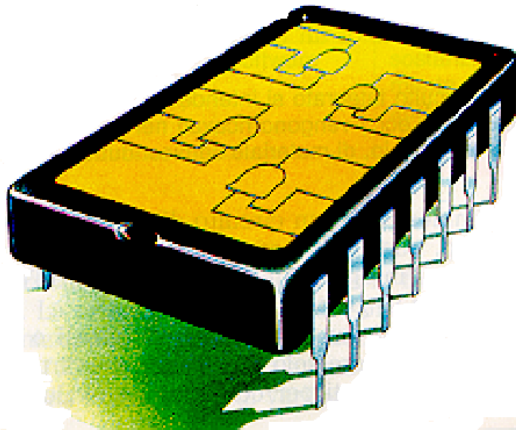


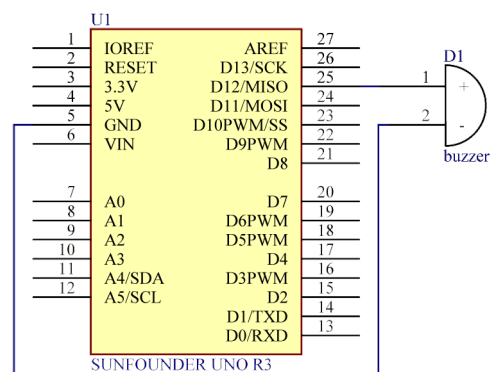
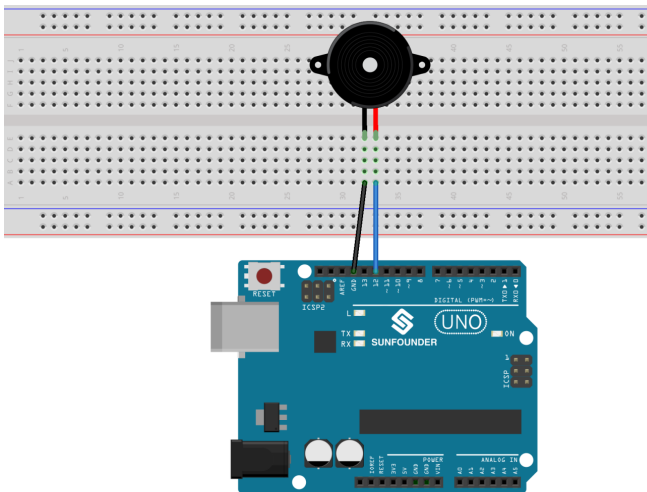
Lez. N. 13 Buzzer attivo



Materiali

Arduino
1 Buzzer attivo
Jumper vari

Tanta pazienza!



Introduzione

Si costruisce un semplice circuito con un cicalino attivo.

È un cicalino elettronico con struttura integrata, i cicalini, che sono alimentati dalla corrente continua, sono ampiamente utilizzati nei computer, stampanti, fotocopiatrici, allarmi, giocattoli elettronici, dispositivi elettronici automobilistici, telefoni, timer e altri prodotti elettronici per dispositivi vocali. I buzzer possono essere classificati come attivi e passivi (vedere la figura seguente). Ruota i perni di due cicalini a faccia in su, e quello con un circuito verde è un cicalino passivo, mentre l'altro con un nastro nero è attivo.



La differenza tra un cicalino attivo e un cicalino passivo è:

Un buzzer attivo ha una sorgente oscillante incorporata, quindi emetterà suoni quando elettrificato. Ma un cicalino passivo non ha tale sorgente, quindi non twitterà se vengono usati segnali DC; invece, è necessario utilizzare onde quadre la cui frequenza è compresa tra 2 K e 5 K per guidarla. Il buzzer attivo è spesso più costoso di quello passivo a causa di più circuiti oscillanti incorporati.

In questo esperimento, usiamo il cicalino attivo.

Procedure sperimentali

Step 1: Costruisci il circuito

Il diagramma schematico

Passaggio 2: Programma

Passaggio 3: compila il codice

Passaggio 4: caricare lo Sketch su Arduino

Ora, dovresti sentire il segnale acustico del cicalino.

Codice

```
int buzzer = 12;//the pin of the active buzzer
void setup()
{
  pinMode(buzzer,OUTPUT);//initialize the buzzer pin as an output
}
void loop()
{
  unsigned char i;
  while(1)
  {
    //output an frequency
    for(i=0;i<80;i++)
    {
      digitalWrite(buzzer,HIGH);
      delay(1);//wait for 1ms
      digitalWrite(buzzer,LOW);
      delay(1);//wait for 1ms
    }
    //output another frequency
    for(i=0;i<100;i++)
    {
      digitalWrite(buzzer,HIGH);
      delay(2);//wait for 2ms
      digitalWrite(buzzer,LOW);
      delay(2);//wait for 2ms
    }
  }
}
```

char	È un tipo di dato e memorizza un carattere (es. "a", "d")
unsigned char i	È un tipo di dato senza segno che codifica numeri da 0 a 255, questo per poter svolgere calcoli aritmetici con questi simboli.
while(1)	while(condizione), il ciclo while esegue ripetutamente un'istruzione o un blocco in cui una condizione è vera.