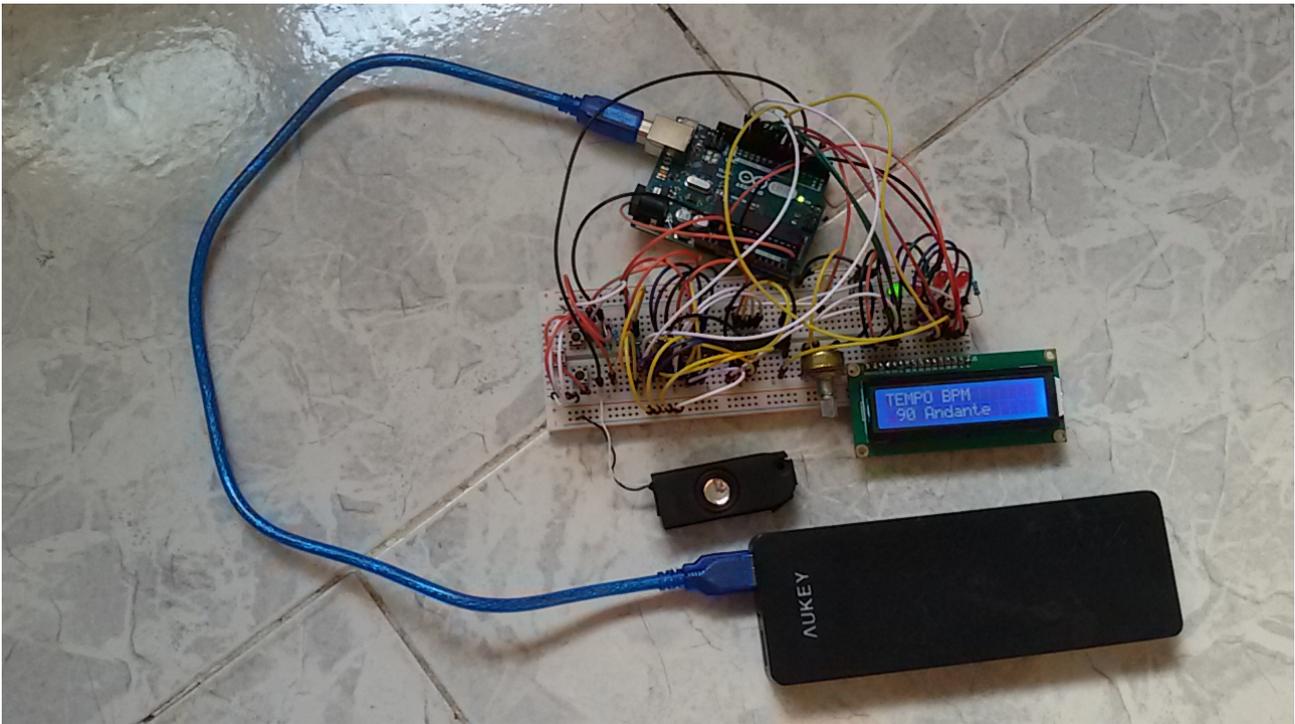


Metronomo

Realizzato da Andrea Corda



Progetto

Il dispositivo realizzato si chiama metronomo e permette di misurare il tempo e fare la scansione ritmica di un brano musicale.

Viene utilizzato dai musicisti durante lo studio di un brano per evitare di accelerare o rallentare durante l'esecuzione, in quanto il battito del metronomo fornisce il tempo di esecuzione.

Il metronomo fornisce un battito costante a un determinato intervallo fornito dall'utilizzatore, stabilito in BPM, battiti per minuto (60 bpm corrisponde a un battito al secondo).

Permette di suddividere i battiti in battute musicali (una battuta musicale corrisponde ad un insieme di note la cui somma ritmica fa un determinato valore come 2/4, 3/4, 4/4).

Il primo battito di ogni battuta deve avere un suono diverso per permettere al musicista che sta suonando di riconoscere l'inizio della battuta successiva e poter verificare se è a tempo.

Per brani un po' più complessi, si rivela necessario avere una suddivisione intermedia di un battito a multipli di due o di tre (nei tempi irregolari), e per questo il metronomo offre la possibilità di suddividere un battito musicale in 2 battiti intermedi (croma di durata 1/8), 4 battiti (semicroma di durata 1/16), 8 battiti (durata 1/32), o tre battiti intermedi (terzina).

Metronomo

Il metronomo permette di modificare le impostazioni dei battiti in tempo reale, tramite 3 pulsanti e il display.

Componenti utilizzati

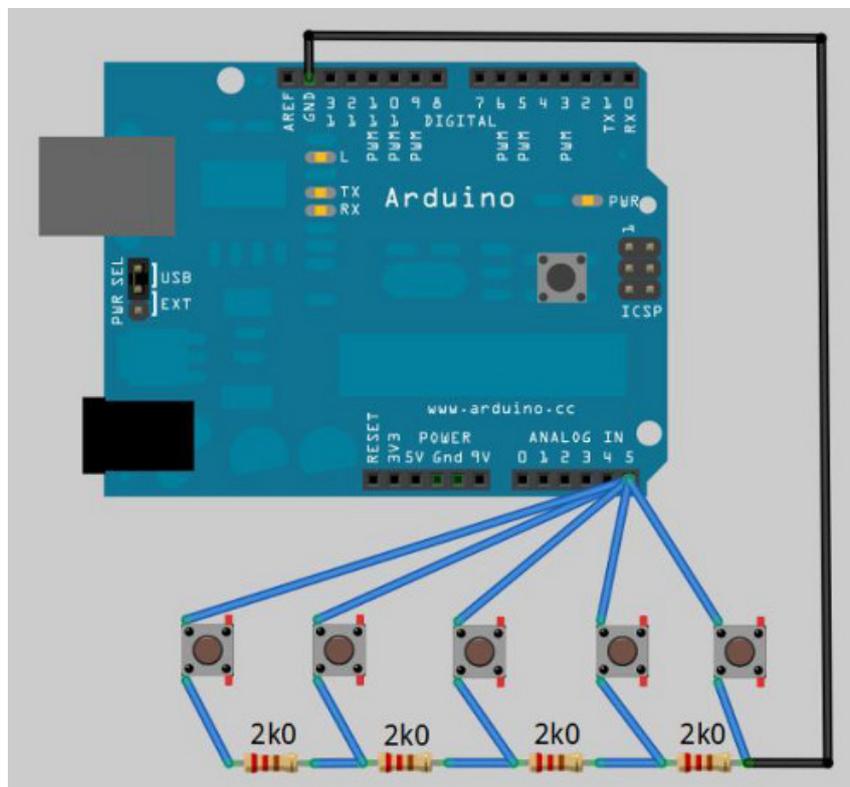
Componenti utilizzati.

- Arduino uno.
- Display LCD con 2 righe e 16 caratteri per riga.
- Potenziometro per regolare il contrasto del display.
- Demultiplexer HD74LS139P con tecnologia TTL.
- Resistenze da 1kOhm.
- 3 Bottoni.
- Breadboard.
- Cavi.

Circuito

Per ricevere in input il segnale dei bottoni, viene utilizzato un analogPin e i 3 bottoni.

Quando viene schiacciato un bottone, Arduino riconosce il bottone schiacciato in base all'intensità della corrente stabilita dal numero delle resistenze nel circuito.



Metronomo

Per gestire i LED vengono utilizzati due pin per i 4 led principali e un pin per un led separato.

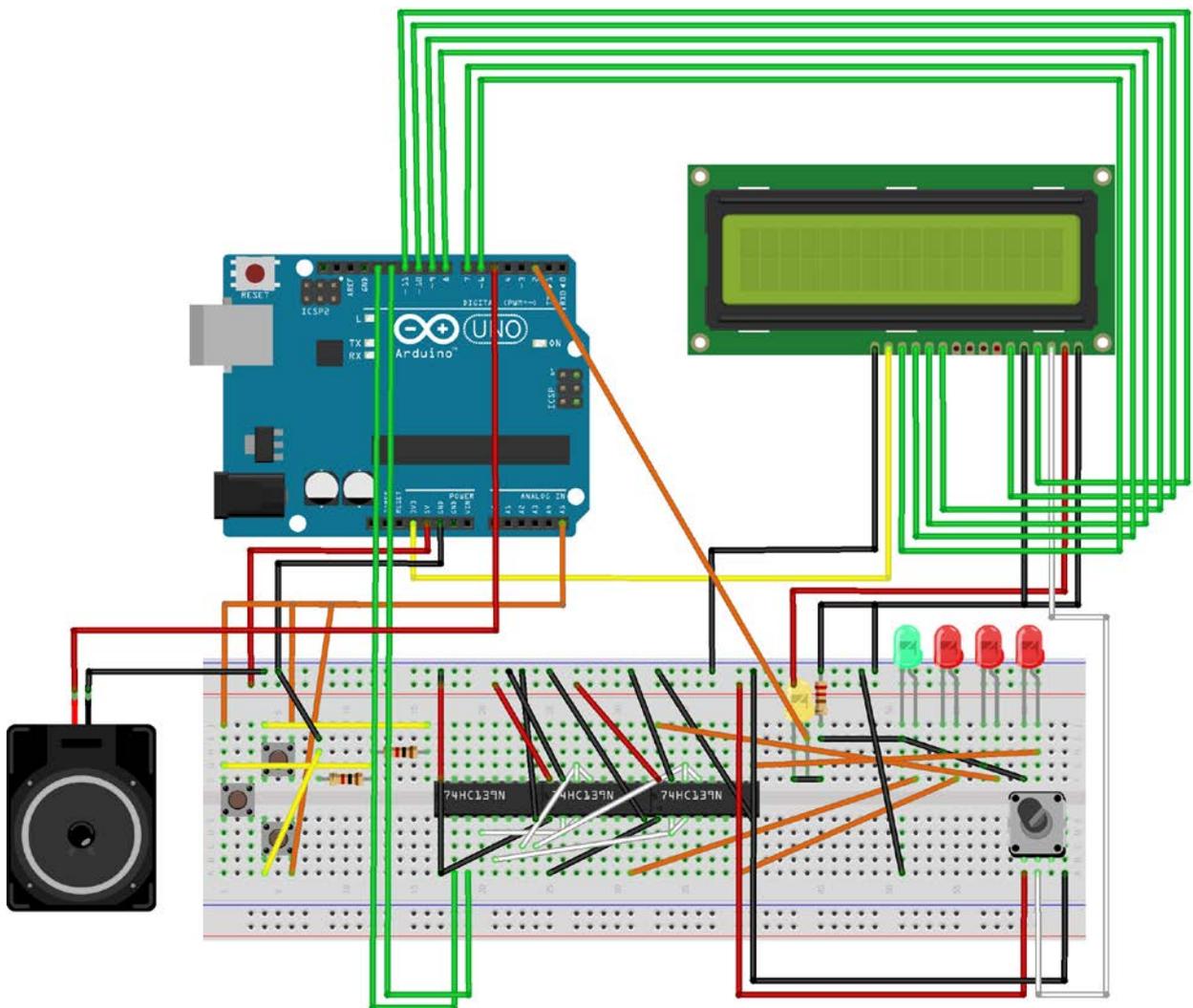
I 4 led principali sono gestiti tramite un circuito integrato HD74LS139P demultiplexer.

Il demultiplexer restituisce l'output negato, per negare il segnale in output si utilizzano altri 2 demultiplexer, che in base all'input ricevuto (1,1) o (0,0), nella porta di output numero 4 restituiscono il segnale negato.

Il 5 led serve per le suddivisioni interne delle battute nei tempi intermedi.

Il potenziometro si occupa di regolare il voltaggio in entrata al led, così da regolare la luminosità del display.

L'altoparlante si occupa di suonare un bip per scandire il tempo.



fritzing

Gestione del sistema di clock dei bottoni

Ogni ciclo di calcolo di Arduino dura meno di 1 millisecondo, perciò tenendo premuto un pulsante in input, il tasto risulta attivo per centinaia di cicli, quindi, per questo avere un solo segnale di input, si memorizza l'ultimo stato.

Quando lo stato precedente risulta LOW e quello attuale risulta HIGH arduino riconosce che il bottone è stato cliccato.

Per supportare la pressione continua del tasto, quando lo stato precedente e lo stato corrente risultano HIGH sale un contatore, ogniqualevolta il contatore arriva a un determinato valore, il software riceve l'input e resetta il contatore.

```
if(pushDown == HIGH && old_pushDown == LOW) {
    faiConfigurazione(-1);
    Serial.write("DOWN\n");
    contPushDown = 1;
}

if(pushDown == HIGH && old_pushDown == HIGH) {
    contPushDown++;
    if(contPushDown % VELOCE == 0) {
        Serial.write("DOWN VELOCE\n");
        faiConfigurazione(-10);
    }
}

//Cambiamenti di stato
old_conf = conf;
old_selected = selected;
old_pushOk = pushOk;
old_pushUp = pushUp;
old_pushDown = pushDown;
clock_lampeggio++;
```

Gestione del calcolo del battito del metronomo

Per il battito del metronomo sfrutto il calcolo dei millisecondi fra ciclo e ciclo. Calcolo i millisecondi necessari in tempo reale per quel determinato battito svolgendo l'operazione $1000 / (\text{BPM}/60)$ e il risultato indica quanti millisecondi devono passare fra ogni battito.

Quando il tempo i vari clock supera questo valore, si resetta il contatore dei millisecondi e si effettua il battito e si cambia lo stato dei led.

Il margine di errore è di 2/3 millisecondi che sono ininfluenti all'orecchio umano.

```
void battito(){
  if(avvio == 0)return;
  int num_battiti = ottieniNumeroBattiti(battiti);
  float battiti_al_secondo = (float)tempo/(float)60;
  long delta = (long)(1000/(battiti_al_secondo*num_battiti));

  if(millis() - mill >= delta){
    Serial.print(delta);
    Serial.write(" ");
    Serial.print(millis() - mill);
    Serial.write(" ");
    Serial.print(battiti_al_secondo);
    Serial.write(" ");
    Serial.print(pulsazione);

    Serial.write("\n");
    mill = millis();

    if(pulsazione != old_pulsazione){
      if(pulsazione == 0){
        digitalWrite(USCITA0,LOW);
        digitalWrite(USCITA1,LOW);
        tone (ALTOPARLANTE, FREQ_BATTUTA, DURATA);
      }else if(pulsazione == 1){
        digitalWrite(USCITA0,HIGH);
        digitalWrite(USCITA1,LOW);
        tone (ALTOPARLANTE, FREQ_BASE, DURATA);
      }
    }
  }
}
```

Utilizzo del display

Il display permette di regolare tutte le impostazioni del metronomo:

1. **Tempo:** misurato in battiti al minuto regolabile da 20 fino a 200, può essere modificato di una unità per un singolo click del bottone, o di 10 unità alla volta tenendo premuto il bottone.
2. **Cadenza tempo:** semiminima (tempo normale), croma (tempo suddiviso in 2 sotto battiti), semicroma (tempo suddiviso in 4 battiti), e biscroma (tempo suddiviso in 8 battiti) e il più importante che è la terzina, che è un tempo irregolare in quanto suddivide il battito in 3 sotto battiti.
3. **Battuta:** regola la battuta da 2/4, 3/4, 4/4.
4. **Metronomo:** Avvia o interrompi il battito.