Haptic feedback in Internet of musical things

Czoguly Tünde

May 29, 2020

Cuprins

Lista de figuri

1 Introducere

1.1 Tema abordată

1.1.1 Internetul obiectelor muzicale

Internetul obiectelor este o platformă vastă și în plină dezvoltare care a evoluat datorită convergenței a mai multor tehnologii. Din ce în ce mai multe obiecte de zi cu zi sunt combinate cu conectivitatea la internet, astfel încât să poate colecta și schimba date între ele cu ajutorul unor senzori și actuatori. Scopul principal al acestor creații este desigur de a ne ușura și de a ne susține activitățile noastre de zi cu zi.

Internetul lucrurilor muzicale este derivat din acest domeniu, fiind o ramură relativ nouă și emergentă, mai mult bazată de cercetare și experimentare, având multe provocări în ceea ce privește latența. După cum se poate observa și din nume, rolul principal este jucat de domeniul muzicii, de instrumente și tot ceea ce ține de producție și recepție muzicală, fie că este vorba despre compoziție,de procesul de învățare sau doar de pură distracție. Întrucât majoritatea instrumentelor pot fi conectate la internet, există numeroase posibilități care facilitează atât publicul cât și interpretul pentru o performanță cât mai bună pe scenă și o interacțiune cât mai bună între audiență și interpreți.[9]

Sistemul senzorial joacă un rol decisiv în ceea ce privește experiența noastră la un concert, fie că este vorba despre muzică electrică sau clasică. Pe lângă simțul auditiv și cel vizual ar trebuie inclusă pentru a avea o satisfacție cât mai mare la sfârșit. Cu ajutorul internetului lucrurilor muzicale această experiență poate fi îmbunătățită și chiar poate veni în ajutor la persoanele cu dizabilități. Acesta este și scopul principal al acestei lucrări, de a creea un dispozitiv purtabil cu ajutorul căruia oamenii cu deficiență de auz să poată resimți vibrațiile muzicii astfel având și ei parte de o experiență mai bogată in ceea ce privește muzica. De ce întocmai această temă? Pentru mine muzica este și a fost mereu o parte foarte importantă din viața mea, eu cântând la pian de când mă știu. Datorită faptului că știam că vreau să fac ceva bazat pe IoT [1], marea provocare a fost de a găsi o modalitatea de a îmbina acesta cu pasiunea mea astfel încăt să iasă ceva interesant și frumos. Am avut două direcții de orientare: să fac ceva pentru persoana care cântă la un instrument sau pentru audiență. Ideea de a face un dispozitiv pentru oamenii cu deficiență de auz nu este o idee originală însă nici una foarte populară nu e.

[1] Internet of Things

S-au făcut experimente cu astfel de wearable device-uri pe piață cu rezultate pozitive însă cele mai multe n-au fost scoase la vânzare, au rămas numai pe plan experimental. Fiind un subiect atât de rar abordat decizia mea a fost să experimentez și eu acest topic. Astfel am ales ca instrument un pian electric, din motive evidente.

1.1.2 Pe scurt despre muzica în comunitatea surdă

Pentru noi este un lucru firesc și obișnuit de a aude sunetele din jurul nostru și mai ales de a asculta muzică fie pe telefon sau prin participarea la concerte. Într-o situație puțin mai neplăcută se află cei surzi, care cu toate că pot comunica prin limbajul semnelor, muzica pentru ei nu e ceva ce pot auzi, ci mai degrabă este ceva ce pot simți numai cu ajutorul vibrațiilor amplificate. În cazul lor celelalte simțuri, prin plasticitatea creierului, lucrează împreună pentru a compensa pierderea auzul [5], simțul lor tactil astfel fiind mult mai dezvoltat.

O persoană cu deficiență de auz care cântă la un instrument muzical are o altă percepție asupra muzicii decât audiența. În cazul unui interpret canalul haptic este implicat într-o buclă complexă de acțiune. Muzicantul interacționează fizic cu instrumentul, pe de o parte, pentru a genera sunet, iar pe de altă parte, pentru a recepționa și percepe răspunsul fizic al instrumentului care este simțit sub forma unei vibrații [7]. În cazul audienței surde acest set de vibrații nu este resimțită sau este este resimțită cu o intensitate foarte slabă. Crearea unui dispozitiv cu ajutorul căruia s-ar simți aceste vibrații intensificate ar duce la o experiență mult mai bogată în ceea ce privește participarea la concerte live a comunității surde.

1.2 Prezentarea capitolelor

În primul capitol al acestei lucrări sunt prezentate aspecte generale despre tot ce înseamnă internetul obiectelor muzicale în comunitatea surdă. Totodată este și descrisă motivația alegerii acestei teme de licență, greutățile întâmpinate cât și posibilele extinderi a acestei lucrări.

Cel de a doilea capitol are ca scop informarea cititorului despre tehnologiile hardware folosite în realizarea lucrării. Sunt amănunțite detalii despre placa de dezvoltare RaspberryPi și funcționalitatea motoarelor de vibrații folosite la crearea dispozitivului purtabil.

În capitolul al treailea sunt prezentate toate tehnologiile software folosite atât în crearea aplicației de pe placa de dezvoltare cât și cele folosite în programarea aplicației Android. Este rezervat și o secțiune unde se explică folosirea mesajelor MIDI[2] transmise de la instrumentele muzicale.

Capitolul al patrulea este rezervat pentru prezentarea aplicației. Este prezentat separat aplicația python de pe placa de dezvolatre RaspberryPi și separat aplicația Android pentru dispozitivul respectiv.

Capitolul cinci are ca scop prezentarea unei concluzii în ceee ce privește proiectarea, programarea și testarea acestui dispozitiv pentru oamenii cu deficiență de auz.

În ultimul capitol este prezentat bibliografia folosită în această documentație.

1.3 Rezultate și Contribuții personale

1.4 Posibile extinderi

Posibilitatea de extindere a acestei aplicații este destul de mare ținând cont că dispozitivul creat este doar un prototip și Interentul lucrurilor muzicale are un potențial de creștere destul de mare, fiind un domeniu mai nou.

Datorită faptului că acest proiect a fost făcut să redea vibrațiile unui singur instrument muzical, din motive clare, este firesc ca această funcționalitate să poate să fie extinsă pe mai multe instrumente, nu numai pian. Astfel utilizatorul n-ar simți numai numai vibrațiile pianului ci mai degrabă ar simți un cumul de vibrații de la mai multe instrumente. Având o aplicație Android unde utilizatorul poate regla în momentul de față intensitățile și poate alege ce fel de vibrație vrea să simtă, îndată cu această extindere și această aplicație ar avea mai multe opțiuni de personalizare. Extindere asta ar implica și schimbări de hardware. În momentul de față este folosit un singur Raspberry Pi atât pentru culegrea datelor din pian cât și pentru punerea în funcțiune a motoarelor. Cazul idealar fi ca fiecare instrument să aibă propria lui placă, la fel ca și device-ul purtabil. Provocarea care trebuie rezolvată în acest caz este latența care se va inpune din cauza faptului că datele vor trebui să fie transmise de la un dispozitiv la altul, asta implicând o nesincronizare între muzica de pe scenă și vibrațiile simțite.

[2]Musical Instrument Digital Interface

2 Tehnologii hardware folosite

2.1 Placa de dezvoltare RaspberryPi

2.1.1 Prezentare generală

Când vine vorba despre un calculator, mulți dintre noi se gândesc la PC-uri clasice sau la laptopuri. Cu toate că aceste dispozitive fac o treabă excelentă în rezolvarea task-urilor, nu sunt o alegere bună mai ales când vrem să lucrăm cu senzori pentru a culege date sau dorim să punem în funcțiune niște actuatori pe baza datelor colectate. Pentru astfel de operațiuni, și nu numai, ai fost create plăcile de dezvoltare sau SBC [3]. Un astfel de calculator este și Rasbperry Pi-ul care a fost creat în Marea Britanie cu scopul de a promova predarea informaticii în școlile și țările în curs de dezvoltare [4] dar poate fi utilizat pentru o varietate de scopuri precum și robotica sau într-o gamă largă de aplicații în domeniul IoT. Aceste plăci de mărimea a unui card de credit au devenit repede populare din cauza dimensiunilor și a accesibilității în ceea ce privește prețul acestora.

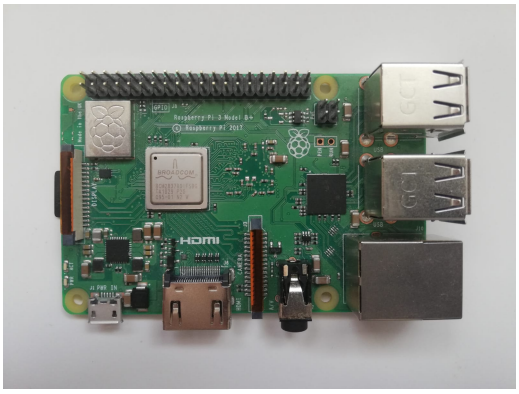

Figura 1: Raspberry Pi 3 model B+

În decursul anilor s-au lansat mai multe versiuni al acestei plăci, fiecare venind cu o îmbunătățire pe partea de software sau hardware. Pentru realizarea acestui proiect eu am folosit un Raspberry Pi 3 model B+ care, din punctul meu de vedere, este un model perfect de a începe a lucra la proiecte Iot. Datorită faptului că este un single-board computer, componentele sale sunt plasate într-un singur circuit imprimat. Acest circuit sprijină mecanic și conectează electric componenetele electrice folosind piste conductoare. În

[3] Single-board computer

cazul acestei plăci printre componentele atașate pe circuit se poate enumăra microprocesorul, memoria RAM [4], diferitele porturi de intrare și ieșire sau pinurile GPIO [5].

2.1.2 Folosirea pinurilor

Pe fiecare model de placă RaspberryPi se pot observa de-a lungul marginii superioare pinii GPIO. Pe plăcile curent se află 40 de astfel de pini, fiecare putând fi programat ca un pin de intrare sau ieșire având astfel șansa de a le utiliza într-o gamă largă de scopuri.

Există câte două pinuri cu un voltaj de 5, respectiv 3.3 și opt pini la sol care nu pot fi configurate. Celelalte sunt pinuri generale de 3V3 asta înseamnând că la ieșire sunt setate la 3V3 și la intrare sunt tolerate la 3V3 [2]. O caracteristică importantă a acestori pini generali este modularea lățimii pulsurilor (PWM[6]) cu ajutorul căruia putem controla dispozitivele analogice cu o ieșire digitală. Cu un control digital putem realiza numai o undă pătrată adică un semnal de oprit și pornit. Acest fel de model poate simula tensiuni de 3.3V și 0V, adică pornire și oprire completă.

Fiecare pin în parte (înafară de cele de ground, de 3V3 și 5V) au un număr specific care vine în ajutor în momentul programării acestora.

2.1.3 Conectarea la internet

Conectarea plăcii Raspberry Pi la internet se poate face în două feluri: prin cablu de internet, placa având mufă de internet, sau prin WIFI. În abele cazuri, dacă vrem să accesăm linia de comandă a plăcii respective, conectarea se face prin prin SSH[7]. Pentru asta trebuie aflată adresa IP a plăcii respective. Acest lucru se poate face în mai multe feluri, existând mai multe programe de căutare a adreselor IP.

2.1.4 Conectarea plăcii la instrumente muzicale

Conectarea unei plăci Raspberry Pi la un instrument muzical se face cu ajutorul unui cablu MIDI USB. Pentru a fi posibilă această operațiune instrumentul respectiv trebuie să fie electric și să suporte interfață digitală. Partea

[4]Random access memory
[5]General-purpose input/ouput
[6]Pulse Width Modulation
[7]Secure Shell

USB a cablului respectiv se va conecta la unul dintre porturile plăcii de dezvoltare iar cealaltă parte se va conecta la porturile MIDI ale instrumentului respectiv. Partea conectată la pian a cablului se termină în mai multe conectoare DIN [8] cu cinci pini de 180°. Un singur concetor ar transporta mesajele

Figura 2: Mufă MIDI

doar într-o singură direcție, deci pentru a avea o comunicare în două sensuri avem nevoie de cel puțin două astfel de conectori (vezi figura 2). Pentru că majoritatea instrumentelor nu copiază mesajele în porturile de ieșire, există și varinate de cablu care au trei mufe, al treilea fiind cel de THRU care este destinat pentru transmiterea datelor către un alt instrument. În cazul nostru sunt deajunse concetorii de IN și OUT. Însă trebuie avut grijă pentru că etichetarea acestor mufe ca fiind IN și OUT ne poate duce în eroare când vine vorba despre conectare pentru că acestea nu vor funcționa dacă sunt conectate la aceleași porturi MIDI etichetate pe un instrument electronic. Acest lucru se datorează faptului că fluxul de date indicat pe mufe arată direcția în care datele vor curge către calculator, nu portul de pe instrument la care trebuie conectat fiecare cablu [8]. Deci mufa IN se va conecta la portul OUT și mufa OUT la cel de IN.

2.2 Actuatori folosiți pentru realizarea feedback-ului haptic

2.2.1 Motoare de vibrați LRA

Motoarele de vibrațiii pot fi de mai mult tipuri astfel alegerea unui tip de motor depinde în cea mai mare parte de cerințe, de unde și pentru ce va fi

[8]Deutsches Institut für Normung

folosit. Există două mari categorii de astfel de motoare și anume ERM și LRA. Diferența dintre acestea, înafară de mărimea lor, este modul în care funcționează. Primul tip de motor folosește o masă mică dezechilibrată pe un motor cu curent continuu atunci când se rotește creând o forță care se traduce prin vibrații. Servomotoarele rezonante liniare (LRA) conțin o masă internă mică atașată la un arc, care creează o forță atunci când sunt conduse de curent [1]. În tehnologiile unde e prezent feedback-ul haptic, de cele mai multe ori se folosesc motoare de tip LRA din cazua mărimii și a faptului că modularea amplitudinii este foarte ușoară. În realizarea acestui proiect s-au folosit mai multe motoare de vibrații de acest tip pentru obținerea vibrațiilor de intensitate dorită.

Modul în care aceste motore sunt puse în funcțiune este prin conectarea firelor la pinurile corespunzătoare de pe placa Raspberry Pi. acestor culor este de a ști la ce fel de pini trebuie conectați: culoarea roșie se conectează la un pin de tip GPIO, iar cel negru la pamânt. Cu ajutorul PWM putem controla valoarea la care aceste motoare vor vibra încât și frecvența acestora. Modelul folosit de mine are firele de culoare negru și roșu, dar sunt și motoare

Figura 3: Motor de vibrații LRA

care merg pe varianta de roșu și albastru. Ideea principală a acestor culori este de a ști la ce pini trebuie conectați: culoarea roșie se conectează la un pin GPIO iar cea neagră la pământ (ground pin).

3 Tehnologii software folosite

3.1 Sistemul de operare Raspbian

Raspbian este un sistem de operare pentru placa de dezvoltare Rasberry Pi, bazat pe Debian, o distribuție a sistemului de operare Linux.

3.2 Protocolul de comunicare MIDI

3.2.1 Prezentarea mesajelor MIDI

Comunicarea între un instrument muzical electric și un calculatori se întâmplă de cele mai multe ori printr-un protocol de comunicare numit MIDI (Digital Interface for Musical Interface). Prin acest protocol putem trimite sau primi date de la instrumentul respectiv sub forma unor mesaje MIDI, fiecare astfel de mesaj fiind o instrucțiune. Putem să ne gândim la ele ca fiind o altfel de partitură în care sunt marcate înălțimea și puterea (velocitatea) cu care notele muzicale au fost cântate. Aceste date primite se pot reda audio sau se pot chiar modifica facând acest tip de comunicare un instrument puternic în industria muzicală.

Un mesaj MIDI constă dintr-un octet de stare, care indică tipul mesajului, urmat de până la doi octeți de date care conțin parametrii. Tipul mesajului poate fi de mai multe feluri, acestea variând de la un instrument la altul. Principalele tipuri de mesaje sunt cele de NOTE ON sau NOTE OFF, care indică faptul că interpretul a atins tastatura muzicală respectiv când s-a dat drumul la tastă. Diferența de timp dintre aceste două tipuri ne poate da timpul în care a fost ținută o notă muzicală. Cu toate că aceste mesaje sunt unele standard trebuie avut în vedere că nu fiecare instrument poate genera astfel de mesaje. Instrumentele mai vechi au o gamă mai închisă in ceea ce privește multitudinea de tipuri de mesaje ce pot trimite.

Parametrii unui mesaj MIDI, după cum s-a discutat și în primul paragraf, pot conține nota muzicală și velocitatea cu care aceasta a fost cântată. Reprezentarea acestei note muzicale se face printr-un număr de la 0 pana 127 [3], 0 fiind nota cea mai joasă cu cea mai mică frecvență. Astfel de exemplu numărul 60 se asociază notei Do cu frecvența 261.63. Puterea cu care nota respectivă a fost apăsată se caracterizează printr-un număr de la 1 la 127 [3], o apăsare decentă (fără prea multă forță) fiind aproximativ 64. Datorită faptului că pianul folosit în acest proiect are o reprezentare imprecisă a acestui număr, în aplicație ne vom folosi numai numărul notei MIDI

din care vom calcula frecvența, folosind o formulă, cu care trebuie să vibreze motoarele când o clapă e apăsată. Când vine vorba despre programarea acestor mesaje, modul în care sunt reprezentate depinde de librăria folosită în limbajul de programare respectiv. Despre parsarea acestora vom detalia în subcapitolele ce urmează.

3.3 Limbajul de programare Python

3.4 Prezentare generală

Python este un limbaj de programare devenit foarte popular datorită faptului că este proiectat să fie ușor de învățat și de înțeles. Este un limbaj de scriptare ceea ce înseamnă că codul scris nu este compilat ci mai degrabă interpretat. Ca și celelalte limbaje de nivel înalt și Python suportă mai multe paradigme de programare cum ar fi programare procedurală, orientată pe obiect sau funcțională sprijinind astfel dezvoltarea unei game largi de aplicații. Poate fi folosit atât pentru realizarea proiectelor de dimensiuni mai mici cât și pentru cele mai mari, fiind cel mai des utilizat în crearea aplicațiilor web (este folosit de companiile Google și Yahoo!), științifice sau de divertisment, cum ar fi jocuri. Este foarte mult folosit și în proiectele ce țin de Internetul lucrurilor, având destule biblioteci ce pot fi folosite pentru programarea senzorilor și actuatorilor.

Faptul că limbajul de programare Python oferă o gamă largă de biblioteci și extinderi, constituie un avantaj în programarea aplicațiilor complexe. Biblioteca standard a limbajului este deseori citat și ca unul dintre cele mai mari puncet forte ale sale, oferind multe tool-uri pentru difeite sarcini.

Sintaxa și semantica limbajului de programare sunt concepute astfel încât codul să fie cât mai ușor de scris și de înțeles de către programator. Spre deosebire de alte limbaje de programare, Python nu folosește acolade pentru delimitarea blocurilor, aceste delimitări făcându-se cu ajutorul indentării. Un cod indenatat incorect va arunca eroare de tipul IndentationError. Se poate vedea în figura 4 o bucată de cod indentat corect. O scăderii a indentării seminfică sfârșitul unui bloc curent în timp ce creșterea indică începerea unei noi afirmații. Un statement (afirmație) este o unitate sintactică a limbajelor de programare imperative care exprimă unele acțiuni care trebuie efectuate.

În Python, ca și în celelalte limbaje de programare, putem face diferența dintre o metodă și o funcție. Ambele trebuie să înceapă cu cuvântul cheie def urmat de denumirea funcției/metodei. Spre deosebire de alte limbaje de pro-

```
def _setValues(self, value, switch_type):
    print("In the set values method")
    print(str(value) + str(switch_type))
    if value == MESSAGES[Message.ON]:
        DEFAULT_VALUES[switch_type] = True
    elif value == MESSAGES[Message.OFF]:
        DEFAULT_VALUES[switch_type] = False
```

Figura 4:

gramare, în cazul metodelor în Python trebuie specificat explicit cuvântul self în parametrii metodei. Acest cuvânt indică instanța clasei de care aparține metoda respectivă.

3.4.1 Biblioteca Gpiozero pentru programarea pinurilor

Gpiozero este o bibliotecă care oferă o interfață simplă pentru programarea GPIO-urilor de pe placa de dezvoltare Raspberry Pi. Spre deosebire de alte biblioteci, aceasta oferă o curbă de învățare mai lină pentru începători și posibilitatea dezvoltării proiectelor complicate cu mai multă ușurință și mai puține linii de cod. La început Gpiozero a fost doar un layout peste biblioteca RPi.GPIO, ulterior fiind adăugate diferite suporturi. În prezent RPi.GPIO este folosit ca bibliotecă implicită pentru Gpiozero, această configurație putând fii schimbată, existând ma multe astfel de biblioteci, cum ar fi de exemplu pigpio. Instalarea acestei biblioteci este necesară numai în cazul în care se folosește pe alt sistem de operare decât Raspbian, pe acesta fiind deja inclus. Odată cu instalarea ei putem rula comanda pinout din linia de comandă pentru a vedea vizual detalii despre pinii disponibili de pe Raspberry Pi. Acest lucru devine în ajutor în momentul în care, la configurarea pinurilor, trebuie să precizăm numărul pinului ales astfel fiind mult mai ușor de urmat poziționarea fiecărui pin în parte.

În comparație cu alte biblioteci, în Gpiozero se pote observa o abordare orientată pe obiecte, acesta folosind clase în loc de funcții. Pe lângă niște clase specifice, Gpiozero oferă două clase mari de bază, InputDevice și OutputDevice, celelalte fiind derivate din acestea, fiecare având metode și proprietăți specifice care sunt adaptate dispozitivului controlat. Clasa OutputDevices și tot ce derivă din ea este folosită pentru controlarea dispozitivelor de ieșire cum ar fi de exemplu LED-urile, pe când cea de InputDevice este pentru dispozitivele care sunt folosite pentru a furniza date și semnale

de control. Asftel alegerea clasei de lucru este mult mai vizibilă și mult mai ușoară.

Pentru controlarea motoarelor de vibrații în proiectul de față s-au folosit clasa PWMOutputDevice derivat din OutputDevice care permit și modularea lățimii pulsului. Importarea bibliotecii se face simplu, conform figurei 5. În momentul instanțierii clasei, în constructor putem da de la unu pana la

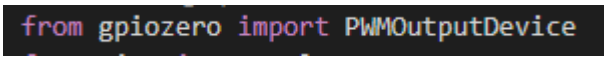

Figura 5: Importare biblioteca Gpiozero

cinci parametrii, numai primul fiind obligatori, celelalte având deja valorile setate implicit. Parametrul obligatoriu este numărul pinului de pe Raspberry Pi cu care lucrăm. Numerotarea pinurilor în cazul acestei biblioteci se face cu ajutorul sistemului Broadcom. Ceilalți parametrii ai constructorului sunt pentru a seta valoarea de început (duty cycle), frecvența, dacă să fie activ sau nu și nu în ultimul rând putem schimba fabricii de pini, asta fiind o caracteristică avansată pe care majoritatea utilizatorilor o pot ignora. Un exemplu de instanțiere a acestei clase se poate vedea în figura 6, unde parametrul gpio indică numărul pinului setat. .

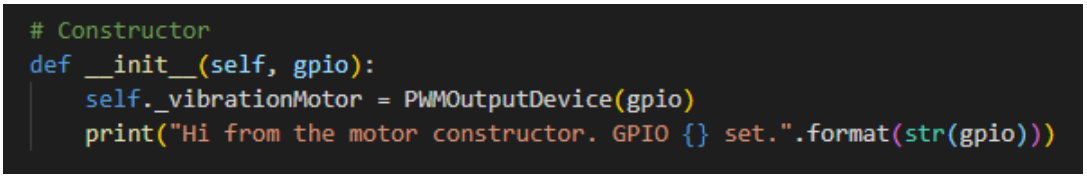

Figura 6: PWMOutputDevice constructor

Datoriă faptului că clasa permite modularea lățimii pulsurilor, se pot seta intensități între valorile 1 și 0, 1 fiind intensitatea maximă cu care motorul poate să vibreze și 0 fiind starea în care se oprește din vibrat. Totodată se poate seta și frecvența, aceasta implicit fiind 100Hz, numărul acesta indicând rata de apariție a unui eveniment repetitiv [6]. Exemplu de setare a acestor valori poate fi observat în figura 7. Clasa dispune mai multe metode și proprietăși cu ajutorul cărora putem porni sau opri dispozitivul. În cazul nostru este deajuns să setăm proprietățile value și frequency pentru a da drumul sau a opri vibrațiile.

```
    self._vibrationMotor.value = value
    self._vibrationMotor.frequency = frequency
```

Figura 7: Setarea valorilor motorului

Biblioteca Gpiozero pune la dispoziție și o clasă Tone destinată utilizatorilor care folosesc TonalBuzzer pentru a putea reprezenta cu ușurință notele muzicale. Cu toate că în proiectul de față nu este folosit un astfel de dispozitiv, clasa asta ne poate deveni în ajutor la calcularea frecvențelor pornind de la numărul notei MIDI transmis de pian. Datorită faptului că clasa poate fi construită într-o varietate de moduri, asta incluzând și prin numere MIDI, se poate cu ușurință obține frecvența notei respective cu ajutorul atributului frequency. Codul din spatele clasei calculează frecvența cu formula

$$f = 2^{(d-69)/12} * 440Hz$$

unde parametrul d reprezintă numărul notei muzicale MIDI. Pe lângă nota MIDI, clasa se poate construi pornind și de la frecvență sau de la specificarea notei prin sistemul alfabetic (acesta consistând dintr-o literă majusculă de la A până la G urmat de un modificator opțional și numărul de octave). Folosirea acestei clase se poate vedea în figura 8.

```
def convert_midi_number_to_frequency(midiNumer):
    try:
        tone = Tone(midi=midiNumer)
        return tone.frequency
    except:
        print("Operation problem (convert midi number to freuency)")
```

Figura 8: Calcularea frecvenței

3.4.2 Folosirea bibliotecii Mido

Parsarea și reprezentarea datelor primite de la instrumentul muzical este un pas important în ceea ce privește rularea acestui proiect. Din fericire limbajul de programare Python ne oferă mai multe posibilități prin care putem să ajungem la același rezultat, depinde doar de noi cu ce vrem să lucrăm și care modalitate se potrivește task-ului respectiv. Pentru citirea, parsarea și modificare mesajelor/fișierelor MIDI există mai multe tipuri de biblioteci, fiecare făcând în mare parte același lucru, cu puține diferențe. Biblioteca cu

care s-a lucrat în acest proiect se numește Mido și datorită documentației bine pusă la punct este relativ ușor de urmărit ce metode/funcționalități are. Pe parcursul dezvoltării proiectului am încercat să lucrez cu mai multe astfel de tipuri de biblioteci, însă cele mai multe s-au dovedit a fi mai greu de folosit, ducând lipsă de funcționalități sau având numai metodele minime necesare.

Instalarea bibiotecii pe sistemul de operare Raspbian se face cu executarea comenzii pip install mido, iar pentru utilizarea porturilor trebuie instalată și biblioteca rtmidi cu ajutorul comenzii pip install python-rtmidi. Rtmidi este backend-ul folosit în mod implicit care se poate schimba cu altele, cum ar fi PortMidi sau Pygame. Este recomandat să se folosească rtmidi din datorită faptului că este mult mai ușor de instalat și are toate caracteristicile celor-lalte biblioteci.

Pentru a putea primi date de la pian, trebuie deschis un port de intrare specificând numele portului respectiv (a se vedea figura 9). Pentru a putea

```
def __init__(self, port):
    self._midiIn = mido.open_input(port)
```

Figura 9: Port de intrare

găsi numele portului există metoda `mido.get_input_names()` a bibliotecii, care afișează toate proturile disponibile. Sunt incluse și alte tipuri de porturi în bibliotecă înafară de cele de intrare și de ieșire, precum multi portul care ne permite să citim și să scriem mesaje pe mai multe porturi, SocketPort-ul sau IOPort-ul care este o combinație între cel de intrare și ieșire. După nevoi se pot implementa și altfel de tipuri.

Mesajele MIDI vor veni pe acest port deschis iar în funcție de arhitectura instrumentului muzical acestea pot veni fără întreruperi, asta însemnând că programul primește date chiar și atunci când nicio o tastă nu a fost apărată pe pian, sau pot veni doar doar când a fost cântată o notă muzicală. În funcție de asta poate fi implementată metoda de parsare a acestor mesaje. În cazul nostru pianul trimite date încontinuu asta însemnând că vor trebui selectate mesajele de care suntem interesate. Iterarea peste mesaje se întâmplă cu ajutorul unui for până când portul se va închide. Pentru o parsare mai ușoară, mesajele de intrare vor fi mai întâi convertate la octeți MIDI ceea ce înseamnă că mai departe se va lucra cu un vector de valori, fiecare valoare corespându-se cu un parametru a notei MIDI respective. Un exemplu de astfel de parsare

se poate vedea în figura 10. Vectorul găsit conține 3 parametrii printre care

```
for message in self._midiIn:
    if bridge.ev.is_set():
        self._processEvent()
    bytes_array = message.bytes()
    if(self._is_pressed_note(bytes_array)):
        note, velocity = self._get_note_and_velocity(bytes_array)
        ansi_note = Helper.number_to_note(note)
```

Figura 10: Citire mesaje MIDI

ultimele două indică valoarea notei MIDI respectiv puterea cu care acesta a fost cântat/apăsat. Biblioteca mido suportă majoritatea tipurilor de mesaje descrise în prezentarea mesajelor MIDI.

3.4.3 Programarea asincrona cu biblioteca Asyncore

3.5 Limbajul de programare Java

3.5.1 Folosirea firelor de executie in Java

3.6 Sistemul de operare Android

3.6.1 Baza de date SQLite

4 Prezentarea aplicației

4.1 Aplicația RaspberryPi

4.1.1 Prezentare dispozitiv purtabil

4.2 Aplicația Android

4.2.1 Prezentare UI

5 Concluzii

6 Bibliogragie si webografie

[1] An Introduction To Vibration Motors. https://www.precisionmicrodrives.com/vibration-motors/.

[2] GPIO. https://www.raspberrypi.org/documentation/usage/gpio/.

[3] MIDI Tutorial. https://www.cs.cmu.edu/~music/cmsip/readings/MIDI\%20tutorial\%20for\%20programmers.html.

[4] James Cusick, William Miller, Nicholas Laurita, and Tasha Pott. Design, construction, and use of a single board computer beowulf cluster: Application of the small-footprint, low-cost, insignal 5420 octa board. *Arxiv*, 12 2014.

[5] Rachel Elaine. How Deaf People Experience Music, 2017.

[6] Michael Lombardi. *Frequency Measurement*, page 24 p. 02 1999.

[7] Stefano Papetti and Charalampos Saitis. *Musical Haptics: Introduction*, pages 1–7. 05 2018.

[8] Alexis Rohlin. How to Use MIDI Cable In & Out Plugs. https://smallbusiness.chron.com/use-midi-cable-out-plugs-49492.html.

[9] Luca Turchet, Carlo Fischione, and Mathieu Barthet. Towards the internet of musical things. 07 2017.