



# INFORMATICA MUSICALE

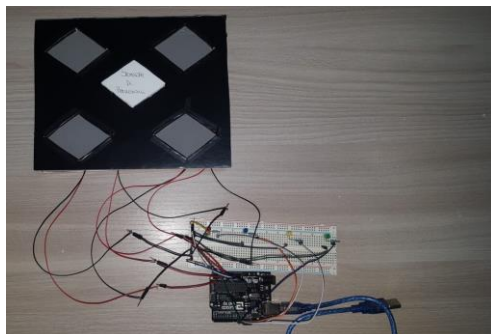
**UNIVERSITA' DEGLI STUDI DI CATANIA**  
**DIPARTIMENTO DI MATEMATICA E INFORMATICA**  
**LAUREA TRIENNALE IN INFORMATICA**  
**A.A. 2018/19**  
**Prof. Filippo L.M. Milotta**

**ID PROGETTO:** 13

**TITOLO PROGETTO:** Sensore di Percussioni

**AUTORE 1:** Baldacchino Irene

**AUTORE 2:** Gaglio Pierluigi



Il progetto consiste nella realizzazione mediante Arduino di un sistema di percussioni, ovvero la simulazione di un ***Pad Elettronico***.

## Indice

<b>1. Obiettivi del progetto .....</b>	<b>2</b>
<b>2. Metodo Proposto / Riferimenti Bibliografici .....</b>	<b>3</b>
<b>3. Risultati Attesi / Argomenti Teorici Trattati .....</b>	<b>4</b>

## 1. Obiettivi del progetto

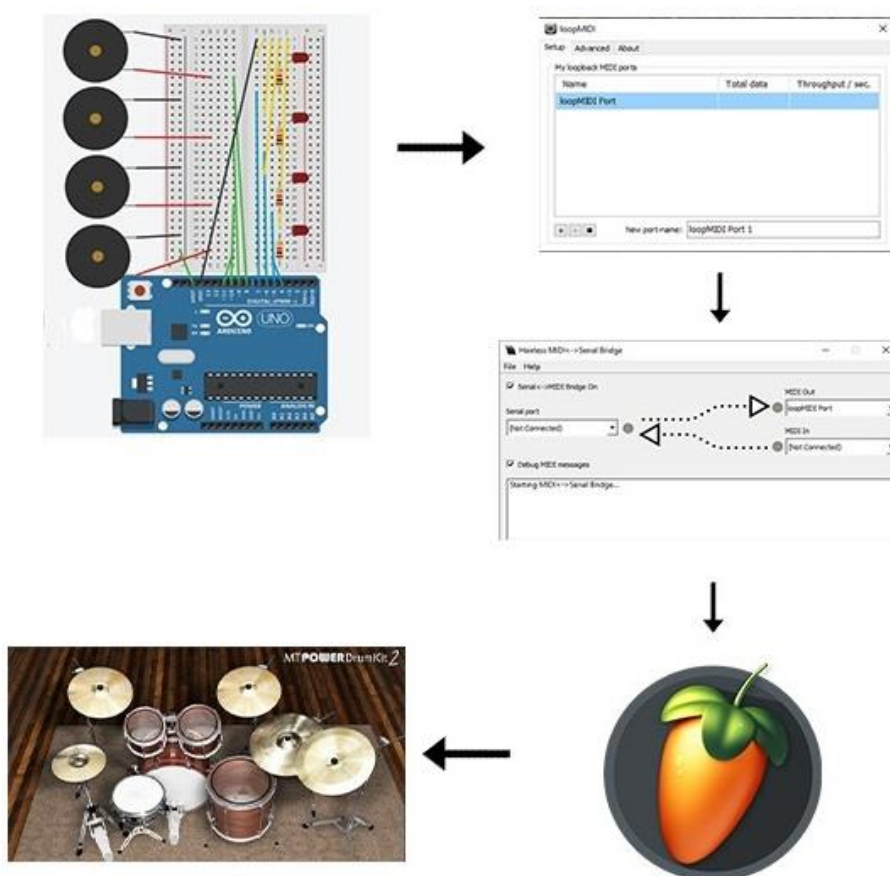
L'obiettivo principale che ci siamo proposti è quello di riprodurre nel modo più affidabile possibile il **"groove"** di una batteria. Il tutto viene realizzato mediante Arduino, il quale mediante dei piezi elettrici collegati ad esso permettono di realizzare il suono dei rispettivi componenti:

1. Hit-hat
2. Snare
3. Crash
4. Bass

Il **piezo elettrico** emana un segnale analogico ad Arduino, il quale a sua volta converte il segnale in formato digitale. Quest'ultimo viene indirizzato al software **"Hairless MIDI"** in grado di aprire una comunicazione tra la porta seriale di Arduino e una porta **"virtuale"** MIDI creata tramite il software **"Loop MIDI port"**. La porta MIDI è necessaria affinché **FL Studio** sia in grado di ricevere i messaggi MIDI che vengono inviati al plugin VST **"Power-Drum-Kit"** il quale dopo averli decodificati genera il suono associato al canale specificato.

Inoltre ad ogni componente viene associato un **led** in grado di mantenersi acceso prolungando il tocco sul piezo.

Mediante FL Studio si è anche in grado di registrare il **groove** prodotto.



## 2. Metodo Proposto



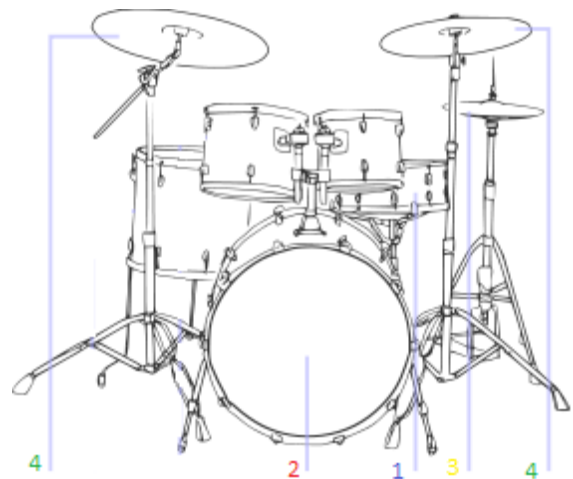
I **Pad Elettronici** sono dispositivi portatili e flessibili a percussione dotati di suoni preimpostati. Essi presentano una varietà di controlli interattivi per espandere e sviluppare l'esperienza di *drumming* e stanno diventando sempre più popolari tra principianti e musicisti professionisti.

Lista dei componenti e software utilizzati:

- Arduino Uno
- 4x Piezi elettrici
- 4x Led (Blu, Rosso, Verde, Giallo)
- 4x Resistenze da 330 ohm
- Connettori e Breadbord
- Loop MIDI port
- Hairless MIDI
- FL Studio 20
- VST Power-Drum-Kit
- Materiale utile per la realizzazione fisica

I vari componenti del Pad sono associati ad uno specifico canale in grado di definire la tipologia di suono desiderato. Nel plugin VST utilizzato i nostri componenti corrispondono ai seguenti canali e colori dei led:

- |            |   |          |   |        |
|------------|---|----------|---|--------|
| 1. Snare   | → | Canale 1 | → | Blu    |
| 2. Bass    | → | Canale 2 | → | Rosso  |
| 3. Hit-hat | → | Canale 3 | → | Giallo |
| 4. Crash   | → | Canale 4 | → | Verde  |



### 3. Risultati Ottenuti

L'elemento principale del codice è l'utilizzo della libreria **<MIDI.h>**.

Per l'inizializzazione è stata utilizzata una porta seriale:

```
Serial.begin(115200);
```

adatta per la gestione dei *messaggi* MIDI.

Inoltre sono stati inizializzati i Pin di Arduino tramite funzione

```
PinMode(pin, stato);
```

con **stato OUTPUT**.

Ad ogni singolo suono è stato associato un blocco

```
if(vibrazione_n >= soglia)
```

dove:

- **vibrazione\_n** è il ritorno della funzione **"analogRead()"**, in grado di leggere il valore di uno specifico pin analogico.
- **n** indica il numero del piezo utilizzato
- **soglia** è il valore minimo affinché possa essere emanato il suono (ovvero l'intensità del tocco).

All'interno dei vari blocchi if si trovano le seguenti funzioni:

1. **MIDI.sendNoteOn(Numero della nota, velocity, canale)**
2. **MIDI.sendNoteOff(Numero della nota, velocity, canale)**

Ove i parametri descritti rappresentano:

- **Numero della nota**: byte identificativi del suono scelto
- **Velocity**: caratterizza la velocità di completamento della fase di inizio e fine del suono
- **Canale**: canale di uscita

Si trovano in contemporanea anche le funzioni:

3. **digitalWrite(pin\_led, HIGH);**
4. **digitalWrite(pin\_led, LOW);**

Dove:

- **pin\_led** indica il pin inizializzato in uscita (tramite pinMode)
- **HIGH** e **LOW** indicano gli stati di **ON/OFF**

**In conclusione:**

Lo scopo finale del progetto è stato raggiunto.

L'unico problema riscontrato è relativo alla fisicità della base costruita e del piezo elettrico (in quanto "scauso") che riproducono in base al colpo dato un suono duplicato.

Ovviamente è una versione ridotta di quella che potrebbe estendersi in un sistema di percussioni complesso.