CATALOG

Module List

S/N	Product Name	Product Image	S/N	Product Name	Product Image
1	0.96 inch yellow and blue two-color OLED screen		10	SR501 Infrared Sensor	
2	1602 Liquid Crystal Display Module		11	Soil Moisture Detection Module	
3	1-Way Relay Module		12	Obstacle Avoidance Sensor	
4	DHT11 Temperature and Humidity Sensor		13	Breadboard(4.5*3.5)	
5	HC-SR04 Ultrasonic Sensor		14	SG90 Servo Motor	
6	8*8 Red Dot Matrix Screen		15	Diode (red)	
7	Joystick Module		16	Diode (green)	
8	4-digit Digital Tube		17	Resistance	
9	TTP223B Touch Sensor		18	DuPont Cable (Male to female)	

Module 1:0.96 inch OLED Screen

1. Introduction

0.96 Inch OLED Module for showing graphical & textual information directly on your micro-controller projects. It supports many chips: Arduino UNO and Mega, Raspberry pi, 51 MCU, STIM 32, etc.,

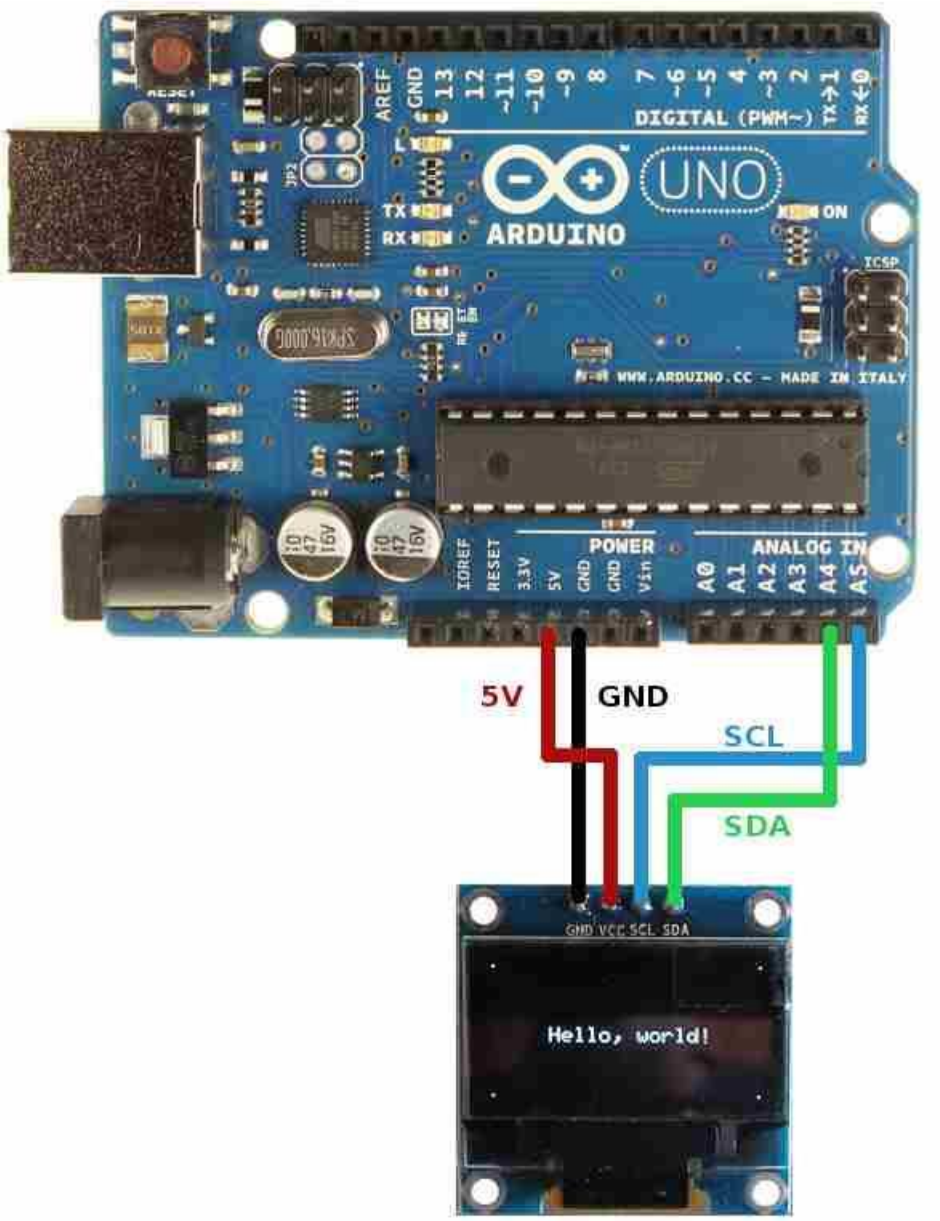
Resolution: 128 x 64, View angle: > 160° , Support voltage: 3.3V-5V DC, Power consumption: 0.04W during normal operation, full screen lit 0.08W

Embedded Driver IC: SSD1306. Communication: I2C/IIC Interface, only need two I / O ports

This OLED display module is small, only 0.96” diagonal, it is made of 128x64 individual yellow and blue OLED pixels, each one is turn on or off by the controller chip.

The Driver chip of this OLED is SSD1306, which is compatible with IIC communication. So this module can be controlled by I2C. That is, except the VCC and GND, 2 wires would be needed when using 4-wires I2C mode.

2. Example(Work with Arduino)



2.1 Pin Map:

0.96 OLED	Arduino
VCC	3.3V / 5V
GND	GND
A4	SDA
A5	SCL

2.2 Arduino Code:

Add two class libraries to the Arduino IDE(You can save it to your computer by right clicking on the two attachments below):

Adafruit_GFX.zip

Adafruit_SSD1306.zip

```
#include <SPI.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

#define OLED_RESET 4
Adafruit_SSD1306 display(OLED_RESET);
//Define display height
#define LOGO16_GLCD_HEIGHT 16
//Define display width
#define LOGO16_GLCD_WIDTH  16

#if (SSD1306_LCDHEIGHT != 64)
#error("Height incorrect, please fix Adafruit_SSD1306.h!");
#endif

void setup()   {
  Serial.begin(9600);
//by default, we'll generate the high voltage from the 3.3v line //internally! (neat!)
//initialize with the I2C addr 0x3D (for the 128x64)
  display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
  // init done

  display.clearDisplay(); //Clear screen
  display.setTextSize(1); //Set the font size
  display.setTextColor(WHITE);  //Set font color white
  display.setCursor(0,0); //Set the starting position of the font
  display.println("Hello, world!"); //Output characters and wrap
  //Set font black, font background white
  display.setTextColor(BLACK, WHITE);
  display.println("0.123456789"); //Output numbers and wrap
  display.setTextSize(1);
  display.setTextColor(WHITE);
  display.println("--------------------");

  display.setTextSize(2);
  display.setTextColor(WHITE);
  display.print("ideaSpark!");
  display.print("0.96 OLED!");

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

Module 2:1602 Liquid Crystal Display Module

1. Introduction

As we all know, though LCD and some other displays greatly enrich the man-machine interaction, they share a common weakness. When they are connected to a controller, multiple IOs will be occupied of the controller which has no so many outer ports. Also it restricts other functions of the controller. Therefore, LCD1602 with an I2C port is developed to solve the problem. It is also simple to handle. It has only two bidirectional data lines, a serial data line and a serial clock one.

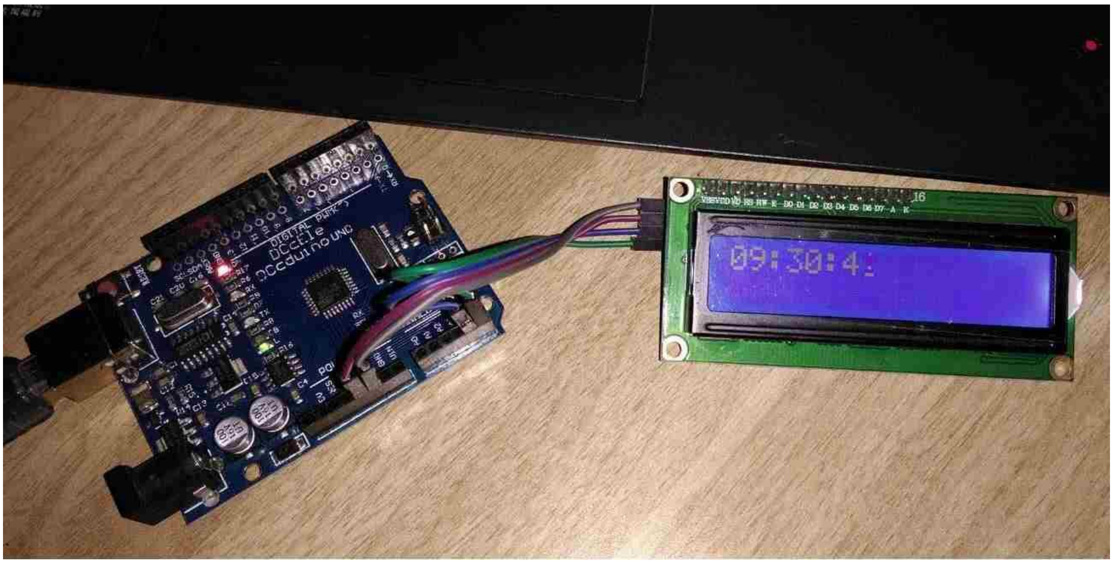
Easy to use. Less I/O ports are occupied, only four - VCC, GND, SDA (serial data line), SCL (serial clock line).

Support IIC protocol. The I2C LCD1602 library is provided, so you can call it directly.

With a potentiometer used to adjust backlight and contrast.

Power supply: +5V,Address of the module: ox27

2. Example(Work with Arduino)



2.1 Pin Map:

1602 LCD	Arduino
VCC	5V
GND	GND
A4	SDA
A5	SCL

2.2 Arduino Code:

Add class libraries to the Arduino IDE(You can save it to your computer by right clicking on the attachments below):

LiquidCrystal_I2C-master.zip

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h> //add I2C library
//Set the LCD1602 device address,0x3F,0x20,or 0x27
LiquidCrystal_I2C lcd(0x27,16,2);
void setup(){
  lcd.init(); //Initialize the LCD
  lcd.backlight();  //Set LCD background bright
}
void loop(){
  lcd.setCursor(0,0);
  lcd.print("1602 LCD Display");
  lcd.setCursor(0,1);
  lcd.print("    by ideaSpark");
  delay(1000);
}
```

Module 3:1-Channel Relay Module

1. Introduction

This Relay Board provides a single relay that can be controlled by any 5V digital output from your microcontroller

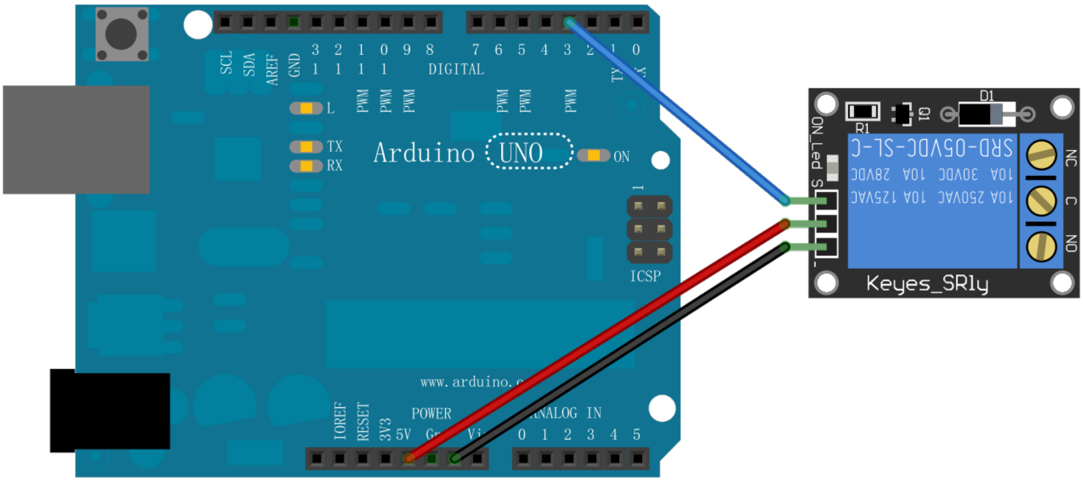
Peak Load of NO Connector: AC125~250V/10A, DC28~30V/10A; Trigger Current: 5mA; 4pcs * 3.1 mm Fixed Bolt Hole

The isolation circuit prevent damages to I / O port by relay switch current

These 5v relay board works well with Arduino/ ARM /PIC /AVR /MCU/Raspberry/CNC machine/ PS4 etc

All interfaces can be directly wired out through the terminals

2. Example(Work with Arduino)



2.1 Pin Map:

Relay Module	Arduino
IN	3
VCC	5V
GND	GND

2.2 Arduino Code:

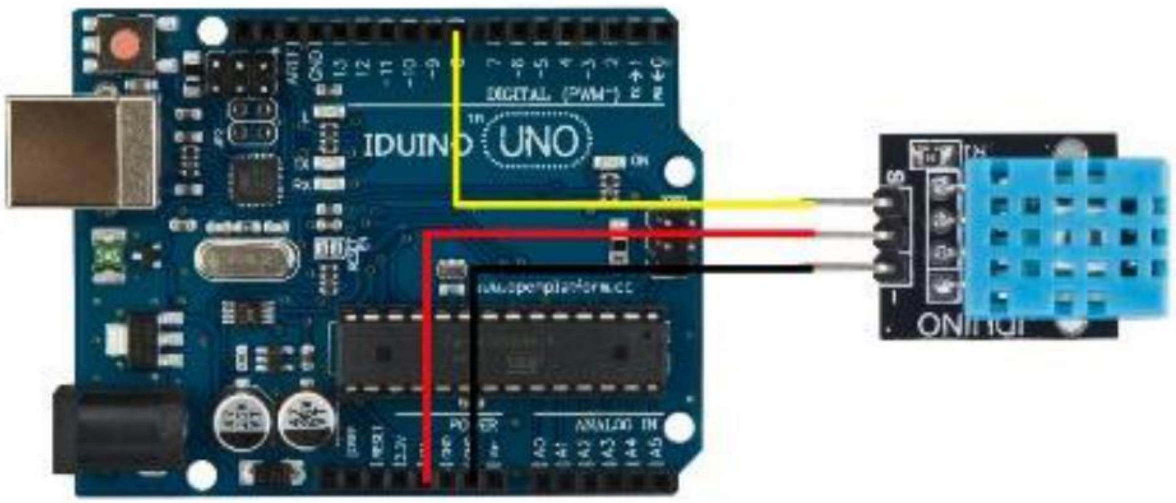
```
int sign= 3;
void setup(){
 pinMode(sign, OUTPUT);
}

void loop(){
  digitalWrite(sign, HIGH);
  delay(2000);
  digitalWrite(sign, LOW);
  delay(2000);
}
```

Module 4:DHT11 temperature and humidity sensor

1. Introduction

DHT11 digital temperature and humidity sensor is a composite Sensor contains a calibrated digital signal output of the temperature and humidity. Application of a dedicated digital modules collection technology and the temperature and humidity sensing technology,to ensure that the product has high reliability and excellent long-term stability. The sensor includes a resistive sense of wet components and an NTC temperature measurement devices, and connected with a high-performance 8-bit microcontroller

2. Example(Work with Arduino)



2.1 Pin Map:

DHT11	Arduino
+	5V
-	GND
OUT	D8

2.2 Arduino Code:

```
int DHpin = 8;
byte dat[5];

byte read_data () {
 byte data;
 for (int i = 0; i < 8; i ++) {
 if (digitalRead (DHpin) == LOW) {
    //wait for 50us
    while (digitalRead (DHpin) == LOW);
//determine the duration of high level to determine the data is '0 'or '1'
    delayMicroseconds (30);
    if (digitalRead (DHpin) == HIGH)
    //high front and low in the post
    data |= (1 << (7-i));
    //data '1 ',wait for the next one receiver
    while (digitalRead (DHpin) == HIGH);
  }
 }
    return data;
}

void start_test (){
  //bus down, send start signal
  digitalWrite (DHpin, LOW);
  //delay greater than 18ms, so DHT11 start signal can be detected
  delay (30);
  digitalWrite (DHpin, HIGH);
  //Wait for DHT11 response
  delayMicroseconds (40);
  pinMode (DHpin, INPUT);
  while (digitalRead (DHpin) == HIGH);
  //DHT11 response, pulled the bus 80us
  delayMicroseconds (80);
  if(digitalRead (DHpin) == LOW);
  //DHT11 80us after the bus pulled to start sending data
  delayMicroseconds (80);
  //receive temperature and humidity data, the parity bit is not considered
  for (int i = 0; i < 4; i ++)
  dat[i] = read_data ();
  pinMode (DHpin, OUTPUT);
 //send data once after releasing the bus, wait for the host to open the next Start signal
  digitalWrite (DHpin, HIGH);
}

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

void loop () {
 start_test ();
 Serial.print ("Current humdity =");
 //display the humidity-bit integer
```

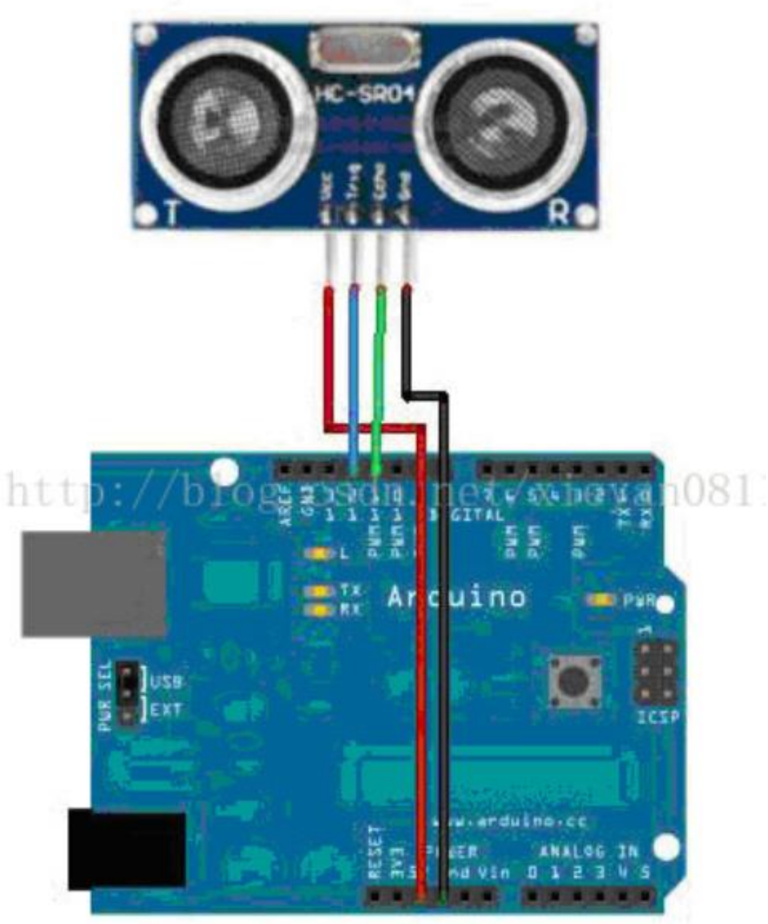
```
Serial.print (dat [0], DEC);
//display the humidity decimal places;
Serial.print ('.'); Serial.print (dat [1], DEC);
Serial.println ('%');
Serial.print ("Current temperature =");
//display the temperature of integer bits
Serial.print (dat [2], DEC);
Serial.print ('.');
//display the temperature of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

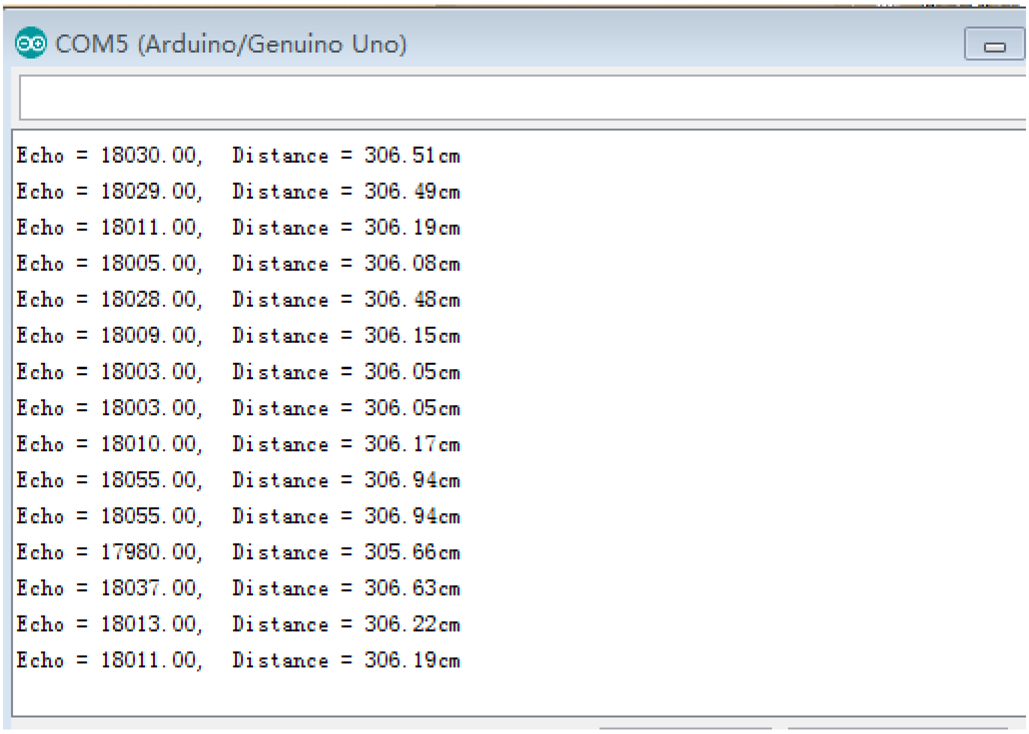
Module 5:HC-SR04 Ultrasonic sensor

1. Introduction

The HC-SR04 ultrasonic ranging module can provide 2cm-400cm non-contact distance sensing function, the ranging accuracy can be up to 3mm; the module includes ultrasonic transmitter, receiver and control circuit

2. Example(Work with Arduino)





```
Echo = 18030.00,  Distance = 306.51cm
Echo = 18029.00,  Distance = 306.49cm
Echo = 18011.00,  Distance = 306.19cm
Echo = 18005.00,  Distance = 306.08cm
Echo = 18028.00,  Distance = 306.48cm
Echo = 18009.00,  Distance = 306.15cm
Echo = 18003.00,  Distance = 306.05cm
Echo = 18003.00,  Distance = 306.05cm
Echo = 18010.00,  Distance = 306.17cm
Echo = 18055.00,  Distance = 306.94cm
Echo = 18055.00,  Distance = 306.94cm
Echo = 17980.00,  Distance = 305.66cm
Echo = 18037.00,  Distance = 306.63cm
Echo = 18013.00,  Distance = 306.22cm
Echo = 18011.00,  Distance = 306.19cm
```

2.1 Pin Map:

HC-SR04	Arduino
VCC	5V
GND	GND
Echo	D11
Trig	D12

2.2 Arduino Code:

```
#define PIN_TRIG 12
#define PIN_ECHO 11
float cm;
float temp;

void setup(){
  Serial.begin(9600);
  pinMode(PIN_TRIG, OUTPUT);
  pinMode(PIN_ECHO, INPUT);
}

```

```
void loop() {
  digitalWrite(PIN_TRIG, LOW);
  delayMicroseconds(2);
  digitalWrite(PIN_TRIG, HIGH);
  delayMicroseconds(10);
  digitalWrite(PIN_TRIG, LOW);
  temp = float(pulseIn(PIN_ECHO, HIGH));
  cm = (temp * 17 )/1000;
  Serial.print("Echo = ");
  Serial.print(temp);
  Serial.print(",  Distance = ");
  Serial.print(cm);
  Serial.println("cm");
  delay(300);
}
```

Module 6:8x8 Dot Matrix Display

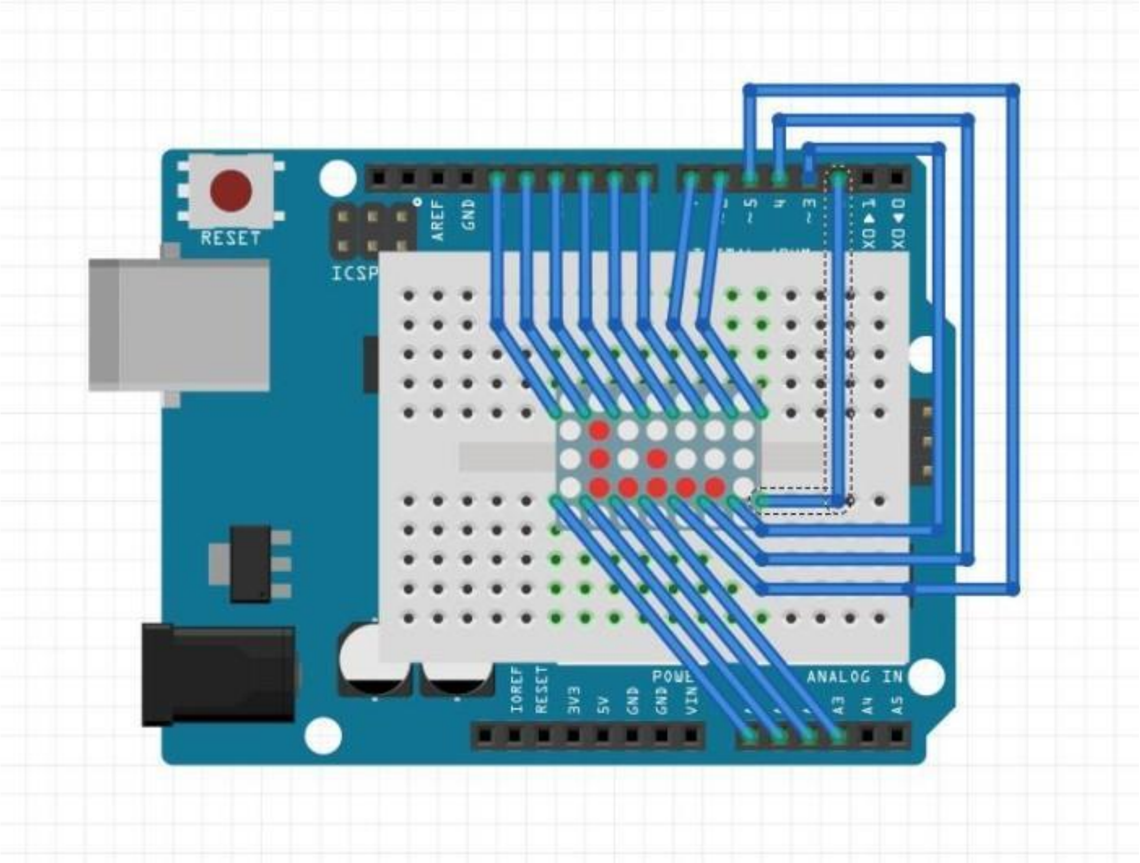
1. Introduction

This 8x8 dot matrix LED (red) module is ideally suited for use with various microcontrollers such a Pic, Atmel AVR, and Arduino.

- Display colour: Red
- Type: LED dot matrix
- Common Anode
- Brightness: High brightness
- Model: 1588BS
- Module size: 38mm X 38mm X 8mm(L * W * H)
- Pixel diameter: 3.75 (mm)
- Pixel pitch: 4.76 * 7 (mm)

Two-axis pushbutton rocker sensor module by joystick potentiometer with (X, Y) 2 axis analog output, (Z) 1 button digital output. With the Arduino sensor expansion board can be produced such as remote control interactive works. In addition, this product In order to allow customers to more easily with Arduino expansion board and other standard interface, the design of the X, Y, Z axis of the circuit leads alone, the user can use 3-pin ARDUINO special line really plugged into the expansion board for Use.

2. Example(Work with Arduino)



2.1 Pin Map:

Refer to the above picture

2.2 Arduino Code:

```
//Dot matrix positive pin
int leds[8] = {6, 11, 5, 9, 14, 4, 15, 2};
//Dot matrix negative pin
int gnds[8] = {10, 16, 17, 7, 3, 8, 12, 13};
//Font library
char test[] = {
  0x00, 0x7C, 0x8A, 0x92, 0xA2, 0x7C, 0x00, 0x00, // -0-
  0x00, 0x00, 0x42, 0xFE, 0x02, 0x00, 0x00, 0x00, // -1-
  0x00, 0x46, 0x8A, 0x92, 0x92, 0x62, 0x00, 0x00, // -2-
  0x00, 0x84, 0x82, 0x92, 0xB2, 0xCC, 0x00, 0x00, // -3-
  0x00, 0x18, 0x28, 0x48, 0xFE, 0x08, 0x00, 0x00, // -4-
  0x00, 0xE4, 0xA2, 0xA2, 0xA2, 0x9C, 0x00, 0x00, // -5-
  0x00, 0x3C, 0x52, 0x92, 0x92, 0x8C, 0x00, 0x00, // -6-
  0x00, 0x80, 0x8E, 0x90, 0xA0, 0xC0, 0x00, 0x00, // -7-
  0x00, 0x6C, 0x92, 0x92, 0x92, 0x6C, 0x00, 0x00, // -8-
  0x00, 0x62, 0x92, 0x92, 0x94, 0x78, 0x00, 0x00, // -9-
  0x00, 0x3E, 0x48, 0x88, 0x48, 0x3E, 0x00, 0x00, // -A-
  0x00, 0xFE, 0x92, 0x92, 0x92, 0x6C, 0x00, 0x00, // -B-
  0x00, 0x7C, 0x82, 0x82, 0x82, 0x44, 0x00, 0x00, // -C-
  0x00, 0xFE, 0x82, 0x82, 0x82, 0x7C, 0x00, 0x00, // -D-
  0x00, 0xFE, 0x92, 0x92, 0x92, 0x82, 0x00, 0x00, // -E-
  0x00, 0xFE, 0x90, 0x90, 0x90, 0x80, 0x00, 0x00, // -F-
};

void setup() {
  for (int i = 0; i < 8; i++){
    pinMode(leds[i], OUTPUT);
    pinMode(gnds[i], OUTPUT);
//The negative pin is pulled high to turn off all LEDs
    digitalWrite(gnds[i], HIGH);
  }
}

void ledclean(){
//Pull the positive screen of the dot matrix low
//the negative pole is pulled high,the display is turned off
  for (int i = 0; i < 8; i++){
    digitalWrite(leds[i], LOW);
    digitalWrite(gnds[i], HIGH);
  }
}
```

```

//Character display
void ledShow(char num, char dat){
  digitalWrite(gnds[num], LOW);
  for (int i = 0; i < 8; i++){
    digitalWrite(leds[i], dat & 0x80); // 1 0 0 0 0 0 0 0
    dat <<= 1;
  }
  delayMicroseconds(100);
  digitalWrite(gnds[num], HIGH);
  ledclean();
}

void loop() {
  for(int a=0;a<120;a++){ //Control flow display content
  for(int i=0;i<200;i++){//Loop display achieve delay effect
      for (int c = 0; c < 8; c++){ //8-column loop scan
        ledShow(c, test[c + a]);
      }
    }
  }
}
```

Module 7:JoyStick Module

1. Introduction

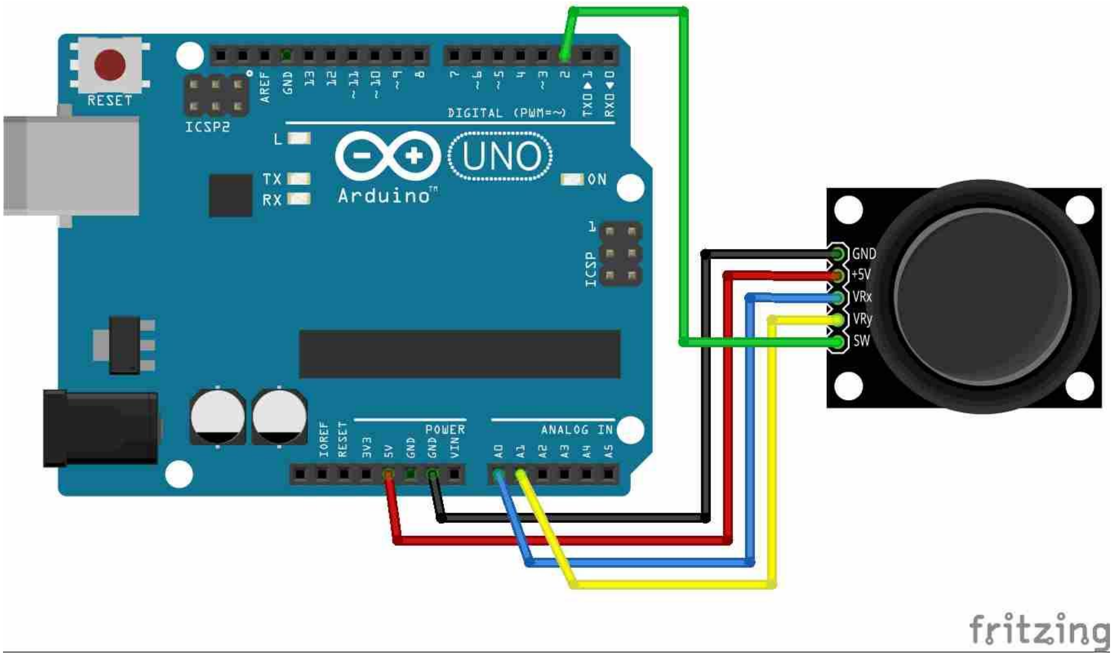
High-quality rocker, long life, stable performance.Two analog outputs,all the way to digial output

X, Y-axis output for the two potentiometers, you can read through the AD conversion twist angle

Like the next press the joystick, you can move all the way to touch the authority for the digital output, has been pulled

For two degrees of freedom servo PTZ control or other remote proportional control

2. Example(Work with Arduino)



2.1 Pin Map:

JoyStick	Arduino
GND	GND
+5V	5V
VRx	A0
VRy	A1
SW	2

2.2 Arduino Code:

```
#define joyX A0
#define joyY A1
int xValue = 0,yValue = 0;

```

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

void loop() {
  // put your main code here, to run repeatedly:
  xValue = analogRead(joyX);
  yValue = analogRead(joyY);

  //print the values with to plot or view
  Serial.print(xValue);
  Serial.print("\t");
  Serial.println(yValue);
}
```

Module 8:4-digit Digital Tube

1. Introduction

Product Name: LED Digital Display Tube; Type: Common Anode; Model: 5641BH

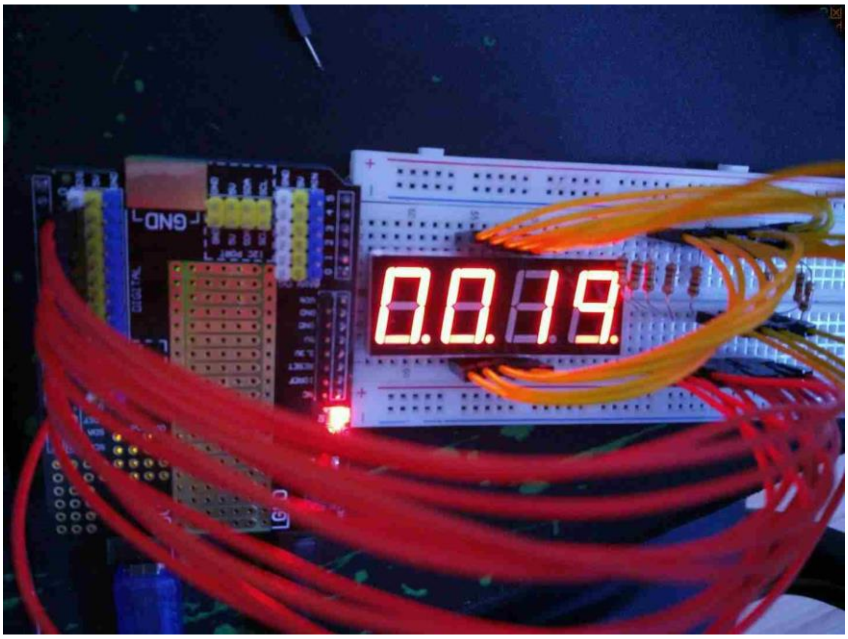
Common Anode: 12-9-8-6; Digital Display: 4 Digit; Digital Number: 4 Bit 7 Segment; Emitted Color: Red

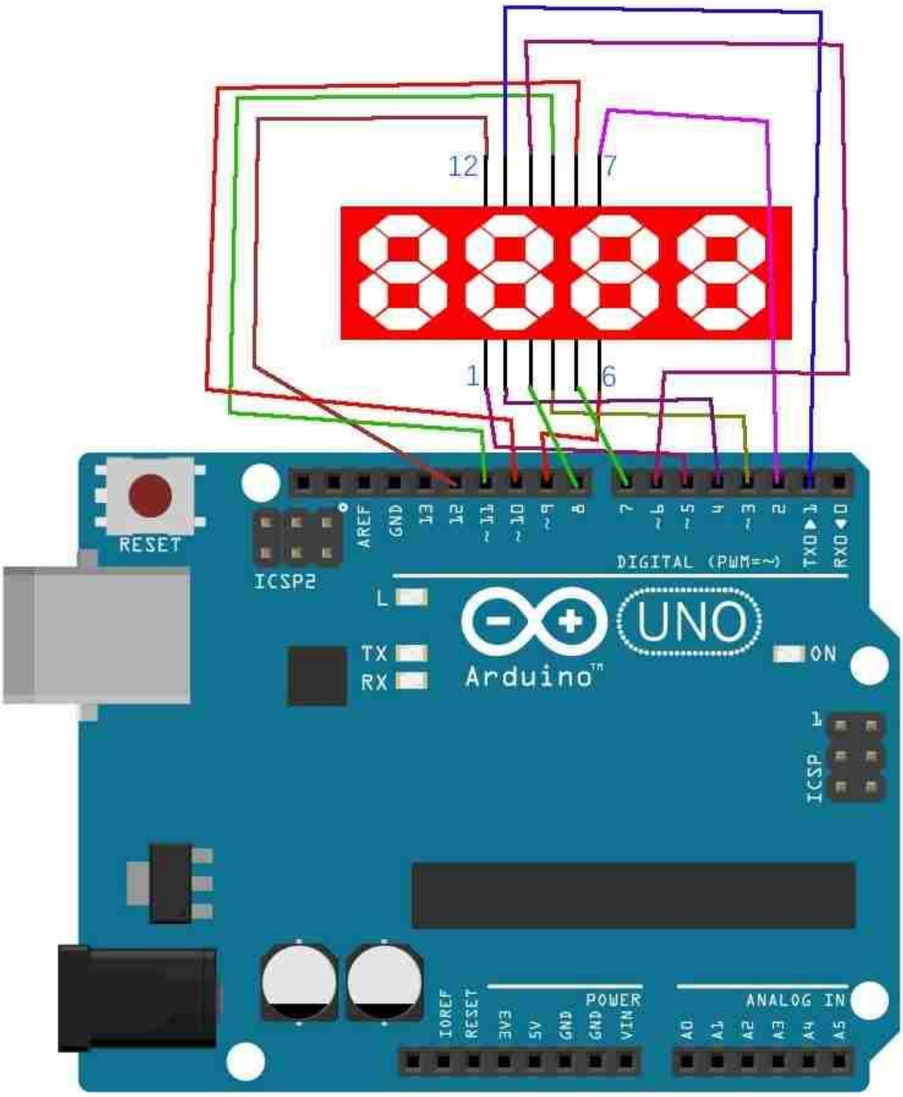
Pin Number: 12; Continuous Forward Current: 20mA; Average Forward Voltage: 2V; Power Consumption: 36mW

Luminous Intensity: 11mcd; Number Height: 0.55 inch; Panel Size: 50.3 x 19 x 8mm / 1.98" x 0.75" x 0.31" (L*W*H); Total Size: 50.3 x 19 x 14.5mm / 1.98" x 0.75" x 0.57" (L*W*H)

Material: Plastic, Metal; Color: Black, White; Package Content: 5 x LED Digital Display Tube

2. Example(Work with Arduino)




12
7
1
6
RESET
ICSP2
AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX0▶1
RX0◀0
DIGITAL (PWM=~)
L
TX
RX
Arduino™
UNO
ON
1
ICSP
IOREF
RESET
3V3
5V
GND
GND
VIN
POWER
ANALOG IN
A0
A1
A2
A3
A4
A5

2.1 Pin Map:

Digital Tube	Arduino	Digital Tube	Arduino
11	1	11	6
7	2	7	2
4	3	4	10
2	4	2	12
1	5	1	13
10	6	10	5
5	7	5	9
3	8	3	11
6	9	6	8
8	10	8	3
9	11	9	4
12	12	12	7

2.2 Arduino Code:

```
int a = 1;
int b = 2;
int c = 3;
int d = 4;
int e = 5;
int f = 6;
int g = 7;
int dp = 8;
int d1 = 9;
int d2 = 10;
int d3 = 11;
int d4 = 12;

void setup() {
    pinMode(d1, OUTPUT);
    pinMode(d2, OUTPUT);
    pinMode(d3, OUTPUT);
    pinMode(d4, OUTPUT);
    pinMode(a, OUTPUT);
    pinMode(b, OUTPUT);
    pinMode(c, OUTPUT);
    pinMode(d, OUTPUT);
    pinMode(e, OUTPUT);
    pinMode(f, OUTPUT);
    pinMode(g, OUTPUT);
    pinMode(dp, OUTPUT);
}

void loop() {
  showNumber(1,9);
  showNumber(2,1);
  showNumber(3,0);
  showNumber(4,0);
}

void showNumber(unsigned int pos,unsigned char num){
  weizhi(pos);
  getNumber(num);
  delay(1);
  digitalWrite(a,HIGH);
  digitalWrite(b,HIGH);
  digitalWrite(c,HIGH);
  digitalWrite(d,HIGH);
  digitalWrite(e,HIGH);
  digitalWrite(f,HIGH);
  digitalWrite(g,HIGH);
  digitalWrite(dp,HIGH);
}

void weizhi(unsigned int pos){
  switch(pos){
    case 1:
      digitalWrite(d1,HIGH);
      digitalWrite(d2,LOW);
      digitalWrite(d3,LOW);
      digitalWrite(d4,LOW);
      break;
    case 2:
      digitalWrite(d1,LOW);
      digitalWrite(d2,HIGH);
      digitalWrite(d3,LOW);
      digitalWrite(d4,LOW);
      break;
    case 3:
      digitalWrite(d1,LOW);
      digitalWrite(d2,LOW);
      digitalWrite(d3,HIGH);
      digitalWrite(d4,LOW);
      break;
    case 4:
      digitalWrite(d1,LOW);
      digitalWrite(d2,LOW);
      digitalWrite(d3,LOW);
      digitalWrite(d4,HIGH);
      break;
    default:
      digitalWrite(d1,LOW);
      digitalWrite(d2,LOW);
      digitalWrite(d3,LOW);
      digitalWrite(d4,HIGH);
  }
}

void getNumber(unsigned char num){
  switch(num){
    case 0:
      digitalWrite(a,LOW);
      digitalWrite(b,LOW);
      digitalWrite(c,LOW);
      digitalWrite(d,LOW);
```

```
          digitalWrite(e,LOW);
          digitalWrite(f,LOW);
          digitalWrite(g,HIGH);
          digitalWrite(dp,HIGH);
          break;
        case 1:
          digitalWrite(a,HIGH);
          digitalWrite(b,LOW);
          digitalWrite(c,LOW);
          digitalWrite(d,HIGH);
          digitalWrite(e,HIGH);
          digitalWrite(f,HIGH);
          digitalWrite(g,HIGH);
          digitalWrite(dp,HIGH);
          break;
        case 2:
          digitalWrite(a,LOW);
          digitalWrite(b,LOW);
          digitalWrite(c,HIGH);
          digitalWrite(d,HIGH);
          digitalWrite(e,LOW);
          digitalWrite(f,HIGH);
          digitalWrite(g,LOW);
          digitalWrite(dp,HIGH);
          break;
        case 3:
          digitalWrite(a,LOW);
```

```
      digitalWrite(b,LOW);
      digitalWrite(c,LOW);
      digitalWrite(d,LOW);
      digitalWrite(e,HIGH);
      digitalWrite(f,HIGH);
      digitalWrite(g,LOW);
      digitalWrite(dp,HIGH);
      break;
    case 4:
      digitalWrite(a,HIGH);
      digitalWrite(b,LOW);
      digitalWrite(c,LOW);
      digitalWrite(d,HIGH);
      digitalWrite(e,HIGH);
      digitalWrite(f,LOW);
      digitalWrite(g,LOW);
      digitalWrite(dp,HIGH);
      break;
    case 5:
      digitalWrite(a,LOW);
      digitalWrite(b,HIGH);
      digitalWrite(c,LOW);
      digitalWrite(d,LOW);
      digitalWrite(e,HIGH);
      digitalWrite(f,LOW);
      digitalWrite(g,LOW);
      digitalWrite(dp,HIGH);
```

```
          break;
        case 6:
          digitalWrite(a,LOW);
          digitalWrite(b,HIGH);
          digitalWrite(c,LOW);
          digitalWrite(d,LOW);
          digitalWrite(e,LOW);
          digitalWrite(f,LOW);
          digitalWrite(g,LOW);
          digitalWrite(dp,HIGH);
          break;
        case 7:
          digitalWrite(a,LOW);
          digitalWrite(b,LOW);
          digitalWrite(c,LOW);
          digitalWrite(d,HIGH);
          digitalWrite(e,HIGH);
          digitalWrite(f,HIGH);
          digitalWrite(g,HIGH);
          digitalWrite(dp,HIGH);
          break;
        case 8:
          digitalWrite(a,LOW);
          digitalWrite(b,LOW);
          digitalWrite(c,LOW);
          digitalWrite(d,LOW);
          digitalWrite(e,LOW);
```

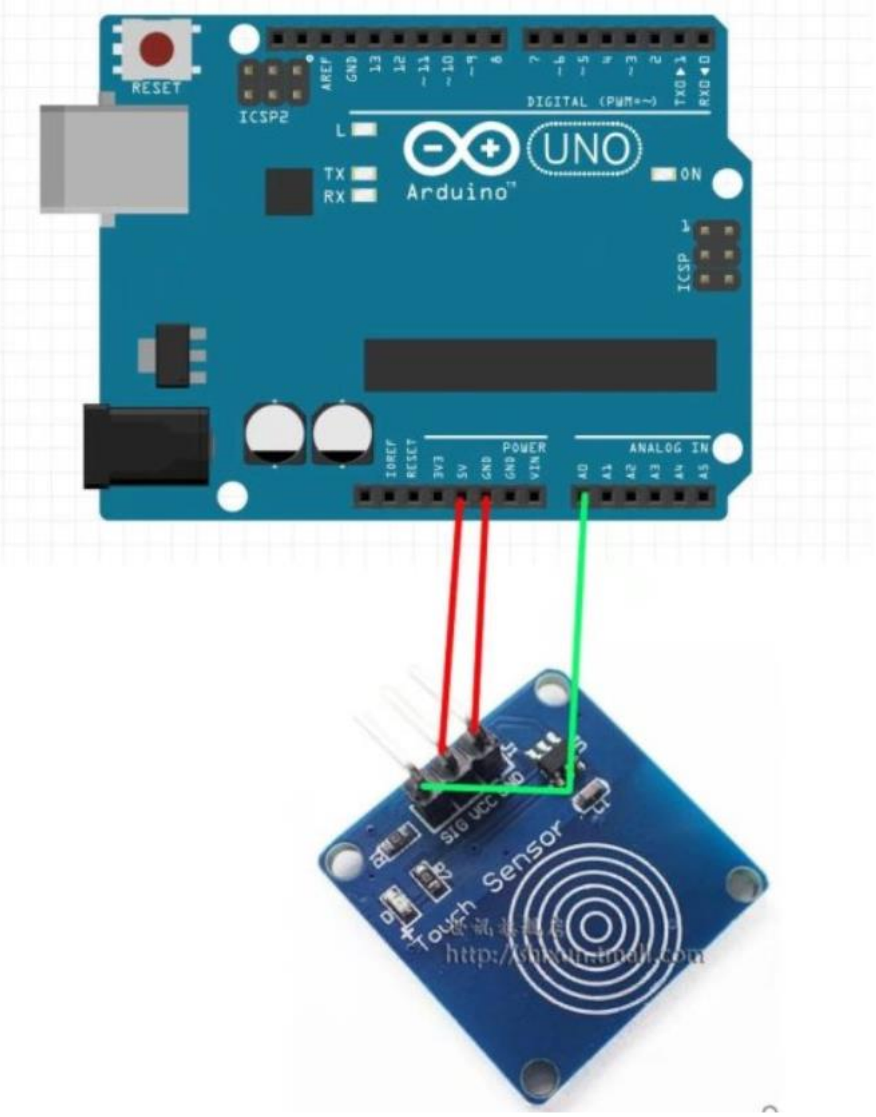
```
      digitalWrite(f,LOW);
      digitalWrite(g,LOW);
      digitalWrite(dp,HIGH);
      break;
    case 9:
      digitalWrite(a,LOW);
      digitalWrite(b,LOW);
      digitalWrite(c,LOW);
      digitalWrite(d,LOW);
      digitalWrite(e,HIGH);
      digitalWrite(f,LOW);
      digitalWrite(g,LOW);
      digitalWrite(dp,HIGH);
      break;
    default:
      digitalWrite(a,HIGH);
      digitalWrite(b,HIGH);
      digitalWrite(c,HIGH);
      digitalWrite(d,HIGH);
      digitalWrite(e,HIGH);
      digitalWrite(f,HIGH);
      digitalWrite(g,HIGH);
      digitalWrite(dp,HIGH);
  }
}
```

Module 9:TTP223B Touch sensor

1. Introduction

TTP233B touch sensor

2 . Example(Work with Arduino



2.1 Pin Map:

TTP223B	Arduino
VCC	5V
GND	GND
SIG	A0

2.2 Arduino Code:

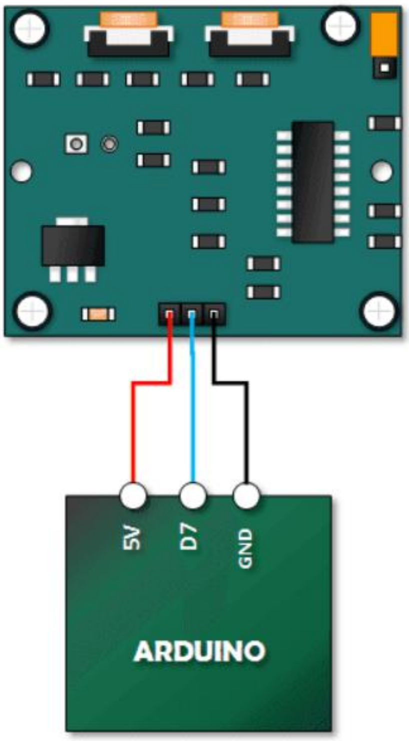
```
//When the sensor is touched, LED on the Arduino board lights up; otherwise, the LED goes out
void setup(){
//Connect sensor's to A0 of Arduino to receive the touch signal
 pinMode(A0,INPUT);
 pinMode(13, OUTPUT); //3 port set output mode
}

void loop(){
//Judge if the returned data is greater than 0, port 13 high voltage, light LED bulb
 if(analogRead(A0)>0){
   digitalWrite(13,HIGH);
  }else{ //low voltage, turn off the light bulb
   digitalWrite(13,LOW);
  }
  delay(100);
}
```

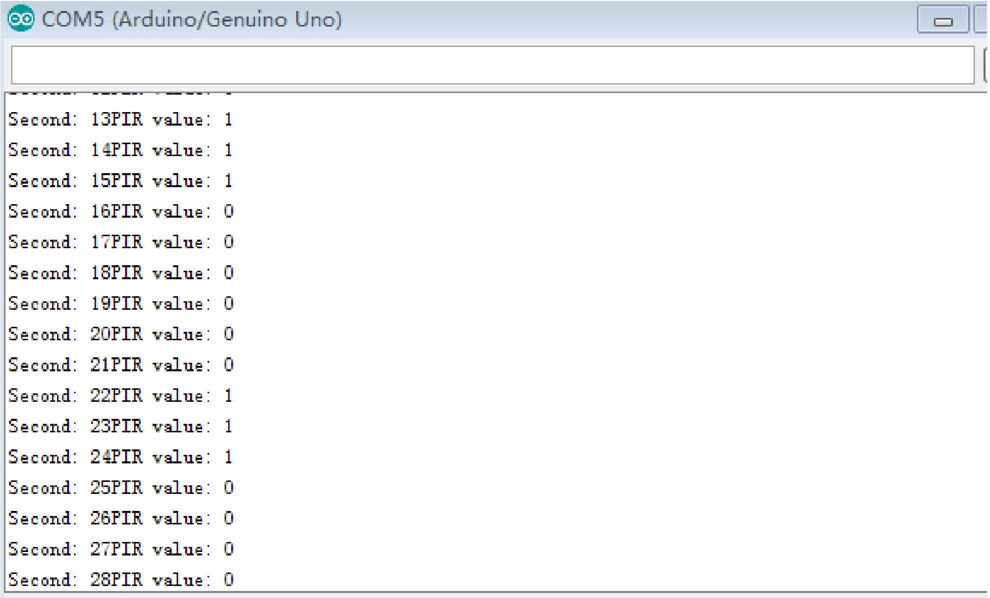
Module 10:HC-SR501 Infrared sensor

1. Introduction

The HC-SR501 is a pyroelectric motion sensor based on the pyroelectric effect that detects infrared rays emitted from humans or animals. This sensor module can be adjusted by two knobs to detect the range of 3 ~ 7 meters, 5 seconds to 5 minutes of delay time, and jumper to select single trigger and repeat trigger mode.

2. Example(Work with Arduino)





```
Second: 13PIR value: 1
Second: 14PIR value: 1
Second: 15PIR value: 1
Second: 16PIR value: 0
Second: 17PIR value: 0
Second: 18PIR value: 0
Second: 19PIR value: 0
Second: 20PIR value: 0
Second: 21PIR value: 0
Second: 22PIR value: 1
Second: 23PIR value: 1
Second: 24PIR value: 1
Second: 25PIR value: 0
Second: 26PIR value: 0
Second: 27PIR value: 0
Second: 28PIR value: 0
```

2.1 Pin Map:

HC-SR501	Arduino
VCC	5V

GND	GND
OUT	D7

2.2 Arduino Code:

```
//Observe the sensor output status
int ledPin = 13;
int pirPin = 7;
int pirValue;
int sec = 0;
void setup(){
    pinMode(ledPin, OUTPUT);
    pinMode(pirPin, INPUT);
    digitalWrite(ledPin, LOW);
    Serial.begin(9600);
}

void loop(){
    pirValue = digitalRead(pirPin);
    digitalWrite(ledPin, pirValue);
    //Observe the sensor output status
     sec += 1;
     Serial.print("Second: ");
     Serial.print(sec);
     Serial.print("PIR value: ");
     Serial.print(pirValue);
     Serial.print('\n');
     delay(1000);
}
```

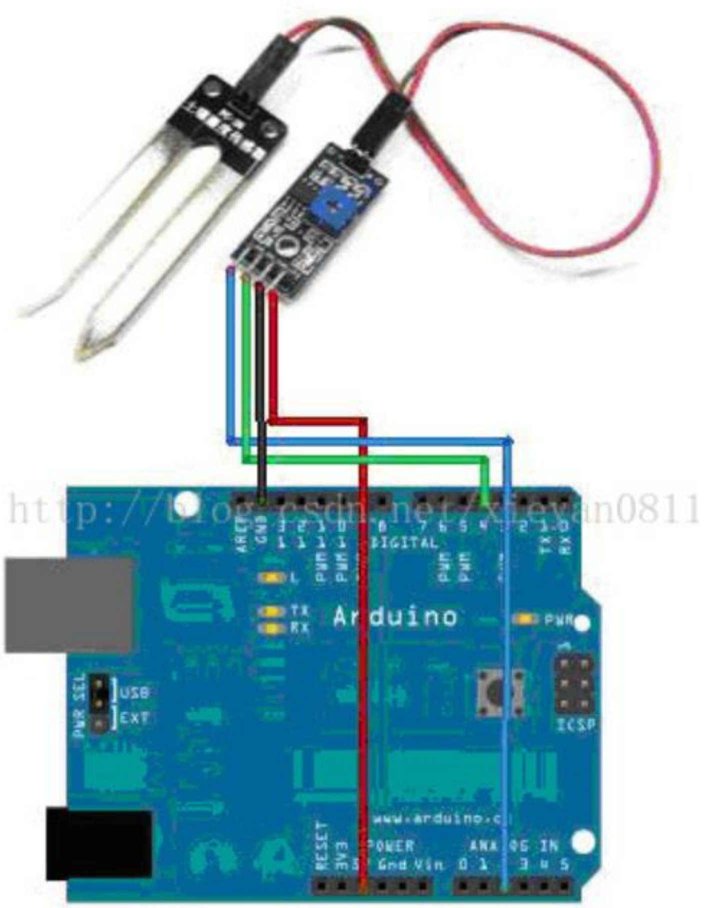
Module 11:Soil Moisture Sensor

1. Introduction

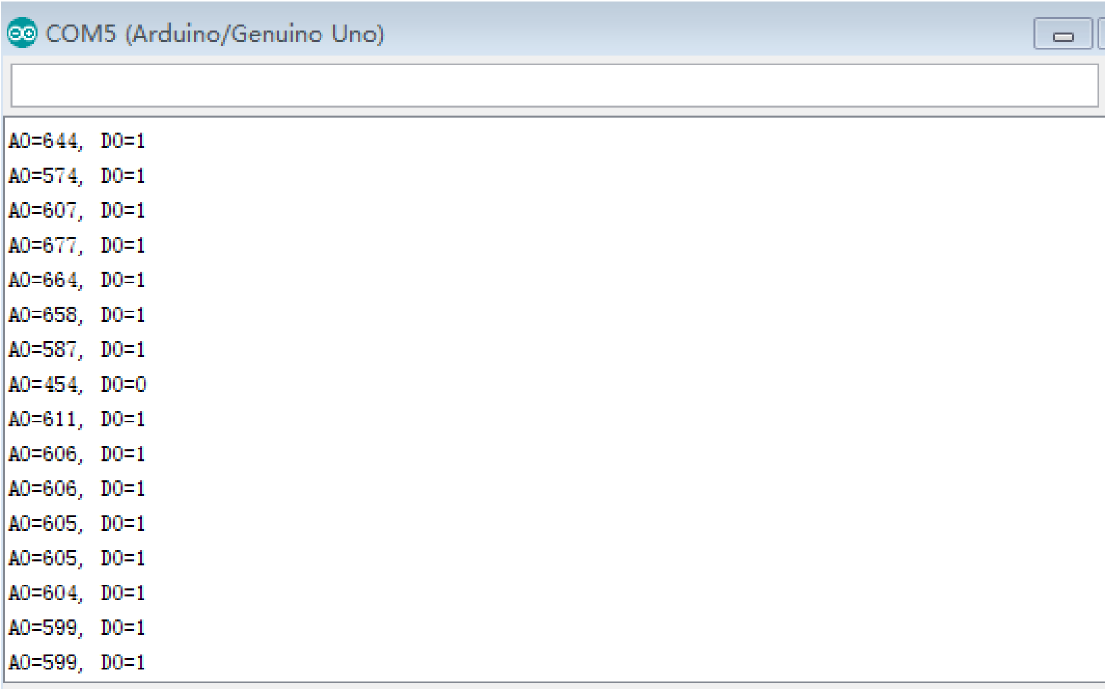
Used for moisture detection in soil.The threshold of soil moisture can be adjusted by potentiometer, clockwise adjustment, the humidity of control will be larger, the counterclockwise will be smaller; when the humidity is lower than the set value, DO will output high level, the module prompts light; humidity is higher than the setting When the value is low, the DO output is low and the module prompts off.

The working voltage is 3.3V-5V. At 3V, the maximum value of AO reading in air is 695, and the minimum value of immersion in water is 245; at 5V, the maximum value of AO reading in air is 1023, and the minimum value of immersion in water is 245.

2. Example(Work with Arduino)



http://blog.csdn.net/xieyan0811



```
A0=644, D0=1
A0=574, D0=1
A0=607, D0=1
A0=677, D0=1
A0=664, D0=1
A0=658, D0=1
A0=587, D0=1
A0=454, D0=0
A0=611, D0=1
A0=606, D0=1
A0=606, D0=1
A0=605, D0=1
A0=605, D0=1
A0=604, D0=1
A0=599, D0=1
A0=599, D0=1
```

2.1 Pin Map:

Soil Moisture Sensor	Arduino
VCC	5V
GND	GND
AO	A2
DO	D4

2.2 Arduino Code:

```
#define PIN_AO 2
#define PIN_DO 4
void setup(){
  pinMode(PIN_AO, INPUT);
  pinMode(PIN_DO, INPUT);
  Serial.begin(9600);
}

void loop(){
  Serial.print("AO=");
  Serial.print(analogRead(PIN_AO));
  Serial.print(", DO=");
  Serial.println(digitalRead(PIN_DO));
  delay(500);
}
```

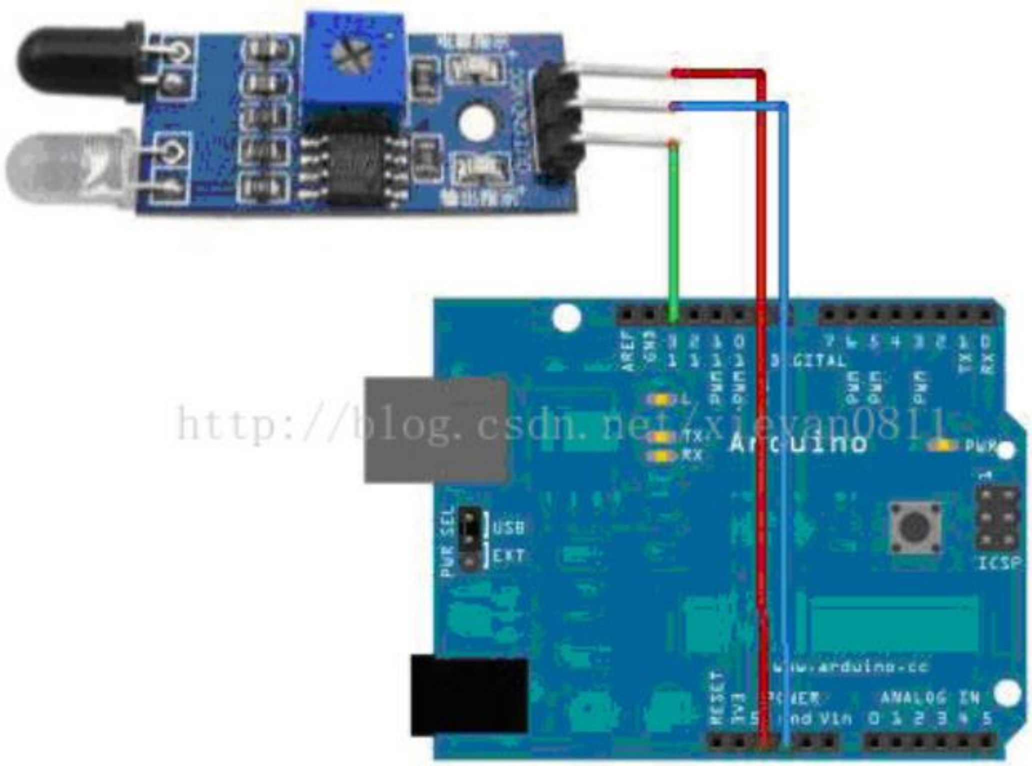
Module 12:Obstacle Avoidance Sensor

1. Introduction

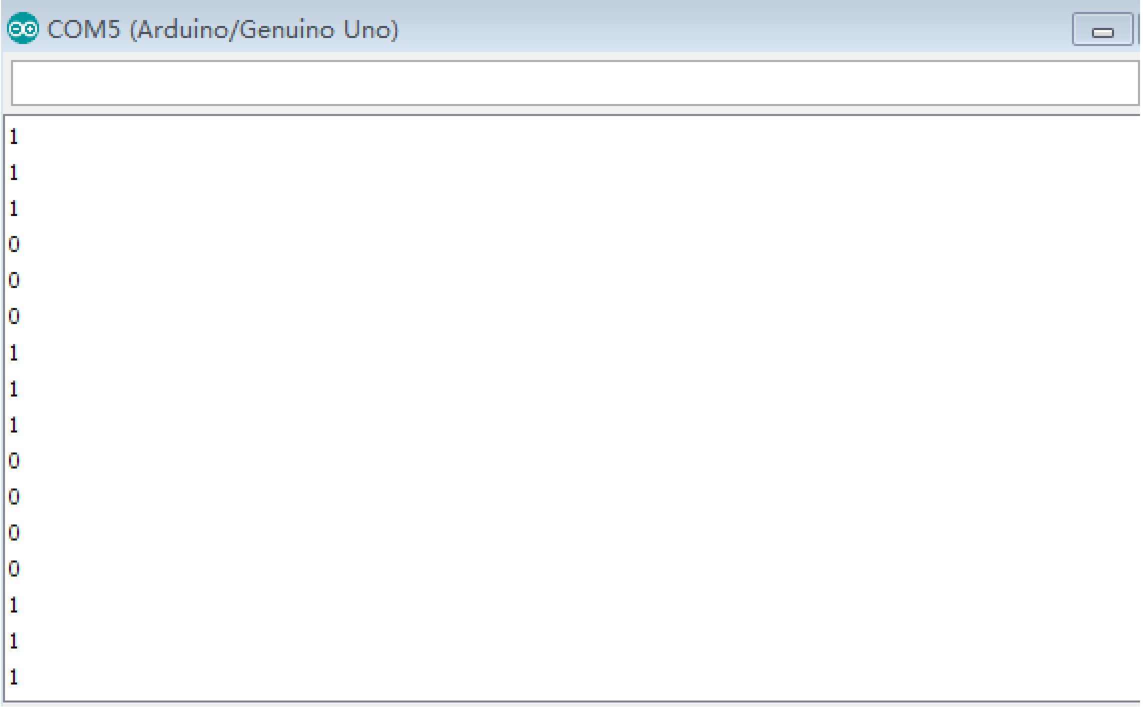
The infrared obstacle avoidance sensor has a pair of infrared emitting and receiving tubes, and the transmitting tube emits infrared rays of a certain frequency. When the detecting direction encounters an obstacle (reflecting surface), the infrared rays are reflected and received by the receiving tube. It is often used to mount on a cart to determine if there are obstacles in front. The threshold can be set via a potentiometer. When there is an obstacle in front, the green light is on, the OUT pin is low, and vice versa.

Since daylight also contains infrared light, most inexpensive infrared modules can be problematic when used outdoors.

2. Example(Work with Arduino)



http://blog.csdn.net/xieyan0811



```
1
1
1
0
0
0
1
1
1
0
0
0
0
1
1
1
```

2.1 Pin Map:

KY-018	Arduino
VCC	5V
GND	GND
OUT	D13

2.2 Arduino Code:

```
int PIN_SENSOR = 13;
void setup() {
  pinMode(PIN_SENSOR, INPUT);
  Serial.begin(9600);
}
void loop() {
  int x = digitalRead(PIN_SENSOR);
  Serial.println(x);
}
```

Module 13:Bread Board

1. Introduction

Bread Board

2. Example(Work with Arduino)

2.1 Pin Map:

… …

2.2 Arduino Code:

… …

Module 14:SG90 9G Servo Motor

1. Introduction

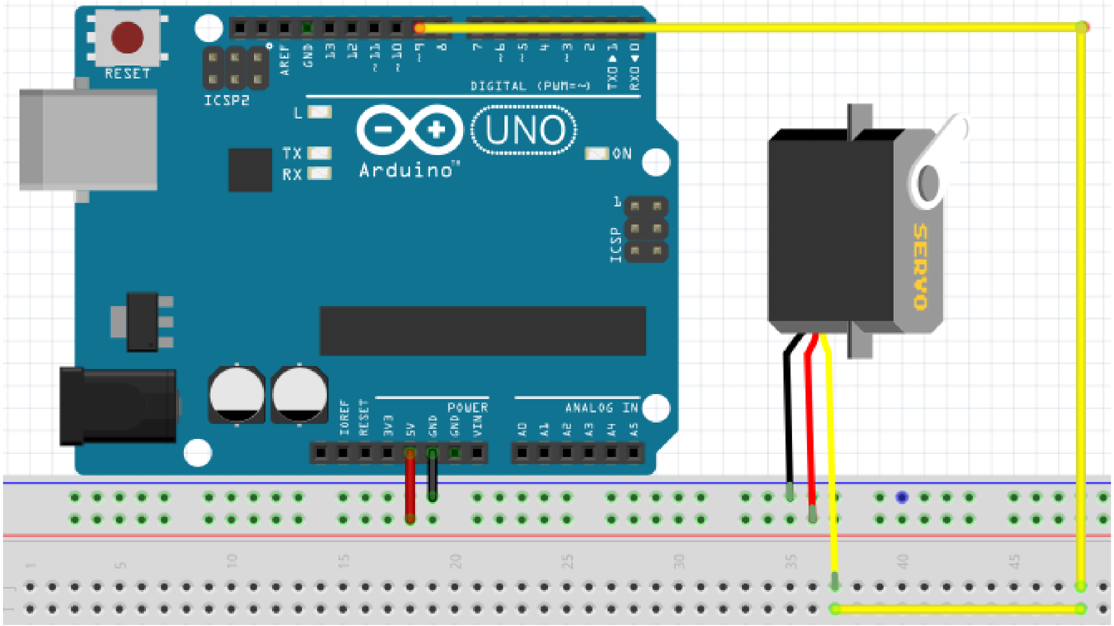
Application Fields: remote control helicopters, micro robot, robot arm and boats. ALL kind of R/C Toys and Arduino experiments

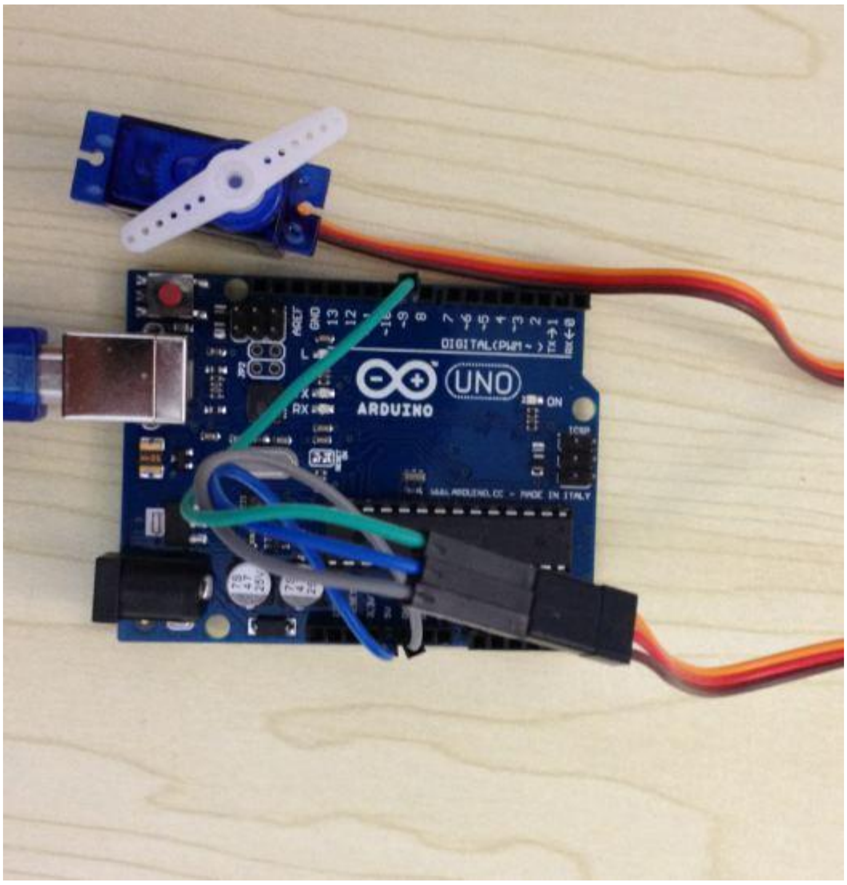
Interface Type: compatible with JR & Futaba interface. No Load Running Speed: 0.09±0.01 sec/60° at 4.8V Rotary Angle: 120°

Stall Torque (4.8V): 17.5oz /in (1kg/cm) Dead band width: 7usec

Amplifier type:Analog controller Operating voltage: 4.8V～6.0V

2. Example(Work with Arduino)





2.1 Pin Map:

SG90 Servo	Arduino
VCC(red)	5V
GND(brown)	GND
PWM(orange)	10

2.2 Arduino Code:

```
#include <Servo.h>
//Define steering gear objects, up to eight
Servo myservo;
//Define the steering position of the steering gear
int pos = 0;
void setup(){
  myservo.attach(9); //Set the servo control pin
}

 void loop(){
 //Rotating servo from 0 to 180 with a delay of 15 milliseconds
   for(pos = 0; pos < 180; pos += 1){
     myservo.write(pos);
     delay(15);
   }
 //Rotating servo from 180 to 0 with a delay of 15 milliseconds each time
   for(pos = 180; pos>=1; pos-=1){
     myservo.write(pos);
     delay(15);
   }
 }
```