ITT “GUGLIELMO MARCONI” ROVERETO
Anno Scolastico 2012-2013
Classe 5C Informatica

MickSynth

Sintetizzatore FM Digitale

Progetto di Michele Sordo

INDICE

1. Abstract

Lo scopo iniziale di questo progetto è stato quello di realizzare un sintetizzatore musicale riciclando una tastiera vecchia e non più funzionante ma è nei mesi evoluto nella volontà di creare un vero e proprio software di sintetizzazione, capace di lasciare all' utente la possibilità di modificare e modulare i suoni a suo piacimento, al fine di realizzare delle intere tracce musicali.
Il progetto si articola quindi nella costruzione di un hardware capace di comunicare con il sintetizzatore per la riproduzione in real-time di suoni ed effetti.
Al progetto è stato correlato uno studio della composizione e modulazione del suono, della teoria dei sintetizzatori e della loro realizzazione.

2. OpenSource come scelta

2.1 Significato

OpenSource, (termine inglese che significa codice sorgente aperto), in informatica, indica un software i cui autori (più precisamente i detentori dei diritti) ne permettono e favoriscono il libero studio e l'apporto di modifiche da parte di altri programmatori indipendenti. Questo è realizzato mediante l'applicazione di apposite licenze d'uso.
Esistono differenti tipologie di licenza libera in base al maggiore o minore numero di operazioni permesse sulla modifica e distribuzione del programma in oggetto.
La GNU General Public License, comunemente indicata con l'acronimo GNU GPL o semplicemente GPL, è una licenza per software libero, originariamente stesa nel 1989 da Richard Stallman per distribuire i programmi creati nell'ambito del Progetto GNU della Free Software Foundation (FSF). La versione 2.0 di tale licenza è attualmente la licenza di software libero per antonomasia.
Contrapponendosi alle licenze per software proprietario, la GNU GPL assicura all'utente libertà di utilizzo, copia, modifica e distribuzione. La GPL ha incontrato un gran successo fra gli autori di software sin dalla sua creazione, ed è oggi la più diffusa licenza per il software libero, arrivata ormai alla versione 3. Come ogni licenza software, la GPL è un documento legale associato al programma rilasciato sotto tale licenza. Come ogni licenza di software libero, essa concede ai licenziatari il permesso di modificare il programma, di copiarlo e di ridistribuirlo con o senza modifiche, gratuitamente o a pagamento. Chi distribuisce è tenuto a rendere disponibile il codice sorgente del software alle persone che ne hanno ricevuto una copia o, in alternativa, accompagnare il software con una offerta scritta di rendere disponibile il sorgente su richiesta a prezzo nominale. La GPL impone delle condizioni a chi ridistribuisce il software. La base giuridica di questo sta nella licenza specifica: qualora l'utente non accetti le condizioni specificate, essa diventa nulla e quindi non concede alcun permesso. In particolare, essendo il software protetto dalla legge sul diritto d'autore e dalle norme internazionali sul copyright, chi ottiene il software non ha alcun diritto di modifica, copia o ridistribuzione al di fuori di quelle concesse dalla licenza. Rispetto alle altre licenze di software libero, la GPL è classificabile come:

- "persistente" perché impone un vincolo alla redistribuzione: se l'utente distribuisce copie del software, deve farlo secondo i termini della GPL stessa.
- "propagativa" perché definisce nel testo una particolare interpretazione di "codice derivato", tale che in generale l'unione di un programma coperto da GPL con un altro programma coperto da altra licenza può essere distribuita sotto GPL, o in alternativa non essere distribuita affatto.

2.2 Perché Software Libero

La scelta duplice di utilizzare tecnologie OpenSource per realizzare il progetto, e quindi di rilasciare il progetto stesso in OpenSource, porta con sé molteplici vantaggi:

- Sicurezza - Visto l'elevato numero di sviluppatori che hanno accesso al codice, eventuali bug di sicurezza possono essere corretti più velocemente.
- Personalizzazione - il software può essere migliorato dall'intera comunità di sviluppatori, che possono migliorare le caratteristiche esistenti e svilupparne di nuove. Lo stesso può anche essere fatto dal singolo utente, come forma di personalizzazione, se necessario.
- Libertà – le decisioni che riguardano le politiche di sviluppo e il destino del software sono prese dall'intera comunità.
- Flessibilità – il software open si presta in maniera migliore al funzionamento su svariati tipi di hardware, anche di architettura diversa (es. Pc e singleboard computers)
- Interoperabilità – L'adesione a standard aperti per I dati ed I formati, consente di facilitare l'integrazione con sistemi di diversa natura.
- Supporto - è possibile ottenere supporto dalla community attraverso i più svariati canali: documentazione sul web, forum, mailing list, newsgroup e chat. I progetti commerciali offrono spesso anche supporto professionale a pagamento.
- Costi - Molti pacchetti, completi di codice, sono distribuiti in forma gratuita. Inoltre, se il software prevede un costo può' essere provato prima di essere acquistato.

2.3.GitHub

2.3.1 Cos'è GitHub

GitHub è un piattaforma online di social coding basata sul software di controllo di versione “git”. Essa permette l'interazione tra più utenti che condividono i loro repositories contenenti i codici sorgente dei loro progetti. Tramite questa piattaforma ogni iscritto può proporre delle modifiche al codice di terzi e ricevere dei suggerimenti riguardanti i propri. Oltre a ciò ogni progetto può essere clonato nei propri repositories offline e compilato oppure si può decidere di aprire una fork del progetto, continuando lo stesso in parallelo a quello ufficiale. Il codice postato è ovviamente opensource ed il software è totalemnte gratuito (pagando è possibile sbloccare la funzionalità che permette di mantenere online repositories nascosti).

2.3.2 Come funziona Git

Come per molte grandi cose nella vita, Git è iniziato con un po' di distruzione creativa e polemiche di fuoco. Il kernel di Linux è un progetto software open source di portata abbastanza ampia. Per la manutenzione del kernel Linux, per diverso tempo (1991-2002), le modifiche al software venivano passate sotto forma di patch e file d'archivio. Nel 2002, il progetto del kernel di Linux iniziò ad utilizzare un sistema proprietario chiamato DVCS BitKeeper.
Nel 2005, il rapporto tra la comunità che ha sviluppato il kernel Linux e la società commerciale che ha sviluppato BitKeeper si ruppe, e l'uso gratuito di questo strumento fu revocato. Ciò ha indotto la comunità di sviluppo di Linux (e in particolare Linus Torvalds, il creatore di Linux) a sviluppare il proprio strumento, basandosi su alcune delle lezioni apprese durante l'utilizzo di BitKeeper.

Alcuni degli obiettivi di questo sistema sono i seguenti:

- Velocità
- Design semplice
- Forte supporto allo sviluppo non-lineare (migliaia di rami paralleli)
- Completamente distribuito
- Capacità di gestire, in modo efficiente (velocità e dimensione dei dati), grandi progetti come il kernel Linux

Fin dalla sua nascita nel 2005, Git si è evoluto e maturato per essere facile da usare e tuttora mantiene le sue qualità iniziali. E' incredibilmente veloce, è molto efficiente con grandi progetti, ed ha un incredibile sistema di ramificazioni, per lo sviluppo non lineare.

3. Tematiche Trattate

3.1 Il Suono

3.1.1 Cos'è il suono

Possiamo definire il suono come una particolare sensazione percepita dall'organo dell'udito eccitato da un agente esterno. Esso ha origine dal movimento di un corpo dotato di caratteristiche elastiche, e si propaga modificandosi attraverso un mezzo solido, liquido o gassoso, dotato anch'esso di proprietà elastiche.
I suoni che noi udiamo sono di solito molto complessi, tuttavia possiamo oggettivamente notare delle caratteristiche che possono creare una prima distinzione. Ad esempio ci sono suoni che sono molto forti, altri appena percettibili. Notiamo anche che questa caratteristica varia in misura notevole dalla distanza che abbiamo dall'evento che ha generato il suono. Notiamo anche che varia a seconda della qualità dell'evento. Per esempio se percuoto leggermente un tavolino sentirò un piccolo suono, ma se lo percuoto con tutta la forza, il suono sarà senz'altro più forte. Possiamo chiamare questa caratteristica intensità del suono. Notiamo anche che a parità di intensità alcuni suoni sono più acuti ed altri più gravi: denominiamo questa caratteristica altezza del suono. Osserviamo che in genere i suoni più acuti sono generati da oggetti di piccole dimensioni, mentre quelli gravi da oggetti più grossi.

3.1.2 Origine del Suono

Come abbiamo già detto un suono ha origine dalla vibrazione di un corpo elastico: un diapason percosso con un colpo secco, una corda tesa strofinata da un archetto, l'aria contenuta all'interno di un flauto posta in vibrazione soffiandoci contro altra aria. Si può osservare che la vibrazione di questi corpi elastici ha un andamento ondulatorio, ossia descrive nel tempo un movimento che può essere rappresentato graficamente con un onda. Tale onda sarà caratterizzata, a seconda della natura, delle dimensioni, e dello stato di tensione del corpo elastico che entra in vibrazione, nonché dalla causa che origina tale vibrazione, da alcune grandezze che ne definiscono l'andamento: innanzitutto la frequenza, ovviamente espressa in hertz. Si nota che tale grandezza definisce l'altezza del suono che abbiamo prima osservato come parametro oggettivo: in particolare con l'aumentare della frequenza un suono diviene più acuto, col diminuire diviene più grave. Vi è poi da considerare l'ampiezza dell'oscillazione che determina l'intensità del suono. Più difficile è stabilire una unità di misura utile per le nostre considerazioni future. Indubbiamente l'ampiezza dell'oscillazione è proporzionale all'energia dell'onda, quindi, nell'unità di tempo alla potenza, tuttavia esprimere in watt l'ampiezza di un'onda, sebbene fisicamente corretto, ci porterebbe ad avere una grandezza che non comunica efficacemente la percezione dell'intensità del suono. Tale intensità è da noi

percepita secondo un andamento non lineare ma logaritmico. Per questi motivi è stato introdotto il decibel (dB).

3.1.3 Il Decibel

Il decibel è forse la misura più usata in acustica, esso esprime secondo una scala logaritmica in base 10, il rapporto fra due grandezze omogenee.
Il decibel è 10 volte il logaritmo in base 10 del rapporto di due grandezze omogenee.
Supponiamo di voler esprimere in decibel la differenza fra due potenze: w1 e w2.

$$dB = 10\log\left(\frac{w1}{w2}\right)$$

Notiamo quindi che affinché la precedente espressione abbia senso occorre che esistano entrambe le grandezze w1 e w2 ed in particolare che sia w2 ≠ 0. Notiamo anche che non ha importanza di che tipo siano le grandezze w1 e w2 , occorre solo che siano omogenee, ossia dello stesso tipo.
D'altra parte possiamo immaginare per il decibel anche valori negativi. Questo si ha quando fra le grandezze di cui vogliamo esprimere il rapporto, quella che appare al numeratore è minore di quella che appare al denominatore.

Il decibel si può quindi considerare una misura relativa, che non dipende infatti da una sola grandezza ma dal rapporto di due grandezze omogenee. Tuttavia ci sono certi casi in cui tale misura si può immaginare assoluta. Questo avviene quando al denominatore si pone una misura stabilita per convenzione internazionale. Un esempio di misura assoluta in decibel è quella dell'intensità sonora, che convenzionalmente è espressa in riferimento alla soglia di udibilità:

Livello di intensità dB	Condizione ambientale	Effetto sull'uomo
140	Soglia del dolore	Lesioni dell'orecchio nel caso di ascolto prolungato
120	Clacson potente, a un metro	
110	Picchi d'intensità di una grande orchestra	Zona pericolosa per l'orecchio
100	Interno della metropolitana	
90	Picchi di intensità di un pianoforte	
80	Via a circolazione media	Zona di fatica
75	Voce forte, a un metro	
70	Conversazione normale, a un metro	
60	Ufficio commerciale	
50	Salotto calmo	Zona di riposo (giorno)
40	Biblioteca	
30	Camera da letto molto calma (notte)	Zona di riposo (notte)
20	Studio di radiodiffusione	
0	Soglia di udibilità	

3.1.4 Composizione - Armoniche

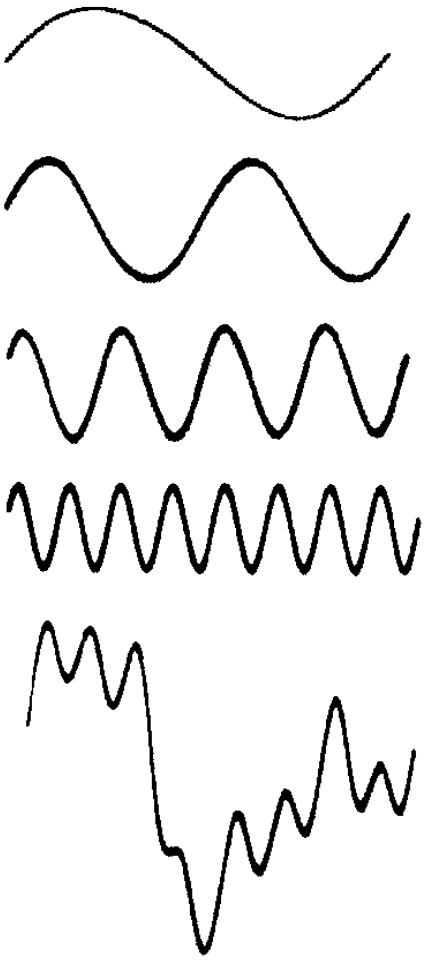
I suoni fin'ora citati sono stati rappresentati rispetto al tempo con una sinusoide. In realtà si tratta di un caso molto particolare in quanto i suoni hanno un andamento di solito più complesso. Tuttavia non è sbagliato limitare la nostra attenzione al suono sinusoidale, infatti a qualsiasi suono complesso, potendosi comunque rappresentare con una funzione periodica, possiamo applicare il teorema di Fourier:

- Qualsiasi funzione periodica di frequenza υ può essere decomposta in un solo modo nella somma di funzioni sinusoidali semplici di frequenza multipla di υ, le cui ampiezze sono univocamente determinate.

Quindi ogni funzione suono complesso che abbia frequenza υ può essere considerato la somma di tanti suoni sinusoidali con frequenza multipla di υ, ed ampiezza univocamente determinata. Denominiamo tali suoni sinusoidali armoniche del suono complesso, per cui possiamo dire:

- Tutti i suoni complessi della medesima frequenza differiscono fra loro unicamente per l'ampiezza delle armoniche.

La frequenza del suono complesso, che è poi la più bassa fra le armoniche, prende il nome di fondamentale. Le armoniche superiori prendono il nome di seconda armonica, terza armonica, quarta armonica ecc. a seconda se la frequenza è 2, 3, 4.... volte la fondamentale.

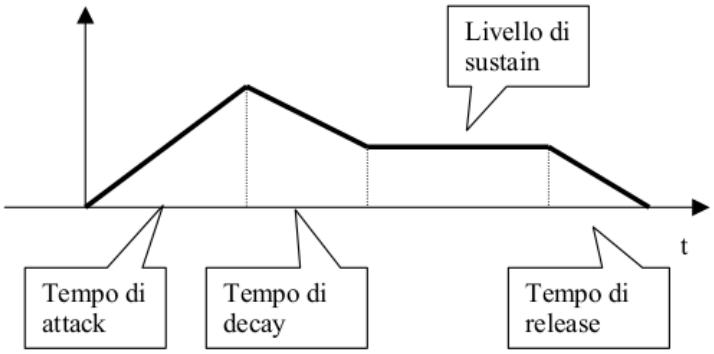


Nel disegno a lato possiamo vedere la rappresentazione della fondamentale di un suono complesso, della seconda armonica, della quarta armonica e dell'ottava armonica. L'ultimo disegno in basso rappresenta il suono complesso risultante dalla somma di queste armoniche.

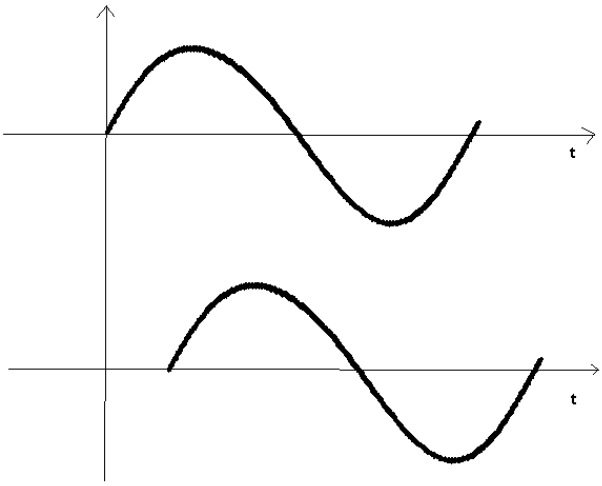
Definiamo onda quadra, quell'oscillazione che contiene tutte le (infinite) armoniche con la stessa intensità. Possiamo quindi osservare che più la rappresentazione di un suono tende all'onda quadra, più questo suono aumenta l'intensità delle sue armoniche superiori. Questo è il motivo per cui quando si utilizza un amplificatore di potenza ad una potenza che eccede quella massima, e il suono tende a squadrarsi (clipping) , i primi componenti a soffrire sono i tweeter. L'intensità delle armoniche determina il timbro di un suono, ossia la differenza che esiste fra i suoni emessi dai vari strumenti musicali. Con l'equalizzazione interveniamo sul timbro degli strumenti variando proprio l'intensità delle armoniche. I suoni naturali hanno in genere un'intensità delle armoniche che decresce col crescere del loro ordine. Si è provato a generare dei suoni sintetici che hanno le intensità delle armoniche crescenti, ottenendo risultati interessanti. Esistono poi dei generatori di subarmoniche che aggiungono al suono una armonica con frequenza pari alla metà della fondamentale, col risultato di aggiungere corposità al suono.
La composizione delle armoniche può non essere costante nel tempo in cui il suono è emesso. Un suono può ad esempio iniziare con poche armoniche, arrivare ad averne una certa quantità, diminuirne il numero al momento che il suono si estingue. Questo fenomeno prende il nome di inviluppo del suono (in inglese contour) riferito alle armoniche. L'inviluppo di un suono è di solito definito da quattro grandezze: attack, decay, sustain, release (si puo chiamare inviluppo

ADSR).
Definiamo concettualmente tali grandezze:

- Attack (time): il tempo che un suono impiega a raggiungere il livello massimo di armoniche
- Decay (time): il tempo che un suono impiega a raggiungere il livello normale di armoniche
- Sustain (level): il livello normale di armoniche
- Release (time): il tempo che un suono, dopo il termine della sua emissione impiega per annullarsi (e quindi annullare tutte le sue armoniche)

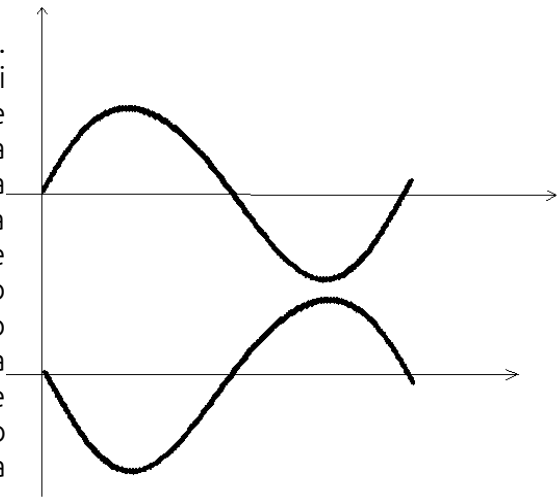


3.1.5 Fase di un Suono



Due suoni possono differire oltre che per la frequenza, l'intensità e la composizione delle armoniche anche per il momento in cui vengono emessi. In genere si parla di fase quando questo tempo è minore del periodo, ossia del tempo necessario a compiere un ciclo completo. La differenza di tempo fra due suoni (figura a lato) dipende dal cosiddetto angolo di fase. La funzione che esprime una rappresentazione sinusoidale è :

$$I = I0 \sin(\omega t + \varphi)$$



La grandezza φ rappresenta appunto l'angolo di fase. Esprimere la fase come un angolo ci permette di prescindere da altre grandezze, tipo la frequenza, che definiscono il nostro suono. Infatti qualunque sia la frequenza, un angolo di fase pari a π/2 corrisponde a 1⁄4 del periodo, un angolo di fase π corrisponde a metà del periodo, un angolo di fase 3/2 π corrisponde a 3⁄4 del periodo, e un angolo di fase di 2 π all'intero periodo. É facilmente osservabile che il suono risultante fra due suoni uguali in tutto fuorché nella fase, è molto diverso dai suoni originari, in particolare nella composizione delle armoniche. Un caso particolarmente significativo si ha quando la

differenza di fase fra i due suoni è pari a π/2. Si dice che i due suoni sono in opposizione di fase, o in controfase. Il suono risultante dalla somma di due suoni in controfase ha intensità nulla.

3.2 Synth

3.2.1 Cos'è un Synth

Il sintetizzatore (abbreviato anche a synth dal termine in inglese) è uno strumento musicale che appartiene alla famiglia degli elettrofoni. È un apparato in grado di generare autonomamente segnali audio, sotto il controllo di un musicista o di un sequencer. Si tratta infatti di uno strumento che può generare imitazioni di strumenti musicali reali o creare suoni ed effetti non esistenti in natura.
Esistono essenzialmente due tipi differenti di sintetizzatori: quelli analogici che creano i suoni attraverso degli oscillatori elettronici e quelli virtuali (generalmente chiamati VST) che utilizzano esclusivamente appositi software per creare e modulare i suoni e si appoggiano sulla scheda audio montata dal singolo computer per quanto riguarda l' output.
Il sintetizzatore è generalmente comandato per mezzo di una tastiera simile a quella del pianoforte, ma non mancano realizzazioni destinate ad essere gestite mediante il fiato, la pressione, le corde di una chitarra o altri tipi di controller come sensori a raggi infrarossi.

3.2.2 Tipi di Sintetizzazione

3.2.2.1 Sintesi Additiva

Secondo gli studi di Fourier (dei quali si è già parlato) è possibile ricreare un suono naturale partendo dalla somma di un certo numero di frequenze fondamentali (segnali sinusoidali) e distribuendole nello spettro sonoro. Tale tecnica, pur permettendo teoricamente di poter riprodurre qualsiasi suono esistente, in realtà è di estrema complessità; infatti, laddove la sintesi sottrattiva agisce su un consistente numero di armoniche, già patrimonio del segnale grezzo originale, qui abbiamo la necessità di controllare un numero elevatissimo di componenti, che molto probabilmente andranno modulate individualmente, per ottenere una risposta convincente all'ascolto. Si tratta pertanto di una tecnica complessa, che non ha incontrato molto successo nella produzione industriale di strumenti musicali elettronici (è però interessante nell'ambito della ricerca).

3.2.2.2 Sintesi Sottrattiva

Da un generatore di segnale con elevata produzione di armoniche (ad esempio onda quadra, onda triangolare, dente di sega, etc) si interviene con un sistema di filtri allo scopo di modificare il timbro e quindi la forma d'onda. Esempi di sintesi sottrattiva si possono ritrovare anche negli strumenti musicali tradizionali dove la selezione del timbro è ottenuta in maniera meccanica tramite la cassa armonica come nella chitarra o nel violino. È inoltre possibile variare frequenza del filtro (cut off) ed il fattore di qualità Q (Peak o resonance) tramite opportuni controlli. I filtri possono essere realizzati con tecnologia analogica (reti RC o componenti discreti) oppure nel dominio digitale tramite dsp. Altri parametri fondamentali nella catena di sintesi analogica sono l'inviluppo (ADSR), il controllo del volume (VCA) e gli effetti di vibrato (LFO). I primi sintetizzatori a sintesi sottrattiva erano implementati tramite sistemi modulari analogici dove era possibile interconnettere e controllare ogni modulo a piacimento. In seguito sono stati sviluppati i primi sintetizzatori normalizzati dove l'utente poteva scegliere fra alcune configurazioni di base scelte dal costruttore. Una grande rivoluzione è stata l'implementazione della sintesi sottrattiva nel dominio digitale, dove un semplice DSP può sostituire le funzioni centinaia di moduli analogici.

3.2.2.3 Sintesi FM/PM

Sperimentata da John Chowning presso il centro CCRMA della Stanford University, questa tecnica è divenuta di grande popolarità tramite una fortunata serie di sintetizzatori prodotta dalla giapponese Yamaha, a partire dal 1982. Il concetto parte dalla possibilità di modulare in banda audio la frequenza di una fondamentale mediante un altro segnale (puro cioè sinusoidale nelle prime versioni commerciali Yamaha, di complessità differente nelle implementazioni successive): sotto questa azione, il segnale modulato modifica la propria fase in funzione del segnale modulante, perdendo così la caratteristica di segnale puro e arricchendosi di nuove armoniche; il risultato è estremamente variabile in funzione del rapporto aritmetico fra le frequenze e dell'ampiezza del segnale modulante: maggiore è l'ampiezza del segnale modulante, maggiore sarà la distribuzione di armoniche nel segnale fondamentale. Ciò permette di ottenere timbriche di eccezionale verosimiglianza, soprattutto operando con combinazioni di più generatori (nel caso di Yamaha, fino a sei nel sintetizzatore DX-1) e operando sullo schema di combinazione dei generatori (detti operatori), sull'inviluppo di ampiezza e di frequenza degli stessi.
Questo procedimento è molto più vicino alla generazione naturale del suono di quanto si immagini; nel momento in cui il suono viene prodotto, ad esempio con una chitarra acustica, la corda viene spostata dal proprio stato di quiete e rilasciata: ciò provoca un'oscillazione della corda corrispondente alla sua fondamentale, sommata allo "sforzo" del pizzico. In questo esempio, la fondamentale della corda è l'oscillatore modulato, mentre l'andamento nel tempo della componente del pizzico rappresenta l'oscillatore modulante. L'ampiezza di entrambi degrada con l'andare del tempo, fino al naturale smorzamento del suono, ovvero si delineano due differenti curve di inviluppo per i due generatori. Il suono risultante sarà pertanto diversamente colorato in funzione dell'intensità e della modalità (dita o plettro) del pizzico.
Un esempio ancora più evidente lo si trova nella tecnica slap per il basso elettrico, o nelle varianti del pianoforte acustico (piano elettrificato Yamaha, piano a puntine). Nel caso degli archi, è lo sfregamento dell'archetto sulla corda a creare la componente modulante. Non a caso, la modulazione di frequenza e di fase trova l'eccellenza nella riproduzione proprio di queste categorie di strumenti.

3.2.2.4 Campionamento

Un segnale audio può essere registrato sia in modo analogico, trasferendone la modulazione di ampiezza su un supporto magnetico, sia in modo digitale, effettuando una misurazione a campioni della sua ampiezza nel dominio del tempo, e riportandone i valori in un elaboratore, sotto forma di numeri. Se la frequenza di campionamento è sufficientemente elevata (2 volte la frequenza da campionare per un sistema binario, secondo il teorema di Shannon), il segnale audio può essere così trasferito nella memoria di un elaboratore e successivamente riprodotto, procedendo alla sua rigenerazione inviando, con velocità regolare, i valori misurati ad un Convertitore digitale-analogico (DAC), che provvederà a fornire una tensione analogica in uscita, destinata ad essere amplificata ed ascoltata.
Realizzando un registratore digitale di questo tipo, è possibile ottenere una riproduzione eccezionalmente realistica dei suoni con ampiezza costante (es. organo, strumenti a fiato, archi), un po' più complessa invece la riproduzione dei suoni con un andamento variabile nel tempo, come nel caso del pianoforte, degli strumenti a corde pizzicate e delle percussioni.
Un campionatore, quindi, deve necessariamente fornire le opportune possibilità di agire sul segnale campionato, modificandone la distribuzione delle armoniche e dell'ampiezza nel tempo, in modo da restituirgli quella naturalezza corrispondente allo strumento originario.

3.2.2.5 Sintesi VC

Una certa quantità di parametri da controllare (intonazione, frequenza dei filtri, ampiezza del segnale, andamento nel tempo e così via) richiederebbe un'operatività estremamente articolata per poter suonare un sintetizzatore in tempo reale; infatti, i primi esperimenti erano

realizzati sfruttando la registrazione multitraccia per poter ascoltare insieme il prodotto di più sessioni di registrazione; successivamente, si pensò di poter controllare tutti questi parametri per mezzo di una tensione variabile: il musicista modificava una tensione ruotando una manopola, e una serie di oscillatori poteva variare tutta insieme la propria intonazione. Andando oltre, un secondo problema riguarda la riproduzione delle armoniche nello spettro dell'esecuzione musicale; un sintetizzatore a sintesi additiva, con filtro regolato per produrre un suono di trombone alle basse frequenze, produrrà un suono via via più flebile man mano che ci si avvicina alle note più alte, poiché il filtro ha una frequenza fissa e non si adatta al nuovo segnale che lo attraversa. L'adozione del V.C. ha permesso di realizzare il Keyboard Follow, ovvero una tensione che varia con la posizione del tasto premuto sulla tastiera, che va ad alterare contemporaneamente l'oscillatore controllato in tensione (VCO, Voltage-controlled oscillator) in modo da intonare la nuova nota, ed il filtro (Voltage-controlled filter, VCF) per adattarlo alla nuova frequenza, per estrarne le armoniche nel modo opportuno. Una terza implementazione del sistema a controllo di tensione riguarda l'unità di amplificazione del suono (VCA, Voltage Controlled Amplifier), permettendo di creare suoni con intensità programmabile e variabile nel tempo.

3.2.2.6 Sintesi LFO

Per fornire un effetto piacevole ed espressivo al suono di una forma d'onda filtrata in modo statico da un sistema di filtri, si possono sommare oscillazioni periodiche a frequenza subsonica (es. da 0 a 10 Hz) alla tensione che controlla l'oscillatore, il filtro e/o l'amplificatore d'uscita. Gli effetti così ottenuti prendono il nome di vibrato quando il LFO modula la frequenza di lavoro del VCO, tremolo quando il LFO modula l'ampiezza del VCA, mentre la modulazione del filtro (VCF) permette di creare effetti del tipo "wah-wah".

3.2.2.7 Generatore di Inviuppo

Una caratteristica importante del suono di qualsiasi strumento musicale è l'espressione, ovvero la possibilità, per il musicista, di indurre lo strumento a variare un poco la timbrica, l'intensità e anche l'intonazione durante l'esecuzione, per rendere il suono più gradevole: si pensi ad esempio all'esecuzione di una sonata per pianoforte, o ad un assolo di violino, parti nelle quali la differenza di intensità, di ricchezza acustica e di vibrato permette di sottolineare alcuni passaggi, trasmettendo una determinata emozione all'ascoltatore.
Nel caso dei sintetizzatori, si può intervenire su vari parametri, ma resta sempre il problema delle esecuzioni dal vivo, che richiederebbero una squadra di tecnici per manovrare tutti i controlli nel modo e nella sequenza desiderate dall'esecutore. Inoltre, va osservato che molte di queste variazioni seguono uno schema fisso: si pensi ad esempio all'attacco in crescendo di una sezione d'archi, sempre uguale per ogni tasto che viene premuto.
Per riprodurre questo genere di modulazioni non periodiche, si ricorre a generatori di transitori di tensione, programmabili, in modo da generare il medesimo profilo della grandezza controllata ad istanti preimpostati (es. pressione di un tasto della tastiera). Alla pressione del tasto, il generatore di transienti genera una tensione crescente, che raggiunge un proprio massimo, per poi decadere ad estinguere l'effetto dopo il rilascio del tasto (coda, filtro passa-basso che diminuisce la frequenza di taglio, oscillatore che stona leggermente e così via).
Esistono in genere diversi schemi di generatore di transienti (o di inviluppo): AR, ADSR, AHDSR, AHDBDR e molti altri. L'AR (Attack, Release) definisce un tempo di salita della tensione alla pressione del tasto, ed un tempo di discesa al suo rilascio: adatto quindi per iterazioni semplici quali archi, fiati e voci.
ADSR (Attack, Decay, Sustain, Release) permette di creare un transitorio più vicino a strumenti caratterizzati da un attacco specifico (pianoforte, tromba, percussioni); alla pressione del tasto, la tensione di controllo sale in un tempo definito dal parametro Attack fino ad un picco massimo fisso; subito dopo, la tensione scende con la velocità definita dal parametro Decay,

fino a stabilizzarsi sul valore impostato dal parametro Sustain; infine, al rilascio del tasto, il parametro Release provvede a determinare in quanto tempo la tensione controllata ritornerà allo zero.
Quelli più complessi come AHDSR (Attack-Hold-Decay-Sustain-Release) e AHDBDR (Attack-Hold-Decay-Breakpoint-Decay-Release) sono meno diffusi, ma cominciano diffondersi nei più evoluti sintetizzatori assieme alla possibilità di creare inviluppi completamente personalizzati.

3.2.3 Sequencer

Il sequencer è un dispositivo (hardware o software), utilizzato nel campo musicale, che permette di creare e riprodurre delle sequenze di segnali di controllo, per comandare uno strumento elettronico.
Sebbene l'utilizzo del sequencer abbia un fine musicale, non è da confondere con un dispositivo di registrazione audio. A differenza del registratore, dove sono le forme d'onda di un suono a essere memorizzate, nel sequencer non viene memorizzato alcun segnale audio, solo quello di controllo. Si può immaginare un sequencer come una "mano elettronica" automatica e programmabile che suona strumenti e regola pulsanti e potenziometri di sintetizzatori e processori audio. Con l'avvento negli anni 80 del protocollo MIDI le possibilità dei sequencer si ampliarono: Il MIDI permetteva di trasmettere 16 esecuzioni polifoniche contemporaneamente con tutto il relativo corredo di espressioni esecutive. Ma questo di per sé grande salto di qualità fu ampliato da un altro salto di qualità che all'epoca la tecnologia stava compiendo: il computer da pachidermico strumento sperimentale dalle prestazioni modeste acquisiva sempre maggiori capacità di calcolo a costi e ingombri sempre minori, così da diventare sempre più un oggetto comune. Aziende come Atari e Commodore producevano macchine a 16 bit alla portata del proprietario di uno "studio", e fu così che il computer cominciò ad essere utilizzato come sequencer, grazie a opportuni software e alle interfacce MIDI che lo mettevano in comunicazione con qualsiasi apparecchiatura compatibile.

4. Componenti Progetto

4.1 Tastiera

4.1.1 Funzionamento

La tastiera è uno strumento musicale elettronico in grado di riprodurre i timbri di molti strumenti musicali attraverso un sintetizzatore, azionato mediante la pressione di tasti, analoghi a quelli del pianoforte. Spesso è munita di altoparlanti interni, mentre alcuni modelli necessitano di essere collegati a cuffie o amplificatori esterni.
Nelle tastiere odierne la qualità dei timbri degli strumenti è aumentata enormemente. Per favorire l'interoperabilità fra diversi sintetizzatori (anche non a tastiera) gli strumenti sono poi stati organizzati secondo lo standard general MIDI. Questo standard presenta tuttavia delle limitazioni, a cui Roland e Yamaha hanno ovviato creando nuovi standard, rispettivamente il General Standard (GS) e l'Extended General MIDI (XG), implementati nei loro strumenti.
Spesso le tastiere più ricche permettono di suddividere i tasti assegnando a due ottave a sinistra una funzione di accompagnamento e alle altre la parte solista. L'accompagnamento può consistere nell'assegnazione alle ottave apposite di uno strumento come organo o archi o addirittura di un arrangiamento automatico creato al momento in base agli accordi suonati con la mano sinistra. Gli strumenti di quest'ultimo tipo sono anche detti arranger.

4.1.2 Matrice R x C

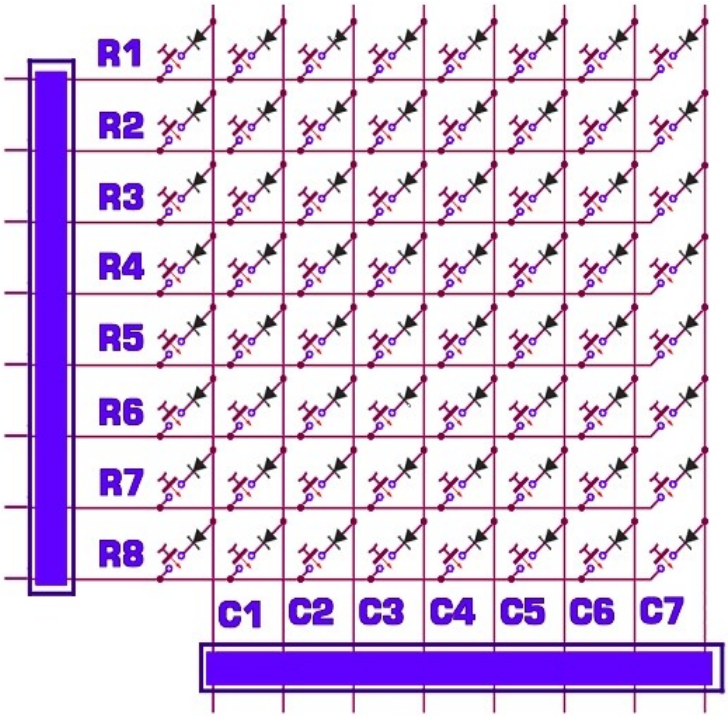
Da una prima analisi della tastiera elettronica utilizzata nel progetto, è stato possibile individuare al suo interno due sezioni principali:
- la scheda principale, con le sezioni di alimentazione e gli integrati per la produzione dei suoni;
- la tastiera vera e propria, con la relativa scheda di interfaccia, collegata alla scheda principale con 15 segnali digitali, e un diodo in corrispondenza di ogni tasto (in totale 49).

L'idea poi realizzata, consiste nel recuperare la sola elettronica di lettura della tastiera ed interfacciarla direttamente con la Raspberry. Ciò evita da un lato di dover prelevare manualmente il segnale di ogni tasto, ma implica per contro di dover conoscere nel dettaglio le modalità di interfacciamento dei tasti e la loro relazione con I segnali forniti in uscita dalla scheda.

L'impossibilità di disporre degli schemi originali dell'apparecchiatura, ha reso necessaria un'analisi empirica, realizzata parzialmente a vista, e parzialmente cercando di ricostruire le connessioni partendo dalle piste del circuito stampato, analizzando la continuità elettrica dei vari punti.

Considerate la totale assenza di integrati ed altre parti attive sulla scheda di interfaccia dei tasti (rendendo di fatto impossibile qualunque tecnica di multiplex digitale), e la presenza di 16 linee, distribuite su due connettori da 8 pin, si è potuto dedurre (ipotesi poi confermata) che la tastiera fosse gestita con la classica tecnica a “matrice di tasti”.

Tale tecnica, consente di gestire un numero elevato di tasti (49) partendo con un numero più limitato di linee (15), senza alcun componente aggiuntivo, secondo questo schema:



La pressione dei tasti, viene rilevata in questo modo:
- metto una colonna alla volta a livello logico alto.
- leggo lo stato delle righe

Le righe, collegate su ingressi configurati in pull down, risulteranno tutte a livello logico "basso", tranne quella collegata al tasto premuto.
Ripetendo tale lettura per ogni riga e per ogni colonna, attraverso dei cicli a frequenza elevata, è dunque possibile ottenere in tempo reale la mappatura dei tasti eventualmente premuti.
Il diodo, in corrispondenza di ogni tasto, garantisce che la corrente possa circolare solo in ingresso verso le linee di riga, evitando di fatto cortocircuiti indesiderati qualora fossero premuti più tasti collegati alla stessa colonna.

Il software dunque realizza i cicli necessari per porre i livelli logici sulle colonne e leggere le righe come sopra descritto, e ad ogni pressione di un qualsiasi tasto, ne fornisce la coordinata riga/colonna.

4.2 Raspberry

4.2.1 Struttura

La scheda “Raspberry Pi” è un single-board computer sviluppato sull'architettura RISC ARM.
La versione utilizzata in questo progetto (model B) monta un System-on-a-chip (SoC) Broadcom BCM2835, che incorpora un processore ARM1176JZF-S a 700 MHz, una GPU VideoCore IV, e 512 MB di memoria.
A bordo troviamo anche un'interfaccia Ethernet Broadcom (collegata in realtà al bus USB), e il connettore dei “General Purpose Input Output” (GPIO), ossia linee d'ingresso/uscita programmabili dall'utente mediante il software.

Per la memoria di massa, viene utilizzata una scheda SD, su cui viene caricata l'immagine del sistema operativo. Benchè in linea teorica possa essere montato qualunque sistema compatibile con l'architettura ARM, la comunità del progetto Raspberry fornisce numerose immagini già pronte per l'installazione, derivate da vari sistemi operativi quali:
- Debian, con un fork dedicato a Raspberry (Raspbian), che rispetto alla release ufficiale Debian ARM, aggiunge al suo interno anche I moduli del kernel per l'interfaccia Ethernet Broadcom (altrimenti non inclusi), e I moduli di gestione dei GPIO.
- Fedora, Android, e altri...

La scelta è caduta su Raspbian, in quanto basato su Debian, sistema operativo particolarmente indicato per applicazioni headless (senza interfaccia grafica) come quella in oggetto, e ampiamente documentato.

4.2.2 GPIO

Come detto, gli ingressi/uscite della Raspberry (GPIO) sono gestite da un apposito modulo del Kernel Linux eseguito dal sistema operativo.

Il kernel espone I GPIO al sistema operativo, come un device di sistema raggiungibile al percorso:

$ /sys/class/gpio/

L'utente può quindi scegliere di gestire il device in vari modi quali ad esempio:

- Da shell, scrivendo direttamente caratteri sul device, come se si trattasse di un dispositivo seriale:

Esempio:

```
#!/bin/bash

# GPIO4 come uscita
echo "4" > /sys/class/gpio/export
echo "out" > /sys/class/gpio/gpio4/direction

# GPIO7 come ingresso
echo "7" > /sys/class/gpio/export
echo "in" > /sys/class/gpio/gpio7/direction

# Valore “1” sull'usita
echo "1" > /sys/class/gpio/gpio4/value
```

 # Leggo il valore in ingresso
cat /sys/class/gpio/gpio7/value

 # Resetto i GPIO
echo "4" > /sys/class/gpio/unexport
echo "7" > /sys/class/gpio/unexport

- Attraverso una libreria Python (Rpi.GPIO)

import RPi.GPIO as GPIO

 # uso la numerazione pin del connettore anzichè I nomi simbolici
GPIO.setmode(GPIO.BOARD)

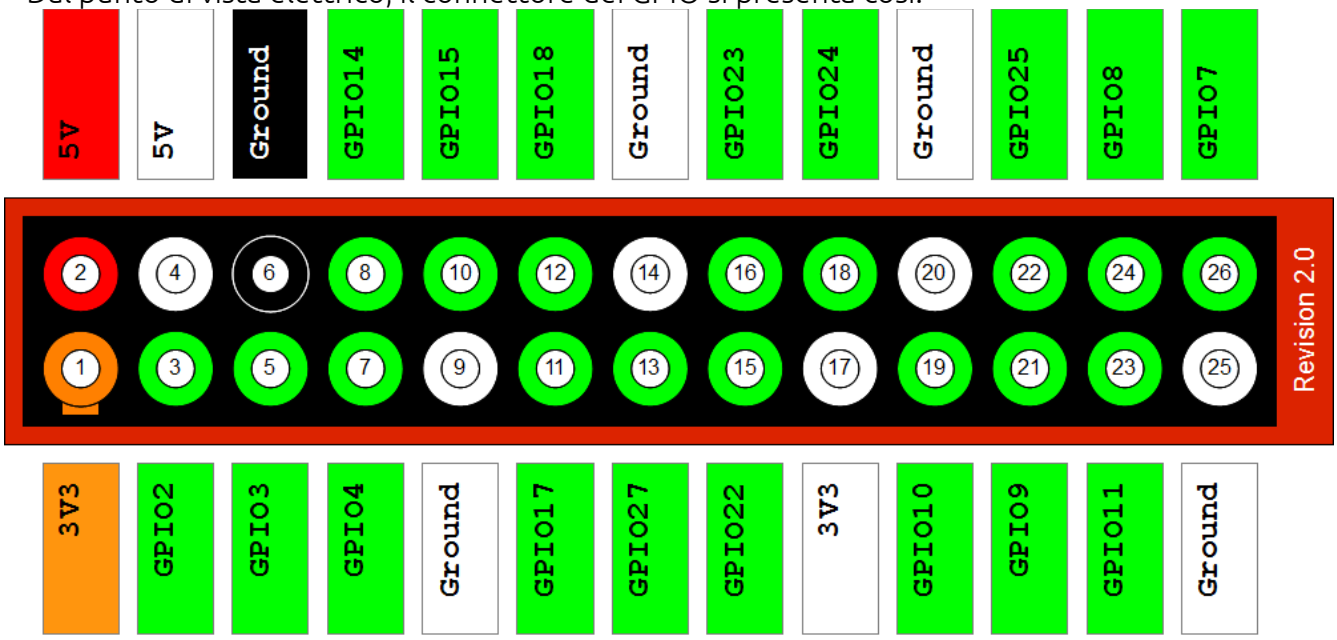
 # Imposto il pin 4 come uscita, e il pin 7 come ingresso
GPIO.setup(7, GPIO.IN)
GPIO.setup(4, GPIO.OUT)

 # Leggo il valore dal pin 7
input_value = GPIO.input(7)

 # Imposto a livello alto il pin 4
GPIO.output(4, GPIO.HIGH)

 # Resetto I GPIO
GPIO.clear()

Dal punto di vista elettrico, il connettore dei GPIO si presenta così:



La tensione 5V è fornita solo per scopi di servizio, in quanto le uscite, a livello logico alto, presentano tensione 3.3V, così come gli ingressi, che vengono rilevati “alti” quando ricevono tensione 3,3V.

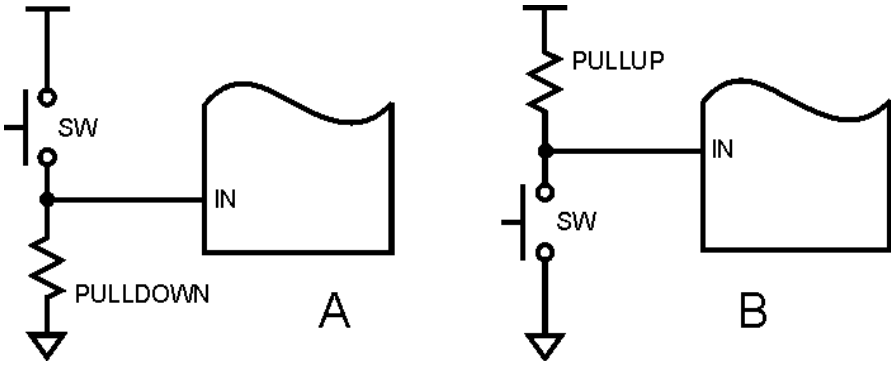
Gli ingressi, possono essere configurati anche per utilizzare resistenze di pullup o di pulldown, che non devono quindi essere inserite esternamente.
E' sufficiente intervenire via software. Ad esempio, utilizzando la libreria Python:

GPIO.setup(numero_pin, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
oppure
GPIO.setup(numero_pin, GPIO.IN, pull_up_down=GPIO.PUD_UP)

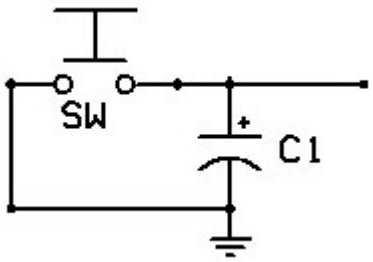
Configurando l'ingresso in pulldown, viene applicata una resistenza da 10K verso massa (da utilizzare se l'ingresso deve essere basso a riposo).

Configurando l'ingresso in pullup, viene allo stesso modo applicata una resistenza da 10K verso 3.3V (da utilizzare se l'ingresso deve essere alto a riposo).

I livelli logici "a riposo" vengono così vincolati ad uno stato certo. Ciò evita letture errate dei livelli in ingresso, dovute ad esempio ad interferenze sull'alimentazione.



Un altro problema relativo alla lettura di tasti con ingressi digitali, riguarda l'antirimbalzo (debouncing).
Per motivi di costruzione (presenza di piccole molle al loro interno), le commutazioni dei tasti potrebbero non essere sempre precise (ossia potrebbero dar luogo a numerose e velocissime commutazioni indesiderate al momento del loro rilascio).
Per risolvere questo problema è possibile intervenire su due fronti:
- via hardware, inserendo in parallelo al tasto un condensatore da 100nF, in modo da ottenere un filtro passa-basso, che consente così di eliminare la componente di alta frequenza generata dall'oscillazione del tasto.



- via software, inserendo dei ritardi (ordine dei 50-200ms) tra una lettura e l'altra, in modo da evitare le letture indesiderate (il software rimane in attesa...)

```
while True:
        if GPIO.input(channel):
                print('L'ingresso è alto')
        else:
                print('L'ingresso è basso')
        time.sleep(0.2)
```

Utilizzando gli interrupt, la procedura fornita in libreria consente di configurare l'ingresso per avere un ritardo di debouncing integrato:

```
#Esempio di un ingresso configurato in interrupt con 200ms di debouncing
GPIO.add_event_detect(channel, GPIO.RISING, callback=evento, bouncetime=200)
```

4.3 Python

4.3.1 Accenni sul Linguaggio

Python è un linguaggio di programmazione ad alto livello orientato agli oggetti adatto allo sviluppo di quasi ogni tipo di programmazione. Python infatti si presenta come linguaggio multipiattaforma e performante sia in ambiente linux che windows o unix e permette lo sviluppo sia di applicazioni lato desktop con opportune librerie grafiche che applicazioni lato server e scripting vario.
Python è un linguaggio multi-paradigma, che fa della dinamicità, semplicità e flessibilità i suoi principali obiettivi. Supporta il paradigma object oriented, la programmazione strutturata e molte caratteristiche di programmazione funzionale e riflessione. Le caratteristiche più immediatamente riconoscibili di Python sono le variabili non tipizzate e l'uso dell'indentazione per la definizione dei blocchi. Altre caratteristiche distintive sono l'overloading di operatori e funzioni tramite delegation, la presenza di un ricco assortimento di tipi e funzioni di base e librerie standard, sintassi avanzate quali slicing e list comprehension.
Il controllo dei tipi è comunque forte (strong typing) e viene eseguito al runtime (dynamic typing). In altre parole una variabile è un contenitore al quale viene associata un'etichetta (il nome) che può essere associata a diversi contenitori anche di tipo diverso durante il suo tempo di vita. Usa un garbage collector per la liberazione automatica della memoria.

4.3.2 Librerie NSound

Nsound è un framework sviluppato in C++ per la sintetizzazione audio. Il suo scopo è quello di essere potente con Csound implementando però delle features rese possibili dall'evoluzione ci C in C++. Nsound cerca di rendere facile lo sviluppo di suoni, anche complessi, ai programmatori, in modo da poter integrare queste librerie in maniera piuttosto diffusa.
Esempi:

4.3.2.1 Librerie Libao & Portaudio

Le librerie Libao e Portaudio sono due librerie per interfacciare le riproduzioni di suoni in real-time. Esse permettono all' utente di riprodurre automaticamente il suono dalla applicazioni senza doverlo salvare in file .wav e riprodurre tramite stream.
Nel mio progetto ho utilizzato in particolare libao poiché si è presentato più funzionante con il server audio utilizzato (pulseaudio).

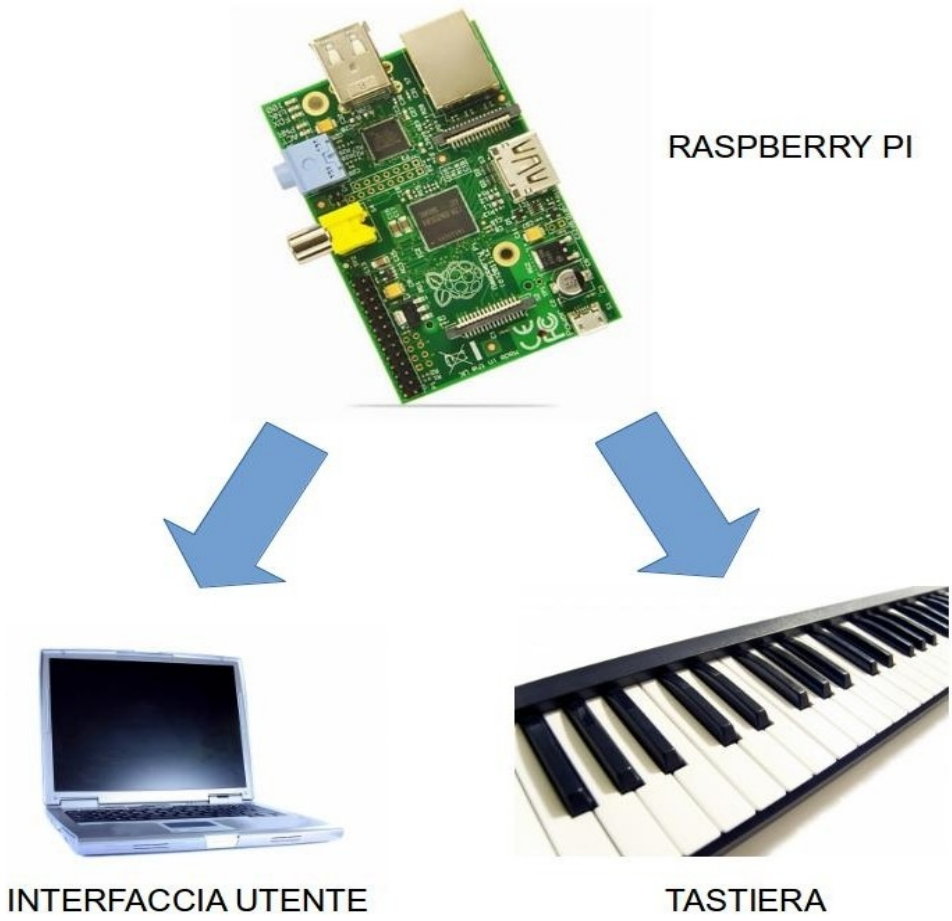
5. Struttura Progetto

5.1 Schema Generale

1) Tastiera elettronica
La vecchia tastiera elettronica, è stata riutilizzata nei tasti e nella relativa interfaccia.

2) Interfaccia di connessione
Per l'adattamento delle connessioni dal connettori 8 poli (riga/colonna) esistenti sulla tastiera, al connettore GPIO della scheda Raspberry a matrice RxC.

3) Raspberry
Single board computer, che gestisce i segnali della matrice per la lettura dei tasti, ospita il software per la generazione dei suoni, l'interfaccia con il personal computer dell'utente.



SOFTWARE

1) Python:
Linguaggio di programmazione nativo di Raspberry.

2) Nsound
Librerie Python per la generazione di suoni, descritti attraverso le singole armoniche

3) Librerie LibAo
Librerie di interfaccia con il sottosistema sonoro PulseAudio, per portare i suoni generati da Nsound sul device audio predefinito del sistema

4) Librerie QT
Librerie grafiche per la creazione dell'interfaccia utente.

5.2 Difficoltà Incontrate & Imprevisti

Il progetto è stato costellato da numerosi imprevisti. Il primo e più dispendioso dal punto di vista temporale è stata la rottura della raspberry, causata probabilmente da un sovraccaricamento della tensione della linea elettrica. Questa problematica ha causato innanzitutto la perdita di una parte del lavoro svolto ma soprattutto dover aspettare più di un mese per l'arrivo della seconda.
Una difficoltà iniziale è stata rappresentata dal dover riutilizzare l' interfaccia nativa della tastiera senza riconoscerne gli schemi: tramite un attento e minuzioso lavoro è stato però ricostruito interamente il funzionamento con relativo mappamento dei tasti.
Un'ulteriore difficoltà è stata quella di riprodurre i suoni tramite libao nello stesso programma di gestione delle gpio: la libreria libao richiede infatti di non essere eseguita come utente root mentre le librerie Rpi necessitano i permessi di amministrazione per l'esecuzione.
Il problema finale, che non ha permesso lo sviluppo di un intero sintetizzatore, è stata la scarsa capacità computazionale della raspberry, che impiega diversi secondi a sintetizzare un suono: è stato quindi solamente possibile sviluppare un software capace di riprodurre il suono di un solo tasto alla volta (sinusoide fondamentale). Si è deciso di strutturare, in secondo momento, un' interfaccia di collegamento tra raspberry e pc in modo da leggere i tasti con la raspberry e riprodurre i suoni in real-time tramite computer.

6. Riferimenti

Il materiale ed i documenti usati per lo studio e la realizzazione di questo documento sono tratti, anche integralmente, da:

http://www.sintetizzatore.com
www.dei.unipd.it/~musica/IM/cap5.pdf
https://en.wikipedia.org/wiki/Synthesizer
https://en.wikipedia.org/wiki/Music_sequencer
https://en.wikipedia.org/wiki/Sound
http://en.wikipedia.org/wiki/Yamaha_DX7
www.fmboschetto.it/didattica/pdf/il_suono.pdf
http://www.suonoelettronico.com/cap_v_4_2_tecsin.htm
https://en.wikipedia.org/wiki/Frequency_modulation_synthesis
http://people.ece.cornell.edu/land/courses/ece4760/Math/GCC644/FM_synth/Chowning.pdf

7. Ringraziamenti

Un sentito ringraziamento va a tutte le persone che hanno collaborato con me per la realizzazione di questo progetto, che mi hanno supportato e che hanno sempre creduto in me.

Ringrazio i professori che mi hanno aiutato nella realizzazione di questo progetto, in particolare Brunetti Marco per la parte riguardate elettronica/telecomunicazioni.
Ringrazio tutti coloro che in questi anni si sono impegnati per trasmettermi le loro conoscenze e infondermi le loro passioni.
Ringrazio tutti gli amici che mi hanno sopportato e stimolato durante quest'ultimo anno di scuola superiore.
Ringrazio Anna, la mia ragazza, per essere sempre riuscita ad infondermi fiducia e forza di proseguire anche nei momenti più difficili e per avermi aiutato a credere di poter realizzare un progetto di quest'entità.
Ringrazio tutta la mia famiglia per la vicinanza, compresi i nonni che sono stati felici di ascoltare le mie esposizioni e si sono interessati nel mio progetto.
Un grazie speciale e forse il più grande va, infine, all'amico Nick (Nicola Ferrari) che mi ha aiutato e consigliato durante tutta la stesura della tesina e la realizzazione del progetto.

“A volte il vincitore è semplicemente chi non ha mai mollato.
Chi rinuncia ai sogni è destinato a morire.”

(Jim Morrison)

Grazie.

Michele Sordo

_____

Questo documento, insieme alla presentazione ed al codice di questo progetto, è disponibile online sulla piattaforma github all' indirizzo:
https://github.com/mickfenneck/micksynth

Riferimenti all' autore del progetto presso: http://mickfenneck.github.io/

Prossimamente presso il sito ufficiale http://www.mickfenneck.com/