

Arduino Happy Birthday

Obiettivo: Realizzare un bigliettino di auguri di buon compleanno utilizzando la piattaforma Arduino.

Componenti elettronici:

- Arduino UNO
- Breadboard
- 1 Buzzer

Pre-requisiti:

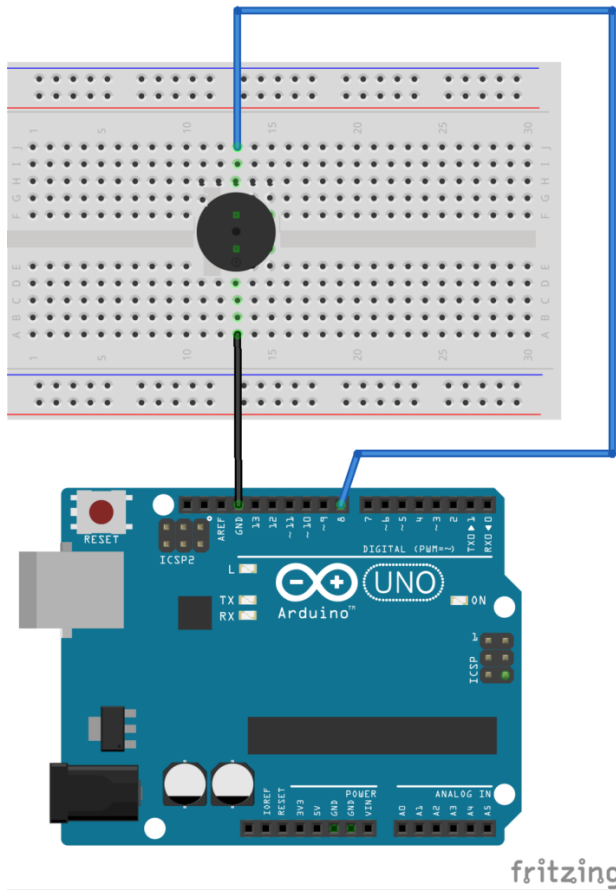
[*Buzzer Passivo*](#)

Teoria: Ogni melodia musicale è composta da note e pause. Se le pause possono essere riprodotte utilizzando la funzione **delay()** di Arduino (già vista negli esempi precedenti), le singole note possono essere facilmente generate grazie all'utilizzo di un buzzer passivo e della funzione **tone()**. Nel dettaglio, l'impiego della funzione **tone** permette di selezionare la frequenza riprodotta dal buzzer e la relativa durata della nota.

Nel caso in questione, l'intera melodia (comprensiva di note, durata delle note e pause, viene salvata in due differenti array (vettori) e riprodotta sequenzialmente come un vero

spartito musicale.

Collegamento Circuitale:



Collegamento Circuitale

Codice:

fritzing

Il Sensore di Parcheggio

Obiettivo: Realizzare un sensore di parcheggio

Componenti elettronici:

- Arduino UNO
- Breadboard
- 1 Sensore Ultrasuoni (e.g., HC-SR04)
- 1 Buzzer

Pre-requisiti:

[*Il Sensore a Ultrasuoni*](#)

[*Buzzer Attivo*](#)

Teoria: Il **sensore di parcheggio** rappresenta una delle moderne tecnologie che permettono al guidatore di un autoveicolo di essere a conoscenza della distanza tra la propria automobile ed un eventuale altro veicolo. Questo permette di gestire in modo facile e rapido l'operazione di parcheggio evitando danneggiamenti o urti agli autoveicoli in questione.

Nel caso specifico, il sensore di parcheggio si basa su un

seniore di prossimità ad ultrasuoni (INPUT) ed un buzzer (OUTPUT) utilizzato come segnalatore acustico.

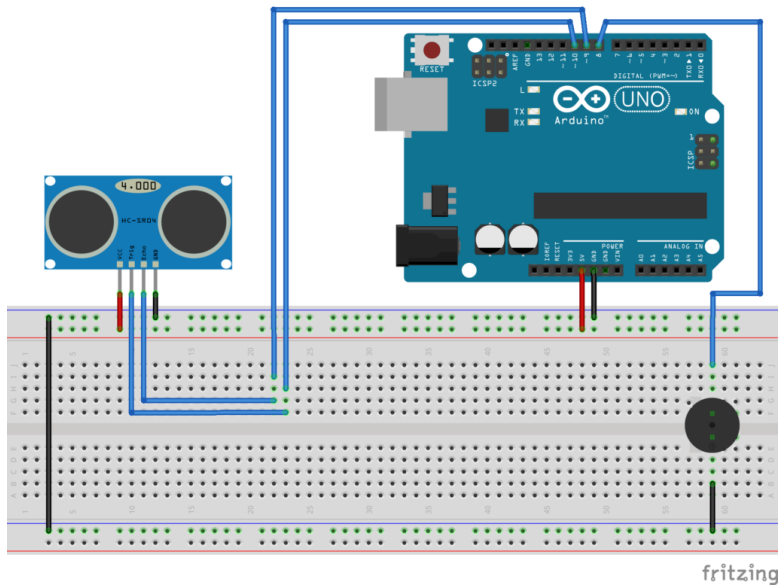
Nel dettaglio, maggiore è la distanza maggiore sarà il ritardo tra una segnalazione acustica e la successiva, analogamente, minore è la distanza minore sarà il ritardo tra una segnalazione acustica e la successiva. In particolare, la relazione ingresso uscita che lega la distanza al ritardo impiegato nella segnalazione acustica è funzione dei seguenti quattro parametri:

- DistanzaMassima: la distanza massima dopo la quale non viene più segnalato un'ostacolo.
- DistanzaMinima: la distanza minima per la quale il buzzer emette un tono continuo
- RitardoMassimo: Il ritardo tra una segnalazione acustica e la successiva nel caso di massima distanza.
- RitardoMinimo: Il ritardo tra una segnalazione acustica e la successiva nel caso di minima distanza.

Questi valori vengono utilizzati al fine di determinare l'equazione fondamentale per il calcolo del ritardo:

$$\text{ritardo} = \text{distanza} * ((\text{RitardoMax} - \text{RitardoMin}) / (\text{DistanzaMax} - \text{DistanzaMin}))$$

Collegamento Circuitale:



Collegamento Circuitale

Codice:

Personalizzazioni: E' possibile inserire un led al fine di avere un segnalatore visivo che indichi la distanza dall'ostacolo.

Buzzer Passivo

Obiettivo: Realizzazione di un semplice segnalatore acustico (basato su un buzzer passivo).

Componenti elettronici:

- Arduino UNO
- Breadboard
- 1 Buzzer passivo

Teoria: Il Buzzer (cicalino) è uno dispositivo elettronico che permette di emettere un suono (bzzz) se correttamente alimentato.

I buzzer si suddividono in:

- Buzzer Attivi: alimentati con una tensione opportuna riproducono un tono ad una frequenza pre-impostata.
- Buzzer Passivi: non emettono alcun suono pre-impostato, ma necessitano di una forma d'onda specifica per fare vibrare la membrana interna. Possono produrre toni differenti in funzione del segnale di alimentazione utilizzato.



- Buzzer Attivo (Funzione DigitalWrite)



- Buzzer Passivo (Funzione Tone)

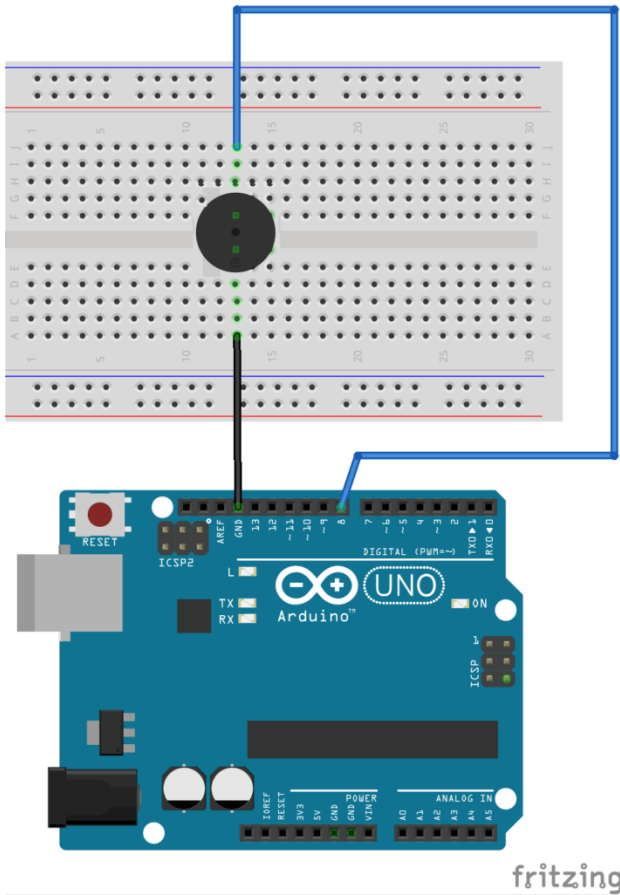
L'utilizzo di un buzzer passivo permette di creare delle melodie personalizzate o differentemente riprodurre piccole canzoni famose. Questo perché il buzzer passivo, a differenza del buzzer attivo, può generare differenti tonalità (note). Le differenti note sono generate utilizzando la funzione tone (nativa nella libreria Arduino). Nello specifico viene riportata in seguito la sintassi di tale funzione.

```
tone(pin, frequenza)
```

```
tone(pin, frequenza, durata)
```

Per interrompere un tono è possibile utilizzare la funzione noTone()

Collegamento Circuitale:



Collegamento Circuitale

Codice:

[crayon-6634d788c7e54719894389/]

Personalizzazioni: E' possibile modificare il comportamento del circuito in questione intervenendo sul valore delle frequenze caratteristiche dei toni e sulle pause introdotte attraverso la funzione delay.

Buzzer Attivo

Obiettivo: Realizzazione di un semplice segnalatore acustico (basato su un buzzer attivo).

Componenti elettronici:

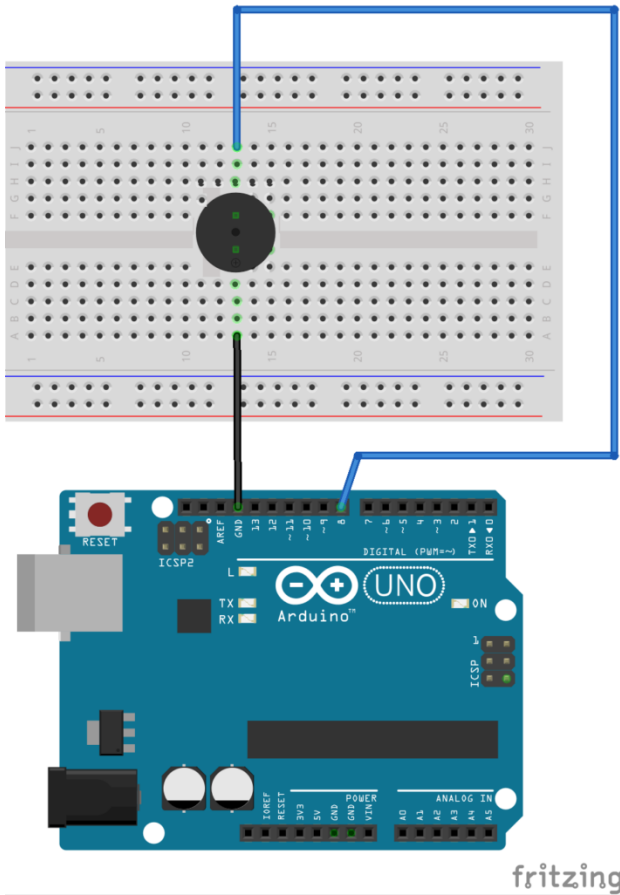
- Arduino UNO
- Breadboard
- 1 Buzzer Attivo

Teoria: Il Buzzer (cicalino) è uno dispositivo elettronico che permette di emettere un suono (bzzz) se correttamente alimentato.

I buzzer si suddividono in:

- Buzzer Attivi: alimentati con una tensione opportuna riproducono un tono ad una frequenza pre-impostata.
- Buzzer Passivi: non emettono alcun suono pre-impostato, ma necessitano di una forma d'onda specifica per fare vibrare la membrana interna. Possono produrre toni differenti in funzione del segnale di alimentazione utilizzato.

Collegamento Circuitale:



Collegamento Circuitale

Codice:

[crayon-6634d788c8cc9452129679/]

Personalizzazioni: E' possibile modificare il comportamento del circuito in questione intervenendo sul valore della variabile *buzzerTime*. Modificando il suo valore infatti cambia la pausa generata tra un tono del buzzer ed il successivo.