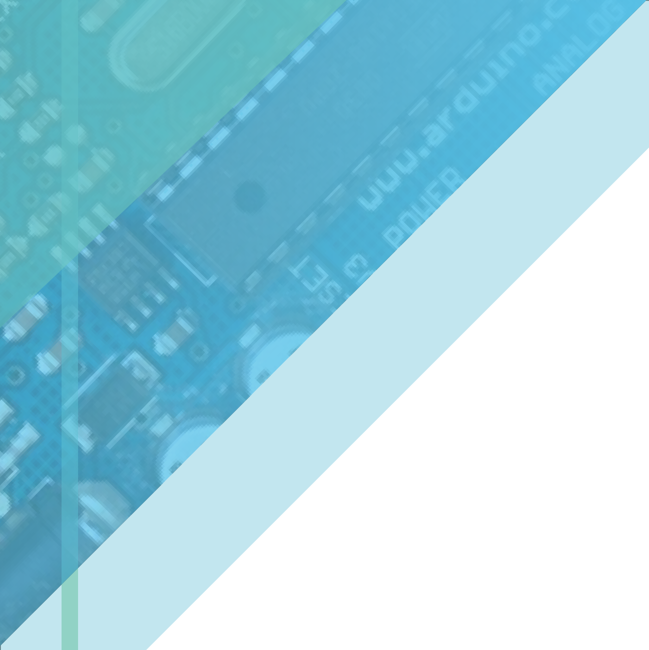


BAIJIA DAGU

Arduino

Starter kit

Catalog

LED module

（Red/Yellow/Green/Blue/White）

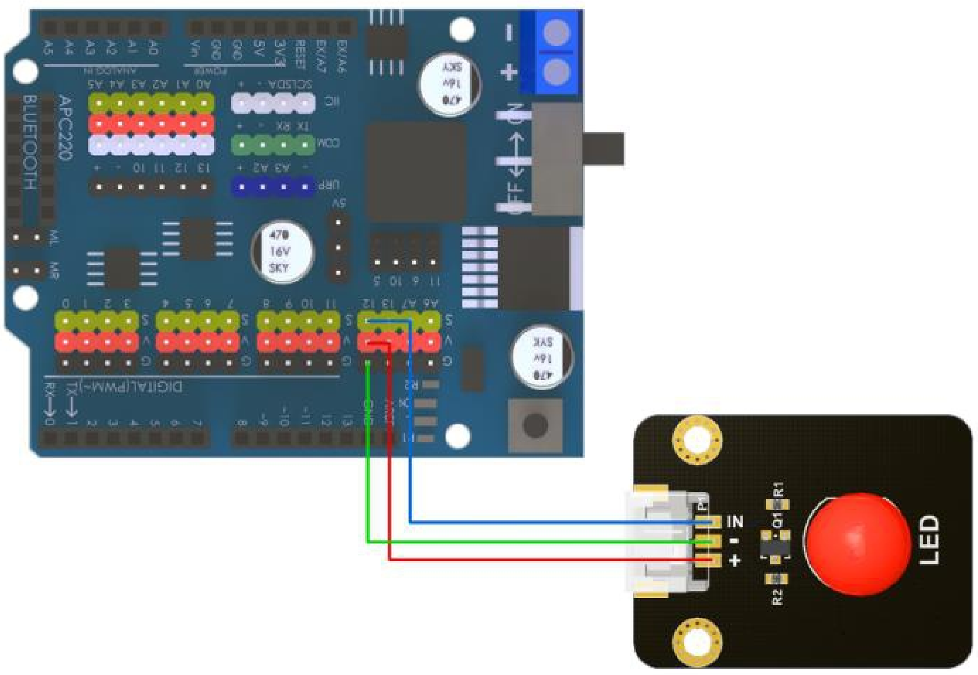
Module description: LED is the abbreviation of light-emitting diode. It is usually made of gallium arsenide, gallium phosphide semiconductor material. LED has two electrodes, one positive and one negative. When the positive current passes through, it will light up. It can be red, blue, green or yellow light, and the color of the light depends on the material it uses.

Working voltage:3.3V~5V
Output type:Digital signal. (For analog voltages from 0V to 5V, values (0-255) are also allowed to be entered as digital values.)
Working mode:4 type
Interface mode:PH2.0~3P
Module size:35*26.3mm
Module weight:47g

Pin denifition:

IN	Input
+	VCC
–	GND

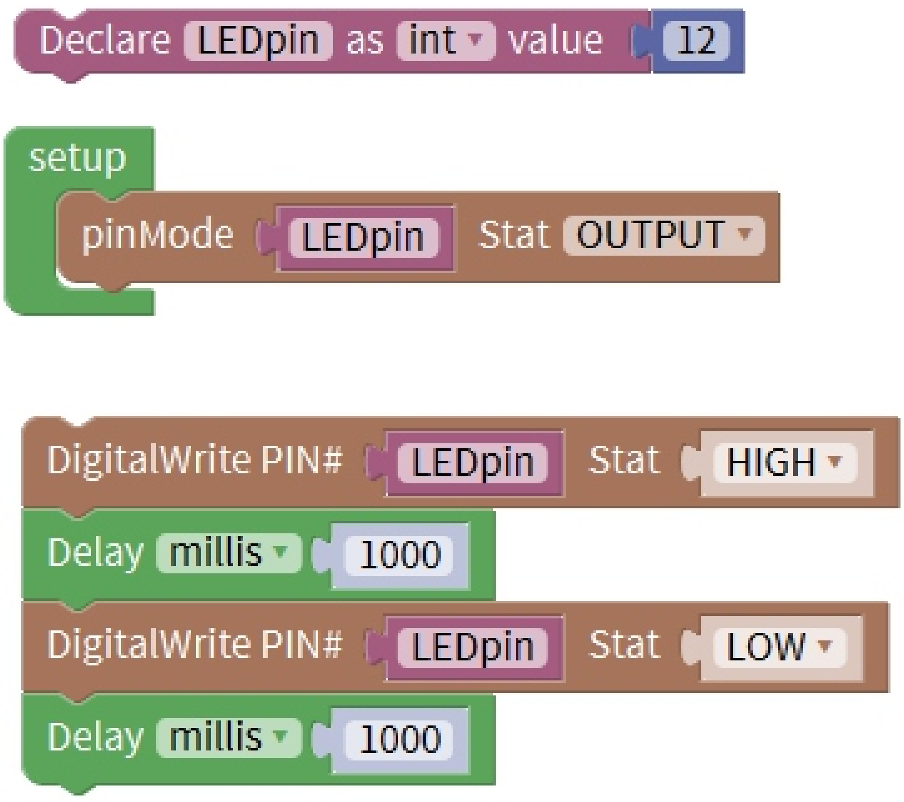
Arduino uno
Example 1
Connection diagram



Code

```
int LEDpin = 12;        //  Pin  12  of  LED

void setup() {
   pinMode(LEDpin, OUTPUT);    // Define LED pin as output pin
}

void loop() {
   digitalWrite(LEDpin, HIGH);      // Write output value is high
   delay(1000);
   digitalWrite(LEDpin, LOW);       // Write output value is low
   delay(1000);
}
```

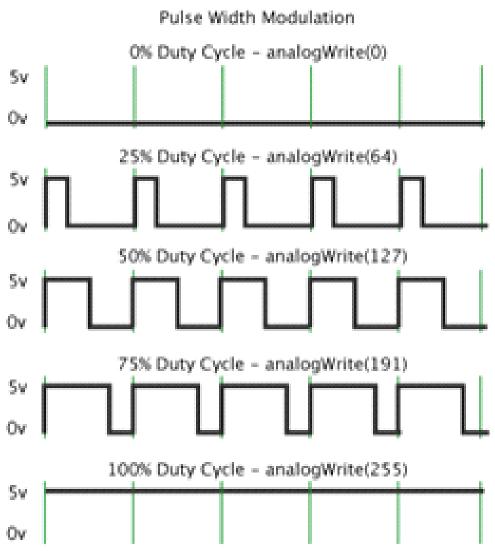
Mixly graphic programming



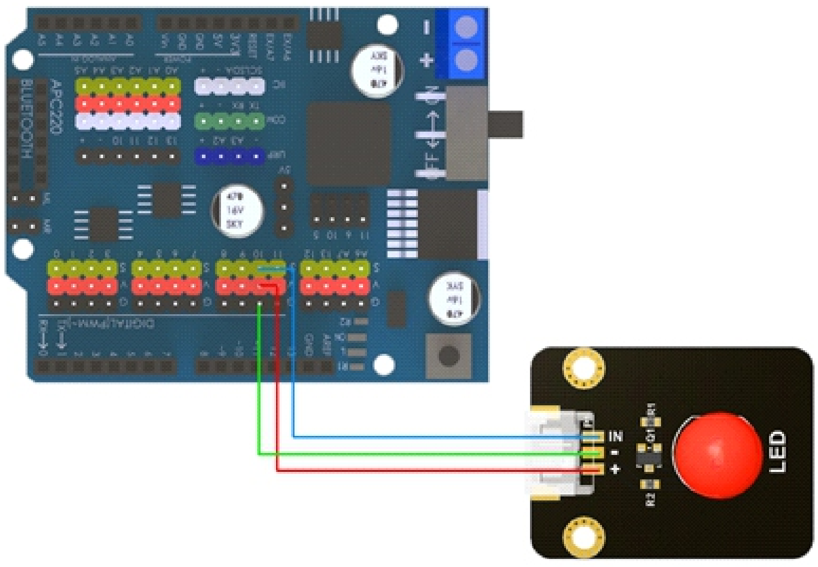
```
Declare LEDpin as int value 12

setup
   pinMode LEDpin Stat OUTPUT

DigitalWrite PIN# LEDpin Stat HIGH
Delay millis 1000
DigitalWrite PIN# LEDpin Stat LOW
Delay millis 1000
```

Result: after pin 12 is plugged in the LED light, the LED light will turn on for one second and off for one second.

Pulse Width Modulation or PWM is a king of technical and digital means to obtain simulation results. Digital control is used to generate a square wave, and a signal exchange is between on and off. These switch modes simulate voltage on 5V and off 0V between voltages. By changing the time part of the signal, the duration is compared with the signal stop time. The duration of the voltage on is called the pulse width. To get different analog values, you can change the pulse width. For example, led, the result is like a signal between 0V and 5V stable voltage, to control the brightness of LED. In the following figure, the green line represents a fixed time period. In other words, the pulse width modulation frequency with Arduino is about 50Hz, and the green line measurement is 20ms each time. Call the analog Write (pin, value), where the value is 0~255, so that the analog Write(pin, 255) requests 100% duty cycle (always open). For example, analog Write(pin, 127) is a 50% duty cycle (half time).



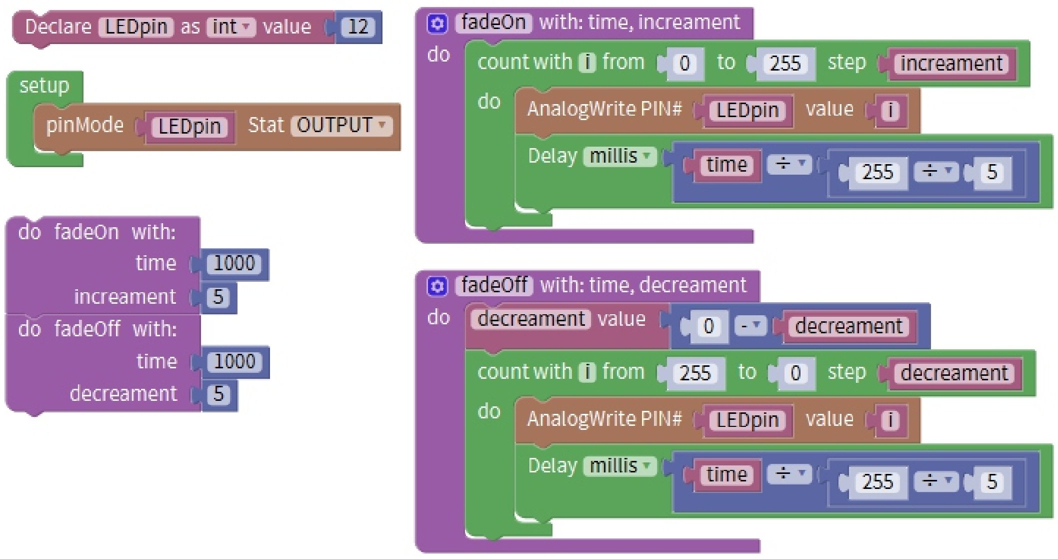
Example 2
Connection diagram



Code

```
int ledPin = 10;        //   Pin   10   ofLED

void setup() {
  pinMode(ledPin,OUTPUT);    // DefineLEDpinasoutputpin
}
void loop(){
fadeOn(1000,5);
fadeOff(1000,5);
}
// Define fuction, the analog output value of LED is increased one by one according to the
custom increament
void fadeOn(unsigned int time,int increament){
  for (byte value = 0 ; value < 255; value+=increament)
  { analogWrite(ledPin, value);
  delay(time/(255/5));
  }
}
// Define fuction, the analog output value of LED is decreased one by one according to the
custom increament
void fadeOff(unsigned int time,int decreament){
  for (byte value = 255; value >0; value-=decreament)
  { analogWrite(ledPin, value);
  delay(time/(255/5));
  }
}
```

Mixly graphic programming



```
Declare LEDpin as int value 12

setup
  pinMode LEDpin Stat OUTPUT

do fadeOn with:
  time 1000
  increament 5
do fadeOff with:
  time 1000
  decreament 5

fadeOn with: time, increament
do
  count with i from 0 to 255 step increament
  do
    AnalogWrite PIN# LEDpin value i
    Delay millis time ÷ 255 ÷ 5

fadeOff with: time, decreament
do
  decreament value 0 - decreament
  count with i from 255 to 0 step decreament
  do
    AnalogWrite PIN# LEDpin value i
    Delay millis time ÷ 255 ÷ 5
```

Result: the LED on pin 10 has a slow process from on to off, rather than directly on and off.。

RGB module

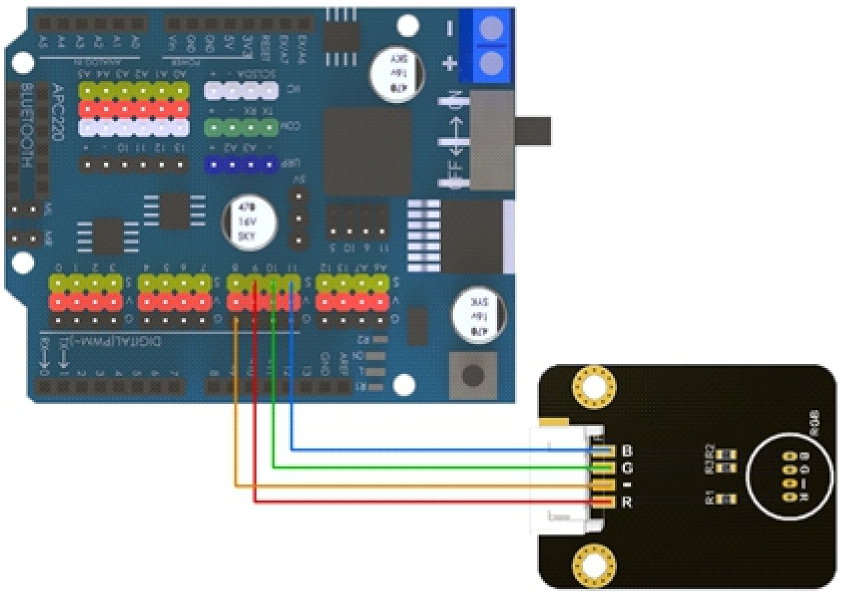
Product description: RGB LED consists of 3 leds. Each led has a red light, a green light and a blue light. These three colors leds can produce any color. The RGB LED has red, green and blue light transmitters and is usually connected to a common lead (anode or carthode) using three wires. The module is a commong cathode led.

Working voltage: 3V~5V
Input type: PWM. The value (0~255) is input into RDDB three interfaces as digitial value. The color of RGB LED is controled by PWM.
Interface mode: pH2.0~4P
Module size: 35*26.3mm
Module weight: 49g

Pin denifition:

B	Blue intput
G	Green input
-	GND
R	Red input

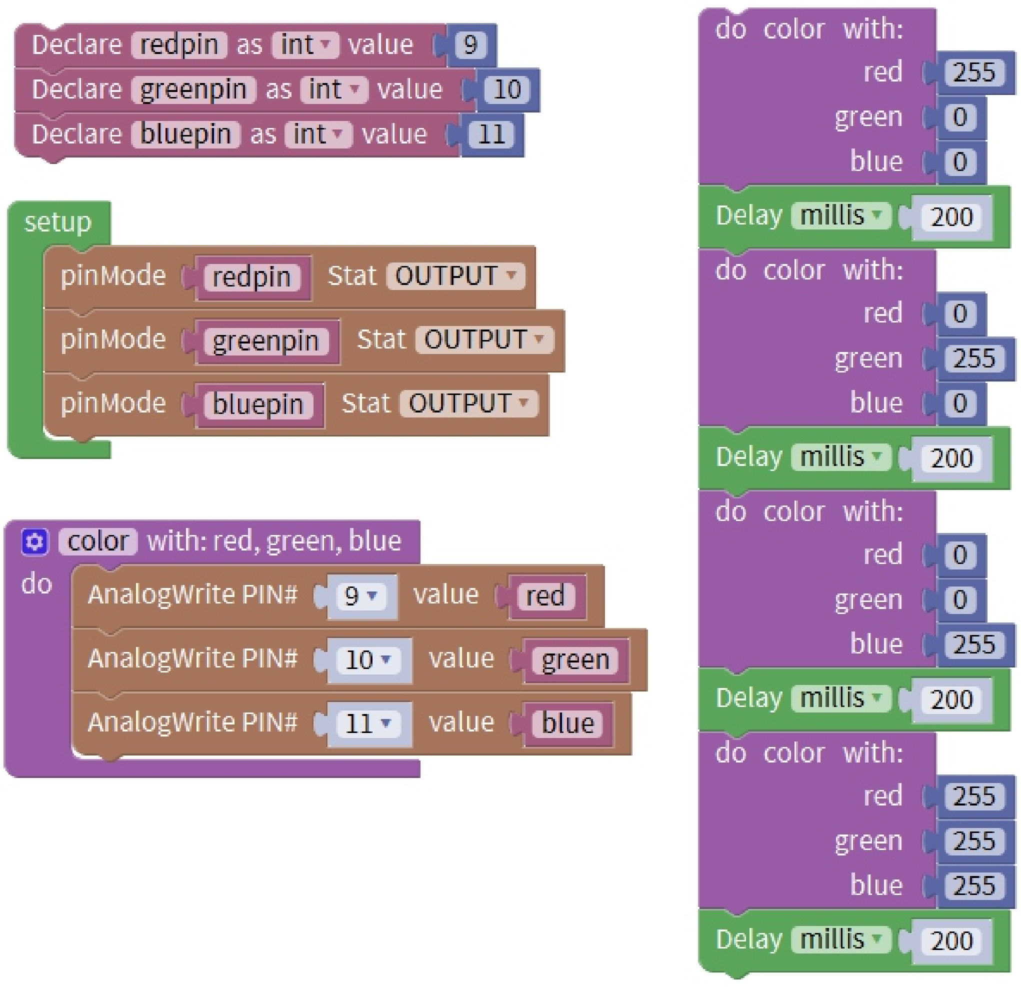
Arduino uno
Example 1
Connection diagram



Code

```
int redpin = 9;              // Pin 9 of RGB Red LED
int greenpin  = 10;         // Pin 10 of RGB Green LED
int bluepin = 11;           // Pin 11 of RGB Blue LED
void setup() {
   pinMode(redpin,OUTPUT);                // Define three color of LED pin as output pin
   pinMode(greenpin,OUTPUT);
   pinMode(bluepin,OUTPUT);
}

void loop() {
   color(255,0,0);
   delay(200);
   color(0,255,0);
   delay(200);
   color(0,0,255);
   delay(200);
   color(255,255,255);
   delay(200);
}

// Define fuction, define write output values separately
void color(int red,int green ,int blue){
   analogWrite(redpin,red);
   analogWrite(greenpin,green);
   analogWrite(bluepin,blue);
}
```

Mixly graphic programming



```
Declare redpin as int value 9
Declare greenpin as int value 10
Declare bluepin as int value 11

setup
    pinMode redpin Stat OUTPUT
    pinMode greenpin Stat OUTPUT
    pinMode bluepin Stat OUTPUT

color with: red, green, blue
do
    AnalogWrite PIN# 9 value red
    AnalogWrite PIN# 10 value green
    AnalogWrite PIN# 11 value blue

do color with:
        red 255
        green 0
        blue 0
Delay millis 200
do color with:
        red 0
        green 255
        blue 0
Delay millis 200
do color with:
        red 0
        green 0
        blue 255
Delay millis 200
do color with:
        red 255
        green 255
        blue 255
Delay millis 200
```

Result: Pin R of module connect to Pin 9, G connect to Pin 10, B connect to Pin 11. The RGB led blinks in cycles of red, green, blue and white.

Button module

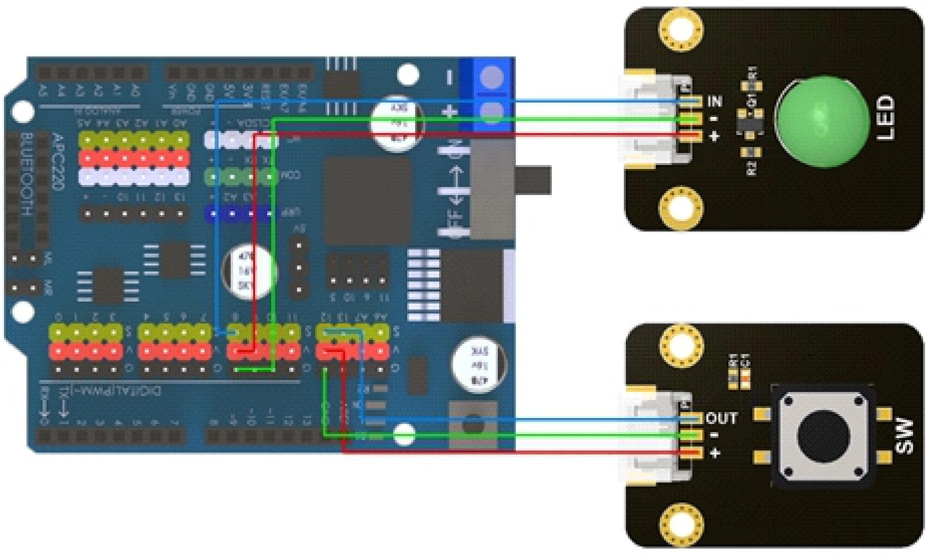
Product description: Buttons are common components used to control electronic devices. They are usually used as switches to connect or disconnect circuits. Under normal circumstances, the two contacts of the button are in the open state, and they only close when the button is pressed.

Working voltage:3.3V to 5V
Output type:Digital signal. Press the key, high level;Release the key, low level.
Interface mode:PH2.0~3P
Module size:35*26.3mm
Module weight:49g. 3mm

Pin definition:

OUT	Output
+	VCC
-	GND

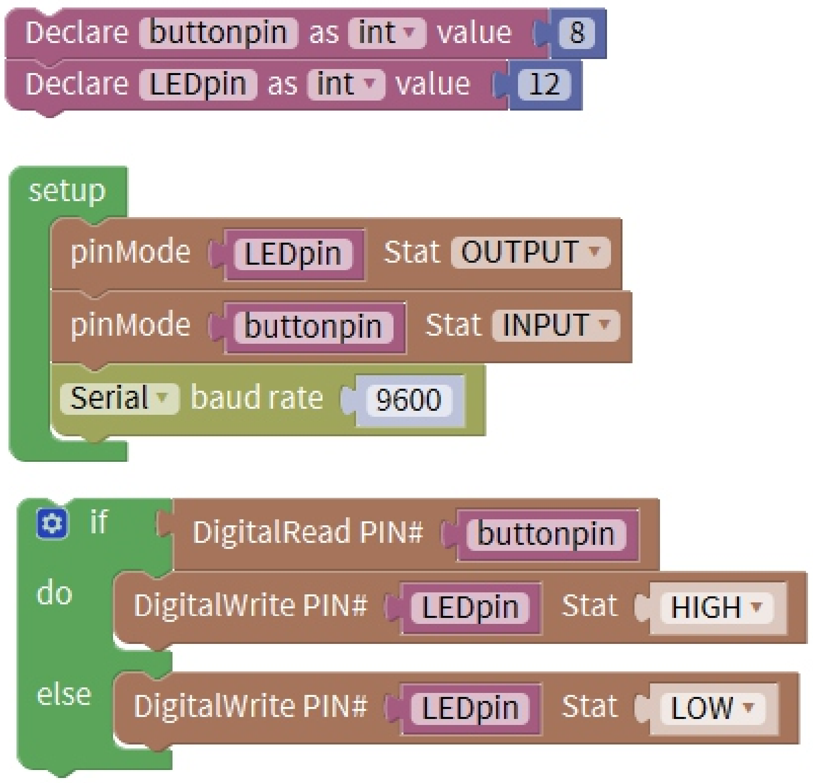
Arduino uno
Example 1
Connection diagram



Code

```
int buttonpin = 8;         // Pin    8 button module

int LEDpin = 12;           // Pin 12 of LED

void setup() {
    pinMode(LEDpin, OUTPUT);          // Define LED pin as output pin
    pinMode(buttonpin,INPUT);          // Define button pin as input pin
}

void loop() {
   int state = digitalRead(buttonpin);          // Read output value
   if(state == HIGH ){             //Check whether the input is high. Press button is high.
      digitalWrite(LEDpin,  HI G H );             //      Light      on      status
   }
   else{
      digitalWrite(LEDpin, LOW);              // Light off status
   }
}
```

Mixly graphic programming



```
Declare buttonpin as int value 8
Declare LEDpin as int value 12

setup
    pinMode LEDpin Stat OUTPUT
    pinMode buttonpin Stat INPUT
    Serial baud rate 9600

if DigitalRead PIN# buttonpin
do   DigitalWrite PIN# LEDpin Stat HIGH
else DigitalWrite PIN# LEDpin Stat LOW
```

Result: when you press the button, the LED of Pin 13 will be on. When you release the button, the LED will be off.

Photoresistor module

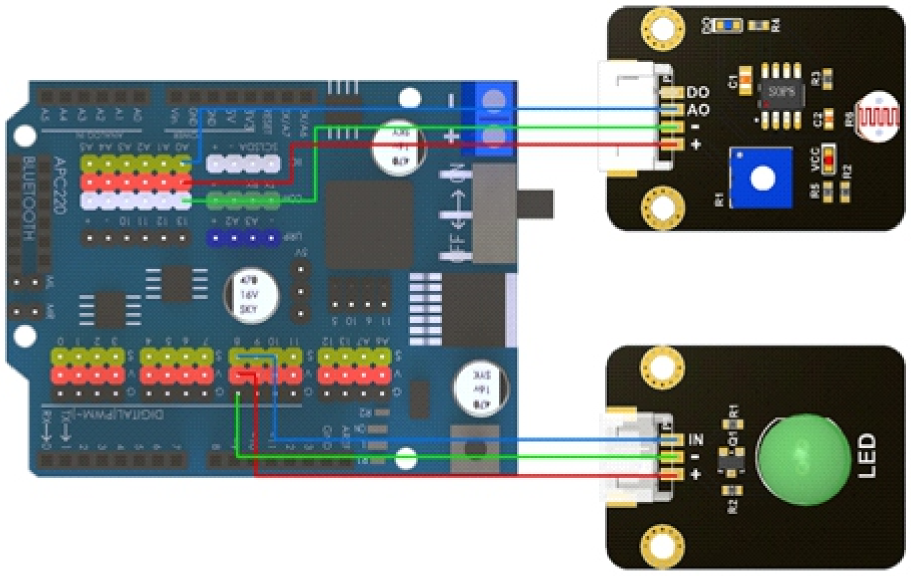
Module description: Photoresistor is a kind of light controlled variable resistor.Photoresistors have photoconductivity and can be used in photodetectors. Photoresistors are made of high resistance semiconductors. In the dark, the resistance of photoresist can be as high as several megaohms (m Ω), while in sufficient light, the resistance of photoresist can be as low as several hundred ohms. If the incident light on the photoresist exceeds acertain frequency, the photons are absorbed by the semiconductor to the bound electrons enough energy to jump into the conduction band. The resulting free electrons conduct electricity, thereby reducing resistance.

Working voltage: 3.3V~5V
Output type: Digital and analog interfaces. In this module, the stronger the incident light is, the lower the output analog value is.
Interface mode: PH2.0~4P
Module size: 35*26.3mm
Module weight: 42g

Pin denifition:

DO	Digital output
AO	Analog output
-	GND
+	VCC

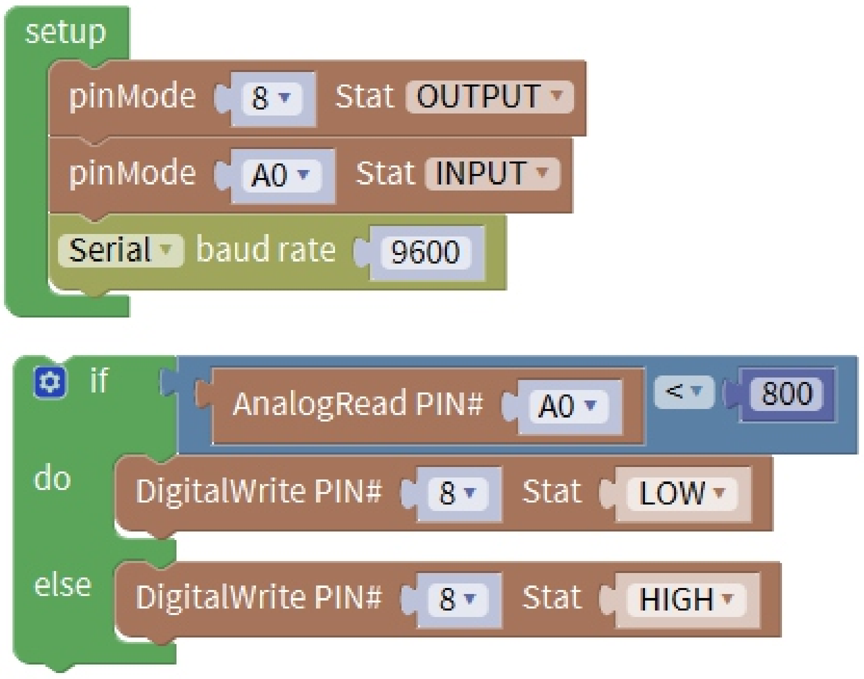
Arduino uno
Example 1
Connection diagram



Code

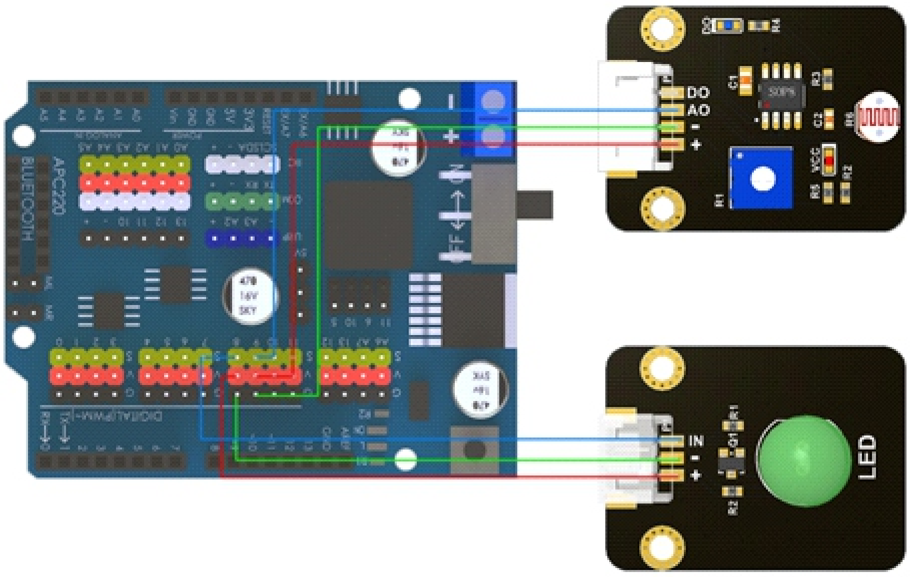
```
void setup() {
  pinMode(8, OUTPUT);           // Pin 8 of LED as output pin
  pinMode(A0,INPUT);            // Pin A0 of photoresistor as input pin
  Serial.begin(9600);           // Open 9600 serial port with baud rate
}
void loop(){
  int state = analogRead(A0);   // Read the analog input value of photoresistor

    Serial.println(state);           // Serial port output the analog input value of photoresistor
    if(state<800){  // Judge whether the analog value of photoresistor input is less than 800
        digitalWrite(8,LOW);      // Light on status
    }
    else{
        digitalWrite(8,HIGH);       // Light off status
    }
}
```

Mixly graphic programming



Result: The analog port of photoresistor conncet to Pin A0, Led connect to Pin 8. Illuminate the photoresistor module with a flashlight, whether the analog output value is higher than 800, the LED is on. Otherwise, the LED is off.

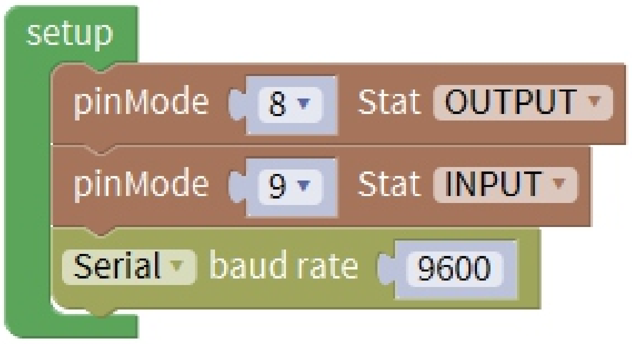
Example 2
Connection diagram



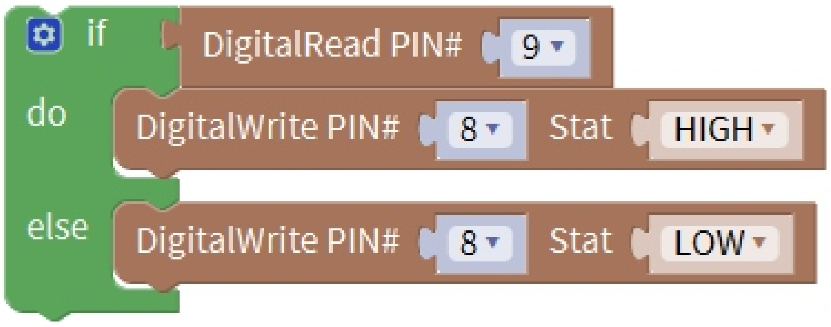
Code

```
void setup() {
  pinMode(8, OUTPUT);        // Pin 8 of LED as output pin
  pinMode(9,INPUT);            // Pin 9 of photoresistor as input pin S
  erial.begin(9600);    Open 9600 serial port with baud rate
}
void loop(){
  int state = digitalRead(9);   // Read the digital input value of photoresistor

    Serial.println(state);          // Serial port output the digial value of photoresistor
    if(state){    // Judge where whether the analog value of photoresistor input is  high
        digitalWrite(8, HIGH);        // Light on status
    }
    else{
        digitalWrite(8, LOW);        // Light off status
    }
}
```

Mixly graphic programming



```
setup
    pinMode 8 Stat OUTPUT
    pinMode 9 Stat INPUT
    Serial baud rate 9600
```



```
if DigitalRead PIN# 9
do   DigitalWrite PIN# 8 Stat HIGH
else DigitalWrite PIN# 8 Stat LOW
```

Result: Digital port of photoresistor connect to Pin 9, LED connect to Pin 8. When the output digital sigal of pin 9 is 0, led is on. Otherwise, led is off.

Active buzzer module

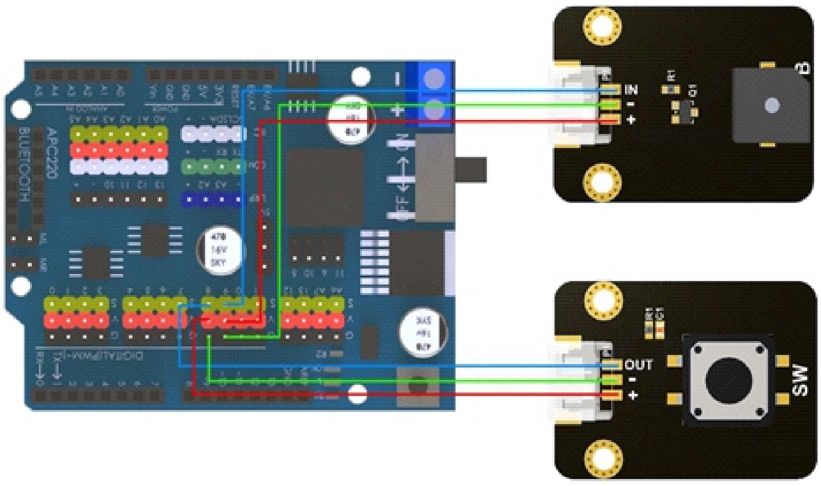
Module description: Buzzer is a kind of audio signal device. As an integrated electronic buzzer, it is powered by DC voltage and widely used in computers, printers, copiers, alarms,electronic toys, automotive electronic equipment, telephones, timers and other electronic product voice equipment. According to its driving mode, buzzer can be divided into activebuzzer and passive buzzer.
The difference between the active buzzer and the passive buzzer: the source here does not refer to the power supply, but refers to the vibration source. That is to say, there is a vibration source inside the active buzzer, so as long as the power is on, it will ring, and there is no vibration source inside the passive buzzer, so it is impossible to make it ring with DC signal. The buzzer described here is an active buzzer. As long as there is power. We output alternating high and low levels to make the buzzer sound.

Working voltage: 3.3V~5V
Output type: Digital output
Interface mode: pH2.0~3P
Module size: 35*26.3mm
Module weight: 47g

Pin denifition:

IN	Digital input
+	GND
-	VCC

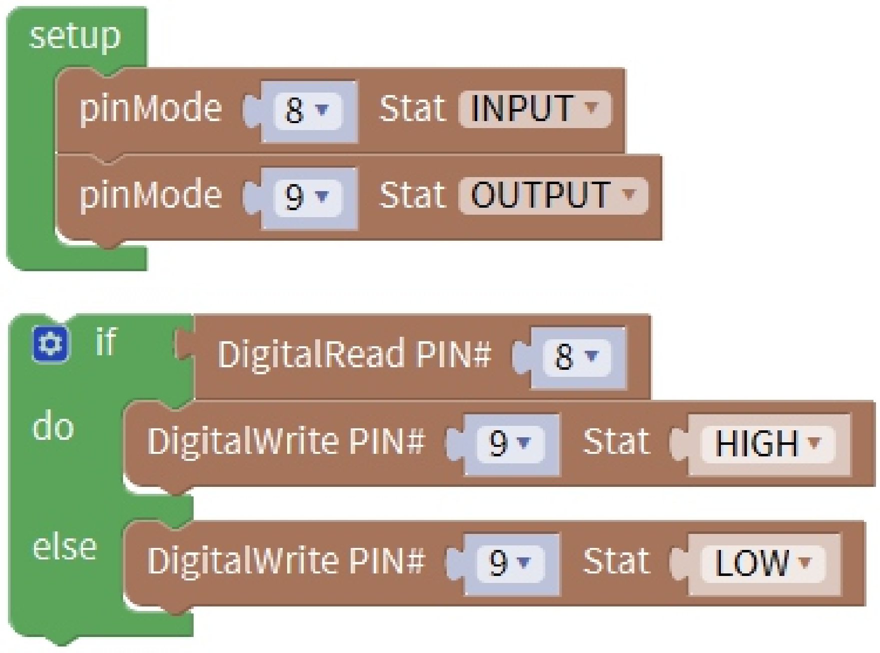
Arduino uno
Example 1
Connection diagram



Code

```
void setup() {
   pinMode(8,INPUT);     // Define button of Pin 8 as input pin
   pinMode(9,OUTPUT);  // Define active buzzer of Pin 9 as output pin
}
void loop() {
    int state = digitalRead(8);       // Read the button input value
    if(state){                   // Check whether input is high,  Press button is high.
       digitalWrite(9,HIGH);         // Buzzer sounds
    }
    else{
       digitalWrite(9,LOW);         // Buzzer off
   }
}
```

Mixly graphic program



Result: button connect to pin 8, buzzer connect pin 9. When press the button, buzzer sounds, when release the button, buzzer off.

Passive buzzer module

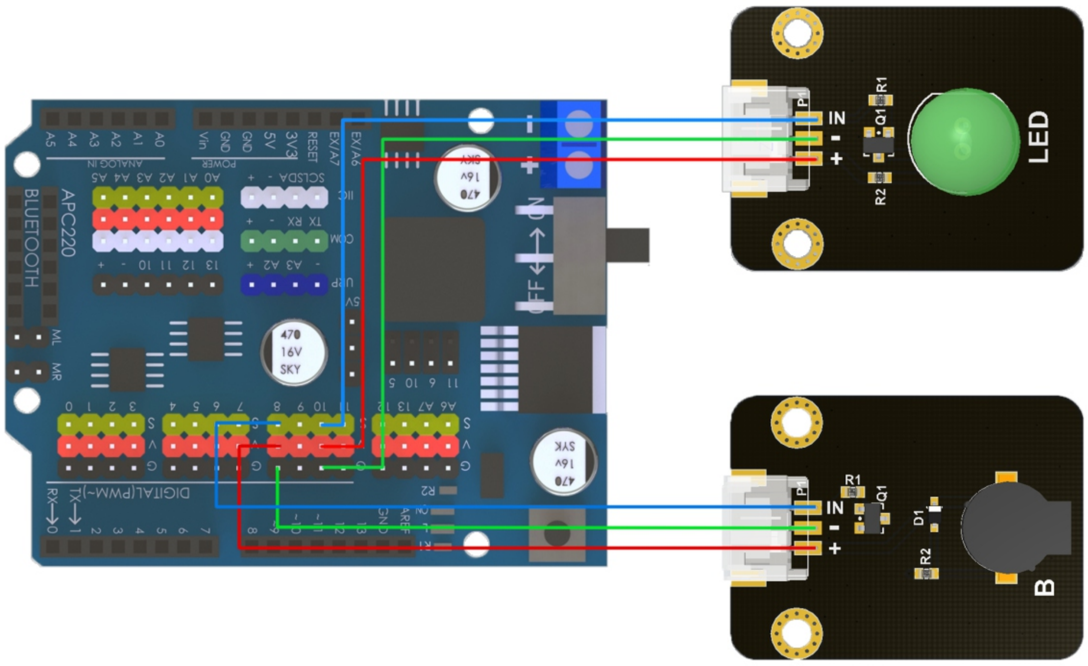
Module description: Buzzer is a kind of audio signal device. As an integrated electronic buzzer, it is powered by DC voltage and widely used in computers, printers, copiers, alarms,electronic toys, automotive electronic equipment, telephones, timers and other electronic product voice equipment. According to its driving mode, buzzer can be divided into active buzzer and passive buzzer. Is the input module, digital interface.
The difference between the active buzzer and the passive buzzer: the source here does not refer to the power supply, but refers to the vibration source. That is to say, there is a vibration source inside the active buzzer, so as long as the power is on, it will ring, and there is no vibration source inside the passive buzzer, so it is impossible to make it ring with DC signal. The buzzer described here is a passive buzzer.

Working voltage: 3.3V~5V
Output type: Digital output
Interface mode: pH2.0~3P
Module size: 35*26.3mm
Module weight: 40g

Pin denifition:

IN	Input
+	GND
-	VCC

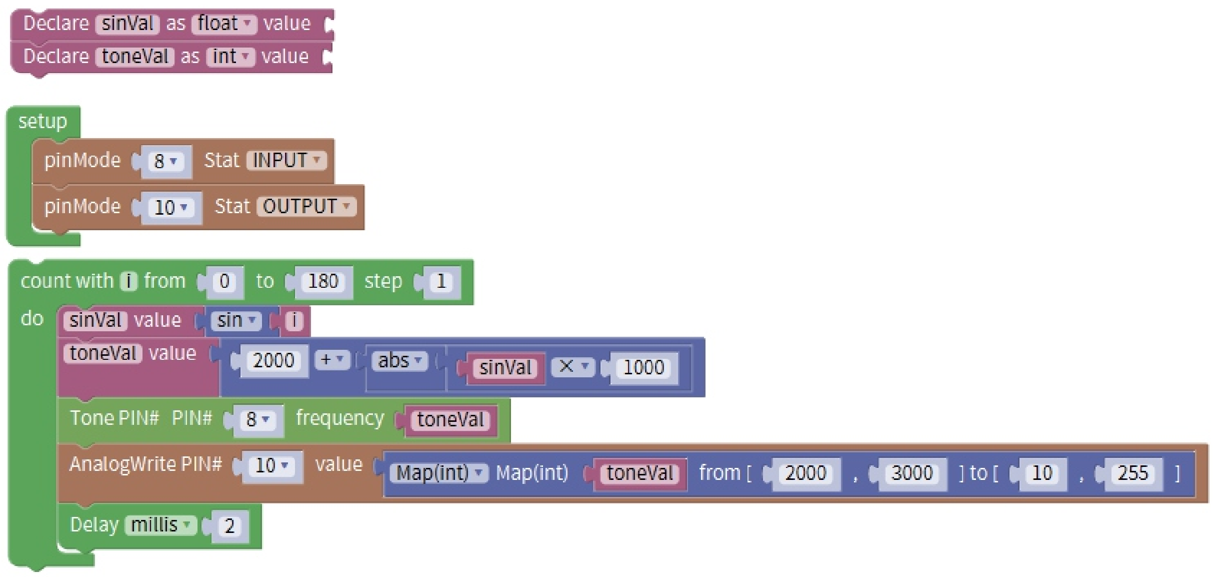
Arduino uno
Example 1
Connection diagram



代码

```
float sinVal;
int toneVal;
void setup() {
   // put your setup code here, to run once:
   pinMode(8,OUTPUT);
   pinMode(10,OUTPUT);
}

void loop() {
   // put your main code here, to run repeatedly:
   for(int x =0;x<180;x++){
      sinVal = (sin(x*(3.1412/180)));
      toneVal = 2000+(int(sinVal*1000));
      tone(8,toneVal);
      analogWrite(10,map(toneVal,2000,3000,10,255));
      delay(2);
   }
}
```

Mixly graphic program

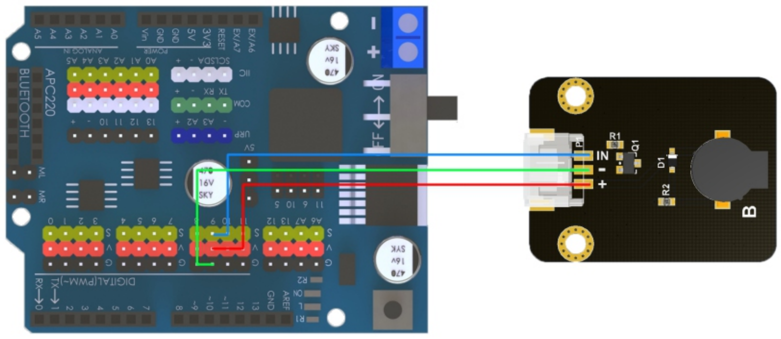


```
Declare sinVal as float value
Declare toneVal as int value

setup
   pinMode 8 Stat INPUT
   pinMode 10 Stat OUTPUT

count with i from 0 to 180 step 1
do
   sinVal value sin i
   toneVal value 2000 + abs sinVal × 1000
   Tone PIN# PIN# 8 frequency toneVal
   AnalogWrite PIN# 10 value Map(int) Map(int) toneVal from [ 2000 , 3000 ] to [ 10 , 255 ]
   Delay millis 2
```

Result: Led connect to pin 10, buzzer connect to pin 8. The buzzer will make a similar alarm like sound, and the brightness of the light will change with the fluctuation of the sound.

Example 2

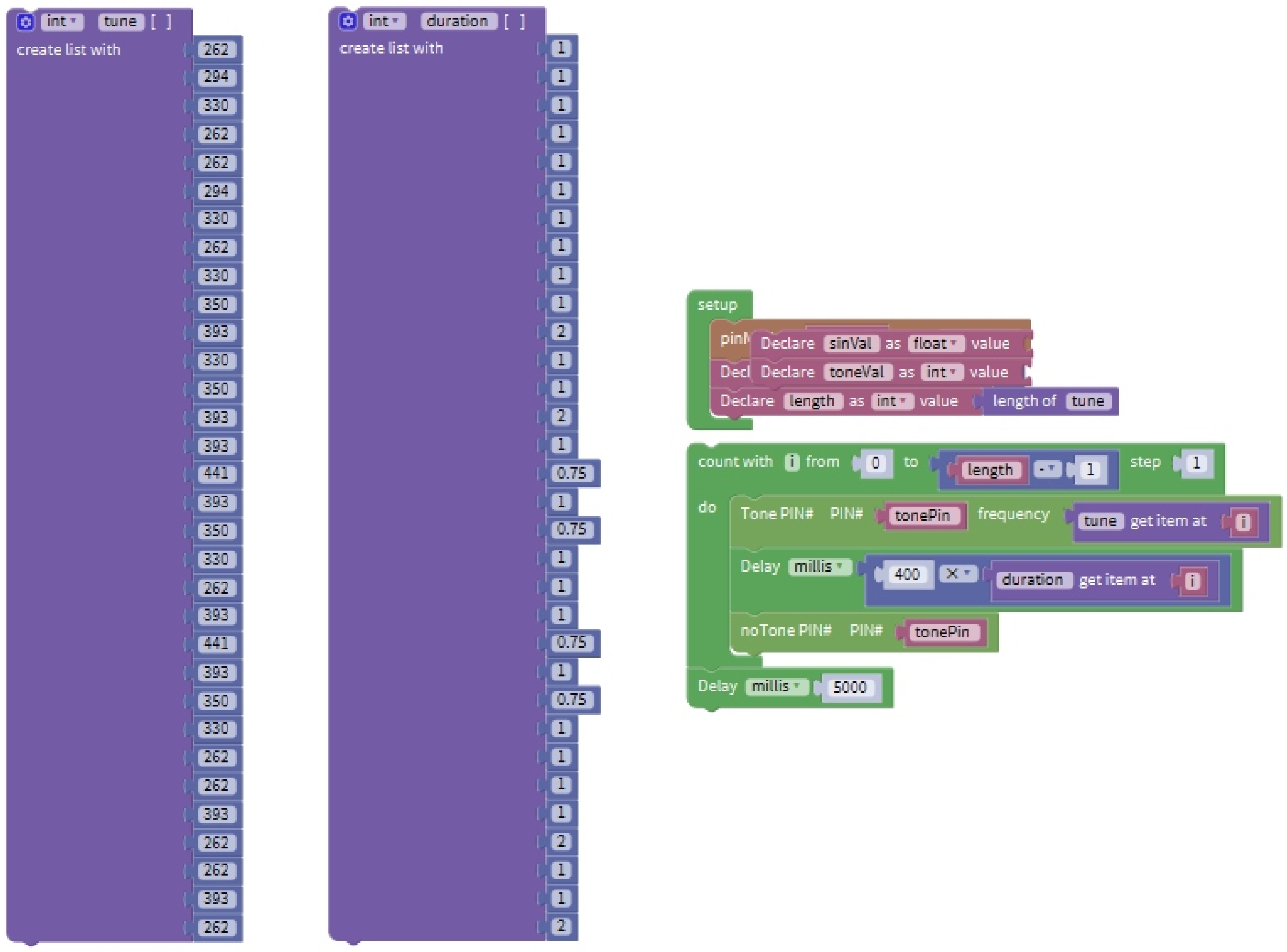
Connection diagram



Code

```
#define NOTE_D0 -1
#define NOTE_D1 262
#define NOTE_D2 294
#define NOTE_D3 330
#define NOTE_D4 350
#define NOTE_D5 393
#define NOTE_D6 441
#define NOTE_D7 495
#define NOTE_DL1 131
#define NOTE_DL2 147
#define NOTE_DL3 165
#define NOTE_DL4 175
#define NOTE_DL5 196
#define NOTE_DL6 221
#define NOTE_DL7 248

#define NOTE_DH1 525
#define NOTE_DH2 589
#define NOTE_DH3 661
#define NOTE_DH4 700
#define NOTE_DH5 786
#define NOTE_DH6 882
#define NOTE_DH7 990
//The definition above is to match each note with the frequency value

int tune[] =
{
  NOTE_D1,NOTE_D2,NOTE_D3,NOTE_D1,
  NOTE_D1,NOTE_D2,NOTE_D3,NOTE_D1,
  NOTE_D3,NOTE_D4,NOTE_D5,
  NOTE_D3,NOTE_D4,NOTE_D5,
  NOTE_D5,NOTE_D6,NOTE_D5,NOTE_D4,NOTE_D3,NOTE_D1,
  NOTE_D5,NOTE_D6,NOTE_D5,NOTE_D4,NOTE_D3,NOTE_D1,
  NOTE_D1,NOTE_D5,NOTE_D1,
NOTE_D1,NOTE_D5,NOTE_D1};    //This part is the note part of the whole piece, which is
defined by a sequence as tune, integer

float duration[]=

{
   1,1,1,1,
   1,1,1,1,
   1,1,1+1,
   1,1,1+1,
   0.5+0.5,0.25+0.5,0.5+0.5,0.25+0.5,1,1,
   0.5+0.5,0.25+0.5,0.5+0.5,0.25+0.5,1,1,
   1,1,1+1,
   1,1,1+1};
  //This part is the beat part of the whole song, and also defines the sequence duration and
floating point (the number of arrays is the same as the number of previous notes, one-to-
one correspondence)
int length;    //Define a variable, followed by how many notes there are

int tonePin=9;      //pin of buzzer

void setup()
{
   pinMode(tonePin,OUTPUT);     // Setting the pin of buzzer as output mode
length = sizeof(tune)/sizeof(tune[0]);      // Using a sizeof function, you can find out how
many notes there are in the tone sequence
}
void loop()
{   for(int x=0;x<length;x++)        //Number of circular notes
    {
      tone(tonePin,tune[x]);  //This function plays the array in the tune sequence in turn, that is, each note
      delay(400*duration[x]); //The duration of each note is the larger the adjustment time is, the slower
the music speed is, and the smaller the music speed is, the faster the music speed is


      noTone(tonePin);     // Stop the current note and enter the next note
      }
   delay(5000);        // Wait 5 seconds and the cycle starts again
}
```

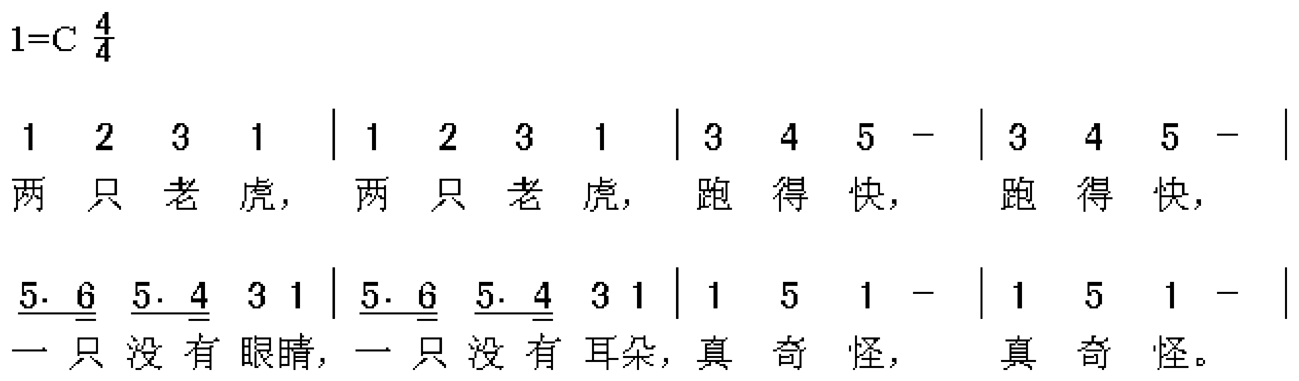
Mixly graphic programming



```
int tune [ ]
create list with 262, 294, 330, 262, 262, 294, 330, 262, 330, 350, 393, 330, 350, 393, 393, 441, 393, 350, 330, 262, 393, 441, 393, 350, 330, 262, 262, 393, 262, 262, 393, 262

int duration [ ]
create list with 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 2, 1, 0.75, 1, 0.75, 1, 1, 1, 0.75, 1, 0.75, 1, 1, 1, 1, 2, 1, 1, 2

setup
  pinM  Declare sinVal as float value
  Decl  Declare toneVal as int value
  Declare length as int value  length of tune

count with i from 0 to length - 1 step 1
do  Tone PIN#  PIN# tonePin  frequency  tune get item at i
    Delay millis  400 × duration get item at i
    noTone PIN#  PIN# tonePin
Delay millis 5000
```

Result: buzzer connect to pin 9, buzzer sounds "two tiger" song.
Expansion: use the buzzer to play a song you like!
Step 1: get a list of the corresponding sounds (note frequency table) from the song's score and the buzzer. Here are two tigers.

两 只 老 虎

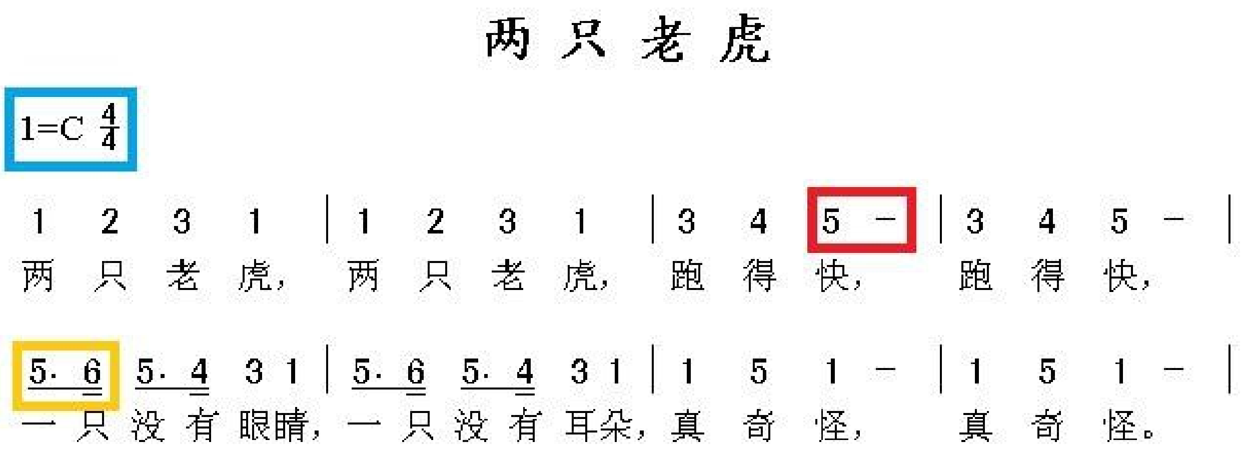

1=C 4/4
两 只 老 虎，两 只 老 虎，跑 得 快，跑 得 快，
一 只 没 有 眼睛，一 只 没 有 耳朵，真 奇 怪，真 奇 怪。

Tone Note	$\underset{.}{1}$	$\underset{.}{2}$	$\underset{.}{3}$	4	$\underset{.}{5}$	$\underset{.}{6}$	$\underset{.}{7}$
A	221	248	278	294	330	371	416
B	248	278	294	330	371	416	467
C	131	147	165	175	196	221	248
D	147	165	175	196	221	248	278
E	165	175	196	221	248	278	312
F	175	196	221	234	262	294	330
G	196	221	234	262	294	330	371

Tone Note	1	2	3	4	5	6	7
A	441	495	556	589	661	742	833
B	495	556	624	661	742	833	935
C	262	294	330	350	393	441	495
D	294	330	350	393	441	495	556
E	330	350	393	441	495	556	624
F	350	393	441	495	556	624	661
G	393	441	495	556	624	661	742

Tone Note	$\dot{1}$	2	$\dot{3}$	4	5	$\dot{6}$	$\dot{7}$
A	882	990	1112	1178	1322	1484	1665
B	990	1112	1178	1322	1484	1665	1869
C	525	589	661	700	786	882	990
D	589	661	700	786	882	990	1112
E	661	700	786	882	990	1112	1248
F	700	786	882	935	1049	1178	1322
G	786	882	990	1049	1178	1322	1484

Step 2: Look at the music score



In the notation, the blue box shows the tune in C key, so we only need to look at the C line in the note frequency table (marked in red font). Because there is no high and low tone in the notation, this piece is only used for the C-key part (marked with yellow background) ofthe mid note in the note frequency table. So we get the note sequence of two tigers. Only notes are not enough. We need to add rhythm to notes. Rhythm can be divided into one beat, half beat, one quarter beat, one eighth beat and two beats

There is a horizontal line behind 5 in the red box of the notation, indicating beat + 1. There is a dot on the right of 5 in the yellow box of the shorthand score, which means beat + 0.5; there is an underscore below 5, which means beat is 0.5; there are two underscores below 6, which means beat is 0.25. The rule is that there is no underline for a single note in time,that is, one beat, half beat (0.5) if there is underline, and quarter beat (0.25) if there are two underscores. After the note, there is "-" equal to beat + 1 of the previous note, and after the note, it is a little equal to beat + 0.5 of the previous note. So we get the beat sequence of two tigers.

Touch sensor module

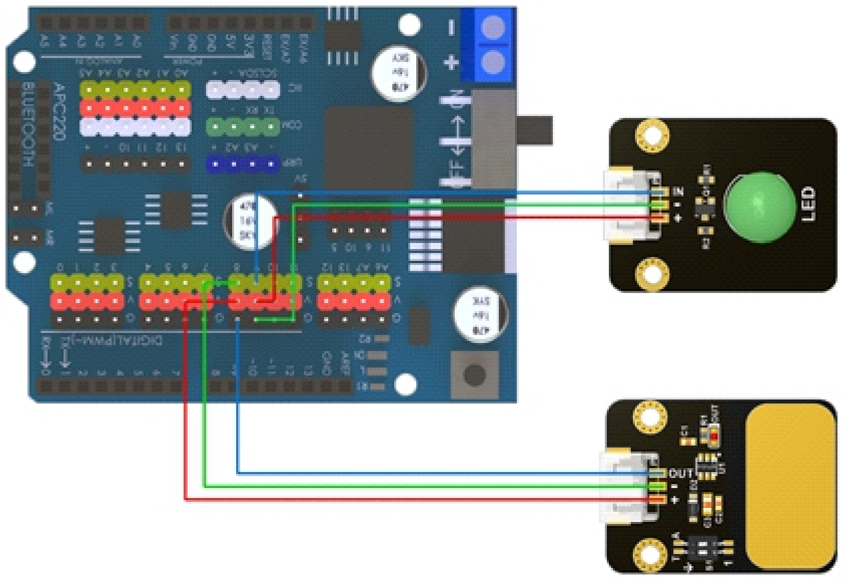
Module description: Touch module is a capacitive touch switch module based on touch detection. Normally, the module outputs low level;When touching the corresponding position with fingers, the module will output high level.The module can be installed on the surface of non-metallic materials such as plastic and glass.In addition, a thin piece of paper (non- metallic) can be covered on the surface of the module.As long as the touch position is correct, it can be made into a key hidden in the wall, desktop and other places.This module can help you to avoid the trouble of pressing buttons.

Working voltage:3.3V~5V
Output type:Digital signal
Working mode:4 type
Interface mode:PH2.0~3P
Module size:35*26.3mm
Module weight:36g

Pin denifition:

OUT	Output
-	GND
+	VCC

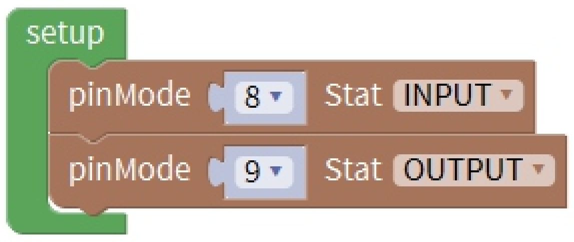
Arduino uno
Example 1
Connection diagram



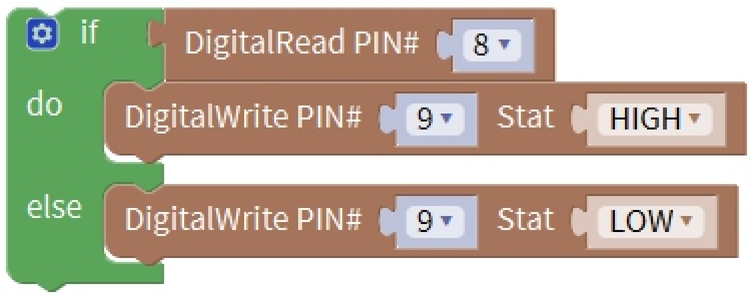
Code

```
void setup() {
   pinMode(8,INPUT);        // Pin 8 of touch sensor as input pin
   pinMode(9,OUTPUT);     // Pin 9 of LED as output pin
}

void loop() {
   int state = digitalRead(8);     //Read module level
   if(state){           //Judge the level of touch module
      digitalWrite(9,HIGH);       //LED on
   }
   else{
      digitalWrite(9,LOW);       //LED off
   }
}
```

Mixly graphic programming



```
setup
  pinMode 8 Stat INPUT
  pinMode 9 Stat OUTPUT
```



```
if DigitalRead PIN# 8
do   DigitalWrite PIN# 9 Stat HIGH
else DigitalWrite PIN# 9 Stat LOW
```

Result: touch module connect to pin 8, led connect to pin 9. When touch the sensor of module, led is on. The LED will not turn off until the touch module is no longer touched.

Tracking moudle

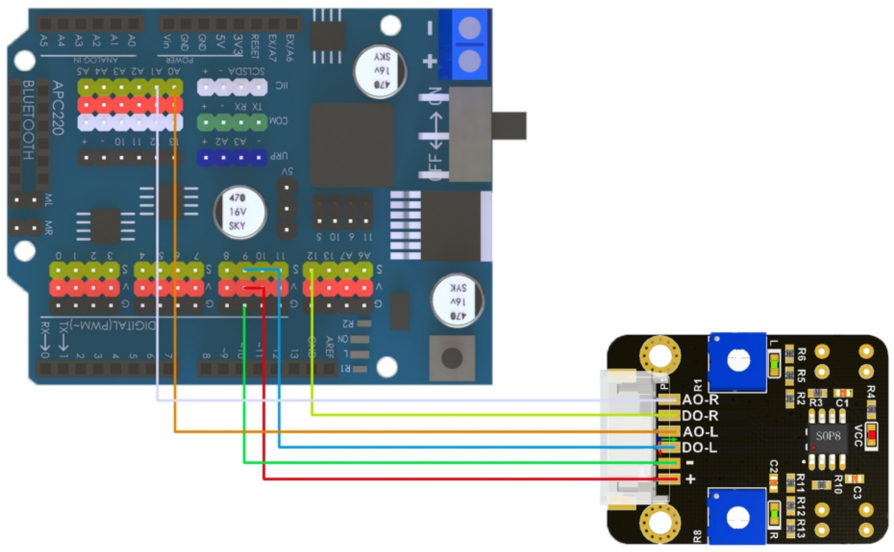
Module description: The tracking module is used to transmit light to the road by the infrared transmitting tube. When the infrared light encounters black, it is absorbed. The receiving tube does not receive the reflected light and outputs high level. When the red light meets other colors, the receiving tube receives the reflected light and outputs a low level. When using analog output, the module can be used as a gray-scale sensor. The gray-scale sensor is an analog sensor, which can sense different colors of the ground or desktop and generate corresponding signals.

Operating voltage: 3.3V~5V;
Output mode: digital signal, analog signal;
Interface mode: pH2.0~6P
Module size: 35*26.3mm
Module weight: 51g. 3mm

Pin definition:

AO-R	Right analog input
DO-R	Right digital input
AO-L	Left analog input
DO-L	Left digital input
-	GND
+	VCC

Arduin uno Example
1 Connection
diagram



Code

```
int data[2];
void setup() {
   pinMode(9,INPUT);//LD
   pinMode(12,INPUT);//RD
   pinMode(A0,INPUT);//LA
   pinMode(A1,INPUT);//RA
   Serial.begin(9600);
}

void loop() {
   distanceonD();
}

void distanceonD(){
   data[0] = digitalRead(9);
   data[1] = digitalRead(12);
   if(data[0] && data[1]){
      Serial.println("stop");
   }
   if(!data[0] && !data[1]){
      Serial.println("go");
   }
   if(!data[0] && data[1]){
      Serial.println("right");
   }
   if(data[0] && !data[1]){
      Serial.println("left");
   }
}
```

Result: when the infrared on both sides is blocked, the signal will not be received, and the high-level output will prompt "stop"; when the infrared on both sides is not blocked, the signal will prompt "go"; when the infrared on the left is blocked, the signal will prompt "left";when the infrared on the right is blocked, the signal will prompt "right".
Tips: Try to use a screwdriver to twist the cross port on the L and R of the module. In the test example, increase and reduce the distance between the detected object and the detected object, and observe the output content of the serial port and the movement of the car.

Example 2
Code

```
int data[2];
void setup() {
   pinMode(A0,INPUT);//LA
   pinMode(A1,INPUT);//RA

   Serial.begin(9600);
}

void loop() {
   distanceonA();
}

void distanceonA(){
   data[0] = analogRead(9);
   data[1] = analogRead(12);

   if(data[0]>500 && data[1]>500){
      Serial.println("stop");
   }
   if(data[0]<500 && data[1]<500 ){
      Serial.println("go");
   }
   if(data[0]<500   && data[1]>500 ){
      Serial.println("right");
   }
   if(data[0]>500   && data[1]<500 ){
      Serial.println("left");
   }
}
```

Result: when the infrared tracking module uses the analog signal to input the signal, it can be regarded as the gray-scale sensor. When the gray value on both sides is greater than 500, prompt "stop"; when the gray value on both sides is less than 500, prompt "go"; when the gray value on the left is greater than 500, and the gray value on the right is less than 500, prompt "right"; when the gray value on the left is less than 500, and the gray value on the right is greater than 500, prompt "right"; (500 in the code can be adjusted according to the actual situation)

Fan module

Module description: DC motor is a kind of motor that converts DC electric energy into mechanical energy. The most common type depends on the force produced by the magnetic field. Almost all types of DC motors have some internal mechanism, either electromechanical or electronic, to periodically change the current flow direction of some motors. Most types produce rotational motion; linear motors produce force and linear motion directly

Working voltage: 5V.
Input type: digital signal, PWM
Interface mode: ph2.0-4p
Module size: 35*26.3mm

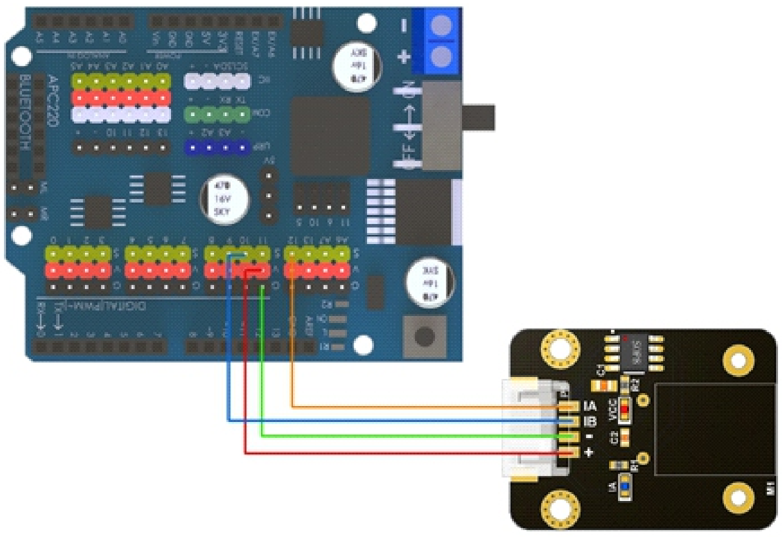
Pin denifition:

IA	Forward digital input
IB	Reverse digital input
-	GND
+	VCC

Control method

IA	IB	Function
0	0	Stop
0	1	Reverse
1	0	Forward

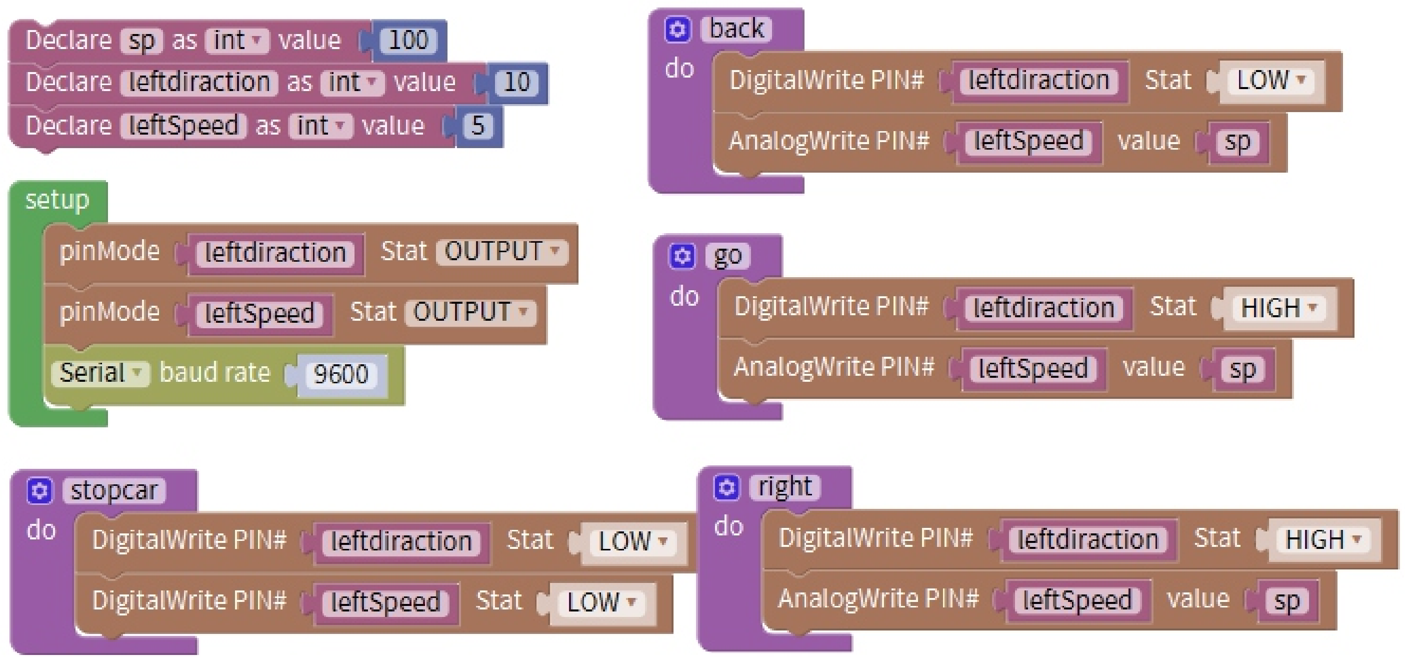
Arduino uno
Example 1
Connection diagram



Code

```
const int IB = 6;
const int IA = 5;
int buttonpin = A1;
boolean state = false;

void setup() {

    Serial.begin(9600);
    pinMode(IA, OUTPUT);
    pinMode(IB, OUTPUT);
    digitalWrite(IA, LOW);
    digitalWrite(IB, LOW);
    pinMode(buttonpin, INPUT);
    pinMode(4, INPUT);
}

void loop()
{

      analogWrite(IA, 0);
        analogWrite(IB, 0);
        delay(1000);

      analogWrite(IA, 0);
        analogWrite(IB, 255);
        delay(1000);

}
```

Mixly graphic programming



```
Declare sp as int value 100
Declare leftdiraction as int value 10
Declare leftSpeed as int value 5

setup
    pinMode leftdiraction Stat OUTPUT
    pinMode leftSpeed Stat OUTPUT
    Serial baud rate 9600

back
do  DigitalWrite PIN# leftdiraction Stat LOW
    AnalogWrite PIN# leftSpeed value sp

go
do  DigitalWrite PIN# leftdiraction Stat HIGH
    AnalogWrite PIN# leftSpeed value sp

stopcar
do  DigitalWrite PIN# leftdiraction Stat LOW
    DigitalWrite PIN# leftSpeed Stat LOW

right
do  DigitalWrite PIN# leftdiraction Stat HIGH
    AnalogWrite PIN# leftSpeed value sp
```

Result: IB interface connect to pin 6, IA interface connect to pin 5. After motor rotates for 1 second, stop rotating for 1 second as periodic cycle movement.

Infrared receiving module

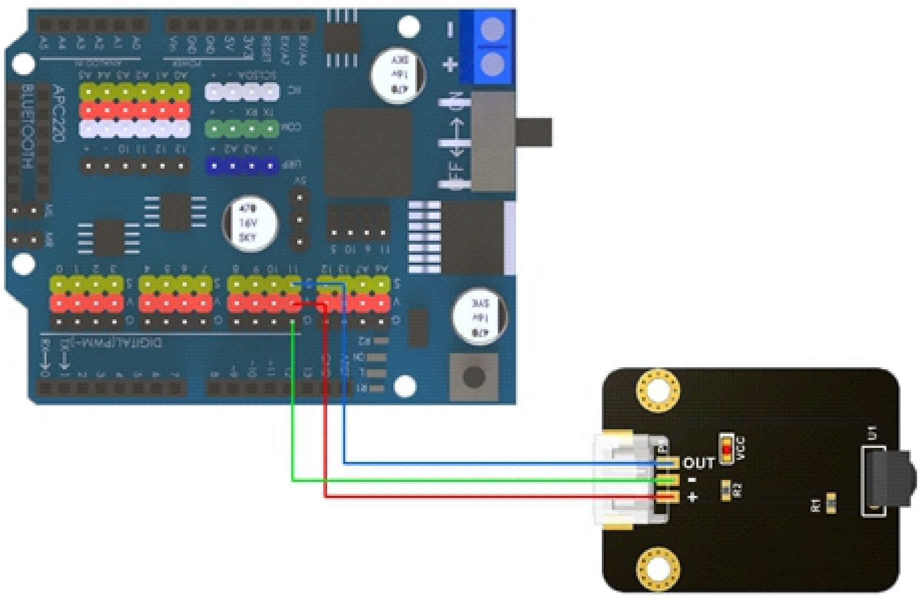
Module description: The external receiving tube is an electronic device that receives infrared light. For example, our TV sets, air conditioners and other household appliances need infrared receivers. We all know that the remote control emits infrared light. It is necessary for the TV to have an infrared receiver to receive the infrared signal from the remote control.

Working voltage: 3.3V~5V
Output type:digital signal
Interface mode: pH2.0~3P
Module size: 35*26.3mm
Module weight: 40g

Pin denifition:

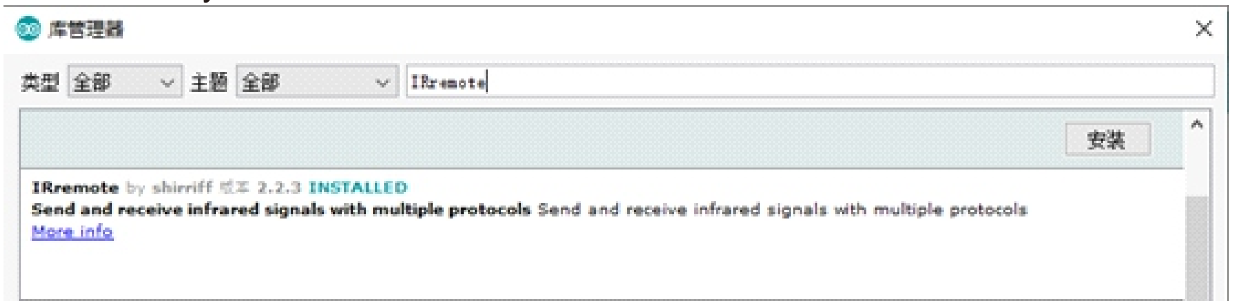
OUT	Output
-	GND
+	VCC

Arduino uno
Example 1
Connection diagram



Code

Before we start programming, we need to add the library "irremote. h" for infrared reception. Click "project" - > "load library" - > "manage library" - > "search" irremote ",and install the following irremote library.


库管理器

类型 全部 主题 全部 IRremote

安装

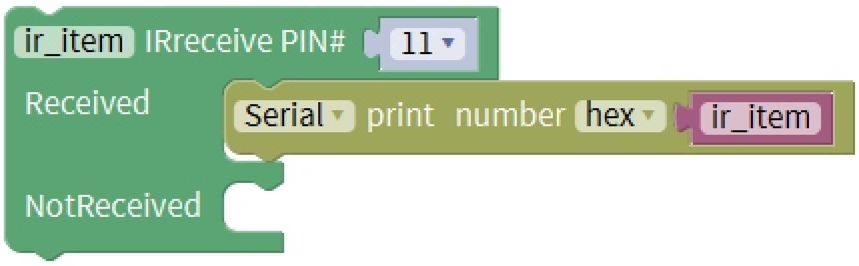
IRremote by shirriff 版本 2.2.3 INSTALLED
Send and receive infrared signals with multiple protocols Send and receive infrared signals with multiple protocols
More info

After installing the library, you can program it. We found irremote under "file" - >"example" - > under "third party library example", which provides some examples of the library using infrared reception.

code

```
#include <IRremote.h>
int RecvPin =11;//Pin 11 of infrared module
IRrecv irrecv(RecvPin);// Initialize irrecv instance
decode_results results;//Data received
void setup() {
   Serial.begin(9600);//Open the serial port with baud rate of 9600
   irrecv.enableIRIn();// Instance allowed to receive data
}
void loop() {
   if(irrecv.decode(&results)){//Decode results and determine whether data is received
      Serial.println(results.value,HEX);// The serial port outputs the received data value
      and outputs it in hexadecimal
      irrecv.resume();// Instance allowed to receive data
   }
}
```

Mixly graphic programming



Result: pin 11 is connected to the infrared receiving module, and the serial port window is opened. When the infrared remote control sends information to the infrared receiver, the infrared receiving module receives the information and displays it on the serial port window.Later, the received hex code can be used to control the lights, steering gear, motor and so on

Microphone module

（Sound sensor module)

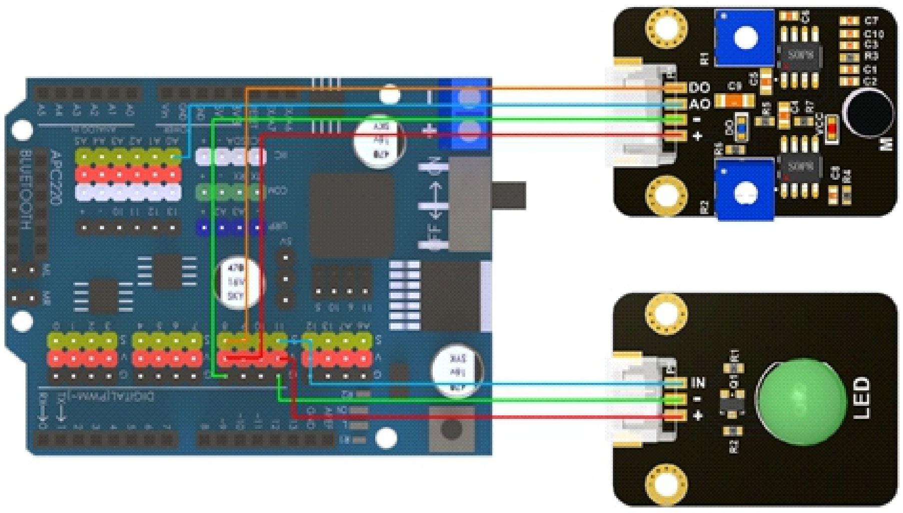
Module description: Output module, digital analog interface. It acts as a microphone. It is used to receive sound waves and display vibration images, but it can not measure the intensity of noise. The sound wave makes the electret film in the microphone vibrate,resulting in the change of capacitance, thus generating the corresponding change of micro voltage, which is received by the data collector through digital / analog conversion, and transmitted to the control board.
Ao analog output microphone's received sound signal in real time Do when the sound intensity reaches a certain threshold, the low-level signal is output (which can be adjusted by R2 on the module). Because the time of outputting the low-level signal is very short, in order to capture useful information, information filtering or interruption is used to present, here information filtering is used.

Working voltage: 3.3V~5V
Output type: Digital output, analog output
Interface mode: pH2.0~4P
Module size: 35*26.3mm
Module weight: 50g

Pin denifition:

DO	Digital output
AO	Analog output
-	GND
+	VCC

Arduino
Example 1
Connection diagram



Code

```
void setup() {
   // put your setup code here, to run once:
   Serial.begin(9600);
   pinMode(8,INPUT);
   pinMode(A0,INPUT);
   pinMode(11, OUTPUT);
}

void loop() {
  int astate = analogRead(A0);
   int dstate = digitalRead(8);
   if(dstate==0){
      Serial.print("Digital interface："); 
      Serial.println(dstate);
      Serial.print("Analog interface: ");
      Serial.println(astate);
      analogWrite(11,(map(astate, 1, 1023, 1, 255)));
   }
else {
      digitalWrite(11,LOW);
   }
}
```

Results: the do of the module was connected to pin 8, clapped at the microphone, and the LED light at the do of the module was on; on the contrary, the LED light was off. When rotating the crosshead on R2, it will be found that the higher the sensitivity of the module to sound will be if it rotates anticlockwise; when an active buzzer is added and the crosshead on R1 is rotated, it will be found that the module will amplify the received sound signal in different multiples, and the greater the received sound, the more obvious the amplification effect.

Potentiometer module

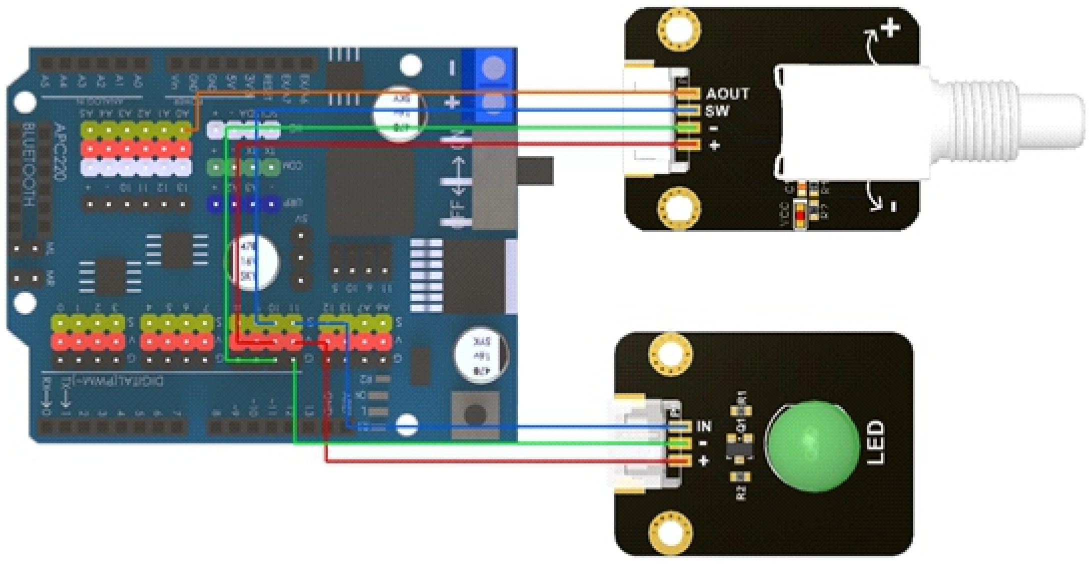
Module description: Potentiometer is a kind of three terminal resistance element whose resistance can be changed according to certain rules. It usually consists of a resistance element and a movable brush. When the brush moves along the element, a resistance or voltage is generated at the terminal relative to its moving distance.

Working voltage: 3.3V~5V
Output type: Obtain the digital output of the potentiometer switch and the analog output of the potentiometer value (0-1023).
Interface mode: PH2.0~4P
Module size: 35*26.3mm
Module weight: 93g

Pin denifition:

AUTO	Analog output
SW	Digital output
-	GND
+	VCC

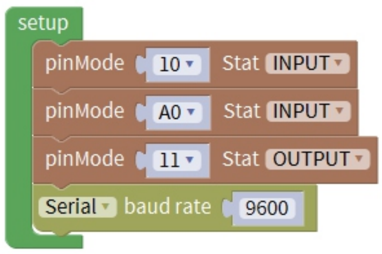
Arduino uno
Example 1
Connection diagram

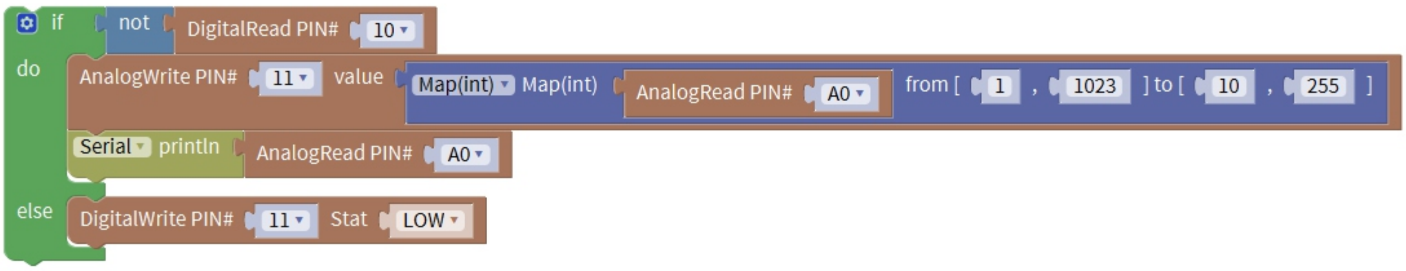


Code

```
void setup() {
   pinMode(10,INPUT);          // Pin 10 of pin SW of potentiometer as input pin
   pinMode(A0,INPUT);          // Pin A0 of pin AOUT of potentiometer as input pin
   pinMode(11,OUTPUT);         // Pin 11 of LED as output
   Serial.begin(9600);         // Open 9600 serial port with baud rate

}
void loop() {
   int state = digitalRead(10);    // Read the level of pin SW, High level means the key is not
pressed, low level means the key is pressed
   int value = analogRead(A0);  // Read the analog value of pin AOUT, 0~1023
   if(state==0){  // Judge whether you press the key
      analogWrite(11,map(value,0,1023,10,255));  // Change the brightness according to the
   analog value of potentiometer input
       }
   else{
      digitalWrite(11,LOW);     // LED off
   }
   Serial.println(value);  // Serial port output potentiometer analog value
}
```

Mixly graphic programming


```
setup
  pinMode 10 Stat INPUT
  pinMode A0 Stat INPUT
  pinMode 11 Stat OUTPUT
  Serial baud rate 9600
```


```
if not DigitalRead PIN# 10
do AnalogWrite PIN# 11 value Map(int) Map(int) AnalogRead PIN# A0 from [ 1 , 1023 ] to [ 10 , 255 ]
   Serial println AnalogRead PIN# A0
else DigitalWrite PIN# 11 Stat LOW
```

Results: pin 10 was connected to the digital interface of the potentiometer, pin A0 to the analog interface of the potentiometer, and pin 11 to the LED lamp. The brightness of the LED light changes with the twist of the potentiometer.

Ultrasonic Module

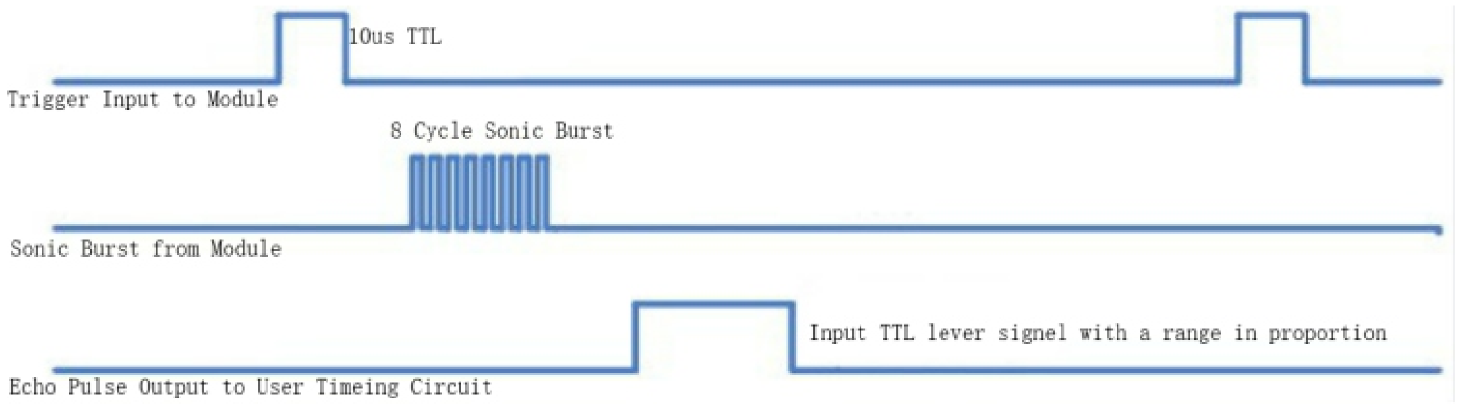
Module description: This module is used to measure the distance. By sending and receiving ultrasonic waves, it measures the time required for the sound to rebound from the object and return to the sensor. Using the time difference and sound propagation speed, it calculates the distance between the module and the obstacles in front.

Working voltage: 3.3V~5V
Minimum measurement distance measurement: 2cm
Maximum measurement distance measurement: 350cm
Type: Echo is the input digital signal, trig is the output digital signal.
Interface mode: Ph2.0-4p
Module size: 35*26.3mm
Module weight: 86g

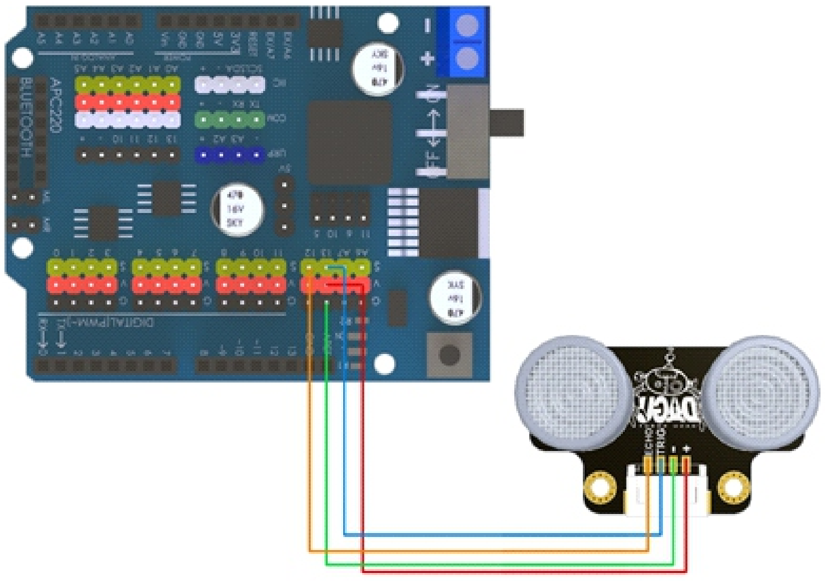
Pin definition:

Echo	Receive input
Trig	Send output
-	GND
+	VCC

Working principle: input a high level of more than 10us to trig pin to trigger module ranging. When the ranging is finished, the echo pin will output a high level, and the level width is the sum of the ultrasonic round-trip time.


10us TTL
Trigger Input to Module
8 Cycle Sonic Burst
Sonic Burst from Module
Input TTL lever signel with a range in proportion
Echo Pulse Output to User Timeing Circuit

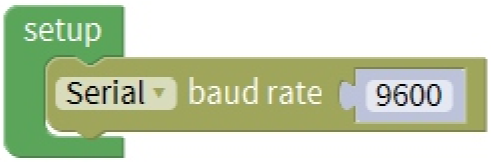
Arduino uno
Example 1
Connection diagram



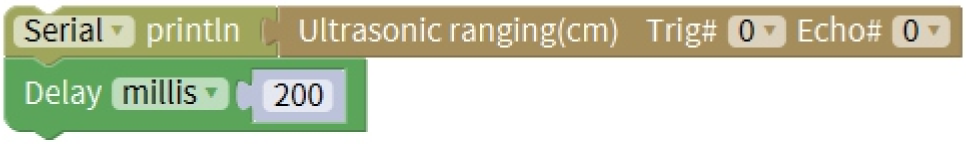
Code

```
void setup() {
    pinMode(13,OUTPUT);     // Pin 13 of send as output pin
    pinMode(12,INPUT);      // Pin 12 of receiver as input pin
    Serial.begin(9600);     // Open 9600 serial port with baud rate
}

void loop() {
    digitalWrite(13,LOW);       // Trigger ultrasonic ranging
    delayMicroseconds(2);
    digitalWrite(13,HIGH);
    delayMicroseconds(10);
    digitalWrite(13,LOW);
    int distance = pulseIn(12,HIGH)/58;        // Calculate distance
    Serial.println(distance);                  // Output distance of serial port
    delay(200);     // Delay 200ms
}
```

Mixly graphic programming



```
setup
  Serial baud rate 9600
```



```
Serial println Ultrasonic ranging(cm) Trig# 0 Echo# 0
Delay millis 200
```

Result: Trig connect to pin 13, Echo connect to pin 12. Open serial port, the obstacles in front of the mobile ultrasonic ranging module and the data output to the serial port change.

Explanation: the time returned by pulsein() is the time when the ultrasound is sent to the receiver, the unit is subtle, and the sound speed is 340m / s. because distance = speed *time, the distance (CM) = 340 * 100 / 1000000 * pulsein() / 2, that is, distance (CM) =pulsein() / 58.