

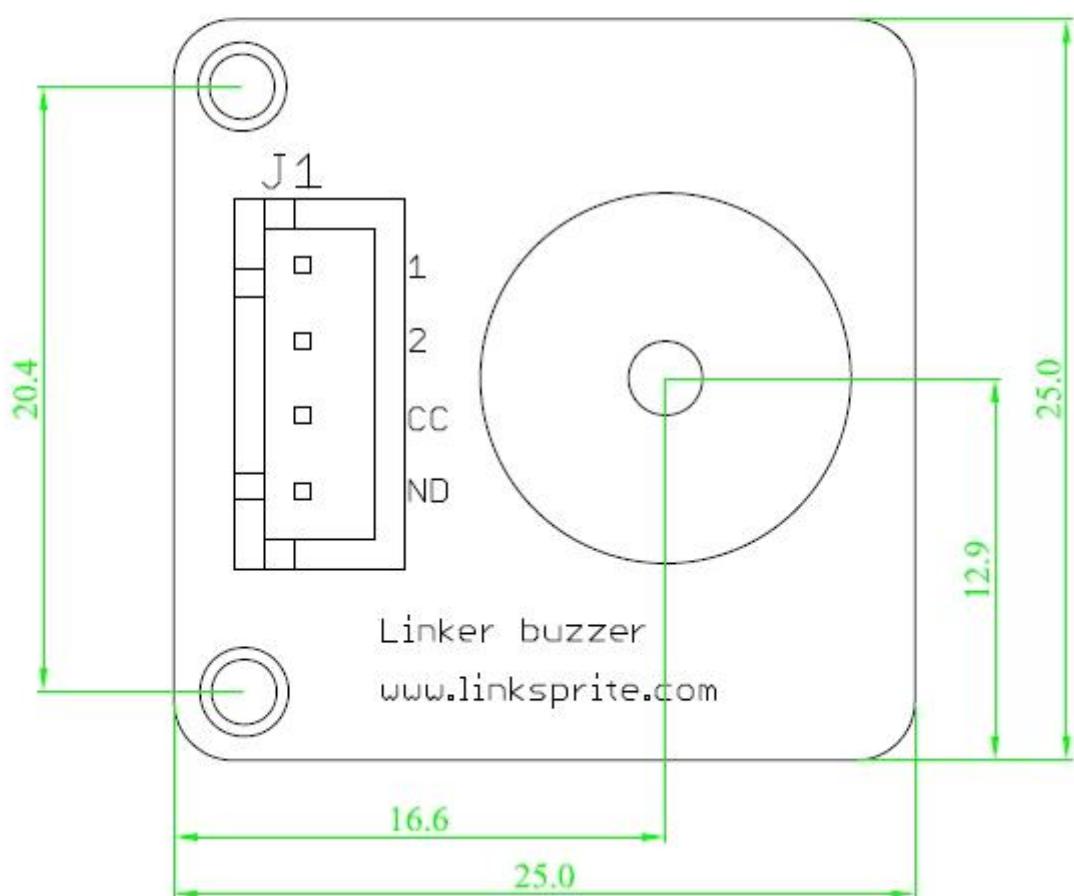
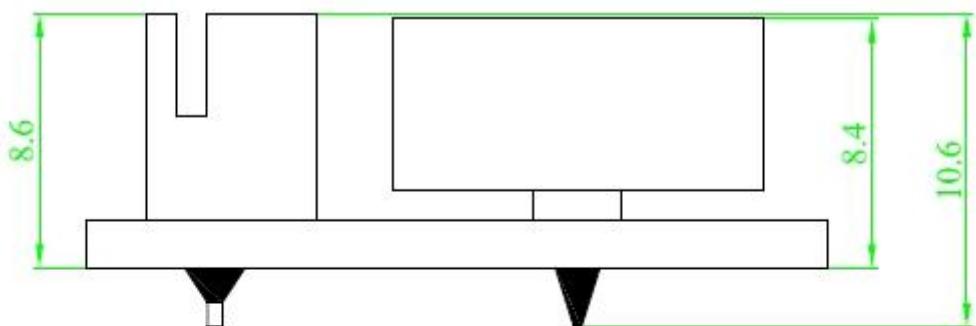
## Linker Kit Platine mit Buzzer

This buzzer module will generate around the audible 2kHz range.



## Features

Dimensions: 25.0×25.0×10.6mm



## Schematics

- [Schematics](#)

## Application Ideas

```

int speakerPin = 9;
int length = 15; // the number of notes
char notes[] = "ccggaagffeeddc "; // a space represents a rest
int beats[] = { 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 2, 4 };
int tempo = 300;
///////////////////////////////
    
```

```

void playTone(int tone, int duration)
{
    for (long i = 0; i < duration * 1000L; i += tone * 2)
    {
        digitalWrite(speakerPin, HIGH);
        delayMicroseconds(tone);
        digitalWrite(speakerPin, LOW);
        delayMicroseconds(tone);
    }
}
///////////////////////////////
void playNote(char note, int duration)
{
    char names[] = { 'c', 'd', 'e', 'f', 'g', 'a', 'b', 'c' };
    int tones[] = { 1915, 1700, 1519, 1432, 1275, 1136, 1014, 956 };      // play the tone corresponding to the note name
    for (int i = 0; i < 8; i++)
    {
        if (names[i] == note)
        {
            playTone(tones[i], duration);
        }
    }
}
/////////////////////////////
void setup()
{
    pinMode(speakerPin, OUTPUT);
    pinMode(11, OUTPUT);
    digitalWrite(11, LOW);
}
/////////////////////////////
void loop()
{
    for (int i = 0; i < length; i++)
    {
        if (notes[i] == ' ')
        {
            delay(beats[i] * tempo); // rest
        }
        else
        {
            playNote(notes[i], beats[i] * tempo);
        }          // pause between notes
        delay(tempo / 2);
    }
}

```

