

Buzzer

Definizione: componenti elettronici in grado di emettere un suono se alimentati o se soggetti ad un segnale.

Si distinguono in buzzer attivi e buzzer passivi.

Buzzer attivi: sono dotati di un circuito integrato con un oscillatore interno; se sottoposti ad una alimentazione in continua, emettono un suono caratteristico la cui frequenza non può essere controllata. Il loro utilizzo è molto semplice.

Caratteristiche:

- tensione di alimentazione 3.3 V; 5 V
- corrente attivo < 30 mA
- frequenza di risonanza 2500 Hz \pm 300 Hz
- suono minimo in uscita 85 dB a 10 cm

Esempio:

```
#define PIN 11

void setup() {

  pinMode(PIN,OUTPUT); //Impostiamo il pin come uscita
}

void loop() {

  digitalWrite(PIN,HIGH);

  delay(1000);

  digitalWrite(PIN,LOW);

  delay(1000);

}
```

Buzzer passivi: sono meno costosi di quelli attivi perché non dotati di circuito interno. Tuttavia, risulta più difficoltoso il loro utilizzo in quanto bisogna fornire un'onda quadra per generare frequenze ben precise. Per un corretto uso con la scheda Arduino, si collega un resistore da 220 Ω

Caratteristiche:

- tensione di alimentazione 3.3 V; 5 V
- frequenza in uscita 50 Hz-14 kHz
- suono minimo in uscita 85 dB a 10 cm

Esempio:

```
// elenco note con le relative frequenze
```

```
#define NOTE_B0 31
```

```
#define NOTE_C1 33
```

```
#define NOTE_CS1 35
```

```
#define NOTE_D1 37
```

```
#define NOTE_DS1 39
```

```
#define NOTE_E1 41
```

```
#define NOTE_F1 44
```

```
#define NOTE_FS1 46
```

```
#define NOTE_G1 49
```

```
#define NOTE_GS1 52
```

```
#define NOTE_A1 55
```

```
#define NOTE_AS1 58
```

```
#define NOTE_B1 62
```

```
#define NOTE_C2 65
```

```
#define NOTE_CS2 69
```

```
#define NOTE_D2 73
```

```
#define NOTE_DS2 78
```

```
#define NOTE_E2 82
```

```
#define NOTE_F2 87
```

```
#define NOTE_FS2 93
```

```
#define NOTE_G2 98
```

```
#define NOTE_GS2 104
```

```
#define NOTE_A2 110
```

```
#define NOTE_AS2 117
```

```
#define NOTE_B2 123
```

```
#define NOTE_C3 131
#define NOTE_CS3 139
#define NOTE_D3 147
#define NOTE_DS3 156
#define NOTE_E3 165
#define NOTE_F3 175
#define NOTE_FS3 185
#define NOTE_G3 196
#define NOTE_GS3 208
#define NOTE_A3 220
#define NOTE_AS3 233
#define NOTE_B3 247
#define NOTE_C4 262
#define NOTE_CS4 277
#define NOTE_D4 294
#define NOTE_DS4 311
#define NOTE_E4 330
#define NOTE_F4 349
#define NOTE_FS4 370
#define NOTE_G4 392
#define NOTE_GS4 415
#define NOTE_A4 440
#define NOTE_AS4 466
#define NOTE_B4 494
#define NOTE_C5 523
#define NOTE_CS5 554
#define NOTE_D5 587
#define NOTE_DS5 622
```

```
#define NOTE_E5 659
#define NOTE_F5 698
#define NOTE_FS5 740
#define NOTE_G5 784
#define NOTE_GS5 831
#define NOTE_A5 880
#define NOTE_AS5 932
#define NOTE_B5 988
#define NOTE_C6 1047
#define NOTE_CS6 1109
#define NOTE_D6 1175
#define NOTE_DS6 1245
#define NOTE_E6 1319
#define NOTE_F6 1397
#define NOTE_FS6 1480
#define NOTE_G6 1568
#define NOTE_GS6 1661
#define NOTE_A6 1760
#define NOTE_AS6 1865
#define NOTE_B6 1976
#define NOTE_C7 2093
#define NOTE_CS7 2217
#define NOTE_D7 2349
#define NOTE_DS7 2489
#define NOTE_E7 2637
#define NOTE_F7 2794
#define NOTE_FS7 2960
#define NOTE_G7 3136
```

```
#define NOTE_G5 3322

#define NOTE_A7 3520

#define NOTE_AS7 3729

#define NOTE_B7 3951

#define NOTE_C8 4186

#define NOTE_CS8 4435

#define NOTE_D8 4699

#define NOTE_DS8 4978

//inizio programma vero e proprio

#define PIN 3

#define NOTE_C5 523

#define NOTE_D5 587

#define NOTE_E5 659

#define NOTE_F5 698

#define NOTE_G5 784

#define NOTE_A5 880

#define NOTE_B5 988

#define NOTE_C6 1047

//Note melodia:

unsigned int melody[] = {

    NOTE_C5, NOTE_D5, NOTE_E5, NOTE_F5, NOTE_G5, NOTE_A5, NOTE_B5, NOTE_C6};

unsigned int tempo = 500;

void setup() {

    pinMode(PIN, OUTPUT);

}

void loop() {

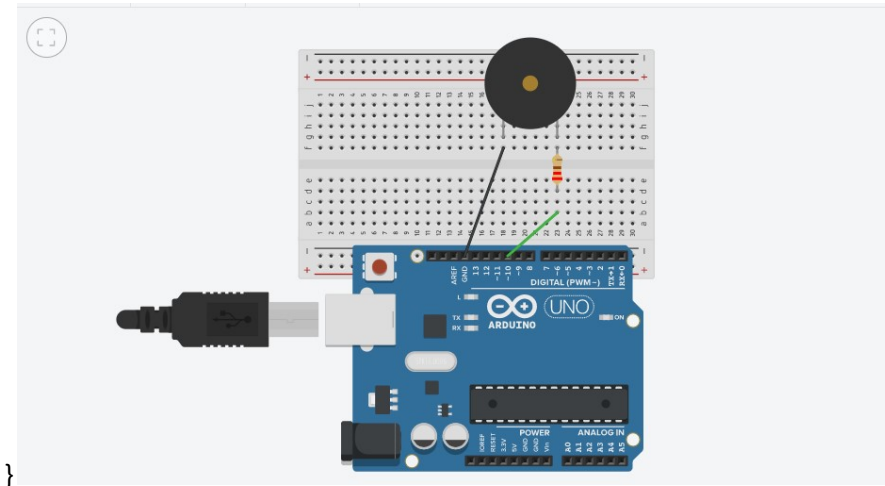
    for (int i = 0; i < 8; i++) {
```

```
tone(PIN, melody[i], tempo);
```

```
delay(1000);
```

```
}
```

```
delay(2000);
```



Tre musichette a scelta con il monitor seriale

```
int uscita = 10;
```

```
//suoneria Tetris
```

```
int melodia[] = {660, 660, 660, 510, 660, 770, 380, 510, 380, 320, 440, 480, 450, 430, 380, 660, 760, 860, 700, 760, 660, 520, 580, 480, 510, 380, 320, 440, 480, 450, 430, 380, 660, 760, 860, 700, 760, 660, 520, 580, 480, 500};
```

```
int durata[] = {100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 80, 100, 100, 100, 100, 80, 50, 100, 80, 50, 80, 80, 80, 80, 100, 100, 100, 100, 80, 100, 100, 100, 100, 80, 50, 100, 80, 50, 80, 80, 80, 80, 100};
```

```
int ritardo[] = {150, 300, 300, 100, 300, 550, 575, 450, 400, 500, 300, 330, 150, 300, 200, 200, 150, 300, 150, 350, 300, 150, 150, 500, 450, 400, 500, 300, 330, 150, 300, 200, 200, 150, 300, 150, 350, 300, 150, 150, 500, 300, 300};
```

```
//suoneria Super Mario
```

```
int melodia_s[] = {659, 493, 523, 587, 523, 493, 440, 440, 523, 659, 587, 523, 493, 523, 587, 659, 523, 440, 440, 440, 493, 523, 587, 698, 880, 783, 698, 659, 523, 659, 587, 523, 493, 493, 523, 587, 659, 523, 440, 440};
```

```
int durata_s[] = {80, 50, 50, 80, 50, 50, 80, 50, 50, 80, 50, 50, 100, 50, 80, 80, 80, 80, 50, 50, 50, 50, 100, 50, 80, 50, 50, 100, 50, 80, 50, 50, 80, 50, 50, 80, 80, 80, 80, 80};
```

```
int ritardo_s[] = {80, 50, 50, 80, 50, 50, 80, 50, 50, 80, 50, 50, 100, 50, 80, 80, 80, 80, 50, 50, 50, 50, 100, 50, 80, 50, 50, 100, 50, 80, 50, 50, 80, 50, 50, 80, 80, 80, 80, 120};
```

```
//suoneria Jingle bells
```

```
int melodia_j[] = {20, 20, 20, 20, 20, 20, 20, 25, 16, 18, 20, 22, 22, 22, 22, 22, 20, 20, 20, 18, 18, 20, 18, 25};
```

```
int durata_j[] = {200, 200, 400, 200, 200, 400, 200, 200, 200, 200, 400, 200, 200, 200, 200, 200, 200, 200, 200, 200, 200, 200, 200, 200, 200, 400, 400};
```

```
int ritardo_j[] = {400, 400, 800, 400, 400, 800, 400, 400, 400, 400, 800, 400, 400, 400, 400, 400, 400, 400, 400, 400, 400, 400, 400, 400, 400, 400, 400, 400, 400, 400, 400, 400, 400, 800, 800};
```

```
//Proseguiamo inizializzando tre contatori, che utilizzeremo dopo, la porta seriale e il display LCD con l'opportuno numero di colonne e righe. Nel nostro esempio, il display è un 16x2.
```

```
void setup() {
```

```
    pinMode(uscita,OUTPUT);
```

```
    Serial.begin(9600);
```

```
}
```

```
//Passiamo quindi al cuore del codice, ovvero il loop().
```

```
void loop() {
```

```
char key = Serial.read(); //ritorna il valore del tasto che è stato premuto
```

```
if(key == '1'){
```

```
    int size = sizeof(melodia_s);
```

```
    for (int thisNote = 0; thisNote < size; thisNote++) {
```

```
        tone(uscita, melodia_s[thisNote], durata_s[thisNote]*2);
```

```
        delay(ritardo_s[thisNote]*4);
```

```
    }
```

```
}
```

```
if(key == '2'){
```

```
    int size = sizeof(melodia);
```

```
    for (int thisNote = 0; thisNote < size; thisNote++) {
```

```
        tone(uscita, melodia[thisNote], durata[thisNote]*2);
```

```
        delay(ritardo[thisNote]);
```

```
    }
```

```
}
```

```
if(key == '3' ){
```

```
    int size = sizeof(melodia_j)/sizeof(int);
```

```
    for (int thisNote = 0; thisNote < size; thisNote++) {
```

```
        tone(uscita, melodia_j[thisNote], durata_j[thisNote]*2);
```



```
delay(ritardo_j[thisNote]);  
  
}}}
```

N.B La scheda Arduino riesce a far funzionare un buzzer alla volta e non due insieme perché i suoi comandi sono sequenziali

Note e frequenze

Nota	Frequenza Hz	Tempo Alto (1/2f)
Do	261	1915
Do#	277	1805
Re	293	1700
Re#	311	1607
Mi	329	1519
Fa	349	1432
Fa#	370	1351
Sol	392	1275
Sol#	415	1204
La	440	1136
La#	466	1072
Si	493	1014