduino (pin 13) se enciende cuando la señal del sensor es alta. También el LED propio del módulo se enciende cuando la señal esta en alto.

Ejemplo:

```
int Led = 13 ; // define LED Interface
int SENSOR = 10 ; // define the Hall magnetic sensor interface
int val ; // define numeric variables val
void setup ()
{
  pinMode (Led, OUTPUT) ; // define LED as output interface
  pinMode (SENSOR, INPUT) ; // define the Hall magnetic sensor output
  interface
}
void loop ()
{
  val = digitalRead (SENSOR) ; // read digital interface is assigned a
  value of 3 val
  if (val == HIGH) // When the shock sensor detects a signal, LED
  lights
  {
    digitalWrite (Led, HIGH);
  }
  {
    digitalWrite (Led, LOW);
  }
}
```

Arduino KY-004: Modulo Interruptor Tipo Botón

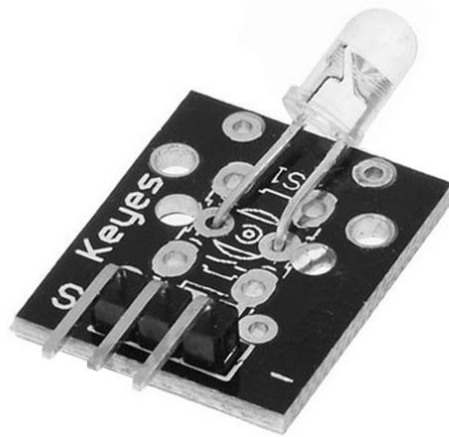


Este módulo tiene integrado un interruptor común, se muestra un ejemplo en el cual, al presionar el botón se encenderá el LED conectado al PIN 13 del Arduino.

Ejemplo:

```
int Led = 13 ;// define LED Interface
int buttonpin = 10; // define the key switch sensor interface
int val ;// define numeric variables val
void setup ()
{
  pinMode (Led, OUTPUT); // define LED as output interface
  pinMode (buttonpin, INPUT); // define the key switch sensor output
  interface
}
void loop ()
{
  val = digitalRead (buttonpin); // digital interface will be assigned
  a value of 3 to read val
  if (val == HIGH) // When the key switch when the sensor detects a
  signal, LED flashes
  {
    digitalWrite (Led, HIGH);
  }
  else
  {
    digitalWrite (Led, LOW);
  }
}
```


Arduino KY-005: Modulo Sensor Emisor Infrarrojo



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

- Corriente: 30 ~ 60 mA
- Corriente de Pulso: 0.3 ~ 1 A
- Voltaje Inverso: 5V
- Potencia de disipación: 90 mW
- Rango de Temperatura: -25 ~ +80°C
- Temperatura de Almacenamiento: -40 ~ +100°C

Ejemplo:

Para este ejemplo se necesita el siguiente material:

- Arduino × 1
- Cable USB × 1
- Modulo transmisor infrarrojo × 1
- Modulo receptor infrarrojo × 1

Código:

```
# Include <IRremote.h>
int RECV_PIN = 11; // define input pin on Arduino
IRrecv irrecv (RECV_PIN);
decode_results results;
void setup ()
{
  Serial.begin (9600);
  irrecv.enableIRIn (); // Start the receiver
}
void loop () {
  if (irrecv.decode (& results)) {
    Serial.println (results.value, HEX);
    irrecv.resume (); // Receive the next value
  }
}
Main emission part of the code:
# Include <IRremote.h>
IRsend irsend;
void setup ()
{
  Serial.begin (9600);
}
void loop () {
  for (int i = 0; i <50; i ++ ) {
    irsend.sendSony (0xa90, 12); // Sony TV power code
    delay (40);
  }
}
```

Arduino KY-006 Modulo Buzzer Pasivo



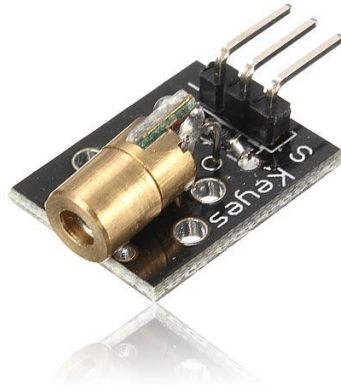
Las aplicaciones más comunes con Arduino son utilizando LED, pero también es posible realizar un sinnúmero de aplicaciones con sonido, para ello podemos utilizar este módulo con un buzzer integrado.

Los buzzer tienen un rol muy importante en la estructura de transductores electrónicos, son ampliamente utilizados en computadoras, impresoras, fotocopiadoras, alarmas, juguetes electrónicos, teléfonos, timers, productos de electrónica para equipos de sonido y muchos más.

Ejemplo: Para este ejemplo el buzzer deberá ser conectado en el PIN 8 del Arduino.

```
int buzzer = 8 ;// setting controls the digital IO foot buzzer
void setup ()
{
  pinMode (buzzer, OUTPUT) ;// set the digital IO pin mode, OUTPUT out
  of Wen
}
void loop ()
{
  unsigned char i, j ;// define variables
  while (1)
  {
    for (i = 0; i <80; i++) // Wen a frequency sound
    {
      digitalWrite (buzzer, HIGH) ;// send voice
      delay (1) ;// Delay 1ms
      digitalWrite (buzzer, LOW) ;// do not send voice
      delay (1) ;// delay ms
    }
    for (i = 0; i <100; i++) // Wen Qie out another frequency sound
    {
      digitalWrite (buzzer, HIGH) ;// send voice
      delay (2) ;// delay 2ms
      digitalWrite (buzzer, LOW) ;// do not send voice
      delay (2) ;// delay 2ms
    }
  }
}
```

Arduino KY-008: Modulo Sensor Laser

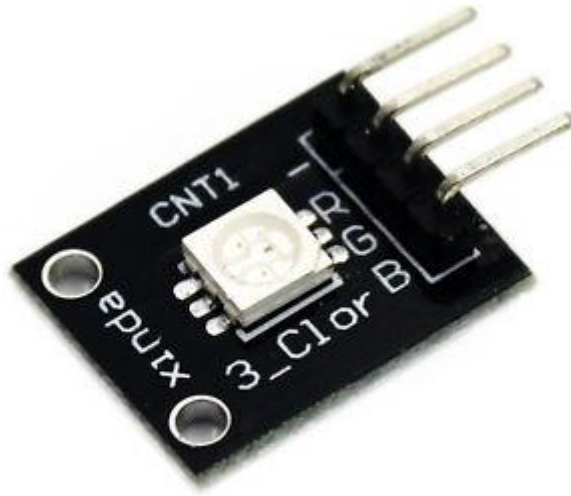


Este módulo tiene integrado un transmisor laser y una interfaz digital, además cuenta con un LED incorporado.

Ejemplo:

```
void setup ()
{
  pinMode (13, OUTPUT); // define the digital output interface 13
  feet
}
void loop () {
  digitalWrite (13, HIGH); // open the laser head
  delay (1000); // delay one second
  digitalWrite (13, LOW); // turn off the laser head
  delay (1000); // delay one second
}
```

Arduino KY-009: Modulo LED SMD tricolor



Módulo LED SMD RGB, consiste en un LED capaz de crear cualquier color utilizando RGB, tiene una entrada de voltaje PWM con tres pines. Los colores primarios (rojo / verde / azul) son usados con el fin de lograr el efecto de mezclar colores y obtener cualquier color deseado. Con este módulo se pueden crear efectos de color muy llamativos.

Características:

- Corriente máxima en el LED: 20mA
- Voltaje en cada color: Rojo 1.80V (2.4 max), Verde y Azul 2.8V (3.6V max)
- Voltaje de operación: 5V

Ejemplo:

En este ejemplo el Arduino deberá ser conectado de la siguiente forma:

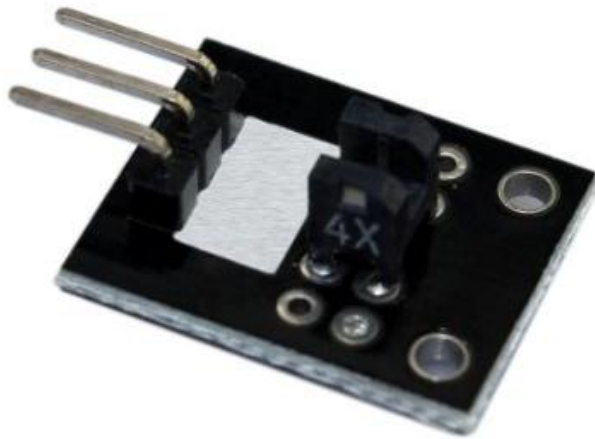
- Arduino pin 9 --> 180 Ohm --> Pin 'R' del módulo KY-009
- Arduino pin 10 --> 100 Ohm --> Pin 'G' del módulo KY-009
- Arduino pin 11 --> 100 Ohm --> Pin 'B' del módulo KY-009
- Arduino GND --> pin '-' del módulo KY-009

Código:

```
int redpin = 11; // select the pin for the red LED
int bluepin = 10; // select the pin for the blue LED
int greenpin = 9; // select the pin for the green LED
int val=0;
void setup () {
  pinMode (redpin, OUTPUT);
  pinMode (bluepin, OUTPUT);
  pinMode (greenpin, OUTPUT);
  Serial.begin (9600);
}

void loop () {
  for (val=255; val>0; val--)
  {
    analogWrite (11, val);
    analogWrite (10, 255-val);
    analogWrite (9, 128-val);
    delay (1);
  }
  for (val = 0; val <255; val++)
  {
    analogWrite (11, val);
    analogWrite (10, 255-val);
    analogWrite (9, 128-val);
    delay (1);
  }
  Serial.println (val, DEC);
}
```

Arduino KY-010: Modulo Sensor Fotoeléctrico Tipo U



Modulo foto-interruptor que funciona con Arduino. Si el sensor es accionado entonces el LED se enciende.

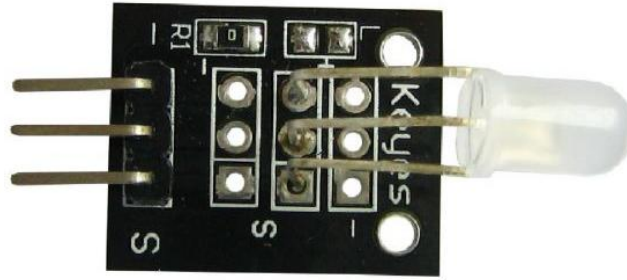
El modulo debe conectarse de la siguiente manera con el Arduino:

- Pin izquierdo: GND
- Pin medio: +5V
- Pin derecho: Señal
 - Pin GND del Arduino con GND del modulo
 - Pin 3 del Arduino con Pin de señal del modulo
 - Pin 5V del Arduino con el Pin 5V del modulo

Código:

```
int Led = 13 ;// define LED Interface
int buttonpin = 3; // define the photo interrupter sensor interface
int val ;// define numeric variables val
void setup (){
  pinMode (Led, OUTPUT) ;// define LED as output interface
  pinMode (buttonpin, INPUT) ;// define the photo interrupter sensor
  output interface
}
void loop ()
{
  val = digitalRead (buttonpin) ;// digital interface will be assigned
  a value of 3 to read val
  if (val == HIGH) // When the light sensor detects a signal is
  interrupted, LED flashes
  {
    digitalWrite (Led, HIGH);
  }
  else
  {
    digitalWrite (Led, LOW);
  }
}
```

Arduino KY-011: Modulo LED bicolor



- Colores: Verde + Rojo
- Diámetro: 3mm
- Tipo: Difusión
- Voltaje (V) :2.0-2.5
- Corriente (mA): 10
- Angulo de visión: 150
- Longitud de onda (NM): 571 +644
- Intensidad luminosa (MCD) :20-40; 40-80

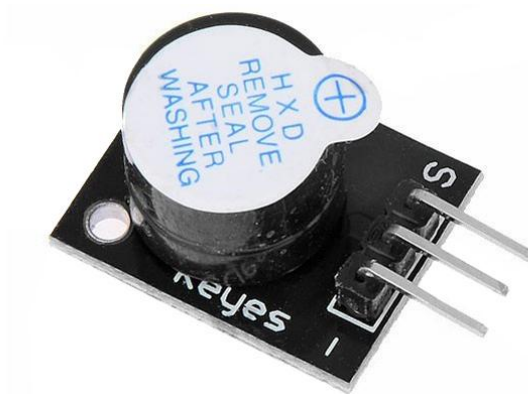
Conexión:

- Arduino pin 10 --> resistencia de 330 Ohm --> Pin de señal del módulo.
- Arduino pin 11 --> resistencia de 330 Ohm --> Pin medio del módulo.
- Arduino GND --> Pin GND del módulo.

Ejemplo:

```
// Arduino test code for KY011
int redpin = 11; // select the pin for the red LED
int greenpin = 10; // select the pin for the green LED
int val;
void setup () {
  pinMode (redpin, OUTPUT);
  pinMode (greenpin, OUTPUT);
}
void loop () {
  for (val = 255; val > 0; val--)
  {
    analogWrite (greenpin, val);
    analogWrite (redpin, 255-val);
    delay (15);
  }
  for (val = 0; val < 255; val++)
  {
    analogWrite (greenpin, val);
    analogWrite (redpin, 255-val);
    delay (15);
  }
}
```


Arduino KY-012: Modulo Buzzer Activo



Este módulo cuenta con un buzzer activo, el cual emite un sonido justo cuando se le aplica un voltaje, su desventaja es que no es posible determinar la frecuencia de los sonidos, pero para ello podemos utilizar el buzzer pasivo incluido en este KIT.

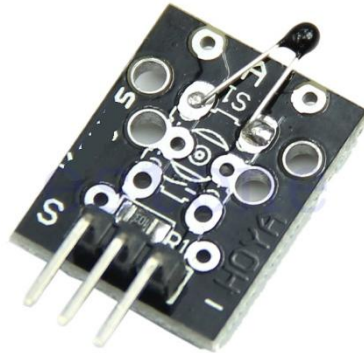
Conexión:

- Pin 8 del Arduino --> Pin de Señal del módulo.
- Pin GND del Arduino --> Pin GND del módulo.
- Pin 5V del Arduino --> Pin 5V del módulo.

Código:

```
//Example code KY012 active buzzer
int speakerPin = 8;
void setup () {
  pinMode (speakerPin, OUTPUT);
}
void loop () {
  analogWrite (speakerPin, 255);
  delay (50);
  analogWrite (speakerPin, 0);
  delay (10);
}
```

Arduino KY-013: Modulo Sensor de Temperatura



El modulo contiene un termistor (resistencia que varía según la temperatura del ambiente), censa en tiempo real la temperatura y varia sus valores internos, conectando al Arduino en sus pines analógicos es posible realizar un gran número de implementaciones, por ejemplo, alamar térmicas.

- Temperatura de medición: $-55^{\circ}\text{C} \sim +125^{\circ}\text{C}$
- Precisión: $\pm 0.5^{\circ}\text{C}$

Conexión:

- Pin analógico 5 del Arduino--> Pin de señal del módulo.
- Pin GND del Arduino --> Pin GND del módulo.
- Pin 5V del Arduino --> Pin intermedio del módulo (5V)

Código:

```
#include <math.h>
int sensorPin = A5; // select the input pin for the potentiometer

double Thermistor(int RawADC) {
    double Temp;
    Temp = log(10000.0*((1024.0/RawADC-1)));
    Temp = 1 / (0.001129148 + (0.000234125 + (0.0000000876741 * Temp *
Temp ))* Temp );
    Temp = Temp - 273.15;           // Convert Kelvin to Celcius
    //Temp = (Temp * 9.0)/ 5.0 + 32.0; // Convert Celcius to Fahrenheit
    return Temp;
}

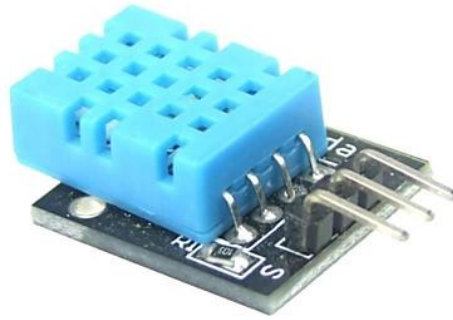
void setup() {
    Serial.begin(9600);
}

void loop() {
    int readVal=analogRead(sensorPin);
    double temp = Thermistor(readVal);

    Serial.println(temp); // display tempature
    //Serial.println(readVal); // display tempature

    delay(500);
}
```

Arduino KY-015: Módulo Sensor de Temperatura y Humedad



Este módulo cuenta con un sensor de temperatura y humedad DHT11, el cual, envía una señal digital a su salida. Tiene un tamaño ultra compacto, bajo consumo de energía, señal de transmisión a distancia de hasta 20 metros, por lo que es de gran utilizada en un gran número de aplicaciones básicas e incluso en aplicaciones más exigentes y robustas.

Características:

- Voltaje de Suministro: 3.3 ~ 5.5V DC
- Salida: Señal Digital.
- Rango de Medición: Humedad 20-90% RH, Temperatura 0 ~ 50 °C.
- Precisión: Humedad + -5% RH, Temperatura + -2 °C.
- Resolución: Humedad 1% RH, Temperatura 1 °C.
- Estabilidad a Largo Plazo: $\pm 1\% \text{ RH} / \text{Año}$.

Ejemplo:

Conexión:

- Arduino pin 8 --> Pin Señal del módulo.
- Arduino GND --> Pin GND del módulo.
- Arduino +5V --> Pin Medio del módulo.

Código:

```
//KY015 DHT11 Temperature and humidity sensor
int DHpin = 8;
byte dat [5];
byte read_data () {
  byte data;
  for (int i = 0; i < 8; i ++ ) {
    if (digitalRead (DHpin) == LOW) {
      while (digitalRead (DHpin) == LOW); // wait for 50us
      delayMicroseconds (30); // determine the duration of the high
level to determine the data is '0 'or '1'
      if (digitalRead (DHpin) == HIGH)
        data |= (1 << (7-i)); // high front and low in the post
```

```

        while (digitalRead (DHpin) == HIGH); // data '1 ', wait for the
next one receiver
    }
}
return data;
}

void start_test () {
    digitalWrite (DHpin, LOW); // bus down, send start signal
    delay (30); // delay greater than 18ms, so DHT11 start signal can be
detected

    digitalWrite (DHpin, HIGH);
    delayMicroseconds (40); // Wait for DHT11 response

    pinMode (DHpin, INPUT);
    while (digitalRead (DHpin) == HIGH);
    delayMicroseconds (80); // DHT11 response, pulled the bus 80us
    if (digitalRead (DHpin) == LOW);
    delayMicroseconds (80); // DHT11 80us after the bus pulled to start
sending data

    for (int i = 0; i < 4; i ++) // receive temperature and humidity
data, the parity bit is not considered
        dat[i] = read_data ();

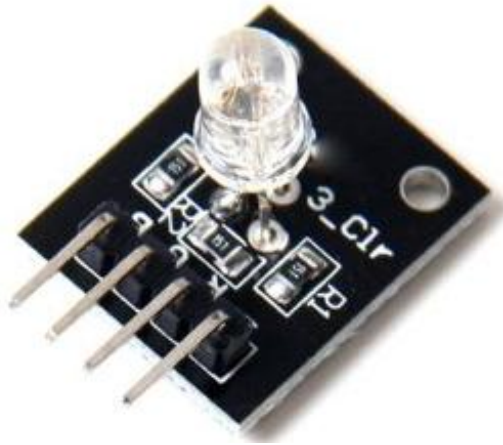
    pinMode (DHpin, OUTPUT);
    digitalWrite (DHpin, HIGH); // send data once after releasing the
bus, wait for the host to open the next Start signal
}

void setup () {
    Serial.begin (9600);
    pinMode (DHpin, OUTPUT);
}

void loop () {
    start_test ();
    Serial.print ("Current humdity =");
    Serial.print (dat [0], DEC); // display the humidity-bit integer;
    Serial.print ('.');
    Serial.print (dat [1], DEC); // display the humidity decimal places;
    Serial.println ('%');
    Serial.print ("Current temperature =");
    Serial.print (dat [2], DEC); // display the temperature of integer
bits;
    Serial.print ('.');
    Serial.print (dat [3], DEC); // display the temperature of decimal
places;
    Serial.println ('C');
    delay (700);
}

```

Arduino KY-016: Modulo LED tricolor



Este módulo consta de un LED capaz de encender en cualquier color por medio de la mezcla de los colores del sistema RGB (rojo, verde y azul), esto con el fin de lograr muchas combinaciones posibles y lograr efectos muy colorido y llamativos.

Características:

- Contiene una resistencia previa al LED RGB para prevenir posibles accidentes.
- Utilizando PWM es posible modificar los valores para obtener las diferentes combinaciones de color.
- Voltaje de funcionamiento: 5V.
- Tipo de LED: Cátodo común.

Conexión:

- Arduino pin 11 --> Pin R del módulo.
- Arduino pin 10 --> Pin G del módulo.
- Arduino pin 9 --> Pin B del módulo.
- Arduino pin GND --> Pin GND del módulo.

No es necesario conectar ninguna resistencia, el modulo incluye las resistencias necesarias.

Código:

```
//KY016 3-color LED module
int redpin = 11; // select the pin for the red LED
int bluepin = 10; // select the pin for the blue LED
int greenpin = 9 ;// select the pin for the green LED
int val;
void setup () {
  pinMode (redpin, OUTPUT);
  pinMode (bluepin, OUTPUT);
  pinMode (greenpin, OUTPUT);
  Serial.begin (9600);
}
void loop ()
{
  for (val = 255; val > 0; val --)
  {
    analogWrite (11, val);
    analogWrite (10, 255-val);
    analogWrite (9, 128-val);
    delay (10);
    Serial.println (val, DEC);
  }
  for (val = 0; val < 255; val ++ )
  {
    analogWrite (11, val);
    analogWrite (10, 255-val);
    analogWrite (9, 128-val);
    delay (10);
    Serial.println (val, DEC);
  }
}
```

Arduino KY-017: Modulo Óptico de Mercurio



El modulo óptico de mercurio es utilizado para detectar inclinación y encendiendo un LED informa al usuario, este sensor es el precursor de los ahora llamados acelerómetros utilizados en los dispositivos móviles como celulares inteligentes y tabletas.

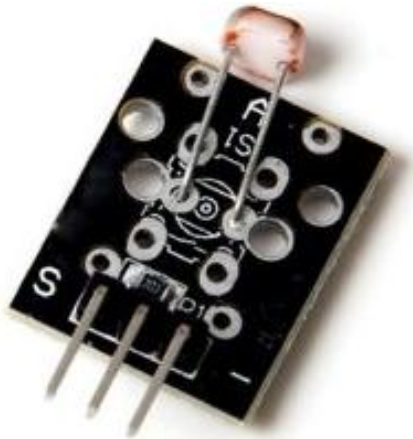
Conexión:

- Arduino GND --> Pin GND del módulo.
- Arduino +5V --> Pin intermedio del módulo.
- Arduino pin 3 --> Pin de señal del módulo.

Código:

```
/KY017 Mercury open optical module
int Led = 13 ;// define LED Interface
int buttonpin = 3; // define the mercury tilt switch sensor interface
int val ;// define numeric variables val
void setup ()
{
  pinMode (Led, OUTPUT) ;// define LED as output interface
  pinMode (buttonpin, INPUT) ;// define the mercury tilt switch sensor
  output interface
}
void loop ()
{
  val = digitalRead (buttonpin) ;// read the values assigned to the
  digital interface 3 val
  if (val == HIGH) // When the mercury tilt switch sensor detects a
  signal, LED flashes
  {
    digitalWrite (Led, HIGH);
  }
  else
  {
    digitalWrite (Led, LOW);
  }
}
```

Arduino KY-018: Modulo con fotorresistencia



Este módulo contiene una fotorresistencia, la cual es una resistencia variable dependiente de la cantidad de luz en su entorno. En la oscuridad, su resistencia es muy alta, a veces hasta 1 M Ω , pero cuando el sensor LDR se expone a la luz, la resistencia se reduce drásticamente, incluso a unos pocos ohm, dependiendo de la intensidad de la luz.

El sensor LDR es ampliamente utilizado en cámaras, lámparas de jardín y calle, detectores, relojes, luces automáticas y un sinfín de aplicaciones interesantes.

Los valores de su resistencia, sensibilidad, coeficiente de temperatura y su curva de voltaje-corriente dependen directamente de la cantidad de luz que recibe el sensor.

Conexión:

- Arduino pin 5 --> Pin de señal del módulo.
- Arduino 5V --> Pin medio del módulo.
- Arduino GND --> Pin GND del módulo (-).

Código:

```
/KY018 Photo resistor module
int sensorPin = A5; // select the input pin for the potentiometer
int ledPin = 13; // select the pin for the LED
int sensorValue = 0; // variable to store the value coming from the
sensor
void setup() {
  pinMode(ledPin, OUTPUT);
  Serial.begin(9600);
}
void loop() {
  sensorValue = analogRead(sensorPin);
  digitalWrite(ledPin, HIGH);
  delay(sensorValue);
  digitalWrite(ledPin, LOW);
  delay(sensorValue);
  Serial.println(sensorValue, DEC);
}
```


Arduino KY-019: Modulo con Relevador



Con este módulo, el cual tiene adicionado un relevador podrás conectar circuitos de 240V AC o 28V DC. Los relevadores son utilizados como sistemas de control y swicheo en alarmas, juguetes y muchas otras aplicaciones. Comúnmente son usados en circuitos de control automático con una pequeña corriente de control y una gran corriente de operación.

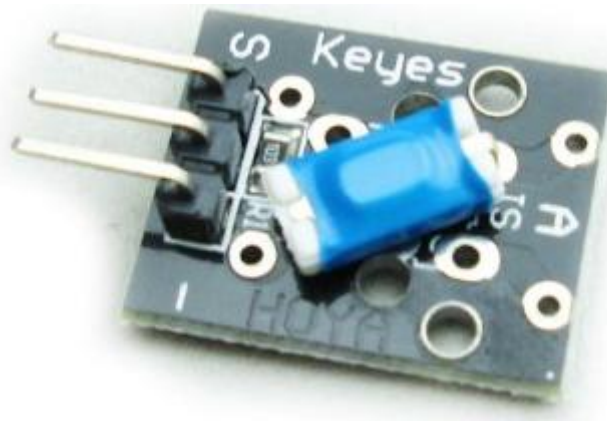
Conexión:

- Arduino pin digital 10 --> Pin de señal del módulo.
- Arduino GND --> Pin GND (-) del módulo.
- Arduino +5V --> Pin medio (+) del módulo.

Código:

```
//KY019 5V relay module
int relay = 10; // relay turns trigger signal - active high;
void setup ()
{
  pinMode (relay, OUTPUT); // Define port attribute is output;
}
void loop ()
{
  digitalWrite (relay, HIGH); // relay conduction;
  delay (1000);
  digitalWrite (relay, LOW); // relay switch is turned off;
  delay (1000);
}
```

Arduino KY-020: Modulo Interruptor de Inclinación



Módulo interruptor de inclinación, este sensor envía un pulso al detectar una inclinación. Se mostrara un código ejemplo en el cual será necesario construir un circuito simple que contara de un LED, dicho LED encenderá al percibirse una inclinación por parte del sensor, de no detectar nada, el LED permanecerá apagado.

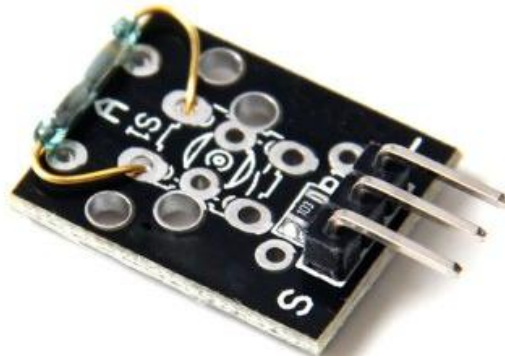
Conexión:

- Pin 3 del Arduino -> Pin de señal del módulo.
- Pin GND del Arduino -> Pin GND (-) del módulo.
- Pin +5V del Arduino -> Pin medio del módulo.

Código:

```
int Led = 13 ;// define LED Interface
int buttonpin = 3; // define the tilt switch sensor interfaces
int val ;// define numeric variables val
void setup ()
{
  pinMode (Led, OUTPUT) ;// define LED as output interface
  pinMode (buttonpin, INPUT) ;//define the output interface tilt
  switch sensor
}
void loop ()
{
  val = digitalRead (buttonpin) ;// digital interface will be assigned
  a value of 3 to read val
  if (val == HIGH) //When the tilt sensor detects a signal when the
  switch, LED flashes
  {
    digitalWrite (Led, HIGH);
  }
  else
  {
    digitalWrite (Led, LOW);
  }
}
```

Arduino KY-021: Modulo de Contacto con Mini Lámina Magnética



Una lámina magnética es un sensor que se encuentra comúnmente abierto y al detectar un campo magnético se cierra, permitiendo el paso del voltaje.

Para el siguiente ejemplo vamos a encender y apagar un LED conectado al pin 13, necesitaremos un imán cerca del módulo para generar un campo magnético.

El modulo incluye una resistencia de 10 k Ω y no es necesario adicionar ninguna otro resistencia.

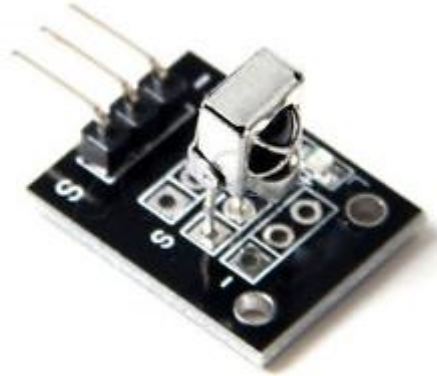
Conexión:

- Arduino GND --> Pin GND (-) del módulo.
- Arduino +5V --> Pin intermedio del módulo.
- Arduino pin digital 3 --> Pin de señal del módulo.

Código:

```
// Example code for sensor KY021
int Led = 13 ;// define LED Interface
int buttonpin = 3; // define the Reed sensor interfaces
int val ;// define numeric variables val
void setup ()
{
  pinMode (Led, OUTPUT) ;// define LED as output interface
  pinMode (buttonpin, INPUT) ;// output interface as defined Reed
  sensor
}
void loop ()
{
  val = digitalRead (buttonpin) ;// digital interface will be assigned
  a value of 3 to read val
  if (val == HIGH) // When the Reed sensor detects a signal, LED
  flashes
  {
    digitalWrite (Led, HIGH);
  }
  else
  {
    digitalWrite (Led, LOW);
  }
}
```

Arduino KY-022: Modulo Sensor Receptor Infrarrojo



Este es el nuevo y ultra delgado 38k infrarrojo para control remoto universal, utilizado en la codificación NEC, principalmente en vehículos con MP3, marco de fotos digital, iluminación equipada y otras aplicaciones. Debido a que es no lineal es de fácil uso y aprendizaje de parte del usuario.

Características:

- Distancia del control remoto infrarrojo: Más de 8 metros.
- Longitud de onda infrarroja: 940Nm.
- Frecuencia del cristal: 455 kHz.
- Frecuencia portadora: 38 kHz.
- Codificación: Formato de codificación de NEC.
- Tamaño: 86 * 40 * 6mm.
- Potencia: CR2025/1600mAH.

Recuerde que el control remoto y el receptor IR son un conjunto, tiene que descifrar el código y mostrarlo en hexadecimal.

Código:

```
#include <IRremote.h>

int RECV_PIN = 11;

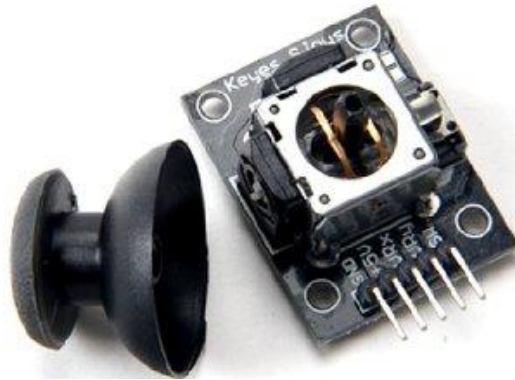
IRrecv irrecv(RECV_PIN);

decode_results results;

void setup()
{
  Serial.begin(9600);
  irrecv.enableIRIn(); // Start the receiver
}

void loop() {
  if (irrecv.decode(&results)) {
    Serial.println(results.value, HEX);
    irrecv.resume(); // Receive the next value
  }
}
```

Arduino KY-023: Modulo Joystick de Eje-XY



Joystick similar al incluido en el juego PS2, el joystick por un potenciómetro en cada eje que es variado con los movimientos, además, incluye un botón Z que es activado al presionar el joystick.

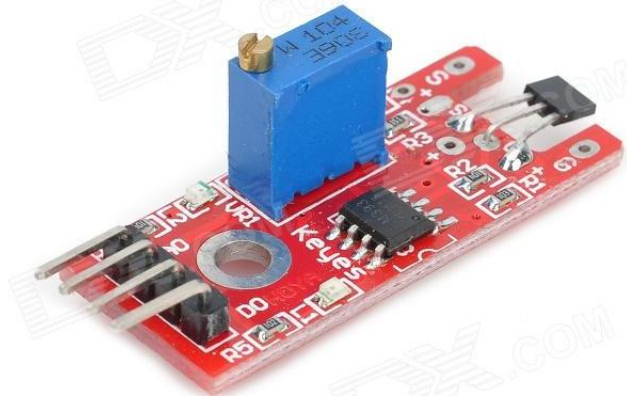
Conexión:

- Arduino GND --> Pin GND (-) del módulo.
- Arduino +5V --> Pin +5V del módulo.
- Arduino pin analógico 0 --> VRx del módulo.
- Arduino pin analógico 1 --> VRy del módulo.
- Arduino pin digital 3--> Pin SW del módulo.

Código:

```
// Module KY023
axis_joystick_module
int JoyStick_X = A0; // x
int JoyStick_Y = A1; // y
int JoyStick_Z = 3; // key
void setup ()
{
  pinMode (JoyStick_X, INPUT);
  pinMode (JoyStick_Y, INPUT);
  pinMode (JoyStick_Z, INPUT);
  Serial.begin (9600); // 9600 bps
}
void loop ()
{
  int x, y, z;
  x = analogRead (JoyStick_X);
  y = analogRead (JoyStick_Y);
  z = digitalRead (JoyStick_Z);
  Serial.print (x, DEC);
  Serial.print (",");
  Serial.print (y, DEC);
  Serial.print (",");
  Serial.println (z, DEC);
  delay (100);
}
```

Arduino KY-024: Modulo Sensor de Campo Magnético Lineal



Este módulo cuenta con un sensor de campo magnético y un LED el cual deberá conectarse al pin 13 del Arduino, al detectar un campo el LED se encenderá y de esta manera generara una alerta. De no detectar ningún campo el LED permanecerá apagado.

Código:

```
int Led = 13 ;// define LED Interface
int buttonpin = 3; // define the linear Hall magnetic sensor
interface
int val ;// define numeric variables val
void setup ()
{
pinMode (Led, OUTPUT) ;// define LED as output interface
pinMode (buttonpin, INPUT) ;// define linear Hall magnetic sensor
output interface
}
void loop ()
{
val = digitalRead (buttonpin) ;// digital interface will be assigned
a value of 3 to read val
if (val == HIGH) // When the linear Hall sensor detects a magnetic
signal, LED flashes
{
digitalWrite (Led, HIGH);
}
else
{
digitalWrite (Led, LOW);
}
}
```

Arduino KY-025: Modulo Sensor de Reflexión

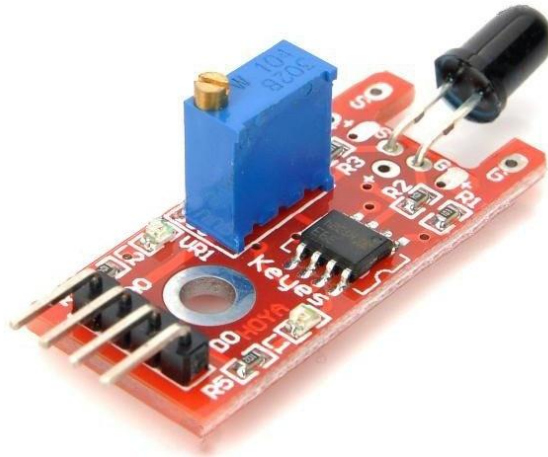


Este módulo cuenta con una lámina detectora de campos magnéticos (Reed Switch) y un LED el cual deberá conectarse al pin 13 del Arduino, al detectar un campo el LED se encenderá y de esta manera generara una alerta. De no detectar ningún campo el LED permanecerá apagado.

Código:

```
int Led = 13 ;// define LED Interface
int buttonpin = 3; // define the Reed sensor interfaces
int val ;// define numeric variables val
void setup ()
{
  pinMode (Led, OUTPUT) ;// define LED as output interface
  pinMode (buttonpin, INPUT) ;// output interface as defined Reed
  sensor
}
void loop ()
SunFounder{
  val = digitalRead (buttonpin) ;// digital interface will be assigned
  a value of 3 to read val
  if (val == HIGH) // When the Reed sensor detects a signal, LED
  flashes
  {
    digitalWrite (Led, HIGH);
  }
  else
  {
    digitalWrite (Led, LOW);
  }
}
```

Arduino KY-026: Modulo Sensor de Llamas



El sensor funciona entre las longitudes de onda infrarroja 760 nm a 1100 nm, es más sensible utilizando dos salidas a 60 grados. Salida analógica: señal analógica de tensión conectada a una resistencia térmica. Señal digital: cuando la temperatura alcanza ciertos umbrales la salida del sensor sube y baja en conjunto con la temperatura, esta señal a la salida es variable con el potenciómetro.

Conexión:

- Arduino pin GND --> Pin G del módulo.
- Arduino pin +5V --> Pin “+” del módulo.
- Arduino pin digital 4 --> Pin “D0” del módulo.
- Arduino pin analógico 3 --> Pin “A0” del módulo.

Código:

```
/Example for KY-026
//TkkrLab
int Led = 13 ;// define LED Interface
int buttonpin = 3; // define the flame sensor interface
int analog = A3; // define the flame sensor interface

int val ;// define numeric variables val
float sensor; //read analog value

void setup ()
{
  pinMode (Led, OUTPUT) ;// define LED as output interface
  pinMode (buttonpin, INPUT) ;// output interface defines the flame
  sensor
  pinMode (analog, INPUT) ;// output interface defines the flame
  sensor
  Serial.begin(9600);
}
```



```
void loop ()
{
  sensor = analogRead(analog);
  Serial.println(sensor); // display tempature

  val = digitalRead (buttonpin) ;// digital interface will be assigned
a value of 3 to read val
  if (val == HIGH) // When the flame sensor detects a signal, LED
flashes
  {
    digitalWrite (Led, HIGH);
  }
  else
  {
    digitalWrite (Led, LOW);
  }
  delay(1000);
}
```

Arduino KY-027: Modulo Taza Mágica de Luz



Este módulo tiene dos partes - un LED y un interruptor de inclinación de mercurio.

Conexión:

- Pin +5V Arduino -> Pin +5V del módulo.
- Pin GND Arduino -> Pin GND (G) del módulo.
- Pines 5 y 6 Arduino -> Pines de señal del módulo.

Al realizar la conexión mostrada con anterioridad se alternara entre un pulso en alto (5V) y uno bajo (0V) al detectar una inclinación. Si se conecta una resistencia limitadora de corriente, se puede utilizar como un LED parpadeante.

Código Ejemplo:

```
int LedPinA = 5;
int LedPinB = 6;
int ButtonPinA = 7;
int ButtonPinB = 4;
int buttonStateA = 0;
int buttonStateB = 0;
int brightness = 0;
void setup () {
  pinMode (LedPinA, OUTPUT);
  pinMode (LedPinB, OUTPUT);
  pinMode (ButtonPinA, INPUT);
  pinMode (ButtonPinB, INPUT);
}
void loop () {
  buttonStateA = digitalRead (ButtonPinA);
  if (buttonStateA == HIGH && brightness != 255)
  {
    brightness ++;
  }
  buttonStateB = digitalRead (ButtonPinB);
  if (buttonStateB == HIGH && brightness != 0)
  {
    brightness --;
  }
  analogWrite (LedPinA, brightness);
  analogWrite (LedPinB, 255 - brightness);
  Delay (25);
}
```

Arduino KY-028: Modulo Sensor de Temperatura Digital



Módulo de temperatura digital, cuenta con un LED que se conecta al PIN 13 del Arduino. El sensor de temperatura digital es activado al detectar una variación en la temperatura, cuando esto sucede el LED se enciende, de lo contrario el LED permanecerá apagado.

Código Ejemplo:

```
int Led = 13 ;// define LED Interface
int buttonpin = 3; // define the digital temperature sensor interface
int val ;// define numeric variables val
void setup ()
{
  pinMode (Led, OUTPUT) ;// define LED as output interface
  pinMode (buttonpin, INPUT) ;// define digital temperature sensor
  output interface
}
void loop ()
{
  val = digitalRead (buttonpin) ;// digital interface will be assigned
  a value of 3 to read val
  if (val == HIGH) // when the digital temperature sensor detects a
  signal, LED flashes
  {
    digitalWrite (Led, HIGH);
  }
  else
  {
    digitalWrite (Led, LOW);
  }
}
```

Arduino KY-029: Modulo LED bicolor3mm



Características:

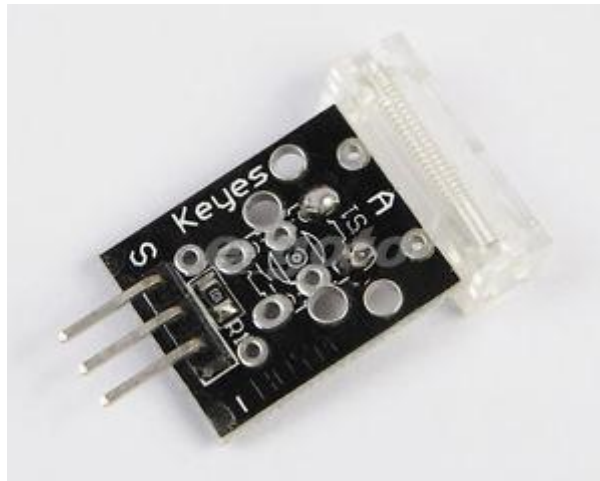
- Colores: Verde y Rojo.
- Diámetro: 5mm
- Tipo: Difusión.
- Voltaje color Verde: 2.3 V ~ 2.6 V.
- Voltaje color Rojo: 1.9 V ~ 2.2 V.
- Corriente: 20 mA
- Longitud de Onda (NM): 571 +625
- Intensidad luminosa (MCD):20-40 (verde); 60-80 (rojo).

Este tipo de LED es ampliamente utilizado en diccionarios electrónicos, PDA's, MP3, auriculares, cámaras digitales, VCD, DVD, audio de coches, comunicaciones, computadoras, cargadores, amplificadores de potencia, instrumentación, regalos, juguetes electrónicos, teléfonos móviles, y muchos otros campos.

Código Ejemplo:

```
int redpin = 11; // select the pin for the red LED
int bluepin = 10; // select the pin for the blueLED
int val;
void setup () {
  pinMode (redpin, OUTPUT);
  pinMode (bluepin, OUTPUT);
  Serial.begin (9600);
}
void loop (){
  for (val = 255; val > 0; val -)
  {
    analogWrite (11, val);
    analogWrite (10, 255-val);
    delay (15);
  }
  for (val = 0; val <255; val +)
  {
    analogWrite (11, val);
    analogWrite (10, 255-val);
    delay (15);
  }
  Serial.println (val, DEC);}
```

Arduino KY-031: Modulo Sensor de Golpe



El modulo sensor de golpe debe ser conectado a un circuito muy simple que constara de un LED conectado al PIN 13 del Arduino, el cual, parpadeara en el momento que el modulo detecte un golpe, de lo contrario el LED permanecerá encendido.

Código Ejemplo:

```
int Led = 13 ;// // define LED Interface
int Shock = 3 // // define the percussion Sensor Interface
int val ;// // define numeric variables val
void setup ()
{
  pinMode (Led, OUTPUT) ;// // define LED as output interface
  pinMode (Shock, INPUT) ;// // define knock sensor output interface
}
void loop ()
{
  val = digitalRead (Shock) ;// // read digital interface is assigned a
  value of 3 val
  if (val == HIGH) // // When the percussion when the sensor detects a
  signal, LED flashes
  {
    digitalWrite (Led, LOW);
  }
  else
  {
    digitalWrite (Led, HIGH);
  }
}
</syntaxhighlight lang="C">
```

Arduino KY-032: Modulo Sensor de Evasión de Obstáculos.



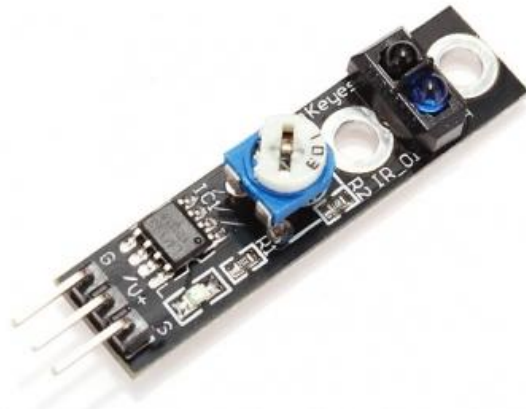
Sensor de evitación de obstáculos por infrarrojo, está diseñado para la elaboración robots de ruedas evasores de obstáculos. Este sensor de luz ambiental es adaptable, de alta precisión, que tiene un par de infrarrojos de transmisión y recepción, el infrarrojo transmisor emite una cierta frecuencia, al detectar la dirección de un obstáculo (reflector), el receptor de infrarrojos recibe el reflejo, cuando esto sucede el indicador LED enciende, a través del circuito, la señal digital de salida puede ser ajustada junto con la distancia de detección por medio de los potenciómetros, la distancia efectiva es de 2 ~ 40cm, voltaje de trabajo de 3.3V-5V, el voltaje de funcionamiento es amplio.

- Voltaje de Trabajo: DC 3.3V-5V
- Corriente: $\geq 20\text{mA}$
- Temperatura: $-10\text{ }^{\circ}\text{C} - +50\text{ }^{\circ}\text{C}$
- Distancia de Detección :2-40cm
- Interfaces de entrada/salida: (- / + / S / EN)
- Ajuste: Ajuste de resistencia para sensibilidad de detección.
- Angulo Efectivo: 35 °
- Tamaño: $28\text{mm} \times 23\text{mm}$
- Peso: 9g

Código Ejemplo:

```
int Led = 13 ;// define LED Interface
int buttonpin = 3; // define the obstacle avoidance sensor interface
int val ;// define numeric variables val
void setup ()
{
  pinMode (Led, OUTPUT) ;// define LED as output interface
  pinMode (buttonpin, INPUT) ;// define the obstacle avoidance sensor
  output interface
}
void loop (){
  val = digitalRead (buttonpin) ;// digital interface will be assigned
  a value of 3 to read val
  if (val == HIGH) // When the obstacle avoidance sensor detects a
  signal, LED flashes
  {
    digitalWrite (Led, HIGH);
  }
  else
  {
    digitalWrite (Led, LOW);
  }
}
```

Arduino KY-033: Modulo Sensor de Rastreo

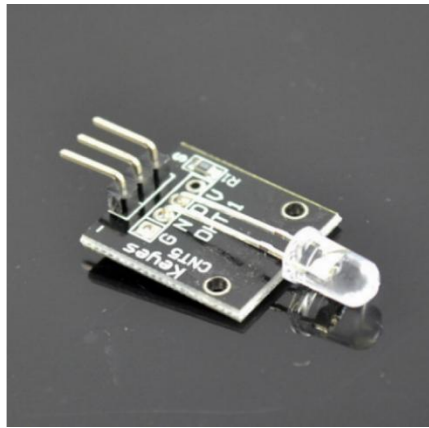


Modulo con sensor magnético, deberá construirse un circuito simple que contara con un LED que debe conectarse al PIN 13 del Arduino, cuando el sensor detecta un destello magnético el LED encenderá, si el sensor no detecta nada el LED permanecerá apagado.

Código Ejemplo:

```
int sensorPin = A5; // select the input pin
int ledPin = 13; // select the pin for the LED
int sensorValue = 0; // variable to store the value coming from the
sensor
void setup () {
  pinMode (ledPin, OUTPUT);
  Serial.begin (9600);
}
void loop () {
  sensorValue = analogRead (sensorPin);
  digitalWrite (ledPin, HIGH);
  delay (sensorValue);
  digitalWrite (ledPin, LOW);
  delay (sensorValue);
  Serial.println (sensorValue, DEC);
}
```

Arduino KY-034: Modulo Flash Automático de 7 colores

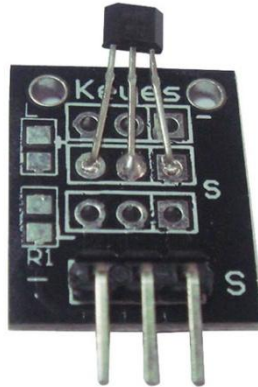


- Tipo: LED
- Modelo: YB-3120B4PnYG-PM
- Forma: LED de 5mm tipo DIP
- Colores: rosa, amarillo, verde (ultra brillante)
- Tipo de Lente: Niebla Blanca
- Voltaje :3.0 ~ 4.5 V

Código Ejemplo:

```
/*
Blink
Turns on an LED on for two second, then off for two second,
repeatedly.
This example code is in the public domain.
*/
void setup () {
  // Initialize the digital pin as an output.
  // Pin 13 has an LED connected on most Arduino boards:
  pinMode (13, OUTPUT);
}
void loop () {
  digitalWrite (13, HIGH); // set the LED on
  delay (2000); // wait for a second
  digitalWrite (13, LOW); // set the LED off
  delay (2000); // wait for a second
}
```


Arduino KY-035: Modulo Sensor Magnético Clase Bihor

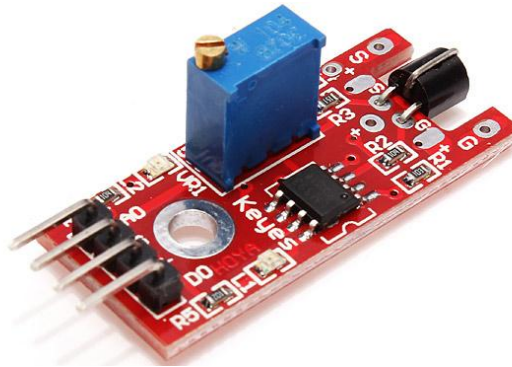


Módulo sensor magnético analógico, se debe construir un circuito simple con un LED para crear un campo magnético Makers. Cuando el sensor detecta un campo magnético el LED conectado deberá encender, de lo contrario el LED permanecerá apagado.

Código Ejemplo:

```
int sensorPin = A5; // select the input pin
int ledPin = 13; // select the pin for the LED
int sensorValue = 0; // variable to store the value coming from the
sensor
void setup () {
  pinMode (ledPin, OUTPUT);
  Serial.begin (9600);
}
void loop () {
  sensorValue = analogRead (sensorPin);
  digitalWrite (ledPin, HIGH);
  delay (sensorValue);
  digitalWrite (ledPin, LOW);
  delay (sensorValue);
  Serial.println (sensorValue, DEC);
}
```

Arduino KY-036: Modulo Sensor de Sonido Metálico



Módulo sensor metálico táctil, cuenta con un LED que deberá conectarse con un simple circuito al pin número 13 del Arduino. Cuando el sensor detecta un el toque de una persona el LED conectado deberá encender, de lo contrario el LED permanecerá apagado.

Código Ejemplo:

```
int Led = 13 ;// define LED Interface
int buttonpin = 3; // define Metal Touch Sensor Interface
int val ;// define numeric variables val
void setup ()
{
  pinMode (Led, OUTPUT) ;// define LED as output interface
  pinMode (buttonpin, INPUT) ;// define metal touch sensor output
  interface
}
void loop ()
{
  val = digitalRead (buttonpin) ;// digital interface will be assigned
  a value of 3 to read val
  if (val == HIGH) // When the metal touch sensor detects a signal, LED
  flashes
  {
    digitalWrite (Led, HIGH);
  }
  else
  {
    digitalWrite (Led, LOW);
  }
}
```

Arduino KY-037: Modulo Sensor de Micrófono Sensible



Para la detección de sonido el módulo tiene dos salidas:

- AO, salida analógica, la señal de tensión de salida en tiempo real del micrófono
- DO, cuando la intensidad del sonido alcanza un cierto umbral, la salida de señal de alta y baja

El umbral de sensibilidad se puede ajustar mediante el potenciómetro en el sensor.

Conexión:

- Pin +5V del Arduino ->Pin 5+ del módulo.
- Pin GND del Arduino ->Pin “-” del módulo.
- Pin A0 del Arduino ->Pin A0 del módulo. (Para Salidas Analógicas)
- Pin D0 del Arduino ->Pin D0 del módulo. (Para Salidas Digitales)

Código Ejemplo Salidas Digitales:

```
int Led = 13 ;// define LED Interface
int buttonpin = 3; // define D0 Sensor Interface
int val = 0;// define numeric variables val

void setup ()
{
  pinMode (Led, OUTPUT) ;// define LED as output interface
  pinMode (buttonpin, INPUT) ;// output interface D0 is defined sensor
}

void loop ()
{
  val = digitalRead(buttonpin);// digital interface will be assigned a
  value of pin 3 to read val
```

```
    if (val == HIGH) // When the sound detection module detects a
signal, LED flashes
    {
        digitalWrite (Led, HIGH);
    }
    else
    {
        digitalWrite (Led, LOW);
    }
}
```

Código Ejemplo Salidas Analógicas:

```
int sensorPin = A0; // select the input pin for the potentiometer
int ledPin = 13; // select the pin for the LED
int sensorValue = 0; // variable to store the value coming from the
sensor

void setup ()
{
    pinMode (ledPin, OUTPUT);
    Serial.begin (9600);
}

void loop ()
{
    sensorValue = analogRead (sensorPin);
    digitalWrite (ledPin, HIGH);
    delay (sensorValue);
    digitalWrite (ledPin, LOW);
    delay (sensorValue);
    Serial.println (sensorValue, DEC);
}
```

Arduino KY-038: Modulo Micrófono con sensor de Sonido



Para la detección de sonido el módulo tiene dos salidas:

- AO, salida analógica, la señal de tensión de salida en tiempo real del micrófono
- DO, cuando la intensidad del sonido alcanza un cierto umbral, la salida de señal de alta y baja

El umbral de sensibilidad se puede ajustar mediante el potenciómetro en el sensor.

Conexión:

- Pin +5V del Arduino ->Pin 5+ del módulo.
- Pin GND del Arduino ->Pin “-” del módulo.
- Pin A0 del Arduino ->Pin A0 del módulo. (Para Salidas Analógicas)
- Pin D0 del Arduino ->Pin D0 del módulo. (Para Salidas Digitales)

Código Ejemplo Salidas Digitales:

```
int Led = 13 ;// define LED Interface
int buttonpin = 3; // define D0 Sensor Interface
int val = 0;// define numeric variables val

void setup ()
{
  pinMode (Led, OUTPUT) ;// define LED as output interface
  pinMode (buttonpin, INPUT) ;// output interface D0 is defined sensor
}

void loop (){
  val = digitalRead(buttonpin);// digital interface will be assigned a
  value of pin 3 to read val
  if (val == HIGH) // When the sound detection module detects a
  signal, LED flashes
  {
    digitalWrite (Led, HIGH);
  }
  else
  {
    digitalWrite (Led, LOW);
  }
}
```

Código Ejemplo Salidas Analógicas:

```
int sensorPin = A0; // select the input pin for the potentiometer
int ledPin = 13; // select the pin for the LED
int sensorValue = 0; // variable to store the value coming from the
sensor

void setup ()
{
  pinMode (ledPin, OUTPUT);
  Serial.begin (9600);
}

void loop ()
{
  sensorValue = analogRead (sensorPin);
  digitalWrite (ledPin, HIGH);
  delay (sensorValue);
  digitalWrite (ledPin, LOW);
  delay (sensorValue);
  Serial.println (sensorValue, DEC);
}
```

Arduino KY-039: Modulo Detector de Latidos



Este módulo utiliza un LED infrarrojo brillante y un fototransistor para detectar el pulso del dedo, el LED enciende en rojo con cada pulso. Monitor de pulso funciona de la siguiente manera: El LED genera la luz por un lado del dedo, y el fototransistor en el otro lado del dedo, el fototransistor es utilizado para obtener el flujo emitido, cuando el pulso de la presión arterial en el dedo varia la resistencia del fototransistor cambia ligeramente.

Conexión:

- Sensor pin S -> Arduino pin Analógico 0 / A0.
- Sensor pin "+"->Arduino pin 5+.
- Sensor pin "-"->Arduino pin GND.

Código Ejemplo:

```
// Pulse Monitor Test Script
int sensorPin = 0;
double alpha = 0.75;
int period = 100;
double change = 0.0;
double minval = 0.0;
void setup ()
{
  Serial.begin (9600);
}
void loop ()
{
  static double oldValue = 0;
  static double oldChange = 0;

  int rawValue = analogRead (sensorPin);
  double value = alpha * oldValue + (1 - alpha) * rawValue;

  Serial.print (rawValue);
  Serial.print (",");
  Serial.println (value);
  oldValue = value;

  delay (period);
}
```

Arduino KY-040: Modulo Codificador Rotatorio



Los codificadores incrementales producen una onda cuadrada de dos fases, la diferencia de fase entre ellos es de 90° , a menudo son llamados como canales A y B. Uno de los canales proporciona la Información de velocidad-, al mismo tiempo, mediante hace una comparación de forma secuencial de señales en los dos canales. Hay también una señal especial llamada Z o canal cero, lo que da la codificación de posición, la señal es una onda cuadrada que coincide con la onda cuadrada del canal A.

A(Hacia las manecillas del reloj)	B(En contra de las manecillas del reloj)
1	1
0	1
0	0
1	0
1	1
1	0
0	0
0	1

Para saber a cuánto equivale un ángulo eléctrico con un ángulo mecánico de 360 grados se puede calcular con la siguiente fórmula:

$$360^\circ \text{ Eléctricos} = 360^\circ \text{ Mecánicos} / n^\circ \text{ Impulsos/Revoluciones.}$$

Si deseas comprender mejor el funcionamiento de este codificador rotatorio visita el siguiente link para ver un video:

<https://www.youtube.com/watch?v=-mSuN5M4tHs&hd=1>

Código Ejemplo:

```
int redPin = 2;
int yellowPin = 3;
int greenPin = 4;
int aPin = 6;
int bPin = 7;
int buttonPin = 5;
int state = 0;
int longPeriod = 5000; // Time at green or red
int shortPeriod = 700; // Time period when changing
int targetCount = shortPeriod;
int count = 0;
void setup ()
{
  pinMode (aPin, INPUT);
  pinMode (bPin, INPUT);
  pinMode (buttonPin, INPUT);
  pinMode (redPin, OUTPUT);
  pinMode (yellowPin, OUTPUT);
  pinMode (greenPin, OUTPUT);
}
void loop ()
{
  count++;
  if (digitalRead (buttonPin))
  {
    setLights (HIGH, HIGH, HIGH);
  }
  else
  {
    int change = getEncoderTurn ();
    int newPeriod = longPeriod + (change * 1000);
    if (newPeriod >= 1000 && newPeriod <= 10000)
    {
      longPeriod = newPeriod;
    }
    if (count > targetCount)
    {
      setState ();
      count = 0;
    }
  }
  delay (1);
}
int getEncoderTurn ()
{
  // Return -1, 0, or +1
  static int oldA = LOW;
  static int oldB = LOW;
  int result = 0;
  int newA = digitalRead (aPin);
  int newB = digitalRead (bPin);
  if (newA != oldA || newB != oldB)
  {
    //Something has changed
    if (oldA == LOW && newA == HIGH)
    {
      result = - (oldB * 2 - 1);
    }
  }
}
```

```
    oldA = newA;
    oldB = newB;
    return result;
}
int setState ()
{
    if (state == 0)
    {
        setLights (HIGH, LOW, LOW);
        targetCount = longPeriod;
        state = 1;
    }
    else if (state == 1)
    {
        setLights (HIGH, HIGH, LOW);
        targetCount = shortPeriod;
        state = 2;
    }
    else if (state == 2)
    {
        setLights (LOW, LOW, HIGH);
        targetCount = longPeriod;
        state = 3;
    }
    else if (state == 3)
    {
        setLights (LOW, HIGH, LOW);
        targetCount = shortPeriod;
        state = 0;
    }
}
void setLights (int red, int yellow, int green)
{
    digitalWrite (redPin, red);
    digitalWrite (yellowPin, yellow);
    digitalWrite (greenPin, green);
}
```