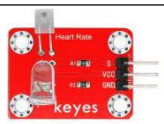
Keyes 37 in 1 Sensor Kit for Arduino

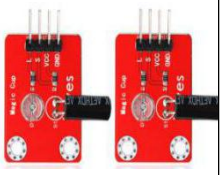
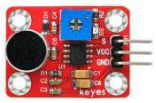
Content

1. Description

This sensor kit contains 37 kinds of commonly used sensor modules while playing the microcontroller, such as active buzzer module, 5V relay module, temperature and humidity module and so on. It is compatible for various microcontrollers and Raspberry Pi. At the same time, we also provide some learning courses for each sensor based on development board, including wiring method, test code, etc. Helping you to have further understanding of those sensor modules and development board.

2. Component List

No.	Component	Description	Quantity	Image
1	Keyes module	keyes LED cap white to white module (pad holes) red and eco-friendly	1	
2	Keyes module	keyes reed switch (pad holes) red and eco-friendly	1	
3	Keyes module	keyes dual-color LED module (pad holes) red and eco-friendly	1	
4	Keyes module	keyes heart rate module (pad holes) red and eco-friendly	1	
5	Keyes module	keyes LED RGB SMD module (pad holes) red and eco-friendly	1	
6	Keyes module	keyes active buzzer module (pad holes) red and eco-friendly	1	
7	Keyes module	keyes passive buzzer module (pad holes) red and eco-friendly	1	
8	Keyes module	keyes rotary encoder module (pad holes) red and eco-friendly	1	
9	Keyes module	keyes rotation pot module (pad holes) red and eco-friendly	1	

10	Keyes module	keyes 5V one relay module (pad holes) red and eco-friendly	1	
11	Keyes module	keyes RGB (pad holes) red and eco-friendly	1	
12	Keyes sensor	keyes thermistor sensor (pad holes) red and eco-friendly	1	
13	Keyes sensor	keyes button switch sensor (pad holes) red and eco-friendly	1	
14	Keyes sensor	keyes magical cup sensor (pad holes) red and eco-friendly	2	
15	Keyes sensor	keyes DHT11 temp.and humidity sensor (pad holes) red and eco-friendly	1	
16	Keyes sensor	keyes photocell sensor (pad holes) red and eco-friendly	1	
17	Keyes sensor	keyes tilt module sensor (pad holes) red and eco-friendly	1	
18	Keyes sensor	keyes photo-interrupter sensor (pad holes) red and eco-friendly	1	
19	Keyes sensor	keyes ADXL345 acceleration sensor (pad holes) red and eco-friendly	1	
20	Keyes sensor	keyes microphone sound sensor (pad holes) red and eco-friendly	1	
21	Keyes sensor	keyes Hall sensor (pad holes) red and eco-friendly	1	

22	Keyes sensor	keyes Crash sensor (pad holes) red and eco-friendly	1	
23	Keyes sensor	keyes IR transmitter sensor (pad holes) red and eco-friendly	1	
24	Keyes sensor	keyes ultrasonic sensor	1	
25	Keyes sensor	keyes MQ-2 gas sensor (pad holes) red and eco-friendly	1	
26	Keyes sensor	keyes knock module sensor (pad holes) red and eco-friendly	1	
27	Keyes sensor	keyes capacitive touch sensor (pad holes) red and eco-friendly	1	
28	Keyes sensor	keyes IR receiver sensor (pad holes) red and eco-friendly	1	
29	Keyes sensor	keyes joystick module sensor (pad holes) red and eco-friendly	1	
30	Keyes sensor	keyes MQ-3 alcohol sensor (pad holes) red and eco-friendly	1	
31	Keyes sensor	keyes obstacle detector sensor (pad holes) red and eco-friendly	1	
32	Keyes sensor	keyes LM35 temperature sensor (pad holes) red and eco-friendly	1	
33	Keyes sensor	keyes PIR motion sensor (pad holes) red and eco-friendly	1	
34	Keyes sensor	keyes laser sensor module (pad holes) red and eco-friendly	1	

35	Keyes sensor	keyes line tacking sensor (pad holes) red and eco-friendly	1	
36	Keyes sensor	keyes 18B20 temperature sensor (pad holes) red and eco-friendly	1	
37	Keyes sensor	keyes TEMT6000 ambient light sensor (pad holes) red and eco-friendly	1	

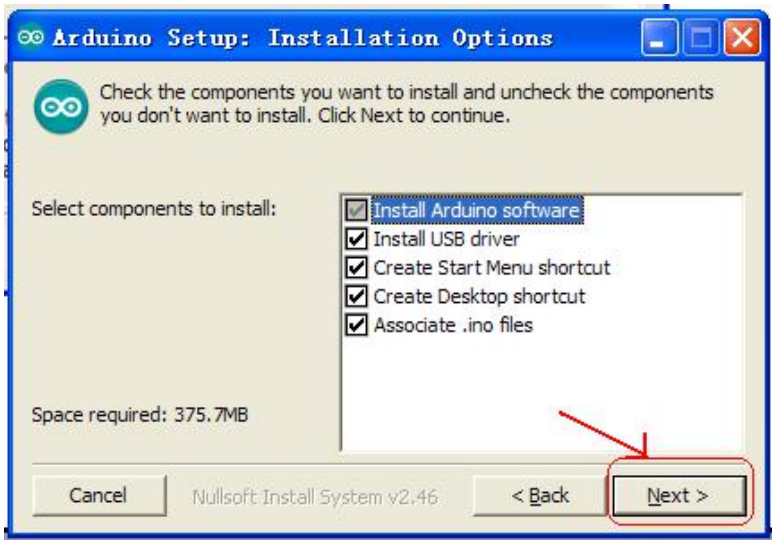
3. Arduino IDE and Driver Installation

When getting the keyes development board, first of all you have to install the Arduino IDE and the driver, and all relevant files can be found on the official website.
The following link includes various systems, various versions of the Arduino IDE and drivers whatever you choose.
https://www.arduino.cc/en/Main/OldSoftwareReleases#1.5.x

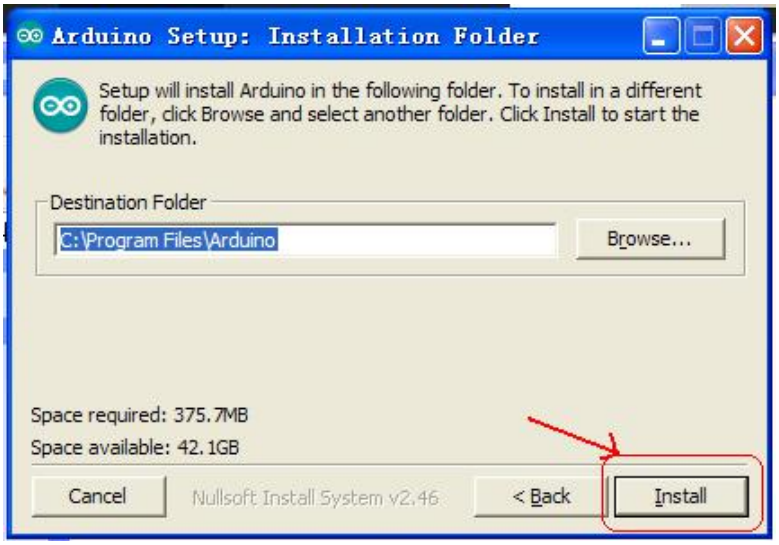
Next, we first introduce the installation method of Arduino IDE-1.5.6 version in the Windows system.

The file downloaded is an arduino-1.5.6-r2-windows.zip compression folder, please unzip it to the hard disk. Double-click Arduino-1.5.6 .exe file.
Please refer to the following setup figures:

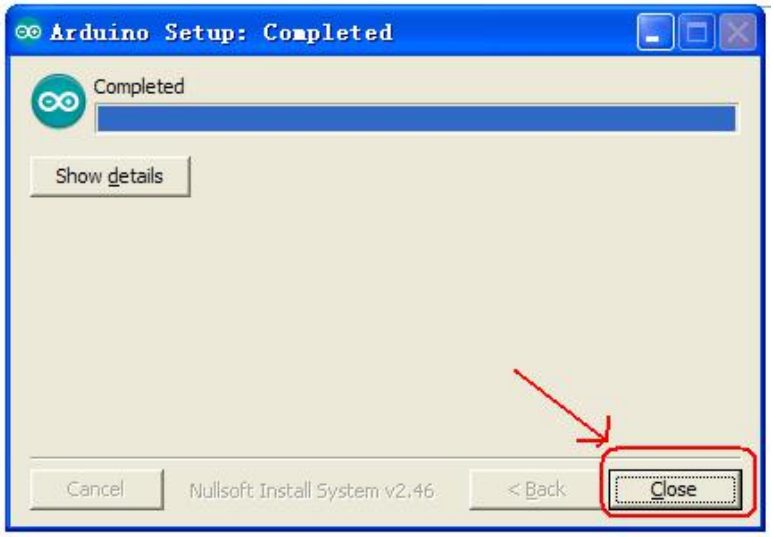
Click "I Agree". Then, click "Next"



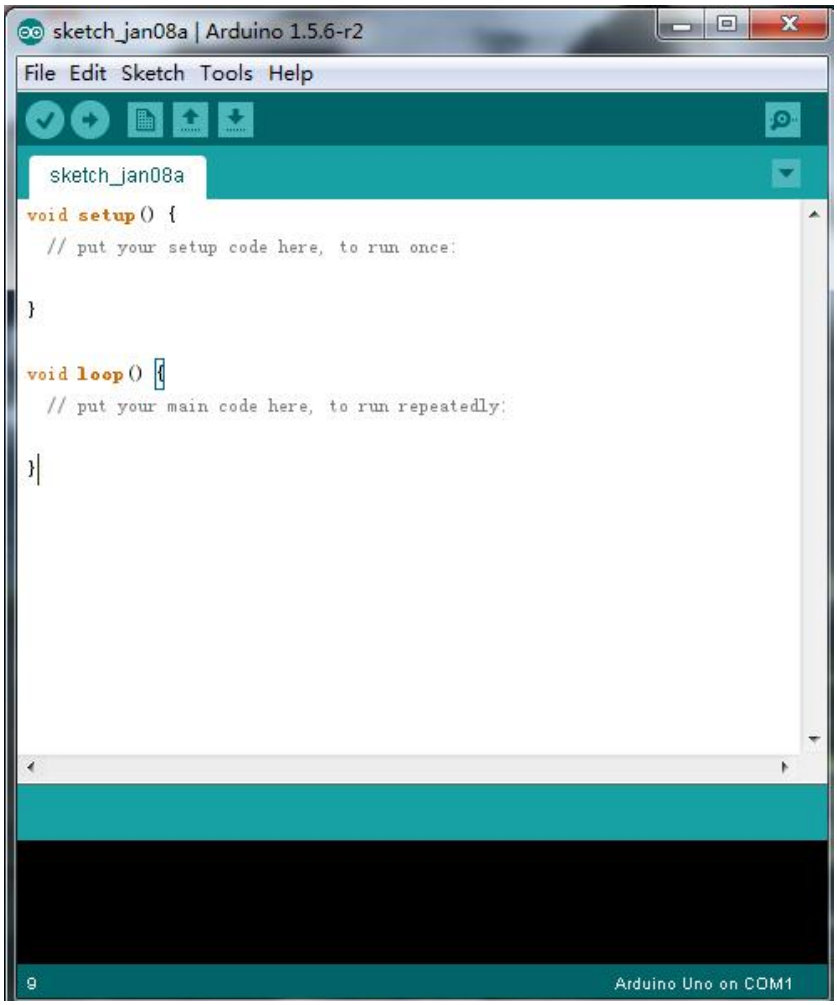
Next, click "Install".



Finally, click "Close" after completing the installation.



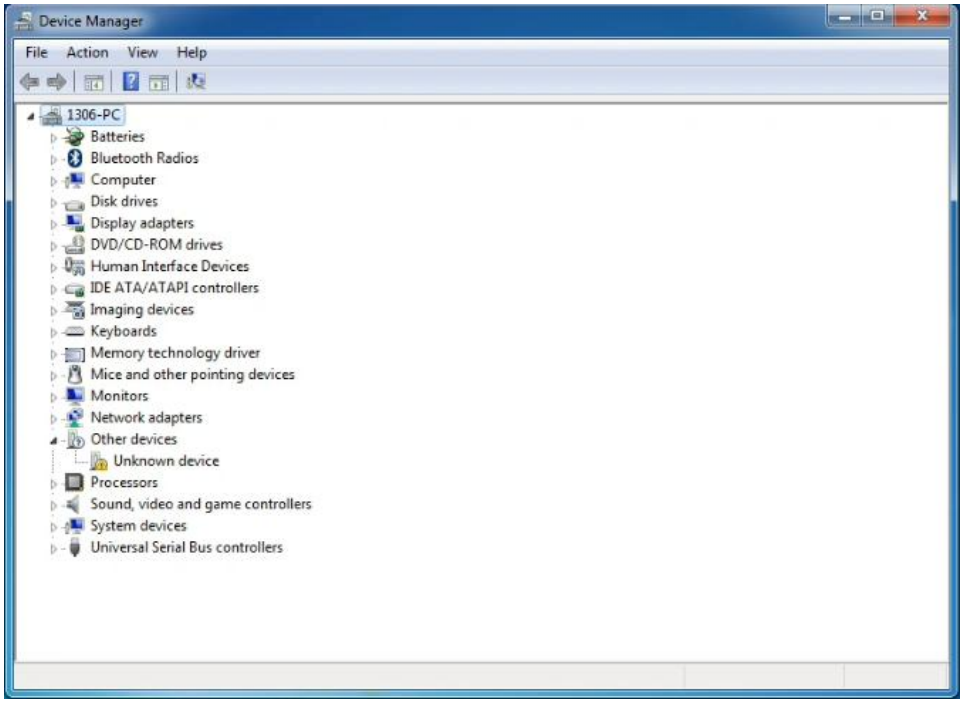
The figure below shows the successful installation of 1.5.6 version:



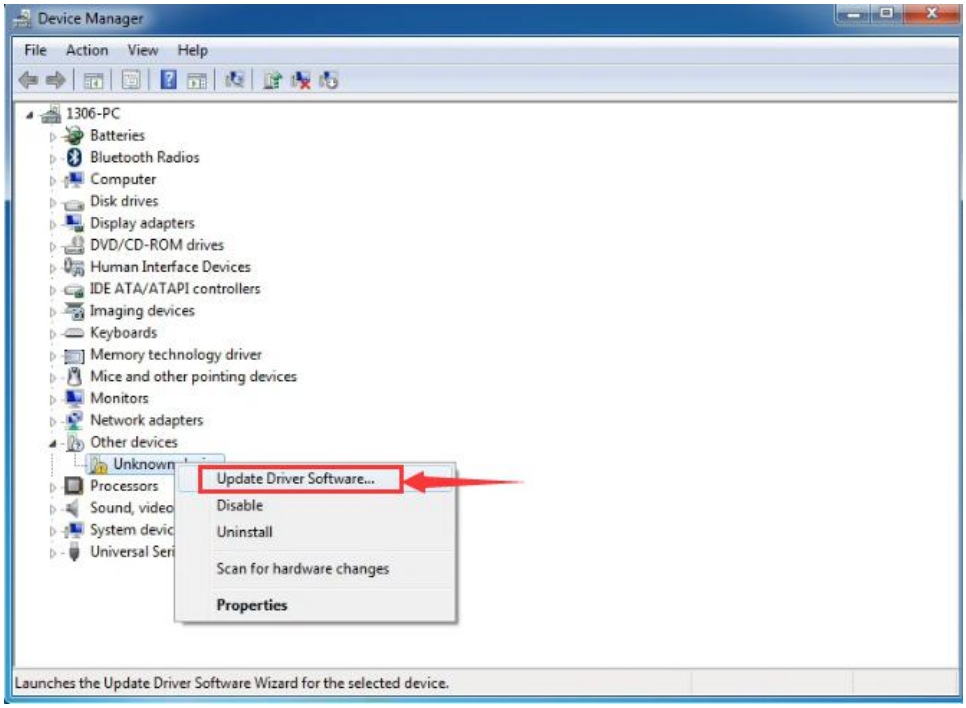
Next, we will introduce the driver installation of Keyes UNO R3 development board. Actually you can use the same method to install the driver of Keyes 2560 R3 development board.

Let's move on to the driver installation in the WIN 7 system.

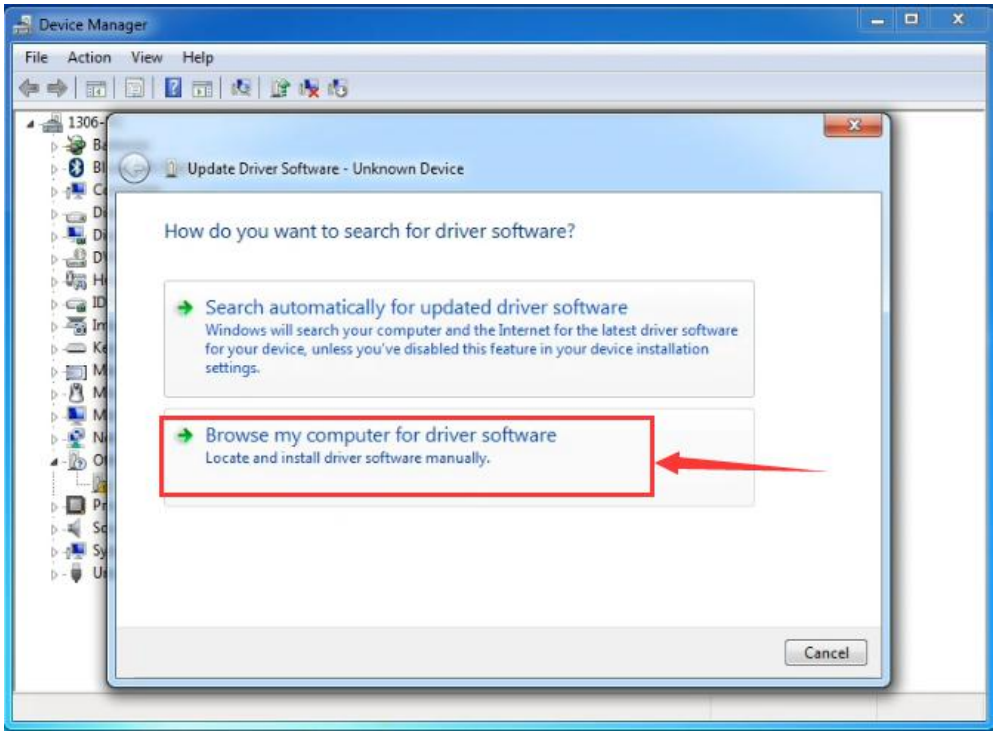
A. When you connect Keyes UNO R3 to your computer at the first time, right click "Computer" —>"Properties"—> "Device manager", you can see "Unknown devices".



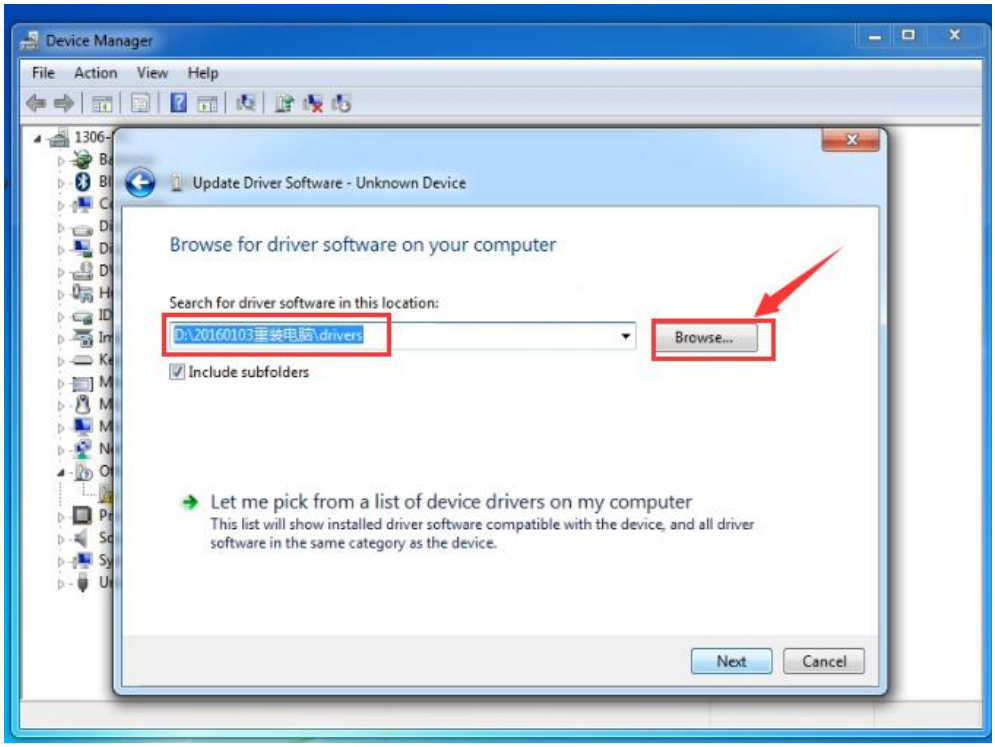
B. Click "Unknown devices", select "Update Driver software".



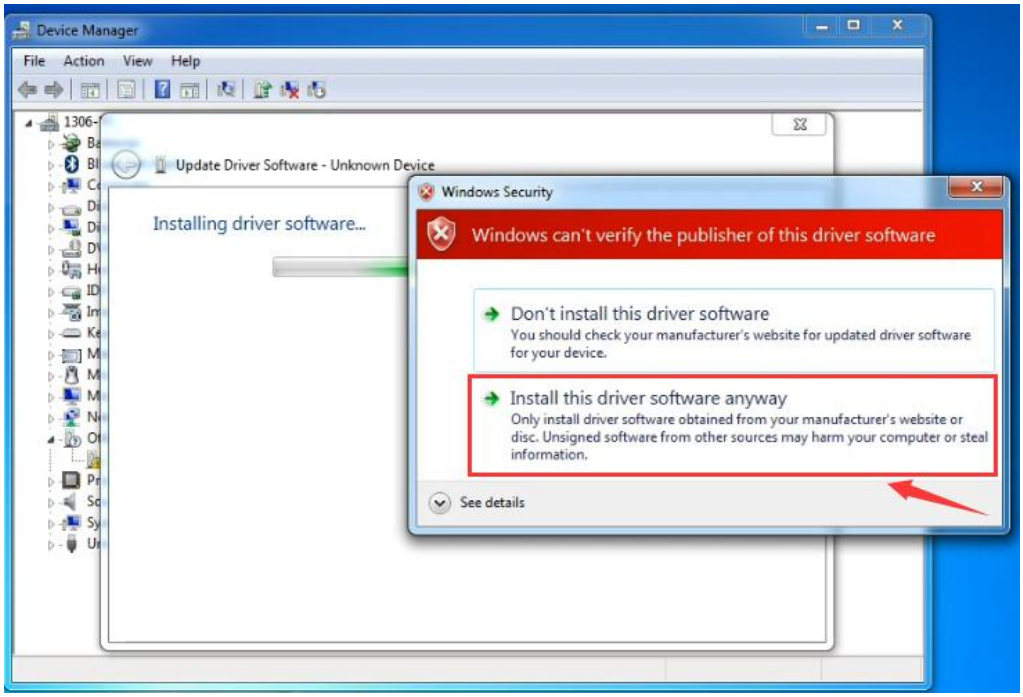
C. In this page, click “Browse my computer for driver software”.



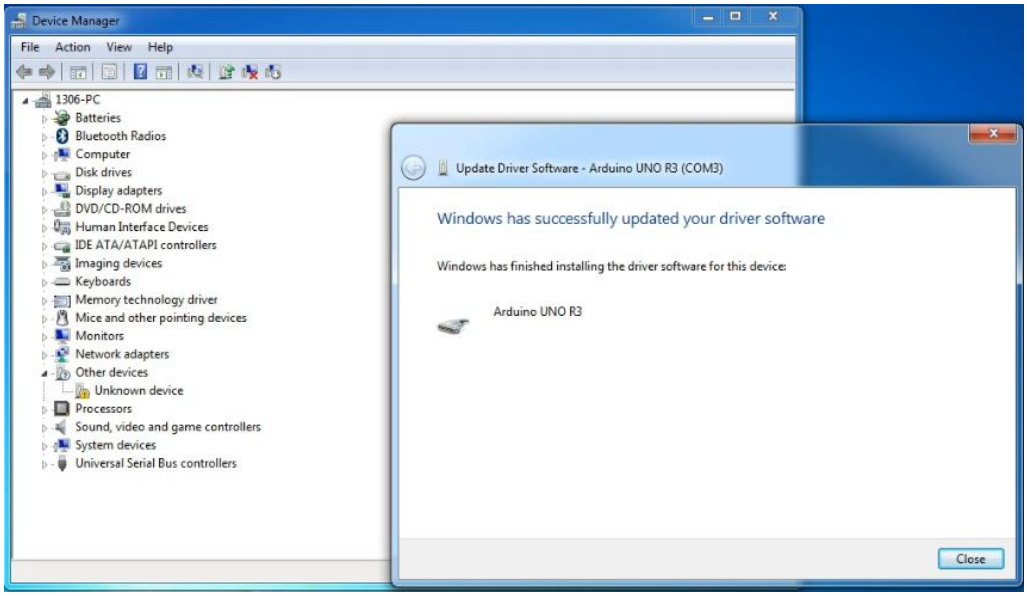
D. Find the “drivers” folder.



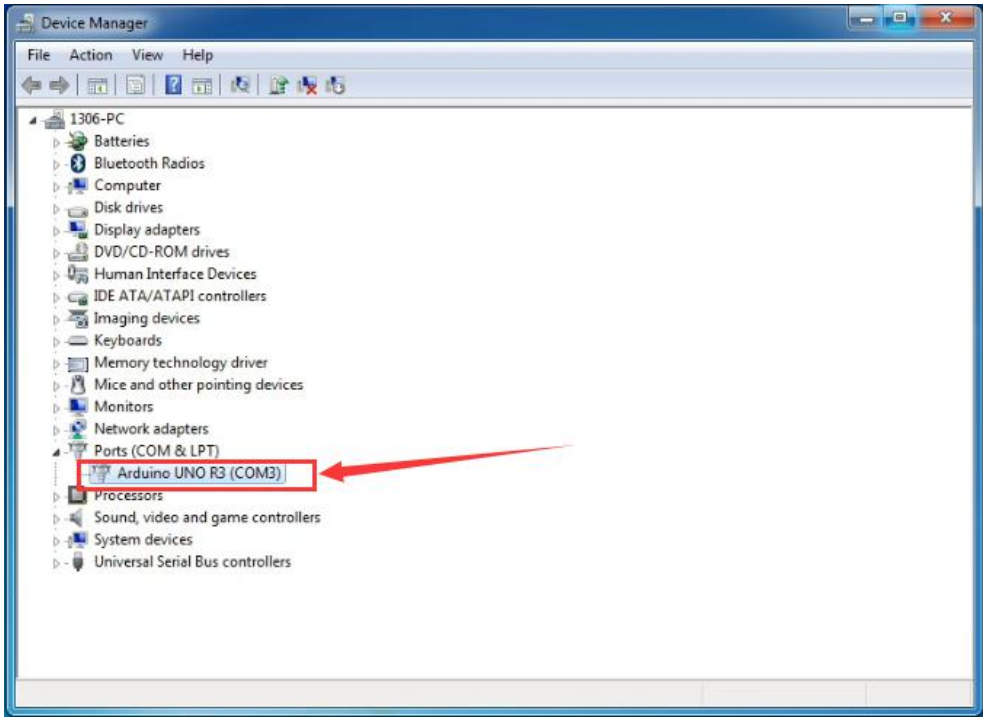
E. Click “Next”; select “Install this driver software anyway” to begin the installation.



F. Installation completed, click "Close".



G. After installation, go to see the "Device manager" again. right click "Computer" —> "Properties"—> "Device manager", you can see the device as below figure shown.



4. Using Method of Arduino IDE

When successfully installing the USB driver of Keyes UNO R3 development board, you can find the corresponding serial port in Windows Device Manager.
Next, we will show you the first program showing the "Hello World!" on the serial monitor.

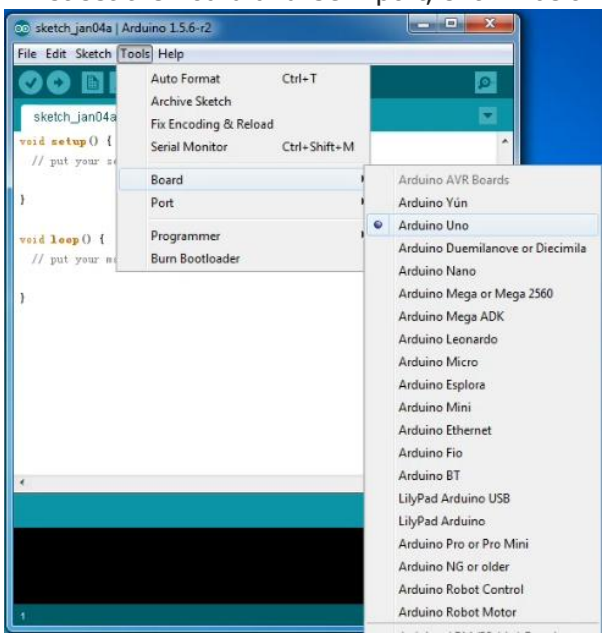
Sample Code as below:

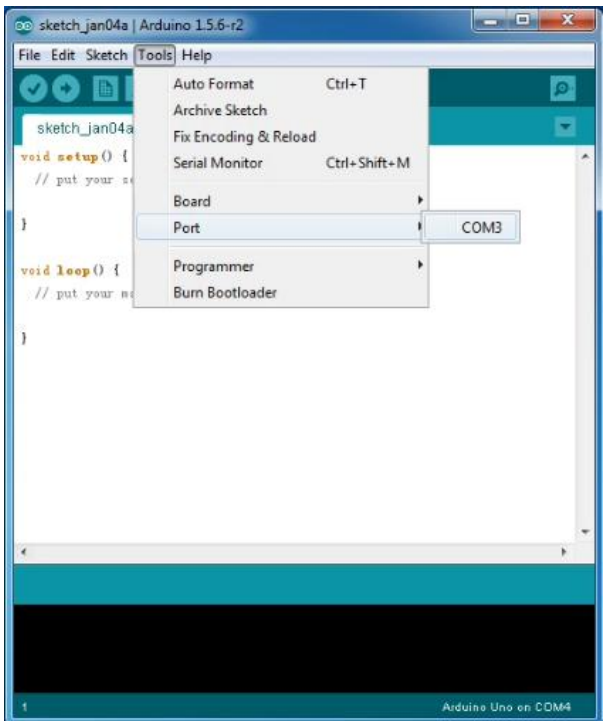
```
//////////////////////////////////////////
int val;
int ledpin=13;
void setup()
{
Serial.begin(9600);
pinMode(ledpin,OUTPUT);
}
void loop()
{
val=Serial.read();
if(val=='R')
{
digitalWrite(ledpin,HIGH);
delay(500);
digitalWrite(ledpin,LOW);
```

```
delay(500);
Serial.println("Hello World!");
}
}
//////////////////////////////////////////
```

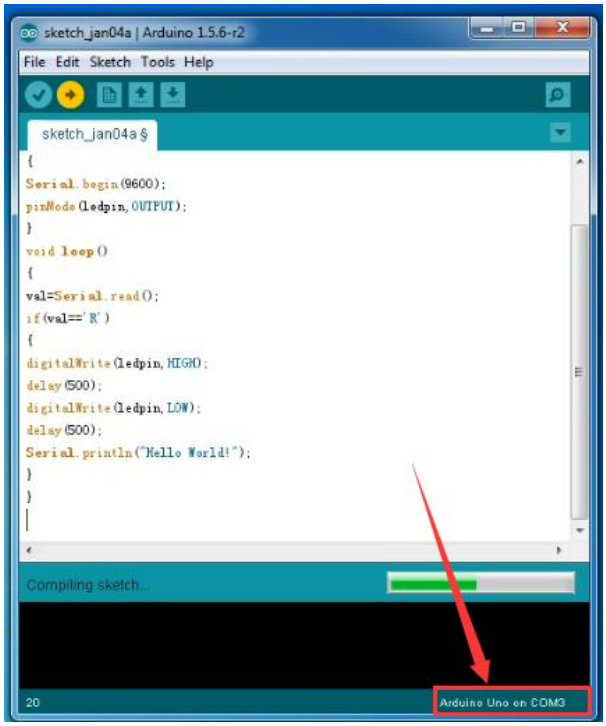
Open Arduino software, print the program to make keyes UNO R3 development board display the character "Hello World！ ". When the board receives the instruction, D13 indicator light on the board blinks and "Hello World！ " is displayed on the monitor.

First set the Board and COM port, shown below.



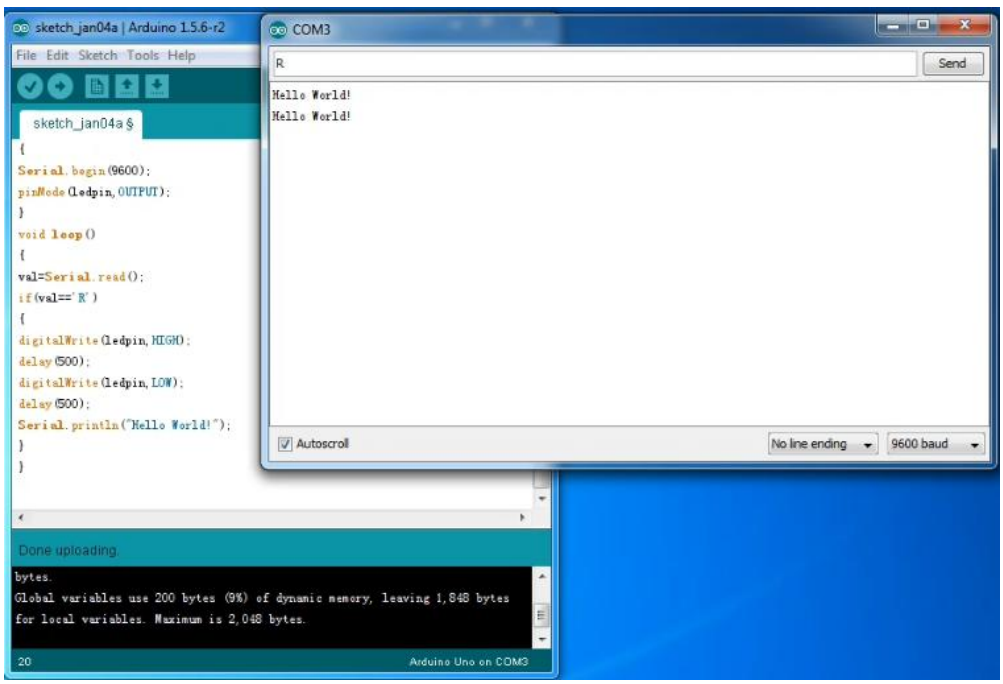


If setting well the board and port, you can see the display on the bottom right corner, which is the same as the Device Manager display.



Then, click the verify to compile the sketch, if no mistake, click upload to upload the program.

Done uploading, open the serial monitor, print an "R" and click "Send", you can see the D13 indicator on the Keyes UNO R3 development board blinks once, and the "Hello World!" is displayed on the serial monitor. Shown below.



Congrats. Your first programming is done well!

5. Example Projects

Project 1: LED Cap

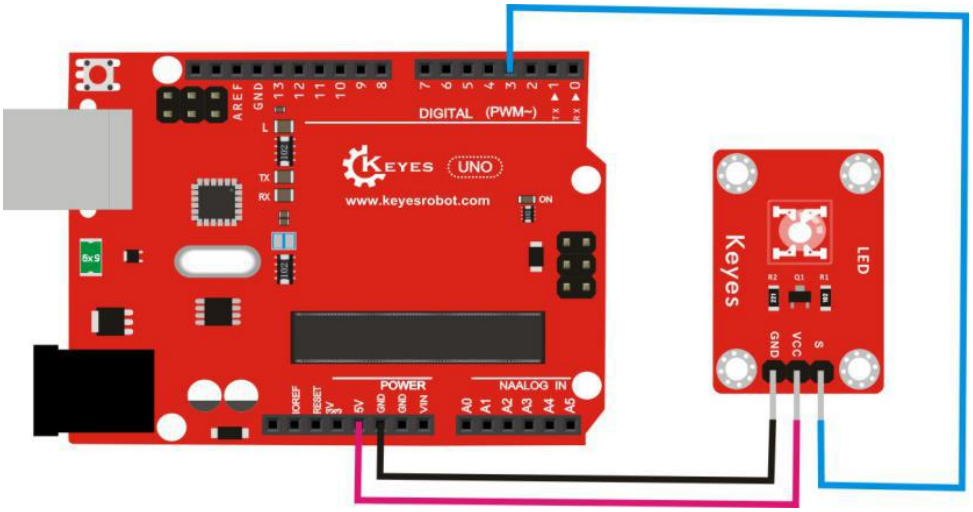
Description

Now we are going to detect this LED cap module. In the experiment, we connect the signal pin of LED module to digital port 3 of UNO board, which is also the PWM port. There are two choices to detect it. One is to make the LED blink. The other is to control the brightness of LED by PWM port, letting the cap LED gradually become bright and gradually darken so as to simulate the phenomenon like breathing.

Part List

Development Board*1
USB Cable*1
LED cap module*1
Jumper Wire*3

Wiring Diagram



Sample Code

Code A:

```
//////////////////////////////////////////////////////////
int led = 3;                          //set the digital 3
void setup()
{
```

```
  pinMode(led, OUTPUT);      //set led as output
}
void loop()
{
  digitalWrite(led, HIGH);    //turn on led
    delay(1000); //delay 1 second
  digitalWrite(led, LOW);     //turn off led
  delay(1000);//delay 1 second
}
//////////////////////////////////////////////////////////
```

Code B:

```
//////////////////////////////////////////////////////////
int ledPin = 3; // set the digital 3
void setup()
{
pinMode(ledPin, OUTPUT);// set ledPin as output
}
void loop()
{
for (int a=0; a<=255;a++)// make LED gradually brighten
{
analogWrite(ledPin,a); // turn on led, adjust the brightness, range is within 0-255, led is the brightest at 255.
delay(10); // delay 0.01 second
}
for (int a=255; a>=0;a--) // make LED gradually darken
{
analogWrite(ledPin,a); //    turn on led, adjust the brightness, range is within 0-255, led is the brightest at 255.
delay(10); // delay 0.01 second
}
delay(1000);// delay 1 second
}
//////////////////////////////////////////////////////////
```

Phenomenon

Wire it up and upload well the code A, you can see the LED module continue to blink, with an interval about one second. If upload the code B, you can see the LED first gradually brighten, then gradually dims, alternately and circularly.

Project 2: Reed Module

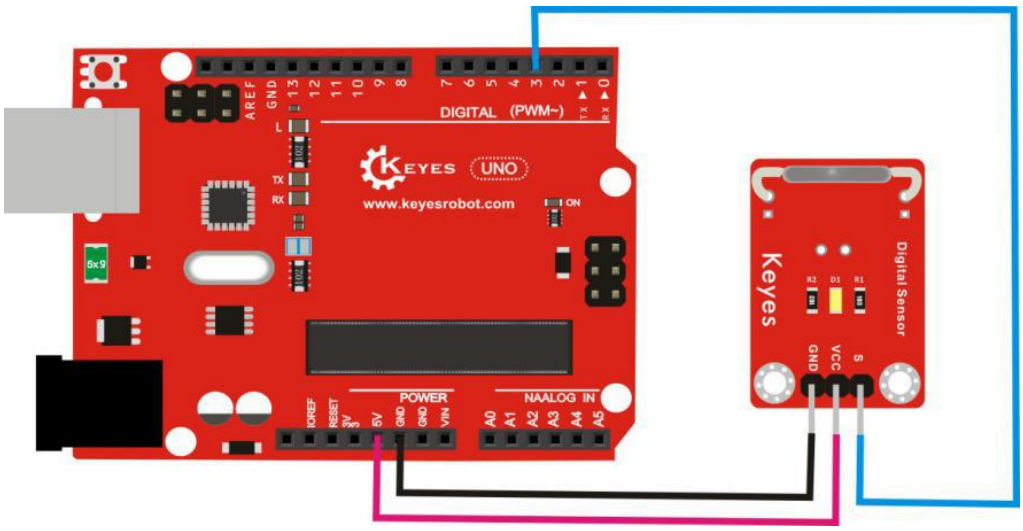
Description

Now we are going to detect the reed module. The module is mainly composed of a reed pipe. Powered up, the signal end of module outputs high level, and the LED light on the sensor dims. When adding a magnetic field to the module, its signal end outputs low level and LED becomes bright. In this experiment, we will make use of built-in D13 indicator on the UNO board, to control the D13 indicator on and off via sensor.

Part List

Development Board*1
USB Cable*1
Reed module*1
Jumper Wire*3

Wiring Diagram



Sample Code

```
//////////////////////////////////////////////////////////
int Led=13;//set the digital 13
int buttonpin=3; //set the digital 3
int val;//set the digital variable val
void setup()
{
pinMode(Led,OUTPUT);//set Led as output
pinMode(buttonpin,INPUT);//set buttonpin as input
  }
void loop()
```

```
{

val=digitalRead(buttonpin);// read the value of digital and assign it to val
if(val==LOW)//if val is low level
{
digitalWrite(Led,HIGH);     //LED lights up
}
else
{
digitalWrite(Led,LOW);     //LED goes out
}
}
//////////////////////////////////////////////////////
```

Phenomenon

Hook up and upload the code well, D13 indicator on the UNO board is off, and D1 led on the module is also off. When there is a magnet close to the module, D13 indicator on the UNO board is on, and D1 led on the module is also on.

Project 3: Dual-Color LED

Description

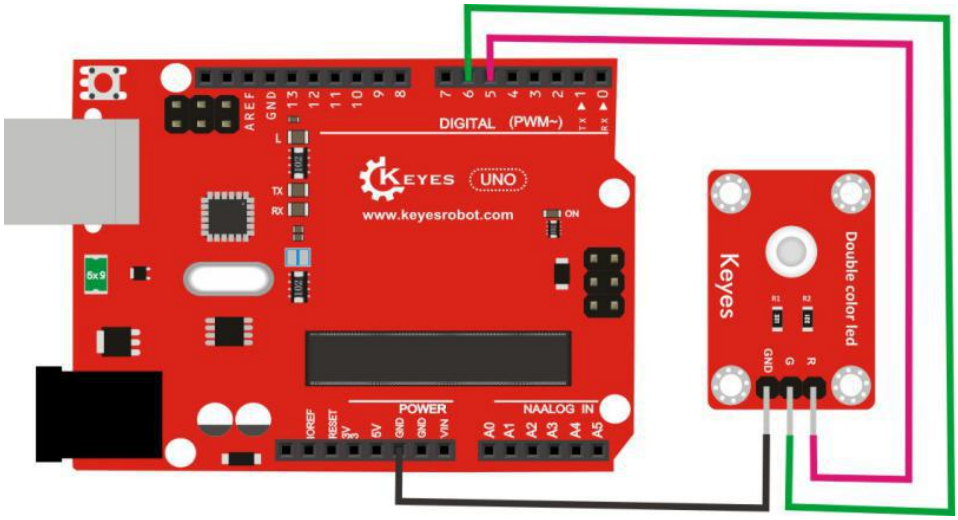
Next we are going to detect this dual color LED module. The module is driven by common cathode, with emitting colors Red and Green. By two PWM ports, we can adjust the proportion of red and green light display to mix into other colors.

In this experiment, connect the signal pins of dual color module to digital 5 and 6 of UNO, finally make the dual color module display red and green light circularly.

Part List

Development Board*1
USB Cable*1
Dual-color LED module*1
Jumper Wire*3

Wiring Diagram



Sample Code

```
//////////////////////////////////////////////////////////
int redPin = 5;   // control pin of Red LED is connected to Arduino pin 5
int greenPin = 6;   // control pin of green LED is connected to Arduino pin 6
void setup()
{
        pinMode(redPin, OUTPUT); //set redPin 5 as output
        pinMode(greenPin, OUTPUT); //set greenPin 6 as output
}

void loop()
{
         // Basic colors:
        color(255, 0); // red light on
        delay(1000); // delay one second
        color(0,255); //green light on
        delay(1000); // delay one second
}
void color (unsigned char red, unsigned char green)   //color control function
{
         analogWrite(redPin, red);
         analogWrite(greenPin,green);
}
//////////////////////////////////////////////////////////
```

Phenomenon

Powered up and upload well the code, you will see the dual color LED module display red light for one second, green light for one second, alternately and circularly.

Project 4: Heart Rate

Description

The module mainly consists of an infrared transmitter and a photosensitive receiver. The infrared transmitter is on one side of the finger, while photosensitive receiver is on the other side of the finger.

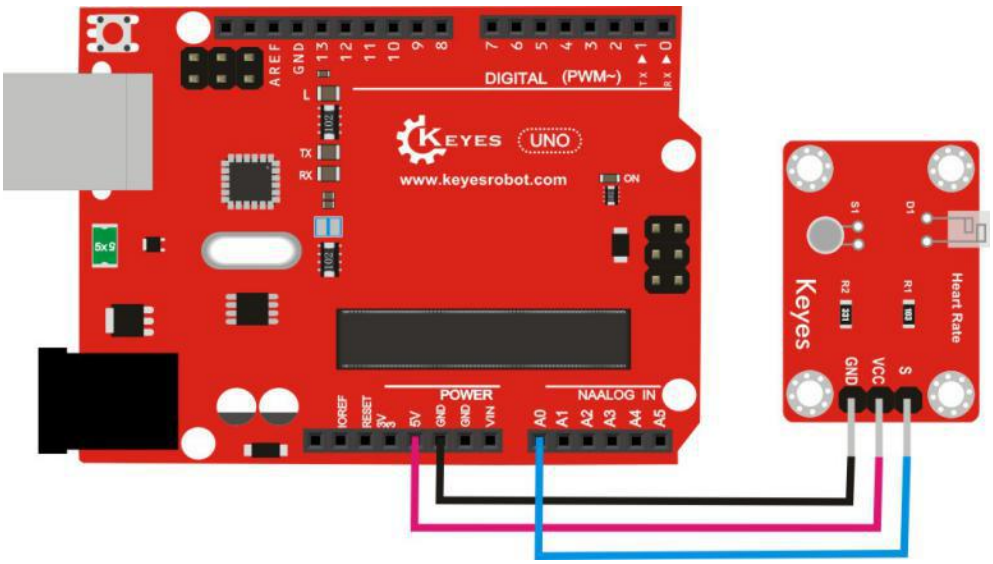
The photosensitive receiver is used to capture the emitted light flux. When the blood pressure pulsates through the finger, the resistance of photosensitive receiver changes slightly.

In this project, connect signal end of sensor to analog A0 of UNO board, so as to display the corresponding data on the serial monitor of Arduino IDE.

Part List

Development Board*1
USB Cable*1
Heart rate module*1
Jumper Wire*3

Wiring Diagram



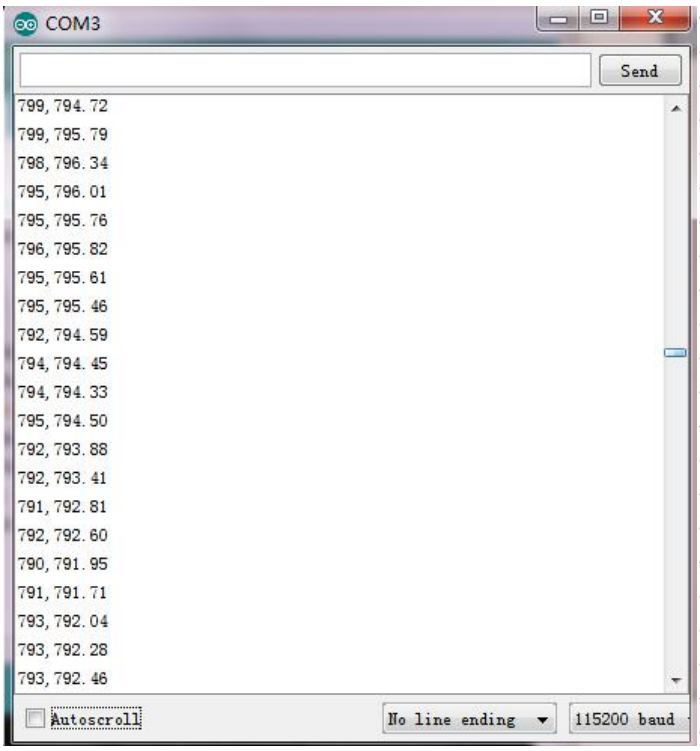
Sample Code

```
//////////////////////////////////////////////////////////
int ledPin = 13;
int sensorPin = 0;
double alpha = 0.75;
int period = 20;
double change = 0.0;
void setup()
```

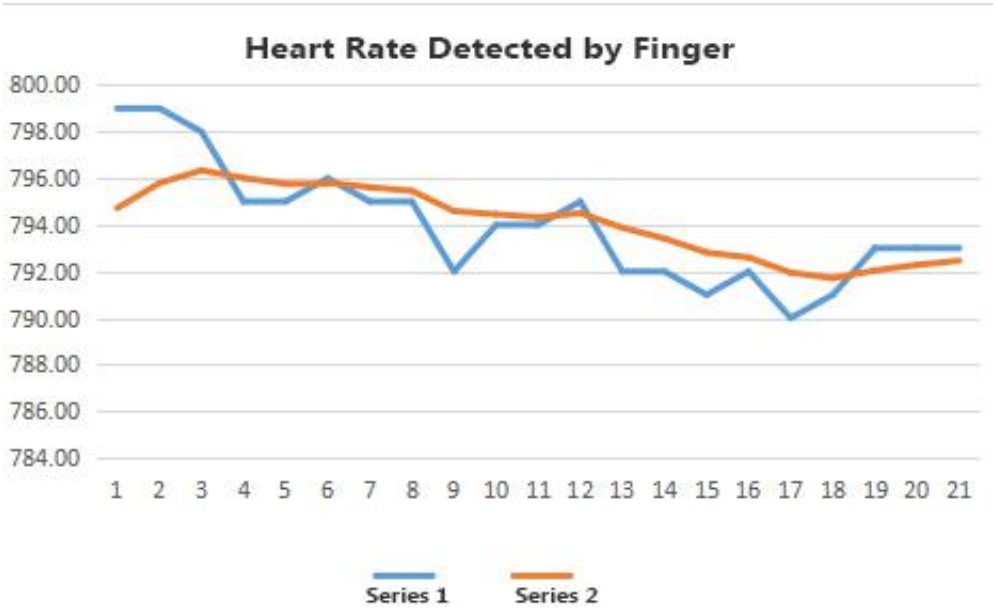
```
{
pinMode(ledPin, OUTPUT);
Serial.begin(115200);
}
void loop()
{
static double oldValue = 0;
static double oldChange = 0;
int rawValue = analogRead(sensorPin);
double value = alpha * oldValue + (1 - alpha) * rawValue;
Serial.print(rawValue);
Serial.print(",");
Serial.println(value);
oldValue = value;
delay(period);
}
//////////////////////////////////////////////////////////
```

Phenomenon

Powered up and upload well the code, then open the serial monitor and set the baud rate as 115200, you will see the corresponding data. Directly copy the data to excel, it will be generated into picture. Shown below.



799.00	794.72
799.00	795.79
798.00	796.34
795.00	796.01
795.00	795.76
796.00	795.82
795.00	795.61
795.00	795.46
792.00	794.59
794.00	794.45
794.00	794.33
795.00	794.5
792.00	793.88
792.00	793.41
791.00	792.81
792.00	792.6
790.00	791.95
791.00	791.71
793.00	792.04
793.00	792.28
793.00	792.46



Project 5: SMD RGB

Description

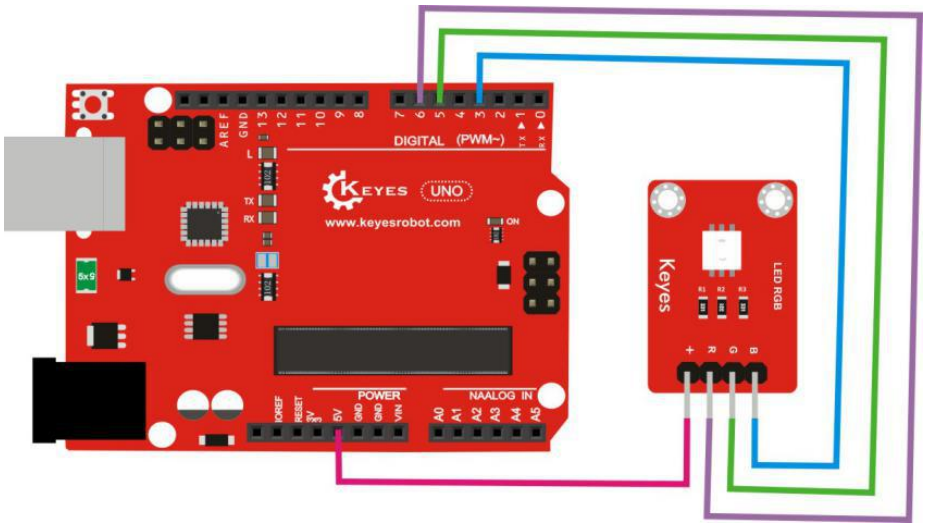
This RGB module is formed with one full-color LED SMD. The PWM voltage input of R, G, B pins can adjust the intensity of three primary colors (red/blue/green) to realize the full-color mixing effect.

This project we use the UNO R3 board to control the LED RGB module to achieve cool lighting effects, displaying different colors circularly.

Part List

Development Board*1
USB Cable*1
SMD RGB module*1
Jumper Wire*4

Wiring Diagram



Sample Code

```
//////////////////////////////////////////////////////////
int redPin = 6;   // Red LED control pin is connected to Arduino pin 6
int greenPin = 5;   // green LED control pin is connected to Arduino pin 5
int bluePin = 3;   // blue LED control pin is connected to Arduino pin 3
void setup()
{
        pinMode(redPin, OUTPUT); //set redPin 6 as output
        pinMode(greenPin, OUTPUT); //set greenPin 5 as output
        pinMode(bluePin, OUTPUT); //set bluePin 3 as output
}

void loop()
{
        // Basic colors:
      color(255, 0, 0); // red light on
      delay(1000); // delay one second
      color(0,255, 0); //green light on
      delay(1000); //delay one second
      color(0, 0, 255); // blue light on
      delay(1000); //delay one second

        // Example blended colors:
        color(255,255,0); // yellow light on
        delay(1000); //delay one second
        color(128,0,255); // purple light on
        delay(1000); //delay one second
        color(255,255,255); // white light on
        delay(1000); //delay one second
        color(0,0,0); // turn off led
```

```
        delay(1000); //delay one second
}
void color (unsigned char red, unsigned char green, unsigned char blue)  //color control function
{
        analogWrite(redPin, 255-red);
        analogWrite(greenPin,255-green);
        analogWrite(bluePin, 255-blue);
}
//////////////////////////////////////////////////////////
```

Phenomenon

Powered up and upload well the code, RGB module will continue to display the red light for one second, green light for one second, blue light for one second, yellow light for one second, purple light for one second, and white light for one second, then stop display for one second, alternately and circularly.

Project 6: Active Buzzer

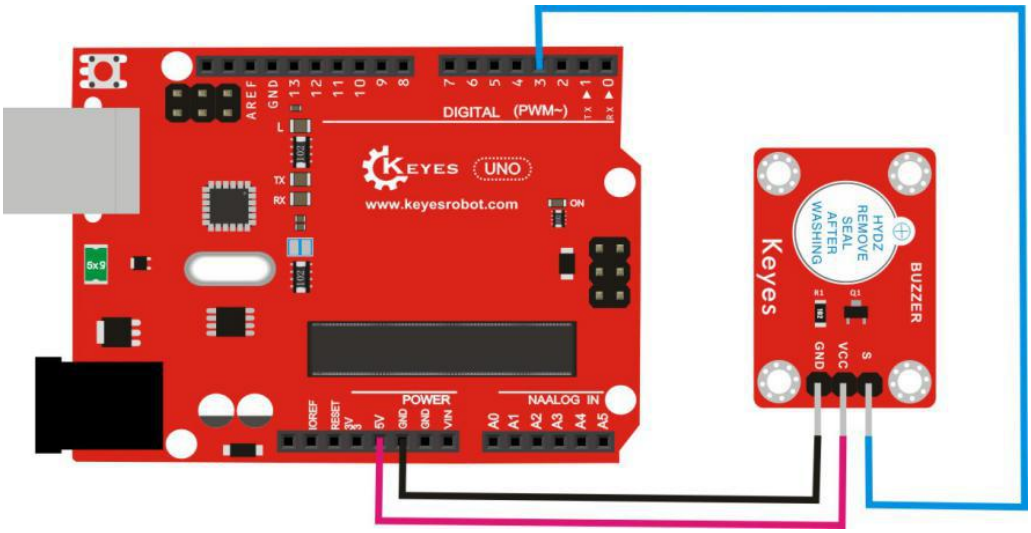
Description

Now we are going to detect the active buzzer module, making a sound circularly. This module is mainly composed of an active buzzer device. It is an integrated structure of electronic sounder and is powered by a DC power supply. After the module is connected to the power supply, the buzzer will beep when directly input a high level signal to its signal terminal.

Part List

Development Board*1
USB Cable*1
Active buzzer module*1
Jumper Wire*3

Wiring Diagram



Sample Code

```
//////////////////////////////////////////////////////////
int buzzPin = 3;     //define the digital 3
 void setup()
 {
  pinMode(buzzPin, OUTPUT);   //set buzzPin as output
}
 void loop()
{
  digitalWrite(buzzPin, HIGH);//active buzzer beeps
  delay(2000); //delay 2 seconds
  digitalWrite(buzzPin, LOW);    //active buzzer not sound
  delay(2000);  //delay 2 seconds
}
//////////////////////////////////////////////////////////
```

Phenomenon

Powered up and upload well the code, active buzzer will beep for two seconds, then stop beeping for two seconds, alternately and circularly.

Project 7: Passive Buzzer

Description

The passive buzzer module does not have an oscillation source inside, and the DC signal cannot make it buzz. It must be driven by a square wave.

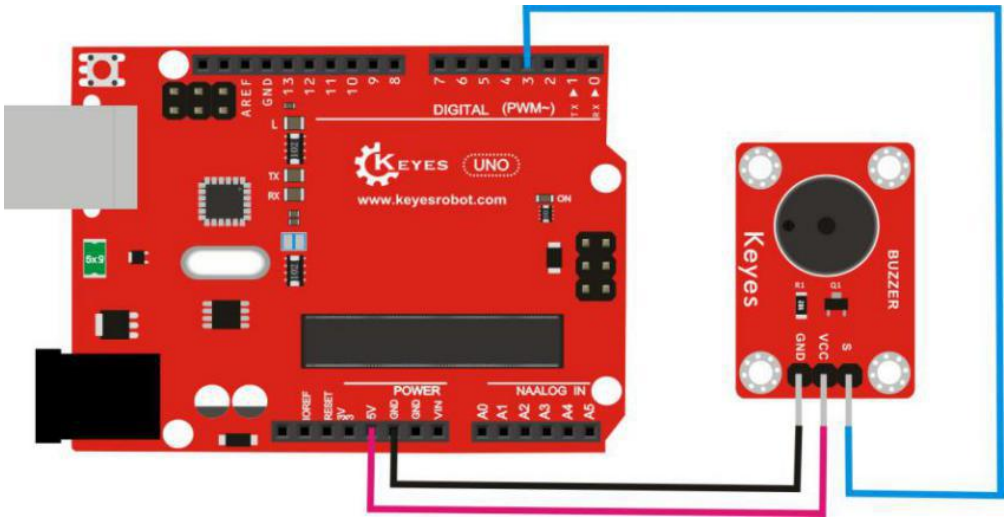
In the experiment, we connect the signal end of passive buzzer to the digital port 3 of UNO board. This way control the digital port 3 to output the square wave, thereby driving the passive buzzer.

In the experiment, one is to let the digital port 3 output a square wave of two frequencies to drive the passive buzzer to sound. The other is that let the digital port 3 output a square wave of various frequencies and set well rhythm, so as to allow the passive buzzer to play the song of Ode to Joy.

Part List

Development Board*1
USB Cable*1
Passive buzzer module*1
Jumper Wire*3

Wiring Diagram



Sample Code

Code A:

```
//////////////////////////////////////////////////////////
int buzzer=3;                //define the digital 3
void setup()
{
pinMode(buzzer,OUTPUT);//set buzzer as output
```

```
}
void loop()
{
unsigned char i,j;//define variable i，j
while(1)
{
for(i=0;i<80;i++)// output the sound of a frequency
{
digitalWrite(buzzer,HIGH);
delay(1);//delay 1ms
digitalWrite(buzzer,LOW);
delay(1);//delay 1ms
}
for(i=0;i<100;i++)//   output the sound of another frequency
{
digitalWrite(buzzer,HIGH);
delay(2);//delay 2ms
digitalWrite(buzzer,LOW);
delay(2);//delay 2ms
}
}
}
//////////////////////////////////////////////////////////////
```

Code B:

```
//////////////////////////////////////////////////////////////
#define D0 -1
#define D1 262
#define D2 293
#define D3 329
#define D4 349
#define D5 392
#define D6 440
#define D7 494
#define M1 523
#define M2 586
#define M3 658
#define M4 697
#define M5 783
#define M6 879
#define M7 987
#define H1 1045
#define H2 1171
#define H3 1316
```

```
#define H4 1393
#define H5 1563
#define H6 1755
#define H7 1971
///list out the frequency of all D tone
#define WHOLE 1
#define HALF 0.5
#define QUARTER 0.25
#define EIGHTH 0.25
#define SIXTEENTH 0.625
///list out all the beats
int tune[]=             // list out all the frequencies according to the numbered musical notation
{
  M3,M3,M4,M5,
  M5,M4,M3,M2,
  M1,M1,M2,M3,
  M3,M2,M2,
  M3,M3,M4,M5,
  M5,M4,M3,M2,
  M1,M1,M2,M3,
  M2,M1,M1,
  M2,M2,M3,M1,
  M2,M3,M4,M3,M1,
  M2,M3,M4,M3,M2,
  M1,M2,D5,D0,
  M3,M3,M4,M5,
  M5,M4,M3,M4,M2,
  M1,M1,M2,M3,
  M2,M1,M1
};
float durt[]=           //list out all the beats according to the numbered musical notation
{
  1,1,1,1,
  1,1,1,1,
  1,1,1,1,
  1+0.5,0.5,1+1,
  1,1,1,1,
  1,1,1,1,
  1,1,1,1,
  1+0.5,0.5,1+1,
  1,1,1,1,
  1,0.5,0.5,1,1,
```

```
  1,0.5,0.5,1,1,
  1,1,1,1,
  1,1,1,1,
  1,1,1,0.5,0.5,
  1,1,1,1,
  1+0.5,0.5,1+1,
};
int length;
int tonepin=3;     // use pin3
void setup()
{
  pinMode(tonepin,OUTPUT);
  length=sizeof(tune)/sizeof(tune[0]);     //calculate the length
}
void loop()
{
  for(int x=0;x<length;x++)
  {
    tone(tonepin,tune[x]);
    delay(500*durt[x]);      //here used to adjust the delay time according to the beats. The number 500 can be adjusted.
    noTone(tonepin);
  }
  delay(2000);
}
//////////////////////////////////////////////////////////////
```

Phenomenon

Done wiring and powered up, upload the code A, passive buzzer will make two different sounds alternately. If upload the code B, the buzzer will play the song of Ode to Joy.

Project 8: Rotary Encoder

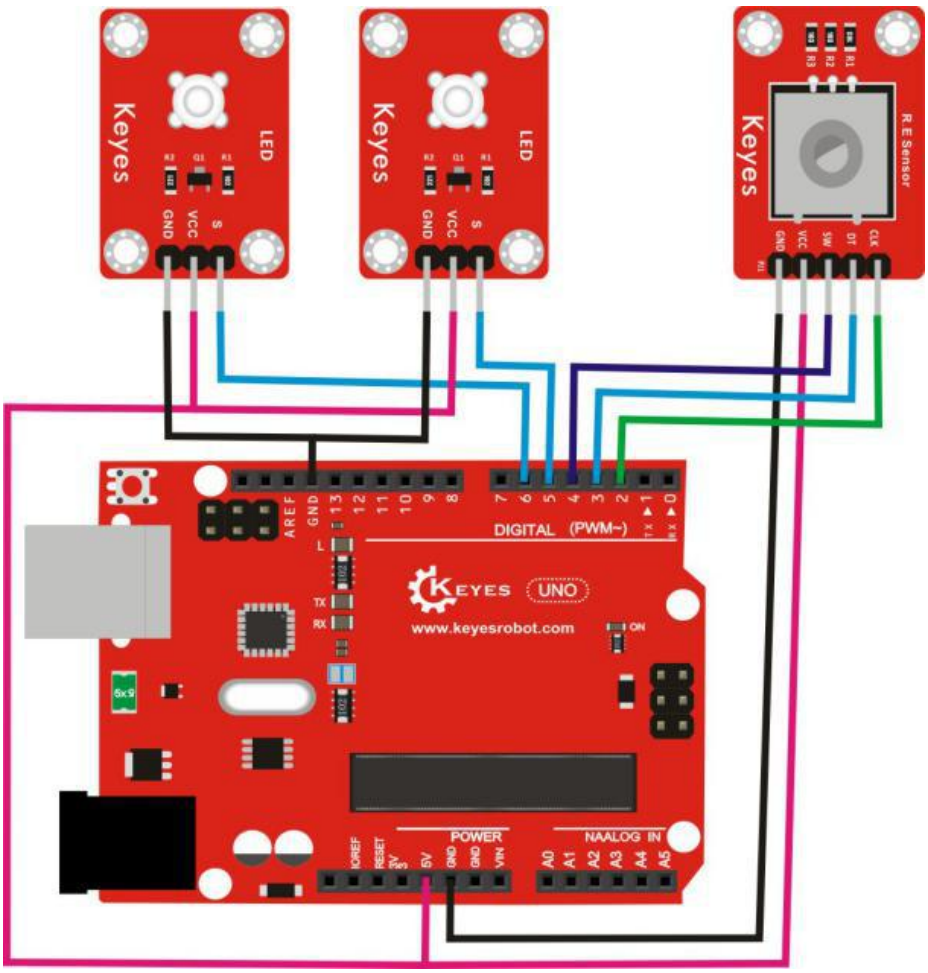
Description

The rotary encoder module is mainly composed of a rotary encoder, which can count the number of output pulses in the forward and reverse direction rotation. This rotation count is unlimited, if reset to the initial state, that is, counting starts from 0.
In the experiment we are going to use a rotary encoder module to control two LEDs on and off.

Part List

Development Board*1
USB Cable*1
Rotary encoder module*1
LED cap module *2
Jumper Wire*8

Wiring Diagram



Sample Code

```
//////////////////////////////////////////////////////
const int interruptA = 0; //interrupt 0 is in digital 2
const int interruptB = 1;//interrupt 1 is in digital 3
int CLK = 2; // define the digital 2
int DAT = 3; // define the digital 3
int BUTTON = 4; // define the digital 4
int LED1 = 5; // define the digital 5
int LED2 = 6; // define the digital 6
int COUNT = 0;//set the number variable COUNT as 0
void setup()
{
attachInterrupt(interruptA, RoteStateChanged, FALLING);
// when digital 2 is changed from high level to low level, trigger interrupt.
pinMode(CLK, INPUT);//set CLK as input
digitalWrite(2, HIGH); //   set the digital 2 as high level
pinMode(DAT, INPUT); //set DAT as input
digitalWrite(3, HIGH); //set the digital 3 as high level
pinMode(BUTTON, INPUT); //set BUTTON as input
digitalWrite(4, HIGH); //set the digital 4 as high level
pinMode(LED1, OUTPUT);//set LED1 as output
pinMode(LED2, OUTPUT);//set LED1 as output
Serial.begin(9600); //set the baud rate
}
void loop()
{
if (digitalRead(BUTTON)==LOW)//if digital 4 is low level
{
COUNT = 0; //set the number variable COUNT as 0
Serial.println("STOP COUNT = 0");//display the content
digitalWrite(LED1, LOW);//LED1 dims
digitalWrite(LED2, LOW);//LED2 dims
delay (2000);//delay 2S
}
Serial.println(COUNT);//display COUNT data
}
void RoteStateChanged() //when the digital 2 is changed from high level to low level
{
if (digitalRead(DAT)==HIGH) //   if digital 3 is high level
{
COUNT++;//number variable COUNT plus 1
digitalWrite(LED1, HIGH);//LED1 lights up
digitalWrite(LED2, LOW);//LED2 dims
delay(200);//delay 0.2S
```

```
}
else
{
COUNT--;//number variable COUNT subtract 1
digitalWrite(LED2, HIGH);//LED2 lights up
digitalWrite(LED1, LOW);//LED1 dims
delay(200);//delay 0.2S
}
}
//////////////////////////////////////////////////////////
```

Phenomenon

Done wiring and powered up, you can rotate the rotary encoder to control two LEDs on and off.

Project 9: Potentiometer

Description

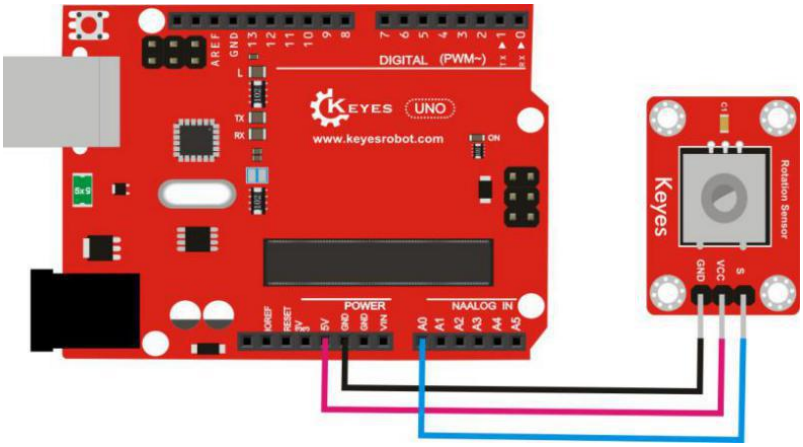
The rotation pot module is mainly composed of an adjustable potentiometer.

In the experiment, connect the signal pin of module to analog A0 of UNO board, powered up, you can rotate the potentiometer on the module to adjust the analog input value displayed on the serial monitor of Arduino IDE.

Part List

Development Board*1
USB Cable*1
Rotation pot module*1
Jumper Wire*8

Wiring Diagram

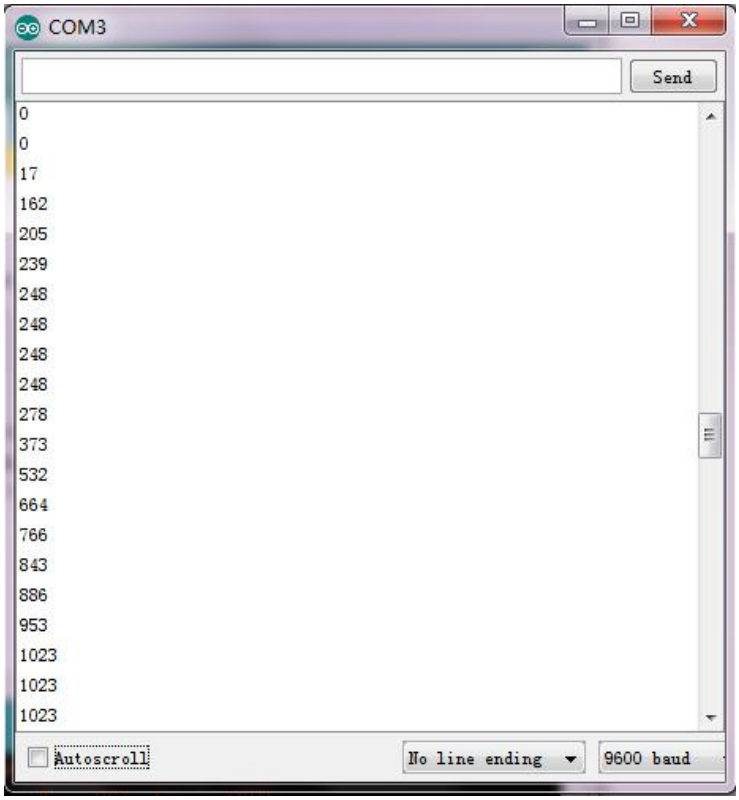


Sample Code

```
//////////////////////////////////////////////////////////
int sensorPin =A0 ;   //define the analog port A0
int value = 0;          //set value as 0
void setup()
{
 Serial.begin(9600);//set the baud rate

 }
void loop()
{
value = analogRead(sensorPin);     //set the value as the value of A0
Serial.println(value, DEC);       //display value and line wrap
delay(100);   //delay 0.1 second
}
//////////////////////////////////////////////////////////
```

Phenomenon

Wire it up as the above diagram, upload well the code, open the serial monitor, you will see the value of analog port A0, then rotate the potentiometer knob on the module, the data will be changed within 0-1023. Shown below.



Project 10: One Relay

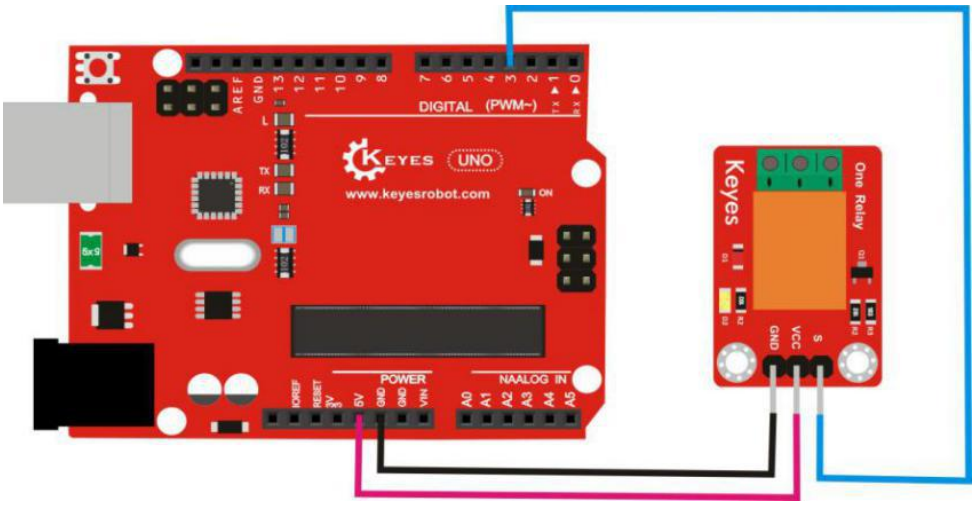
Description

The single 5V relay module is active high level. In the experiment, connect the signal pin of relay module to digital 3 of UNO, aimed to control the relay on the module on and off circularly.

Part List

Development Board*1
USB Cable*1
Relay module*1
Jumper Wire*3

Wiring Diagram



Sample Code

```
//////////////////////////////////////////////////////////
int Relay = 3; //define the digital 3
void setup()
{
pinMode(Relay, OUTPUT); //set Relay as output
}
void loop()
{
digitalWrite(Relay, HIGH); //relay on
delay(2000); //delay 2 seconds
digitalWrite(Relay, LOW); //relay off
delay(2000); //delay 2 seconds
}
//////////////////////////////////////////////////////////
```

Phenomenon

Wire it up as the above diagram, upload well the code, relay is turned on (ON end is connected, NC end is disconnected) for two seconds, stop (ON end disconnected, NC end connected) for two seconds, circularly and alternately. And D2 led on the relay lights up when it is on.

Project 11: RGB Module

Description

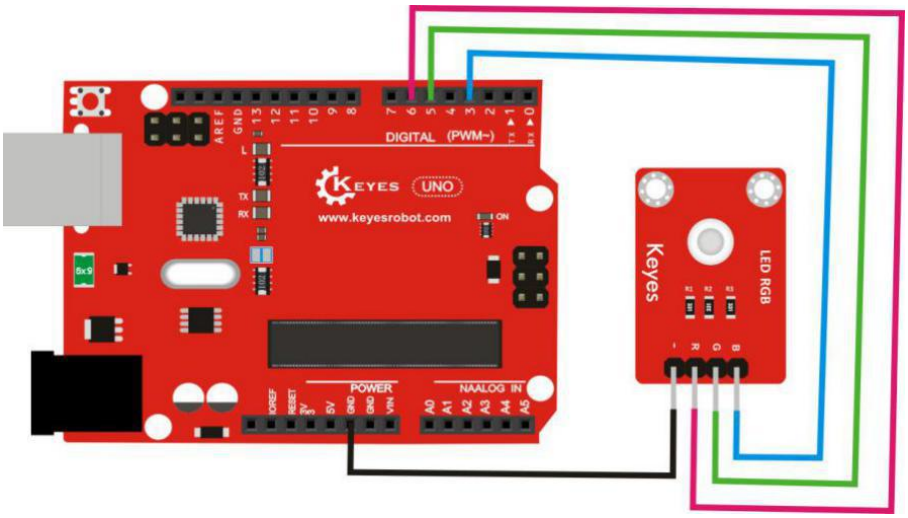
In the following we are going to detect this plugin RGB module. This module RGB module is mainly made of a full-color LED. Using PWM voltage input of R, G, B pin can adjust the intensity of three primary colors (red/ blue/green) so as to realize the full color mixing effect.

Next, we will make this RGB module display different colors circularly.

Part List

Development Board*1
USB Cable*1
RGB module*1
Jumper Wire*4

Wiring Diagram



Sample Code

```
//////////////////////////////////////////////////////////
int redPin = 6;   //Red LED control pin is connected to Arduino pin 6
int greenPin = 5;   // Green LED control pin is connected to Arduino pin 5
int bluePin = 3;   // blue LED control pin is connected to Arduino pin 3
void setup()
{
          pinMode(redPin, OUTPUT); //set redPin 6 as output
          pinMode(greenPin, OUTPUT); //set greenPin 5 as output
          pinMode(bluePin, OUTPUT); //set bluePin 3 as output
}

void loop()   // run over and over again
{
          // Basic colors:
         color(255, 0, 0); //   red light on
         delay(1000); //   delay one second
         color(0,255, 0); // green light on
         delay(1000); //delay one second
         color(0, 0, 255); //   blue light on
         delay(1000); //delay one second

          // Example blended colors:
          color(255,255,0); //   yellow light on
          delay(1000); //delay one second
          color(128,0,255); //   purple light on
          delay(1000); //delay one second
          color(255,255,255); //   white light on
          delay(1000); //delay one second
          color(0,0,0); // led off
          delay(1000); //delay one second
}
void color (unsigned char red, unsigned char green, unsigned char blue)   //color
control function
{
          analogWrite(redPin, red);
          analogWrite(greenPin,green);
          analogWrite(bluePin, blue);
}
//////////////////////////////////////////////////////////
```

Phenomenon

Powered up and upload well the code, RGB module will continue to display the red light for one second, green light for one second, blue light for one second, yellow light

for one second, purple light for one second, and white light for one second, then stop display for one second, alternately and circularly.

Project 12: Thermistor

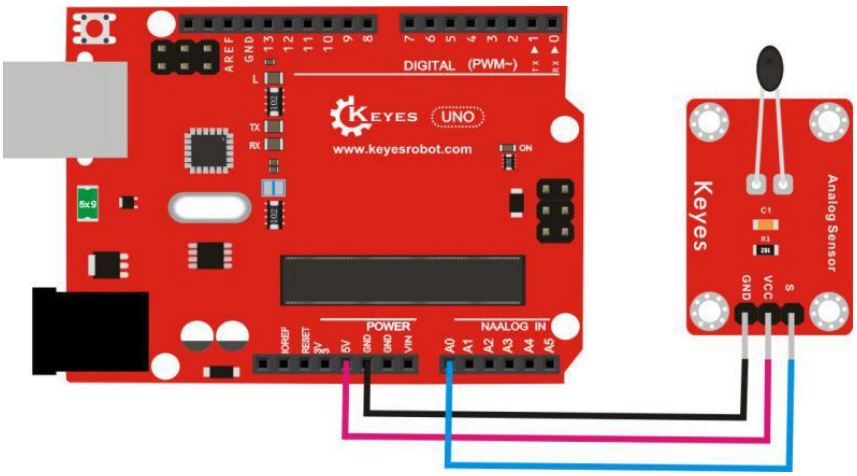
Description

In the following we are going to detect this thermistor sensor. It is based on the working of thermistor, which can sense the temperature changes of current ambient in real time. In the experiment, connect the signal end of sensor to analog A0 of UNO board, after simple programming, convert the data the sensor outputs into Celsius, showing you the temperature value on the serial monitor of Arduino IDE. You can apply it to gardening, family alarm system and other devices.

Part List

Development Board*1
USB Cable*1
Thermistor sensor*1
Jumper Wire*3

Wiring Diagram



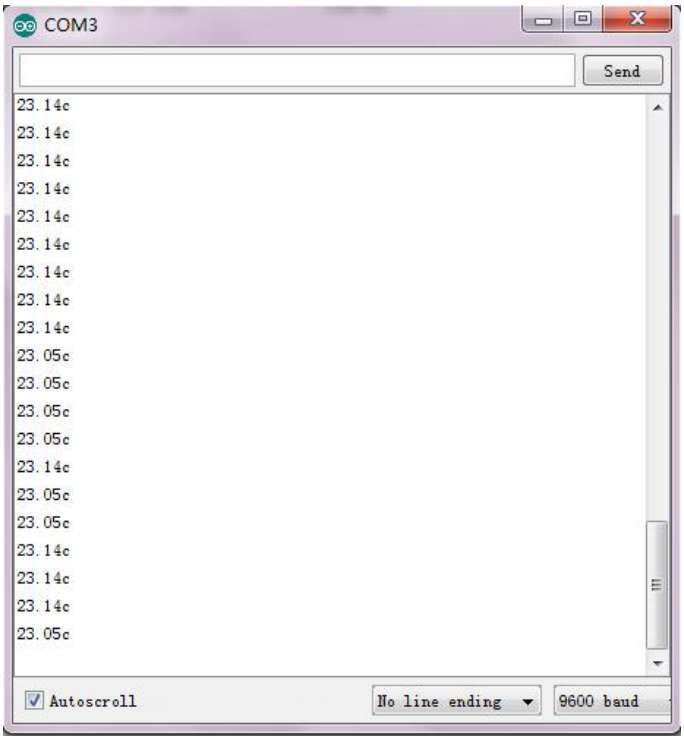
Sample Code

```
//////////////////////////////////////////////////////////
#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); //set the baud rate
}
 void loop()
{
Serial.print(Thermister(analogRead(0))); // display the temperature value
Serial.println("c");   // display c，and line wrap
delay(500);   // delay 0.5S
}
////////////////////////////////////////////////////////////
```

Phenomenon

Powered up and upload well the code, then open the serial monitor, you will see the temperature value of current ambient. Shown below.



Project 13: Button Control

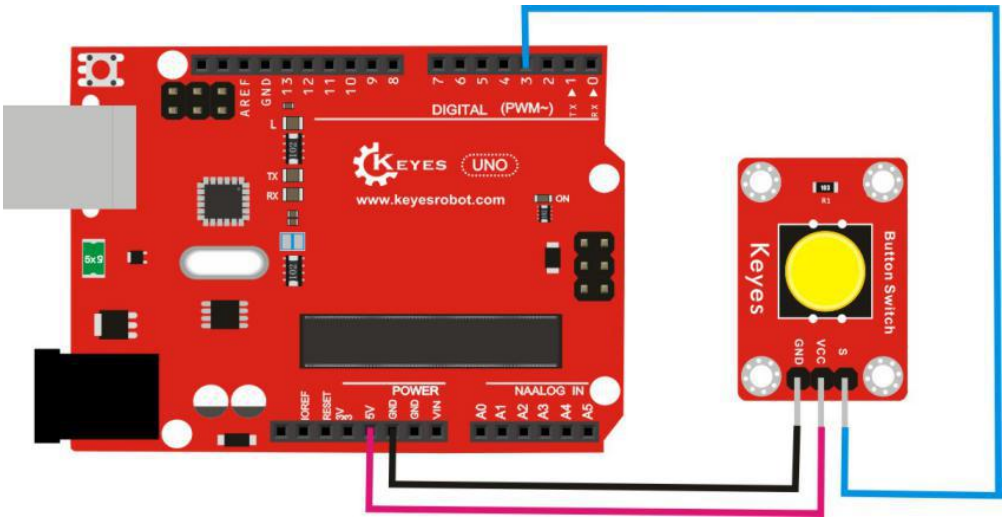
Description

In the following we are going to detect this button sensor. When press down the button on the module, its signal end will output low level, while release the button, the signal end will keep high level. In this experiment, we are going to control the D13 indicator on UNO board on and off combined with this button module.

Part List

Development Board*1
USB Cable*1
Button sensor*1
Jumper Wire*3

Wiring Diagram



Sample Code

```
//////////////////////////////////////////////////////////
int ledPin = 13;   //set the digital 13
int inputPin = 3; //set the digital 3
void setup()
{
pinMode(ledPin, OUTPUT);     //set ledPin as output
pinMode(inputPin, INPUT); //set inputPin as input
}
void loop()
{
int val = digitalRead(inputPin);
//set the digital variable val, read the value of digital 3 and assign it to val
```

```
if (val == LOW) //if val is low level
{
digitalWrite(ledPin, HIGH); // LED on
}
 else
{
digitalWrite(ledPin, LOW); // LED dims
}
}
//////////////////////////////////////////////////////////
```

Phenomenon

Powered up and upload well the code, press down the button, D13 indicator on UNO board is on. While release the button, D13 indicator on UNO board is off.

Project 14: Magical Light Cap

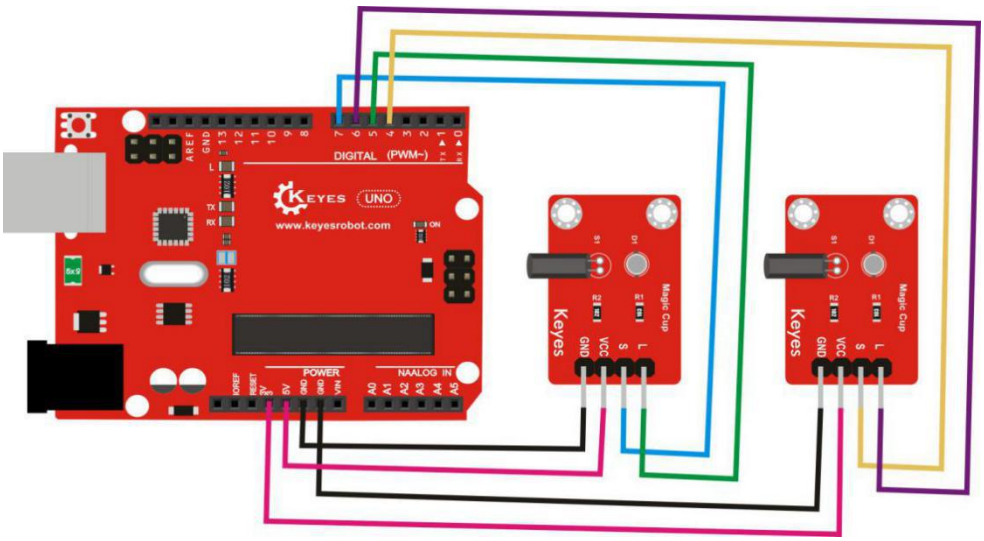
Description

In the following we will detect the magical cap sensor. It can be interactive with Arduino, making use of the theory of PWM controlling brightness, to make the brightness of two sensors change. In the experiment, the ball switch will provide digital signal to trigger the PWM adjustment, via programming, you will see the test phenomenon that two caps seem like pouring the light.

Part List

Development Board*1
USB Cable*1
Magical cap sensor*2
Jumper Wire*8

Wiring Diagram



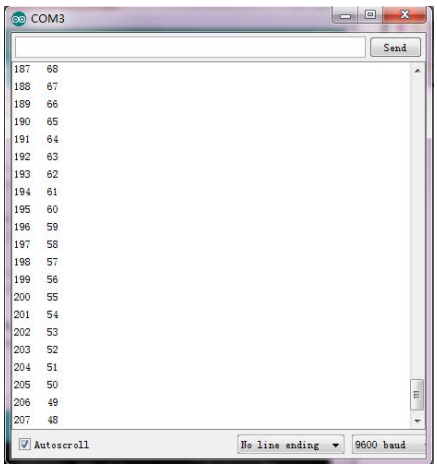
Sample Code

```
//////////////////////////////////////////////////////////
int LedPinA = 5; //define the digital 5
int LedPinB = 6; //define the digital 6
int ButtonPinA = 7;//define the digital 7
int ButtonPinB = 4;//define the digital 4
int buttonStateA = 0;
int buttonStateB = 0;
int brightnessA = 0;
int brightnessB= 255;
void setup()
{
Serial.begin(9600);//set the baud rate
pinMode(LedPinA, OUTPUT);//set digital 5 as output
pinMode(LedPinB, OUTPUT);//set digital 6 as output
pinMode(ButtonPinA, INPUT);//set digital 7 as input
pinMode(ButtonPinB, INPUT);//set digital 4 as input
}
void loop()
{
buttonStateA = digitalRead(ButtonPinA);//read the value of digital 7 and assign it to
buttonStateA
if (buttonStateA == HIGH && brightnessA != 255)
//if buttonStateA is high level and brightnessA is not 255
{
brightnessA ++;//brightnessA plus 1
delay(10);//delay 0.01S
}
if (buttonStateA == LOW && brightnessA != 0)
//if buttonStateA is low level and brightnessA is not 0
```

```
{
brightnessA --;//brightnessA subtract 1
delay(10);//delay 0.01S
}
analogWrite(LedPinB, brightnessA);//assign brightnessA to PWM port 6
Serial.print(brightnessA);//display brightnessA value
Serial.print("     ");
buttonStateB = digitalRead(ButtonPinB);//read the value of digital 4 and assign it to
buttonStateB
if (buttonStateB == HIGH && brightnessB != 0)
//if buttonStateB is high level brightnessA is not 0
{
brightnessB --;//brightnessB subtract 1
delay(10);//delay 0.01S
}
if (buttonStateB == LOW && brightnessB != 255)
//if buttonStateB is low level and brightnessA is not 255
{
brightnessB++;//brightnessB plus 1
delay(10);//delay 0.01S
}
analogWrite(LedPinA, brightnessB); //assign brightnessB to PWM port 5
Serial.println(brightnessB);//display brightnessB value and line wrap
delay(5);
}
////////////////////////////////////////////////////////////
```

Phenomenon

Powered up and upload well the code, tilt two cap sensors toward one side at the same time, you will see one LED on the cap gradually dims, while another one gradually brightens, finally one LED totally goes out but another LED become the brightest. In addition, you can see the detailed data changes on the serial monitor. Shown below.



Project 15: DHT11 Temp and Humidity

Description

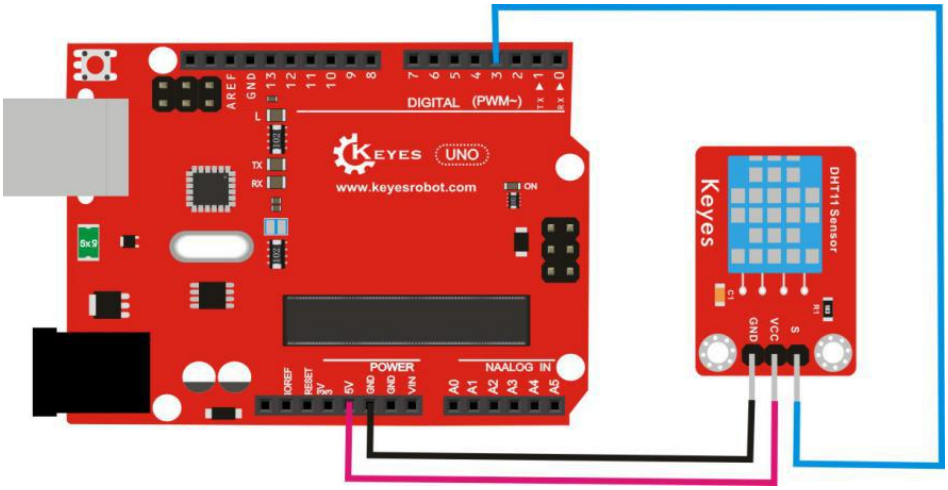
The DHT11 sensor is a temperature and humidity composite sensor with calibrated digital signal output. It uses a dedicated digital module acquisition technology and temperature and humidity sensing technology, to ensure the high reliability and excellent long-term stability.

Next, we are going to detect this DHT11 sensor. Connect the signal pin of sensor to digital 3 of UNO board, finally show you the temperature and humidity value on serial monitor of Arduino IDE.

Part List

Development Board*1
USB Cable*1
DHT11 temp. and humidity sensor*1
Jumper Wire*3

Wiring Diagram



Sample Code

```
//////////////////////////////////////////////////////////////
#include <dht11.h>
dht11 DHT;
#define DHT11_PIN 3
void setup(){
  Serial.begin(9600);
  Serial.println("DHT TEST PROGRAM ");
  Serial.print("LIBRARY VERSION: ");
  Serial.println(DHT11LIB_VERSION);
  Serial.println();
```

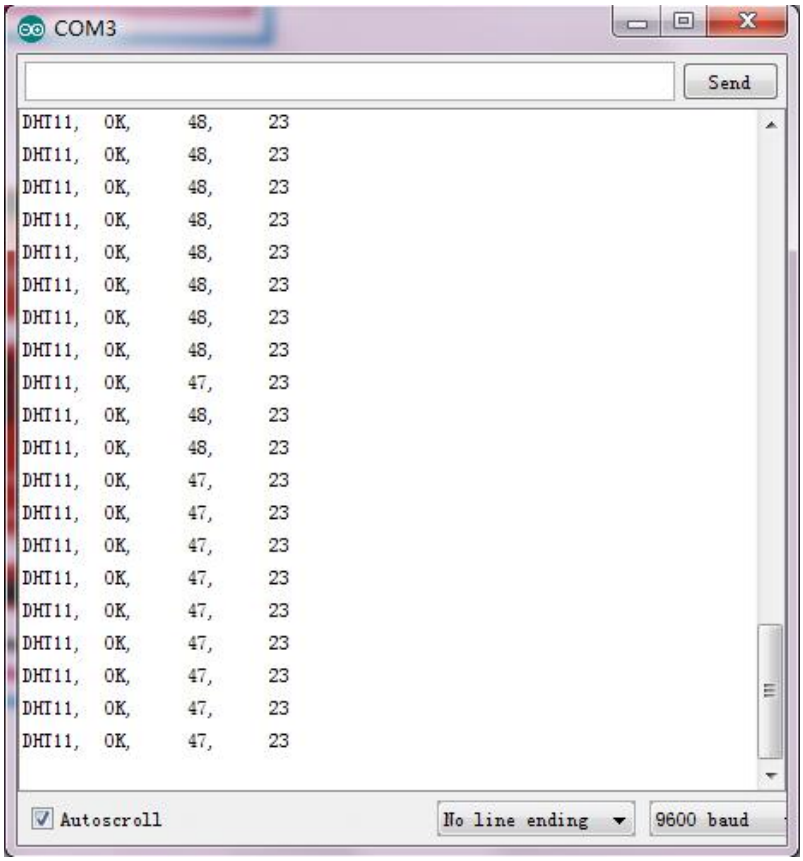
```
  Serial.println("Type,\tstatus,\tHumidity (%),\tTemperature (C)");
}
void loop(){
  int chk;
  Serial.print("DHT11, \t");
  chk = DHT.read(DHT11_PIN);      // READ DATA
  switch (chk){
    case DHTLIB_OK:
                Serial.print("OK,\t");
                break;
case DHTLIB_ERROR_CHECKSUM:
                Serial.print("Checksum error,\t");
                break;
    case DHTLIB_ERROR_TIMEOUT:
                Serial.print("Time out error,\t");
                break;
    default:
                Serial.print("Unknown error,\t");
                break;
  }
 // DISPLAT DATA
  Serial.print(DHT.humidity,1);
  Serial.print(",\t");
  Serial.println(DHT.temperature,1);
  delay(1000);
}
//////////////////////////////////////////////////////////////
```

Libraries Download Link:

https://pan.baidu.com/s/1eSlMYD8

Phenomenon

Powered up and upload well the code, then open the serial monitor on IDE, you will see the temperature and humidity value of current ambient. Shown as the below figure.



Project 16: Photocell

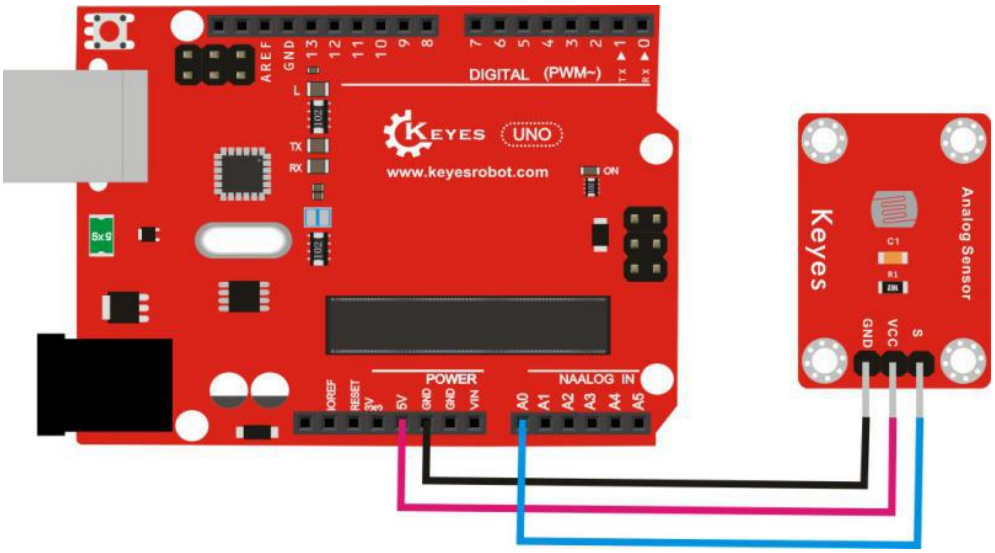
Description

In the following we will detect the photocell sensor. This sensor is extremely sensitive to ambient light, so it is used to detect the light brightness in ambient, or to trigger microcontroller and relay modules and more. In this experiment, connect the signal end of sensor to analog A0 of UNO board, finally show you the analog value output on the serial monitor of Arduino IDE.

Part List

Development Board*1
USB Cable*1
Photocell sensor*1
Jumper Wire*3

Wiring Diagram

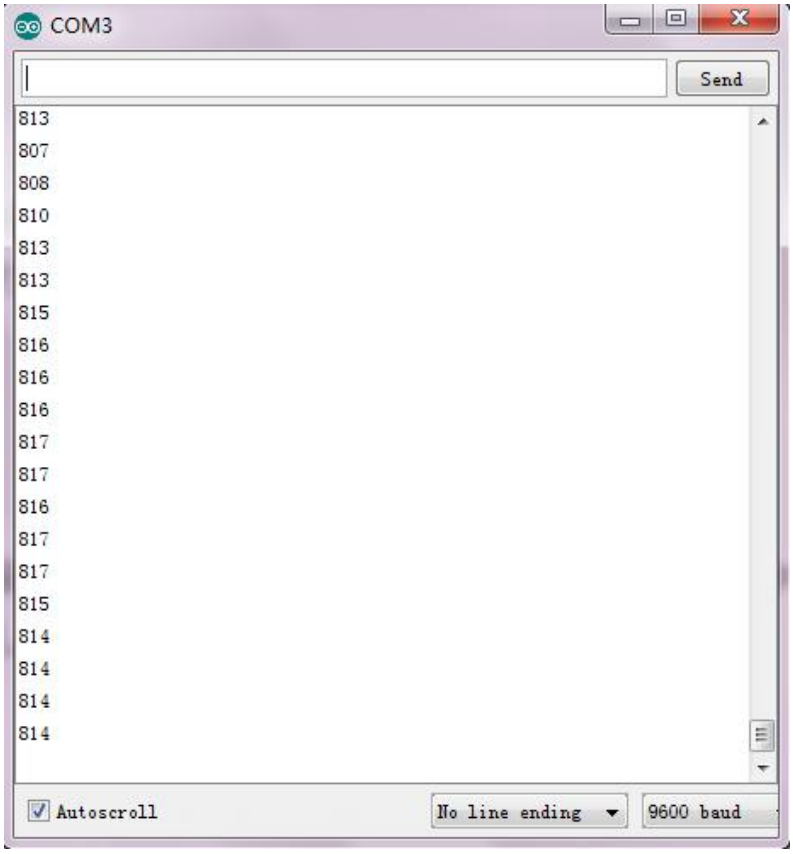


Sample Code

```
//////////////////////////////////////////////////////////
int sensorPin =A0 ;   //set the analog A0
int value = 0;       //set value as 0
void setup()
{
 Serial.begin(9600); //set the baud rate
}
void loop()
{
value = analogRead(sensorPin);   //set the value as the value of A0 it reads
Serial.println(value, DEC);   //display value and line wrap
delay(200);   //delay 0.2 second
}
//////////////////////////////////////////////////////////
```

Phenomenon

Powered up and upload well the code, open the serial monitor of Arduino IDE, you will see the analog value representing the light intensity. The stronger the light, the greater the value. Shown below.



Project 17: Tilt Sensor

Description

In the following we will test the tilt sensor. This sensor uses the working principle that the ball inside the switch with different tilting angle will be able to trigger the circuit. It can be used for tilt detection, alarm device or other detection.

In this experiment, make use of D13 indicator on the UNO board combined with the tilt sensor, finally control D13 indicator on and off.

Part List

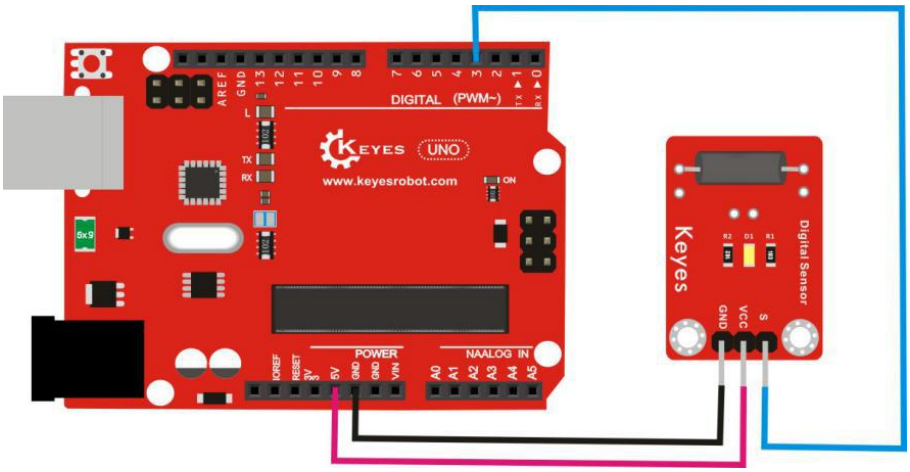
Development Board*1

USB Cable*1

Tilt sensor*1

Jumper Wire*3

Wiring Diagram



Sample Code

```
//////////////////////////////////////////////////////////
int ledPin = 13; //define digital 13
int switcher = 3; // define the digital 3
void setup()
{
pinMode(ledPin, OUTPUT); // set ledPin as output
pinMode(switcher, INPUT); //set switcher as input
}
void loop()
{
if(digitalRead(switcher)==HIGH) //read the digital 3 and found as high level
{
digitalWrite(ledPin, HIGH); // LED lights up
}
else
{
digitalWrite(ledPin, LOW); // LED dims
}
}
//////////////////////////////////////////////////////////
```

Phenomenon

Powered up and upload well the code, tilt the sensor toward one side, D13 indicator on the UNO board is on, while D1 led on the tilt sensor is off. If tilt the sensor toward another side, D13 indicator on the UNO board is off, while D1 led on the tilt sensor is on.

Project 18: Photo Interrupter

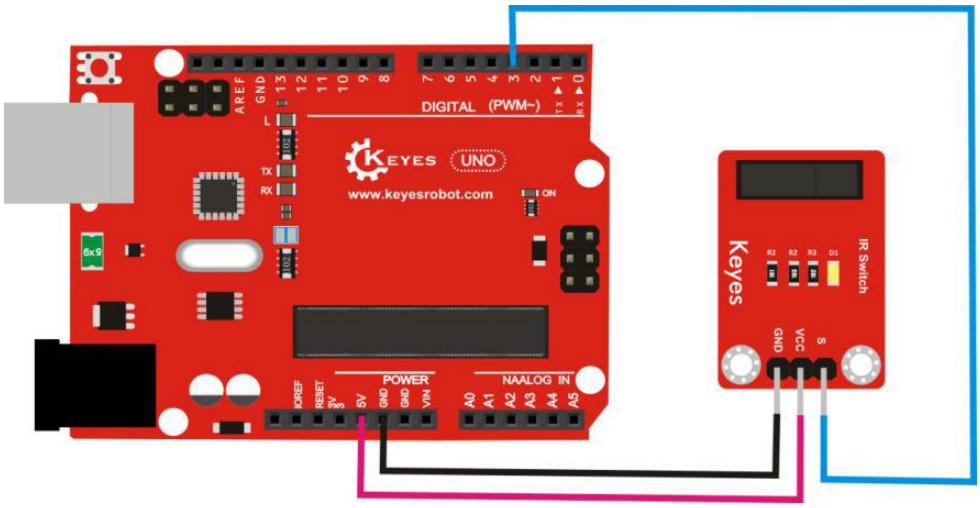
Description

In the following we will test the photo interrupter sensor. This sensor is mainly formed with a photo-interrupter, and it belongs to an optical switch sensor of a photo-interrupting photoelectric switch. In the experiment, we use this module to control the on and off of built-in D13 indicator on the UNO R3 board.

Part List

Development Board*1
USB Cable*1
Photo interrupter sensor*1
Jumper Wire*3

Wiring Diagram



Sample Code

```
//////////////////////////////////////////////////////////
int Led = 13 ;//define the digital 13
int buttonpin = 3; //define the digital 3
int val ;//define the digital variable val
void setup ()
{
pinMode (Led, OUTPUT) ; //set Led as output
pinMode (buttonpin, INPUT) ;//set buttonpin as input
}
void loop ()
{
val = digitalRead (buttonpin) ;//read the value of digital 3 and assign it to val
if (val == HIGH)//if val is high level, LED on
```

```
{
digitalWrite (Led, HIGH);// LED lights up
}
else
{
digitalWrite (Led, LOW);// LED dims
}
}
////////////////////////////////////////////////////////
```

Phenomenon

Done wiring and powered up, upload well the code, D13 led on the UNO R3 board is off. If pick up a paper to cover the groove on the module, D13 led is on.

Project 19: ADXL345 Acceleration

Description

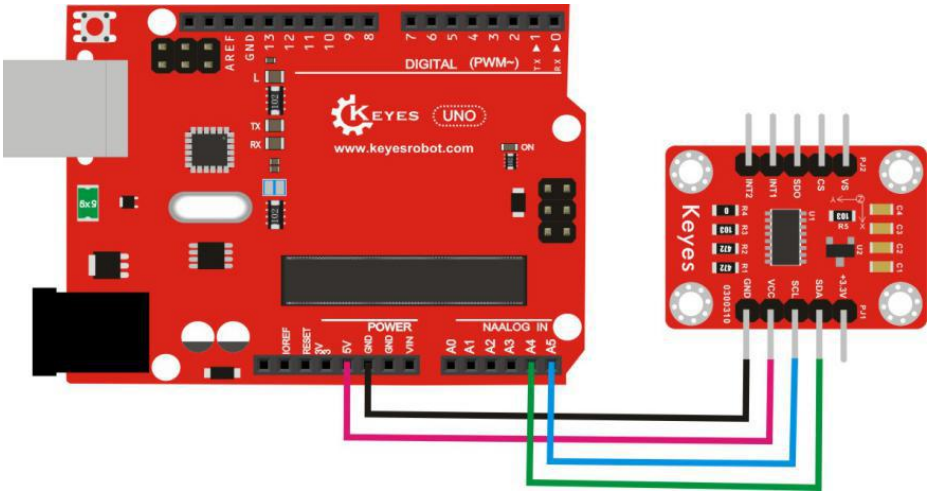
In the following we will test the ADXL345 acceleration sensor. The ADXL345 sensor is a compact, low-power tri-axis accelerometer with a resolution of 13 bits and a measurement range of±16 g. The digital output data is in 16-bit binary complement format. It supports I2C and SPI communications.

In this experiment, we are going to use the I2C communication, showing the data the sensor detects on the monitor of Arduino IDE.

Part List

Development Board*1
USB Cable*1
ADXL345 acceleration sensor*1
Jumper Wire*4

Wiring Diagram



Sample Code

```
//////////////////////////////////////////////////////////
#include <Wire.h>
// Registers for ADXL345
#define ADXL345_ADDRESS (0xA6 >> 1)   // address for device is 8 bit but shift to the
                                      // right by 1 bit to make it 7 bit because the
                                      // wire library only takes in 7 bit addresses
#define ADXL345_REGISTER_XLSB (0x32)
int accelerometer_data[3];
// void because this only tells the cip to send data to its output register
// writes data to the slave's buffer
void i2c_write(int address, byte reg, byte data) {

  // Send output register address
  Wire.beginTransmission(address);
  // Connect to device
  Wire.write(reg);
  // Send data
  Wire.write(data); //low byte
  Wire.endTransmission();
}

// void because using pointers
// microcontroller reads data from the sensor's input register
void i2c_read(int address, byte reg, int count, byte* data) {
  // Used to read the number of data received
  int i = 0;
  // Send input register address
  Wire.beginTransmission(address);
```

```
  // Connect to device
  Wire.write(reg);
  Wire.endTransmission();

  // Connect to device
  Wire.beginTransmission(address);
  // Request data from slave
  // Count stands for number of bytes to request
  Wire.requestFrom(address, count);
  while(Wire.available()) // slave may send less than requested
  {
    char c = Wire.read(); // receive a byte as character
    data[i] = c;
    i++;
  }
  Wire.endTransmission();
}

void init_adxl345() {
  byte data = 0;

  i2c_write(ADXL345_ADDRESS, 0x31, 0x0B);     // 13-bit mode   +_ 16g
  i2c_write(ADXL345_ADDRESS, 0x2D, 0x08);     // Power register


  i2c_write(ADXL345_ADDRESS, 0x1E, 0x00);     // x
  i2c_write(ADXL345_ADDRESS, 0x1F, 0x00);     // Y
  i2c_write(ADXL345_ADDRESS, 0x20, 0x05);     // Z

  // Check to see if it worked!
  i2c_read(ADXL345_ADDRESS, 0X00, 1, &data);
  if(data==0xE5)
    Serial.println("it work Success");
  else
    Serial.println("it work Fail");
}

void read_adxl345() {
  byte bytes[6];
  memset(bytes,0,6);

  // Read 6 bytes from the ADXL345
  i2c_read(ADXL345_ADDRESS, ADXL345_REGISTER_XLSB, 6, bytes);
  // Unpack data
```

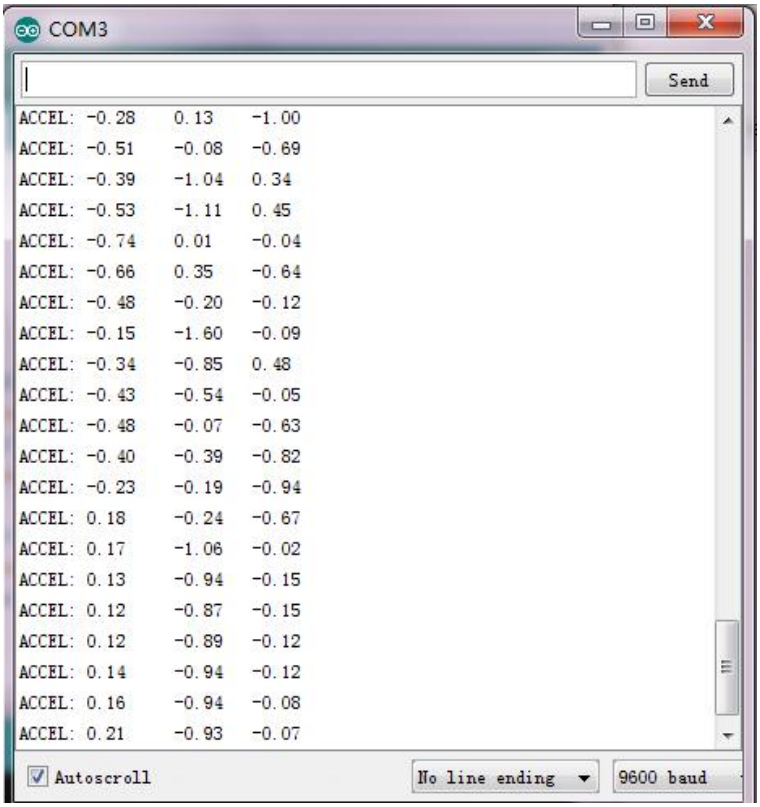
```
  for (int i=0;i<3;++i) {
    accelerometer_data[i] = (int)bytes[2*i] + (((int)bytes[2*i + 1]) << 8);
  }
}
// initialise and start everything
void setup() {
  Wire.begin();
  Serial.begin(9600);
  for(int i=0; i<3; ++i) {
    accelerometer_data[i]   = 0;
  }
  init_adxl345();
}

void loop() {
  read_adxl345();
  Serial.print("ACCEL: ");
  Serial.print(float(accelerometer_data[0])*3.9/1000);//3.9mg/LSB  scale  factor  in
13-bit mode
  Serial.print("\t");
  Serial.print(float(accelerometer_data[1])*3.9/1000);

  Serial.print("\t");
  Serial.print(float(accelerometer_data[2])*3.9/1000);
  Serial.print("\n");
  delay(100);
}
////////////////////////////////////////////////////////////
```

Libraries Download Link:

https://pan.baidu.com/s/1kVkXsm7

Phenomenon

Done wiring and powered up, upload well the code, then open the serial monitor, and rotate the sensor, you will see the data change displayed on the monitor. Shown as the figure below.



Project 20: Microphone Sound

Description

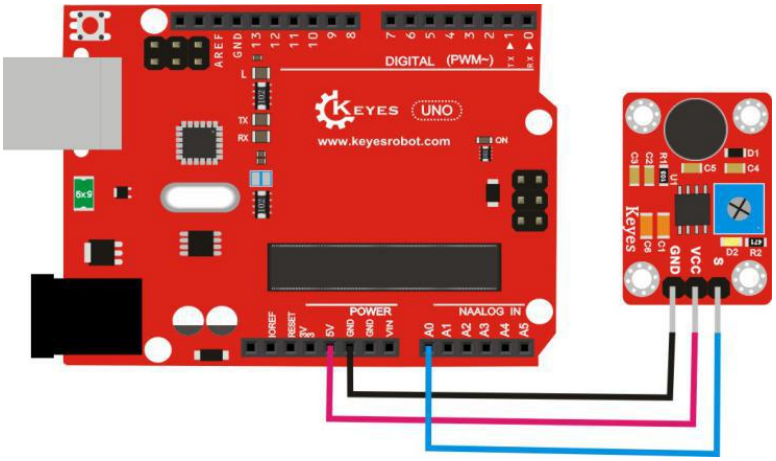
This project aims to test the microphone sound sensor. The S terminal of sensor is an analog output, which is for voltage signal output in real time. The potentiometer on the sensor can be used to adjust the signal gains.

In the experiment, connect the signal pin of sensor to analog A0 of UNO board, finally show you the analog value output on the serial monitor of Arduino IDE.

Part List

Development Board*1
USB Cable*1
Sound sensor*1
Jumper Wire*3

Wiring Diagram

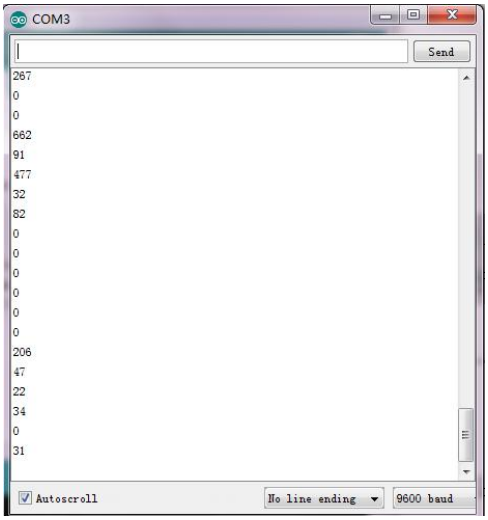


Sample Code

```
//////////////////////////////////////////////////////////
int sensorPin =A0 ;   //define analog port A0
int value = 0;       //set value as 0
void setup()
{
 Serial.begin(9600); //set the baud rate
}
void loop()
{
value = analogRead(sensorPin);   //set value as the value of A0 it read
Serial.println(value, DEC);   //display value and line wrap
delay(100);   //delay 0.1 second
}
//////////////////////////////////////////////////////////
```

Phenomenon

Done wiring and powered up, upload well the code, the open the serial monitor of Arduino IDE, you will see the corresponding analog value. Shown as the figure below. The higher the sound, the greater the analog value.



Project 21: Hall Sensor

Description

We are going to detect this hall sensor. Input magnetic induction intensity, while output a digital voltage signal. It has the characteristics of small size, high sensitivity, fast response, good temperature performance and high reliability.

It can be used for non-contact switch, position speed detection and control, full alarm device, textile control system and so on.

Part List

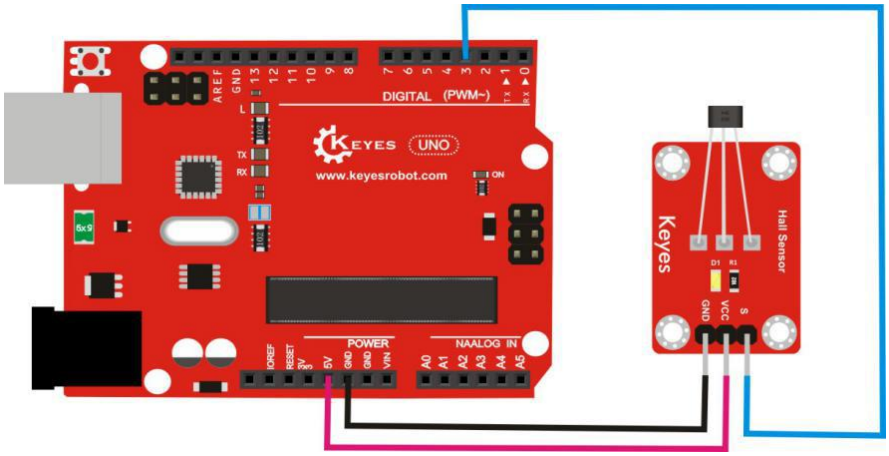
Development Board*1

USB Cable*1

Hall sensor*1

Jumper Wire*3

Wiring Diagram



Sample Code

```
//////////////////////////////////////////////////////////
int ledPin = 13; //define digital 13
int inputPin = 3; //define digital 3
int val = 0; //define digital variable val, and set as 0
void setup()
{
pinMode(ledPin, OUTPUT); //set ledPin as output
pinMode(inputPin, INPUT); //set inputPin as input
}
void loop(){
val = digitalRead(inputPin); //read the value of digital 3 and assign it to val
if (val == LOW) //if val is low level
{
```

```
digitalWrite(ledPin, HIGH); //LED brightens
}
else
{
digitalWrite(ledPin, LOW); //LED dims
}
}
//////////////////////////////////////////////////////////
```

Phenomenon

Done wiring and powered up, upload well the code, D13 indicator on UNO board is off, and D1 led on the module is also off. But when put a magnet close to the hall module, you will see the D13 indicator on UNO board is on, and D1 led on the module is also on.

Project 22: Crash Sensor

Description

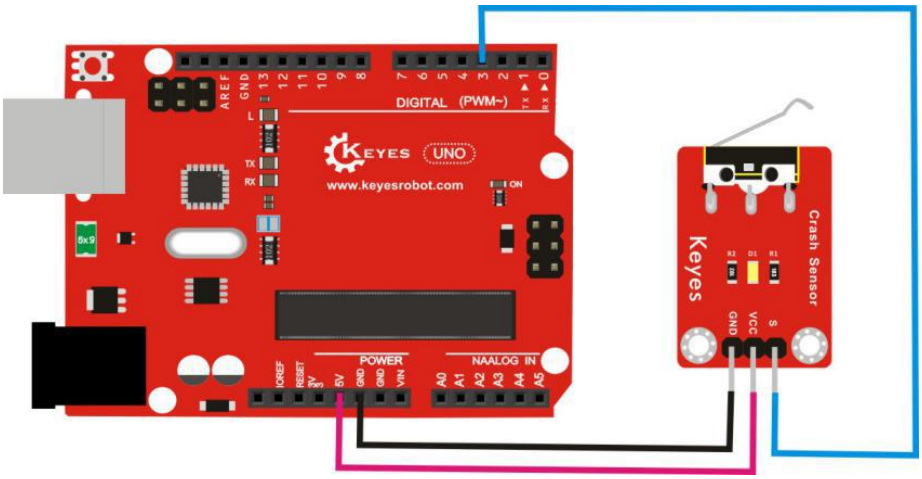
We are going to detect this crash sensor. When the sensor crashes the objects, and the metal switch of the sensor is pressed down, the signal end of sensor will output low level signal. When the switch is released, the signal end of sensor will keep high level.

This sensor can be used as limit switch inside 3D printer. In the experiment, we use the D13 indicator on the UNIO board combined with this sensor, to control the state of D13 indicator on or off.

Part List

Development Board*1
USB Cable*1
Crash sensor*1
Jumper Wire*3

Wiring Diagram



Sample Code

```
//////////////////////////////////////////////////////////
int Led=13;//set LED interface
int Shock=3;//define the crash sensor interface
int val;//define the digital variable val
void setup()
{
pinMode(Led,OUTPUT);//set LED as output
pinMode(Shock,INPUT);//set the crash sensor as input
}
void loop()
{
val=digitalRead(Shock);//read the value of digital 3 and assign it to val
if(val==LOW)//if the sensor detects the signal, LED on
{
digitalWrite(Led,HIGH); //LED on
}
 else
{
digitalWrite(Led,LOW); //LED dims
}
}
//////////////////////////////////////////////////////////
```

Phenomenon

Done wiring and powered up, upload well the code, if the metal switch of sensor is pressed, both D13 indicator on the UNO board and the D1 led on the module are on. Otherwise, those two lights are off.

Project 23: IR Transmitter and Receiver

Description

In this project, we will use both IR transmitter and IR receiver sensor.

The IR transmitter sensor is mainly composed of an infrared transmitting tube. It is a light emitting device that can directly convert electric energy into near-infrared light and finally radiate it out.

The infrared receiver sensor is mainly composed of an infrared receiving head. It is a device that integrates reception, amplification, and demodulation. The internal IC has completed demodulation, and the output is a digital signal. It is mainly used with infrared remote control or infrared transmitter.

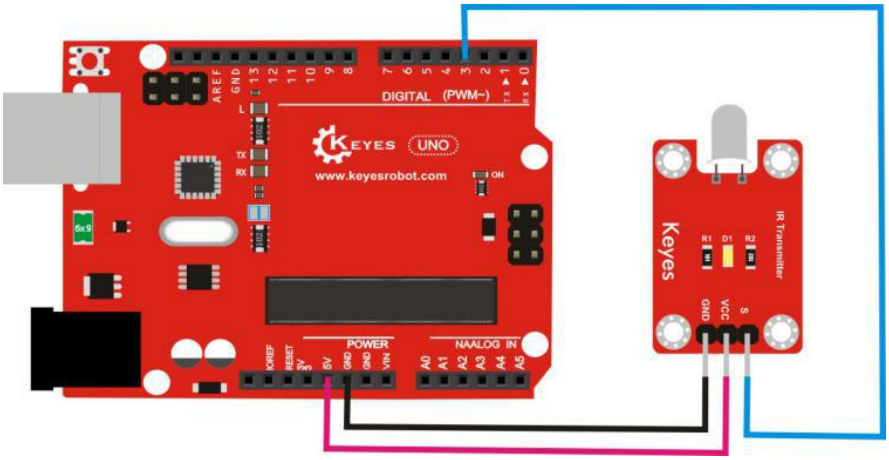
In the experiment, we will use two UNO boards to connect the IR transmitter and IR receiver respectively. The IR transmitter module will transmit the signal, and receiver module will receive the signal and then decode it, showing the decoding result on the serial monitor of Arduino IDE.

Part List

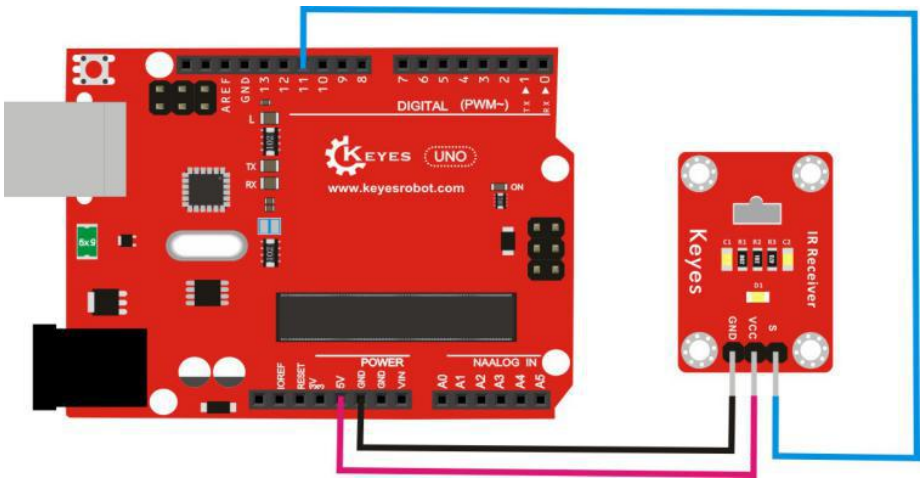
Development Board*1
USB Cable*1
IR transmitter sensor*1
IR receiver sensor*1
Jumper Wire*6

Wiring Diagram

For IR transmitter sensor:



For IR receiver sensor:



Sample Code

For Transmitter:

```
//////////////////////////////////////////////////////////
#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 switch encoding
delay(40);
}
}
//////////////////////////////////////////////////////////
```

For Receiver:

```
//////////////////////////////////////////////////////////
#include <IRremote.h>
int RECV_PIN = 11; //set the digital 11
IRrecv irrecv(RECV_PIN);
decode_results results;
void setup()
 {
 Serial.begin(9600);//set the baud rate
 irrecv.enableIRIn(); //enable IR receiver
 }
```

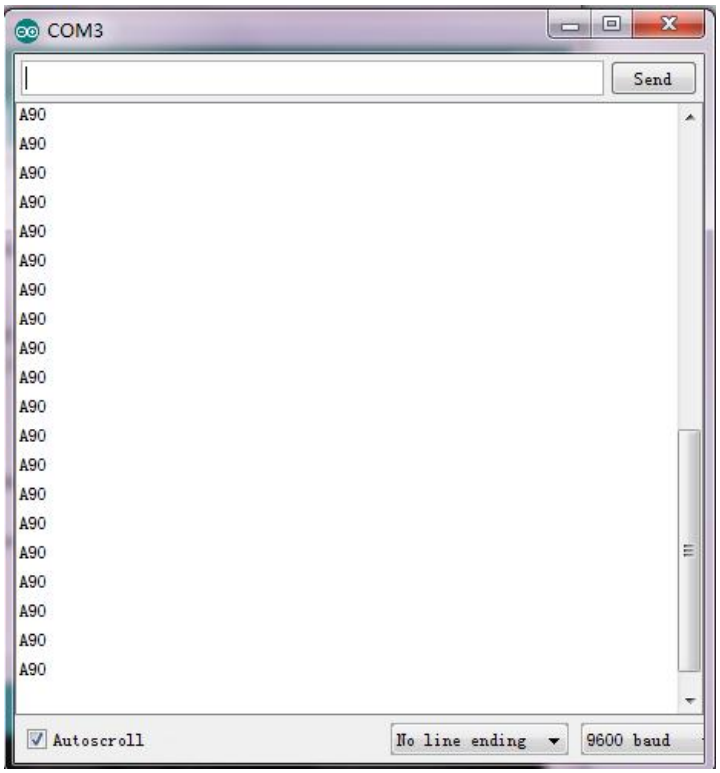
```
void loop()
{
if (irrecv.decode(&results))
{
Serial.println(results.value, HEX); //display the data
irrecv.resume();
}
}
//////////////////////////////////////////////////////////
```

Libraries Download Link:

https://pan.baidu.com/s/1pLDpyg7

Phenomenon

Wire it up as the above diagram, upload well the code, then infrared transmitter sensor will send the signal to IR receiver sensor, D1 led on both transmitter and receiver sensor will quickly flash. After that, connect the UNO board that connected to IR receiver to PC, you can see the monitor will display "A90". Shown below.



Project 24: Ultrasonic Ranging

Description

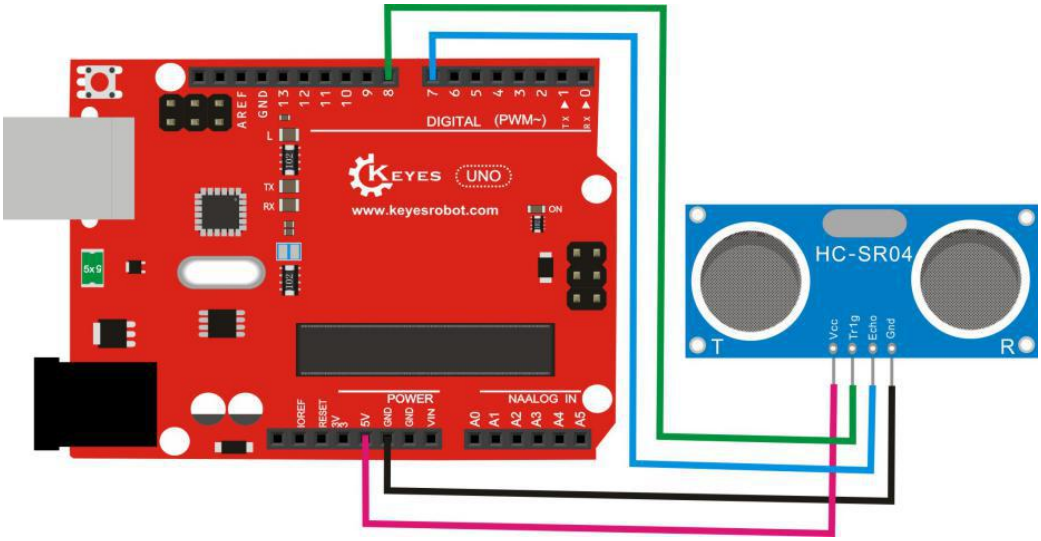
In this experiment, we are going to detect ultrasonic sensor. The sensor is mainly used for distance measurement. It has features of high precision, super closeness in the blind area (2cm), and stable performance.

In the experiment, the signal of the sensor is connected to the digital port 7 and 8 of UNO board, finally you will see that the distance between the ultrasonic sensor and the front obstacle is displayed on the serial monitor of Arduino IDE.

Part List

Development Board*1
USB Cable*1
Ultrasonic sensor*1
Jumper Wire*4

Wiring Diagram



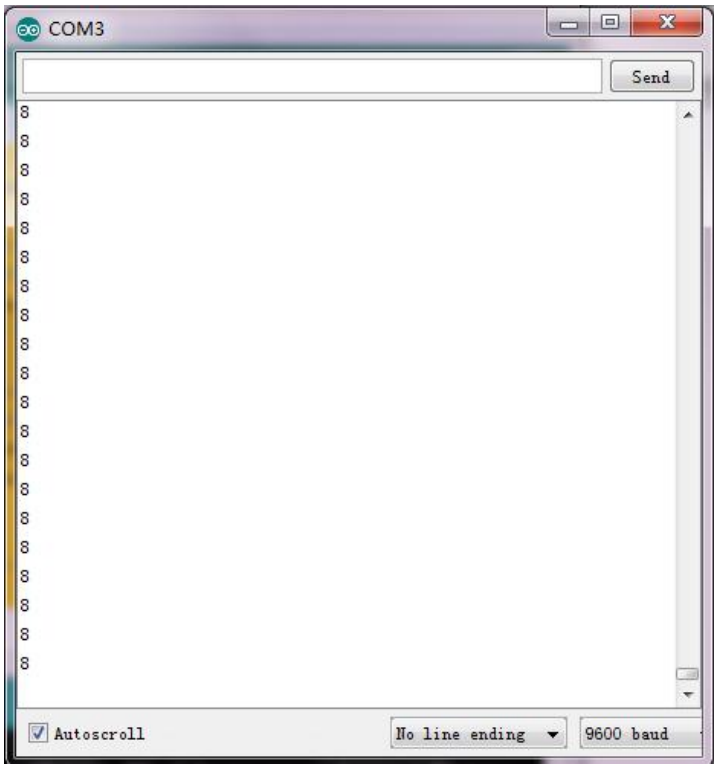
Sample Code

```
////////////////////////////////////////////////////////////////
#define echoPin 7 // Echo Pin
#define trigPin 8 // Trigger Pin
#define LEDPin 13 // Onboard LED

int maximumRange = 200; // Maximum range needed
int minimumRange = 0; // Minimum range needed
long duration, distance; // Duration used to calculate distance

void setup() {
```

```
  Serial.begin (9600);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(LEDPin, OUTPUT); // Use LED indicator (if required)
}

void loop() {
/* The following trigPin/echoPin cycle is used to determine the
  distance of the nearest object by bouncing soundwaves off of it. */
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);

  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  duration = pulseIn(echoPin, HIGH);

  //Calculate the distance (in cm) based on the speed of sound.
  distance = duration/58.2;

  if (distance >= maximumRange || distance <= minimumRange){
  /* Send a negative number to computer and Turn LED ON
  to indicate "out of range" */
  Serial.println("-1");
  digitalWrite(LEDPin, HIGH);
  }
  else {
  /* Send the distance to the computer using Serial protocol, and
  turn LED OFF to indicate successful reading. */
  Serial.println(distance);
  digitalWrite(LEDPin, LOW);
  }

  //Delay 50ms before next reading.
  delay(50);
}
////////////////////////////////////////////////////////////
```

Phenomenon

Done wiring and uploading the code, open the monitor and set the baud rate as 9600, you will see the distance between ultrasonic sensor and obstacle. The unit is cm. Shown as the figure below.



Project 25: Gas Detection

Description

In this experiment, we are going to detect MQ-2 gas sensor. The gas sensitive material used in this sensor is tin dioxide (SnO2) with low conductivity in clean air. When the combustible gas is present in the environment where the sensor is located, the conductivity of the sensor increases as the concentration of combustible gas in the air increases.

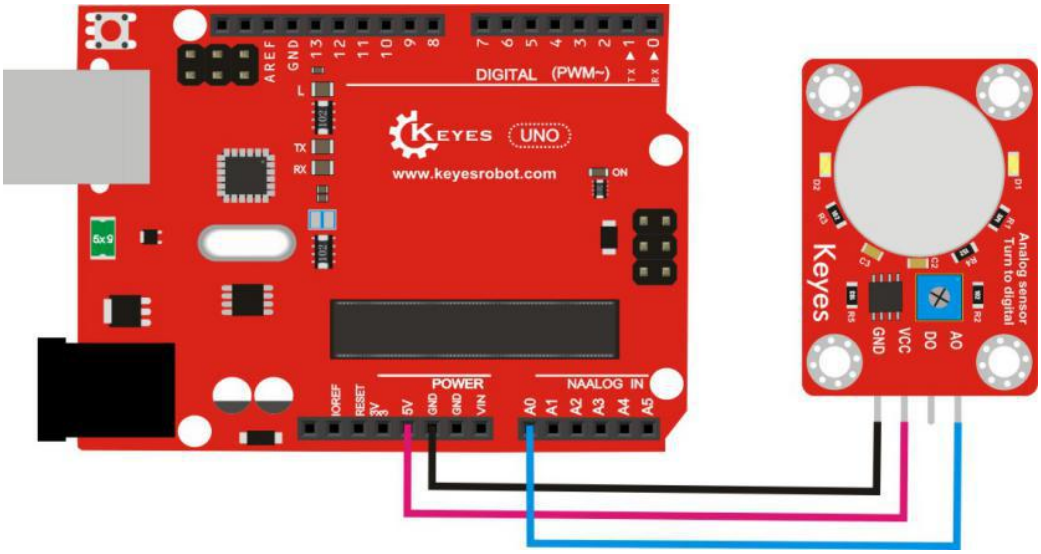
The MQ-2 gas sensor is highly sensitive to liquefied gas, propane, and hydrogen, also ideal for the detection of natural gas and other combustible vapors. It can detect a wide variety of flammable gases and is a low cost sensor for a variety of applications.

In the experiment, we connect the A0 pin of sensor to analog port A0 of UNO board, finally show you the analog output value on the serial monitor of Arduino IDE.

Part List

Development Board*1
USB Cable*1
MQ-2 Gas sensor*1
Jumper Wire*3

Wiring Diagram

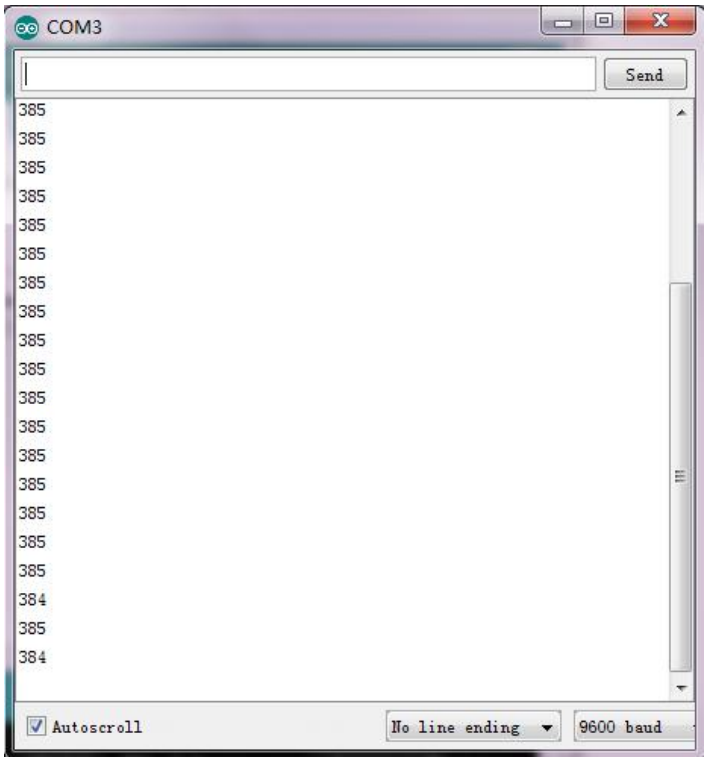


Sample Code

```
//////////////////////////////////////////////////////////////
void setup() {
  Serial.begin(9600); //set the baud rate
}
void loop() {
  int sensorValue = analogRead(A0); //read the value of A0, and assign it to digital
  variable sensorValue
  Serial.println(sensorValue, DEC); //display sensorValue and line wrap
  delay(100);   //delay 0.1 S
}
//////////////////////////////////////////////////////////////
```

Phenomenon

Hook up as the above diagram, powered up and upload well the code, D1 led on the sensor is on. You can rotate the potentiometer on the sensor to adjust the sensitivity. Open the serial monitor of Arduino IDE, you will see the corresponding analog value. The higher the detected gas concentration, the greater the analog value.



Project 26: Knock Sensor

Description

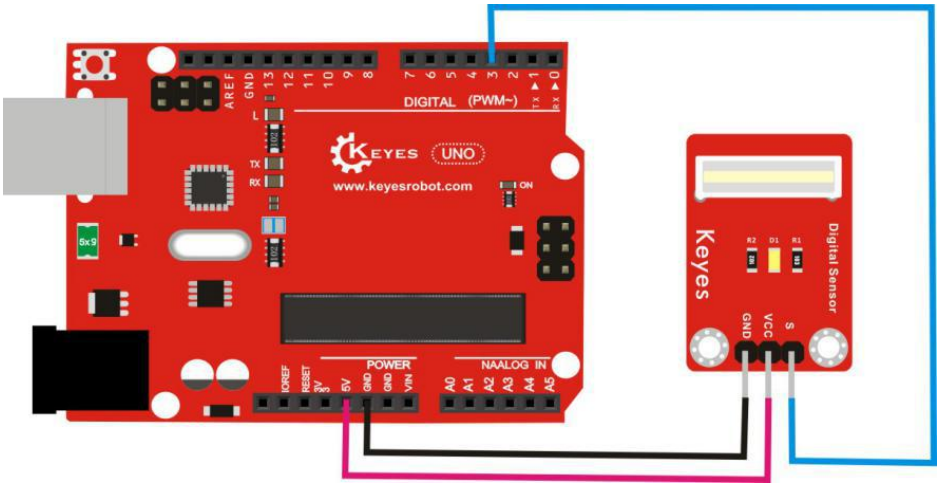
The knock sensor module is mainly composed of SW-280 vibration switch, which is an inductive proximity switch. It is an electronic switch that will transmit the sensing magnitude of the induced vibration force to the circuit device, so as to enable the circuit working.

In this experiment, we are going to test this knock sensor. Combine this sensor with D13 indicator on UNO board, finally control the state of indicator lights.

Part List

Development Board*1
USB Cable*1
Knock sensor*1
Jumper Wire*3

Wiring Diagram



Sample Code

```
//////////////////////////////////////////////////////////
int Led=13;//set the digital 13
int Shock=3;//set the digital 3
int val;//set the digital variable val
void setup()
{
pinMode(Led,OUTPUT);//set Led as output
pinMode(Shock,INPUT);//set Shock as input
}
void loop()
{
val=digitalRead(Shock);//read the value of digital 3 and assign it to val
if(val==LOW)              //if val is low level
{
digitalWrite(Led,HIGH); //Led on
}
else
{
digitalWrite(Led,LOW); //Led dims
}
}
//////////////////////////////////////////////////////////
```

Phenomenon

Hook up as the above diagram, powered up and upload well the code, then knock at the sensor, both D13 led on the UNO board and D1 led on the sensor are on.

Project 27: Capacitive Touch

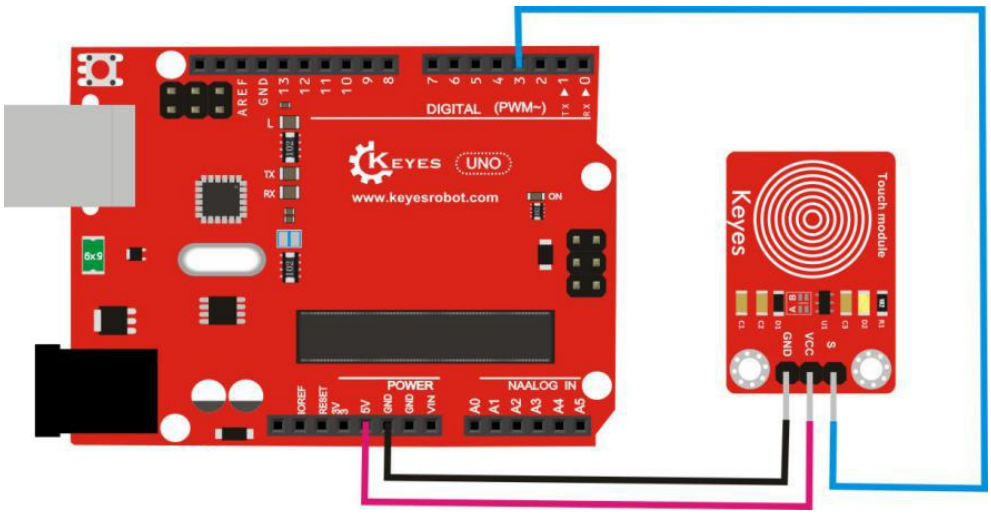
Description

In this project, we will test the capacitive touch sensor. It is mainly composed of touch detection IC. And Touch detection IC is designed to replace traditional button keys with variable area keys, featuring low power consumption and wide operating voltage.
When powered on, the sensor needs a stabilization time of about 0.5 sec. Do not touch the key during that time. At that time, all functions are forbidden. Self-calibration is always performed. When the key is not touched, the recalibration cycle is approximately 4.0sec.
This touch sensor can be used in waterproof appliances, button key substitutes and others.
In the experiment, connect the sensor to UNO board, finally control the D13 indicator on and off.

Part List

Development Board*1
USB Cable*1
Capacitive touch sensor*1
Jumper Wire*3

Wiring Diagram



Sample Code

```
//////////////////////////////////////////////////////////
int ledPin = 13; //define digital 13
int KEY = 3; //define digital 3
void setup()
  {
```

```
pinMode(ledPin, OUTPUT); //set ledPin as output
pinMode(KEY, INPUT);       //set KEY as input
}
void loop()
{
if(digitalRead(KEY)==HIGH) //if digital 3 is read as high level, LED on
{
digitalWrite(ledPin, HIGH); // LED on
}
else
{
digitalWrite(ledPin, LOW); //LED dims
}
}
//////////////////////////////////////////////////////////
```

Phenomenon

Hook up as the above diagram, powered up and upload well the code, then touch the sensor, both D2 led on the sensor and D13 indicator on UNO board are on. Otherwise, those two indicators are off.

Project 28: Joystick

Description

In this project, we will test the joystick module. It uses 5V power supply, and signal terminal X, Y are connected to the analog port.

The readout voltage is about 2.5V in the original state. When pressed along the direction of the arrow, the readout voltage increases, up to 5V. Otherwise, the readout voltage will decrease, down to a minimum of 0V.

The signal pin B of module is connected to the digital port of UNO. In the original state, output 0, if pressed, output 1.

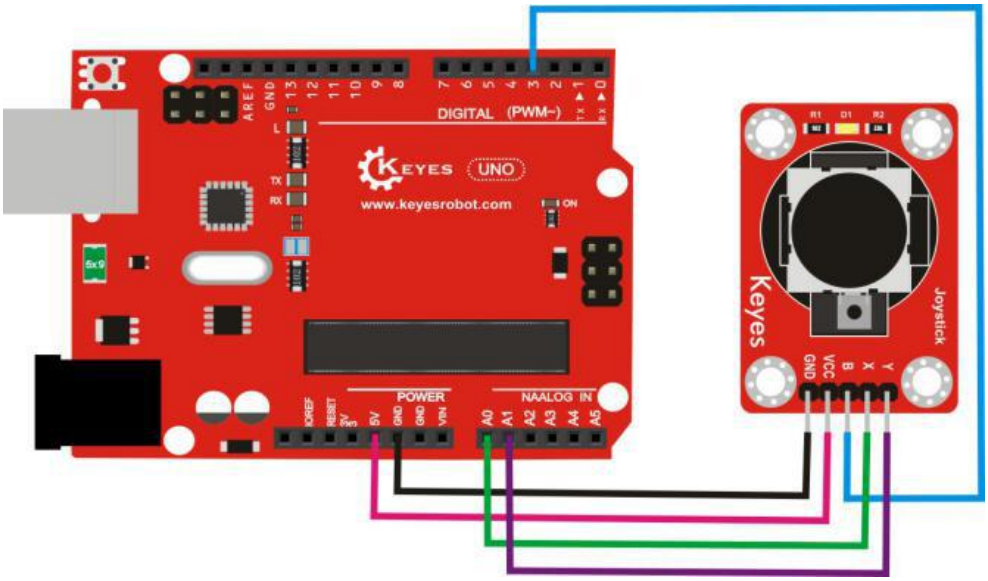
In the experiment, connect the signal ends of module to both two analog ports and one digital port of UNO board, finally the monitor of Arduino IDE will display the data. This way, you can get the specific state of joystick module.

Part List

Development Board*1
USB Cable*1
Joystick module*1

Jumper Wire*5

Wiring Diagram



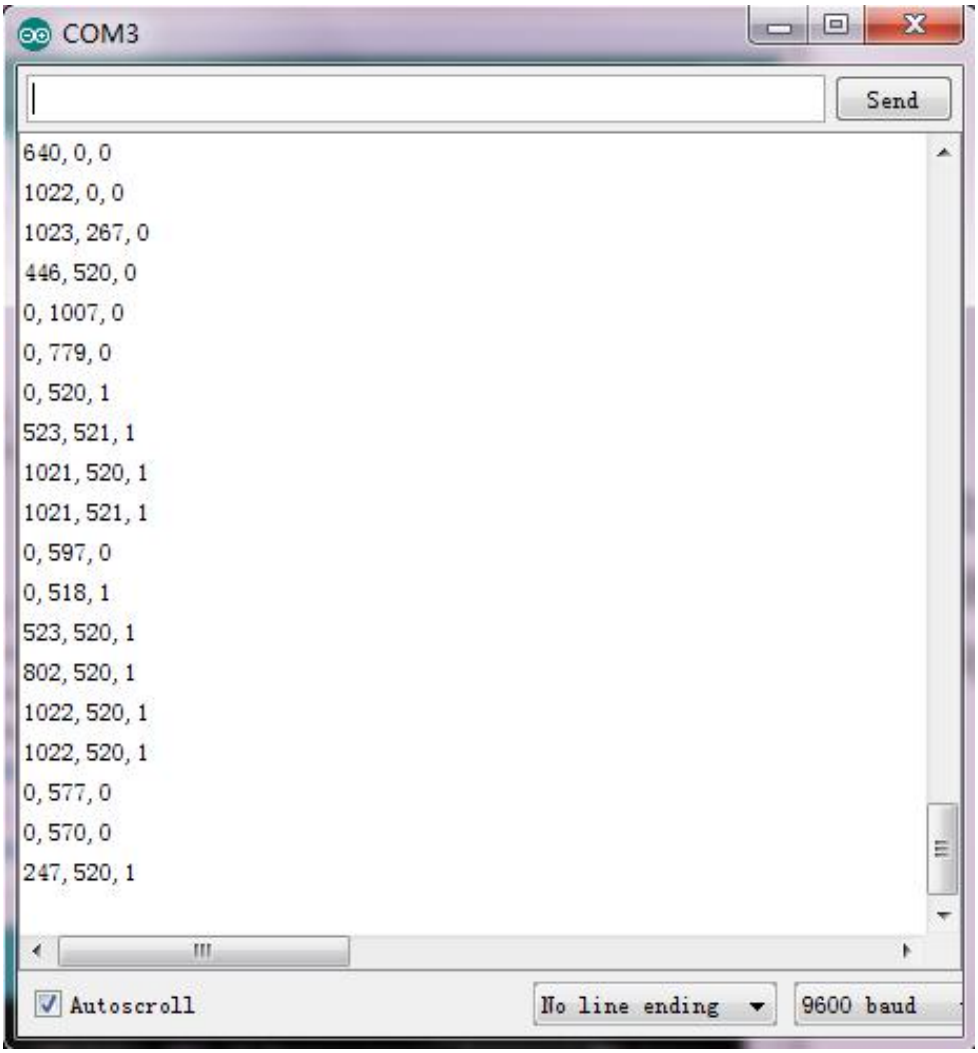
Sample Code

```
//////////////////////////////////////////////////////////
int JoyStick_X = 0; //define the analog portA0
int JoyStick_Y = 1; //define the analog port A1
int JoyStick_Z = 3; //define the digital port 3
void setup()
{
pinMode(JoyStick_Z, INPUT);//set JoyStick_Z as input
Serial.begin(9600); // set the baud rate
}
void loop()
{
int x,y,z; //define the digital variable x y z
x=analogRead(JoyStick_X); //set x as the value of A0 it read
y=analogRead(JoyStick_Y);//set y as the value of A1 it read
z=digitalRead(JoyStick_Z);//set z as the value of digital 3 it read
Serial.print(x ,DEC); //display the value of x
Serial.print(","); //display the comma
Serial.print(y ,DEC);//display the value of y
Serial.print(",");//display the comma
Serial.println(z ,DEC);//display the value of z and line wrap
delay(100);//delay 0.1S
}
//////////////////////////////////////////////////////////
```

Phenomenon

Done wiring and powered up, upload well the code, turn the X axis to the left, data range of X axis is within 0-512. If turned to the right side, data is within 512-1023. Turn the Y axis to upward, the data range of Y axis is within 0-512. If turned to downward, data range is within 512-1023. Shown as the following figure.

The data of Z axis direction is 0, if pressed down the joystick, it will display 1.



Project 29: MQ-3 Alcohol

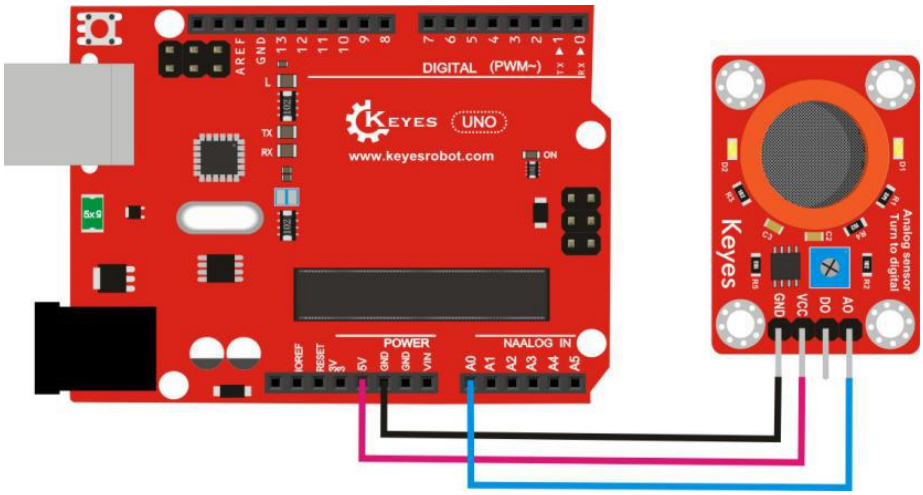
Description

In this project, we are going to detect MQ-3 alcohol sensor. The gas-sensitive material used in the sensor is tin dioxide (SnO2) with low conductivity in clean air. When there is alcohol vapor exist in the environment where the sensor is located, the conductivity of the sensor increases as the concentration of alcohol vapor in the air increases.
In the experiment, connect the A0 pin of sensor to analog port A0 of UNO board, finally show you the analog output value on the monitor of Arduino IDE.

Part List

Development Board*1
USB Cable*1
MQ-3 alcohol sensor*1
Jumper Wire*3

Wiring Diagram

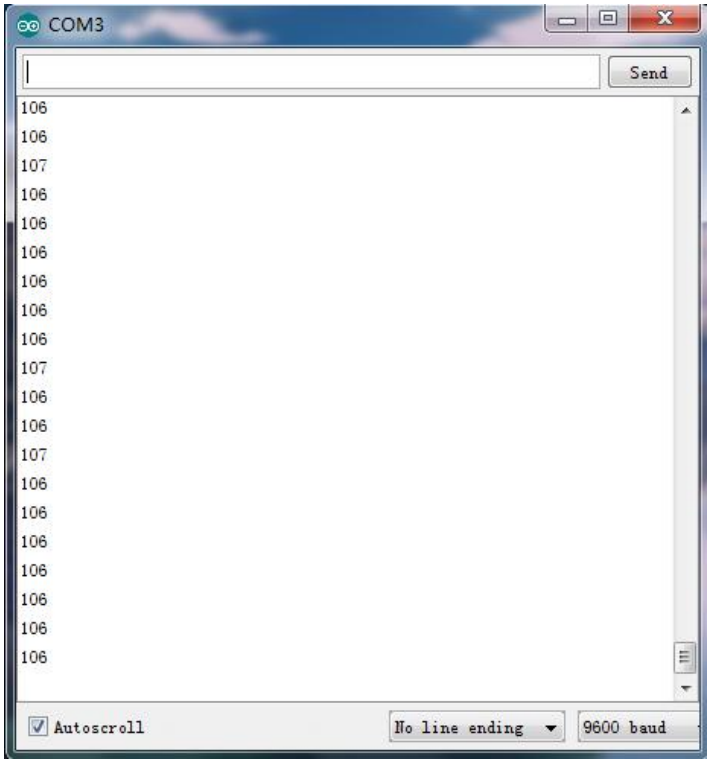


Sample Code

```
//////////////////////////////////////////////////////////
void setup() {
  Serial.begin(9600); //set the baud rate
}
void loop() {
  int sensorValue = analogRead(A0); //read the value of A0, and assign it to digital variable sensorValue
  Serial.println(sensorValue, DEC); //display sensorValue and line wrap
}
//////////////////////////////////////////////////////////
```

Phenomenon

Done wiring and powered up, upload well the code, D1 led on the sensor is on. Then open the serial monitor, you will see the corresponding analog value. You can rotate the potentiometer on the sensor to adjust the sensitivity. When putting the alcohol close to the sensor, the data showed on the monitor will increase.



Project 30: Obstacle Detector

Description

This project is going to introduce the obstacle detector module. When the obstacle module detects an object, its signal terminal outputs 0. If not detected, S terminal outputs 1. Two potentiometers on the module can be used to adjust its sensitivity.

It has fast sensing speed, suitable for obstacle avoidance of intelligent car and so on. In the experiment, combine the module with D13 indicator on UNO board, finally control the state of indicator lights on or off.

Part List

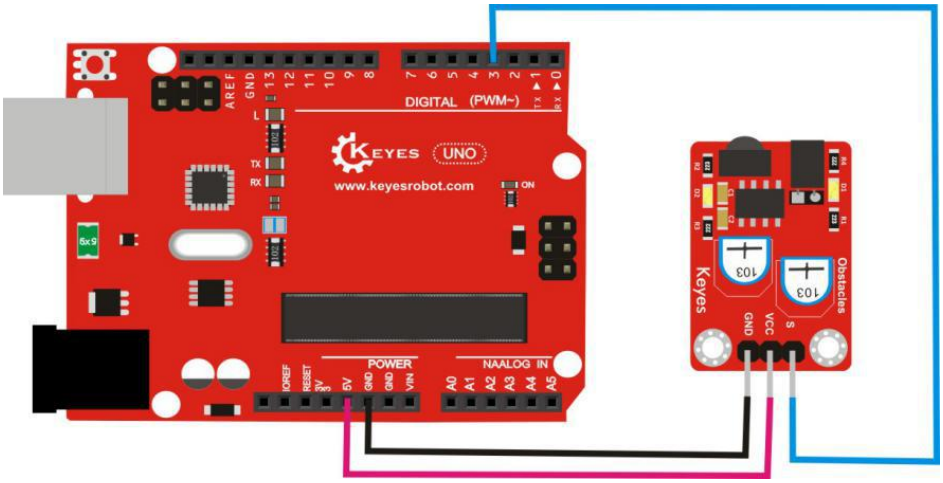
Development Board*1

USB Cable*1

Obstacle detector sensor*1

Jumper Wire*3

Wiring Diagram



Sample Code

```
//////////////////////////////////////////////////////////
const int sensorPin = 3; //define the digital 13
const int ledPin = 13; //define digital 3
int sensorState = 0; //define the digital variable sensorState and set as 0
void setup()
{
pinMode(ledPin, OUTPUT); //set ledPin as output
pinMode(sensorPin, INPUT);//set sensorPin as input
 }
void loop()
{
sensorState = digitalRead(sensorPin);
//read the value of digital 3, and assign it to sensorState
if (sensorState == LOW)   //if sensorState is low level
{
digitalWrite(ledPin, HIGH);//LED ON
}
else
{
digitalWrite(ledPin, LOW);//LED dims
}
}
//////////////////////////////////////////////////////////
```

Phenomenon

Done wiring and powered up, upload well the code, first you can rotate two

potentiometers on the module to adjust its sensitivity. This way keep the indicator light in the critical point that will be lighted up, and this time the sensing distance is the longest.

When there is no obstacle obstructs the infrared obstacle detector module, the D1 light on the sensor is off, and D13 indicator on the UNO R3 board is also off.

Once detecting the obstacles, the D1 light on the sensor lights up, and the D13 indicator on the UNO R3 board is on.

Project 31: LM35 Temperature Detection

Description

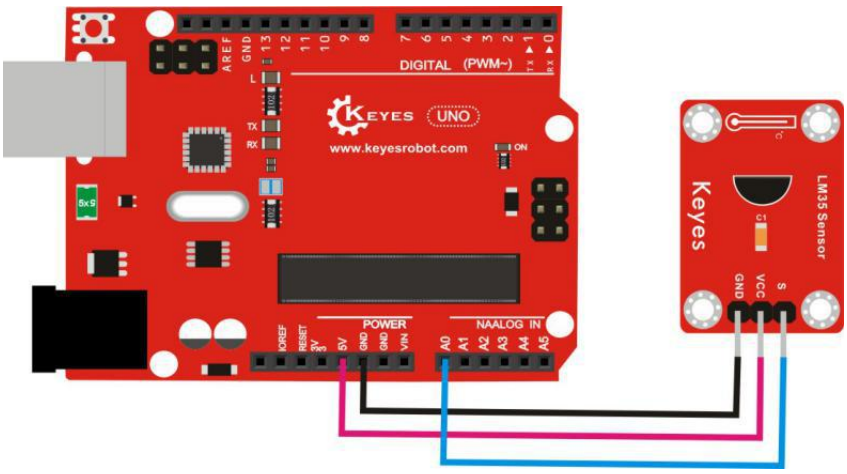
This time let's move on to detect the LM35 temperature sensor. Its output voltage has linear relation with Celsius temperature scale. The conversion formula is as follows. When it's 0℃, the output is 0V. For each 1℃ increase, the output voltage increases by 10mV.

In the experiment, connect the signal pin of sensor to analog A0 of UNO board, finally show you the temperature value of current ambient on the serial monitor of Arduino software.

Part List

Development Board*1
USB Cable*1
LM35 temp sensor*1
Jumper Wire*3

Wiring Diagram

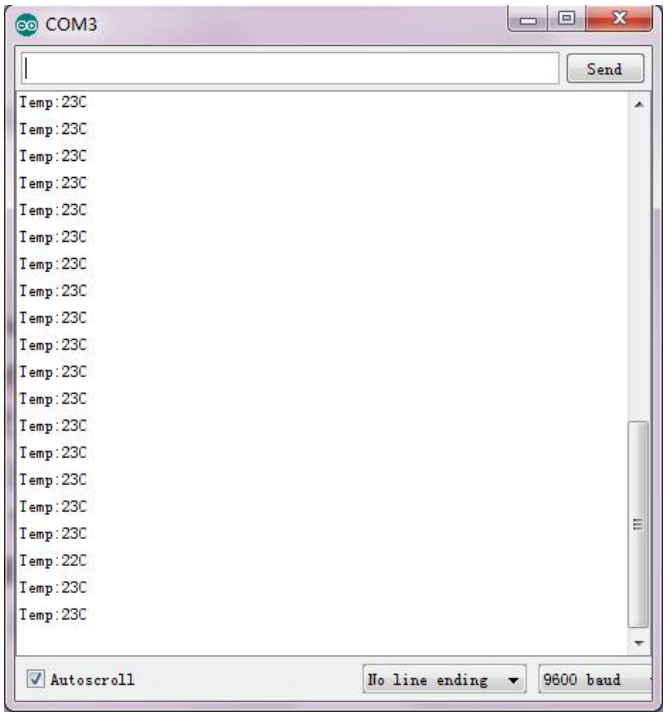


Sample Code

```
//////////////////////////////////////////////////////////
void setup()
{
Serial.begin(9600);//set the baud rate
   }
 void loop()
{
int val; //define the digital variable val
int dat;//define the digital variable dat
val=analogRead(0);//set val as the readout value of analog A0
dat=(500 * val) /1024; //calculate the temperature data dat
Serial.print("Temp:"); //display Temp:
Serial.print(dat);    //display the calculated temperature data
Serial.println("C");//display C and line wrap
delay(500); //delay 0.5S
}
//////////////////////////////////////////////////////////
```

Phenomenon

Done wiring and powered up, upload well the code, then open the serial monitor of Arduino software, you will see the temperature data of current ambient. Shown below.



Project 32: PIR Motion Detection

Description

The human body infrared pyroelectric sensor is an automatic control product based on infrared technology. It has the characteristics of high sensitivity, strong reliability, ultra-low power consumption, ultra-low voltage operation mode and so on.

It is widely used in various types of automatic sensing electrical equipment, especially dry battery-powered automatic control products.

You can adjust the sensing distance and sensing delay by adjusting two potentiometers on the back of sensor. Rotate the potentiometer clockwise to adjust the distance, this way sensing distance will enlarge (about 7 meters), or else it will reduce (about 3 meters).

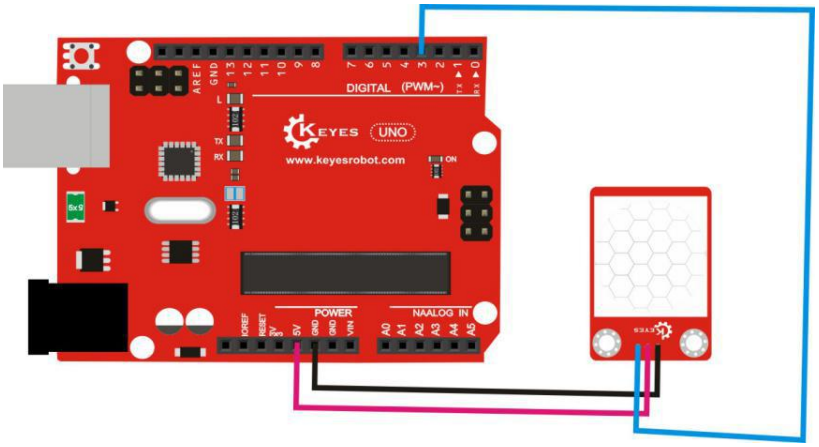
Rotate the potentiometer clockwise to adjust the relay, this way sensing relay will lengthen (about 300S), or else it will reduce (about 5S).

In this project, we are going to use this PIR motion sensor to detect whether there is someone moving nearby, showing you the result on serial monitor.

Part List

Development Board*1
USB Cable*1
PIR motion sensor*1
Jumper Wire*3

Wiring Diagram



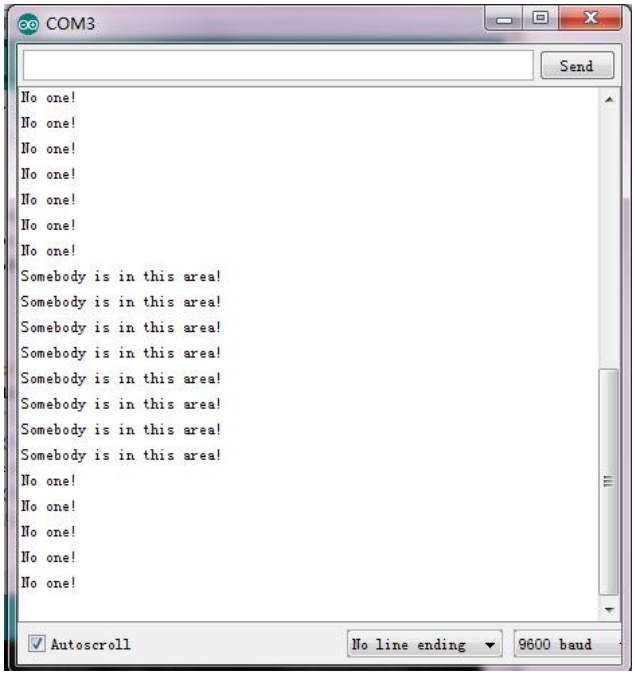
Sample Code

```
//////////////////////////////////////////////////////////
byte sensorPin = 3;//define the digital 3
byte indicator = 13;//define the digital 13
void setup()
{
```

```
  pinMode(sensorPin,INPUT);//set the digital 3 as input
  pinMode(indicator,OUTPUT);//set the digital 13 as output
  Serial.begin(9600);//set the baud rate
}
void loop()
{
  byte state = digitalRead(sensorPin);//readout the value of digital 3 and assign it to
state
  digitalWrite(indicator,state);//control the state of digital 13
  if(state == 1)//if digital 3 is high level, the monitor will print out corresponding
    characters and line wrap.
  Serial.println("Somebody is in this area!");
  else if(state == 0) //if digital 3 is low level, the monitor will print out corresponding
characters and line wrap.
  Serial.println("No one!");
  delay(500);//delay 0.5 S
}
//////////////////////////////////////////////////////////
```

Phenomenon

Done wiring and powered up, upload well the code, if the sensor detects someone moving nearby, D13 indicator on UNO board will light up, and "Somebody is in this area!" is displayed on the serial monitor of Arudino software. If no detecting the movement, D13 indicator on UNO board will be off, and "No one!" is displayed on the serial monitor.



Project 33: Laser Sensor

Description

The laser sensor is mainly composed of a laser head where consists of a light emitting core, a condenser lens, and an adjustable copper sleeve. After connecting the power supply, directly input a high-level digital signal at the signal terminal and the sensor starts to work.

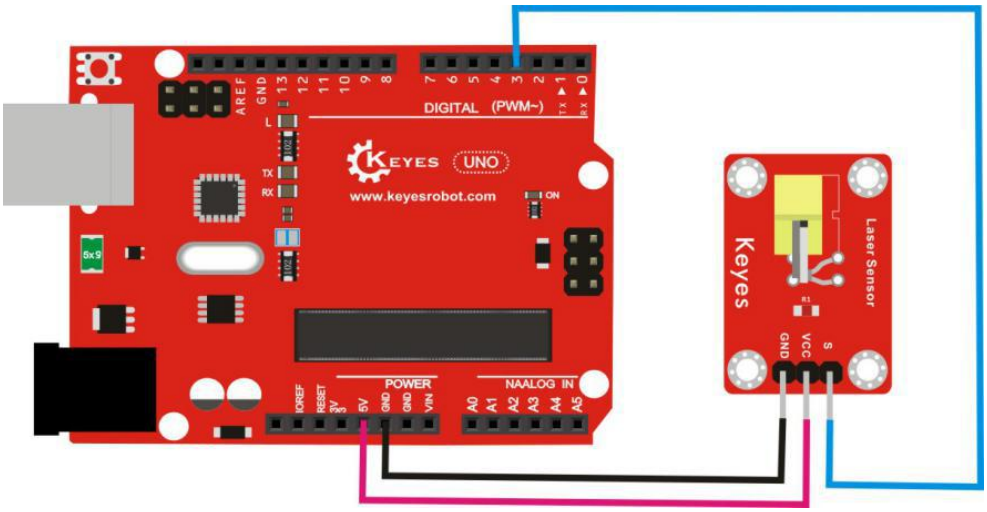
It can be used in laser toys, electronic pointer pens, electronic level gauges, miniature liquid crystal projectors and so on.

In the following, we are going to test this laser sensor to be turned on and off circularly.

Part List

Development Board*1
USB Cable*1
Laser sensor*1
Jumper Wire*3

Wiring Diagram



Sample Code

```
//////////////////////////////////////////////////////////
void setup()
{
pinMode(3, OUTPUT); // set the pin 3 as digital interface
}
void loop() {
digitalWrite(3, HIGH); // turn on laser
delay(1000); // delay one second
```

```
digitalWrite(3, LOW); // turn off laser
delay(1000); //    delay one second
}
//////////////////////////////////////////////////////////
```

Phenomenon

Done wiring and powered up, upload well the code, the laser head is turned on for one second, then off for one second, alternately and circularly.

Project 34: Line Tracking

Description

The line tracking module is mainly composed of TCRT5000 infrared tube. The working principle of TCRT5000 infrared tube is that because infrared reflectivity is different for colors, it can convert the strength of reflected signal into a current signal.

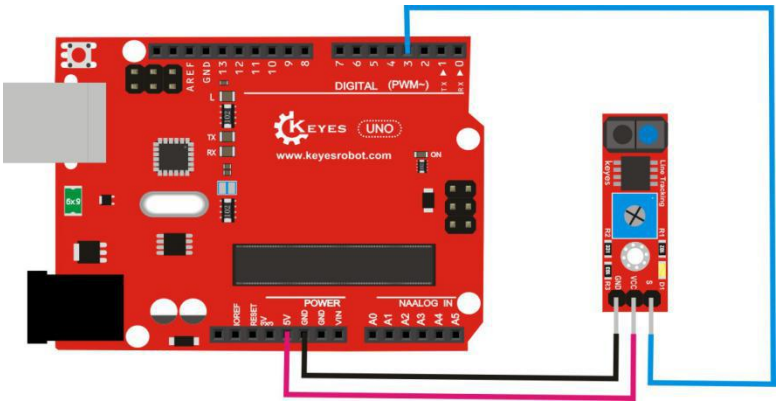
The sensor is active high level at detecting the black line, and is active low level at detecting white line. The detecting height is 0-3 cm. You can rotate the potentiometer on the sensor to adjust the sensitivity for black-white line tracking.

In the following, we are going to detect the line tracking sensor. Connect the signal pin of sensor to digital port 3 of UNO board, finally show the output data on the serial monitor of Arduino software.

Part List

Development Board*1
USB Cable*1
Tracking sensor*1
Jumper Wire*3

Wiring Diagram



Sample Code

```
//////////////////////////////////////////////////////////
void setup()
{
  Serial.begin(9600);//set the baud rate
}
void loop()
{
  Serial.println(digitalRead(3)); // output the value read from digital 3, and line wrap.
  delay(500);//delay 0.5S
}
//////////////////////////////////////////////////////////
```

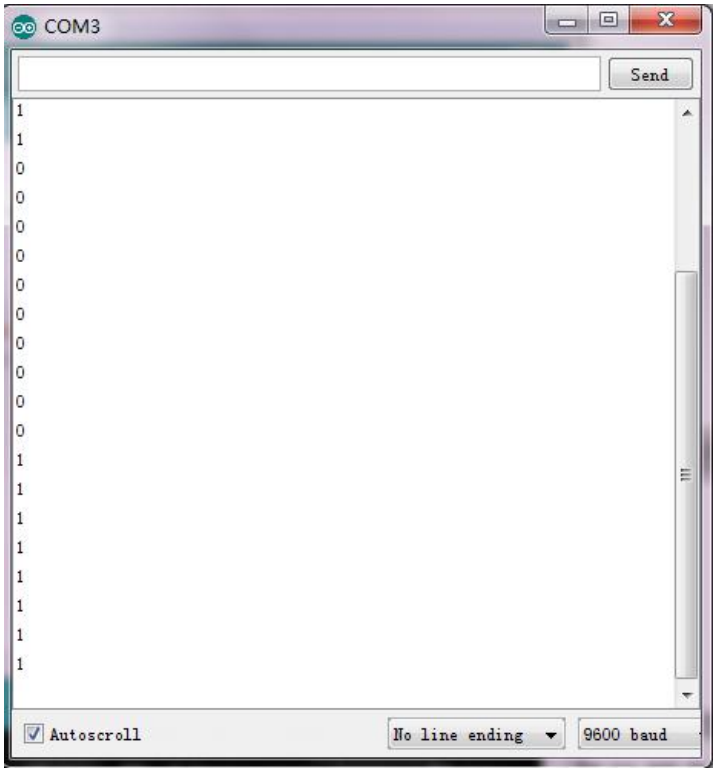
Phenomenon

Done wiring and powered up, upload well the code, then open the serial monitor and set the baud rate as 9600.

When the sensor detects the black line, its signal terminal outputs high level, and monitor displays 1, as well as the D1 indicator on the sensor is off.

When the sensor detects other colors of line, its signal terminal outputs low level, and monitor displays 0, as well as the D1 indicator on the sensor is on.

You can rotate the potentiometer on the module to adjust its sensitivity. When D1 indicator is at the critical point of on and off state, its sensitivity is the best.



Project 35: 18B20 Temperature

Description

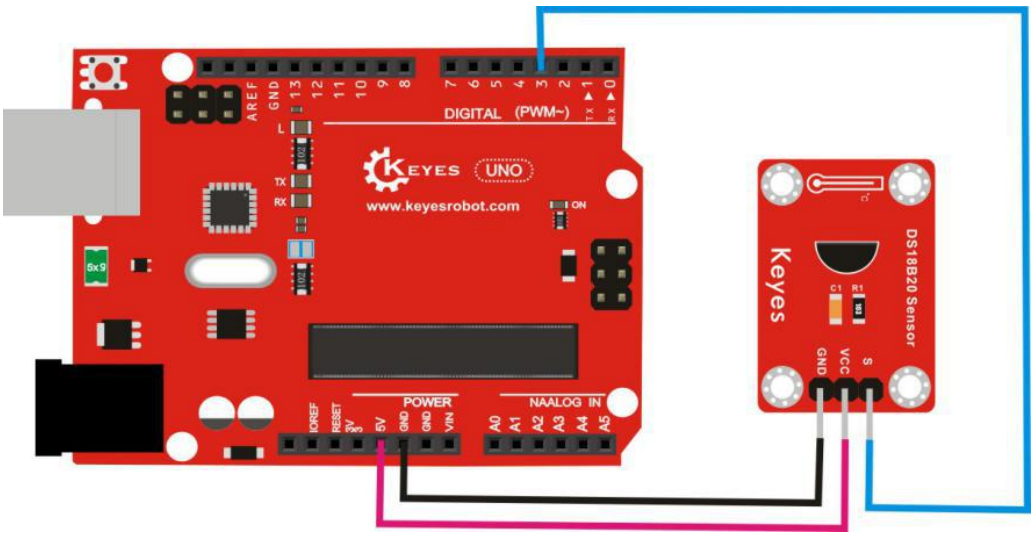
In this experiment we are going to test the 18B20 temperature sensor. The sensor is mainly composed of DS18B20 programmable digital temperature device. It has the characteristics of small volume, strong anti-interference ability and high precision. Its tested temperature range is from -55℃ to +125℃, and the inherent temperature error is 1℃.

In the following, connect the signal pin of sensor to digital port 3 of UNO board, finally show you the temperature value on the monitor of Arduino software.

Part List

Development Board*1
USB Cable*1
18B20 temperature sensor*1
Jumper Wire*3

Wiring Diagram



Sample Code

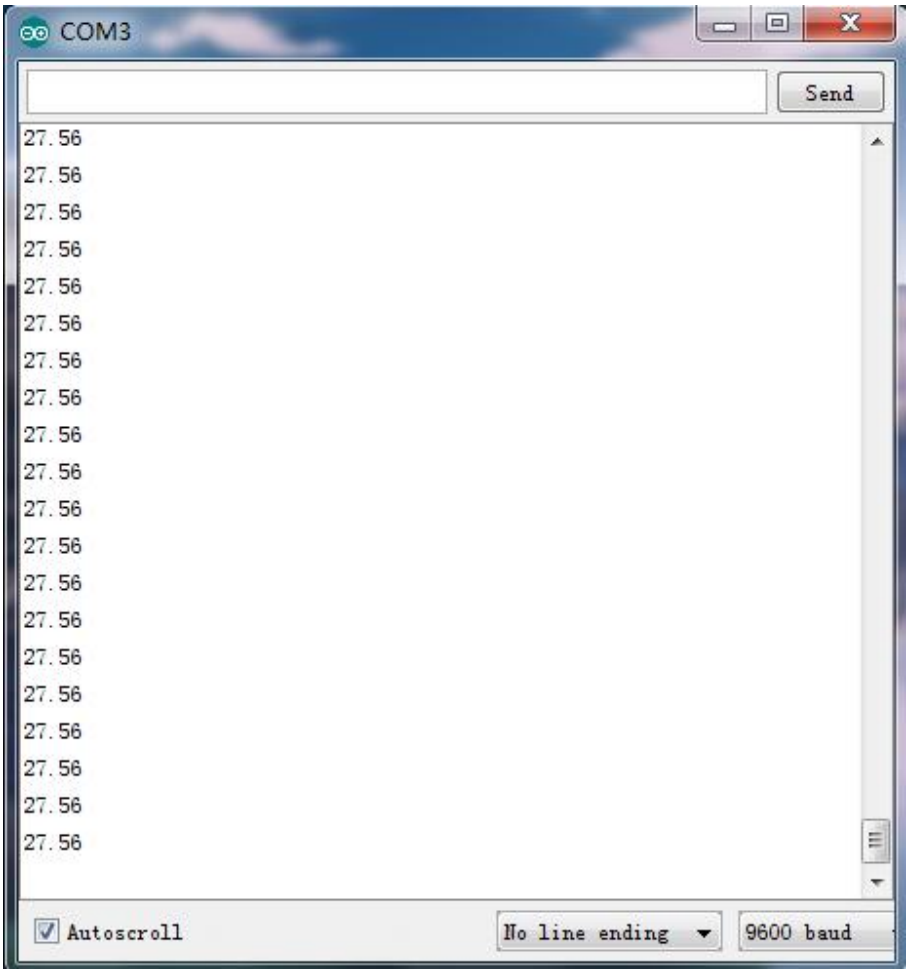
```
//////////////////////////////////////////////////////////////
#include <OneWire.h>
int DS18S20_Pin = 3; //define the digital port 3
OneWire ds(DS18S20_Pin);
 void setup(void) {
  Serial.begin(9600); //set the baud rate
}
 void loop(void) {
  float temperature = getTemp(); //calling the function to calculate the temperature
```

```
value
  Serial.println(temperature); //display the temperature value and line wrap
  delay(100); //delay 0.1S
}
float getTemp(){
  //returns the temperature from one DS18S20 in DEG Celsius
  byte data[12];
  byte addr[8];
  if ( !ds.search(addr)) {
      //no more sensors on chain, reset search
      ds.reset_search();
      return -1000;
  }
  if ( OneWire::crc8( addr, 7) != addr[7]) {
      Serial.println("CRC is not valid!");
      return -1000;
  }
  if ( addr[0] != 0x10 && addr[0] != 0x28) {
      Serial.print("Device is not recognized");
      return -1000;
  }
  ds.reset();
  ds.select(addr);
  ds.write(0x44,1); // start conversion, with parasite power on at the end
  byte present = ds.reset();
  ds.select(addr);
  ds.write(0xBE); // Read Scratchpad
  for (int i = 0; i < 9; i++) { // we need 9 bytes
    data[i] = ds.read();
  }
  ds.reset_search();
  byte MSB = data[1];
  byte LSB = data[0];
  float tempRead = ((MSB << 8) | LSB); //using two's compliment
  float TemperatureSum = tempRead / 16;
  return TemperatureSum;
}
////////////////////////////////////////////////////////////
```

Libraries Download Link:
https://pan.baidu.com/s/1o7HKVKQ

Phenomenon

Done wiring and powered up, upload well the code, then open the serial monitor and set the baud rate as 9600, finally you will see the temperature data in current environment. Shown below.



Project 36: TEMT6000 Lighting

Description

In this project, we are going to test the TEMT6000 light sensor. The sensor is mainly composed of a visible light photosensitive (NPN type) transistor of high sensitivity, which can capture the tiny light changes to magnify it about 100 times, and easily recognized by the microcontroller to make AD conversion.

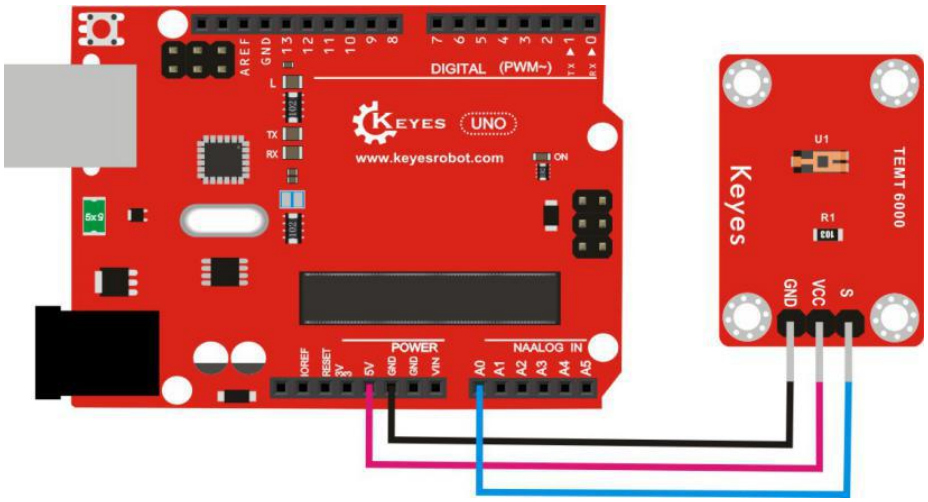
Its response to visible light illumination is similar to that of the human eye, simulating how people judge the intensity of ambient light, and making it easy to make human-friendly interactive works.

In the following, connect the signal pin of sensor to analog port A0 of UNO board, finally show you the analog value output on the monitor of Arduino software.

Part List

Development Board*1
USB Cable*1
TEMT6000 light sensor*1
Jumper Wire*3

Wiring Diagram



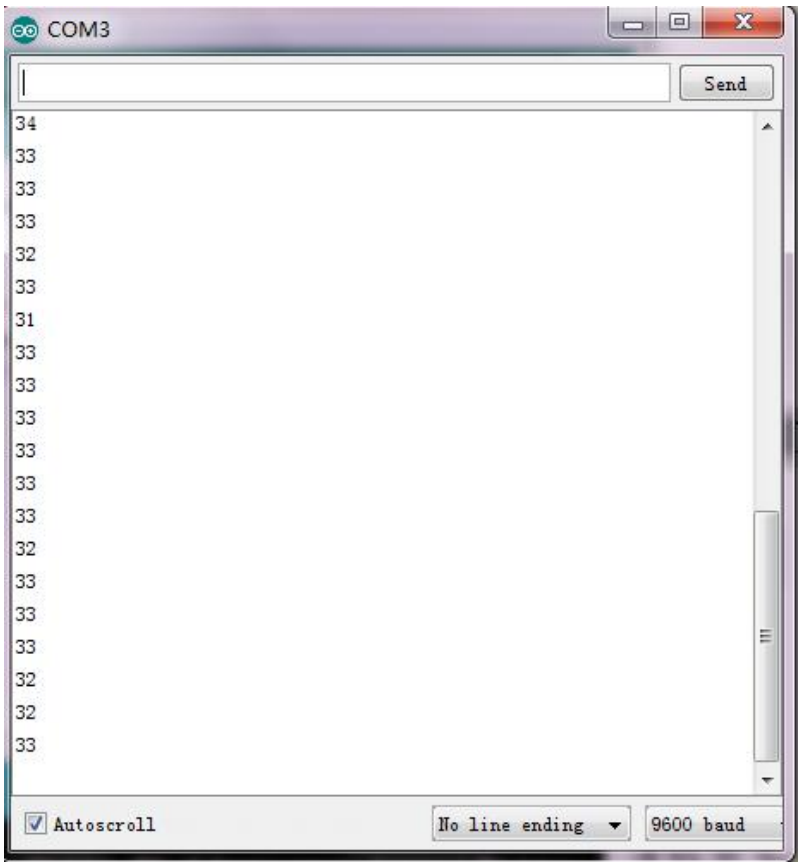
Sample Code

```
//////////////////////////////////////////////////////////
void setup() {
  Serial.begin(9600); //set the baud rate
}
void loop() {
  int sensorValue = analogRead(A0); //read the value of A0 and assign it to digital variable sensorValue
  Serial.println(sensorValue, DEC); //display sensorValue and line wrap
```

```
  delay(100); //delay 0.1S
}
//////////////////////////////////////////////////////////
```

Phenomenon

Done wiring and powered up, upload well the code, then open the serial monitor and set the baud rate as 9600, finally you will see the analog value of current light intensity. Shown below.



6. Related Data Link

https://pan.baidu.com/s/1i51WsFN