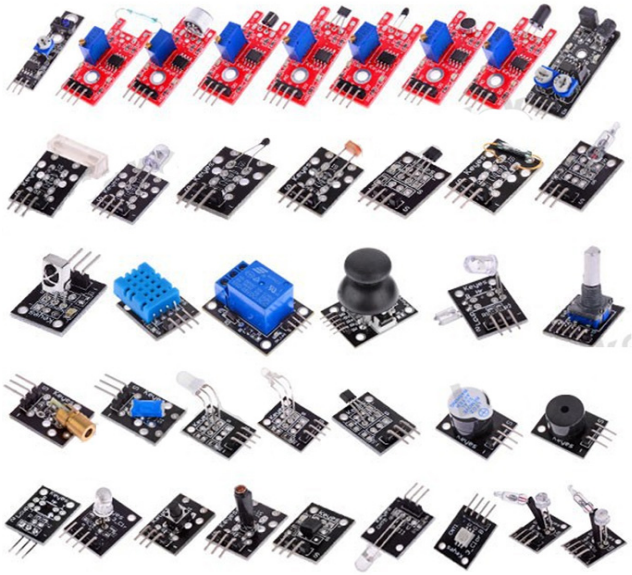
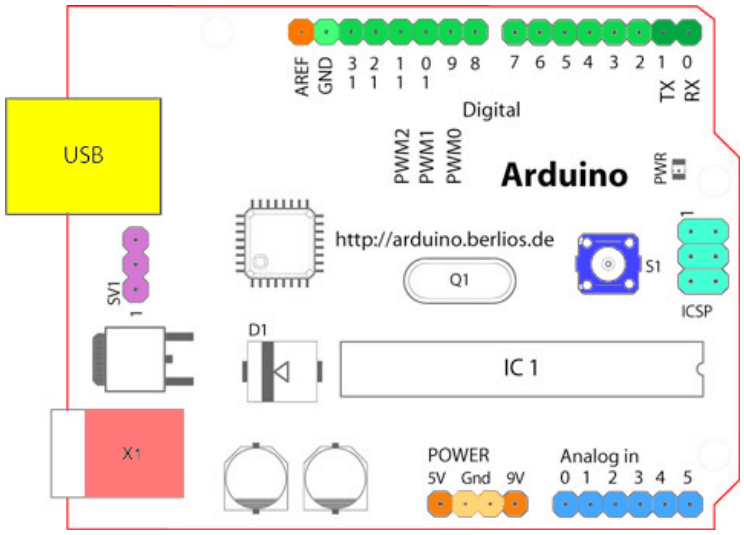
SunFounder

37 in 1 box Sensor Kit For Arduino Starters



Summary:

DIY Maker 37 IN 1 sensor learning package is a highly cost-Learning Arduino sensor package
We carefully build for the beginners， in the whole learning process without welding and wiring, directly by plug 3P universal sensor cable, you can easily go to experience the fun of interactive sensing and electronic technology, in getting started.
Let us into a variety of interactive electronic world. . .



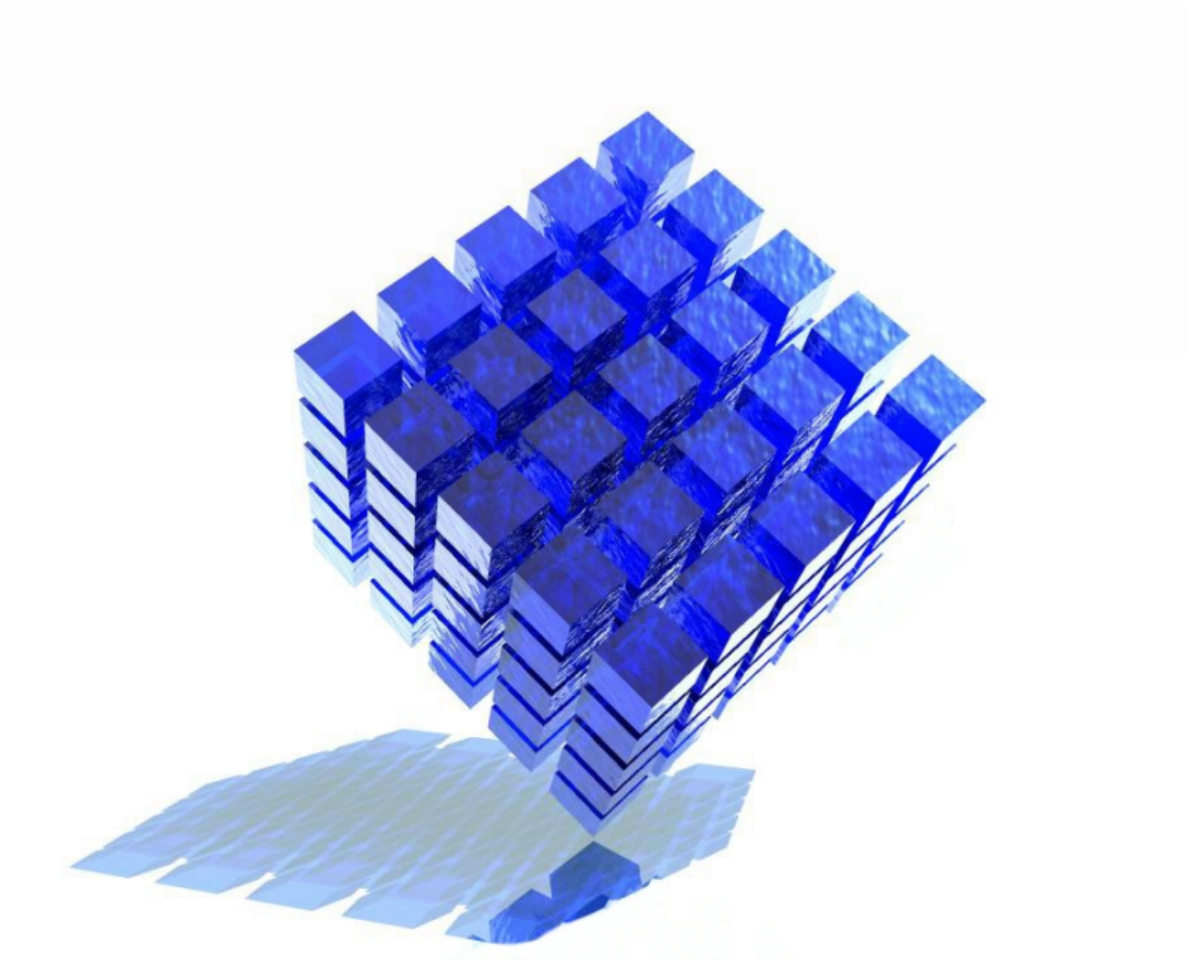
SunFounder

list of kit:

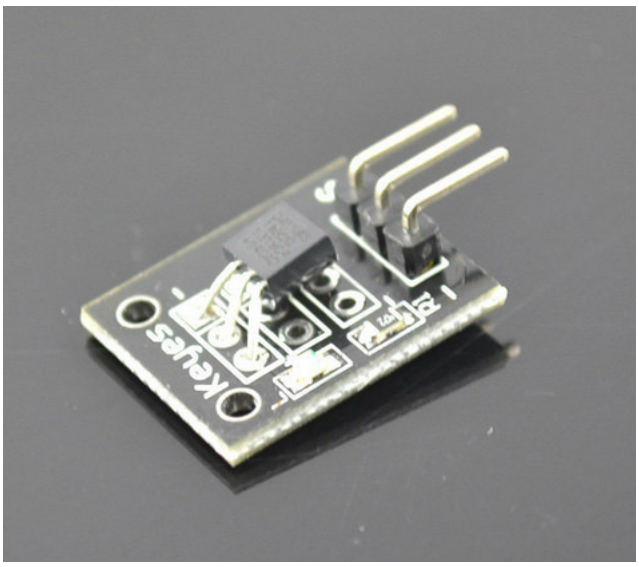
01. DS18B20 sensor module
02. vibration switch module
03. Hall magnetic sensor module
04. key switch module
05 .infrared transmitter module
06 .passive buzzer module
07. laser transmitter module
08. 3 colors - Full Color LED SMD Module
09. photo interrupter module
10. 3 mm red and green LED module (common cathode)
011. active buzzer module
012. analog temperature sensor
013. DHT11 digital temperature and humidity sensor module
014,.3 colors - LED full color module
015. mercury tilt switch module
016. photoresistor module
017. 5V Relay Module
018. tilt switch module
019. MINI reed switch module
020. infrared receiver module
021. PS2 gamepad axis sensor module
022. Linear Hall magnetic module
023. Reed module
024. flame sensor module
025. Magic light cup module
026. Digital Temperature Module
027. 5mm red and green LED module (common cathode)
028. knock sensor module
029. infrared obstacle avoidance sensors
030. 7 color flashing LED module will automatically
031. analog Hall magnetic sensor module
032. Touch Module
033.High sensitivity sound detection module
034. microphone sound detection module
035. finger measuring heartbeat module
036.. Tracking Module
037. rotary encoder modules

SunFounder

List of courses



01 DS18b20 sensor module



I. INTRODUCTION

I believe if we come into contact with this amazing Arduino then the DS18B20 digital thermometerChips are generally not Stranger to it, right! It enables you to keep abreast of the ambient temperature around you, more importantly, as If you have an idea you can also customize Own use Arduino DIY a thermometer, put in your bedroom or in the car,^ - ^ Well, let's work together to uncover the mystery DS18B20, and then you You can use it to make the Kinds of electronic products that match your personality .

二, Products

As in the past the temperature sensor output is analog, we need to add additional A / D And D / A chip into Line conversion, then for Arduino resources are not abundant external interface is a big challenge at the same time Utilization is not high, then we The new DS18B20 Temperature Sensor Module for a good solution to this Issues unique bus line And economic characteristics, fully applicable Arduino platform that allows users to easily set up pass Sensor networks.

Third, the technical parameters

1, the module uses a single-bus digital temperature sensor DS18B20, the external power supply voltage Range is 3.0 V to 5.5 V, No standby power. Measurement temperature range of -55 ° C to +125 ℃, Fahrenheit equivalent 67 ° F to 257 ° F, -10 °C to +85 ° C range accuracy of ± 0.5 ° C.

2, the temperature sensor is a programmable resolution of 9 to 12 temperature conversion to 12-bit digital format With a maximum of 750 milliseconds formula User definable nonvolatile temperature alarm settings.

3, each DS18B20 contains a unique serial number, can be with a plurality ds18b20s Exists in a bus. Temperature sensor can be placed at different places in the detected temperature.

IV Notes

1, because the ordinary transistor DS18B20 and looks similar, we'll be sure to note when using Be careful not to regard it as a generalPass transistor used to avoid damage;

2, in order to prevent damage to the DS18B20 and makes it does not work, we should ensure that the powerLine and ground not reversed.

3, the relevant technical data on the bus did not mention a single number that can be linked to how much DS18B20, But in practical applications are not as many, and we should pay attention to.

4, there is a connection DS18B20 bus length limitations that should be taken in the long-distance communication Consider bus distributed capacitance and resistance Impact resistant.

5, instructions for use Identify DS18B20 Temperature Sensor Module power lines, ground, and data Lines, power lines, ground points connected to the Arduino test board +5 V, GND port number Data bus connected to the digital port.

6, module function test

1, the hardware device

Arduino controller × 1

DS18B20 Temperature Sensor Module × 1

USB data cable × 1

Connect the circuit is very simple, as long as the connected module power supply, ground, and then the module

Data bus

With Arduino digital pin connected to terminal 12 can be, even on the USB data Throughout the test circuit is complete, DS18B20 test results we will use the serial port are displayed on the computer screen. We will test the compiled code downloaded to the board, open the serial port can know that we are What kind of temperatures in the Well, did not talk much, we first look at the test code bar

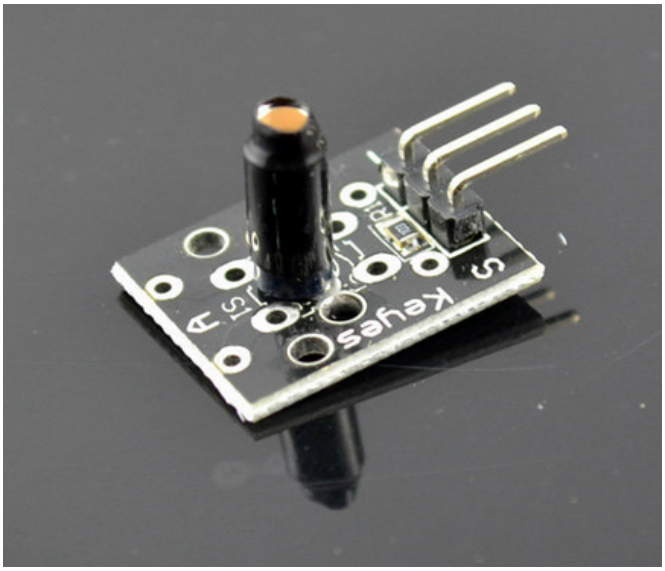
```
# Include <OneWire.h>
/ * DS18S20 Temperature chip i / o
* /
OneWire ds (10); / / on pin 10
void setup (void) {
/ / Initialize inputs / outputs
/ / Start serial port
Serial.begin (9600);
}
void loop (void) {
byte i;
byte present = 0;
byte data [12];
byte addr [8];
int Temp;
if (! ds.search (addr)) {
ÿ / / Serial.print ("No more addresses. \ n");
ÿds.reset_search ();
ÿreturn;
}
Serial.print ("R ="); / / R = 28 Not sure what this is
for (i = 0; i <8; i + +) {
Serial.print (addr [i], HEX);
Serial.print ("");
}
if (OneWire :: crc8 (addr, 7)! = addr [7]) {
ÿSerial.print ("CRC is not valid! \ n");
ÿreturn;
}
if (addr [0]! = 0x28) {
ÿSerial.print ("Device is not a DS18S20 family device. \ n");
ÿreturn;
}
ds.reset ();
```

ds.select (addr);
ds.write (0x44, 1); ÿ / / 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); ÿ / / Read Scratchpad
Serial.print ("P =");
Serial.print (present, HEX);
Serial.print ("");
for (i = 0; i <9; i + +) {ÿ / / we need 9 bytes
data [i] = ds.read ();
When you write the program, download it to arduino inside after running the resulting effect is as follows: In the
When doing experiments here
The temperature was 27 degrees Celsius, touch DS18B20, you can see the serial communication
Significant change in the temperature of the module.
A success! ^ _ ^
VII Conclusion
This section here will come to an end, I believe after reading this section has not DS18B20
And then is so afraid of
It, in fact, as long as we grasp, we can use it arbitrary and designed our personality
Electronic products, their
Try it yourself also. . . . .

Serial.print (data [i], HEX);
Serial.print ("");
}
Temp = (data [1] << 8) + data [0] ;/ / take the two bytes from the response relating to temperature
Temp = Temp >> 4 ;/ / divide by 16 to get pure celcius readout
/ / Next line is Fahrenheit conversion
Temp = Temp * 1.8 +32; / / comment this line out to get celcius
Serial.print ("T =") ;/ / output the temperature to serial port
Serial.print (Temp);
Serial.print ("");
Serial.print ("CRC =");
Serial.print (OneWire :: crc8 (data, 8), HEX);
Serial.println ();
}

****

02 ARDUINO vibration switch module



Vibration module and number 13 comes with interfaces to build a simple circuit LED, producing vibrations flasher.

13 comes with digital interfaces of the LED, the shock sensor access number 3 interface, when a sense of shock sensor

Measure

To a vibration signal, LED flashing light.

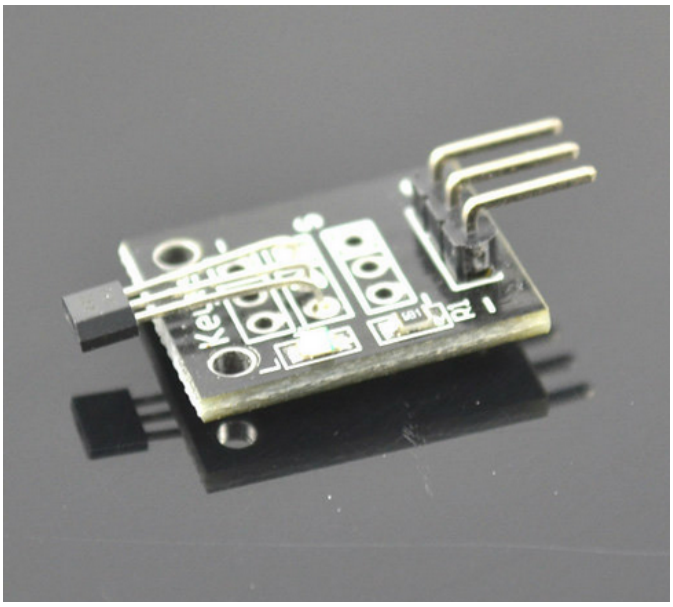
Routines source code:

```
int Led = 13 ;/ / define LED Interface
int Shock = 3 / / 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);
}
```

}

****

03 KEYES Arduino Hall magnetic sensor module



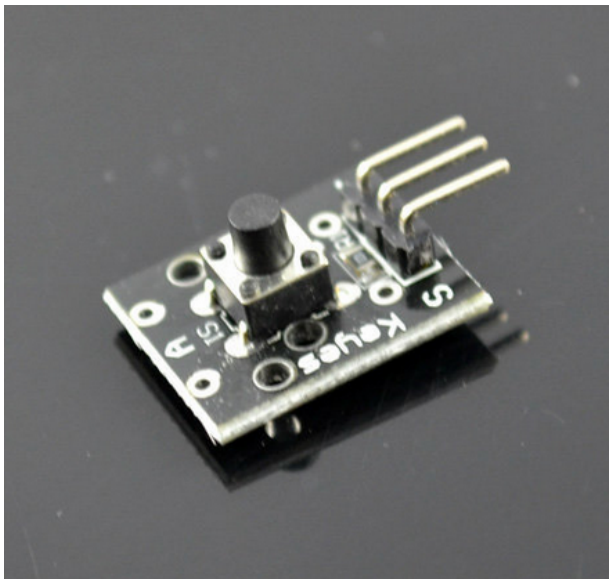
Hall magnetic sensor module and a digital interface, built-in 13 LED build a simple circuit to produce a magnetic flasher.
13 comes with digital interfaces of the LED, the Hall magnetic sensor connected to the force plate number 3 ARDUINO interface
When the Hall magnetic sensor to a magnetic field signal, LED lights, lights off and vice versa.
}
With reference program:

```
int Led = 13 ;/ / define LED Interface
int SENSOR = 3 ;/ / 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);
}
}
```

****

04 KEYES ARDUINO key switch module



Key switch module and a digital interface, built-in 13 LED build a simple circuit to produce key warning lamp

13 comes with digital interfaces of the LED, the access number 3 button switch sensor interfaces, when the key switch

Sensor senses a key signal, LED lights, otherwise off.

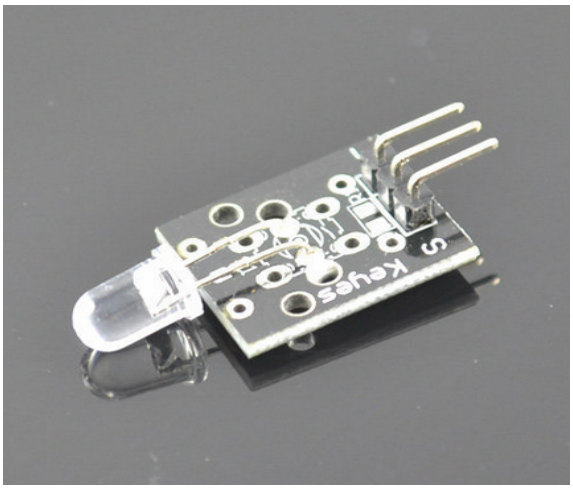
Routines source code:

```
int
Led = 13 ;/ / define LED Interface
int buttonpin = 3 / / 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);
}
}
```

****

05 infrared transmitter module



This is their specific physical map

First, the introduction of This time we want to introduce an infrared transmitter and receiver modules, in fact, they are now in our daily life plays Important role in who is now on a lot of household appliances are used in such a device, such as air conditioning, TV, DVD, etc.,

It is based on wireless sensing, but also can be more remote control, very easy to use, then we today Day it is necessary to study the fundamentals and how to use. Second, Products

1, infrared emission control, also known as infrared emitting diode, it belongs to two Tube can be directly converted into electrical energy near-infrared lightAnd a light emitting device can radiate

out, its structure, with the general principle of the light emitting diode similar, but with a semiconductor
Different materials;
2, the infrared receiver is set to receive, amplify, demodulate one of the devices, which has been completed within the demodulation IC output
Is a digital signal;
3, the principle of infrared communication
Let's look at the structure of the infrared receiver: infrared receiver inside there are two important elements, namely the IC andPD. IC is a receiver processing elements, mainly composed of silicon crystals and circuits, is a highly integrated device, the main
To function as a filter, shaping, decoding, zoom and other functions. Photodiode PD is, the main function is to receive the optical signal Number.
The following is a brief schematic work
The modulated infrared emitting diode signal emitted, infrared receiver after receiving, decoding, filtering, and a series of operations After the signal for recovery;
IV Notes
1, infrared emitting diodes: clean, good condition; various parameters during operation must not exceed limit values (positive
To the current 30 ~ 60 mA, Pulse Forward Current 0.3 ~ 1 A, reverse voltage 5 V, power dissipation 90 mW, the working temperature
Range -25 ~ +80 ℃, storage temperature range of -40 ~ +100 ℃, soldering temperature of 260 ℃) infrared emission tube and then
Closed head should be paired with, otherwise it will affect the sensitivity;
2, the infrared receiver: in a low humidity environment storage and use; Please pay attention to protect the infrared receiver receiving surface,
Contamination or wear will affect reception, and do not touch the surface; Do not wash this; polluting gas in
Body or the sea (salty) environment storage and use; without any external pressure, and affect the quality of the environment
Storage and use;
Five, instructions for use
We first look at the diagram, to understand the infrared transmitter and receiver module specific connection with the Arduino
Note: The above circuit is based on our above that kind protel schematic structures, and meet the specific pin assignment
Shown in the schematic.
Well, the test circuit there, look at the code under test right now
This time we use to two Arduino control board, a main transmitter (Master), one as a slave receiver (Slave),
Own specific set. We can according to the above schematic wiring and fixed infrared transmitter and receiver modules, here I
We can work together to test it.
Six of the test portion
Hardware Requirements

1, Arduino controller × 1
2, USB data cable × 1
3, the infrared transmitter module × 1
3, the infrared receiver module × 1
Here follow the above means to build our test circuit
Well, the whole test code is not long, we understand the code for those specific function of usage, then a
Cut will become simpler, Come!
Another point I must say is: we see the physical map will know, this used a two Arduino
Board, the above code in the download time do to make it clear which is the launch, which was received? The program also
Have noted, if the download is wrong, is not getting the results!
Code download is complete, we open the Serial Monitor window, if you can see the following data show that
It shows you are successful, ^ _ ^
From the receiving part of the code

```
# 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);
}
}
```

The amount of points we can hand to block receiver module, see also the normal communication between them do? The following is the receive window

Ah, looked at the window, and we all know it. . . . .

VII Conclusion

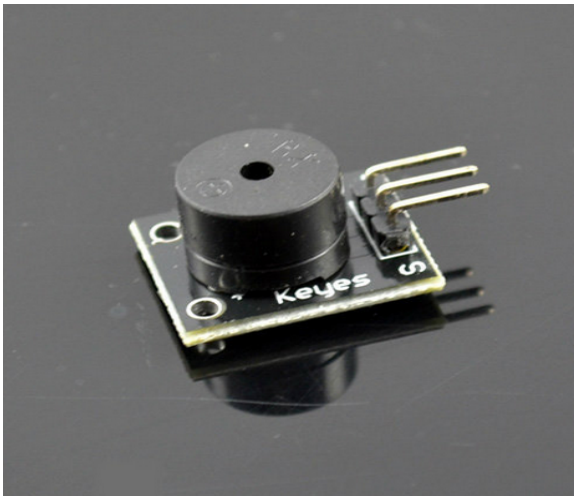
The reason why we feel that infrared is really a wonderful thing, it is because we are invisible, intangible, but

Okay, we do not need that, too, can control it and make it serve us, in fact, we are more magical,

Is not? Oh. . . . . Well, today's introduction on to this, and if you are interested you can contact us, thank you!

****

06 Passive buzzer



Arduino can be done with a lot of interactive work, the most common and most commonly used is the sound and light show in front has been

LED lights are used in the experiments, we let the experiment circuit sound, a voice of the most common

Component is the buzzer and speaker, and comparison of the two buzzer easier and ease the present study, we buzzer.

Buzzer and the principle

(A) the introduction of the buzzer

1. Buzzer Buzzer is an integrated role in the structure of electronic transducers, DC voltage power supply, wide

Pan used in computers, printers, copiers, alarms, electronic toys, automotive electronic equipment, telephones, timers, etc.

Electronic products for sound devices.

2. The classification is divided into buzzer piezo buzzer buzzer and two types of electromagnetic

buzzer.

3. Graphic symbols buzzer buzzer circuit in the circuit by the letter "H" or "HA" (old standard with

"FM", "LB", "JD", etc.) indicates.

(Two) structural principle buzzer

1. Piezo Buzzer Piezo Buzzer mainly by the multivibrator, piezo buzzer, impedance matching and resonance

Boxes, housing and other components. Some piezo buzzer case is also equipped with light-emitting diodes.

Multivibrator constituted by the transistors or integrated circuits. When switched on (1.5 ~ 15V DC working voltage), multi-

Harmonic oscillator start-up, the output 1.5 ~ 2.5kHZ audio signals, impedance matching push piezo buzzer sound.

Piezo buzzer by a lead zirconate titanate or lead magnesium niobate piezoelectric ceramic material. Both sides of the ceramic piece plated silver electrode

The polarization and the aging process, and then with brass or stainless steel sheet stick together.

2. Magnetic Buzzer Magnetic Buzzer by the oscillator, the electromagnetic coil, magnet, diaphragm and housing and other components.

After power on, the audio signal generated by the oscillator current through the electromagnetic coil, the electromagnetic coil generates a magnetic field. Shake

Moving the diaphragm in the electromagnetic coil and magnet interaction, periodically sound vibration.

Active and passive buzzer buzzer What is the difference

Here the "source" does not mean power. But rather refers to the shock source. In other words, the active internal buzzer with shock source, so only

Will be called to an energized.

The passive internal sources without shocks, so if a DC signal can not make it tweet. Must 2K ~ 5K square wave to

Drive it.

Buzzer often than passive expensive, because there multiple oscillator circuit.

Passive buzzer advantages are: 1. Cheap, 2. Sound frequency control, you can make a "more than a meter hair Suola Xi 'efficiency

Fruit. 3. In some special cases, you can reuse a control and LED port active buzzer advantages are: process control

Convenient.

ARDUINO refer to the source:

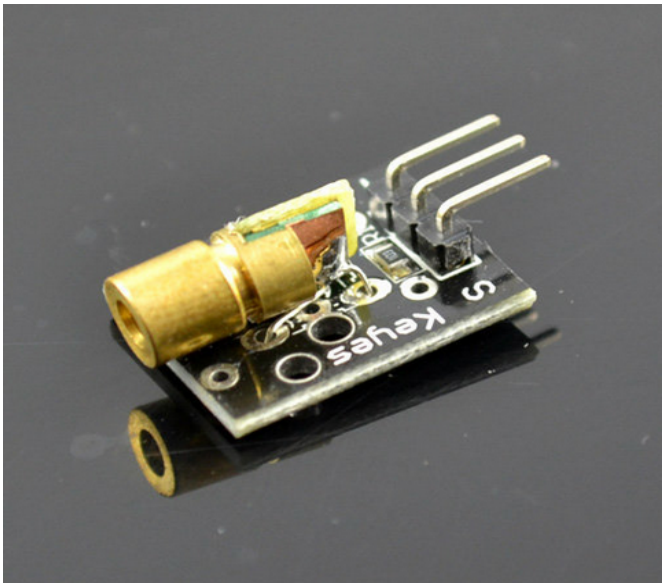
```
int buzzer = 8 ;/ / setting controls the digital IO foot buzzer
void setup ()
{
pinMode (buzzer, OUTPUT) ;/ / set the digital IO pin mode, OUTPUT out of Wen
}
void loop ()
{
unsigned char i, j ;/ / define variables
```

```
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
}
}
}
```

After downloading the program, the buzzer experiments are done.

****

07 KEYES ARDUINO laser transmitter module



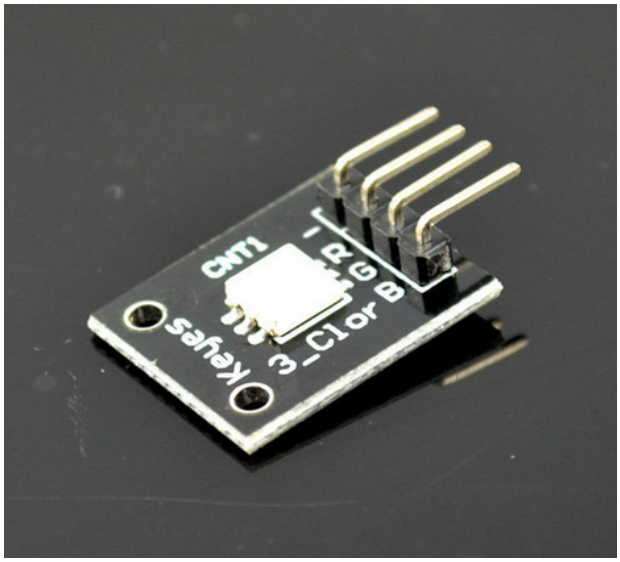
13 laser transmitter module and a digital interface, built-in LED build a simple circuit as shown below

Routines source code:

```
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
}
```

****

08 3 color - full color LED SMD module



I. Overview:

SMD RGB LED module consists of a full-color LED made by R, G, B three pin PWM voltage input can be adjusted

Section three primary colors (red / blue / green) strength in order to achieve full color mixing effect. Control of the module with the Arduino can be achieved

Cool lighting effects.

Second, the product parameters:

Product Features:

1, using 5050 full color LED
2, RGB trichromatic limiting resistor to prevent burnout
3, through the PWM adjusting three primary colors can be mixed to obtain different colors
4, with a variety of single-chip interface
5 Operating voltage: 5V
6.LED drive mode: common cathode driver
Three, Arduino test code:

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

****

9 KEYES ARDUINO photo interrupter module

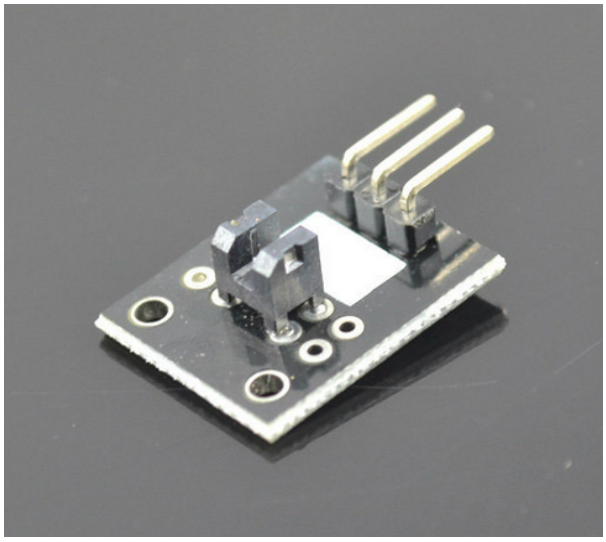


Photo interrupter module and a digital interface, built-in 13 LED build a simple circuit to produce photo-interrupter warning lamp

13 comes with digital interfaces of the LED, the light blocking access number 3 sensor interfaces, sensing when the light interrupter

Device senses a key signal, LED lights, otherwise off.

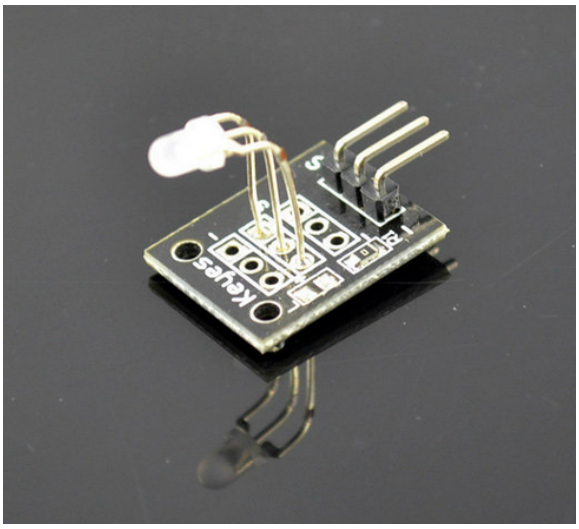
Routines source code:

```
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);
}
}
```

SunFounder

****

010.3mm red and green LED (common cathode) module



Color: Green + Red
Diameter: 3mm
Case Color: None
Package Type: Diffusion
Voltage (V) :2.0-2 .5
Using a current (MA): 10
Viewing angle: 150
Wavelength (NM): 571 +644
Luminous intensity (MCD) :20-40; 40-80
Stent type: long-legged
Our products are widely used in electronic dictionary, PDA, MP3, headphones, digital cameras, VCD, DVD, car audio, communications, computers,
Chargers, power amplifier, instrumentation, gifts, electronic toys and mobile phones, and many other fields.

****
****

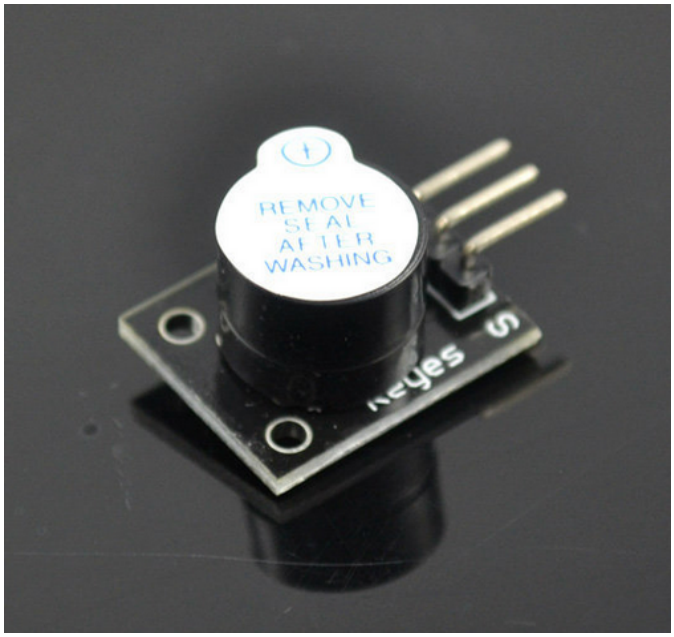
/ / Arduino test code:

```
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);
}
```

****

011.Buzzer Module



One, related presentations

I believe we are not unfamiliar to the buzzer, we will be used in many scenarios buzzer, most of them

Prompted to do is to use the buzzer or alarm, such as button press, to work, or the end of the work breakdown and so on.

Here the microcontroller drives the buzzer to make it on the application description.

Second, drive mode

Conventional drive the buzzer in two ways: one is the PWM output to directly drive, another is the use of

I / O timing flip-level generates a drive waveform for the buzzer to drive.

PWM output PWM output to directly drive is to use a certain port itself can output square wave drive directly

Moving the buzzer. In the SCM software settings, there are several system registers are used to set the PWM output port,

You can set the duty cycle, period, etc. By setting these registers that meets the requirements of the buzzer frequency waves

Shape, simply open the PWM output, PWM output port can output the frequency square wave, this time Lee

This waveform can be driven with a buzzer. Such as frequency of 2000Hz

http://keyes-arduino.taobao.com

Buzzer driver, you can know the cycle of 500μs, so just put the PWM period is set to 500μs,

Duty level is set to 250μs, can generate a frequency of 2000Hz square wave, square wave through reuse

You can go with a transistor drive the buzzer it.

The use of I / O timing flipping to generate drive waveform level would be more trouble that way, you must use regularly

Timing is done, flip through the regular level that meets the requirements of the buzzer frequency waveform This waveform can be

Be used to drive the buzzer. Such as 2500Hz buzzer driver, you can know the cycle of 400μs, which

Samples only need to drive the buzzer I / O port flip once every 200μs level can generate a frequency

2500Hz, 1/2duty duty square wave, and then through the transistor amplifier can drive the buzzer it.

Third, the module uses

We look to the module should understand that he is very easy to use, a power supply terminal, a ground terminal, as well as a

One is the signal input. We just put the power and ground connected, then the signal line connected to IO ports on the line

Fourth, the module function test

Hardware Requirements

Arduino controller × 1

USB data cable × 1
Buzzer Module × 1
Adjustable potentiometer (10K) × 1
The following test examples we mainly learn how to control the buzzer sound, and some simple applications, of course,
Using two different drivers drive the buzzer sound, we can compare the next effect a convenient future use.
Here are the specific connection
Here's the test code is a buzzer on the use of analog control display frequency of the procedure:
Program Description: The first 10-pin to control the buzzer pin.
3 pin to analog pin, the use of adjustable resistor is 10K.
Function: to mobilize adjustable resistor can hear the buzzer sounds obvious frequency changes.

```
int speakerPin = 8 ;/ / control horn pin
int potPin = 4 ;/ / control pin adjustable resistor
int value = 0;
void setup () {
pinMode (speakerPin, OUTPUT);
}
void loop () {
value = analogRead (potPin); reading resistor values pin
digitalWrite (speakerPin, HIGH);
delay (value); adjust the speaker sound of the time;
digitalWrite (speakerPin, LOW);
delay (value); adjust the speaker does not ring a time;
}
```

Here we can say that the delay adjustment potentiometer to achieve the effect of different times, thus changing the buzzer
Audible frequency, we can try to see in the end is not like that, ah ~ ~ ~ ~
Here we added a key switch to control the buzzer, so that we can simulate a simple doorbell, when you press
Key is pressed, the speaker can make any noise.
Physical connections are as follows:
Sample code:

```
const int buttonPin = 4; / / button pin;
const int speakerPin = 8; / / buzzer pin;
/ / Variables will change:
int buttonState = 0; / / read the key pin a value
void setup ()
{
/ / Set button pin to input mode, the buzzer pin output mode;
pinMode (speakerPin, OUTPUT);
pinMode (buttonPin, INPUT);
}
void loop () {
```

```
/ / Read the key one initial value, where I took in the circuit is in the default high, the initial value is high;
buttonState = digitalRead (buttonPin);
/ * If the key is high, then the buzzer did not ring;
Because I just began to take in the hardware circuit initial value is high, so the if condition is true, the buzzer does not sound
* /
if (buttonState == HIGH) {
digitalWrite (speakerPin, LOW);
}
else {
/ / This button is low (also the key is pressed); buzzer sounded
digitalWrite (speakerPin, HIGH);
}
}
```

This procedure is relatively simple, I believe we see to understand, in order to increase people's awareness on the honey me great

Home to write a short code with PWM control buzzer.

The following procedure is to use a PWM (pulse width modulation) control the buzzer, and downloaded to the microcontroller can be heard

Buzzer sounded a different tone, as long as we adjust the notes under the relevant tracks (0,1,2,3,4,5,6,7) can make the beep

Is sing.

Procedures are as follows:

```
int speakerPin = 8;
byte song_table [] = {30, 30, 30, 40, 50, 60, 70, 80, 90, 100,110, 120, 130, 140,
150, 160, 170, 180, 190, 200, 210, 220, 230, 240, 250, 250, 240, 230, 220, 210, 200, 190, 180,
170, 160, 150, 140, 130, 120, 110, 100, 90, 80, 70, 60, 50, 40, 30, 30, 30};
int MAX = 50;
int count = 0;
void setup () {
pinMode (speakerPin, OUTPUT);
}
void loop () {
analogWrite (speakerPin, song_table [count]);
count + +;
if (count> MAX) {
count = 0;
}
delay (50);
}
```

The physical connection is available on specific cases.

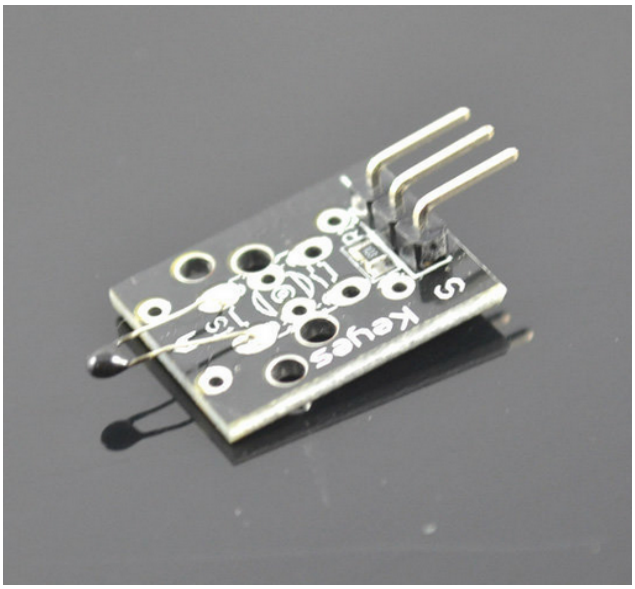
V. Conclusion

As the buzzer control is relatively simple, we do not do too much introduction, we will use the

line, of course, we
The above test case is relatively rough, buzzer sound effect may not be very good, with readers slowly comprehend. . . . .

****

012.Analog temperature sensor

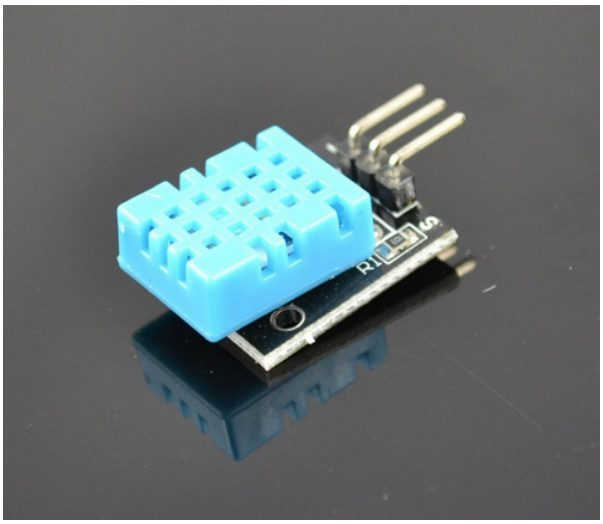


First, the module introduces
The module is based on the thermistor (resistance increases with the ambient temperature changes) works, a sense of real-time
To know the temperature of the surrounding environment changes, we send the data to the Arduino analog IO, then come down as long as we go through Jane
Single programming will be able to convert the sensor output data Celsius temperature values and displayed, it is still easy to use,
It effectively, thereby widely used in gardening, home alarm systems and other devices.
Second, the use
And the general temperature sensor is the same, are three-line package, when we use the power cord connected OK signal output
Into the end on the line, because the module output is an analog signal, we want to signal output

terminal connected to the Arduino analog IO
On sampling, so as to correctly read the temperature value;
Up to this point generally know how to use it, then we need to know about how it is the next temperature measurement?
Third, the module test
Hardware devices
1, Arduino controller × 1
2, USB data cable × 1
3, the analog temperature sensor module × 1
We must have the above things can test to see if an original is a simple thermal how to help us measure the temperature
Degree of it
Circuit is so simple, let's take a look at test program, which will have the answer you want!
}
void loop () {
Serial.print (Thermister (analogRead (0))); / / display Fahrenheit
Serial.println ("c");
delay (500);
}
The reason why the thermistor can measure temperature, because it combines the Steinhart-Hart Thermistor equation, substituting test
Code function double Thermister (int RawADC) is a manifestation of the equation, it is said that, let's look
See the results of it
Test code:
# Include <math.h>
double Thermister (int RawADC) {
double Temp;
Temp = log (((10240000/RawADC) - 10000));
Temp = 1 / (0.001129148 + (0.000234125 + (0.0000000876741 * Temp * Temp)) * Temp);
Temp = Temp - 273.15; / / Convert Kelvin to Celcius
return Temp;
}
void setup () {
Serial.begin (9600);
Is it, or there is a change, huh, huh ..... ^ - ^
Successful completion of the test, then it can later be applied to the design of future we go. . . . . .
Good presentation of the module
Respect to this, thank you!
Tips: code useful about the Steinhart-Hart Thermistor equation applications, specifically we can look online
Look, I will not say more.
Is displayed in the window above the current room temperature, and is so much more. . . . . .
Well, here we hand touched it, to see change is not? ? ? ?

****

013.DHT11 Digital Temperature and Humidity Sensor Module



First, the product introduction

In our daily life, temperature and humidity on our lives has a great impact, especially for factory Production, if we are not well mastered and take relevant measures, then it brings will be a great loss, not Over Well now, there is a temperature sensor that can measure not only but also measured humidity, it really can solve our problems Yet. Well, following up on learning how to use it, it brings convenience to your life. . . . .

Second, the module related presentations

DHT11 digital temperature and humidity sensor is a calibrated digital signal output temperature and humidity combined sensor, which Application-specific modules capture technology and digital temperature and humidity sensor technology to ensure that products with high reliability and excellent Long-term stability. The product has excellent quality, fast response, anti-interference ability, high cost and other advantages. Single Wire serial interface that allows quick and easy system integration. Ultra-small size, low power consumption, signal transmission distance Up to 20 meters, making it to the class of applications and even the most demanding applications is the best choice. Products for the 4-pin single row Pin package, easy connection.

Third, the technical parameters

Supply voltage: 3.3 ~ 5.5V DC

Output: single-bus digital signal

Measuring range: Humidity 20-90% RH, Temperature 0 ~ 50 ℃

Accuracy: Humidity + -5% RH, temperature + -2 ℃
Resolution: Humidity 1% RH, temperature 1 ℃
Long-term stability: <± 1% RH / Year
IV Notes
1, to avoid the use of the condensation conditions
2, long-term storage temperature 10-40 ℃, humidity below 60%
3, the use of power and ground connection to be correct, so as not to damage the sensor
Five, instructions for use
About DHT11 specific timing problems we can refer to its datasheet, see the following modules, such as our company
He connected with the Arduino board
The above is typical of its connection with the processor, the following are specific connection reference
Module "+" Termination +5 V output, "-" Terminal GND, "S" Termination digital port on the 7th pin (when
However, this can also define your own digital pins); connection is very simple, we only have the following test phase. . . . . .
Six, module function test
Hardware Requirements
Arduino controller × 1
USB data cable × 1
DHT 11 module × 1
In order to facilitate testing, we have written a short test code, for reference only

```
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 ()
{
```

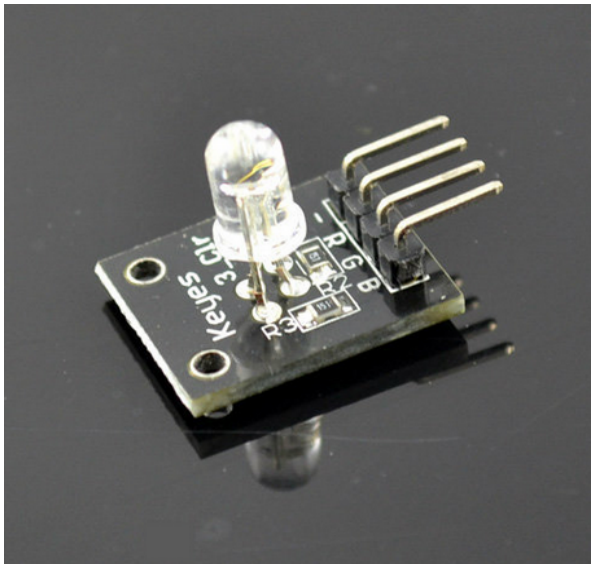
```
digitalWrite (DHpin, LOW); / / bus down, send start signal;
delay (30); / / delay greater than 18ms, so DHT11 start signal can be detected;
digitalWrite (DHpin, HIGH);
delayMicroseconds (40); / / Wait DHT11 response;
pinMode (DHpin, INPUT);
while (digitalRead (DHpin) == HIGH);
delayMicroseconds (80); / / DHT11 a 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 + +) / / receives temperature and humidity data, the parity bit is not considered;
dat [i] = read_data ();
pinMode (DHpin, OUTPUT);
digitalWrite (DHpin, HIGH); / / sending 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);
}
```

Well, we look at the test code compiler, compile the results we can see, really want to see now
At ambient temperature and humidity in the end is how much of that they are invisible, can be really curious **
Anthracene, we burn the program into Arduino board, and then wait to open the Serial
Monitor window, watching the results came out, wow, is not it a little excited!
We then hand hold it, wait a minute. . . . . What happens to look at the screen?
Look Na, elevated temperature display, next time we can use it to test the temperature of the palm,

ha ha. . . . . So then we
Kazakhstan few breaths try it
And imagine the same humidity significantly larger, ha, this thing really is pretty good. . . . Are interested, you can
Be on your own hands frequently be in place, so your heart there at the end.
VII Conclusion
Modules introduced here will tell some, and if you are interested in it, then you can buy your own at home, slowly
Under study, and strive to spend it to make a very creative things out for the rest of your life more fun. . . ^ - ^

****

014 3 color - full color LED module



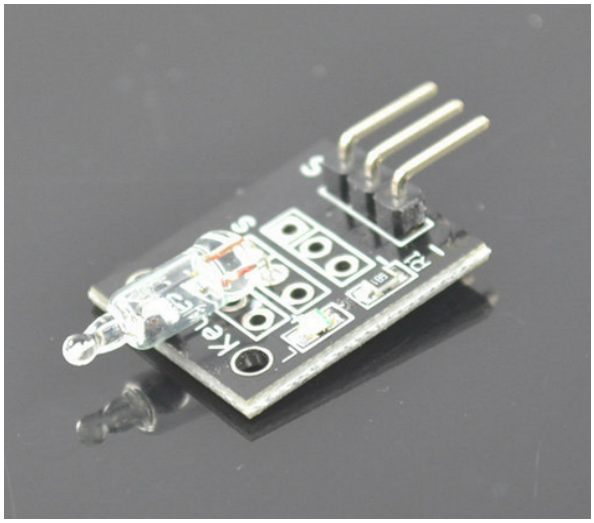
I. Overview:
RGB LED module consists of a plug-in full color LED made by R, G, B three pin PWM voltage input can be adjusted
Section three primary colors (red / blue / green) strength in order to achieve full color mixing effect. Control of the module with the Arduino can be achieved
Cool lighting effects.
Second, the product parameters:
Product Features:
1, the use of plug-in full-color LED
2, RGB trichromatic limiting resistor to prevent burnout
3, through the PWM adjusting three primary colors can be mixed to obtain different colors
4, with a variety of single-chip interface

5, the working voltage: 5V
6, LED drive mode: common cathode driver
Three, Arduino test code:

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

****

015 ARDUINO mercury tilt switch module



Mercury switch module and a digital interface, built-in 13 LED build a simple circuit to produce tilt warning lamp

13 comes with digital interfaces of the LED, the mercury tilt switch sensor interface to access digital 3, when the water

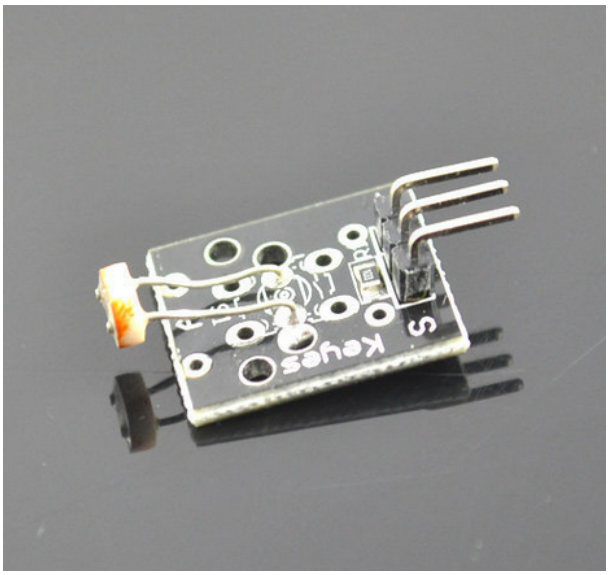
Silver tilt switch sensor senses a key signal, LED lights, otherwise off.

Routines source code:

```
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
http://keyes-arduino.taobao.com
}
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);
}
}
```

****

016 photoresistor module



I. Introduction

Photoresistor in our daily lives but also be able to see, is mainly used in smart switch, giving our students

Live brings some convenience, but, in our daily electronic design will also be used. Then in order to make better use,

We have the appropriate module, easy to use, and efficient.

Second, an overview

Photoresistors are semiconductor photosensitive devices, in addition to having high sensitivity, fast response, consistent with the spectral characteristics and value of r

Good features, but at a high temperature, and humidity in harsh environments, but also to maintain a high degree of stability and reliability, wide

Pan used cameras, solar garden lights, lawn, detectors, clock, music, cups, gift boxes, mini-

Night light, light voice switches, lights automatically switch toys and a variety of light control, light control lighting, lamps and other light automatic opening

OFF control field

Third, the main parameters and characteristics

1, according to the spectral characteristics of the photoresistor has three photoresistor: Ultraviolet photosensitive resistance, infrared light-sensitive resistors,

Visible photosensitive resistance;

2, the main parameters are as follows:

A, dark current, dark resistance: photosensitive resistance at a certain applied voltage, when the light is not irradiated when the flowing

Current is called dark current. Applied voltage and dark current ratio as the dark resistance;

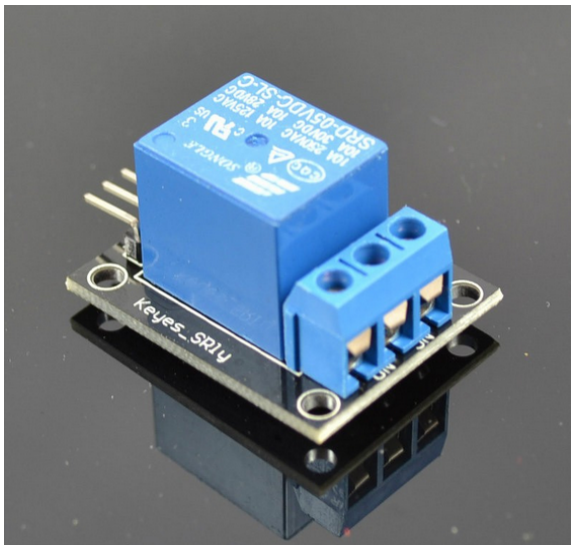
B, Sensitivity: Sensitivity is irradiated by light sensitive resistor, the resistance value (dark resistance) when irradiated with light
Resistance value (light resistance) the relative change in values.
C, volt-ampere characteristic curve. Voltage characteristic curves are used to describe the resistance of the applied voltage and the photosensitive photocurrent relationship,
On the photosensitive devices, the light current with applied voltage increases.
D, temperature coefficient. Photoelectric effect photoresistor affected by temperature, at a low temperature portion photoresistor photoelectric
Sensitive high sensitivity at high temperatures is low.
E, rated power. Photosensitive resistor rated power is allowed for certain lines in the power consumed when the temperature rise
High, its power consumption is reduced.
Fourth, the use
Because we are in the Arduino environment, then we take a look at how they are connected to the corresponding
And general sensors, two power lines and a data cable, wiring is simple;
Then our subsequent test circuit wiring can be like it. Well, since the wiring will be, we take a look at the following
In the end how to use it.
Five, module test
Hardware Requirements
1, Arduino controller × 1
2, USB data cable × 1
3, the photosensitive resistor module × 1

```
int sensorPin = 2;
int value = 0;
void setup () {
Serial.begin (9600);
}
void loop () {
value = analogRead (sensorPin);
Serial.println (value, DEC);
delay (50);
}
```

Well, the next test code
We then tested the photoresistor just read the module's output analog voltage value, the test results, we find that when
There is light, the output voltage is high, the equivalent of the switch is turned on, but there is no light, the output voltage is low, the equivalent of switching off
Open, in practical applications can use this point.
Below are bright and dull when compared to the output data:
The data window is positioned above the light, the following is the data light;

****

017 Relay Module



I. Introduction

At present, the company's products have multiple types of relays, including a relay, two relays, 4 relay,

6 relay, 8 relays, etc., to meet the needs of different users use the relay is when the input (excitation

Reed amount) meet the requirements change, the output circuit in the electrical manipulation occurred predetermined amount charged a step change in an electrical

Makers. The company produces relay module can be connected to 240V AC or 28V DC power into a variety of other electrical parts

Line control. Can be achieved using single-chip timing control switching purposes. Can be used in anti-theft alarm, toys, building

Let other fields. Relay is an electrically controlled device. It has a control system (also known as the input circuit) and the control system

(Also known as the output circuit) the interaction between. Commonly used in automation control circuit, it is actually a small

Current to control a large current operation "automatic switch." Therefore, the circuit automatically adjusts the play, safety protection, transfer

Conversion circuit and so on. Particularly suitable for single-chip control strong electric devices.

In the control and use is also very convenient, just give input corresponding output relay different levels, you can

Achieved by controlling the relay control purposes other devices, in addition, in the multi-channel

relay PCB layout on the use of two lines
Layout, user-lead connections. While in the circuit of a DC diode added greatly improved relay
Module to engage current ability to prevent the transistor being burned. In addition, we added a relay this power indicator
Lights (except relay all the way), the indicator is red. In brightest relay also adds a status indicator. Can
To let everyone real-time observation of the relay switch status.
Second, the module classification introduced
1, one relay
A, the main purpose
Relay is a function of the automatic isolation switching elements, are widely used in remote control, telemetry, communications, automatic control,
Mechatronics and power electronic devices, is one of the most important control elements.
Boils down to the following effect:
1) expand the control range: for example, multi-contact relay control signal reaches a certain value, you can not press the contact group
Different forms, and for access, breaking, connected multi-channel circuits.
2) Zoom: for example, sensitive relays, relays, etc., with a very small amount of control, you can control a large
The power of the circuit.
3) Integrated signal: for example, when a plurality of control signals in the form prescribed input multi-winding relay, by comparison mechanized
Together, to achieve the desired control effect.
4) automatic, remote control and monitoring: for example, the automatic device relays together with other appliances, can be composed of program control
Wire line, in order to achieve automatic operation
B, Note
1) Rated voltage: refers to normal working hours relay coil voltage required,
The control circuit is a control voltage. According to the relay model, can be ac
Pressure, it can be a DC voltage.
2) DC resistance: refers to the relay coil DC resistance, measured by the multimeter.
3) Pick-up current: refers to the relay pull action can produce a minimum current. In normal use, the current will be given
Be slightly larger than the pull current, so that the relay can be operated stably. The work of the coil voltage is applied, generally do not
To more than 1.5 times the rated working voltage, otherwise it will have a greater current to the coil burnt.
4) release current: refers to the relay generates the maximum current release action. When the relay state current is reduced to a
Certain extent, the relay will revert to the release of unpowered state. Then the current is much smaller than the pull current.
5) contact switch voltage and current: is the relay to allow the applied voltage and current. It determines the relay to control
Voltage and current size, use can not exceed this value, it will be very easy to damage the relay

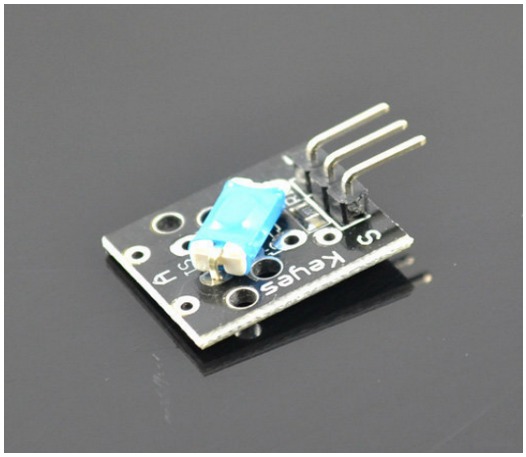
contacts.
C, module test
Pin Description below
Description: COM to VCC, NO then we have to control the LED anode, which
Like when the relay turns on, LED lights will be lit;
To complete the look of this test must be prepared to what what they specifically
Arduino controller × 1
USB data cable × 1
1 relay module × 1
Led indicator × 1
The resistance of the resistance 330 × 1
Of course, we have the following physical connections for a specific reference
Well, Here is a simple relay control test procedure:

```
int relay = 10; / / relay turns trigger signal - active high;
http://keyes-arduino.taobao.com
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);
}
```

Program Description: The program notes in the conduction and disconnection refers to the way that we want that we are using the NO side,
When S relay switches into high and hit the NO side, the switch is turned on, connected to the LED will be lit, otherwise the switch
Hit the NC side, NO direction disconnect, LED light goes out;
You will see the test results led lights flashing interval 1s;

****

018 KEYES ARDUINO tilt switch module



Tilt switch module and a digital interface, built-in 13 LED build a simple circuit to produce tilt warning lamp

13 comes with digital interfaces of the LED, the tilt switch sensor interface to access digital 3, when the tilt open

Off sensor senses a key signal, LED lights, otherwise off.

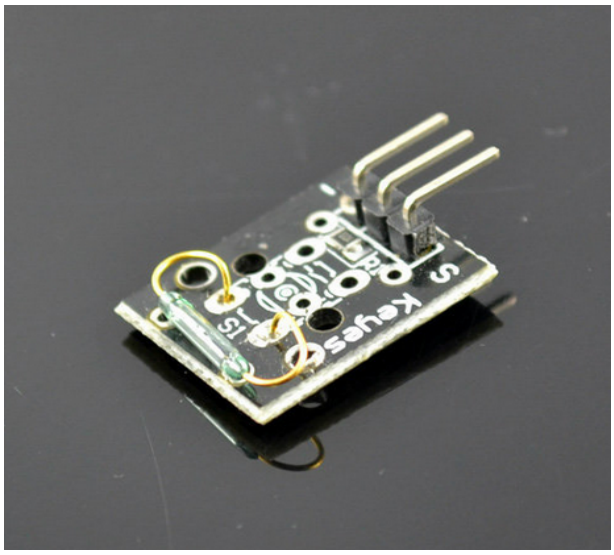
Routines source code:

```
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);
}
}
```

****

SunFounder

019 MINI reed switch module



Reed module and the interface comes with digital 13 LED build a simple circuit to produce a Reed warning lamp

13 comes with digital interfaces of the LED, the Reed sensor access number 3 interface, when Reed sensors

Sensed a key signal, LED lights, otherwise off.

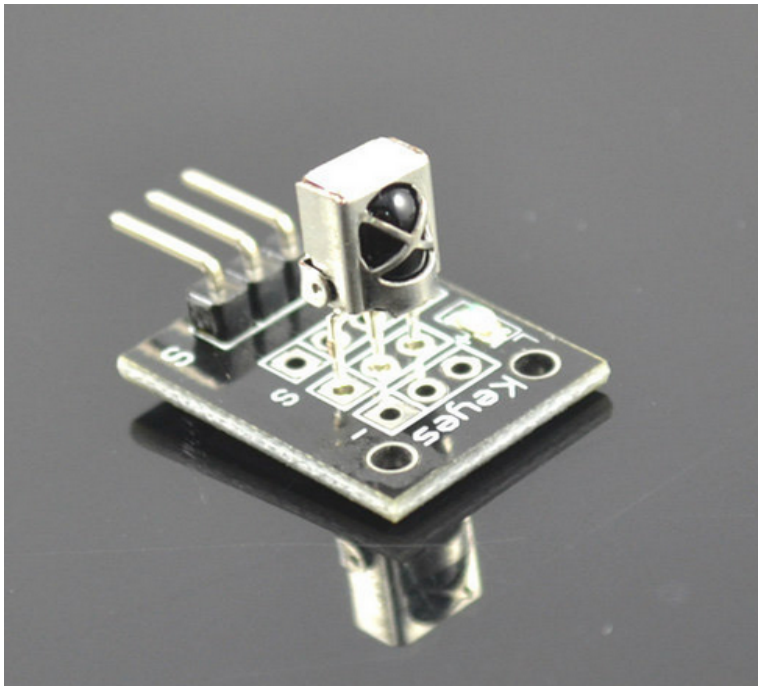
Routines source code:

```
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);
```

```
}
}
```

****

020 Infrared remote control module



I. Introduction

This is a new ultra-thin 38K universal infrared remote control, using the NEC encoding format, mainly for cars

Containing MP3, foot bath, lighting design equipped, digital photo frame, microcontroller development board learning board and other occasions. Because it is based on non-

Line remote control, so people seem easy to use, effective, and now more and more wide application field, then for

This product of our company that we will make the following introduction.

Second, the technical parameters

Infrared remote control distance: more than 8 meters

Launch tube infrared wavelength: 940Nm

Crystal frequency: 455KHZ crystal

Carrier frequency: 38KHZ
Encoding: encoding format for the NEC
Size: 86 * 40 * 6mm
Power: CR2025/1600mAH
Third, the use
We must remember that before using the remote control to the infrared cell, as well as infrared remote control infrared receiver module allows to combine
Use, which is responsible for receiving infrared remote control transmitter over the information and decodes it into hexadecimal code, so as to achieve
Established communications.
The infrared receiver module with Arduino properly connected, where S connection D11, VCC connected +5 V, GND connected GND, and
To be fixed;
Here is its connection with the Arduino specific circuit
Fourth, the module test
1, Arduino controller × 1
2, USB data cable × 1
-3, Infrared remote control × 1
4, the infrared receiver module × 1
Accordance with the instructions connected test circuit,
Well, take a good circuit, then here we start testing it
In this test we will be encoded in the corresponding keys Serial Monitor window displays
Look at the test code:

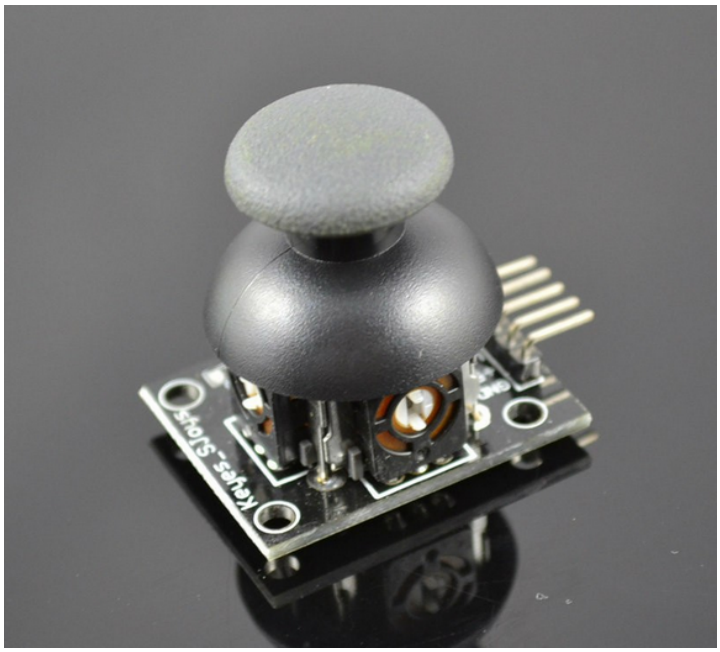
```
# Include <IRremote.h>
int RECV_PIN = 11; / / define input pin on Arduino
IRrecv irrecv (RECV_PIN);
decode_results results;
void setup ()
{
Serial.begin (9600);
irrecv.enableIRIn (); / / Start the receiver
}
void loop () {
if (irrecv.decode (& results)) {
Serial.println (results.value, HEX);
irrecv.resume (); / / Receive the next value
}
}
```

Compiled the above code, we can download the test, pay attention before the test to make sure the remote control has Shanghao
Battery! Here are some test results
In the testing process should pay attention to the infrared remote control and infrared receiver position, making sure the infrared remote receiver can be well received
http://keyes-arduino.taobao.com

- Controller transmitting signals over; by the test results can be seen each key has its own hexadecimal encoding, if I
We long press a button in the Serial Monitor window shows FFFFFFFF.

****

021 PS2 game joystick axis sensor module



First, the product description
The company produces PS2 game joystick axis sensor module consists of using original quality metal PS2 joystick potentiometer system
For, with (X, Y) 2-axis analog output, (Z) 1 digital output channel button. With Arduino sensor expansion board can be made
For remote control and other interactive work. In addition the product in order to allow customers to more easily fit arduino expansion boards and other standard interfaces
Mouth, in the design of the X, Y, Z-axis circuit leads individually, you can use three dedicated lines really pin ARDUINO
Plug into the expansion board for use. It is convenient.
Second, product characteristics
It is like a game console joystick, you can control the joystick module input x, y, z of
http://keyes-arduino.taobao.com
PS2 joystick game controller module Joystick
Values, and to achieve a particular value in a function, it can be considered a combination of buttons and a potentiometer. Data

Type of x, y dimension for the analog input signal is a digital input signal z dimension, therefore, x and y connected to the analog port
Pin sensor end, and z port is connected to the digital port.
Third, the use
On how to use, we first look at how it works now, so we know it is there in the end
How, which we find it helpful to use, there is a functional diagram below, we take a look
Now we should clear it, in fact, it is a potentiometer Well, x, y dimension of the data output is analog
Port readout voltage value, is not a little surprised. Of course, this is not shown above, z-dimensional data output, in fact, it is more
Simple, we know that z-dimensional output only 0 and 1, then it can be achieved through a button bar. Now on we should
Surface of saying, it is a potentiometer and button combination (To be honest, if you do not understand it just to see that
Sentence is a bit foggy it?).
After reading the chart I believe we all know how to use it right Arduino, x, y dimension we received two analog ports
Read their values, and z dimensions we are to the digital port, so that the line, plus the power and ground, so fine. . . .
Fourth, the module test
Let's look at this test what things we have, in fact, not much. . . .
Arduino controller × 1
USB data cable × 1
Game sensor module × 1
Here x I connected an analog port 0, y even an analog port 1, z I connect to the digital port 7, the relevant port
No. You can look at the individual situation, but properties can not be wrong. Code is as follows

```
int sensorPin = 5;
int value = 0;
void setup () {
pinMode (7, OUTPUT);
Serial.begin (9600);
}
void loop () {
value = analogRead (0);
Serial.print ("X:");
Serial.print (value, DEC);
value = analogRead (1);
Serial.print ("| Y:");
Serial.print (value, DEC);
value = digitalRead (7);
Serial.print ("| Z:");
Serial.println (value, DEC);
delay (100);
```

}

Program function: it can play sensor status (x, y, z three-dimensional data) in real time response to the computer screen (we use the

Serial Monitor window), here I cut a figure for everyone to look at it

Is not it, the test is successful, or more I play, we have the opportunity to see for yourself. . . . .

I believe we carefully read the above schematic would say, in fact, I can also DIY

A yo, but might not look so good, it is not so smoothly control it, ah, yes, imitation principle,

Prepare their own two potentiometers and a key entirely possible Yeah, this would leave you complete it, the following test code,

We can take a try. . .

```
int JoyStick_X = 0; / / x
int JoyStick_Y = 1; / / 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);
}
```

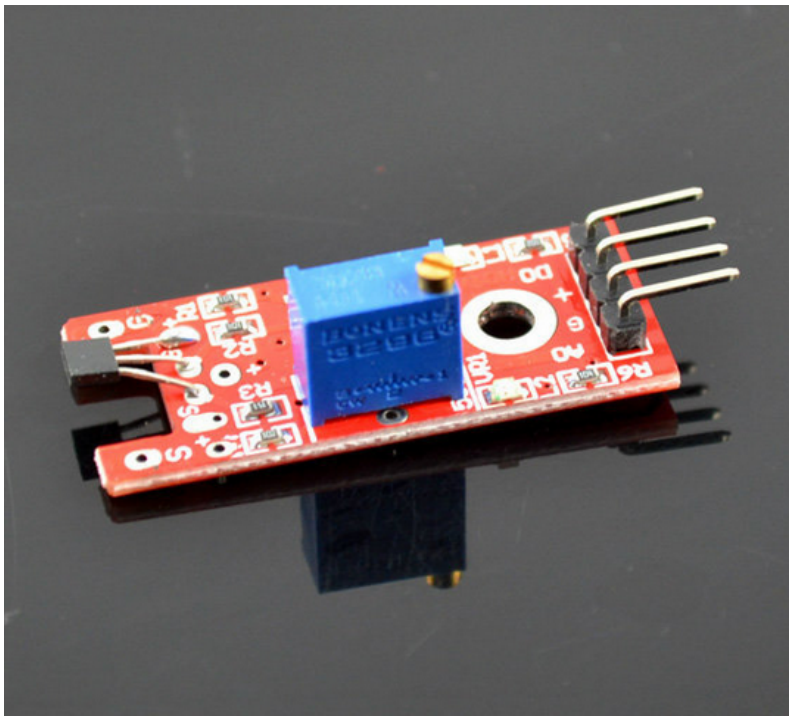
V. Conclusion

The introduction of the game went to this sensor, in fact, use it to make a pretty good thing, we can enjoy the

Imagination, as well as if you have a need, then you can contact us, thank you!

****

022 KEYES ARDUINO Linear Hall magnetic module



Linear Hall magnetic module and a digital interface, built-in 13 LED build a simple circuit to produce a magnetic field warning lamp

13 comes with digital interfaces of the LED, the linear Hall sensor magnetometer access number 3 interface, when linear Hall magnetometer

Sensor senses a key signal, LED lights, otherwise off.

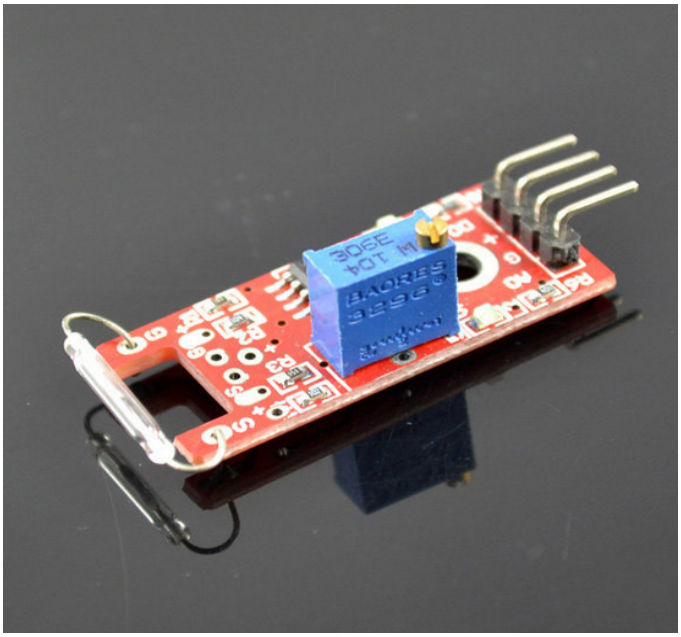
Routines source code:

```
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);
}
}
```

****

023 Reed module



Reed module and the interface comes with digital 13 LED build a simple circuit to produce a Reed warning lamp

13 comes with digital interfaces of the LED, the Reed sensor access number 3 interface, when Reed sensors

Sensed a key signal, LED lights, otherwise off.

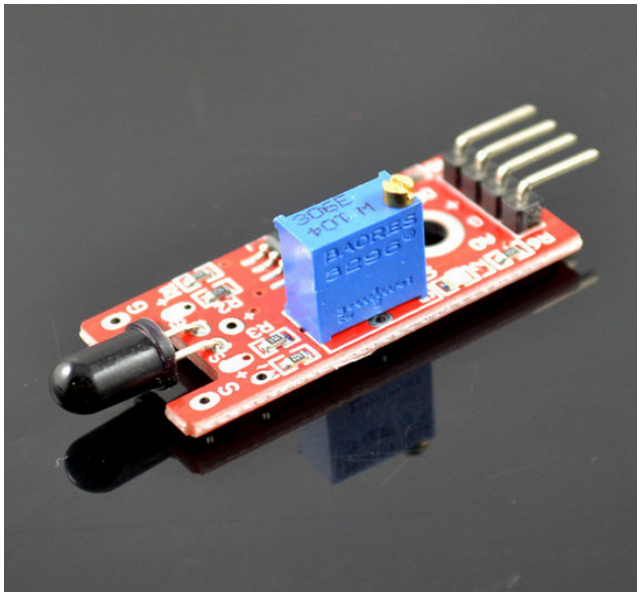
Routines source code:

```
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);
}
}
```

****

024 KEYES ARDUINO flame module



Flame module Interface module and number 13 comes with LED build a simple circuit to produce flame warning lamp

13 comes with digital interfaces of the LED, the flame sensor connected digital three interfaces, when the flame sensor senses

There are key signal detected, LED lights, otherwise off.

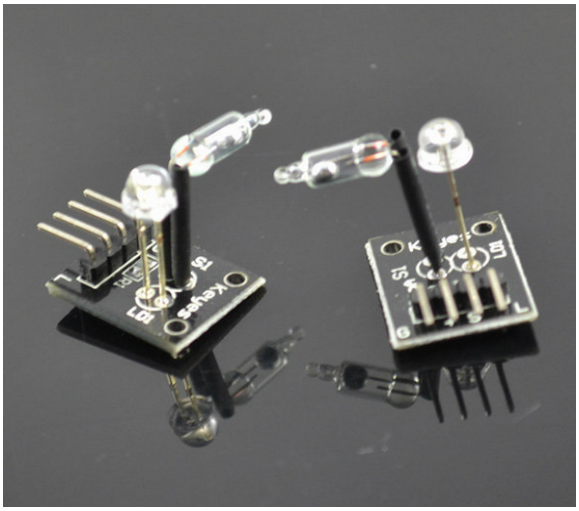
Routines source code:

```
int Led = 13 ;/ / define LED Interface
int buttonpin = 3; / / define the flame sensor interface
int val ;/ / define numeric variables val
void setup ()
{
pinMode (Led, OUTPUT) ;/ / define LED as output interface
pinMode (buttonpin, INPUT) ;/ / output interface defines the flame sensor
```

```
}
void loop ()
{
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);
}
}
```

****

025 Magic light cup module



Magic Light Cup modules are easy to Interactive Technology Division developed a can and ARDUINO interactive modules,

PWM dimming principle is to use the principle of two modules brightness changes.

Mercury switches provide a digital signal that triggers the PWM regulator, through the program design,

We can see the light like two cups filled with the effect of shuffling back and forth.

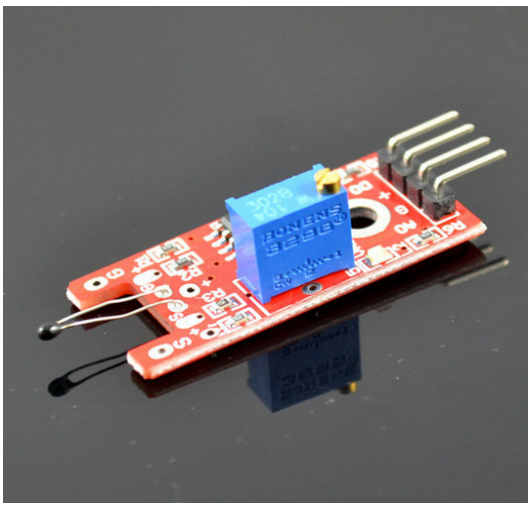
Attach ARDUINO code

```
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); / / A few Guan Yuan (ii)? analogWrite (LedPinB, 2
55 - brightness); / / B Yuan (ii) a few Bang? Delay (25);
}
```

Note: This experiment requires two modules simultaneously, so buy this product is to purchase two groups

****

026 KEYES ARDUINO digital temperature module



Digital temperature module and a digital interface, built-in 13 LED build a simple circuit, making the temperature warning lamp

13 comes with digital interfaces of the LED, the digital temperature sensor connected digital three interfaces, when the digital temperature

Sensor senses a key signal, LED lights, otherwise off.

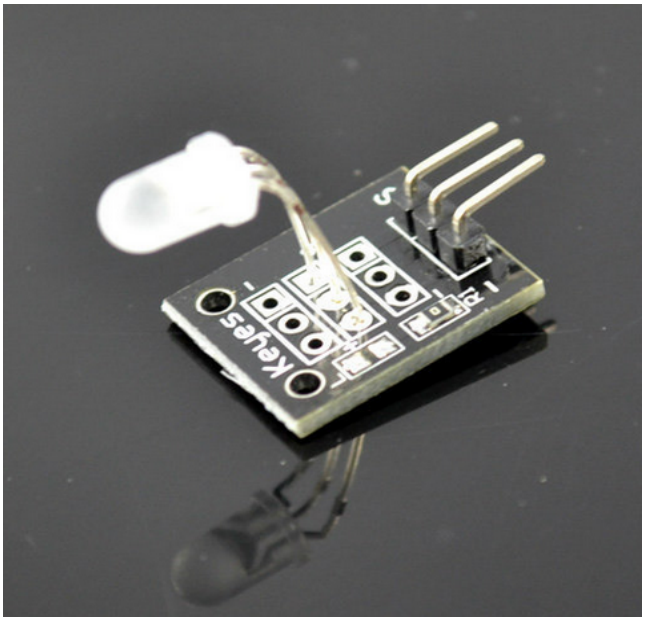
Routines source code:

```
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);
}
}
```

****

027 5mm red and green LED (common cathode) module

SunFounder



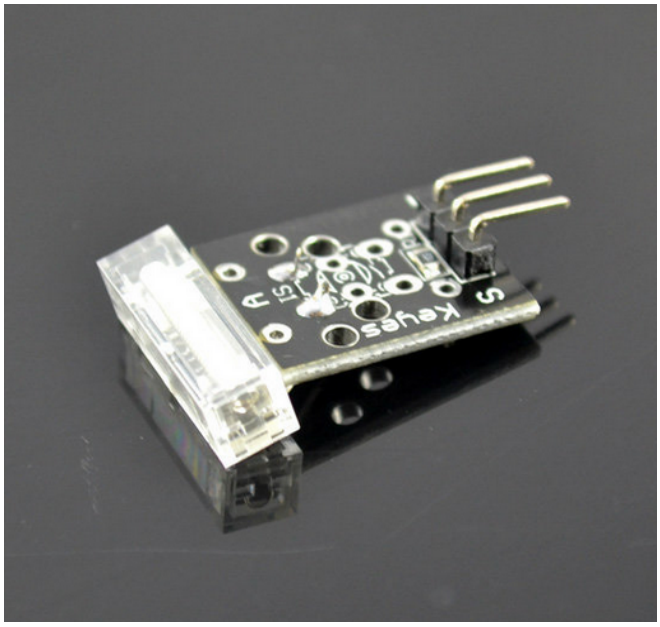
Color: Green + Red
Diameter: 5mm
Case Color: None
Package Type: Diffusion
Voltage (V): G :2.3-2 .6 V; R :1.9-2 .2 V
Using a current (MA): 20
Viewing angle: Wavelength (NM): 571 +625
Luminous intensity (MCD) :20-40; 60-80
Stent type: long-legged
Our products are widely used in electronic dictionary, PDA, MP3, headphones, digital cameras, VCD, DVD, car audio, communications, computers,
Chargers, power amplifier, instrumentation, gifts, electronic toys and mobile phones, and many other fields.

```
/ / Arduino test code:
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);
}
```

****

028 KEYES ARDUINO knock sensor module



13 knock sensor module and a digital interface, built-in LED build a simple circuit to produce percussion flasher.

13 Interface comes with digital LED, will knock sensor connected digital 3 interface, when percussion sensor senses

Measure

To percussive signals, LED flashing light.

Routines source code:

```
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);
}
}
```

****

029 Infrared obstacle avoidance sensors

Infrared obstacle avoidance sensor is designed for the design of a wheeled robot obstacle avoidance sensor distance adjustable. This ambient light sensor

Adaptable, high precision, having a pair of infrared transmitter and receiver, transmitter tubes emit a certain frequency of infrared,

When detecting the direction of an obstacle (reflector), the infrared receiver tube receiver is

reflected back, when the indicator is lit,
Through the circuit, the signal output interface output digital signal that can be detected by means of potentiometer knob to adjust the distance, the effective distance
From 2 ~ 40cm, working voltage of 3.3V-5V, operating voltage range as broad, relatively large fluctuations in the power supply voltage of the situation
Stable condition and still work for a variety of microcontrollers, Arduino controller, BS2 controller, attached to the robot that
Can sense changes in their surroundings.
Specifications:
1 Working voltage: DC 3.3V-5V
(2) Working current: ≥ 20mA
(3) Operating temperature: -10 ℃ - +50 ℃
4 detection distance :2-40cm
5.IO Interface: 4-wire interfaces (- / + / S / EN)
6 Output signal: TTL level (low level there is an obstacle, no obstacle high)
7. Adjustment: adjust multi-turn resistance
8 Effective angle: 35 °
7 Size: 28mm × 23mm
8 Weight Size: 9g

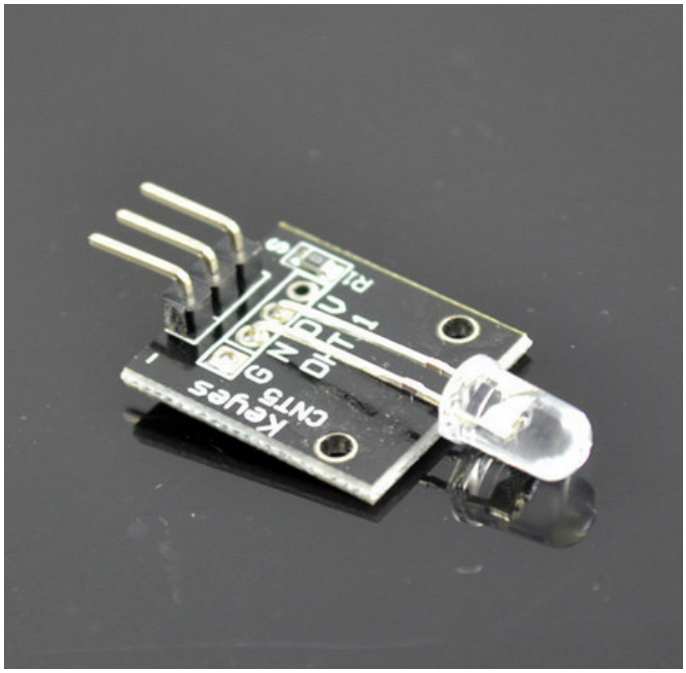
Here we use the obstacle avoidance module and a digital interface, built-in 13 LED build a simple circuit, making avoidance warning lamp, the obstacle avoidance
Sensor Access Digital 3 interface, when obstacle avoidance sensor senses a signal, LED light, and vice versa off.
Routines source code:

```
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);
}
```

}

****

030 7 color flashing LED module automatically



7 color flashing LED module automatically uses 5mm round high-brightness light-emitting diode which has the following characteristics:

1) Product Type: LED
2) Product Model: YB-3120B4PnYG-PM
3) Shape: Round LED 5mm DIP type
4) Color: pink yellow green (high brightness)
5) Lens type: white mist
6) Standard Forward Voltage :3.0-4 .5 V

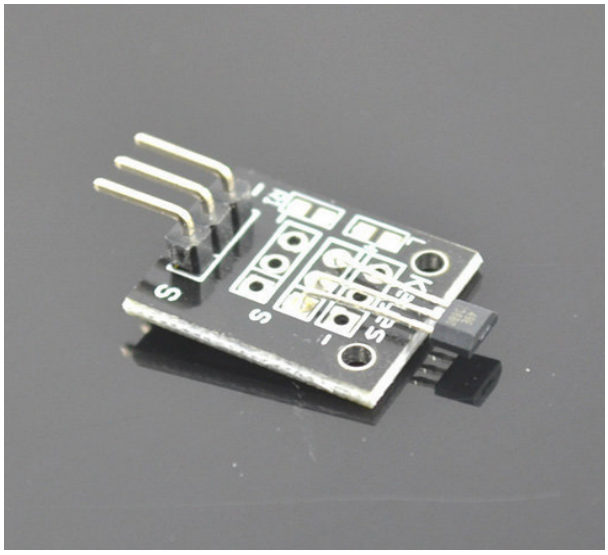
Arduino test cod:

/ *
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
}
```

****

031 analog Hall magnetic sensor module



Analog magnetic sensor module and a digital interface, built-in 13 LED build a simple circuit to produce a magnetic flash

Makers.13 comes with digital interfaces of the LED, the analog magnetic sensor connected to the power board analog 5 ARDUINO Interfaces, when analog magnetic sensor to a signal, LED lights, otherwise the lights out.

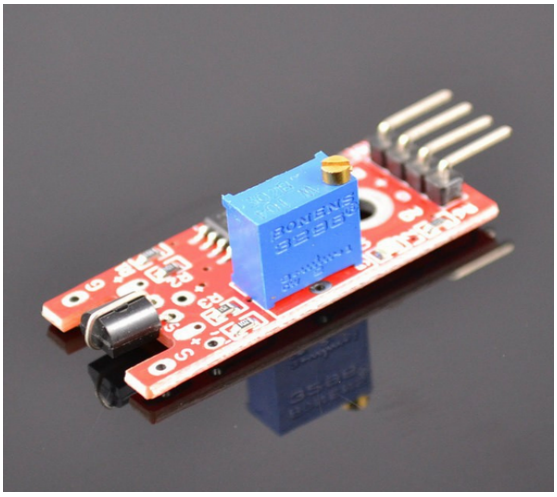
With reference program:

```
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);
}
```

****

032 KEYES ARDUINO touch module



Metal Touch Interface module and number 13 comes with LED build a simple circuit to produce a touch cue lights

13 comes with digital interfaces of the LED, the metal touch sensor connected digital three interfaces, when a metal touch

Sensor senses a key signal, LED lights, otherwise off.

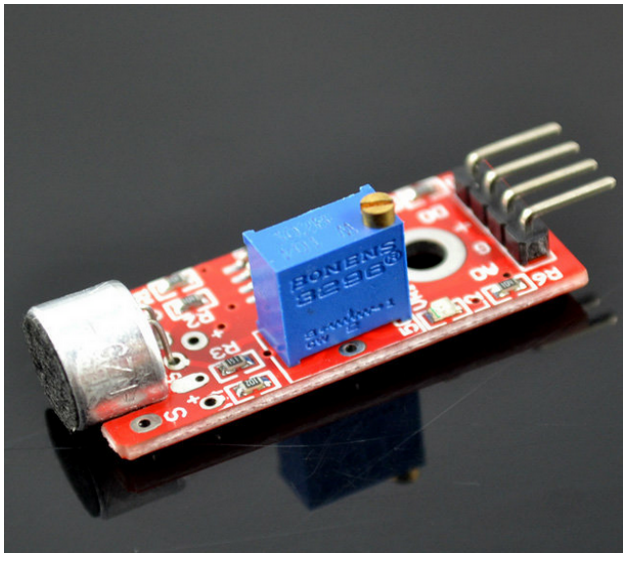
Routines source code:

```
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);
}
}
```

****

033 High sensitivity sound detection module



For sound detection

Module has two outputs:

1, AO, analog output, real-time output voltage signal of the microphone

2, DO, when the sound intensity reaches a certain threshold, the output high and low signal [threshold - sensitivity can be adjusted via potentiometer

Day]

Module features:

2, there is a mounting screw hole 3mm

3, the use 5v DC power supply

4, with analog output

5, there are threshold level output flip

6, high sensitive microphone and high sensitivity.

7, a power indicator light

8, the comparator output is light

ARDUIINO code:

1 Digital output:

```
int Led = 13 ;/ / define LED Interface
int buttonpin = 3 / / define D0 Sensor Interface
; Int val ;/ / 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 3 to read val
if (val == HIGH) / / When the sound detection module detects a signal, LED flashes
{
digitalWrite (Led, HIGH)
;}
else
{
digitalWrite (Led, LOW)
;}
}
```

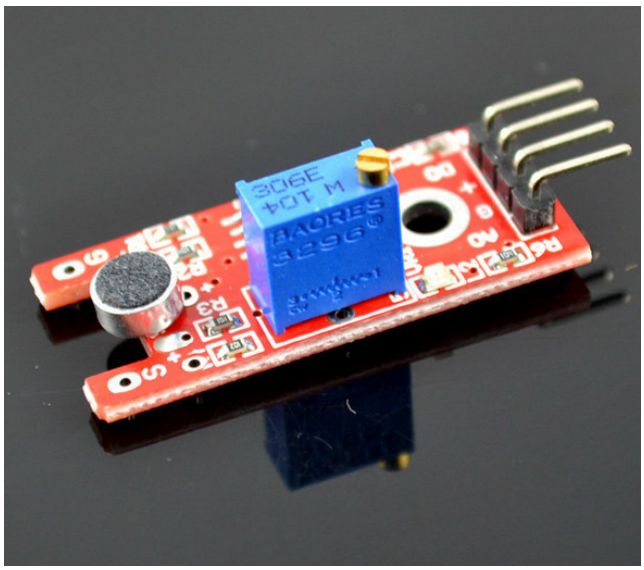
2 analog outputs:

```
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);
}
```

****

034 Microphone sound detection module



For sound detection
Module has two outputs:
1, AO, analog output, real-time output voltage signal of the microphone
2, DO, when the sound intensity reaches a certain threshold, the output high and low signal [threshold - sensitivity can be adjusted via potentiometer
Day]
Module features:
2, there is a mounting screw hole 3mm
3, the use 5v DC power supply
4, with analog output
5, there are threshold level output flip
6, a power indicator light
7, the comparator output is light
ARDUIINO code:
1 Digital output:

```
int Led = 13 ;/ / define LED Interface
int buttonpin = 3 / / define D0 Sensor Interface
int val ;/ / 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 3 to read val
if (val == HIGH) / / When the sound detection module detects a signal, LED flashes
{
digitalWrite (Led, HIGH)
}
else
{
digitalWrite (Led, LOW)
}
}
```

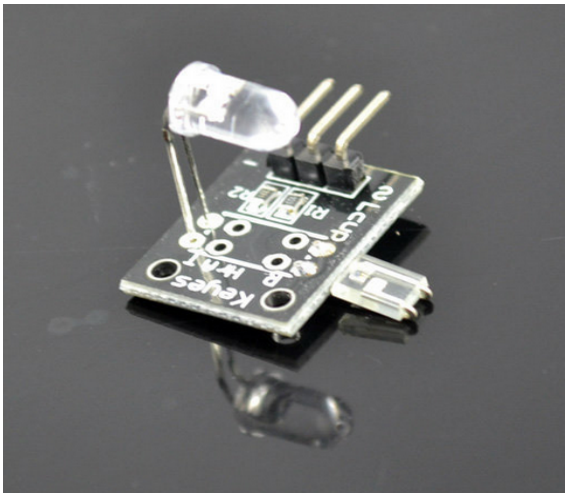
2 analog outputs:

```
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);
}
```

http://keyes-arduino.taobao.com

****

035 finger measuring heartbeat module



This project uses bright infrared (IR) LED and a phototransistor to detect the pulse of the finger, a red LED flashes with each pulse.
Pulse monitor works as follows: The LED is the light side of the finger, and phototransistor on the other side of the finger, phototransistor used to obtain the flux emitted, when the blood pressure pulse by the finger when the resistance of the phototransistor will be slight changed.
The project's schematic circuit as shown,

We chose a very high resistance resistor R1, because most of the light through the finger is absorbed, it is desirable phototransistor sensitive enough. Resistance can be selected by experiment to get the best results.
The most important is to keep the shield stray light into the phototransistor. For home lighting that is particularly important because the lights at home mostly based 50HZ or 60HZ fluctuate, so faint heartbeat will add considerable noise.

ARDUINO test code:

```
/ / Pulse Monitor Test Script

int ledPin = 13;
int sensorPin = 0;

double alpha = 0.75;
int period = 20;
double change = 0.0;

void setup ()
{
  pinMode (ledPin, OUTPUT);
  Serial.begin (115200);
}
```

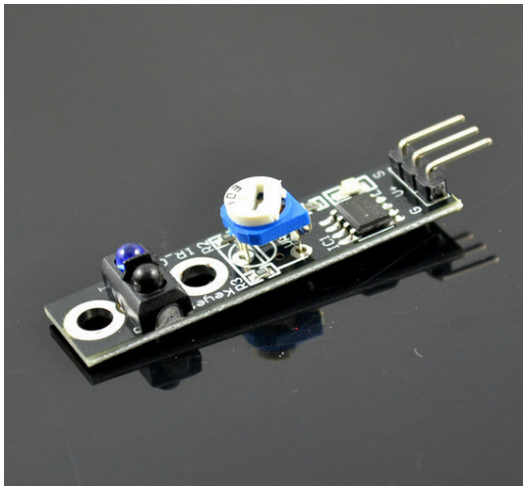
```
void loop ()
{
   static double oldValue = 0;
   static double oldChange = 0;
   int rawValue = analogRead (sensorPin);
   double value = alpha * oldValue + (1 - alpha) * rawValue;

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

   oldValue = value;
   delay (period);
}
```

****

36. KEYES Tracking Module



LED lights use 13 PIN and Tracking sensor combines to make a warning light

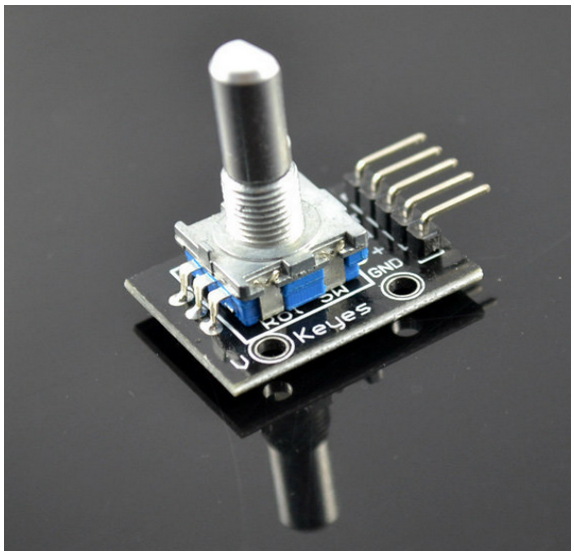
Routines source code:

```
int Led = 13 ;/ / define LED Interface
int buttonpin = 3; / / define Tracking Sensor Interface
int val ;/ / define numeric variables val
void setup ()
{
pinMode (Led, OUTPUT) ;/ / define LED as output interface
pinMode (buttonpin, INPUT) ;/ / define Tracking 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 Tracking sensor detects a signal, LED flashes
{
digitalWrite (Led, HIGH);
}
else
{
digitalWrite (Led, LOW);
}
}
```

****

037 Rotary encoder module



By rotating the rotary encoder can be counted in the positive direction and the reverse direction during rotation of the output pulse frequency, unlike rotary potentiometer counter, which
Species rotation counts are not limited. With the buttons on the rotary encoder can be reset to its initial state, that starts counting from 0.
How it works: incremental encoder is a displacement of the rotary pulse signal is converted to a series of digital rotary sensors. These pulses are used to control
Angular displacement. In Eltra angular displacement encoder conversion using a photoelectric scanning principle. Reading system of alternating light transmitting window and the window is not
Consisting of radial indexing plate (code wheel) rotating basis, while being an infrared light source vertical irradiation light to the code disk image onto the receiving
On the surface. Receiver is covered with a diffraction grating, which has the same code disk window width. The receiver's job is to feel the rotation of the disc
Resulting changes, and change the light into corresponding electrical changes. Then the low-level signals up to a higher level, and generates no interference

Square pulse, which must be processed by electronic circuits. Reading systems typically employ a differential manner, about the same but the phase difference of the two waveforms
Different by 180 ° compared to the signal in order to improve the quality and stability of the output signal. Reading is then the difference between the two signals formed on the basis,
Thus eliminating the interference.
Incremental encoder
Incremental encoders give two-phase square wave, the phase difference between them 90 °, often referred to as A and B channels. One of the channels is given and speed-related
Information, at the same time, by sequentially comparing two channel signals, the direction of rotation of the information obtained. There is also a special signal called Z or
Zero channel, which gives the absolute zero position encoder, the signal is a square wave with the center line of channel A square wave coincide.
Clockwise counterclockwise
A B
11
01
0 0
10
A B
11
10
0 0
01
Incremental encoder accuracy depends on the mechanical and electrical two factors, these factors are: Raster indexing error, disc eccentricity, bearing eccentricity, e-reading
Several means into the optical portion of the errors and inaccuracies. Determine the encoder resolution is measured in electrical degrees, the encoder accuracy depends
Set the pulse encoder generates indexing. The following electrical degrees with a 360 ° rotation of the shaft to said machine, and rotation of the shaft must be a full week of
Period. To know how much electrical equivalent of the mechanical angle of 360 degrees can be calculated with the following formula: Electrical 360 = Machine 360 ° / n ° pulses / revolution
Figure: A, B commutation signals
Encoder indexing error is the electrical angle of the unit two successive pulse maximum offset to represent. Error exists in any encoder, which
Is caused by the aforementioned factors. Eltra encoder maximum error is ± 25 electrical degrees (declared in any condition), equivalent to the rated
Offset values ± 7%, as the phase difference 90 ° (electrical) of the two channels of the maximum deviation ± 35 electrical degrees is equal to ± 10% deviation left Ratings
Right.
UVW incremental encoder signals
In addition to the conventional encoder, there are some other electrical output signal with integrated incremental encoder. And UVW signals
The integrated incremental encoder that instance, it is usually applied to the AC servo motor feedback. These magnetic signals generally appear in the AC servo motor

Machine, UVW through the simulation of the magnetic signal is generally of the original function and design. In Eltra encoder, these optical signals are UVW
Methods of generating, and three square wave form, are offset from each other 120 °. In order to facilitate starting the motor, the control of the starter motor to be
To these the correct signal. The UVW poles in the machine axis rotation pulses repeated many times, because they directly depend on the connected electrical
Machine number of poles, and for 6 or more pole motor UVW signal.
****
****
ARDUINO test code:

```
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);
}
```

****