SOMMARIO PROVVISORIO

# Lato Elettrico-Elettronico:

1. **Presentazione del progetto**
2. **Gestione dei pads**
   1. **Regolazione del segnale**
      1. **Trasduttore piezoelettrico**
      2. **Diodo di Zener**
   2. **Multiplexing degli ingressi**
      1. **Circuito integrato: 4051 Analog Mux/Demux**
   3. Studio e montaggio dell’hardware
3. **Gestione del display**
4. **Gestione dei bottoni**
5. **MIDI output**
6. Audio output
   1. Conversione del segnale
   2. Filtraggio e amplificazione
      1. Filtro passivo: filtro di Butterworth
      2. Circuito integrato: LM386 Low Power Amplifier

# Presentazione del progetto

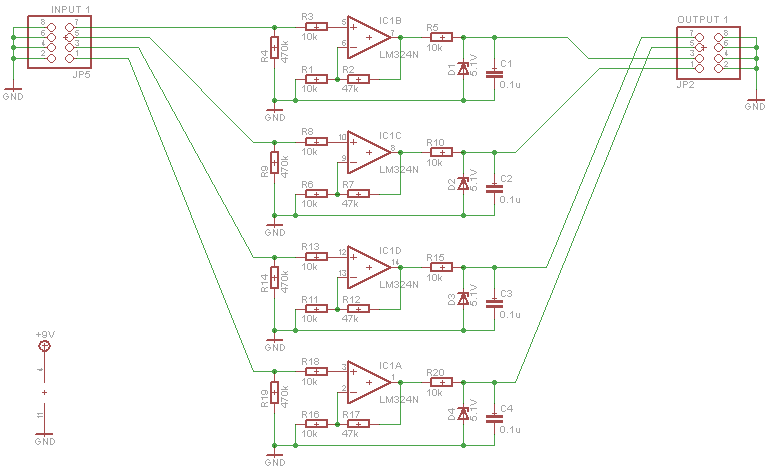
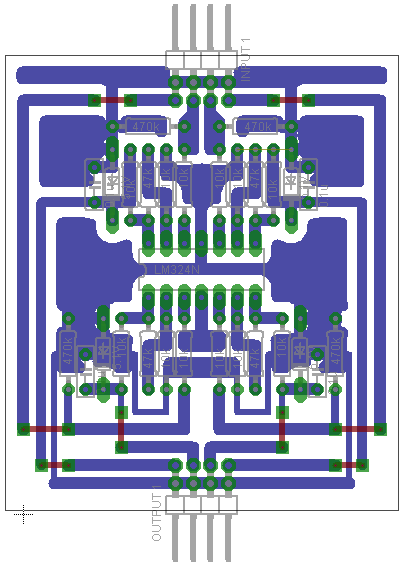
# Gestione dei pads

Una delle parti fondamentali, nonché uno dei primi problemi nel quale si è incappati nel corso della progettazione, è la gestione dei pad, ovvero i cuscinetti semirigidi che verranno suonati. Arduino infatti possiede un numero limitato di ingressi analogici (6 pin), e sfruttarli tutti per la ricezione dei pad non sembra una buona scelta né da un punto di vista progettuale (vista l’impossibilità di sfruttarli in seguito per eventuali upgrade) né pratico (6 pad non sono sufficienti per una drum machine). Si è quindi deciso di convogliare i segnali provenienti dai pad in un multiplexer, occupando solo 4 pin, di cui 3 I/O digitali e uno ingresso analogico. Inoltre, onde evitare escursioni di tensione pericolose o ingestibili, è stato sviluppato un apposito circuito.

## Regolazione del segnale

Nell’ambito del “fai-da-te” i sensori piezoelettrici sono ampiamente usati per rilevare svariati tipi di sollecitazioni meccaniche. Il segnale da essi prodotto però, a causa delle sue caratteristiche di instabilità ed incompatibilità con il range supportato dagli ADC, necessita di un’opportuna elaborazione al fine di renderlo fruibile da Arduino. È stato quindi ideato il seguente schema logico, che raccoglie tutte le fasi del processo e le relative scelte implementative:

Dallo schema logico è stato infine modellato lo schema circuitale, da cui si è ottenuto il seguente circuito stampato:

### Trasduttore piezoelettrico

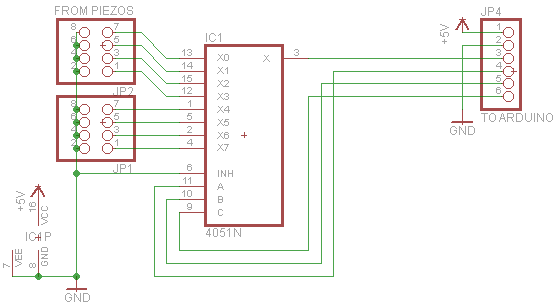
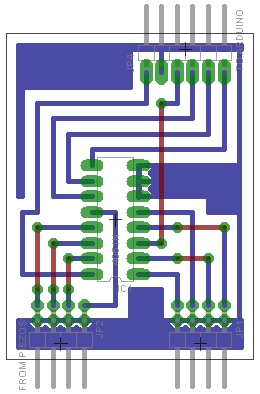
L’effetto piezoelettrico è un fenomeno scoperto da P. Curie e caratterizza alcuni tipi di cristalli (quarzo, tormalina, galena) e ceramiche, che sprigionano una differenza di potenziale quando sottoposti a forze meccaniche. I trasduttori piezoelettrici sono dei piccoli dischi metallici separati una speciale pellicola piezoelettrica che permette la generazione di una tensione proporzionale allo stimolo ricevuto. Le applicazioni di questi elementi sono centinaia: accendigas da cucina, rilevatori sismici, rudimentali altoparlanti, orologi, microfoni, segnalatori acustici, solo per citarne alcune. L’alta temperatura di lavoro (oltre 1000°C per gli elementi al fosfato di gallio), l’ampia banda passante e il costo relativamente basso hanno permesso ai trasduttori e sensori piezoelettrici di imporsi come standard nel mercato. I modelli usati nel progetto sono dei comunissimi “buzzer” dal diametro di 2,8 cm, generalmente usati come segnalatori acustici negli elettrodomestici.

### Diodo di Zener

Si tratta di un particolare tipo di diodo che sfrutta l’effetto Zener, un fenomeno che porta una giunzione P-N percorsa da un’elevata tensione inversa al breakdown, ovvero un repentino aumento di corrente inversa. L’elevato drogaggio cui sono sottoposti i cosiddetti diodi di Zener permette loro di condurre in zona di breakdown, senza che questo precluda il loro funzionamento, come avviene per dei comuni diodi. Questa caratteristica li rende molto apprezzati come stabilizzatori di tensione. Il parametro fondamentale nella scelta di questi diodi è la soglia di zener VZ, che stabilisce la tensione inversa oltre la quale essi tornano a condurre.

## Multiplexing degli ingressi

Avendo scelto di sfruttare uno dei 6 pin analogici per la gestione dei bottini e di dotare la drum machine di 8 pad, si è ritenuto necessario sfruttare un multiplexer esterno, al fine di incanalare tutti i segnali verso un’unica porta analogica. È stato utilizzato il circuito integrato 4051, in grado di multiplex are/demultiplexare fino a 8 canali analogici. Il chip è stato montato su un’apposita scheda, che permette una gestione più agevole de pin. Gli 8 ingressi sono stati collegati alle uscite del circuito dei pad, mentre l’uscita è stata collegata con uno dei 6 pin analogici di Arduino. Il multiplexer è controllato attraverso 3 ingressi, collegati a e controllati da 3 porte analogiche del microcontrollore. Di seguito, lo schema e il circuito stampato della scheda contenente il multiplexer:

### Circuito integrato: 4051 Analog Mux/Demux

L’integrato 4051 è un multiplexer/demultiplexer analogico a 8 canali. Permette di multiplare segnali analogici (quindi continui nel tempo) ampi fino a 15VPP controllando quale ingresso/uscita attivare attraverso 3 ingressi digitali che si attivano con segnali nel range 3-15V. I tre valori inviati a tali ingressi rappresentano il numero (codificato in binario) dell’ingresso/uscita X0-7 che si vuole andare a selezionare. Altre caratteristiche sono: bassa impedenza di uscita (80Ω) e basso assorbimento di corrente sulle alimentazioni e sugli ingressi.

# Gestione del display

Volendo dotare la drum machine di un display, utile per avere sempre sott’occhio le informazioni più importanti e per eseguire una configurazione visuale, si è deciso di valutare un compromesso tra costi e funzionalità. Sono state quindi prese in esame le seguenti caratteristiche:

* Tipo di grafica: grafico, a matrice di caratteri, a 7 segmenti
* Tipo di bus: seriale, I2C, parallelo
* Formato: 20×4, 16×4, 16×2…

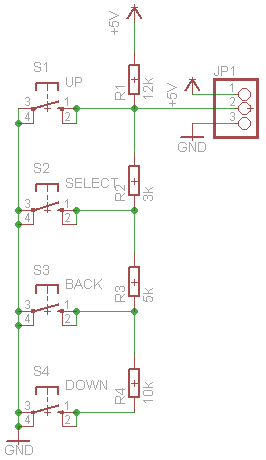
Dopo aver attentamente valutato queste caratteristiche, anche in relazione alla facilità d’uso con Arduino, si è deciso di usare un display LCD a matrice, con 4 righe e 20 colonne di caratteri, comunicante con il microcontrollore attraverso la classica comunicazione parallela prevista dallo standard HD44780. Per l’uso del display è stato necessario il collegamento ai seguenti 6 pin di Arduino:

* 12: pin *RS* del display
* 11: pin *enable* del display
* 2-5: *data* pin del display

Per risparmiare risorse in Arduino, è stato omesso il collegamento del pin *RW* del display e delle 4 *data lines* facoltative. È stato però necessario alimentare il display, e collegarvi un trimmer per regolare il contrasto.

# Gestione dei bottoni

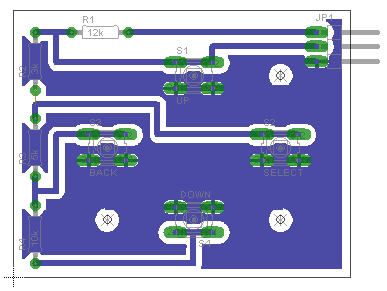
Ad ausilio della navigazione tramite display, si è deciso di inserire una pulsantiera composta di tre bottoni: ⇧, ⇩, ✓ e ⇦ (rispettivamente, scorri su, scorri giù, seleziona e indietro). Avendo la necessità di riservare la quasi totalità dei pin digitali ad altre funzioni si è deciso di adottare un unico ingresso analogico per gestire tutti quattro bottoni. Nel fare ciò è stato sfruttato il principio del partitore di tensione: i terminali degli switch, opportunamente collegati, al momento della pressione bypassano la serie di resistenze (connesse tra 5V e la massa) che li precedono, premettendo di ottenere determinati valori di cadute di tensione, ovvero di stati, rilevabili dall’ingresso analogico. Nel cercare di facilitare il lavoro di programmazione, sono stati adottati diversi valori per le resistenze: dopo una breve simulazione computerizzata è stato possibile vedere come l’uso di resistenze dal valore sempre più grande (seguendo una sorta di andamento geometrico) rendesse più facile la discriminazione dei diversi stati. Di seguito, lo schema del circuito sopra descritto e una tabella riassuntiva con i principali dati ottenuti:



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Bottone | Tensione  nominale  ΔVN (V) | Tensione  misurata  ΔV (V) | Valore  rilevato  *val* | Valore  software  *(val/100)* |
| *idle*  nessuno | 5 | 4.95 | 1023 | 10 |
| ⇧  scorri su | 0 | 0.00 | 0 | 0 |
| ⇩  scorri giù | 3 | 3.01 | 617 | 6 |
| ✓  seleziona | 1 | 1.01 | 204 | 2 |
| ⇦  indietro | 2 | 2.03 | 413 | 4 |

**NOTA:** Nella colonna “Valore software” è stato appuntato il valore usato dal software per riconoscere lo stato di ogni bottone più lo “stato di idle” (nessun bottone premuto). Avendo costatato che i valori rilevati dall’ingresso analogico di Arduino non erano costanti, ma oscillavano di qualche unità, si è ritenuto opportuno aggirare queste oscillazioni avvalendosi di una divisione intera per 100, in quanto l’ordine di grandezza dei dati rilevati si aggirava tra le centinaia.

Si è quindi ricavato un layout definitivo per i componenti, che ha permesso di trasferire il tutto su PCB:



# MIDI output

Tra le diverse funzionalità della drum machine, l’uscita midi è sicuramente la più utile ed interessante, nonché tra le più facili a livello di implementazione hardware. Prevede infatti un collegamento attraverso current-loop con corrente di circa 20mA, instaurato tra due cavi (trasmissione dati e massa, in quanto il dispositivo non prevede l’implementazione di una porta MIDI IN) e una resistenza da 220Ω, collegata tra il pin di output midi di Arduino (nel nostro caso, si tratta del pin TX, usato per la trasmissione seriale). La velocità di trasmissione è di 31.25kbit/s mentre tipo di connettore utilizzato è il DIN a 5 poli, già usato sin dai primi anni ’90 per la connessione di mouse, tastiere o dispositivi audio.