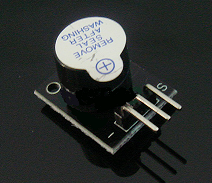
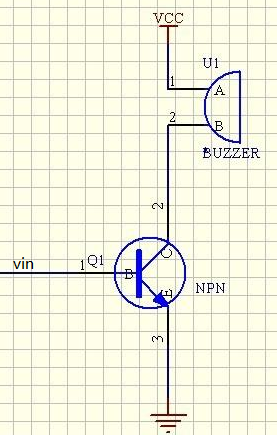
Pls contact us :35007117@ qq.com

----------------------------------------------

Buzzer module



Arduino buzzer module  
Active speaker  
Compatible with PC, printer, car audio system DIY  
  
**Specifications:**  
  
Voltage: 5V  
Color: Black + silver gray  
Package dimension: 77x42x13mm  
Weight: 5g



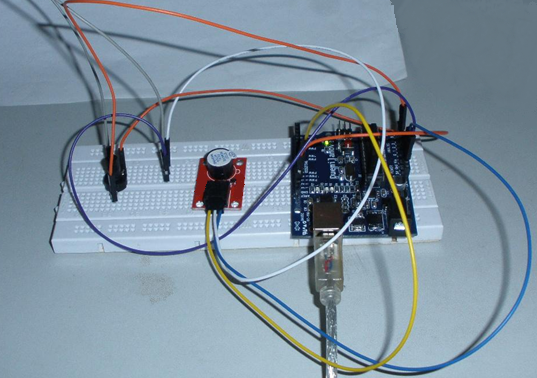
Test Requires

Arduino Controller × 1

USB Cable × 1

Buzzer Module × 1

10K potentiometer（10K） × 1

Connection: 

Below test is use Analog to control buzzer frequency.

Pin10 is to control buzzer.

Pin3 is Analog control, and we use 10K potentiometer.

Function: Adjust potentiometer, we can hear buzzer frequency change.

int speakerPin = 8;//Control horn pin

int potPin = 4;//Pin for controlling the adjustable resistor

int value = 0;

void setup() {

pinMode(speakerPin, OUTPUT);

}

void loop() {

value = analogRead(potPin);Read the value of the resistor pin

digitalWrite(speakerPin, HIGH);

delay(value);Adjust the time of the horn;

digitalWrite(speakerPin, LOW);

delay(value);Adjust the duration of the horn;

}

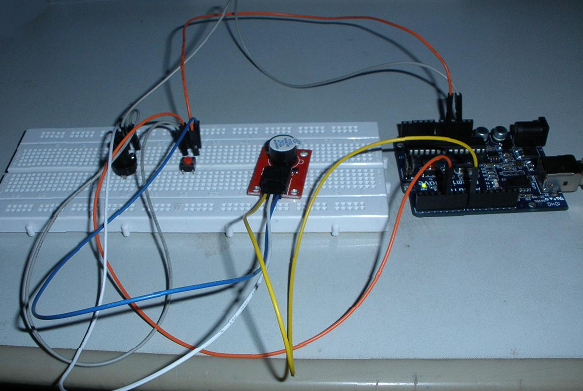
We can use potentiometer to change delay time, then we can change buzzer frequency.

Here we add a button switch to control the buzzer, so we can simulate a simple doorbell, when the

When the button is pressed the speaker can make a sound.

Another Test:

We add button to control buzzer, then we can imitate a simple doorbell. When push button, buzzer can ring.

Connect way:

Code:

const int buttonPin = 4; // Push pin;

const int speakerPin = 8; //Buzzer pin;

// variables will change:

int buttonState = 0; // Read a value of the button pin

void setup()

{

//Set the button pin for the input mode, the buzzer pin for the output mode;

pinMode(speakerPin, OUTPUT);

pinMode(buttonPin, INPUT);

}

void loop(){

// Read the key to an initial value, here in the circuit, I was in the default high, so the initial value is high;

buttonState = digitalRead(buttonPin);

/\*If the button is high, then the buzzer does not ring;

Because I just started in the hardware circuit in the initial value is high, so the if condition was established, the buzzer does not ring

\*/

if (buttonState == HIGH) {

digitalWrite(speakerPin,LOW);

}

else {

//The value of the button here is low (also when the button is pressed).

digitalWrite(speakerPin,HIGH);

}

}

Below is use PWM to control buzzer:

Code:

int speakerPin = 8;

byte song\_table[] = { 30, 30, 30, 40, 50, 60, 70, 80, 90, 100,110, 120, 130, 140,

150, 160, 170, 180, 190, 200, 210, 220, 230, 240, 250, 250, 240, 230, 220, 210, 200, 190, 180,

170, 160, 150, 140, 130, 120, 110, 100, 90, 80, 70, 60, 50, 40, 30, 30, 30 };

int MAX = 50;

int count = 0;

void setup() {

pinMode(speakerPin, OUTPUT);

}

void loop() {

analogWrite(speakerPin,song\_table[count]);

count ++;

if (count > MAX) {

count = 0;

}

delay(50);

}