Rapport de Design

**Ecole supérieure**

Électronique

Salle R112

Métronome électronique

Projet N° : 2318

**Réalisé par :**

Julien Decrausaz

**A l’attention de :**

M. Bovey

M. Moreno

M. Castoldi

**Dates :**

Début : 14 décembre 2023

Fin : 1er février 2024

Table des matières

[1 Description du produit 4](#_Toc157695514)

[1.1 Principe et caractéristiques 4](#_Toc157695515)

[1.2 Schéma bloc 4](#_Toc157695516)

[2 Schématique 5](#_Toc157695517)

[2.1 Alimentation 5](#_Toc157695518)

[2.1.1 Alimentation USB-C et batterie 5](#_Toc157695519)

[2.1.2 Convertisseur +3,3V 7](#_Toc157695520)

[2.2 Général 7](#_Toc157695521)

[2.2.1 Touchscreen 8](#_Toc157695522)

[2.2.2 Touchscreen Controller 8](#_Toc157695523)

[2.2.3 Buzzer 9](#_Toc157695524)

[2.3 Microcontrôleur 10](#_Toc157695525)

[2.3.1 Reset et debug 10](#_Toc157695526)

[2.3.2 PEC12 10](#_Toc157695527)

[2.3.3 Définition des pins du microcontrôleur 11](#_Toc157695528)

[3 Concepts du logiciel 12](#_Toc157695529)

[4 Perspectives et travail restant 13](#_Toc157695530)

[4.1 Travail restant 13](#_Toc157695531)

# Description du produit

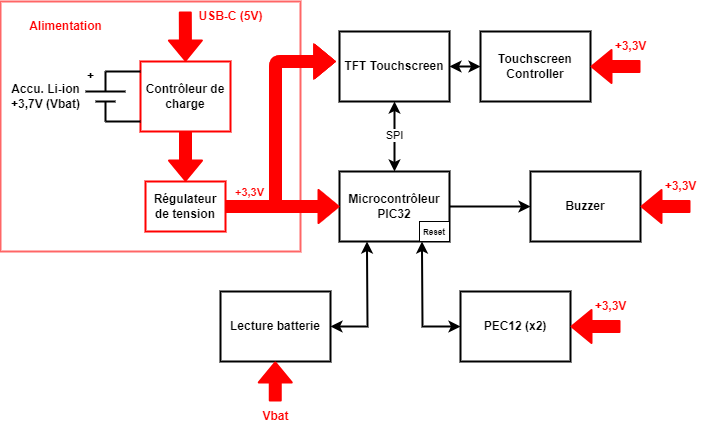
## Principe et caractéristiques

Ce métronome se base sur le modèle du site :

<https://www.imusic-school.com/fr/outils/metronome-en-ligne/>

Il dispose de configurations permettant à l’utilisateur de le paramétrer comme bon lui semble. Il dispose d’un écran TFT Touchscreen tactile offrant une interface minime et facile à utiliser. Deux PEC12 permettent de modifier le nombre de BPM (Beats Per Minutes) ainsi que le rythme du métronome comme sur l’interface du site web (croche, double-croche, etc…). Il peut varier entre 20 et 300BPM et dispose d’une plage de mesures entre 2 et 16 temps par mesure. Le produit est muni d’un boîtier fermé.

## Schéma bloc modifié



# Schématique

## Alimentation

### Alimentation USB-C et batterie



Lecture de tension

Figure 1

Le métronome peut être alimenté par un connecteur USB-c +5V ou par un accu. +3,7V. La tension passe d’abord par un contrôleur de charge, le MCP73871, qui servira à charger la batterie pendant que l’USB-c est branché. Un schéma de lecture de tension de la batterie est présent afin de garantir son bon fonctionnement. Selon si l’USB-C est actif ou inactif, la tension de sortie varie entre 3,7V (batterie) et 5V (USB-C). Cette tension va ensuite dans un convertisseur de tension pour l’abaisser à 3,3V.

#### Lecture de tension de la batterie

Ce montage est repris du projet de M. Zaco (N°2206).

Relié au microcontrôleur, il permet d’activer la lecture de la batterie. Permettant ainsi à la batterie de ne pas être lue en continu et de ne pas tirer un courant constamment sur celle-ci. La sortie du microcontrôleur pilote le MOSFET-N Q302 qui pilote le MOSFET-P Q301.

Le diviseur de tension R312 et R313 est calculé de cette manière :

* Uref = 3V (tension de référence de l’ADC, plage de variation de l’ADC de 0 à 3V)
* Ubat max = 4.2V (tension max sur la batterie)
* R313 = 4.7kΩ (choisie arbitrairement)
* R312 = la résistance à calculer

#### Contrôleur de charge MCP73871

J’ai choisi ce composant car il a déjà été utilisé dans d’autres projets. Ce qui me donne un aperçu du montage et de la schématique des composants autour.

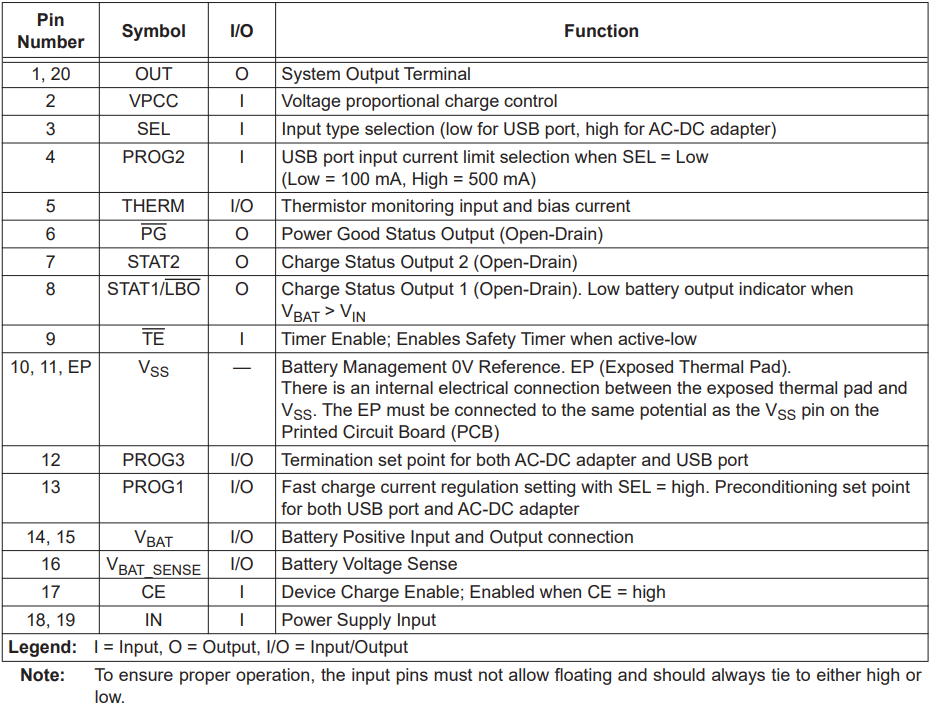


Figure 2 : Pin description du MCP73871 (tiré de la datasheet p17)

Les pins SEL, PROG2, /TE et CE sont des pins pouvant être mises à l’état bas ou haut selon l’utilisation désirée.

* **SEL :** Permet de sélectionner le type de source en entrée pour la limite de courant. Avec le pin en mode HIGH, le MCP73871 est capable de fournir un courant de 1,65A (typical) pour la charge de la batterie Li-Ion et le système. A l’état LOW, la source d’entrée est conçue pour fournir de la tension au système et de charger la batterie depuis un port USB. Dans mon cas, je lui mets un état LOW.
* **PROG2 :** est le pin de régulation du courant d’entrée. Un état logique LOW sélectionne une limite de courant de 100mA et un état logique HIGH limite le courant à 500mA. Dans mon cas, 100mA suffisent.
* **/TE :** pour « Timer Enable » permet d’activer/désactiver un timer interne. LOW pour actif et HIGH pour inactif. Je n’en ai pas l’utilité pour ce projet.
* **CE :** pour « Charge Enable ». Avec le pin à l’état LOW, la charge de la batterie est désactivée. Je l’ai donc mis à l’état haut pour activer la charger.

Comme je n’utilise pas de mesure de température, j’ai remplacé la résistance NTC du pin THERM par une résistance de même valeur (10kΩ).

Les entrées /PG, STAT2 et STAT1 servent d’indicateurs. Des LEDs y sont connectées afin d’avertir l’utilisateur d’un éventuel problème. J’ai choisi trois couleurs de LEDs différentes afin de les différencier. J’ai choisi les LEDs disponibles à l’ES puis j’ai calculé leur résistance en fonction de leur courant de fonctionnement et la tension de 3,3V.

### Convertisseur +3,3V



Figure 3

Ce convertisseur permet d’alimenter tous les composants du métronome. Il sera alimenté de 3,7V (avec la batterie) et 5V (avec USB). J’ai choisi ce modèle car je l’avais utilisé lors d’un précédent projet et qu’il avait bien fonctionné. Pour dimensionner le pont diviseur R314 et R315, j’ai utilisé la formule de la datsheet du composant en page 10.

Avec VADJ = 500mV :

Après discussion avec les professeurs, il a été décidé que l’entrée EN serait gérée par le microcontrôleur.

## Général

Lors de la phase de pré-étude ainsi que dans le cahier des charges, il était indiqué qu’un Ecran Epaper serait utilisé. Cependant, et après discussion avec Messieurs Moreno et Castoldi, nous avons conclu qu’un écran Epaper ne convenait pas à mon projet. C’est pourquoi ils m’ont orienté vers un écran tactile TFT. Grâce à cet écran, j’ai pu réduire grandement le nombre de composants présent dans mes schémas électriques. Car la plupart des réglages et paramètres s’effectueront via l’écran. J’aurai alors plus de Software à concevoir.

L’écran choisi est un ILI9341 de chez Adafruit. C’est un écran de 2,8’’ fonctionnant en SPI ou 8bits. Selon sa datasheet, l’écran doit utiliser un « Touchscreen Controller » car il n’en a pas. Adafruit suggère d’utiliser le TSC20071. C’est donc ce composant que j’ai utilisé pour mon schéma.

L’écran servira d’interface avec l’utilisateur. A partir de celui-ci, il pourra régler le nombre de temps par mesure, ainsi que le volume du buzzer.

### Touchscreen



Figure 4 : Ecran TFT tactile

Comme je vais utiliser l’écran en SPI, tous les pins du côté gauche de l’écran (Figure 4) doivent être en mode no-connect. Les pins X2+, X2-, Y+ et Y- sont reliés TSC20071. Tous les autres pins sont reliés au microcontrôleur sur des entrées SPI afin de gérer l’écran. J’ai placé des points de tests sur chacun de ces pins dans le but de pouvoir mesurer les signaux y passant lors d’un éventuel dépannage.

### Touchscreen Controller

Le « touchscreen controller » utilisé est le TSC20071 proposé par Adafruit. Il permet de convertir les données des pin X et Y en resistive touch. En consultant sa datasheet, j’ai connecté chacun des pins selon ce qui était proposé. Les pins SCL et SDA sont connectées à des résistances de Pullup de 10kΩ. Les pins A0 et A1 correspondent aux pins d’adresse déterminant les deux LSB de l’adresse du composant. Elles ont une adresse initiale qui peut être modifiée en les connectant au 3,3V. Dans mon cas, je n’ai pas besoin de les changer. C’est pour cela que j’au ajouté des jumpers SMD entre les résistances de Pullup et l’alimentation. Dans l’éventuel cas où les adresses devaient être modifiées, il suffit de souder les jumpers.

Figure 5

### Buzzer

Figure 6

Le buzzer que j’utilise est le SMTB-0827. Il a déjà été utilisé sur des projets précédents. C’est un buzzer magnétique de petite taille commandé par un MOSFET-N. Sur sa Gate, j’ai ajouté une Pulldown de **100kΩ** (R106) afin de le bloquer lorsque l’entrée est flottante lors du démarrage du uC. R105 sert de limitation de courant.

J’ai tenu compte du courant maximum que les pins de mon uC peuvent supporter pour la dimensionner (25mA). J’ai trouvé cette valeur dans la datasheet du uC en page 183.

Par question de sécurité, je vais utiliser une valeur plus élevée pour R105. J’ai choisi **180Ω**.

Comme j’utilise un buzzer magnétique, une diode de roue libre est présente afin de décharger le courant présent dans sa bobine. J’ai utilisé un modèle de diode disponible en stock à l’ES.

La résistance R107 permet de limiter le courant sur le buzzer. Pour son dimensionnement, j’ai tenu compte du courant maximum supporté par le buzzer (100mA). Une fois la résistance obtenue, je vais soustraire la valeur de la résistance de la bobine (valeur datasheet de **17Ω**) à celle-ci.

## Microcontrôleur

Le microcontrôleur que j’utilise est le **PIC32MX795F512H** qui possède 64 pins. J’ai choisi ce modèle car il dispose d’assez de pin SPI pour le Touchscreen ainsi qu’assez d’I/Os. Initialement, je devais utiliser un écran Epaper. Je comptais utiliser 16 I/Os connectées à des LEDs pour l’affichage des temps par mesure. Toutefois, j’utilise désormais un TFT Touchscreen. J’ai choisi d’afficher ces 16 temps par mesure sur l’écran plutôt que d’utiliser toutes ces I/Os. C’est pour cela que j’ai beaucoup de pins non connectés.

### Reset et debug

|  |  |
| --- | --- |
| Figure 7 | Figure 8 |
| Ces montages sont repris de la carte du kit de l’ES et sont également utilisés sur d’autres projets.  Le reset permettra de redémarrer le microcontrôleur manuellement. Un connecteur de Debug/Programmation est présent afin d’implémenter le futur Software. | |

### PEC12

Je vais utiliser deux PEC12 afin de régler le nombre de BPM et le mode du métronome. J’ai choisi de ne pas mettre ces paramètres réglables à l’écran afin de ne pas surcharger l’écran d’informations. J’ai repris le montage du PEC12 présent sur le kit PIC32 de l’ES.

Le premier PEC12 (ENC201), règle le nombre de BPM (Beats Per Minutes) allant de 20 à 300.

Le deuxième PEC12 (ENC202) permet à l’utilisateur de choisir le mode dans lequel le métronome va être utilisé. Comme sur le site web de référence, il y aura 5 modes : Croche, Double croche, Triple croche, Quadruple croche et Croche Terner.

Figure 9

### Définition des pins du microcontrôleur



Figure 10

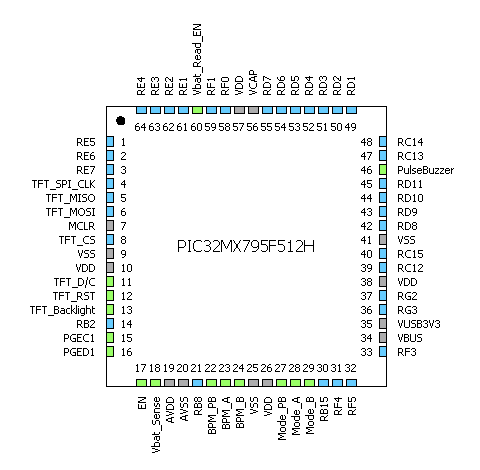


Figure 11 : Pin assignment

# Concepts du logiciel

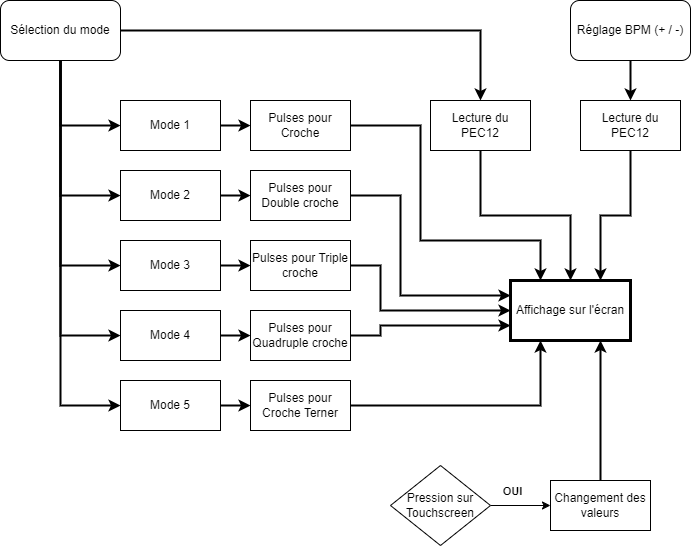


Figure 12

# Perspectives et travail restant

## Travail restant

1. Finir de choisir les composants et réaliser la BOM
2. Ajouter les footprints manquants sur le schéma
3. Choisir Accu Li-Ion
4. Commander les composants critiques
5. Design et routage du PCB
6. Monter le PCB et effectuer les tests hardwares
7. Réaliser le Software
8. Tests et mesures finaux

## Perspectives

Lausanne, le 1er février 2024

Julien Decrausaz