



ENSEA

Beyond Engineering

Projet séquenceur (C-QUENCER 64)

Yaël CAVAILLE-VOISIN
Guilhem BOULARAN
Benoît LOMBARD

Alexis NAUD
Louis GEORGE



Concept & objectifs :

On souhaite créer une interface homme-machine qui permet de configurer une séquence audio et de la jouer (le mode de sortie étant originellement à déterminer).

On s'attachera aux paramètres suivants :

- Nombre de combinaisons offertes par notre outil (maximal)
- Possibilités de personnalisation du signal (maximales)
- Ergonomie des contrôles



Arturia BeatStep – Surface de contrôle MIDI et séquenceur

★★★★★ ~ 397

119⁰⁰ €



Teenage Engineering PO-33 Pocket Operator Sampler / Séquenceur KO

★★★★★ ~ 5 748

93⁰⁰ € Conseillé : 99,00€

Cahier des charges du C-QUENCER 64 :

Thème	Sous-thème	Description/exigences
Alimentation	<i>Alimentation filaire</i>	<i>Compatibilité maximale, faible consommation énergétique</i>
Audio	<i>Puce audio</i>	<i>Au moins 12 notes prises en charge, puce bien documentée</i>
	<i>Sorties</i>	<i>Compatibilité écouteurs, amplis, sortie Mono</i>
Samples	<i>Nombre de step</i>	<i>Au moins 12, maximal dans l'idéal</i>
	<i>Sélection du step</i>	<i>Interface de sélection à boutons (au moins 12 boutons)</i>
	<i>Modification de la hauteur</i>	<i>Sélection de l'octave du dernier step utilisé</i>
		<i>Sélection de la hauteur du dernier step utilisé</i>
	<i>Affichage de la séquence sonore</i>	<i>Affichage du step sélectionné</i>
		<i>Affichage du step en cours de lecture</i>
Modifier le signal audio	<i>Changer de forme d'onde</i>	<i>Sélectionner la waveform parmi les 3 disponibles</i>
	<i>Changer l'enveloppe</i>	<i>Modifier l'attaque, le decay et le release</i>
	<i>Filtrage</i>	<i>Low-pass basique avec fréquence et résonnance réglables</i>
	<i>Changer le BPM</i>	<i>On veut pouvoir changer le BPM rapidement et avec une plage de BPMs maximale</i>
Processeur	<i>Channels ADC</i>	<i>Le processeur doit contenir un nombre maximal de channels ADC (au moins 30)</i>
	<i>Type et accès</i>	<i>Le processeur intégré doit être programmable via l'USB-C en futurproofing</i>
Lecture de l'audio	<i>Mode lecture/pause</i>	<i>La sélection lecture/pause se fait via le bouton tap-tempo</i>
Système	<i>Marche-arrêt</i>	<i>Interrupteur On/Off</i>
	<i>Boîtier</i>	<i>Supporter la carte tout en minimisant le volume du système et en permettant un accès ergonomique à l'utilisateur</i>

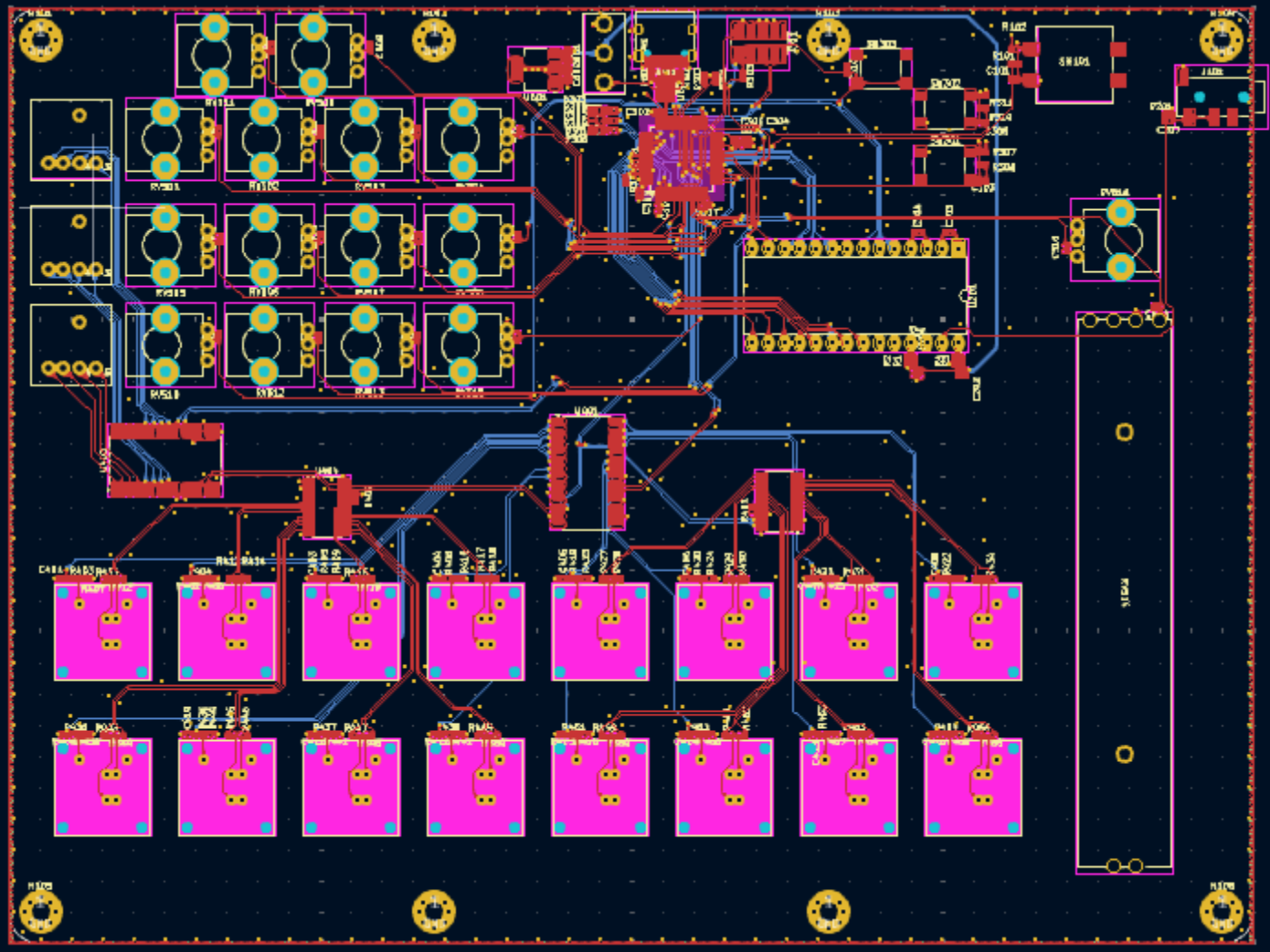
Répartition des Tâches & Déroulement du Projet :

	Yaël	Guilhem	Louis	Alexis	Benoît
Séance 1	Rédaction du cahier des charges				
Séance 2	Bons de commande, solutions technologiques, Màj du cahier des charges				
Séance 3	Conception du PCB, Màj des commandes				
Séance 4	Conception et routage du PCB, dépôt des livrables				
Séance 5	Architecture du code + Code		Soudure		Impression 3D
Séance 6	Code		Soudure + résolution de problèmes		Impression 3D
Séance 7	Code		Soudure + résolution de problèmes		Impression 3D
Séance 8	Code + livrables		Soudure		Impression 3D + livrables

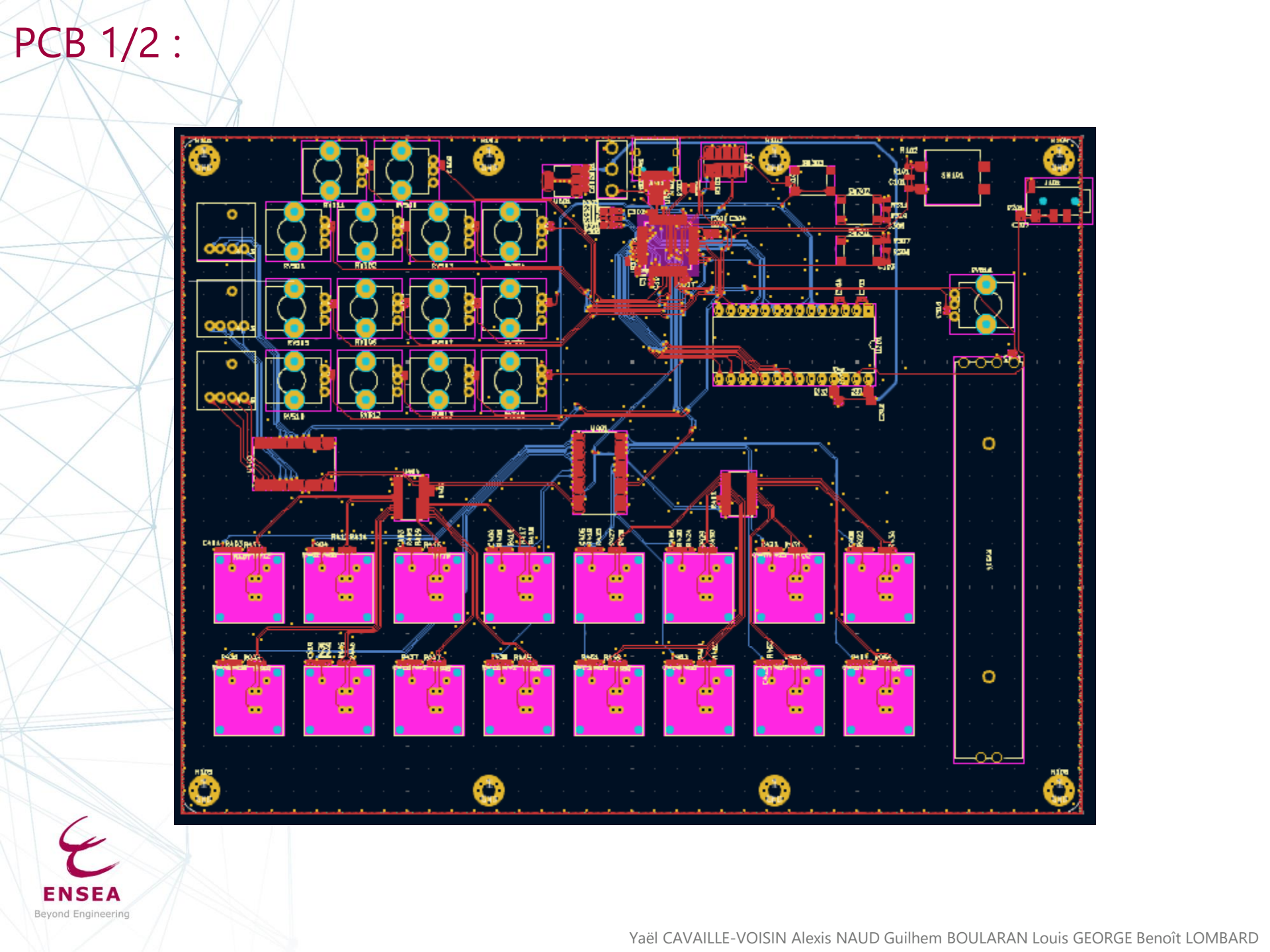
Solutions techniques retenues pour le C-QUENCER 64 :

Thème				Sous-thème	Exigences	Solution
Alimentation				Alimentation filaire	Compatibilité maximale, faible consommation énergétique	Alimentation en USB-C
RS						
Pos	I	P	Art	Désignation	Envoyez un seul fichier par mail	
					Qté commandée	UA
10	Z			Commutateur à glissière 3pos / 154-3587	10	PC
20	Z			Potentiomètre Rotatif / 729-3574	20	PC
30	Z			Bouton de potentiomètre / 465-9381	20	PC
40	Z			Commutateur rotatif 4 pos. / 244-4172	3	PC
50	Z			Interrupteur à bascule On-Off / 185-8229	1	PC
60	Z			Connecteur USB type C / 261-9682	1	PC
70	Z			Capuchon pour potentiomètre / 225-906	10	PC
80	Z			Capuchon pour potentiomètre / 225-883	10	PC
90	Z			Extenseur E/S, 16 Ports / 403-806	10	PC
100	Z					PC
110	Z					PC
120	Z					PC
				Selection de la hauteur du dernier sten utilisé	Potentiometre de selection de la hauteur	
Farnell						
Pos	I	P	Art	Désignation	Envoyez un seul fichier par mail	
					Qté commandée	UA
10	Z			Potentiomètre à glissière / 1688413	1	PC
20	Z					PC
30	Z					PC
				lecture	Bouton Tap-tempo en vert	
Modifier le signal audio		Changer de forme d'onde		Sélectionner la waveform parmi les 3 disponibles	Potentiomètre 4 positions	
		Changer l'enveloppe		Modifier l'attaque, le decay et le release	3 potentiomètres (attaque, decay et release)	
		Filtrage		Low-pass basique avec fréquence et résonnance réglables		
		Changer le BPM		On veut pouvoir changer le BPM rapidement et avec une plage de BPMs maximale	Bouton Tap-tempo	
Processeur		Channels ADC		Le processeur doit contenir un nombre maximal de channels ADC (au moins 30)	STM32-G474RE (48 channels ADC)	
		Type et accès		Le processeur intégré doit être programmable via l'USB-C en futurproofing	STM32-G474RE (programmable en USB-C)	
Lecture de l'audio		Mode lecture/pause		La sélection lecture/pause se fait via le bouton tap-tempo	Passage de Lecture à pause via pression du bouton Tap-tempo	
Système		Marche-arrêt		Interrupteur On/Off	Interrupteur levier	
		Boitier		Supporter la carte tout en minimisant le volume du système et en permettant un accès ergonomique à l'utilisateur	Boitier en 2 parties imprimé en 3D + 8 vis	

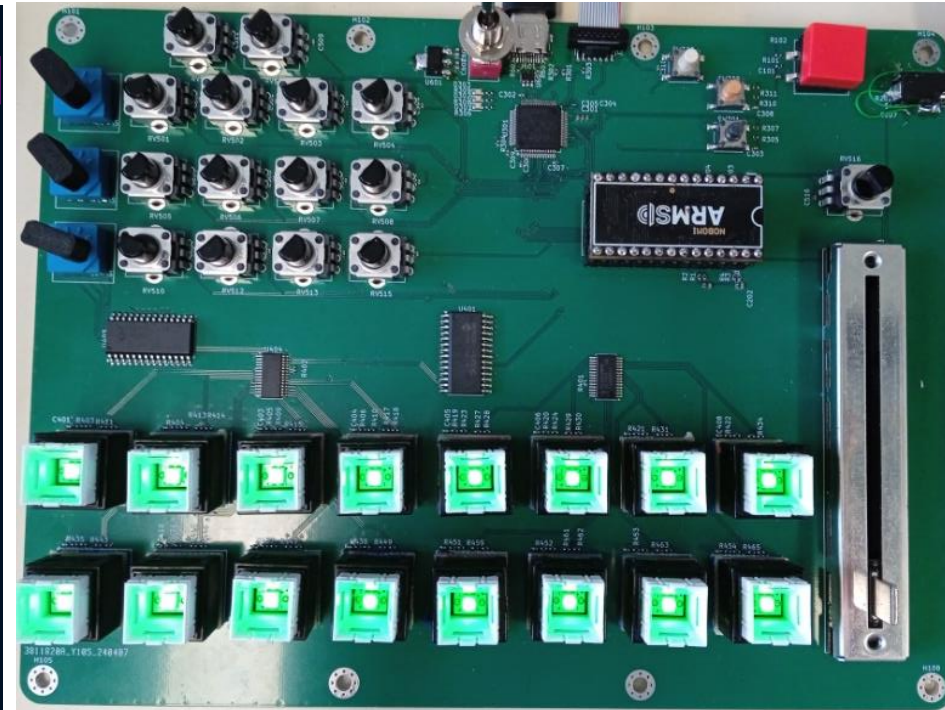
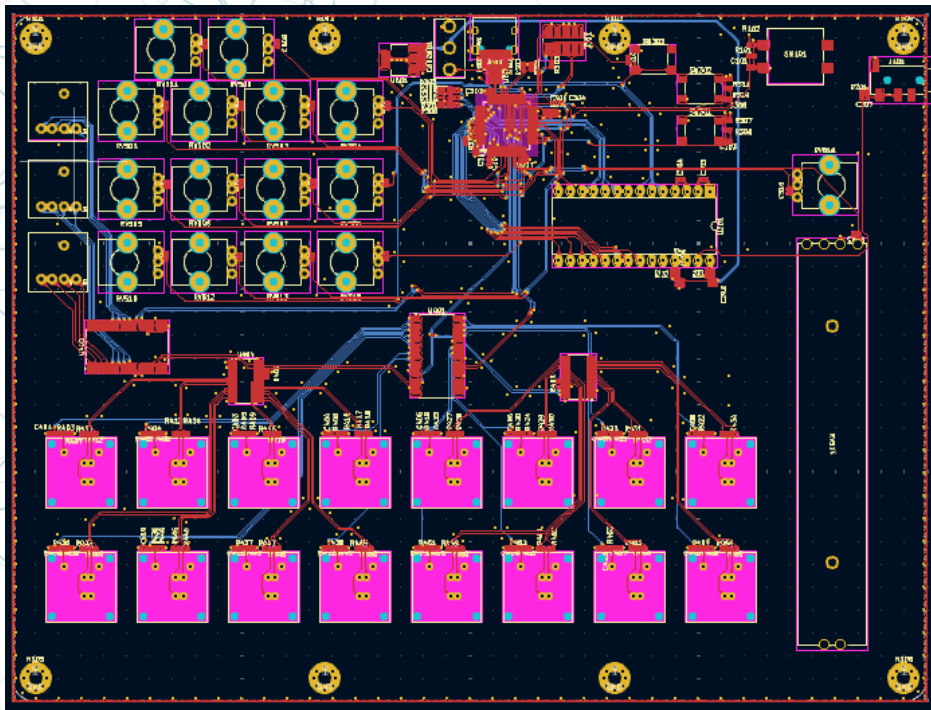
PCB 1/2 :



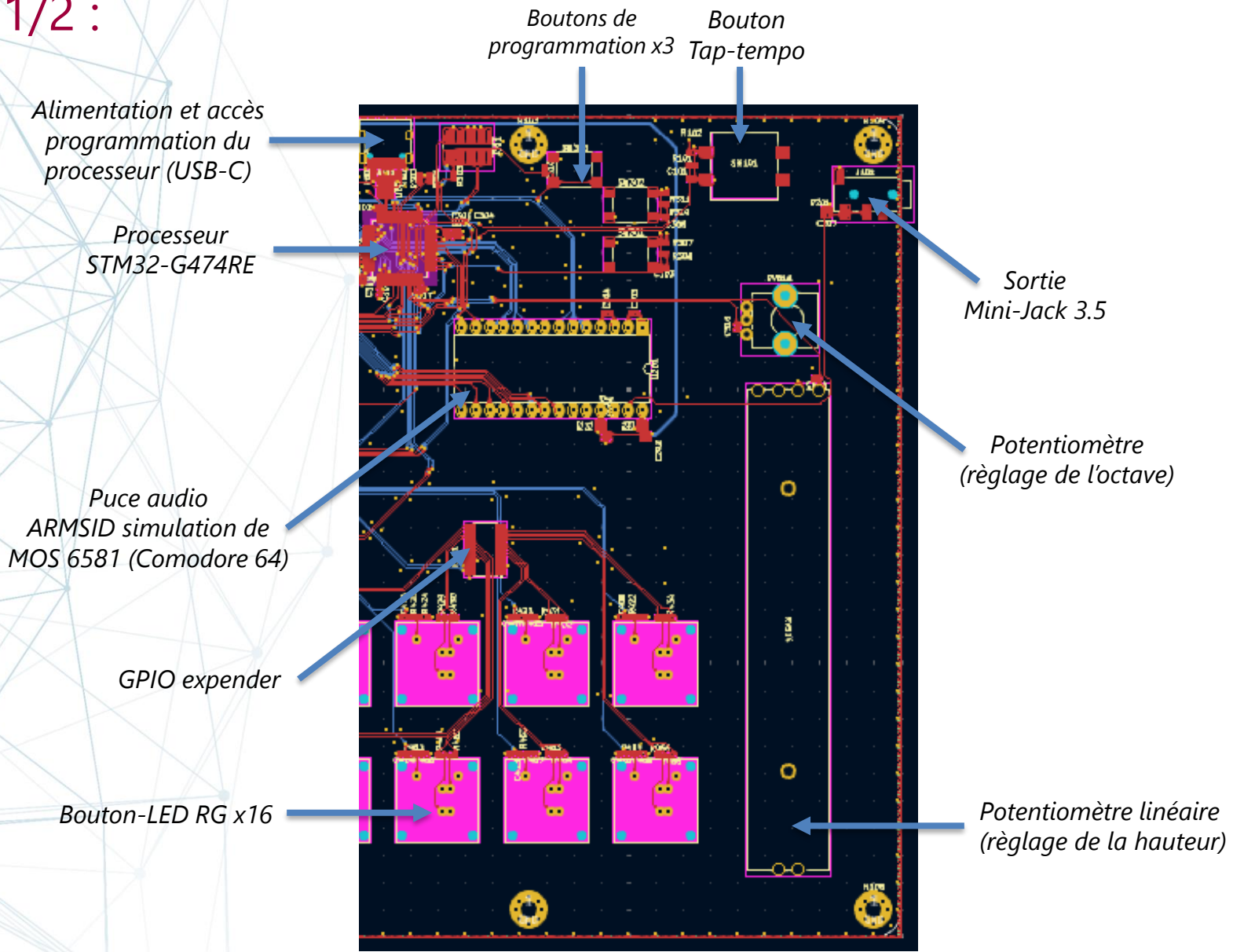
Yael CAVAILLE-VOISIN Alexis NAUD Guilhem BOULARAN Louis GEORGE Benoît LOMBARD



PCB 1/2 :



PCB 1/2 :



PCB 2/2 :

Potentiomètres de filtre
(fréquence et résonance)

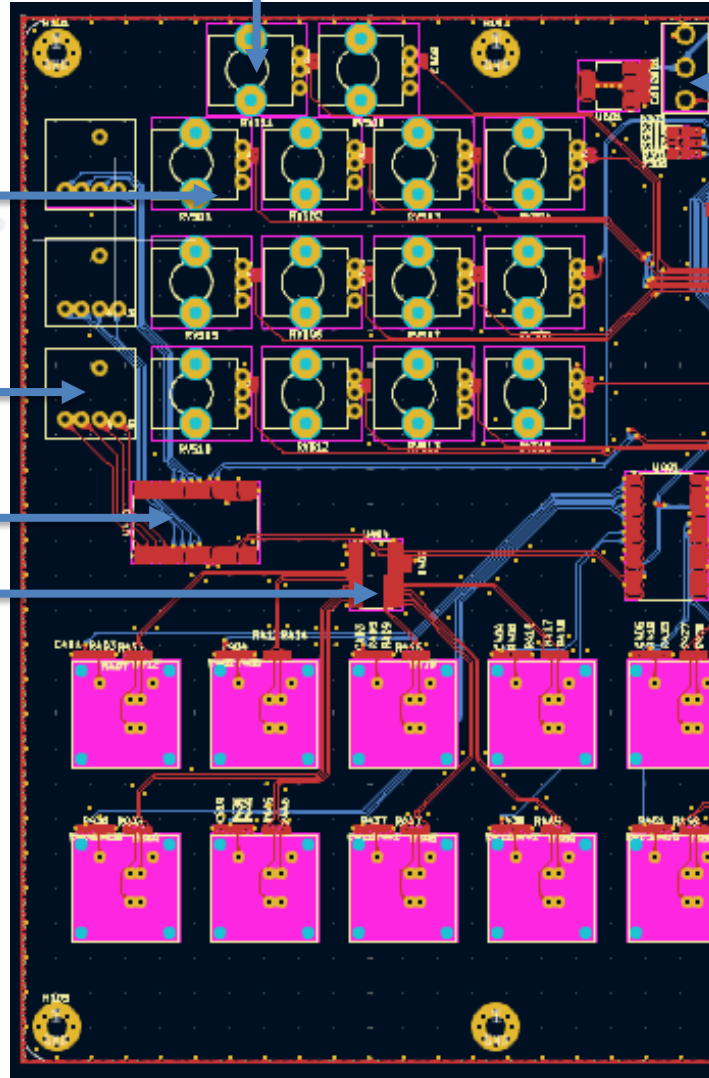
Potentiomètres d'enveloppe
(attack, decay, release)

Sélecteur de waveformes
(potentiomètre à crans x3)

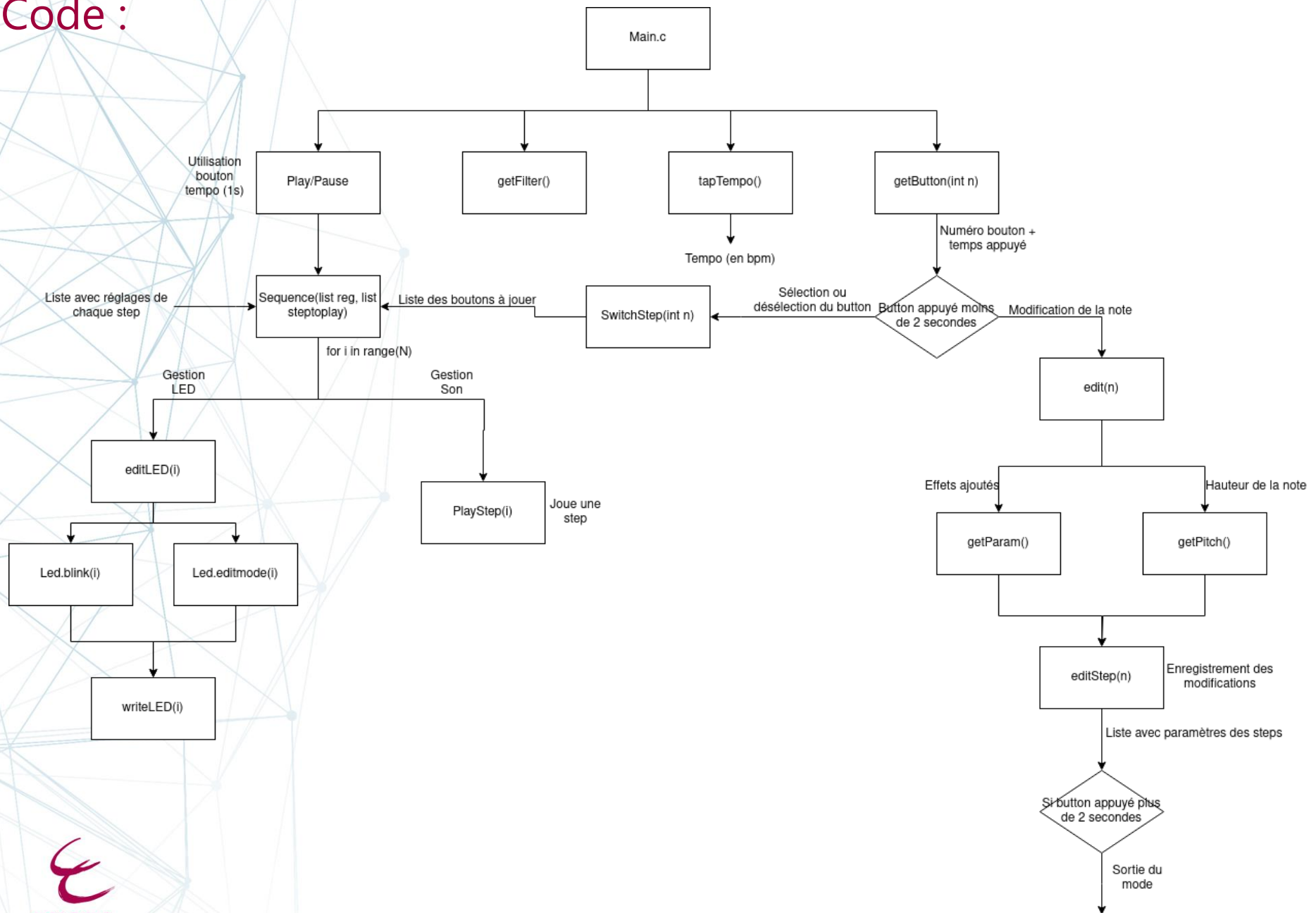
GPIO Expender

LED Driver

Interrupteur général



Code :



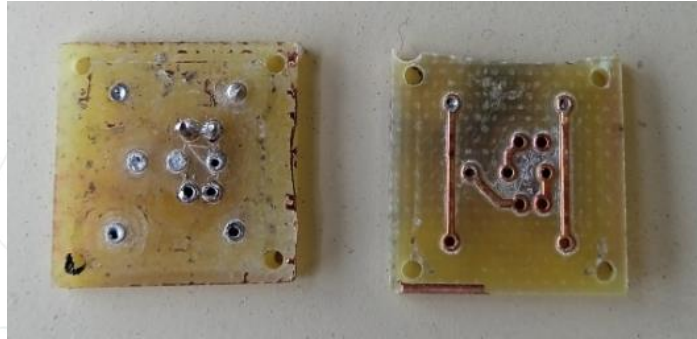
Code :

```
1 void Sequence (float tempo); // joue la sequence suivant le tempo
2 void tapTempo(); // règle le tempo en calculant la moyenne entre temps appuyés
3 void Play( int pitch); //prends un couple (octave,note) et envoie à la puce la note à jouer
4 void ParamVoice(Voice voice); //envoie à la puce les données pour modifier la voix avec les paramètres donnés par la step
5 void editStep(Step step ,int pitch , Filter filter); // change les paramètres de la step en fonction des paramètres actuels des potars
6
7
8 int getPitch (); // renvoie un couple (octave,note) en lisant les valeurs de potentiometres associés
9 int getParam(Voice voice); // recupere les paramètres d'une voix
10 int getFilter(Filter filter); // recupere les paramètres du filtre
11 int getButton(int button);
12 void getGPIOExpander();
13 void writeLED(Led led);
14 void writeMOS(Step step);
15
```


Défis et solutions :

- Erreur de WÜRT Electronik dans les commandes : bouton-LED RG au lieu de RB donc problème d'empreinte avec le PCB.

On a soudé des PCB-adaptateurs imprimés à l'ENSEA entre le PCB et les boutons-LED RG.



- Quelques difficultés lors de la programmation, car les tensions des broches de la puce audio ne correspondaient pas à ce qui est attendu d'après la documentation.

Perspectives :

- Manque de temps par rapport à la complexité du code à réaliser, il paraît difficile d'implémenter toutes les fonctionnalités espérées.

Cependant :





Projet Séquenceur (C-QUENCER 64)