



keyestudio Power Amplifier Module



Description

The keyestudio power amplifier module integrates an adjustable potentiometer, a speaker, an audio amplifier 8002B chip and a 3pin header interface.

It can amplify a small audio signal output of about 8.5 times, and can also play the amplified signal out via a small power speaker.

It can also be used to play music, or used as an external sound amplification device for some music playback devices.

Please notice that you should adjust the volume to the minimum level before use, then gradually increase it, in case to burn the speaker.

When this module is connected to a microcontroller to test this function, you can input square waves of different frequency and different duration at the signal end, so as to edit the sound aired from the speaker.

Parameters

- Amplifier chip: SC8002B
- Operating voltage: DC 5V

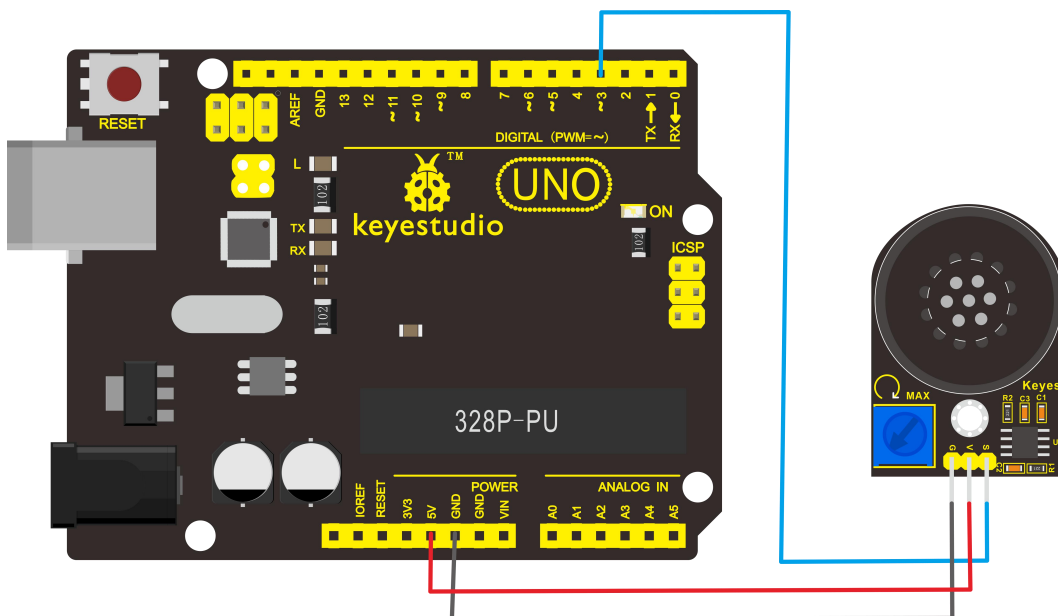


- Operating current: $\geq 500\text{MA}$
- Maximum power: 2W
- Operating temperature: 0-40°
- Speaker power: 0.15W
- Speaker sound volume: 80db
- Property: ROHS

Technical Details

- Dimensions: 47mm*30mm*13mm
- Weight: 8.4g

Connect It Up



Connect the power amplifier module to control board using three jumper wires. Then connect the control board to your PC with a USB cable.

Upload the Code



////////////////////////////////////

```
#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 all the frequencies of all the D tune
#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 the tune according to the 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,
```



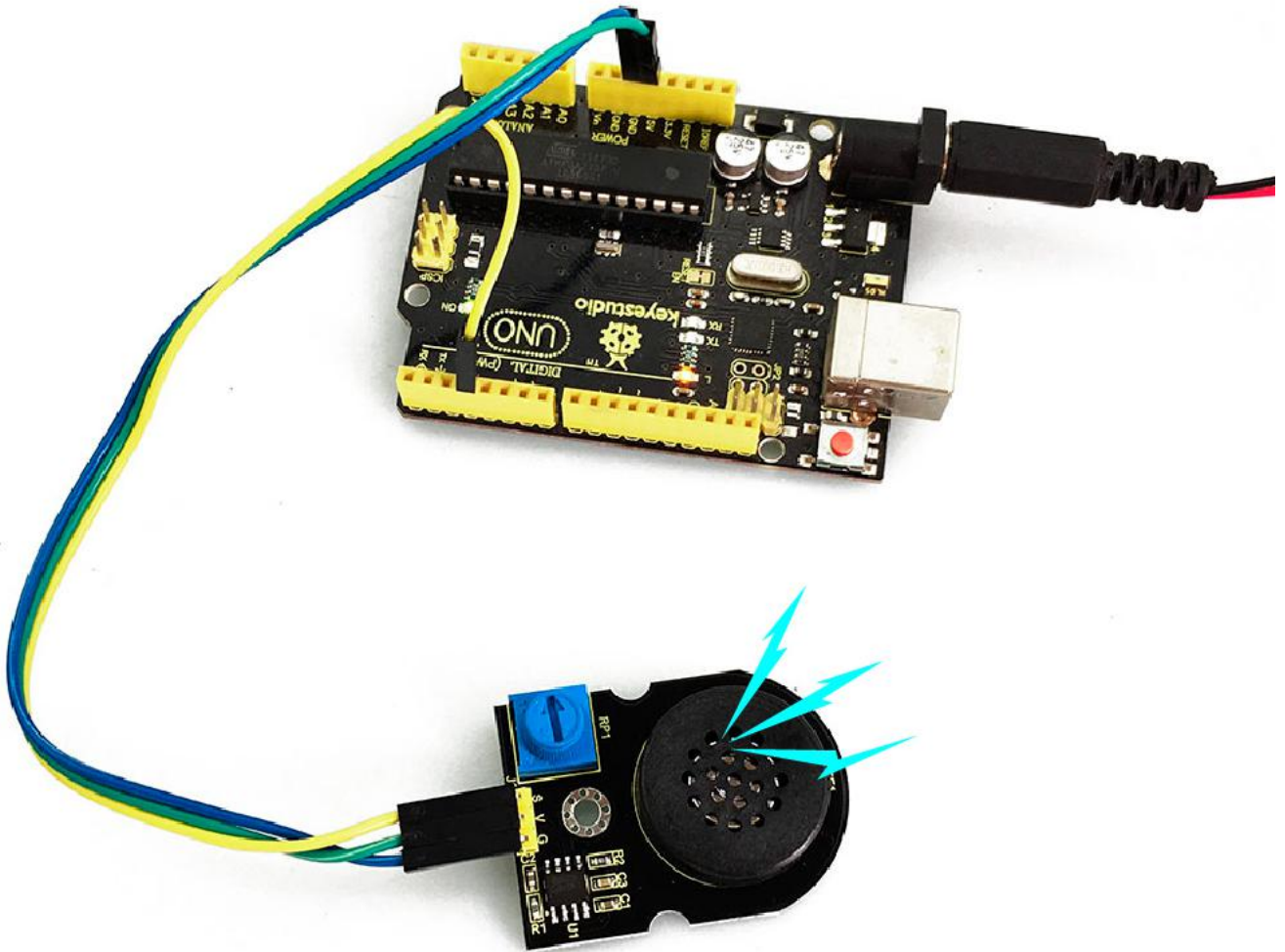
```
M2, M3, M4, M3, M2,
M1, M2, D5, D0,
M3, M3, M4, M5,
M5, M4, M3, M4, M2,
M1, M1, M2, M3,
M2, M1, M1
};
float durt[]=          // list out the beats according to the musical notation
{
    1, 1, 1, 1,
    1, 1, 1, 1,
    1, 1, 1, 1,
    1+0.5, 0.5, 1+1,
    1, 1, 1, 1,
    1, 1, 1, 1,
    1, 1, 1, 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 pin 3
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]);    //adjust the delay time according to tone, you can change the index 500
        for the music.
        noTone(tonepin);
    }
    delay(2000);
}
```



//

What You Should See

After uploading the code, you should be able to hear the song Ode To Joy. If the volume is too small, you can adjust the potentiometer to increase it.



Resource

<https://fs.keyestudio.com/KS0374>