



ECOLE  
POLYTECHNIQUE  
DE BRUXELLES

B-IRCI :2 - 202122

TRAN-H201-INFO3

Rapport de projet

---

# Séquenceur multicanal à commande numérique pour synthétiseurs analogiques (semi-)modulaires

---

Lucas PLACENTINO

Felix ROSSO

Kamil PIETRASZ

Robin AFUERA

GILON Cedric

2021-2022

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>4</b>
<b>2</b>	<b>Conception</b>	<b>5</b>
2.1	Design . . . . .	5
2.1.1	Comparaison d'idées de designs . . . . .	5
2.1.2	Conclusion Design . . . . .	7
2.1.3	Composants . . . . .	9
2.2	Utilisation . . . . .	9
2.2.1	Diagramme d'utilisation . . . . .	9
2.2.2	Exemples de scénario . . . . .	11
2.3	Interface . . . . .	12
2.3.1	Encodeurs rotatifs et bouton . . . . .	12
2.3.2	Écran LCD . . . . .	12
2.3.3	Prises Jacks . . . . .	14
2.4	Électronique . . . . .	14
2.4.1	Composants . . . . .	14
2.4.2	Remarques . . . . .	15
2.4.3	Schéma . . . . .	16
<b>3</b>	<b>Implémentation</b>	<b>20</b>
3.1	Code . . . . .	20
3.1.1	Choix du langage . . . . .	20
3.1.2	Structure . . . . .	20
3.1.3	Librairies . . . . .	21

3.1.4	Fonctionnement et exécution . . . . .	21
3.1.5	Gestion d'erreur . . . . .	21
3.2	Précision du signal . . . . .	21
<b>4</b>	<b>Prototypage</b>	<b>22</b>
4.1	BOM (Bill Of Materials) . . . . .	22
4.2	Fabrication . . . . .	23
4.2.1	Photos de la fabrication . . . . .	23
4.2.2	Référence GPIOs . . . . .	26
4.3	Validation . . . . .	27
4.3.1	Tests . . . . .	27
4.3.2	Résultats . . . . .	27
4.3.3	Limitations . . . . .	28
<b>5</b>	<b>Gestion de projet</b>	<b>29</b>
5.1	SWOT . . . . .	29
5.1.1	SWOT début de projet . . . . .	29
5.2	Risks and Issues (risques et problèmes) . . . . .	29
5.2.1	Risks (risques) . . . . .	29
5.2.2	Issues (problèmes) . . . . .	30
<b>6</b>	<b>Conclusion</b>	<b>31</b>
	<b>Annexes</b>	<b>32</b>
<b>A</b>	<b>Code</b>	<b>33</b>

**Conception of a semi-modular synthesiser sequencer.**

*Mid-term Report*

**Conception d'un séquenceur de synthétiseur semi-modulaire.**

*Rapport de mi-parcours.*

## Table des figures

2.1	Premier design . . . . .	5
2.2	Deuxième design . . . . .	6
2.3	Troisième design . . . . .	6
2.4	Design intermédiaire (Dimensions/échelle non correctes.) . . . . .	7
2.5	Idée design final . . . . .	8
2.6	Design final séquenceur . . . . .	9
2.7	Diagramme de cas d'utilisation du séquenceur . . . . .	10
2.8	Exemple d'utilisation du séquenceur . . . . .	11
2.9	Écran principal . . . . .	12
2.10	Écran affichant le tempo . . . . .	12
2.11	Écran affichant la note du deuxième pas . . . . .	13
2.12	Écran affichant la longueur de la note . . . . .	13
2.13	Écran affichant le CV1 . . . . .	13
2.14	Écran affichant le CV2 . . . . .	13
2.15	Schéma électronique option 1 . . . . .	16
2.16	Schéma électronique option 2 . . . . .	18
3.1	Logo Python . . . . .	20
4.1	Premier assemblage d'une séquence de LEDs . . . . .	23
4.2	Premier assemblage du LCD avec la Raspberry Pi . . . . .	23
4.3	Premier assemblage avec les encodeurs rotatifs . . . . .	24
4.4	Premier test complet sur synthétiseur, encodeurs rotatifs, DACs et prises JACK . . . . .	25
4.5	Liste des GPIOs . . . . .	26

## Liste des tableaux

4.1 Bill Of Materials . . . . .	22
---------------------------------	----

Le projet étudié dans ce rapport se rapporte à la fabrication d'un séquenceur capable d'exécuter une séquence de 8 pas reposant sur une interface ergonomique et permettant une utilisation intuitive. L'utilisateur peut choisir de modifier des paramètres de base comme le tempo, la longueur et hauteur de chaque note (gamme comprise) de la séquence. De plus, trois tensions de commandes (CV1, CV2, CV3) sont disponibles afin de les relier à des paramètres choisis sur le séquenceur. Ces tensions de commande peuvent contrôler le filtrage fréquentiel des notes, leur enveloppe modulaire et leur vitesse d'oscillation.

Tout d'abord, en première partie seront abordés la conception du séquenceur, l'évolution de son design, ses fonctionnalités, son interface (dont l'écran LCD) et ses composants. Ensuite, la seconde section se réfère à l'implémentation du code : le langage, sa structure, les librairies utilisées et avant tout la démarche pour la précision du signal sont examinés. En troisième lieu la phase de prototypage et de fabrication est étudiée ainsi que les essais et leurs résultats. La partie finale porte sur la gestion du projet : le SWOT ainsi qu'un aperçu des risques et des problèmes du groupe de projet.

## 2.1 Design

Pour conceptualiser ce séquenceur, un design a été proposé par chaque membre du groupe puis a fait l'objet d'une comparaison.

### 2.1.1 Comparaison d'idées de designs

#### Premier design proposé

Ce design comprend 2 boutons qui ont pour but de changer de mode (step, tempo, pitch, ...) et un encodeur rotatif permettant de choisir la valeur de ce mode.

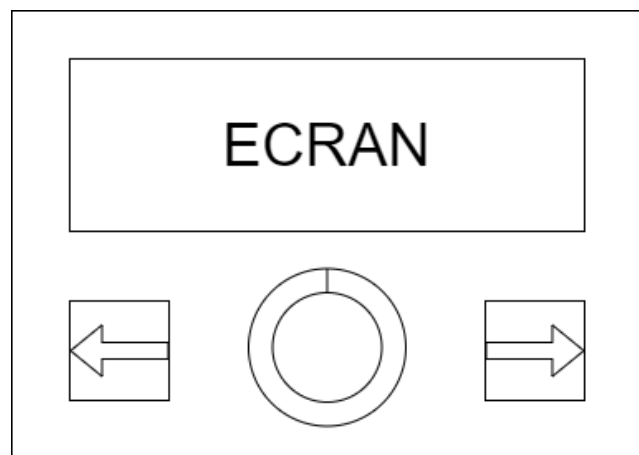


FIGURE 2.1: Premier design

Il a pour avantage de n'avoir que 3 encodeurs analogiques et donc un circuit électrique simple. Ses inconvénients sont que l'utilisation n'est pas très intuitive et que les possibilités de manipulation sont réduites.



### Deuxième design proposé

Ce design comporte un encodeur rotatif par mode et des LEDs qui indiquent le *step* actuel.

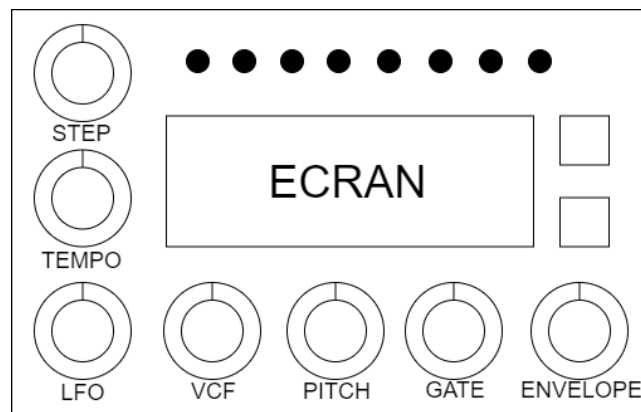


FIGURE 2.2: Deuxième design

En comparaison au précédent, il paraît plus intuitif et permet plus de possibilités de manipulation. Par contre, son circuit électrique sera plus complexe à modéliser et fabriquer.

### Troisième design proposé

Ce design a un fonctionnement similaire au précédent.

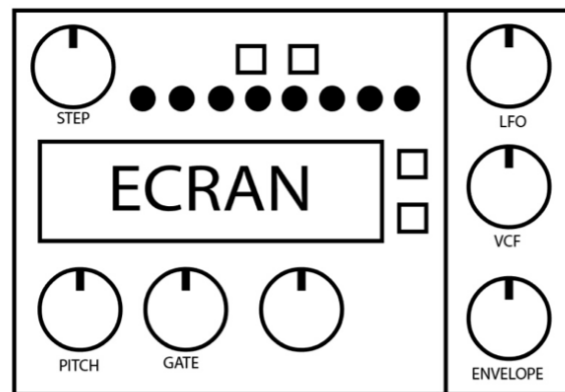


FIGURE 2.3: Troisième design

À la différence du second design, celui-ci n'a pas de bouton de réglage de tempo et a un affichage différent.

### 2.1.2 Conclusion Design

Le design des modèles 2 et 3 présente plus d'avantages et ressemblent davantage au séquenceur recherché. Il a donc été décidé de faire un mélange de ces deux derniers modèles.

La présence d'encodeurs rotatifs pour définir le tempo et les CVs est nécessaire. Mettre les commandes relatives à une note en bas de l'écran et les commandes relatives à l'ensemble des notes sur la droite paraît plus intuitif et logique. Un bouton "silence" a également été ajouté.

Voici le design intermédiaire du séquenceur :

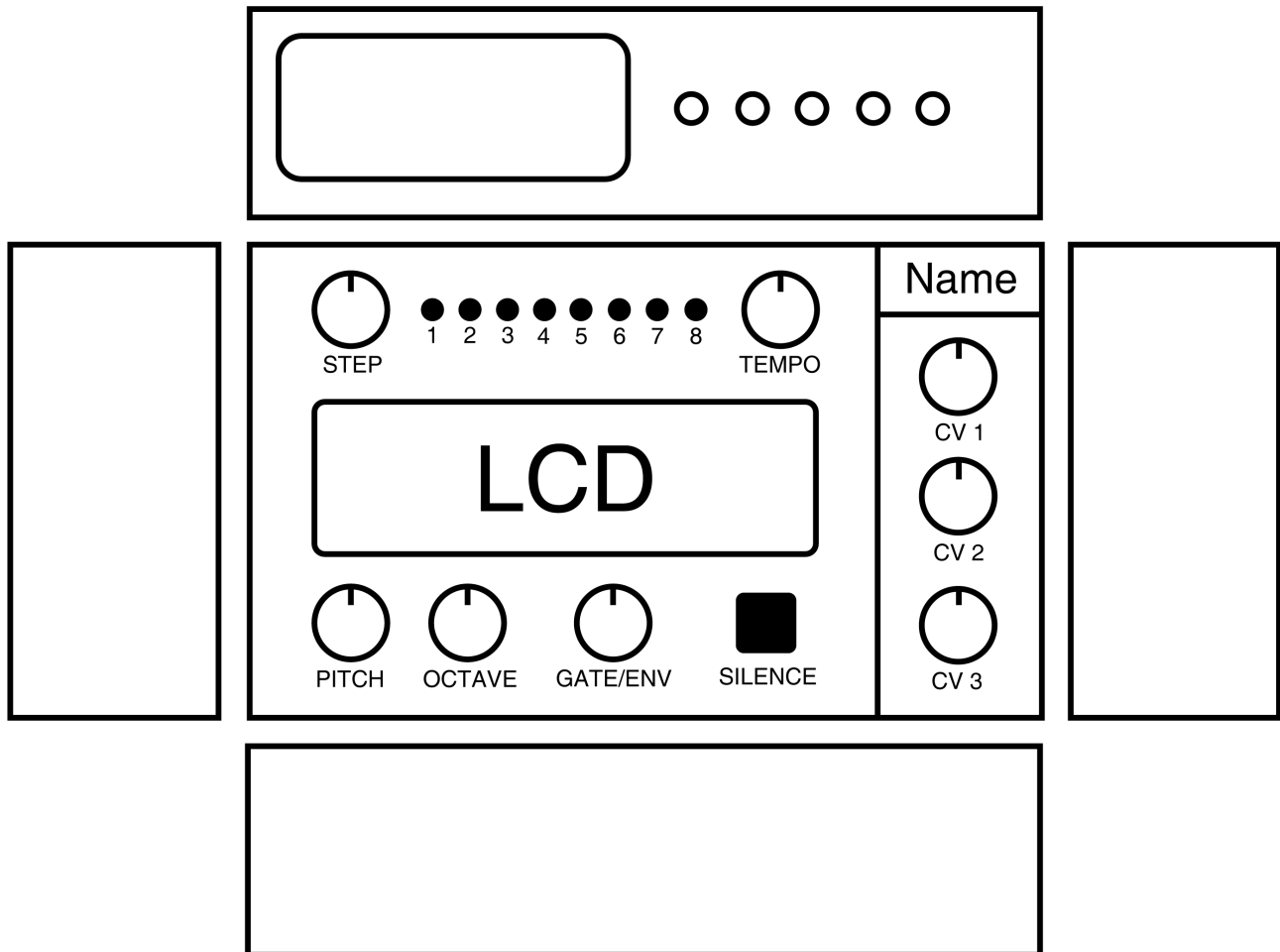


FIGURE 2.4: Design intermédiaire (Dimensions/échelle non correctes.)

Le design est encore à modifier pour rajouter ou changer :

- Un JACK IN et un bouton clock externe - tempo.
- Un JACK IN et un bouton play - pause.
- Des boutons à la place des encodeurs rotatifs pour le pitch, l'octave et la sélection du step.
- Des JACKs sur la face avant.

— Réarrangement en prenant en compte l'ergonomie pour les droitiers.

Comme cette idée :

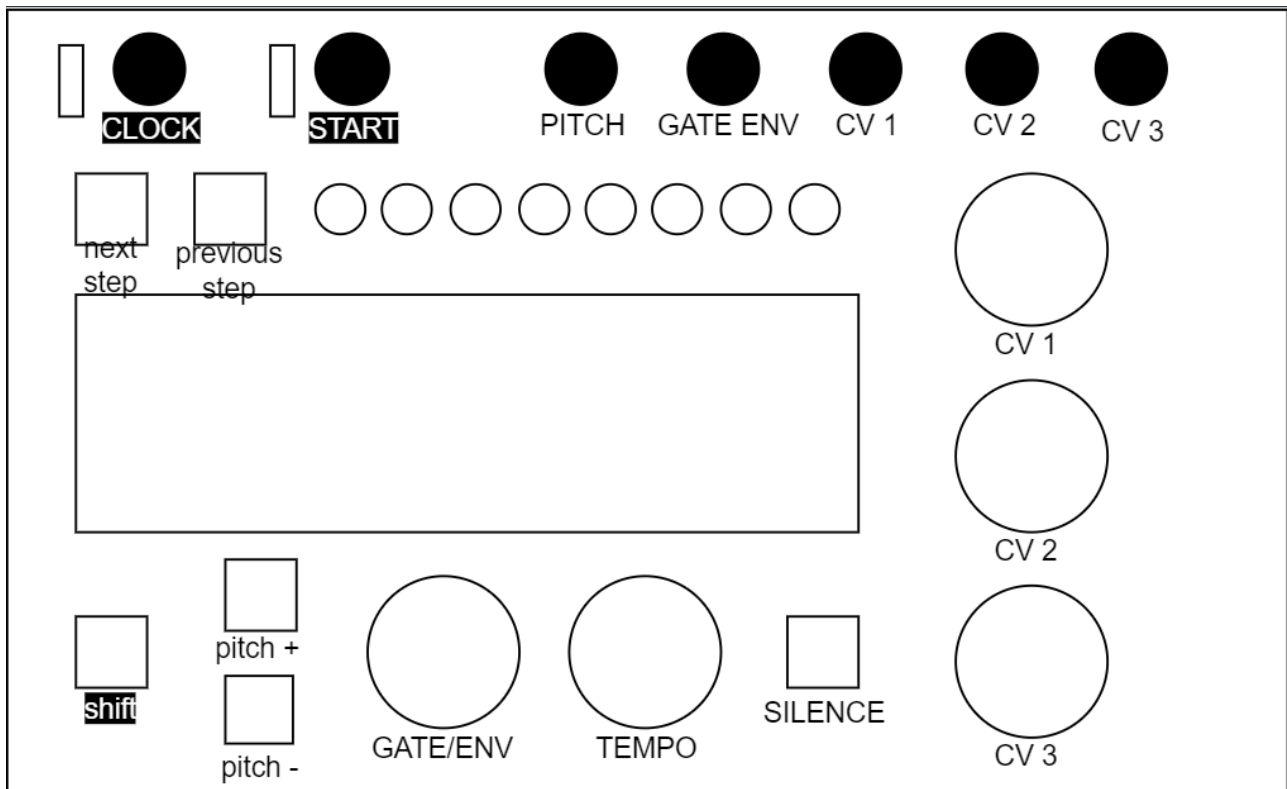


FIGURE 2.5: Idée design final

Enfin, voici le design final du séquenceur :

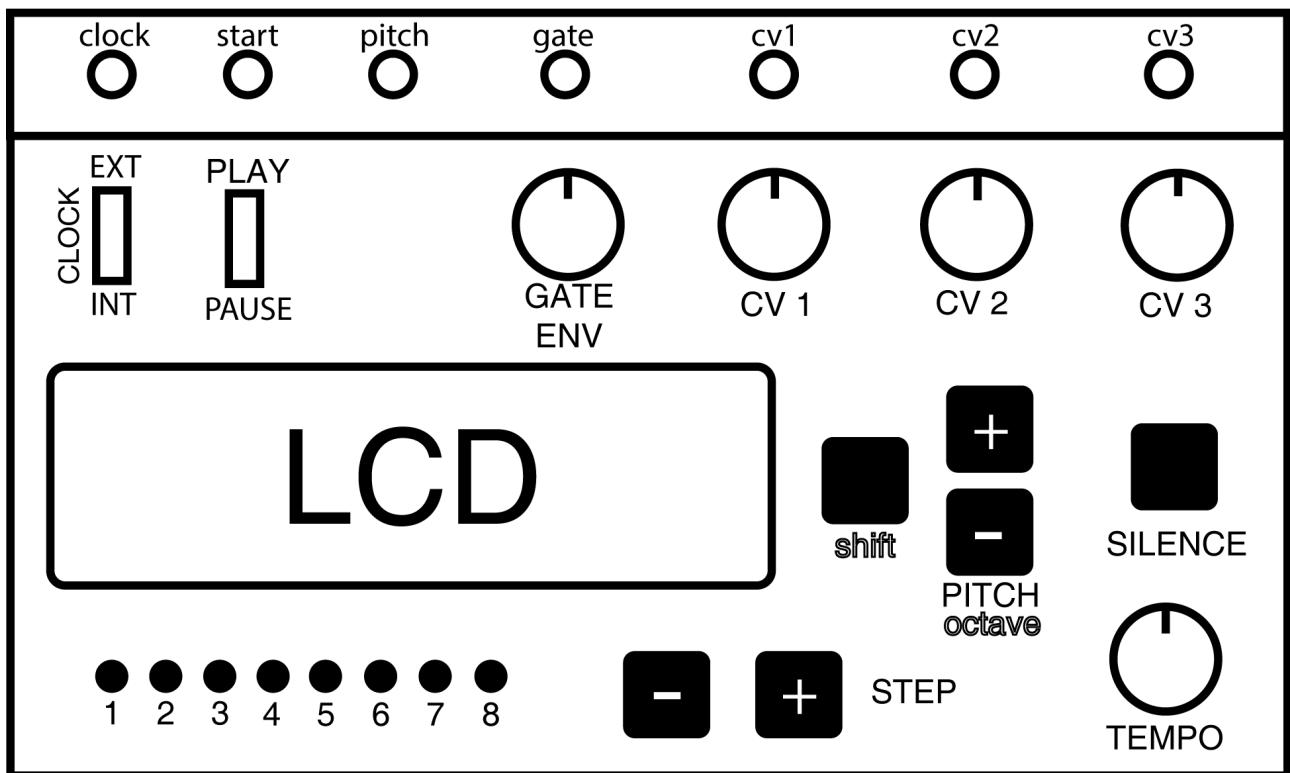


FIGURE 2.6: Design final séquenceur

### 2.1.3 Composants

Un Bill Of Materials se trouve au chapitre 4.1 (Fig. 4.1).

## 2.2 Utilisation

### 2.2.1 Diagramme d'utilisation

Le diagramme d'utilisation (Fig 2.7) décrit toutes les fonctionnalités du séquenceur.

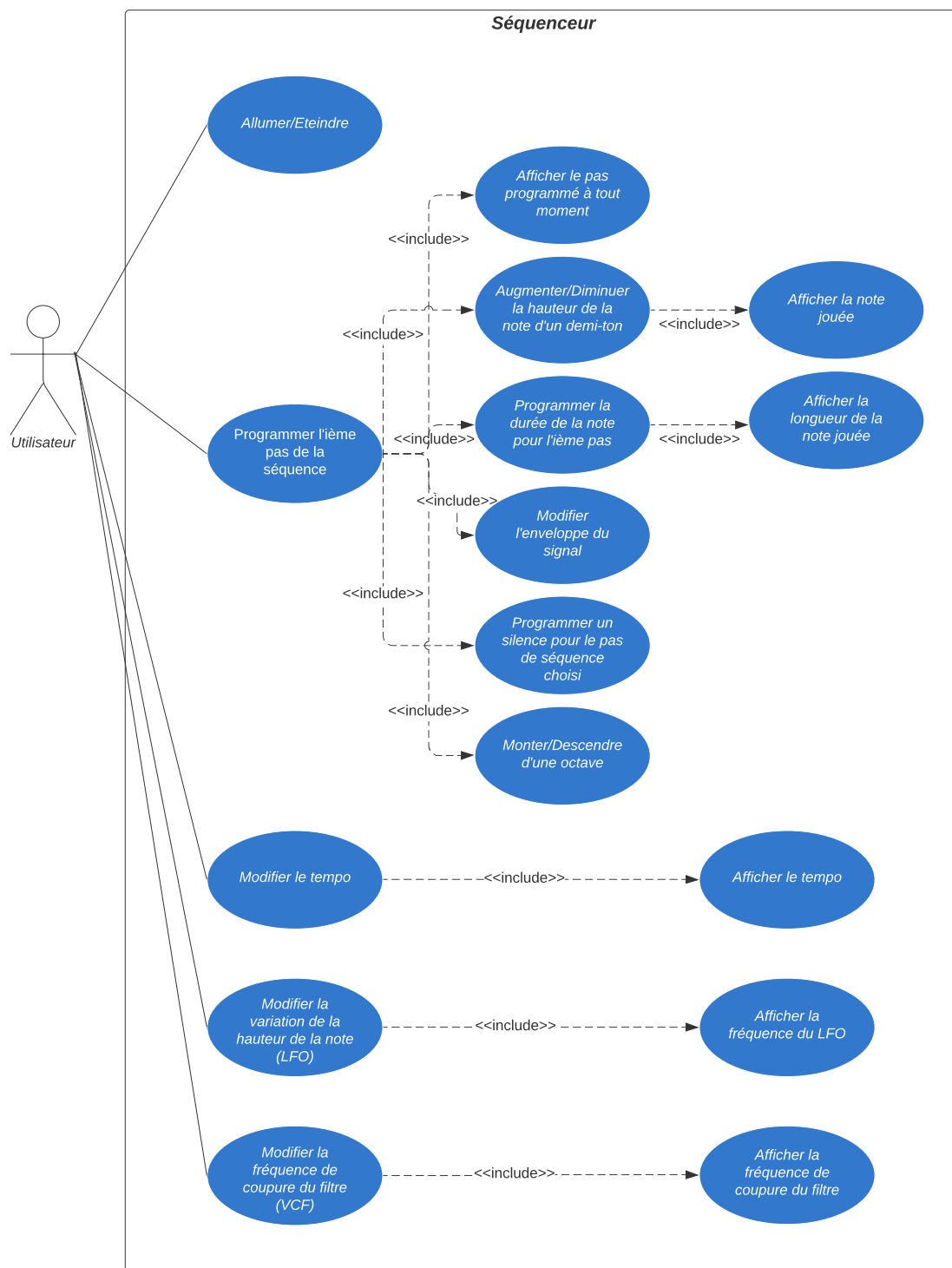


FIGURE 2.7: Diagramme de cas d'utilisation du séquenceur

## 2.2.2 Exemples de scénario

Voici des exemples de scénario d'utilisation du séquenceur.

- L'utilisateur veut définir le *step* n°7 en G3. Il va appuyer sur le bouton correspondant au septième *step*, puis va appuyer sur celui du *pitch* jusqu'à arriver à la note G affichée à l'écran. Enfin, il va utiliser le bouton shift pour modifier l'octave jusqu'à ce que l'écran l'affiche à 3. Le séquenceur va donc savoir que lorsqu'il envoie sa note du *step* n°7 au DAC, il enverra un enchaînement de bits qui sera traduit par le DAC en tension entre 0 et 5 volts qui correspond à la tension requise par le synthétiseur pour générer la bonne note (qui est G3).
- L'utilisateur veut modifier la fréquence *cutoff* du VCF à 200 Hertz. Il va tourner l'encodeur rotatif CV correspondant jusqu'à ce que l'écran lui affiche cette fréquence. Le séquenceur va alors envoyer un enchaînement de bits au DAC qui traduira cela en une tension comprise entre 0 et 5 volts qui correspond à la tension requise par le synthétiseur pour définir le *cutoff* du VCF à la bonne fréquence.
- L'utilisateur veut insérer un silence au *step* n°2. Il va donc sélectionner le bon *step* à l'aide de son encodeur rotatif, il va ensuite appuyer sur le bouton silence. L'écran lui affichera que le *step* n°2 est *mute*. Le séquenceur ne va donc envoyer aucun signal au DAC et au synthétiseur pour qu'aucun son ne soit joué à ce *step* n°2.

La figure 2.2 illustre un autre cas, plus général, d'une utilisation du séquenceur.

*Remarque* : L'ordre des actions une fois le pas sélectionné est arbitraire.

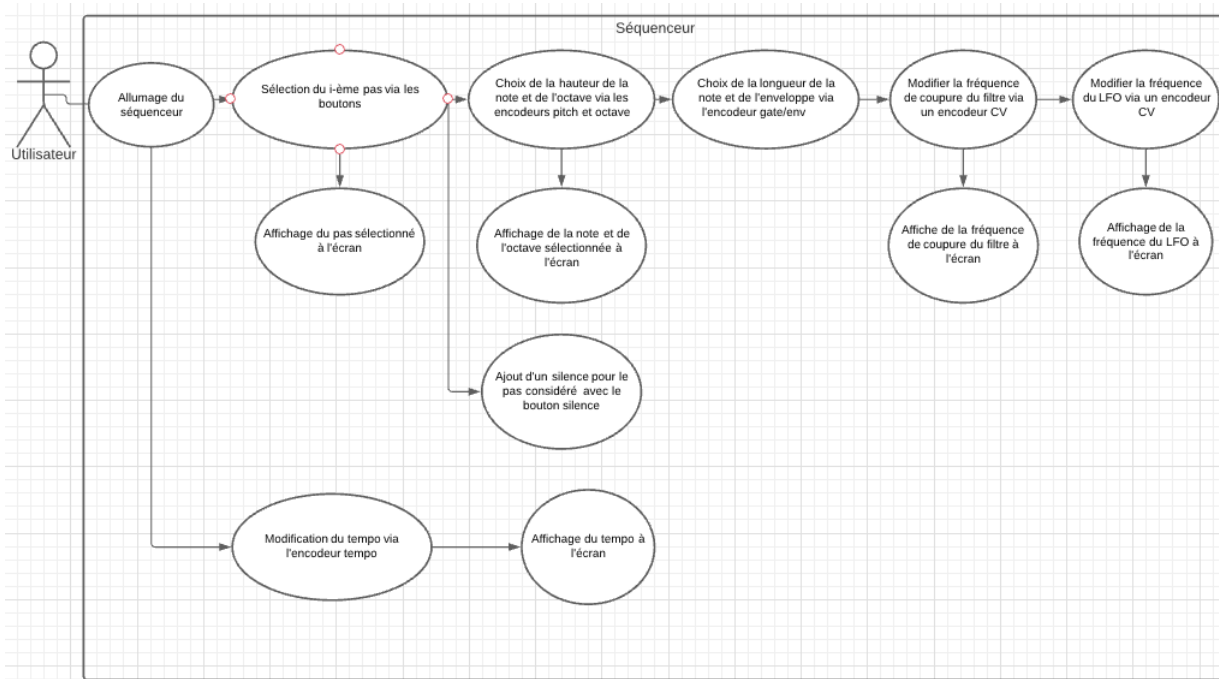


FIGURE 2.8: Exemple d'utilisation du séquenceur

## 2.3 Interface

L'interface du séquenceur repose sur des commandes ergonomiques (boutons, encodeurs rotatifs) et un écran LCD pour permettre à l'utilisateur de naviguer dans la séquence, tout en ayant la possibilité de modifier les paramètres déjà encodés.

### 2.3.1 Encodeurs rotatifs et bouton

L'utilisateur changera les paramètres du séquenceur à l'aide d'encodeurs rotatifs (crantés ou non). Il y a un encodeur rotatif pour chaque paramètre.

Un bouton sera aussi disponible pour ajouter un silence pour le pas considéré

### 2.3.2 Écran LCD

L'écran affichera le dernier paramètre modifié, avec sa valeur, et le pas concerné.

— Voici l'écran "principal" du séquenceur

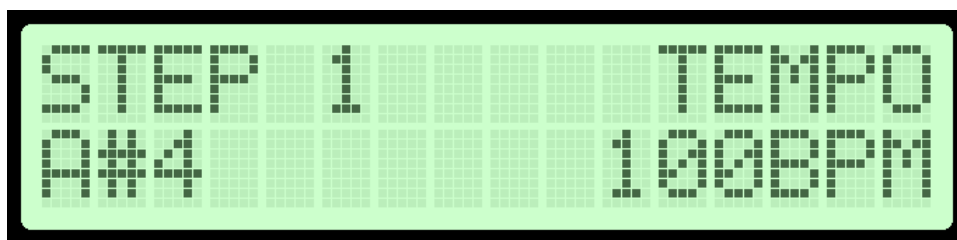


FIGURE 2.9: Écran principal

Voici ce que l'écran affichera si :

— Le tempo est réglé à 200 battements par minute.

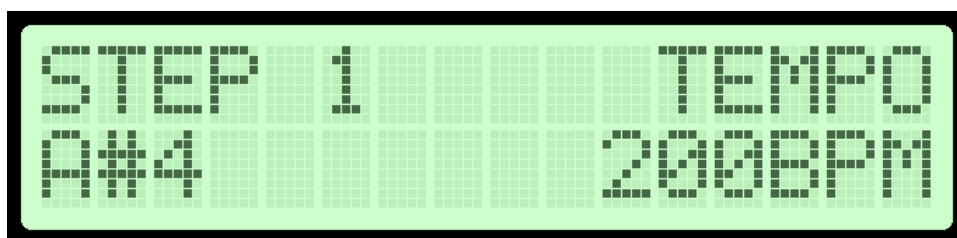


FIGURE 2.10: Écran affichant le tempo

— La note Mi-dièse (E#) de la troisième octave est assignée au deuxième pas.

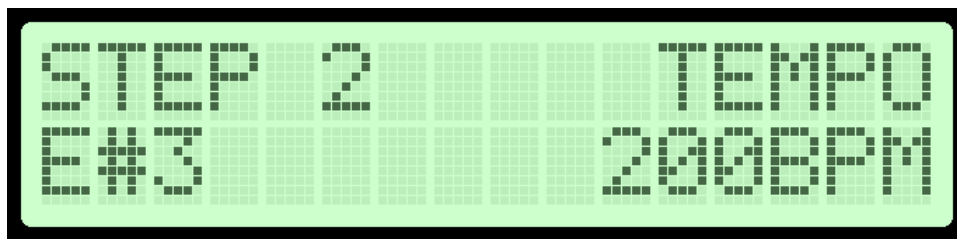


FIGURE 2.11: Écran affichant la note du deuxième pas

- La longueur de la note (GATE/ENV) est réglée à 10% d'un temps.

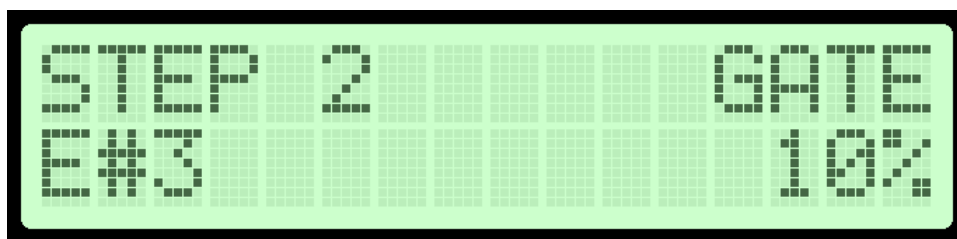


FIGURE 2.12: Écran affichant la longueur de la note

- Le CV1 (par exemple connecté au LFO du synthétiseur) est réglé à 50%.

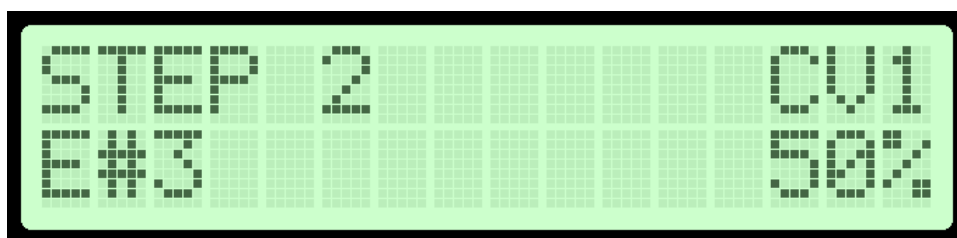


FIGURE 2.13: Écran affichant le CV1

- Le CV2 (par exemple connecté au VCF du synthétiseur) est réglé à 75%.

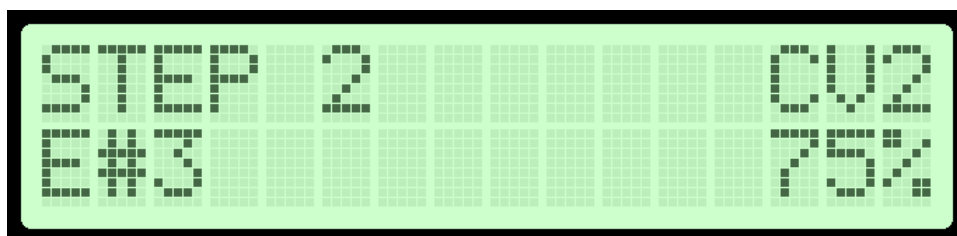


FIGURE 2.14: Écran affichant le CV2

Les images ci-dessus ont été générées à l'aide du site LCD Display Screenshot Generator [AvtSd].



### 2.3.3 Prises Jacks

Des prises Jacks femelles, situées à sur le dessus, serviront à relier le séquenceur au *patchbay* du synthétiseur grâce à des câbles Jacks.

## 2.4 Électronique

Les schémas de circuits (2.4.3) ne sont pas réellement des schémas électriques, mais plutôt des schémas visuels permettant d'avoir une idée du circuit physique du séquenceur.

### 2.4.1 Composants

- Le séquenceur est un Behringer CRAVE [BehSd]. Il s'agit un séquenceur semi-modulaire, mis à disposition par l'Université.
- Les DACs choisis pour ces schémas sont : un DAC haute précision de 12bit pour le *pitch*, et un DAC *4-channels* de 8bit pour les autres paramètres. Il est possible d'utiliser deux DAC *2-channels* de 8bit à la place du quad-DAC. Ces DACs communiquent en SPI [Tex10] avec la Raspberry Pi.
- Un *Port Expander* de huit *I/O* (PCF8574 [Tex01b]) I<sup>2</sup>C [Cyp15] est utilisé pour les huit LEDs afin de ne pas utiliser huit *PINs* de la Raspberry Pi, mais seulement deux. Un même *Port Expander* pourrait être utilisé pour les encodeurs rotatifs. Ces LEDs sont chacune accompagnée d'une résistance de 330Ω.
- Le LCD [AZDSd] est un simple LCD 16x2 avec contrôleur HD44780 [Hit99], sur lequel est attaché un *Backpack* I<sup>2</sup>C pour l'interfacer à la Raspberry Pi avec seulement deux fils (partagés avec le *Port Expander*).
- Huit (cinq) encodeurs rotatifs sont présents pour le contrôle des différents paramètres, et sont reliés directement aux *GPIOs* de la Raspberry Pi.
- Cinq (sept) prises *JACKs* femelles 3.5mm [Swi93] sont branchées aux DACs pour les relier au synthétiseur à l'aide de câbles.
- Un (trois) bouton poussoir *SPST* temporaire est relié directement aux *GPIOs* de la Raspberry Pi.
- (Deux switches, pour contrôler le choix entre tempo interne et clock externe, ainsi que play/-pause.)
- Enfin, le cerveau central est une Raspberry Pi 3B+ [Ras18].

## 2.4.2 Remarques

La liste des composants et les schémas ne sont pas à jour avec la nouvelle idée de design final.

### Remarques composants :

Sur l'option 2 (Fig. 2.4.3), deux *Logic Level Shifters* [SpaSd] (3.3V-5V) sont rajoutés. Ces derniers permettent de s'assurer de l'utilisation des protocoles SPI et I<sup>2</sup>C des composants 5V avec la Raspberry Pi qui, elle, communique en 3.3V sur ses *GPIOs*. Leur utilisation n'est pas obligatoire, mais évite d'éventuels risques de dommages électriques si, lors de la construction du prototype, du 5V était envoyé dans les *GPIOs* 3.3V.

Le choix des composants n'est pas définitif et certains risquent de changer dans peu de temps.

### Remarques schémas :

Pour chaque schéma :

- Il s'agit d'un schéma purement théorique.
- Les connections électriques ne sont pas toutes correctes. Certaines connections servent uniquement à représenter le type de connexion entre deux composants.
- Certains composants ne présentent pas la disposition et le nombre de PINs réel.

Ces schémas ont été réalisés avec Fritzing [Fri13].

### 2.4.3 Schéma

#### Option 1 - sans *Logic Level Shifter*

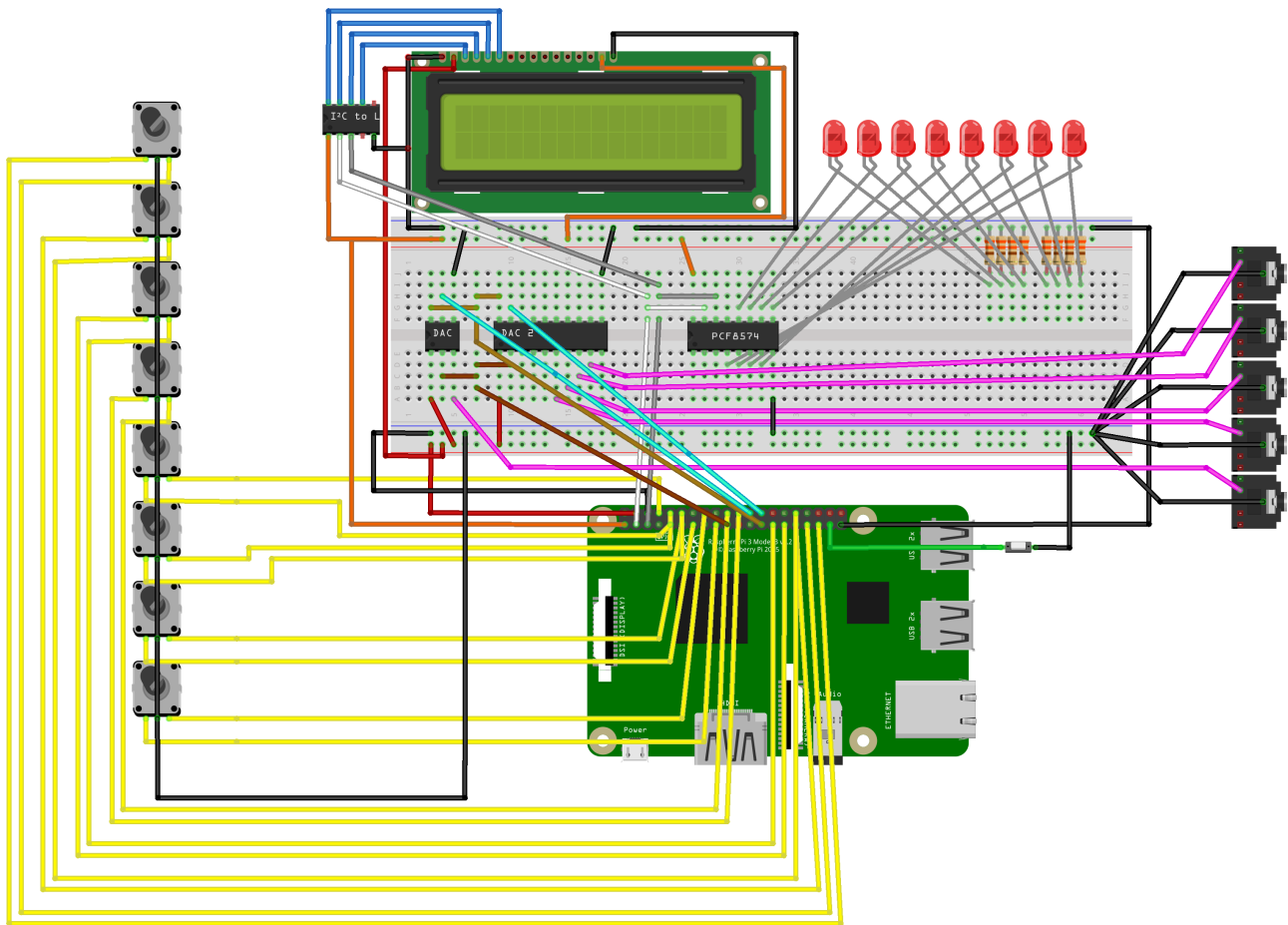
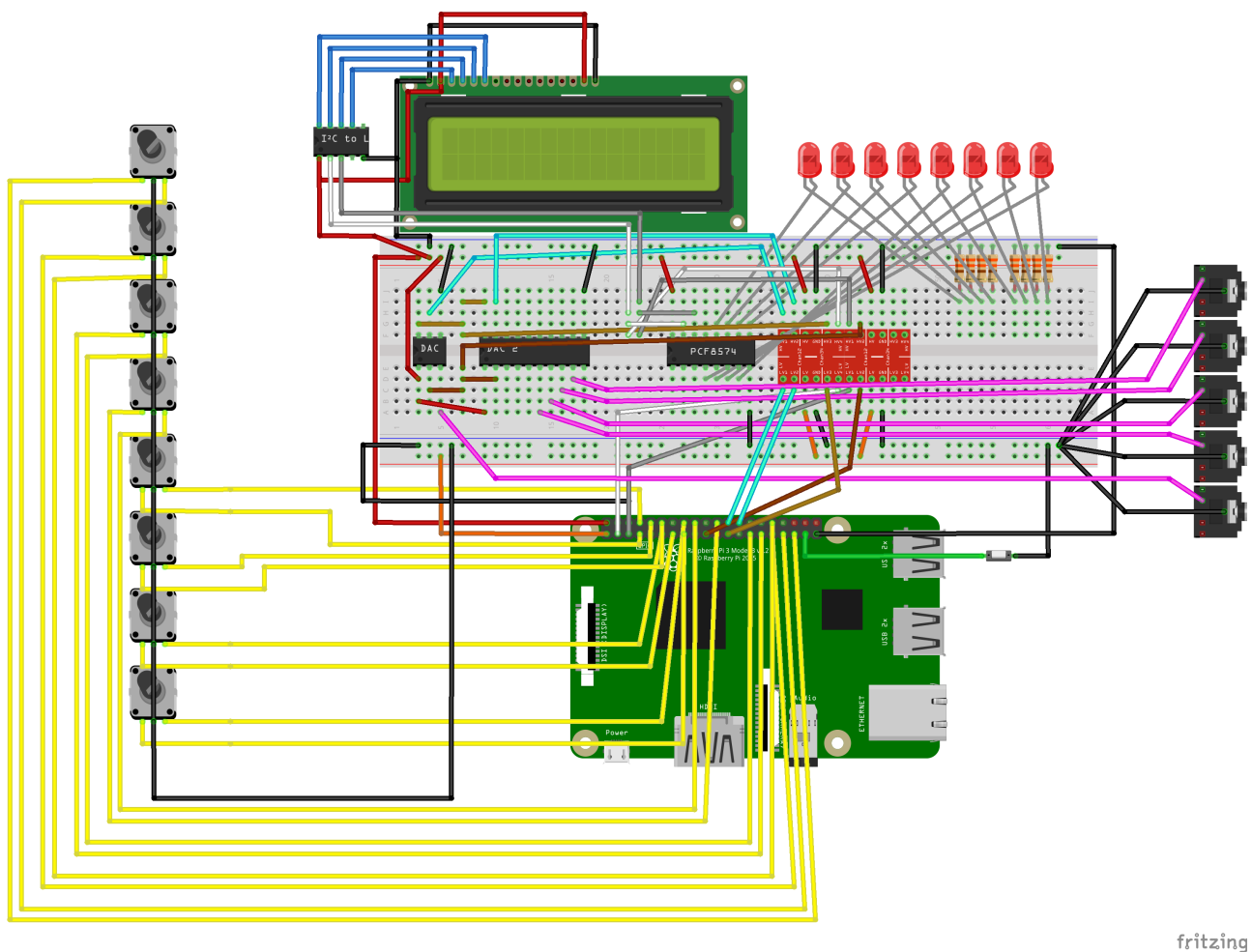


FIGURE 2.15: Schéma électronique option 1

#### Légende :

- I<sup>2</sup>C to LCD IC : *Backpack I<sup>2</sup>C* pour LCD
- LCD : 16x2 HD44780 LCD
- PCF8574 : 8 I/O *Port Expander I<sup>2</sup>C*
- DAC 1 : DAC 12bit
- DAC 2 : quad-DAC 8bit
- Résistance : 330Ω
- Raspberry Pi : modèle 3B+
- LED : LED rouge 5mm
- Fil rouge : 5V
- Fil orange : 3.3V
- Fil noir : *GROUND* (Masse)
- Fil jaune : *GPIO* encodeur rotatif
- Fil cyan : SPI CS (*Chip Select*)
- Fil vert : *GPIO* bouton poussoir

- Fil brun : SPI MOSI (*Master Output - Slave Input*)
- Fil ocre : SPI CLCK (*Clock*)
- Fil blanc : I<sup>2</sup>C SDA (*Serial Data*)
- Fil gris : I<sup>2</sup>C SCL (*Serial Clock*)
- Fil rose : Sortie *JACK*
- Fil bleu : communication *Backpack-LCD*

Option 2 - avec *Logic Level Shifter*

fritzing

FIGURE 2.16: Schéma électronique option 2

**Légende :**

- PCB rouge : *Logic Level Shifter* 5V-3.3V
- I<sup>2</sup>C to LCD IC : *Backpack* I<sup>2</sup>C pour LCD
- LCD : 16x2 HD44780 LCD
- PCF8574 : 8 I/O *Port Expander* I<sup>2</sup>C
- DAC 1 : DAC 12bit
- DAC 2 : quad-DAC 8bit
- Résistance : 330Ω
- Raspberry Pi : modèle 3B+
- LED : LED rouge 5mm
- Fil rouge : 5V
- Fil orange : 3.3V
- Fil noir : *GROUND* (Masse)
- Fil jaune : *GPIO* encodeur rotatif
- Fil cyan : SPI CS (*Chip Select*)
- Fil vert : *GPIO* bouton poussoir
- Fil brun : SPI MOSI (*Master Output - Slave Input*)

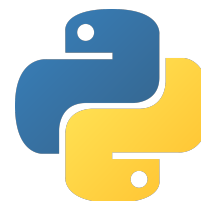
- Fil ocre : SPI CLCK (*Clock*)
- Fil blanc : I<sup>2</sup>C SDA (*Serial Data*)
- Fil gris : I<sup>2</sup>C SCL (*Serial Clock*)
- Fil rose : Sortie *JACK*
- Fil bleu : communication *Backpack*-LCD

## 3.1 Code

Pour retrouver le code, voir **Annexe A** ou **Github**<sup>1</sup> de Lucas Placentino [PLA21].

### 3.1.1 Choix du langage

Le langage Python a été choisi pour le code car il est le plus utilisé dans le monde de la Raspberry Pi (sous Raspberry Pi OS Linux). Ceci permet d'accéder à des milliers de bibliothèques, tutoriels et forums d'aide au *troubleshooting* spécifique à la Raspberry Pi en Python.



De plus, le cours de programmation en Python suivi en première année de Bachelier donne toutes les bases nécessaires pour construire le code relatif au séquenceur.

FIGURE 3.1:  
Logo Python

### 3.1.2 Structure

La programmation orientée objet (POO ou *OOP*) a été choisie pour sa modularité, son organisation et sa clarté (bien que la POO ne nous sera enseignée qu'après le fin de ce projet).

Chaque classe se trouve dans son propre fichier. Le séquenceur se lance en exécutant le fichier `SequenceurMain.py`. Ce dernier contient les *import* des différentes classes et les relie.

---

1. <https://github.com/lucasplacentino/TRANH201INF03-Sequencer/tree/main/code>

### 3.1.3 Librairies

1. gpiozero [NOT15c] :

Interaction avec les GPIOs de la Raspberry Pi. Fonctions pour boutons, encodeurs rotatifs déjà écrites.

2. RPLCD [BAR13] :

Interaction avec un LCD via I<sup>2</sup>C avec la Raspberry Pi.

3. MCP4922 [BAL17] :

Interaction avec un DAC MCP4922 via SPI de la Raspberry Pi.

4. spidev [THO12] :

Interaction avec le bus et controleur SPI de la Raspberry Pi.

### 3.1.4 Fonctionnement et exécution

Le séquenceur écoute les encodeurs rotatifs pour un changement de pitch ou CV, ensuite il va *print* la nouvelle valeur et l'envoyer au DAC correspondant...

### 3.1.5 Gestion d'erreur

Pour l'instant la gestion d'erreur ne se fait que par simple *print* dans la console d'éventuelles erreurs.

## 3.2 Précision du signal

Une précision  $\left(p = \frac{f_1}{f_2} - 1\right)$  à 4 cents ( $c = 4$ ) est souhaitée, où un cent vaut un centième de demi-ton. Le cent calcule un rapport entre 2 fréquences  $\left(\frac{f_1}{f_2} = 2^{c/1200}\right)$  :

$$c = 4 \Rightarrow \frac{f_1}{f_2} = 2^{1/300} \Rightarrow p = 2^{1/300} - 1$$

Soit  $n$  le nombre de bits donc  $2^n$  nombres de possibilités et  $V_{max} = 5V$ . On calcule le nombre de bits nécessaires pour avoir la précision donnée avec la condition suivante :

$$\frac{V_{max}}{2^n} < p \Leftrightarrow n > \log_2 \frac{V_{max}}{p} = 11.07785$$

Il nous faut donc 12 bits pour coder notre signal.



## 4.1 BOM (Bill Of Materials)

<u>Quantity bought</u>	<u>Name</u>	<u>Price</u>	<u>Quantity used</u>	<u>Net Price</u>
1	Raspberry Pi 3B+	41.00€	1	41.00€
1	SD Card	6.43€	1	6.43€
4	DAC(s)	9.60€	3	7.20€
8	Rotary Encoder	9.65€	5	6.03€
1	Power supply Raspberry Pi	14.23€	1	14.23€
10	Button	1.42€	7	0.99€
1	LCD display	5.58€	1	5.58€
3	Breadboard + jumper wires	8.63€	2	5.75€
8	LEDs	0.60€	8	0.60€
8	Resitors	0.15€	8	0.15€
5	Logic Level Shif- ter	8.06€	1	1.61€
12	JACKs female	5.50€	7	3.21€
2	PCF8574	4.28€	1	2.14€
1	Laser cut wood	0.00€	1	0.00€
2	Switch	2.15€	2	2.15€
	<b>TOTAL :</b>	<b>115.13€</b>	<b>NET TOTAL :</b>	<b>97.08€</b>

TABLE 4.1: Bill Of Materials

Dû au *chip shortage* actuel et au très faible stock, le prix de la Raspberry Pi 3B+ est enflé. Un prix plus correct pour cette dernière serait de 35€, on pourrait donc soustraire 6€ au total afin d'avoir un prix rectifié.

## 4.2 Fabrication

### 4.2.1 Photos de la fabrication

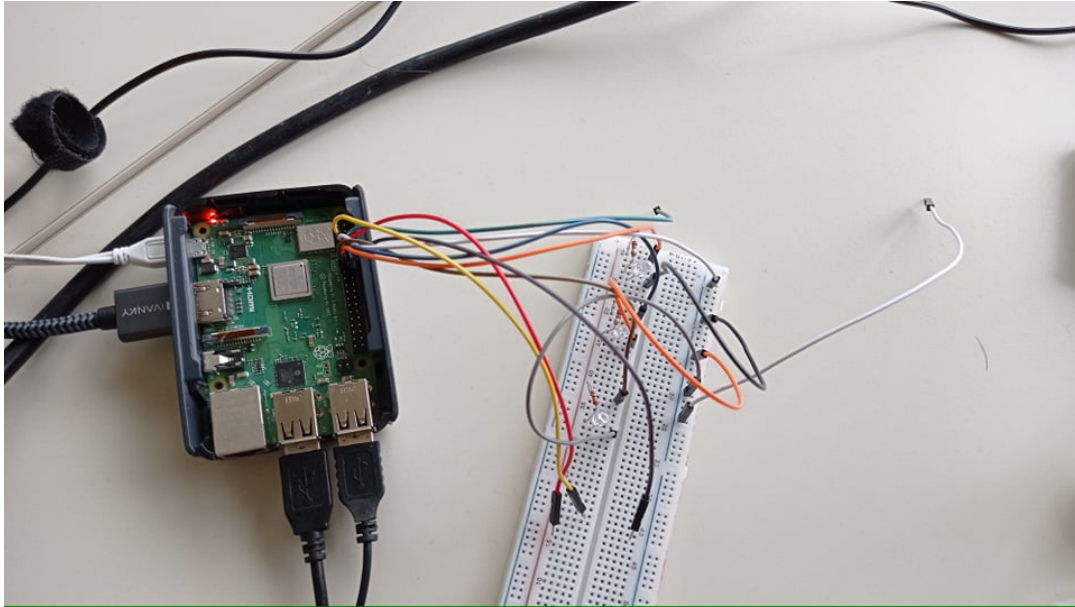


FIGURE 4.1: Premier assemblage d'une séquence de LEDs

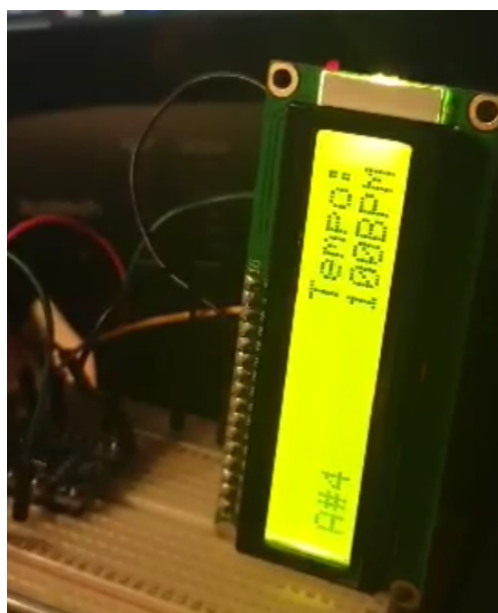


FIGURE 4.2: Premier assemblage du LCD avec la Raspberry Pi

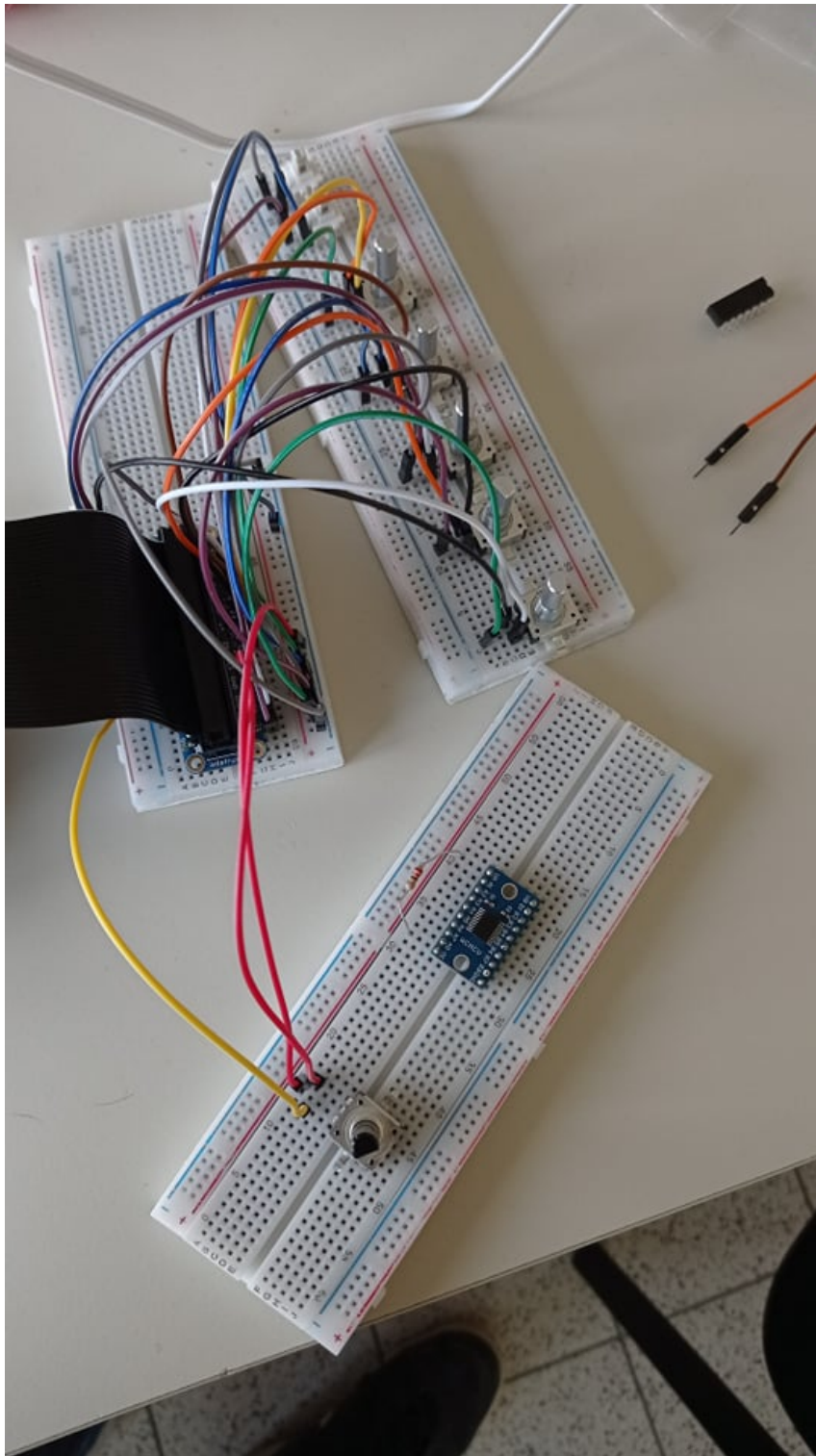


FIGURE 4.3: Premier assemblage avec les encodeurs rotatifs

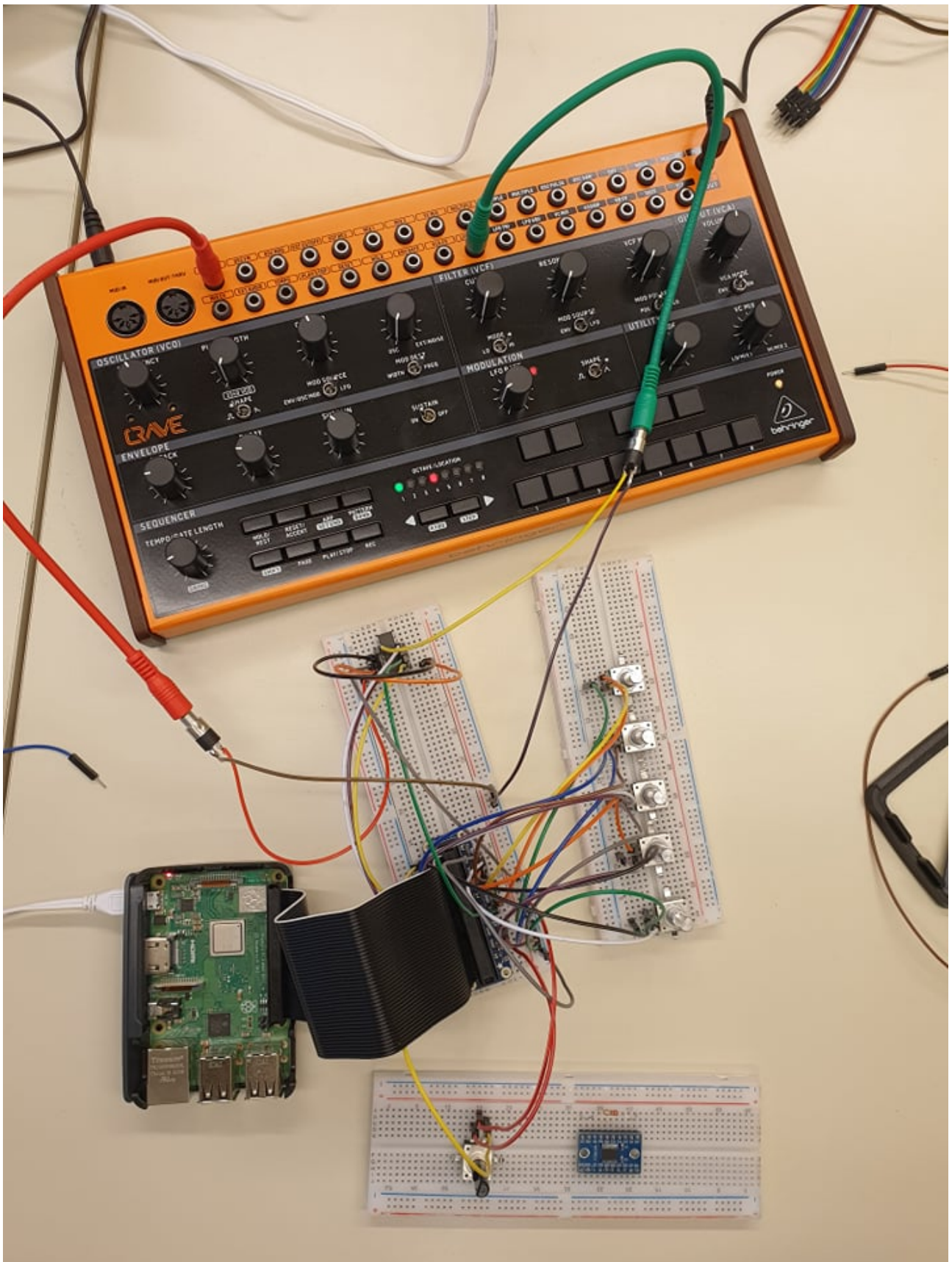


FIGURE 4.4: Premier test complet sur synthétiseur, encodeurs rotatifs, DACs et prises JACK



### 4.2.2 Référence GPIOs

Un tableau qui reprend tous les GPIOs est présenté. Il montre les composants connectés et le type de Pin.

n° GPIO	Utilisé ?	Type	Connecté à	Pin physique
0		GPIO		
1		GPIO		
2		SDA I <sup>2</sup> C		3
3		SCL I <sup>2</sup> C		5
4		GPIO	A RE 1	7
5		GPIO	Button 1 ON	
6		GPIO	Button 2 augm pitch	
7		SPI CE1	DAC 1	26
8		SPI CE0		24
9		GPIO		
10		SPI MOSI		19
11		SPI SCLK		23
12		GPIO	Button 3 dim pitch	
13		GPIO		
14		GPIO	B RE 1	8
15		GPIO	A RE 2	10
16		GPIO		
17		GPIO	A RE 3	11
18		GPIO	B RE 2	12
19		GPIO		
20		GPIO		
21		GPIO		
22		GPIO	A RE 4	15
23		GPIO	B RE 4	16
24		GPIO	A RE 5	18
25		GPIO	B RE 5	22
26		GPIO		
27		GPIO	B RE 3	13
GND		GROUND		6, 9, 14, 20, 25, 30, 34, 39
5V		POWER		2, 4
3.3V		POWER		1, 17

FIGURE 4.5: Liste des GPIOs

## 4.3 Validation

### 4.3.1 Tests

- Premier test : Il consiste à tester une séquence de LEDs et un bouton.  
Il permet de se familiariser avec la librairie `gpiozero` en général.
- Deuxième test : Il consiste à tester le LCD en I<sup>2</sup>C avec un Logic Level Shifter.  
Il permet de se familiariser avec la librairie `RPLCD.i2c` Python.
- Troisième test : Il consiste à tester un encodeur rotatif.  
Il permet de se familiariser avec la librairie `gpiozero`, précisément pour les encodeurs rotatifs.
- Quatrième test : Il consiste à tester un DAC en 3.3V.  
Il permet de se familiariser avec la librairie `MCP4922` Python.
- Cinquième test : Test reprenant les encodeurs rotatifs, les DACs, les boutons et les prises JACK, connectés au synthétiseur.  
Il permet de tester le code rassemblant ces composants, de manière concrète, et avec sortie de son.

La mise en place d'un protocole de testing est prévue pour le Q2.

### 4.3.2 Résultats

Pour les quatre premiers tests individuels des différents composants, procéder par essai-erreur en se référant à la librairie associée permet de comprendre au maximum cette dernière.

Très vite, les test étaient concluants et une bonne idée du fonctionnement de chaque librairie Python a été acquise.

Le premier test a été réalisé avec succès ; cependant, les boutons ne sont pas utilisables dans la cinquième phase reprenant le code avec les encodeurs rotatifs, les DACs, et les prises JACK car le code associé aux boutons provoque un crash de la Raspberry Pi.

Le cinquième test (ainsi que le troisième) a vu le code porter ses fruits. Le séquenceur était relié au synthétiseur et la sortie audio correspondait aux changements de paramètres faits en temps réel via les encodeurs rotatifs.

### 4.3.3 Limitations

Les DACs ne fonctionnent pas encore en 5V. Leur limite de tension est donc pour l'instant à 3.3V. Le séquenceur est tout à fait utilisable avec seulement 3.3V, mais les paramètres ont donc une fourchette de valeurs plus petites.

## 5.1 SWOT

### 5.1.1 SWOT début de projet

- Strengths (Forces) : le codage du séquenceur est plus facile en utilisant la programmation orientée objet avec laquelle Lucas est déjà familier. Il en est de même avec les circuits électroniques pour Lucas et Robin.
- Weaknesses (Faiblesses) : notre équipe est constituée de seulement 4 membres alors que le nombre standard est de 7. Nous n'avons aucune expérience en appareils électronique, que ce soit leur planification ou construction.
- Opportunities (Opportunités) : cette année se déroule en présentiel ce qui facilite l'accessibilité du matériel à tous, que ce soit le Raspberry Pi, ses composants, ou le synthétiseur, notamment lors des réunions de projet. Lucas a une Raspberry Pi personnelle et des composants électroniques à sa disposition.
- Threats (Menaces) : des imprévus peuvent survenir, comme réaliser trop tard que les composants sont incompatibles. Un des membres de l'équipe peut attraper le covid et tomber malade.

## 5.2 Risks and Issues (risques et problèmes)

### 5.2.1 Risks (risques)

- 30/09/2021 **Budget faible** : Le budget limité à 100€ pourrait ne pas être suffisant pour financer l'intégralité du projet.



*Impact : moyen ; Probabilité : haute.*

Fermé le 18/10/2021

- 04/10/2021 **Recherche longue** : Les recherches sur le séquenceur prennent trop de temps et risquent de retarder le planning.

*Impact : bas ; Probabilité : moyenne.*

- 07/10/2021 **Commande retardée** : La commande des composantes pourrait prendre beaucoup de temps ou arriver en retard.

*Impact : moyen ; Probabilité : moyen.*

Fermé le 02/12/2021

- 14/10/2021 **Maladie** : Un des membres pourrait tomber malade du COVID-19.

*Impact : moyen ; Probabilité : moyen.*

- 28/10/2021 **Choix DAC** : Le choix du DAC pourrait prendre plus de temps que prévu.

*Impact : moyen ; Probabilité : moyen.*

Fermé le 26/11/2021

### 5.2.2 Issues (problèmes)

- 23/09/2021 **Équipe de quatre** : Être seulement quatre dans l'équipe limite notre capacité de travail.

*Impact : moyen.*

Fermé le 30/09/2021

- 04/10/2021 **Pénurie de puce** : La fabrication de circuits intégrés est extrêmement limitée. L'achat de tout élément électronique peut être compromise par un prix trop élevé ou une pénurie.

*Impact : Haut.*

Fermé le 7/10/2021

L'idée de conception finale du séquenceur a été formulée et étudiée. Elle sera la base du travail futur, autant pour la construction du prototype, que pour les tests associés. Les réalisations de prototypage du séquenceur sont encore en cours. A cet égard, les objectifs spécifiés en introduction ne sont pas encore atteints. Les schémas finaux de câblage et ainsi que des diagrammes UML sont encore à fournir. D'autre part, les tests ont connu un succès en ce qui concerne les encodeurs rotatifs et les paramètres y étant associés, comme la hauteur de la note et la sortie CV1 reliée à un paramètre du synthétiseur choisi (VCF ou LFO par exemple). Un autre composant a aussi été testé : les boutons. Bien qu'ils fonctionnaient dans la première phase de test avec une simple LED, ils ne fonctionnaient plus lorsqu'ils étaient implémentés dans le code complet (avec encodeurs rotatifs, DAC, et autres composants) provoquant ainsi un crash dans le code forçant donc à redémarrer la Raspberry Pi. En ce qui concerne le DAC, ce dernier est limité à une tension de 3.3V pour une raison encore inconnue.

## **Annexes**



## Code

```
1 from gpiozero import *
2 import time
3 import spidev
4 from signal import pause
5 from MCP4922 import MCP4922
6
7
8 #button1 = Button(5) #29
9 #button2 = Button(38) #31
10 #button3 = Button(40) #32
11
12 rotor1 = RotaryEncoder(4, 14) # 7 et 8 (gate)
13 rotor2 = RotaryEncoder(15, 18) # 10 et 12 (CV1)
14 rotor3 = RotaryEncoder(17, 27) # 11 et 13 (tempo)
15 rotor4 = RotaryEncoder(22, 23) # 15 et 16 (CV2)
16 rotor5 = RotaryEncoder(24, 25) # 18 et 22 (CV3)
17 rotor6 = RotaryEncoder(6, 12) #31 et 32 (pitch)
18
19
20 """
21 led1 = LED(27) #13
22 led2 = LED(23) #15
23 led3 = LED(22) #16
24
```

```

25  l_led = [led1, led2, led3]
26  """
27
28  dac1 = MCP4922(0, 0, 8) #24 (4921 mais ca peut aussi fonctionner en 4922)
29  dac2 = MCP4922(0, 0, 7) #26
30  #dac3 = MCP4922(0, 0, 8) #24
31
32
33  note = ["C", "C#", "D", "D#", "E", "F", "F#", "G", "G#", "A", "A#", "B"]
34  l_step = []
35  for i in range(8): #creation de la liste des notes
36      l_step.append([0, 1]) #cette liste represente l'octave et la note
37
38
39  MAX_STEP = 7
40  MIN_STEP = 0
41  MAX_TEMPO = 359
42  MIN_TEMPO = 2
43  MAX_OCTAVE = 4
44  MIN_OCTAVE = 0
45  MAX_PITCH = 12
46  MIN_PITCH = 1
47  MAX_CV = 25
48
49
50  class Tempo:
51      valeur = 100
52
53  tempo = Tempo()
54
55
56  class Step:
57      n = 0 #indice de la liste a modifier

```

```
58
59 step = Step()
60
61
62 class CV:
63     CV1 = 0
64     CV2 = 0
65     CV3 = 0
66     Gate = 1
67
68 CV = CV()
69
70
71 def led(n):
72     n.on()
73     time.sleep(60/tempo.valeur)
74     n.off()
75
76
77 def led_note():
78     octave = str(l_step[step.n][0])
79     pitch = note[l_step[step.n][1]-1]
80     return "Note : " + pitch + octave
81
82
83 def augmenter_step():
84     step.n += 1
85     step.n = step.n%MAX_STEP
86     print("Step : " + str(step.n))
87
88
89 def diminuer_step():
90     if step.n == MIN_STEP: # step.pitch
```

```

91         step.n = MAX_STEP
92     else:
93         step.n -= 1
94     print("Step : " + str(step.n))
95
96
97 def augmenter_pitch():
98     if l_step[step.n][1] == MAX_PITCH:
99         l_step[step.n][0] += 1
100         l_step[step.n][0] = l_step[step.n][0]%(MAX_OCTAVE + 1)
101         l_step[step.n][1] += 1
102     l_step[step.n][1] += 1
103     l_step[step.n][1] = l_step[step.n][1]%(MAX_PITCH + 1)
104     print(led_note())
105
106     ↪ dac1.setVoltage(0,int(4096*(((l_step[step.n][0]*12)+l_step[step.n][1])/60)))
107
108 def diminuer_pitch():
109     if l_step[step.n][1] == MIN_PITCH:
110         l_step[step.n][1] = MAX_PITCH
111         if l_step[step.n][0] == MIN_OCTAVE:
112             l_step[step.n][0] = MAX_OCTAVE
113         else:
114             l_step[step.n][0] -= 1
115     else:
116         l_step[step.n][1] -= 1
117     print(led_note())
118
119     ↪ dac1.setVoltage(0,int(4096*(((l_step[step.n][0]*12)+l_step[step.n][1])/60)))
120
121 def augmenter_octave():

```

```
122     l_step[step.n][0] += 1
123     l_step[step.n][0] = l_step[step.n][0]%(MAX_OCTAVE + 1)
124     #print(led_note())
125
126
127 def diminuer_octave():
128     if l_step[step.n][0] == MIN_OCTAVE:
129         l_step[step.n][0] = MAX_OCTAVE
130     else:
131         l_step[step.n][0] -= 1
132     #print(led_note())
133
134
135 def augmenter_tempo():
136     if tempo.valeur < MAX_TEMPO:
137         tempo.valeur += 1
138     print("Tempo:",tempo.valeur)
139
140
141 def diminuer_tempo():
142     if tempo.valeur > MIN_TEMPO:
143         tempo.valeur -= 1
144     print("Tempo",tempo.valeur)
145
146
147 def augmenter_CV1():
148     CV.CV1 += 1
149     CV.CV1 = CV.CV1%MAX_CV
150     dac1.setVoltage(1, int(float(4096)*CV.CV1/24))
151     print("CV1:",CV.CV1)
152
153
154 def augmenter_CV2():
```



```
155     CV.CV2 += 1
156     CV.CV2 = CV.CV2%MAX_CV
157     dac2.setVoltage(0, int(float(4096)*CV.CV2/24))
158     print("CV2:",CV.CV2)
159
160
161 def augmenter_CV3():
162     CV.CV3 += 1
163     CV.CV3 = CV.CV3%MAX_CV
164     dac2.setVoltage(1, int(float(4096)*CV.CV3/24))
165     print("CV3:",CV.CV3)
166
167
168 def diminuer_CV1():
169     if CV.CV1 == 0:
170         CV.CV1 = 24
171     else:
172         CV.CV1 -= 1
173     dac1.setVoltage(1, int(float(4096)*CV.CV1/24))
174     print("CV1:",CV.CV1)
175
176
177 def diminuer_CV2():
178     if CV.CV2 == 0:
179         CV.CV2 = 24
180     else:
181         CV.CV2 -= 1
182     dac2.setVoltage(0, int(float(4096)*CV.CV2/24))
183     print("CV2:",CV.CV2)
184
185
186 def diminuer_CV3():
187     if CV.CV3 == 0:
```

```

188         CV.CV3 = 24
189     else:
190         CV.CV3 -= 1
191     dac2.setVoltage(1, int(float(4096)*CV.CV3/24))
192     print("CV3:",CV.CV3)
193
194
195 def augmenter_Gate():
196     if CV.Gate >= 0.95:
197         CV.Gate = 0
198     else:
199         CV.Gate += 0.1
200         #dac3.setVoltage(1, int(4096*CV.Gate))
201     print("Gate:",round(CV.Gate,1))
202
203
204 def diminuer_Gate():
205     if CV.Gate <= 0.05:
206         CV.Gate = 1
207     else:
208         CV.Gate -= 0.1
209         #dac3.setVoltage(1, int(4096*CV.Gate))
210     print("Gate:",round(CV.Gate,1))
211
212 '''
213 def on():
214     p = 'on'
215     while p == 'on':
216         for i in range(8):
217             dac3.setVoltage(1, 12*l_step[i][0] + l_step[i][1])
218             # led(l_led[i])
219             time.sleep(60/tempo.valeur)
220             if not button1.is_pressed:

```

```
221         p = 'off'
222     #if keyboard interrupt:
223         #(led off)
224         #break
225     '''
226     # C0 - (B#4) - C5
227
228     #button1.when_pressed = on
229     #button2.when_pressed = augments_octave
230     #button3.when_pressed = diminuer_octave
231
232
233     rotor1.when_rotated_clockwise = augments_pitch
234     rotor1.when_rotated_counter_clockwise = diminuer_pitch
235     rotor2.when_rotated_clockwise = augments_CV1
236     rotor2.when_rotated_counter_clockwise = diminuer_CV1
237     rotor3.when_rotated_clockwise = augments_Gate
238     rotor3.when_rotated_counter_clockwise = diminuer_Gate
239     rotor4.when_rotated_clockwise = augments_CV2
240     rotor4.when_rotated_counter_clockwise = diminuer_CV2
241     rotor5.when_rotated_clockwise = augments_CV3
242     rotor5.when_rotated_counter_clockwise = diminuer_CV3
243     rotor6.when_rotated_clockwise = augments_tempo
244     rotor6.when_rotated_counter_clockwise = diminuer_tempo
245
246
247     pause()
```

---

## Bibliographie

- [Ana99] ANALOG DEVICES (1999). *LTC1665/LTC1660 - Micropower Octal 8-Bit and 10-Bit DACs*. URL : <https://www.analog.com/media/en/technical-documentation/data-sheets/166560fa.pdf> (visité le 05/10/2021).  
Datasheet de DAC.  
Provient du fabricant réputé Analog Devices.
- [AvtSd] AVTANSKI (S.d.). *LCD/LED Screenshot Generator*. URL : <http://avtanski.net/projects/1cd/> (visité le 22/10/2021).  
Générateur d'image de simulation de Liquid Crystal Display.
- [AZDSd] AZDELIVERY (S.d.). *HD44780 16x02 Blaues Display mit Serielle Schnittstelle Datenblatt*. URL : [https://cdn.shopify.com/s/files/1/1509/1638/files/HD44780\\_1602\\_Blaues\\_LCD\\_Display\\_mit\\_Serielle\\_Schnittstelle\\_I2C\\_Bundle\\_Datenblatt\\_AZ-Delivery\\_Vertriebs\\_GmbH.pdf?v=1591601507](https://cdn.shopify.com/s/files/1/1509/1638/files/HD44780_1602_Blaues_LCD_Display_mit_Serielle_Schnittstelle_I2C_Bundle_Datenblatt_AZ-Delivery_Vertriebs_GmbH.pdf?v=1591601507) (visité le 05/10/2021).  
Datasheet LCD+Backpack I²C.
- [BAL17] BALLUFF, Paul (2017). *MCP4922 : Python-RPi-MCP4922*. URL : <https://github.com/mrwunderbar666/Python-RPi-MCP4922> (visité le 07/11/2021).  
Librairie Python pour DAC MCP4922 avec la Raspberry Pi.
- [BAR13] BARGEN, Danilo (2013). *RPLCD : A Raspberry Pi LCD library for the widely used Hitachi HD44780 controller*. Version 1.3.0. DOCS : <https://rplcd.readthedocs.io/en/stable/>. URL : <https://github.com/dbrgn/RPLCD> (visité le 05/10/2021).  
Librairie Python pour LCD avec une Raspberry Pi.  
Librairie très utilisée. Danilo Bargaen est un ingénieur Software, bachelier en science informatique.
- [BBC70] BBC (1970). *Wendy Carlos demonstrates her Moog Synthesizer in 1970*. URL : <https://www.youtube.com/watch?v=4SBDH5uhs4Q> (visité le 03/10/2021).

Présente de manière concise et intuitive les concepts du projet.

Wendy Carlos est une compositrice de musique électronique, experte dans le domaine.

- [BehSd] BEHRINGER (S.d.). *CRAVE Quick Start Guide*. URL : [https://mediadl.musictribe.com/download/documents/behringer/CRAVE/CRAVE\\_QSG\\_WW.pdf](https://mediadl.musictribe.com/download/documents/behringer/CRAVE/CRAVE_QSG_WW.pdf) (visité le 29/09/2021).  
Manuel du Behringer CRAVE.

- [BER20] BERG, Craig (2020). *Raspberry Pi 4 For Beginners and Intermediates : A Comprehensive Guide for Beginner and Intermediates to Master the New Raspberry Pi 4 and Set up Innovative Projects*. ISBN : 9798650755586. URL : [https://cibleplus.ulb.ac.be/permalink/32ULDB\\_U\\_INST/1hd4301/alma991009823322304066](https://cibleplus.ulb.ac.be/permalink/32ULDB_U_INST/1hd4301/alma991009823322304066).

Explique l'utilisation d'une Raspberry Pi.

Livre présent dans à BST à l'ULB, Prof. Craig Berg est professeur à la University of Wisconsin-Milwaukee.

- [Bun20] BUNKER (2020). *Any interest in a DIY sequencer project?* URL : <https://www.elektronauts.com/t/any-interest-in-a-diy-sequencer-project/124417> (visité le 07/10/2021).  
Contient des informations intéressantes sur les séquenceurs.

Forum d'hobbyistes.

- [cas16] CASPERELECTRONICS SYNTHESIZERS, Peter Edwards (2016). *Casper Electronics DIY synth building. Part 2 : Sequencers*. URL : <https://www.youtube.com/watch?v=t7JY0IYqMEY> (visité le 12/10/2021).

Vidéo très complète reprenant le fonctionnement d'un synthétiseur et séquenceur DIY.

Chaîne Youtube d'un hobbyiste sur les synthétiseurs analogiques.

- [COX18] COX Tim, Dr. Steven L. FERNANDES (2018). *Raspberry Pi 3 cookbook for Python programmers : unleash the potential of Raspberry Pi 3 with over 100 recipes*. ISBN : 978-1-78862-987-4. URL : [https://cibleplus.ulb.ac.be/permalink/32ULDB\\_U\\_INST/1hd4301/alma991009823319704066](https://cibleplus.ulb.ac.be/permalink/32ULDB_U_INST/1hd4301/alma991009823319704066).

Contient des exemples de programmes en Python pour la Raspberry Pi.

Livre présent en bibliothèque à la BST de l'ULB, Tim Cox est ingénieur software et co-fondateur de MagPi (magazine officiel Raspberry Pi).

- [CRISd] CRIMIER (S.d.). *Raspberry Pi : Using 1\$ I2C LCD Backpacks for 1602 Screens & Buttons!* URL : <https://www.instructables.com/Raspberry-Pi-Using-1-I2C-LCD-Backpacks-for-1602-Sc/> (visité le 13/10/2021).

Utilisation du même LCD + Backpack I<sup>2</sup>C.

Instructables est un site de partage de projets d'hobbyistes, CRImier est aussi un hobbyiste.

- [CRO12] CROSTON, Ben (2012). *RPi.GPIO 0.7.0*. URL : <https://pypi.org/project/RPi.GPIO/> (visité le 13/10/2021).  
Documentation pour la librairie Python "RPi.GPIO" pour Raspberry Pi.  
Ben Croston est un hobbyiste.
- [Cur80] CURTIS ELECTROMUSIC SPECIALTIES (1980). *CEM3312 datasheet*. URL : <http://www.cedos.com/datasheets/cem3312pdf.pdf> (visité le 03/10/2021).  
Datasheet du composant CEM3312 pour la création d'enveloppe.  
Provient du fabricant Curtis Electromusic Specialties.
- [Cyp15] CYPRESS SEMICONDUCTOR CORPORATION (2015). *I2C Master/Multi-Master/Slave*. URL : <https://www.cypress.com/file/175671/download> (visité le 24/10/2021).  
Datasheet protocole I<sup>2</sup>C.
- [12] *DIY step sequencer - Technology / Synth-DIY* (2012). URL : <https://forum.mutable-instruments.net/t/diy-step-sequencer/1913> (visité le 07/10/2021).  
Informations sur la fabrication d'un séquenceur.  
Forum d'hobbyistes.
- [DON14] DONAT, Wolfram (2014). *Learn Raspberry Pi Programming with Python*. ISBN : 978-1-4302-6425-5. DOI : 10.1007/978-1-4302-6425-5.  
Présente des exemples de projets en Python pour la Raspberry Pi.  
Livre publié sur Springer; Wolfram Donat est un ingénieur informatique et adepte d'enseignement d'électronique.
- [Ele08] ELECTRIC DRUID (2008). *Voltage Controlled ADSR Envelope Generator (VC ADSR 7B) – Electric Druid*. URL : <https://electricdruid.net/voltage-controlled-adsr-envelope-generator-vc-adsr-7b/> (visité le 20/10/2021).  
Présentation générale des puces CEM 3312 et SSM 2056 génératrices d'enveloppes.  
Le site Electric Druid vend des composants électroniques pour les synthétiseurs.
- [Ele19] — (2019). *VCLFO 10 datasheet*. URL : <https://electricdruid.net/datasheets/VCLFO10Datasheet.pdf> (visité le 03/10/2021).  
Datasheet présentant le VCLFO 10 composant électronique pour le LFO.  
Provient du site de vente Electric Druid.
- [ELL12] ELLINGER, John (2012). *MUSC 101 Unit 1 Sound Basics*. URL : <https://people.carleton.edu/~jellinge/m101s12/Pages/01/01SoundBasics.html> (visité le 09/10/2021).

Explique le rapport entre les notes et leurs fréquences.

John Ellinger donne des cours de musique à la Carleton University.

- [ELSSd] ELSEA, Peter (S.d.). *WHAT TO DO WITH YOUR SYNTHESIZER*. URL : <http://artsites.ucsc.edu/ems/music/equipment/synthesizers/Synthesizing/usesynth.html> (visité le 30/09/2021).

Explique le fonctionnement d'un synthétiseur analogique.

Peter Elsea donne cours aux "Electronic Music Studios" à la University of California, Santa Cruz.

- [ENI19] ENI INFORMATIQUE (2019). *Découvrir l'électronique avec le Raspberry Pi 4*. URL : <https://www.youtube.com/watch?v=jGRV-i0JPGY> (visité le 20/10/2021).

Présentation générale du Raspberry Pi 4, de ses composants et de quelques exemples d'utilisation.

Vidéo produite par ENI Informatique qui propose de nombreux livres et formations dans le domaine informatique.

- [Ext18] EXTRALIFE (2018). *Designing a Eurorack Sequencer - Pt. 1 : Digital to Analog*. URL : <https://www.youtube.com/watch?v=LfQ3woQ3Kn8> (visité le 03/10/2021).

Première vidéo d'une série dédiée à la réalisation d'un séquenceur sur la base d'un arduino, elle donne des informations pratiques sur le DAC.

Vidéo réalisée par un hobbyiste.

- [Fri13] FRITZING (2013). *Fritzing - electronics made easy*. URL : <http://fritzing.org/>.  
Site officiel de Fritzing.

- [GAY17] GAY, Warren (2017). *Custom Raspberry Pi Interfaces : Design and Build Hardware Interfaces for the Raspberry Pi*. ISBN : 978-1-4842-2405-2. DOI : 10.1007/978-1-4842-2406-9.

Explique l'utilisation et l'interfaçage d'un LCD via I<sup>2</sup>C, ainsi que d'autres informations utiles sur la Raspberry Pi.

Livre Publié sur Springer.

- [GAY18] — (2018). *Advanced Raspberry Pi Raspbian Linux and GPIO Integration*. ISBN : 978-1-4842-3948-3. DOI : 10.1007/978-1-4842-3948-3.

Présente des concepts avancés sur la Raspberry Pi.

Livre publié sur Springer.

- [Gén15] GÉNIE ÉLECTRIQUE (2015). *NE555, fonctionnement général*. URL : <http://for-ge.blogspot.com/2015/07/ne555.html> (visité le 20/10/2021).

Présentation du composant NE555, composant électronique utile pour la génération

d'un signal.

L'article provient d'un blog de génie électrique.

- [HAR12] HARMON, Steve (2012). *8 Step Sequencer Assembly Instructions* | Synthrotek. URL : <https://www.synthrotek.com/kit-assembly-instructions/discontinued-products/8-step-sequencer-assembly-instructions/> (visité le 07/10/2021).

Guide de fabrication d'un séquenceur à 8 pas.

Article fait par le propriétaire d'un magasin de musique "Synthrotek".

- [HAS04] HASS, Jeffrey (2004). *Synthesis Chapter Four : Envelopes*. URL : [https://cmtext.indiana.edu/synthesis/chapter4\\_envelopes.php](https://cmtext.indiana.edu/synthesis/chapter4_envelopes.php) (visité le 18/10/2021).

a.

Prof. Jeffrey Hass est professeur en musique électronique et composition à la Indiana University Bloomington.

- [HAW19] HAWES, Stephen (2019). *Designing a Eurorack Sequencer!* URL : <https://www.youtube.com/watch?v=N4pwyMeGUwE> (visité le 03/10/2021).

Présente la réalisation d'un séquenceur à 4 pas en détails et permet de se faire une idée de la démarche à suivre.

La vidéo est réalisée par un hobbyiste.

- [HIL77] HILLEN, Peter (1977). « A Microprocessor Based Sequencer for Voltage Controlled Electronic Music Synthesizers ». In : 57th AES Convention. URL : <https://www.aes.org/e-lib/browse.cfm?elib=3126> (visité le 17/10/2021).

Sujet identique à notre projet.

Peter Hillen est expert en électronique audio, il a présenté plusieurs conférences à la Audio Engineering Society et a été auteur d'articles dans le célèbre magazine Synapse (1976-1979) sur la musique électronique.

- [Hit99] HITACHI (1999). *HD44780U (LCD-II) : (Dot Matrix Liquid Crystal Display Controller/-Driver)*. URL : [http://academy.cba.mit.edu/classes/output\\_devices/44780.pdf](http://academy.cba.mit.edu/classes/output_devices/44780.pdf) (visité le 12/10/2021).

Datasheet HD44780.

- [19] *How many bits are needed for CV DAC? - MOD WIGGLER* (2019). URL : <https://modwiggler.com/forum/viewtopic.php?t=217913> (visité le 09/10/2021).

Informations et conseils sur le nombres de bits de précision d'un DAC pour des CV.

Forum d'hobbyistes.

- [HUA20] HUANG, Andrew (2020). *Modular synthesis EXPLAINED*. URL : <https://www.youtube.com/watch?v=cWslSTTkiFU> (visité le 03/10/2021).



Présentation générale de la modulation et de tous les concepts nécessaires à la réalisation du projet.

Le musicien, compositeur et vidéaste Andrew Huang est réputé dans le domaine de la vulgarisation musicale.

- [JAN14] JANSEE Maartje, Jean-Paul BRODIER (2014). *Raspberry Pi : l'alliance de la programmation et de l'électronique : 46 applications utiles pour l'électronicien*. ISBN : 978-2-86661-196-5. URL : [https://cibleplus.ulb.ac.be/permalink/32ULDB\\_U\\_INST/1hd4301/alma991003693769704066](https://cibleplus.ulb.ac.be/permalink/32ULDB_U_INST/1hd4301/alma991003693769704066).

Présente différents projets avec la Raspberry Pi et explique certaines fonctionnalités de cette dernière.

Livre présent à la BST à l'ULB.

- [JEN19] JENKINS, Mark (2019). *Analog Synthesizers : Understanding, Performing, Buying : From the Legacy of Moog to Software Synthesis*. ISBN : 978-0-429-45399-1. DOI : 10.4324/9780429453991.

Contient des explications sur les synthétiseurs modulaires et leur fonctionnement.

Livre publié sur Taylor & Francis, Mark Jenkins est musicien électronique.

- [joa13] JOAN2937 (2013). *pigpio*. URL : <http://abyz.me.uk/rpi/pigpio/> (visité le 13/10/2021).  
Documentation pour la librairie Python "pigpio" pour Raspberry Pi.  
joan2937 est un.e hobbyiste.

- [KLE20] KLEIN, Moritz (2020). *DIY SYNTH VCF Part 1 : Analog Filtering Basics*. URL : <https://www.youtube.com/watch?v=3tMGNI--ofU> (visité le 03/10/2021).

Guide pour la réalisation d'un VCF, avec une explication intuitive des concepts derrière ce filtre.

Moritz Klein est un vidéaste amateur dont la chaîne traite de l'électronique liée à la musique et la modulation de signaux.

- [LOO17] LOOK MUM NO COMPUTER, Sam Battle (2017). *ARDUINO 8 STEP KEYBOARD SEQUENCER FOR SYNTHESIZERS*. URL : <https://www.youtube.com/watch?v=9oG1CfwCoCw> (visité le 07/10/2021).

Fabrication d'un séquenceur pour un synthétiseur analogique.

Vidéo réalisée par Sam Battle, chanteur, producteur et hobbyiste en électronique musicale.

- [Luc18] LUCIDCHART (2018). *UML Use Case Diagram Tutorial*. URL : <https://www.youtube.com/watch?v=zid-MVo7M-E> (visité le 04/11/2021).

Vidéo de présentation des diagrammes de cas d'utilisation.

- [Luc] LUCIDCHART (2021). *UML Use Case Diagram Tutorial*. en. URL : <https://www.lucidchart.com/pages/uml-use-case-diagram> (visité le 04/11/2021).  
Lien permettant de se familiariser avec les diagrammes de cas d'utilisation et d'en réaliser.
- [Max16] MAXIM INTEGRATED (2016). *MAX11300 PIXI, 20-Port Programmable Mixed-Signal I/O with 12-Bit ADC, 12-Bit DAC, Analog Switches, and GPIO*. URL : <https://www.maximintegrated.com/en/products/analog/data-converters/analog-to-digital-converters/MAX11300.html> (visité le 10/10/2021).  
Caractéristiques d'un DAC 12-bit intéressant pour la réalisation du projet.  
Provient du fabricant réputé Maxim Integrated.
- [MON17] MONK, Simon (2017). *Hacking electronics : learning electronics with Arduino® and Raspberry Pi*. ISBN : 978-1-260-01220-0. URL : [https://cibleplus.ulb.ac.be/permalink/32ULDB\\_U\\_INST/1hd4301/alma991009823319504066](https://cibleplus.ulb.ac.be/permalink/32ULDB_U_INST/1hd4301/alma991009823319504066).  
Explique des concept d'électronique avec la Raspberry Pi.  
Livre présent à la BST à l'ULB; Simon Monk, docteur en Software Engineering, est auteur de plusieurs livres sur l'électronique.
- [Mus19] MUSICIAN ON A MISSION (2019). *Synth Basics (Everything You Need to Know)*. URL : <https://www.youtube.com/watch?v=XmI33zP51Us> (visité le 03/10/2021).  
Explication illustrée des concepts du projet.  
La vidéo a été visionnée de très nombreuses fois et a de très bonnes critiques.
- [NEW17] NEWMARCH, Jan (2017). « Raspberry Pi ». In : *Linux Sound Programming*, p. 537-545. ISBN : 978-1-4842-2496-0. DOI : 10.1007/978-1-4842-2496-0\_31.  
Ne s'intéresse qu'à la synthèse du son en elle-même.  
Livre publié sur Springer, Jan Newmarch est professeur adjoint à la Canberra University.
- [10] *nombre de bits nécessaires* (2010). URL : <https://forums.futura-sciences.com/electronique/524106-nombre-de-bits-necessaires.html> (visité le 10/10/2021).  
Explique l'approche à mettre en oeuvre pour calculer le nombre de bits nécessaires pour la précision donnée d'un signal.  
Il s'agit d'un forum de science où l'on s'échange des questions théoriques ou pratiques de tous niveaux.
- [NOT15a] NOTTALL Ben, Daves JONES (2015a). *13. API - Input Devices — GPIO Zero 1.6.2 Documentation*. URL : [https://gpiozero.readthedocs.io/en/stable/api\\_input.html](https://gpiozero.readthedocs.io/en/stable/api_input.html) (visité le 13/10/2021).

Documentation pour la librairie "gpiozero", partie input.

Librairie utilisée officiellement par Raspberry Pi.

- [NOT15b] NOTTALL Ben, Daves JONES (2015b). *14. API - Output Devices — GPIO Zero 1.6.2 Documentation*. URL : [https://gpiozero.readthedocs.io/en/stable/api\\_output.html](https://gpiozero.readthedocs.io/en/stable/api_output.html) (visité le 13/10/2021).

Documentation pour la librairie Python "gpiozero" pour Raspberry Pi, partie output.

Librairie utilisée officiellement par Raspberry Pi.

- [NOT15c] — (2015c). *gpiozero — GPIO Zero 1.6.2 Documentation*. URL : <https://gpiozero.readthedocs.io/en/stable/index.html> (visité le 13/10/2021).

Documentation pour la librairie Python "gpiozero" pour Raspberry Pi.

Librairie utilisée officiellement par Raspberry Pi.

- [Osk17] OSKITONE (2017). *Four-Step Octaved Sequencer*. URL : <https://github.com/oskitone/four-step-octaved-sequencer> (visité le 03/10/2021).

Guide pour la réalisation d'un séquenceur à 4 pas.

Réalisé par un hobbyiste qui a documenté tous les éléments électroniques nécessaires et un tutoriel vidéo.

- [PAJ21] PAJANKAR, Ashwin (2021). *Practical Linux with Raspberry Pi OS : quick start*. ISBN : 978-1-4842-6510-9. DOI : 10.1007/978-1-4842-6510-9.

Explique comment utiliser l'OS Raspberry Pi.

Livre publié sur Springer; Ashwin Pajankar est un ingénieur informatique.

- [PLA21] PLACENTINO, Lucas (2021). *TRANH201INFO3-Sequencer : Code*. URL : <https://github.com/lucasplacentino/TRANH201INFO3-Sequencer/tree/main/code>.

*Repository* Github de Lucas Placentino, utilisé le stockage et *version control* du code du séquenceur.

- [Pra20] PRATIQUE PC (2020). *Tuto Raspberry Pi 4 : Premier démarrage, installation, configuration et sécurisation*. URL : <https://www.youtube.com/watch?v=aIxZzDWhvvQ> (visité le 20/10/2021).

Présente le premier démarrage et la configuration du Raspberry Pi 4 en serveur.

Vidéo réalisée par un hobbyiste.

- [Ras14] RASPBERRY PI (2014). *Raspberry Pi Documentation - Getting Started*. URL : <https://www.raspberrypi.com/documentation/computers/getting-started.html> (visité le 20/10/2021).

Documentation officielle pour la Raspberry Pi.

Provient du site officiel de Raspberry Pi.

- [Ras18] RASPBERRYPI.ORG (2018). *Raspberry Pi 3 Model B+*. URL : <https://static.raspberrypi.org/files/product-briefs/Raspberry-Pi-Model-Bplus-Product-Brief.pdf> (visité le 24/10/2021).  
Fiche produit Raspberry Pi 3 B+.
- [REU14] REUTER, Jürgen (2014). « Case Study : Building an Out Of The Box Raspberry Pi Modular Synthesizer ». In : Linux Audio Conference 2014. URL : <http://lac.linuxaudio.org/2014/papers/12.pdf> (visité le 02/10/2021).  
Création d'un synthétiseur modulaire sur base de Raspberry Pi.  
Provient d'une conférence Linux Audio.
- [SIE06] SIEVERS, Beau (2006). « A Young Person's Guide to the Principles of Music Synthesis ». In : URL : <http://beausievers.com/synth/synthbasics/> (visité le 18/10/2018).  
Explique les principes de la synthétisation de musique.  
Beau Sievers est un chercheur post-doctorant en psychologie (dont psychologie de la musique) à la Harvard University, et est un compositeur et passionné de musique électronique.
- [SIY19] SIYU, Wu (2019). *i2clcd : driver for LCD1602/2002/2004 with I2C adapter, for Raspberry Pi or other device*. URL : <https://github.com/WuSiYu/python-i2clcd> (visité le 11/10/2021).  
Driver Python pour LCD avec I<sup>2</sup>C.  
Wu Siyu est un.e hobbyiste et étudiant.e à la Beijing University of Technology.
- [Sor18] SOREN (2018). *Using a push button with Raspberry Pi GPIO | Raspberry Pi HQ*. URL : <https://raspberrypi.hq.com/use-a-push-button-with-raspberry-pi-gpio/> (visité le 12/10/2021).  
Contient un tutoriel sur comment utiliser un bouton sur une Raspberry Pi.  
L'auteur, un hobbyiste d'électronique, publie différents articles sur le Raspberry Pi.
- [Sou17] SOUND SEMICONDUCTOR (2017). *SSI2144 datasheet*. URL : [https://synthronics.de/wp-content/uploads/2018/08/SSI2144\\_datasheet\\_rev.2.2.pdf](https://synthronics.de/wp-content/uploads/2018/08/SSI2144_datasheet_rev.2.2.pdf) (visité le 03/10/2021).  
Datasheet d'un VCF utilisable en alimentation 5V.  
Provient du fabricant Sound Semiconductor.
- [SpaSd] SPARKFUN (S.d.). *SparkFun Logic Level Converter - Bi-Directional*. (<http://cdn.sparkfun.com/datasheets/BreakoutBoards/BSS138.pdf>). URL : <https://www.sparkfun.com/products/12009> (visité le 12/10/2021).  
Datasheet Logic Level Shifter.

- [SUI98] SUITS, Bryan H. (1998). *Frequencies of Musical Notes, A4 = 440 Hz*. URL : <https://pages.mtu.edu/~suits/notefreqs.html> (visité le 20/10/2021).  
Table de fréquence de chaque note de musique.  
Prof. Bryan H. Suits de la Michigan Technological University est professeur en physique de la musique.
- [Swi93] SWITCHCRAFT (1993). *35RAPC2AV 3.5 mm Mono PC Board Mount Jacks*. URLbis : [https://www.switchcraft.com/Drawings/35rapc2av\\_cd.pdf](https://www.switchcraft.com/Drawings/35rapc2av_cd.pdf) ; URLtres : [https://www.switchcraft.com/Documents/Jack\\_Schematics.pdf](https://www.switchcraft.com/Documents/Jack_Schematics.pdf). URL : <https://www.switchcraft.com/Specification.aspx?Parent=425> (visité le 24/10/2021).  
Datasheet JACK 3.5mm Femelle Mono.
- [Tex01a] TEXAS INSTRUMENTS (2001a). *DAC7731 datasheet*. URL : [https://www.ti.com/lit/ds/symlink/dac7731.pdf?ts=1634758141424&ref\\_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FDAC7731](https://www.ti.com/lit/ds/symlink/dac7731.pdf?ts=1634758141424&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FDAC7731) (visité le 05/10/2021).  
Datasheet du DAC 7731.  
Provient du fabricant réputé Texas Instruments.
- [Tex01b] — (2001b). *PCF8574 Remote 8-Bit I/O Expander for I<sup>2</sup>C Bus*. URL : <https://www.ti.com/lit/ds/symlink/pcf8574.pdf?ts=1634014562707> (visité le 12/10/2021).  
Datasheet PCF8574.
- [Tex10] — (2010). *KeyStone Architecture : Serial Peripheral Interface (SPI)*. URL : <https://www.ti.com/lit/ug/sprugp2a/sprugp2a.pdf> (visité le 24/10/2021).  
Datasheet protocole SPI.
- [Tex14] — (2014). *NA555, NE555, SA555, SE555 datasheet*. URL : <https://www.ti.com/lit/ds/symlink/ne555.pdf?ts=1633593471682> (visité le 03/10/2021).  
Datasheet des éléments NA555, NE555, SA555, SE555.  
Provient de la marque réputée Texas Instruments.
- [Tex20] — (2020). *TI Precision Labs - DAC : les essentiels du DAC de précision*. URL : <https://www.youtube.com/watch?v=RMngwhh3-3Q> (visité le 10/10/2021).  
Présente les DAC de manière générale puis l'utilisation de leurs caractéristiques.  
Provient de la chaîne Youtube du fabricant réputé Texas Instruments.
- [THO12] THOMS, Volker (2012). *spidev : Python Spidev*. URL : <https://github.com/doceme/py-spidev> (visité le 07/11/2021).  
Librairie Python pour SPI avec la Raspberry Pi.

- [Sd] *Unit 7 : A/D and D/A Converter* (S.d.). URL : [https://www.ebookbou.edu.bd/Books/Text/SST/DCSA/dcsa\\_2301/Unit-07.pdf](https://www.ebookbou.edu.bd/Books/Text/SST/DCSA/dcsa_2301/Unit-07.pdf) (visité le 05/10/2021).  
Montre le fonctionnement et des exemples de DAC.  
Provient de la Bangladesh Open University, programme Computer Science & Application (DCSA 2301, UNIT 7).
- [WAT20] WATKISS, Stewart (2020). *Learn electronics with Raspberry Pi : physical computing with circuits, sensors, outputs, and projects*. ISBN : 978-1-4842-6348-8. DOI : 10.1007/978-1-4842-6348-8.  
Apporte énormément d'informations sur l'électronique avec la Raspberry Pi.  
Livre publié sur Springer; Stewart Watkiss, un ingénieur en électronique, est très actif dans la vulgarisation de l'électronique.
- [WILSd] WILSON, Ray (S.d.). *ADSR Envelope Generator (+/-12V or +/-15V)*. URL : <http://musicfromouterspace.com/index.php?MAINTAB=SYNTHDIY&VPW=1910&VPH=750> (visité le 12/10/2021).  
Explications et schéma d'un générateur d'enveloppe.  
Site d'hobbyiste(s) reprenant énormément d'informations, de projets, de tutoriels et d'explications sur les synthétiseurs analogiques.
- [WOO13] WOODALL Christopher, Benjamin HAVEY (2013). *Analog Synthetiser With AR Enveloping*. URL : <https://cwoodall.com/pdf/ec412-final-report.pdf> (visité le 18/10/2021).  
Présente la génération d'une enveloppe AR.  
Rapport d'un projet de Woodall et Havey de bachelier en ingénierie électrique à la Boston University.