



UNIVERSITÀ
DEGLI STUDI
DI PADOVA



DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

CORSO DI LAUREA IN INGEGNERIA ELETTRONICA

**Elettronica e musica:
design di uno strumento musicale digitale
sul modello dell'astrolabio**

Laureando

Giacomo Nogarín

Relatore

Prof. Sergio Canazza Targon

Correlatore

Alessandro Fiordelmondo

ANNO ACCADEMICO 2023-2024

Data di laurea 24/09/2024

Ai miei amici e alla mia famiglia

Indice

1	Stato dell'arte	1
1.1	Introduzione ai Digital Musical Instruments	1
1.2	Esempi di DMI	1
1.2.1	"Principles for Designing Computer Music Controllers", Perry Cook	2
1.2.2	"The Overtone Violin", Dan Overholt	3
1.2.3	"KNOTTING THE MEMORY//ENCODING THE KHIPU_:", Patricia Cadavid Hinojosa	5
1.3	Considerazioni sulla progettazione del DMI	6
2	Progettazione del Digital Musical Astrolabe (DMA)	7
2.1	Premessa	7
2.2	Componenti hardware	8
2.2.1	Modulo GY-521 MPU-6050	8
2.2.2	Arduino MKR1000 WiFi	9
2.3	Software	10
2.3.1	Arduino IDE	10
2.3.2	Ableton Live	11
2.4	Comunicazione Open Sound Control	12
2.5	Fase di progettazione	13
2.5.1	Collegamento degli elementi hardware	13
2.5.2	Codice	14
2.5.3	Ricezione OSC nella DAW	18
3	Un esempio pratico	21
3.1	Sintetizzatore digitale e sound design	21
3.2	Possibili migliorie future del progetto	25
4	Conclusioni	27

Introduzione

La musica, come tutte le altre forme d'arte, è sempre stata un elemento fondamentale per lo sviluppo culturale. Proprio per questo motivo, l'uomo ha sempre cercato nuovi mezzi per poterla evolvere: partendo da semplici strumenti percussivi nella preistoria, passando poi alla lira greca, solo per citarne alcuni, abbiamo oggi raggiunto uno sviluppo tecnologico che ci ha permesso di ottenere infiniti strumenti e metodologie per poterla realizzare. Il grande cambiamento è avvenuto con la nascita dei primi sintetizzatori analogici negli anni quaranta, dai quali c'è stata una continua evoluzione fino agli attuali synth digitali. Questi capolavori ingegneristici hanno svoltato radicalmente il modo in cui le opere musicali vengono create, essi infatti permettono di avere un ottimo controllo nella modulazione dei suoni, lasciando allo stesso tempo al musicista la possibilità di generare delle sequenze armoniche. Di fatto la loro creazione è avvenuta con il fine di ottimizzare notevolmente il lavoro del musicista, rendendolo più rapido e allo stesso tempo performante; ovviamente non è da trascurare il potenziamento del lato creativo, infatti attraverso questi strumenti è possibile sintetizzare suoni completamente nuovi, lasciando un orizzonte illimitato alle idee e dando uno strumento effettivo per poterle realizzare. Come detto in precedenza i sintetizzatori sono delle vere opere ingegneristiche, tanto perfetti da essere diventati uno standard nell'industria discografica: dei dispositivi che permettono di dar vita alla propria arte. Ma se fosse proprio lo strumento a diventare un'opera d'arte? Questo è il caso dei Digital Musical Instruments (DMIs), strumenti musicali innovativi costituiti da un'interfaccia fisica connessa ad una parte digitale per la generazione del suono. Questi strumenti sono spesso creati per opere d'arte specifiche e, a volte, diventano essi stessi opere d'arte autonome. Il loro sviluppo avviene in un contesto interdisciplinare che unisce la ricerca sulla Human-Computer Interaction (HCI) e le arti performative. Questa tesi presenta lo sviluppo di un DMI sulla base di uno strumento astronomico antico, l'astrolabio. Il DMI è stato ideato per la realizzazione di uno spettacolo teatrale interattivo promosso dal Museo "Giovanni Poleni" - Storia della Fisica del Centro di Ateneo per i Musei (CAM) dell'Università degli Studi di Padova in collaborazione con il Centro di Sonologia Computazionale (CSC) della stessa Università. Lo spettacolo teatrale è un dialogo performativo

tra due attori che ha lo scopo di raccontare la storia e mostrare il funzionamento dei dispositivi astronomici antichi. La tesi si pone come obiettivo l'analisi del DMI, soffermandosi principalmente sulle metodologie impiegate per la progettazione, sia hardware che software, e le conoscenze utilizzate per il suo sviluppo, senza escludere il lato artistico. In seguito verranno illustrati i risultati raggiunti tramite un esempio pratico, mostrando il funzionamento effettivo del DMI e discutendone le possibili migliorie e implementazioni future.

Elenco delle figure

1.1	Frog Maraca, immagine tratta da [1].	2
1.2	TapShoe, immagine tratta da [1].	3
1.3	JavaMug, immagine tratta da [1].	3
1.4	The Overtone Violin, immagine tratta da [1].	4
1.5	Electronic_Khipu_, immagine tratta da [2].	5
2.1	modulo GY-521 MPU-6050.	8
2.2	Arduino MKR1000 WiFi.	9
2.3	Arduino IDE.	10
2.4	Ableton Live Suite.	12
2.5	collegamento tra Arduino MKR1000 WiFi e modulo GY-521 MPU-6050.	13
2.6	output serial monitor dell'IDE.	18
2.7	plugin digitale Max for Live OSC TouchOSC.	19
3.1	sintetizzatore Serum.	22
3.2	sintesi FM tramite gli oscillatori A e B.	22
3.3	LFO.	23
3.4	filtro passa banda e rispettive manopole.	23
3.5	saturatore e riverbero all'interno di Serum.	24
3.6	mappatura dei valori ricevuti da OSC TouchOSC alle MACRO di Serum.	24
3.7	primo vettoriale dell'astrolabio.	25
3.8	secondo vettoriale dell'astrolabio.	25

Capitolo 1

Stato dell'arte

1.1 Introduzione ai Digital Musical Instruments

I DMI sono strumenti musicali costituiti da una parte fisica e una digitale connesse fra di loro. La parte fisica funge da interfaccia per l'utilizzatore dandogli modo di interagire con quella digitale, la quale permette di generare e modulare il suono. Sebbene siano piuttosto recenti, i DMI sono stati studiati approfonditamente e se ne possono trovare moltissime tipologie in tutto il mondo. Al contrario dei sintetizzatori, essi non sono sempre ideati con lo scopo di rendere più efficiente la produzione musicale, ma nella maggior parte dei casi vengono prodotti per essere essi stessi delle opere artistiche. Nel 2001 è nata "New Interfaces for Musical Expression", abbreviato con NIME, una conferenza dedicata alla ricerca e lo sviluppo di nuove tecnologie e le loro applicazioni alle performance musicali. Nata per essere un workshop della "ACM CHI Conference on Human Factors in Computing Systems" e avendo una periodicità annuale, dal 2001 fino al 2024 sono stati trattati innumerevoli argomenti di svariati tipi, di cui molti riguardanti proprio i DMI.

1.2 Esempi di DMI

In questa sezione verranno esposti alcuni articoli presi dalle conferenze NIME (e non solo) che si sono svolte nell'ultimo ventennio, fornendo esempi di DMI con una descrizione del loro funzionamento e l'idea che ha portato alla loro creazione.

1.2.1 "Principles for Designing Computer Music Controllers", Perry Cook

Scritto da Perry Cook, questo articolo [1] si focalizza sull'esposizione dei principi sviluppati nel corso degli anni dall'autore riguardo il design dei controller. Essi non sono da considerare come universali ma derivano dalla sua esperienza acquisita nella progettazione di interfacce musicali. La loro analisi verrà tralasciata, saranno invece discussi gli esempi forniti nell'articolo. Il primo progetto citato è il PhISEM (Physically Inspired Stochastic Event Modeling), incentrato sulla sintesi di percussioni "particle-type" e di suoni reali; esso ha portato alla nascita di nuovi dispositivi, sia percussivi che non. Un esempio è il Frog Maraca, visibile in figura 1.1, un semplice strumento basato su delle maracas a forma di rana tramite le quali è possibile inviare dei segnali Musical Instrument Digital Interface (MIDI) per riprodurre delle sequenze di jazz fusion. Il successo deriva dalla sua semplicità di utilizzo, bastava infatti scuoterlo per poter sentire un trio jazz suonare delle melodie complesse. Il Frog Maraca, come si è potuto intuire, non è stato creato per i musicisti ma piuttosto per chi non lo è: semplice da utilizzare ma allo stesso tempo efficace e divertente.



Figura 1.1: Frog Maraca, immagine tratta da [1].

Proseguendo oltre, si introduce ciò che l'autore categorizza sotto il nome di "Foot, Hand, Kitchen Wear/Ware (1997–2000)". Il principio di funzionamento di questi strumenti è simile a quello del Frog Maraca con la differenza che questi non sono veri e propri strumenti musicali, al contrario sono degli oggetti di uso quotidiano.

Da prima la TapShoe, mostrata in figura 1.2, una scarpa unita a dei sensori di forza e accelerazione direttamente collegati a un Processore di Segnali Digitali (DSP: Digital Signal Processor) progettato in modo da eseguire algoritmi basati su quelli del progetto PhISEM. Questi generavano un pattern ritmico e nel mentre chi indossava la scarpa poteva battere il tacco per aggiungere il suo tocco personale al groove.



Figura 1.2: TapShoe, immagine tratta da [1].

L'oggetto successivo è la JavaMug, più sofisticata rispetto ai precedenti. Come si può vedere in figura 1.3, essa è una tazza da caffè, la parte interessante sta nelle modifiche effettuate per renderla un DMI a tutti gli effetti: sono stati aggiunti sensori di pressione, sensori di inclinazione e dei pulsanti, il tutto per avere un maggiore controllo e poter suonare tramite l'algoritmo gli strumenti di una band techno-latin. Il ruolo principale lo ricoprono i sensori, infatti semplicemente stringendo la tazza in modo diverso o inclinandola leggermente è possibile introdurre grandi variazioni a ciò che viene suonato.



Figura 1.3: JavaMug, immagine tratta da [1].

Seppure questi progetti fossero abbastanza semplici sono tuttora degli ottimi esempi per capire il concetto di DMI e il loro principio di funzionamento.

1.2.2 "The Overtone Violin", Dan Overholt

Nella prima parte dell'articolo [1] lo scrittore esprime il suo interesse riguardo lo sviluppo di nuove interfacce per il design di uno strumento musicale, tuttavia dice che la differenza tra un dispositivo effettivamente utile e un semplice gadget è molto sottile. Prosegue scrivendo che la progettazione basata sugli strumenti acustici esistenti può essere efficace ma provocare al contempo una riduzione della creatività, con il rischio di creare qualcosa di non innovativo, però allo stesso tempo inventare qualcosa di completamente nuovo può rivelarsi un salto nel vuoto. Introduce quindi The Overtone Violin categorizzandolo

come strumento ibrido, cioè uno strumento acustico già esistente ma con l'aggiunta di estensioni per migliorarne le performance. Alcune caratteristiche si discostano da quelle di un generico strumento ibrido, infatti mentre le espansioni di quest'ultimo sono focalizzate sull'acquisizione dei movimenti tipici già esistenti, The Overtone Violin contiene degli appositi sensori per catturare gestualità inedite, ponendosi al contempo l'obiettivo di essere un'evoluzione del classico violino. Esso di conseguenza, contrariamente ai DMI esposti nella sezione precedente, vuole essere un prodotto innovativo, dedicato esclusivamente ai musicisti. È composto da un corpo di legno d'acero e presenta sei corde, rispetto alle usuali quattro, incorpora un sistema di pickup costituito da dei LED IR, sopra le corde, e dei fotodiodi, sotto di esse, collegati a un un circuito stampato (PCB: Printed Circuit Board) in grado di rilevare la luce infrarosso, di conseguenza non è necessario il corpo di risonanza per ottenere un suono di qualità. Nella parte inferiore dello strumento sono disponibili degli output per le singole corde mentre sul lato sinistro troviamo l'uscita audio principale affiancata da tre potenziometri per controllare volume d'uscita, tono e volume degli eventuali segnali MIDI trasmessi. Nella testa del violino si trovano la maggior parte dei sensori per i gesti, sono più di venti e presentano tutti un collegamento con il microcontrollore Microchip PIC16F877 il quale converte i dati raccolti e li invia ad un trasmettitore RF, permettendo una comunicazione wireless con dispositivi esterni. The Overtone Violin è una perfetta dimostrazione dello sviluppo e della complessità che uno strumento classico può raggiungere tramite la sua unione con dei componenti elettronici digitali. In figura 1.4 è possibile vederlo da due prospettive diverse.



Figura 1.4: The Overtone Violin, immagine tratta da [1].

1.2.3 "KNOTTING THE MEMORY//ENCODING THE KHI-PU_:", Patricia Cadavid Hinojosa

I DMI possono rivelarsi un ottimo modo per raccontare le tradizioni e la cultura del proprio paese. L'ultimo progetto che verrà analizzato non si trova in uno degli articoli del NIME ma è invece discusso da Patricia Cadavid Hinojosa nella sua tesi di master [2]. Lei descrive il Khipu come un dispositivo per l'elaborazione di dati attraverso un "codice" visivo e tattile, essendo composto da corde e nodi di svariati colori e dimensioni è possibile attribuire significati precisi alla loro disposizione permettendo di svolgere calcoli matematici, rappresentare eventi storici e molto altro. Esso ricopre un ruolo importante nella storia dell'impero Inca e Patricia ha deciso di utilizzarlo per la creazione di un DMI che prende il nome di *Electronic_Khipu_*, riportato in figura 1.5. Ispirato all'originale, anch'esso è costituito da corde, al contrario dei materiali originali è stata utilizzata della gomma conduttiva in modo da poter captare i segnali. Il suo principio di funzionamento si basa sull'intreccio di nodi, così facendo lo sfregamento e l'allungamento delle corde provoca delle vibrazioni che vengono catturate e processate da un microprocessore Teensy e poi tradotte in segnali MIDI inviabili a un qualsiasi software di musica per controllare effetti audio o sintetizzare suoni. L'interazione fra utilizzatore e strumento è evidenziata dalla versatilità delle corde, semplicemente toccandole in modo diverso o cambiando pressione è possibile modificare il segnale e di conseguenza la modulazione del suono.

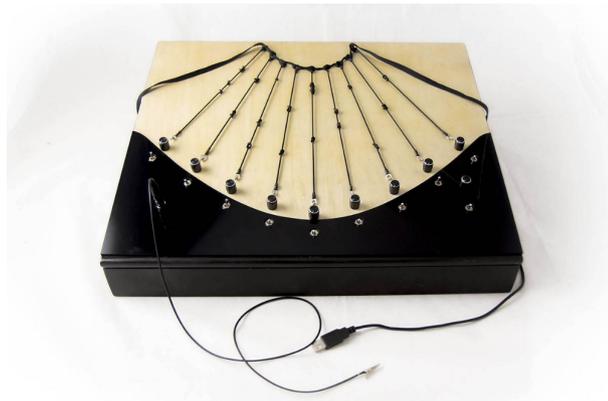


Figura 1.5: *Electronic_Khipu_*, immagine tratta da [2].

1.3 Considerazioni sulla progettazione del DMI

Nelle sezioni precedenti si è avuto modo di esplorare molteplici tipologie di DMI, ci si è soffermati sul loro funzionamento ma sono anche stati esaminati i dispositivi stessi deducendo che il loro scopo non è solo quello di fungere da strumenti musicali, possono invece essere usati a scopo ludico o per riportare in vita il passato sotto una prospettiva moderna tramite l'unione di arte, storia e tecnologia. Il progetto proposto in questo documento tiene conto di molti degli aspetti discussi finora: è destinato ad un pubblico senza competenze in ambito musicale ed offre contemporaneamente una versione moderna e interattiva di uno strumento astronomico dell'antico Egitto, l'astrolabio. Da questi deriva la focalizzazione nella sua facilità di utilizzo e nella cura dell'aspetto esteriore, in modo da poter catturare facilmente l'attenzione del pubblico incuriosendolo e spingendolo a provare ad utilizzarlo.

Capitolo 2

Progettazione del Digital Musical Astrolabe (DMA)

2.1 Premessa

Durante questi mesi di progettazione, dopo vari confronti con docenti e studenti, innumerevoli prove e processi di ottimizzazione, è stato possibile finalizzare il progetto, potendone identificare i tre elementi principali: hardware, software e protocollo di comunicazione. La parte hardware costituisce il cuore del progetto, in termini di DMI identifica l'interfaccia che permetterà l'interazione con lo strumento. Essa è indispensabile per il raccoglimento dei dati e la loro elaborazione; ciò è reso possibile dagli elementi di cui è composta cioè un sensore, per il raccoglimento, e una scheda Arduino, per l'elaborazione. Il software è la mente del dispositivo, la parte digitale del DMI, ed anch'esso ha un duplice scopo. Analogamente all'hardware, è suddiviso in due parti ed entrambe hanno un fine preciso: l'interfaccia digitale di Arduino con il rispettivo codice, per garantire un corretto funzionamento del microprocessore, e la Digital Audio Workstation (DAW) la quale si occupa della sintesi del suono e la sua modulazione. Infine il protocollo di comunicazione risulta essere fondamentale per la corretta comunicazione tra hardware e software favorendo l'invio e la ricezione dei dati tra i due. Grazie all'unione di questi tre elementi è stato possibile completare quella che è ritenuta essere una versione 1.0 del DMI, potendo fungere da punto di partenza per possibili migliorie future. In seguito verrà fornita una descrizione più approfondita di quanto elencato in precedenza, discutendone le specifiche, la modalità di funzionamento e le motivazioni che hanno portato alla loro scelta, proseguendo successivamente con la fase di progettazione.

2.2 Componenti hardware

Nelle considerazioni preliminari è stata esaminata l'importanza dei componenti hardware, infatti senza di loro l'utilizzatore non avrebbe alcun modo di interagire con il DMI. La loro presenza permette di raccogliere le informazioni esterne, come l'inclinazione o la rotazione dello strumento, e conseguentemente controllare il suono, tutto ciò con estrema facilità.

2.2.1 Modulo GY-521 MPU-6050

Il modulo GY-521 è un componente elettronico che integra al suo interno l'MPU-6050 [3], un sensore di MotionTracking che possiede un accelerometro a tre assi, un giroscopio a tre assi, un sensore di temperatura e un Digital Motion Processor (DMP) che permette di elaborare i dati esterni raccolti dai primi due sfruttando sei convertitori analogico-digitale (ADC: Analog to Digital Converter). L'accelerometro e il giroscopio sfruttano il principio di funzionamento dei materiali piezoelettrici, i quali sono in grado di generare una differenza di potenziale, quindi una corrente, se sollecitati da una deformazione meccanica. L'accelerometro ritorna dei valori modificabili in base all'inclinazione del sensore lungo i tre assi x, y e z, mentre il giroscopio offre la stessa possibilità ma cambiando i valori in base alla rotazione attorno i tre assi. In figura 2.1 è possibile vedere il sensore MPU-6050 saldato nel centro del modulo GY-521.



Figura 2.1: modulo GY-521 MPU-6050.

Un altro aspetto fondamentale è il canale di comunicazione I2C attraverso cui è possibile effettuare una trasmissione (o ricezione) seriale dei dati verso le periferiche esterne, in questo caso l'Arduino MKR1000 WiFi, con una frequenza massima di 400kHz. Questo modulo elettronico è con buona probabilità il più importante fra tutti gli elementi del progetto, da questa descrizione si evince che è proprio il suo funzionamento a permettere di ricavare i dati per la modulazione interattiva del suono.

Si è scelto di utilizzare l'MPU-6050 per la sua velocità di trasmissione, le dimensioni ridotte e il costo contenuto, a discapito della precisione, ciononostante durante i test si è dimostrato essere più che adeguato a ricoprire questo ruolo.

2.2.2 Arduino MKR1000 WiFi

L'oggetto che sta per essere introdotto ha un enorme potenziale, se sfruttato pienamente può permettere di creare progetti estremamente avanzati. Arduino è una piattaforma open-source caratterizzata da hardware e software progettati appositamente per permettere una facilità nel loro utilizzo [4], viene usata da ingegneri nei loro progetti, da professori per scopi didattici, ma anche da artisti e molti altri professionisti. La piattaforma offre una vasta gamma di prodotti per qualsiasi tipo di necessità, gli aspetti che più interessavano la scelta erano la presenza di un modulo per il WiFi, compattezza del dispositivo e possibilità di estensioni per eventuali migliorie. Dopo aver confrontato varie schede si è optato per l'MKR1000 WiFi [5], riportato in figura 2.2. Esso presenta tutti i requisiti cercati: dimensioni ridotte rispetto alla maggioranza delle altre schede, un numero elevato di pin per input/output di segnali e soprattutto, come è evidenziato nella sua descrizione, una grande facilità nella connessione WiFi pur avendo poca esperienza riguardo le reti. La scheda ha un'area di 61.5mm x 25mm, presenta 8 pin per i dati analogici, 7 per l'input, 1 per l'output, 8 pin per input/output digitali, 2 pin per la comunicazione I2C, memoria dinamica flash da 256 KB, memoria statica SRAM da 32 KB, è alimentata a 3.3 Volt e integra un modulo WiFi ATSAMW25 a basso consumo di potenza.



Figura 2.2: Arduino MKR1000 WiFi.

All'interno del progetto funge da tramite fra l'MPU-6050 e la parte digitale, si occupa della ricezione dei dati dal sensore e della loro conversione in un formato adeguato ai parametri della DAW, occupandosi successivamente della loro trasmissione tramite OSC.

Per svolgere questi compiti necessita di un codice scritto all'interno del suo IDE, di cui verrà parlato nel capitolo successivo. Nella realizzazione di questa prima versione del DMI sono stati utilizzati solamente 4 pin, due per l'alimentazione e due per la comunicazione I2C, lasciando quindi molta libertà per l'aggiunta di estensioni.

2.3 Software

Per poter ottenere il corretto funzionamento dell'hardware, la parte software risulta indispensabile. Il codice scritto consente al microprocessore e al sensore di svolgere le loro azioni mentre la DAW si occupa dell'impianto sonoro.

2.3.1 Arduino IDE

L'Ambiente di Sviluppo Integrato (IDE: Integrated Development Environment) di Arduino, visibile in figura 2.3, è uno dei software dell'omonima piattaforma [6]. Al suo interno è possibile scrivere del codice per programmare il microprocessore della scheda, permettendole quindi di eseguire le istruzioni desiderate. L'interfaccia è intuitiva e presenta una parte centrale in cui scrivere il proprio programma (sketch), una barra laterale per implementare librerie esterne e tre pulsanti di colore azzurro in alto a sinistra per compilare il programma, caricarlo nel microprocessore ed eseguire il debugging.

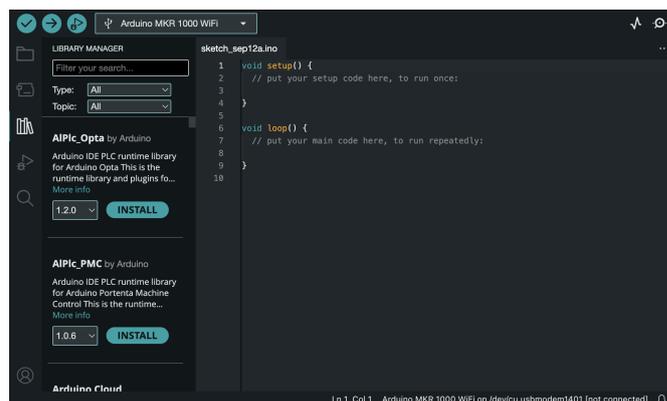


Figura 2.3: Arduino IDE.

Il codice deve essere scritto nel linguaggio di Arduino, basato su C/C++, risultando semplice anche per chi è alle prime armi o non ha mai programmato prima. Il programmatore può utilizzare un grande numero di funzioni di default [7], tra le più importanti troviamo `analogRead()` e `analogWrite()` per la lettura di ingressi analogici e il loro output, `digitalRead()` e `digitalWrite()` per ingresso e uscita di valori digitali, `pinMode()` per la

configurazione dei pin della scheda. Come detto prima l'IDE offre il vantaggio di importare librerie esterne. Queste sono un insieme di più codici sviluppati da programmatori di tutto il mondo, il loro utilizzo risulta essere di grande aiuto quando si ha a che fare con componenti esterni o se si vogliono eseguire funzioni più avanzate. Nella scrittura del codice sono state utilizzate quattro librerie, due per la comunicazione I2C tra sensore e microprocessore, una per la connessione WiFi e l'altra per l'invio di messaggi OSC alla DAW, tutto tramite l'utilizzo delle funzioni in esse contenute.

2.3.2 Ableton Live

Ableton Live è una DAW sviluppata dall'azienda tedesca Ableton [8], nato il 30 ottobre 2001 si dimostra tutt'oggi essere uno dei software musicali più competitivi sul mercato. Nel corso degli anni ha ottenuto un enorme successo soprattutto per quanto riguarda la produzione di musica elettronica (EDM: Electronic Dance Music), sono infatti molti gli artisti che lo utilizzano per le loro produzioni, tra i più famosi citiamo Skrillex, David Guetta e i Daft Punk. Oltre che per la produzione, Live risulta un'ottima scelta per molte altre attività, come il sound design o la registrazione dal vivo di strumenti acustici o analogici. Contrariamente all'Arduino IDE, la sua interfaccia risulta essere molto complessa e tutt'altro che intuitiva. Fortunatamente, data la grande passione per la produzione e il sound design, si è iniziato ad utilizzare questo software e studiarne il suo funzionamento già dal 2020, riuscendo dopo quattro anni ad acquisire una buona conoscenza sul suo utilizzo. In questa sezione ci si limiterà a discutere il suo ruolo nel progetto, data la numerosità di funzioni e plugin interni sarebbe impossibile analizzarli tutti. Ableton Live è disponibile in tre versioni:

- Intro: questa versione contiene 5+ GB di suoni, un numero ridotto di effetti e permette di inserire un massimo di 16 tracce audio o MIDI.
- Standard: rispetto alla versione Intro si può inserire un numero illimitato di tracce audio/MIDI, sono presenti più effetti e 38+ GB di suoni.
- Suite: versione definitiva, 71+ GB di suoni, presenza di tutti gli effetti audio sviluppati da Ableton, numero illimitato di tracce audio/MIDI e possibilità di aggiungere plugin Max for Live.

Fra le tre versioni è stata scelta la Suite, unica opzione possibile per la creazione del DMI, infatti l'ultima feature descritta nell'elenco è quella che permette di implementare il protocollo OSC all'interno del software. Max for Live è una piattaforma nata dall'unione di Live e Max per creare da zero i propri effetti audio o strumenti digitali [9]. Similmente

alle librerie dell'Arduino IDE, il software permette di importare dei plugin Max for Live sviluppati da altri utenti e introdurli nella DAW, ampliandone così le funzionalità. Nel DMI, Ableton Live è il software da cui viene generato il suono, permettendo la sua modulazione tramite la mappatura dei valori ricevuti via OSC dall'MKR1000 WiFi. La sua versatilità e il suo workflow si sono rivelati estremamente comodi per l'ottimizzazione di questi processi. Il figura 2.4 è stata riportata una schermata dell'interfaccia principale della versione Suite.



Figura 2.4: Ableton Live Suite.

2.4 Comunicazione Open Sound Control

Open Sound Control (OSC) è un protocollo di comunicazione sviluppato da Adrian Freed e Mathew Wright al "Center for New Music and Audio Technologies (CNMAT) della University of California nel 1997 e successivamente proposto nel 2002 [10]. Esso è nato come alternativa al suo predecessore, il MIDI, dimostrandosi essere una versione più accurata, con minore latenza e maggiore flessibilità nella comunicazione per il collegamento tra computer, strumenti musicali elettronici, o più in generale dispositivi multimediali, con l'obiettivo di ottimizzare le performance musicali dal vivo. In questi anni il protocollo OSC è stato introdotto in moltissimi progetti, soprattutto quelli presentati alle conferenze NIME. Esso può essere utilizzato per inviare o ricevere vari tipi di dati, numeri interi, decimali e perfino stringhe di caratteri. Dato che presenta una comunicazione wireless è necessario inserire l'indirizzo IP della destinazione, preceduto dal carattere `"/`. Se il dispositivo in ricezione presenta più moduli che implementano l'OSC (nel progetto discusso potrebbero essere Ableton Live e un sintetizzatore digitale stand-alone) c'è la possibilità di attribuire un valore alla porta di comunicazione, in modo che ognuno di loro riceva il messaggio corretto. OSC offre la possibilità di inviare dei bundle di messaggi, funzione che

ad alimentare il dispositivo e permettere la comunicazione I2C, ma inaspettatamente sono bastati per ottenere i dati di entrambi l'accelerometro e il giroscopio. La connessione con l'MKR1000 WiFi è risultata semplice, anch'esso dispone di pin omonimi a quelli del modulo e quindi è stato sufficiente eseguire un corretto collegamento dei cavi.

2.5.2 Codice

La fase successiva, rivelatasi più complicata rispetto alle altre, è stata la scrittura del codice nell'IDE. Come accennato nei precedenti capitoli, il codice è essenziale per garantire il corretto funzionamento dell'MKR1000 WiFi, in sua assenza il microprocessore risulterebbe essere privo di utilità. Per poter eseguire correttamente il collegamento WiFi, la trasmissione dei dati dal sensore e la comunicazione OSC sono state utilizzate alcune librerie:

- "MPU6050.h" [12]: questa libreria, sviluppata da Electronic Cats, è stata progettata appositamente per l'utilizzo del sensore MPU-6050 attraverso una scheda Arduino. Permette di utilizzare più funzioni per la ricezione dei dati sotto forma di vettori o valori, dando l'opzione di ricavarli anche in coordinate polari.
- "I2Cdev.h" [13]: libreria per la comunicazione I2C tra scheda e dispositivi esterni sviluppata da jrowberg. Al suo interno si possono trovare numerose classi e funzioni per interfacciarsi facilmente con il protocollo di comunicazione. Questa libreria è ottimizzata per poter funzionare con Arduino e molti altri microcontrollori (Raspberry PI, MSP430, ESP32 eccetera).
- "ArduinoOSCWifi.h" [14]: libreria sviluppata da hideakitai per l'implementazione del protocollo OSC in Arduino o altre schede. Fornisce metodi e funzioni per la creazione di messaggi o pacchetti di dati delle principali tipologie supportate dal protocollo di comunicazione, agevolando l'invio e la loro ricezione.
- "Wire.h" [15]: libreria creata da Arduino per la comunicazione I2C, in questo caso è necessaria per poter utilizzare alcune funzioni definite in "I2Cdev.h".

La scrittura del codice ha richiesto qualche settimana, essendo disponibili molte librerie per le funzioni elencate si è provato ad utilizzarne diverse, purtroppo rivelatesi non idonee. Con il passare del tempo sono state trovate le quattro librerie descritte poco fa, dopo un'attenta lettura della documentazione e alcuni tentativi si è riuscito ad ottenere il funzionamento desiderato. Qui sotto viene riportato il codice, opportunamente commentato (NB: i commenti sono preceduti dai caratteri "//"), procedendo successivamente con la sua analisi.

```

//Implementazione librerie
#include <ArduinoOSCWiFi.h>
#include "I2Cdev.h"
#include "MPU6050.h"
#include <Wire.h>

//Parametri accelerometro
MPU6050 accelgyro; // creazione istanza MPU-6050
const int MPU_addr=0x68; // indirizzo I2C di default
int16_t AcX, AcY, AcZ; // variabili accelerometro
int16_t GiX, GiY, GiZ; // variabili giroscopio

//Parametri WiFi
const char* ssid = "*****"; //ID del router
const char* pwd = "*****"; //password del router
const IPAddress ip(*****); //indirizzo IP dell'Arduino
const IPAddress gateway(*****); //gateway Arduino
const IPAddress subnet(*****); //subnet Arduino

//Parametri per la comunicazione OSC
const char* host = "*****"; //indirizzo IP dell'host
const int recv_port = ****; //porta per la ricezione di messaggi
const int send_port = ****; //porta per l'invio di messaggi

//Funzione di conversione dei valori
float convert(int16_t c) {
    float fc = c;
    float val = (fc + 17000)/(34000);
    return val;
}

//setup
void setup() {
    //Inizializzazione comunicazione seriale
    Serial.begin(115200);
    delay(2000);
}

```

```

//Inizializzazione comunicazione I2C
#if I2CDEV_IMPLEMENTATION == I2CDEV_ARDUINO_WIRE
    Wire.begin();
#elif I2CDEV_IMPLEMENTATION == I2CDEV_BUILTIN_FASTWIRE
    Fastwire::setup(400, true);
#endif

//Inizializzazione istanza MPU-6050
accelgyro.initialize();

//Disconnessione WiFi precedente
WiFi.disconnect(true);
delay(1000);
WiFi.mode(WIFI_STA);

//Inizializzazione WiFi e tentativo di connessione alla rete
WiFi.begin(ssid, pwd);
WiFi.config(ip, gateway, subnet);
while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(500);
    static int count = 0;
    //blocca il programma se i tentativi sono più di 20
    if (count++ > 20) {
        Serial.println("Tentativo di connessione interrotto, riprovare");
        count = 0;
        exit(0);
    }
}
//Stampa indirizzo IP WiFi dopo la connessione
Serial.print("WiFi connesso, IP = ");
Serial.println(WiFi.localIP());
Serial.println("Invio dei messaggi OSC");
}

```

```

//Istruzioni cicliche
void loop() {
  //Funzione da chiamare per la ricezione e l'invio di messaggi OSC
  OscWiFi.update();

  //Funzione per ricavare i dati dell'accelerometro
  //le variabili sono passate per riferimento
  accelgyro.getMotion6(&AcX, &AcY, &AcZ, &GiX, &GiY, &GiZ);

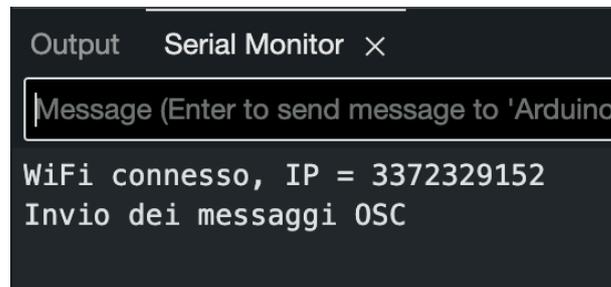
  //Conversione valori
  float aXval = convert(AcX);
  float aYval = convert(AcY);
  float aZval = convert(AcZ);
  float gXval = convert(GiX);
  float gYval = convert(GiY);
  float gZval = convert(GiZ);

  //Invio dei dati tramite OSC
  OscWiFi.send(host, send_port, "/aX", aXval);
  OscWiFi.send(host, send_port, "/aY", aYval);
  OscWiFi.send(host, send_port, "/aZ", aZval);
  OscWiFi.send(host, send_port, "/gX", gXval);
  OscWiFi.send(host, send_port, "/gY", gYval);
  OscWiFi.send(host, send_port, "/gZ", gZval);
}

```

Si può suddividerlo in tre sezioni principali, una per la dichiarazione di variabili e funzioni, una per il setup e l'altra per il loop. Le ultime due sono sempre presenti nei codici di Arduino, servono rispettivamente per l'inizializzazione del programma e lo svolgimento ciclico delle funzionalità desiderate. La prima non è una vera e propria sezione, è uno spazio esterno a setup() e loop() utilizzato solitamente per la creazione di istanze di oggetti, dichiarazione di variabili universali e scrittura delle proprie funzioni. Si è deciso di inserire i commenti all'interno del codice così che chiunque lo legga sia in grado di capire la sua struttura e le funzionalità delle righe; l'aggiunta di commenti è comunque buona norma in ambito informatico. La quasi totalità dei metodi utilizzati provengono dalle librerie, l'unico scritto da zero è il metodo convert(), creato per convertire facilmente i valori letti dall'accelerometro, numeri interi a 16 bit in un range compreso da circa -17000

a +17000, in numeri decimali (float: floating point data type) compresi tra 0 e 1, evitando una ridondanza all'interno del codice. Si possono riconoscere molti degli elementi descritti nel capitolo precedente, le variabili AcX, AcY, AcZ, GiX, GiY, GiZ per i dati raccolti dall'MPU-6050, tre per accelerometro e giroscopio, l'indirizzo IP dell'host, le porte di invio e ricezione per i messaggi OSC e, in aggiunta, ID e password del WiFi utilizzato. Molti di questi si ritrovano nella sezione di setup e di loop passati come parametri all'interno delle funzioni. All'interno del loop vengono eseguite ciclicamente le istruzioni per ricevere i dati dall'accelerometro, convertirli e inviarli al ricevitore. È evidente che l'utilizzo di librerie esterne ha permesso di semplificare notevolmente il codice, sono infatti bastate un centinaio di righe per il suo completamento. Alla fine del setup vengono stampate due frasi per la verifica del funzionamento del codice, producendo sul serial monitor dell'IDE l'output della figura 2.6.



```
Output Serial Monitor ×
Message (Enter to send message to 'Arduino
WiFi connesso, IP = 3372329152
Invio dei messaggi OSC
```

Figura 2.6: output serial monitor dell'IDE.

2.5.3 Ricezione OSC nella DAW

L'ultimo passo per il completamento del progetto è stato la ricerca di un modo per permettere la ricezione dei messaggi OSC all'interno di Ableton Live. Il software in sé non implementa il protocollo OSC, per trovare una possibile soluzione è stata effettuata una ricerca online. In poco tempo è stata scoperta l'esistenza del Connection Kit prodotto da Ableton [16], un bundle di plugin Max for Live dedicati esclusivamente alla comunicazione OSC. Dopo averlo scaricato nel computer è iniziato il test dei dispositivi in esso contenuti. A prima vista sembrava che avessero tutti la sola possibilità di inviare messaggi OSC, il che in parte è vero, l'unico che sembrava essere in grado di riceverli è il plugin OSC TouchOSC. TouchOSC è un'applicazione disponibile per iOS e Android costituita da un'interfaccia simile a un mixer, di fatto è stata programmata apposta per fungere da mixer remoto, integrando dentro di sé, come suggerisce il nome, il protocollo OSC. Il plugin omonimo sembrava potesse comunicare solo con questa applicazione, si è deciso quindi di provare a inviargli dei messaggi dall'MKR1000 WiFi e il risultato ottenuto è

stata una corretta ricezione di questi ultimi. Dopo l'esito positivo di questa prova si è scelto di utilizzare il plugin Max for Live OSC TouchOSC, visibile in figura 2.7, essendosi rivelato più che funzionale. È risultato perfetto sia per il progetto che per la creazione dell'esempio che verrà trattato nel capitolo successivo, infatti la sua interfaccia presenta dei pulsanti digitali "Learn" per dividere in più canali i messaggi OSC ricevuti e dei pulsanti "Map" per mapparli direttamente ai parametri della DAW.



Figura 2.7: plugin digitale Max for Live OSC TouchOSC.

Capitolo 3

Un esempio pratico

Dopo aver analizzato la fase di progettazione, parlando dei componenti e del loro funzionamento, viene proposto ai lettori un esempio pratico di quanto discusso finora.

3.1 Sintetizzatore digitale e sound design

In seguito alla verifica della corretta ricezione dei messaggi OSC all'interno di Ableton è iniziato il processo creativo per la sintesi di un suono e la mappatura di alcuni suoi parametri. Già all'interno della DAW si possono trovare dei sintetizzatori presenti di default, nella creazione di questo suono, invece, si è scelto di utilizzare Serum, prodotto dall'azienda Xfer [17]. Fin dalla sua uscita nel 2014, Serum si è dimostrato essere all'avanguardia rispetto a tutti gli altri sintetizzatori digitali presenti sul mercato. La sua interfaccia è piuttosto ricca, tuttavia si dimostra essere estremamente intuitiva, ovviamente per chi ha un minimo di esperienza in ambito musicale. Si possono vedere molte manopole e alcune schermate, le più importanti sono sicuramente le due centrali, in cui si trovano gli oscillatori da cui poter generare il suono (OSC A e OSC B). A sinistra se ne possono vedere altri due, uno chiamato SUB, progettato per suoni a bassa frequenza, l'altro chiamato NOISE, per la generazione di rumore bianco o di altri tipi. Serum offre un gran numero di wavetable (rappresentazioni digitali di una forma d'onda), passando da quelle base (sinusoide, onda quadra, onda triangolare, dente di sega) ad altre estremamente complesse. In alto a destra si trova il filtro, come per le wavetable se ne possono trovare di tutti i tipi (passa basso, notch, combs e molti altri), con la possibilità di scegliere su quali oscillatori applicarlo. Nella parte inferiore sono presenti gli involucri e gli oscillatori a bassa frequenza (LFO: Low Frequency Oscillator), fondamentali per dare dinamicità al suono collegandoli alle varie manopole per eseguire delle variazioni, cicliche o non, nel tempo. In alto, di fianco alla sezione OSC (in questo caso abbreviativo di oscillatore), si trova la sezione

FX dedicata esclusivamente agli effetti audio come saturazione, equalizzazione, riverbero e altri, seguita da MATRIX e GLOBAL che non verranno discusse, essendo molto più avanzate rispetto alle prime due. Da ultime, in basso a sinistra, si possono vedere quattro manopole nominate MACRO, la cui funzione consiste nel permettere la mappatura dei parametri di Serum da dispositivi esterni. In figura 3.1 è riportata l'interfaccia di Serum.



Figura 3.1: sintetizzatore Serum.

Nell'esempio proposto è stato scelto di sintetizzare un basso modulato in frequenza (FM: Frequency Modulation). Per poter sentire al meglio il funzionamento del DMI, durante la sintesi del suono sono state effettuate delle variazioni nel tempo di alcuni parametri tramite un LFO, unito a tre MACRO per la modulazione da parte dell'utilizzatore. Sono stati impiegati i due oscillatori principali, entrambi generanti una sinuoside di cui una a bassa frequenza e l'altra a frequenza più alta ma con volume d'uscita nullo, fungendo da modulante per la modulazione in frequenza. Un LFO a 0.5 Hz è stato collegato alla manopola FM (FROM B) del primo oscillatore per modificare la percentuale d'uscita del suono modulato in frequenza e non, variandola nel tempo periodicamente, come si può vedere in figura 3.2.



Figura 3.2: sintesi FM tramite gli oscillatori A e B.

La prima delle tre MACRO utilizzate è stata collegata alla frequenza dell'LFO potendola così variare a piacimento, il tutto visibile in figura 3.3.



Figura 3.3: LFO.

Dato che il secondo oscillatore ha volume d'uscita nullo, è stato utilizzato un filtro passa banda a 12 dB/oct (oct: ottava) solo per l'output del primo, creando un picco di risonanza in base alla frequenza di taglio. La seconda MACRO è stata connessa alle manopole CUTOFF e RES del filtro, le quali indicano rispettivamente frequenza di taglio e ampiezza del picco di risonanza. In figura 3.4 è riportata un'immagine del filtro con le sue manopole.



Figura 3.4: filtro passa banda e rispettive manopole.

Nella parte degli effetti si possono trovare un saturatore per distorcere il suono (NB: in ambito musicale, soprattutto per quanto riguarda la produzione, la distorsione è utilizzata frequentemente a scopo artistico) e un riverbero per dargli una percezione di ampiezza e lontananza. La terza e ultima MACRO è legata alle manopole DRIVE del saturatore e MIX del riverbero in modo da variarne la percentuale applicata al suono, come riportato in figura 3.5.



Figura 3.5: saturatore e riverbero all'interno di Serum.

Il criterio di assegnazione delle MACRO è stato effettuato in modo da permettere all'utilizzatore del DMI di sentire facilmente una variazione nel suono in seguito a un cambiamento dei loro valori. Riconducendosi al capitolo precedente, la modifica dei valori delle MACRO si effettua tramite OSC TouchOSC; essendo loro tre in totale, anche i valori inviati tramite OSC sono dello stesso numero, per ottenere un buon controllo del suono tramite l'interfaccia esterna si è scelto di trasmettere alla DAW i valori dell'inclinazione degli assi x e y dell'accelerometro (/X e /Y in OSC TouchOSC) e della rotazione attorno all'asse z del giroscopio (/Z in OSC TouchOSC). Per completare questo esempio è stata creata una clip MIDI in Ableton Live in cui viene suonata una nota singola (un fa) e nella sezione dedicata agli effetti audio della clip sono stati inseriti Serum, per generare il suono di cui si è parlato con le opportune MACRO riportate nell'interfaccia della DAW, insieme a un'istanza di OSC TouchOSC, con i valori ricevuti mappati alle MACRO. Per non confondere le MACRO, sono state rinominate in modo da ricordare la loro funzione nella modulazione del suono: MACRO 1 in LFORATE, MACRO 2 in FILCUTRES e MACRO 3 in DRIVEREVMIX, come si può vedere in figura 3.6. Dopo aver caricato il



Figura 3.6: mappatura dei valori ricevuti da OSC TouchOSC alle MACRO di Serum.

codice nell'Arduino è possibile iniziare ad utilizzare il DMI. Inclinando il sensore lungo l'asse x si possono sentire le frequenze alte del suono oscillare a velocità diverse grazie all'LFO, mentre se viene inclinato lungo l'asse y si sentirà una specie di sweep in frequenza dato dallo spostamento del picco di risonanza del filtro. Infine girando il sensore lungo il suo piano, perpendicolarmente all'asse z, è possibile aumentare le quantità di riverbero e distorsione introdotte.

3.2 Possibili migliorie future del progetto

Nei precedenti capitoli sono stati trattati molti argomenti, si è parlato dell'evoluzione dei DMI nell'ultimo ventennio, analizzandone i principali aspetti che li caratterizzano, è stata successivamente discussa la fase di progettazione di un nuovo DMI introducendo i componenti di cui è costituito e mostrandone il funzionamento complessivo, concludendo con una dimostrazione pratica. Facendo un resoconto si può capire che i margini di miglioramento di questo progetto sono ampi. La prima miglioria consiste nell'aggiunta dell'elemento che caratterizza il progetto e da cui ne deriva il nome, l'astrolabio. Dato il costo elevato, è stato scelto di eseguire una sua stampa 3D. I vettoriali sono completi, come si può vedere nelle figure 3.7 e 3.8, appena il modello fisico sarà disponibile verrà subito inserito nel progetto, collegando sotto di lui il sensore e la scheda in modo che possa fungere da interfaccia per l'utilizzatore.

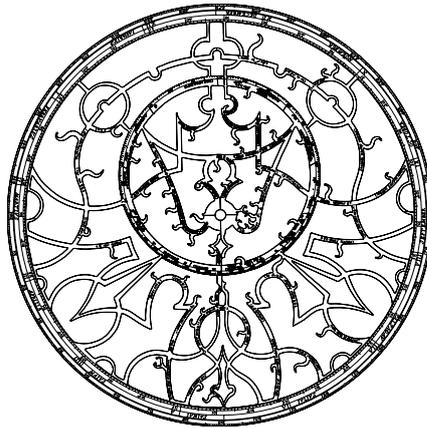


Figura 3.7: primo vettoriale dell'astrolabio.

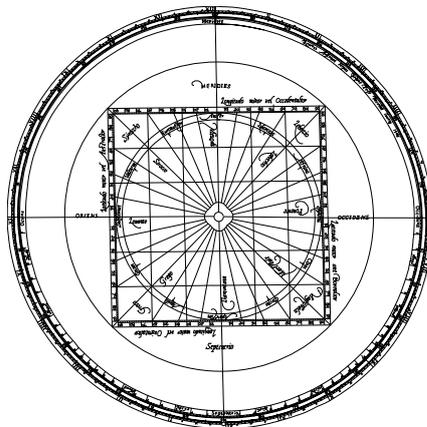


Figura 3.8: secondo vettoriale dell'astrolabio.

Oltre a questo un'idea ottima sarebbe la sostituzione del sensore GY-521 MPU-6050 e dell'Arduino MKR1000 WiFi con un unico PCB in grado di svolgere le loro funzioni, così facendo si otterrebbe un modulo compatto con la possibilità di saldarlo sotto l'astrolabio, riducendo in contemporanea il volume occupato complessivo e il consumo di potenza. Si potrebbe progettare il PCB con il fine di ampliare le funzionalità dello strumento, aggiungendo per esempio altre tipologie di sensori o dei componenti elettronici fisici come pulsanti o leve. Banalmente un sensore di distanza permetterebbe di eseguire delle nuove modulazioni del suono semplicemente alzando o abbassando l'astrolabio rispetto al terreno, allo stesso modo dei sensori di pressione sulla sua superficie renderebbero possibile un nuovo tipo di modulazione toccando lo strumento in punti diversi. Si potrebbe invece pensare di focalizzarsi sull'aspetto esteriore dell'astrolabio. Di fatto le stampe 3D sono tendenzialmente monocromatiche, un rivestimento esterno applicato con criterio migliorerebbe di molto l'impatto visivo sugli osservatori. In aggiunta si potrebbero inserire dei LED RGB collegati al PCB così che possano accendersi, spegnersi e cambiare colore in base ai dati ricavati dai sensori. Tutte le migliorie discusse in questa sezione possono portare dei grandi vantaggi al DMA, data la loro facilità nell'implementazione si potrebbe pensare di introdurle in un futuro non lontano.

Capitolo 4

Conclusioni

L'ingegneria è una disciplina nata con il fine di migliorare la vita dell'uomo tramite l'applicazione di conoscenze scientifiche e matematiche. Grazie al suo approccio basato sull'analisi e la risoluzione di problemi è sempre stata e continuerà ad essere un caposaldo per l'innovazione e il progresso tecnologico. Pur essendo una disciplina puramente razionale, il suo legame con la cultura è risultato evidente, in questo documento è stato possibile vedere la creazione di uno strumento musicale attraverso un approccio analitico basato sulla scomposizione dei problemi in problemi più piccoli. Questo è l'approccio tipico di un ingegnere, tramite la risoluzione di questi è possibile arrivare a progettare e sviluppare prodotti innovativi e all'avanguardia, sia in ambito scientifico che culturale. Nella creazione del DMA sono state applicate molte conoscenze diverse, passando da quelle tipiche dell'elettronica per il corretto utilizzo di sensori e microcontrollori a quelle informatiche per la creazione di un codice finalizzato al loro funzionamento, potendo classificare il lavoro complessivo come un approccio alla robotica. Anche l'aspetto artistico ha giocato un ruolo importante, permettendo la generazione di un suono tramite l'applicazione di concetti di sound design e sintesi digitale. Dato il rapido progresso tecnologico degli ultimi decenni sarà possibile sviluppare progetti sempre più articolati e sofisticati, permettendo la loro unione alle discipline culturali senza limitarsi alla sola musica.

Bibliografia

- [1] A. Refsum Jensenius e M. J. Lyons, *A NIME Reader*. Springer Cham, 2017, ISBN: 978-3-319-47213-3.
- [2] P. Cadavid Hinojosa, «KNOTTING THE MEMORY//ENCODING THE KHI-PU/_:»», 2022. indirizzo: <https://phaidra.kunstuni-linz.at/view/o:2944>.
- [3] «MPU-6050 datasheet.» (2013), indirizzo: <https://invensense.tdk.com/wp-content/uploads/2015/02/MPU-6000-Datasheet1.pdf>.
- [4] «Arduino official website.» (2005), indirizzo: <https://www.arduino.cc/en/Guide/Introduction>.
- [5] «Arduino official store.» (2005), indirizzo: https://store-usa.arduino.cc/products/arduino-mkr1000-wifi?srsltid=AfmB0ooCq3ivbevg-pUkDd2_1I7UsW0DSD9A5qVWHj5PCQCJfTkJI1pY.
- [6] «Arduino official website.» (2005), indirizzo: <https://www.arduino.cc/en/software>.
- [7] «Arduino official website.» (2005), indirizzo: <https://www.arduino.cc/reference/en/>.
- [8] «Ableton official website.» (2013), indirizzo: <https://www.ableton.com/en/live/>.
- [9] «Ableton official website.» (2013), indirizzo: <https://www.ableton.com/en/live/max-for-live/>.
- [10] «OpenSoundControl website.» (2021), indirizzo: <https://opensoundcontrol.stanford.edu/>.
- [11] «AUTODESK instructables website.» (2020), indirizzo: <https://www.instructables.com/GY-521-MPU6050-3-Axis-Acceleration-Gyroscope-6DOF/>.
- [12] «Pagina github di ElectronicCats.» (2024), indirizzo: <https://github.com/ElectronicCats/mpu6050>.

- [13] «Pagina github di jrowberg.» (2023), indirizzo: <https://github.com/jrowberg/i2cdevlib/tree/master>.
- [14] «Pagina github di hideakitai.» (2024), indirizzo: <https://github.com/hideakitai/ArduinoOSC>.
- [15] «Arduino official website.» (2005), indirizzo: <https://www.arduino.cc/reference/en/language/functions/communication/wire/>.
- [16] «Ableton official website.» (2013), indirizzo: <https://www.ableton.com/en/packs/connection-kit/>.
- [17] «Xfer official website.» (2014), indirizzo: <https://xferrecords.com/products/serum/>.