

# Cahier des charges : *LaBalek*

*Plus qu'une MotherBoard moins cher qu'une imprimante!*

## **Introduction**

L'impression 3d résine est pour l'instant un procédé coûteux dont tous les acteurs sont des entreprises proposant des machines avec une architecture fermée.

Notre projet est donc d'apporter une alternative en créant une carte de contrôle pour imprimante à résine de modèle DLP ( [Digital Light Processing](#) ) aussi connu sous le nom MSLA (Mask Stereolithography).

Le but de LaBalek est de faire entrer sur le marché de l'impression 3d personnel une MotherBoard low cost et *Open Source*.

## **Le Principe de LaBalek**

LaBalek est une carte mère conçue spécifiquement pour contrôler une imprimante 3D à résine liquide, qui utilise la lumière UV pour solidifier cette résine sur commande. Le projet vise à contrôler une dalle LCD, une matrice de leds UV ainsi qu'un moteur pas à pas.

Un deuxième écran LCD sert d'interface utilisateur tactile, et un menu permet de sélectionner les modèles à imprimer, d'afficher des informations de debug et de modifier certains réglages.

Les informations liées à l'impression elle-même (telles que les tracés à isoler, le temps de pause et l'épaisseur de chaque couche) sont contenues sur une carte SD.

Le modèle 3d est produit couche après couche en laissant les UV traverser de manière sélective la dalle LCD. Les rayons UV vont solidifier la forme dessinée par les pixels de l'écran LCD. Entre chaque couche, un moteur pas à pas est utilisé pour déplacer le plateau verticalement avec un maximum de précision.

Lorsque l'utilisateur sélectionne un fichier à imprimer, l'impression se lance et la ventilation démarre automatiquement. Des capteurs permettent de vérifier que le couvercle de l'imprimante est bien fermé avant l'allumage des LEDs UV.

## **Descriptif technique :**

- Contrôler un écran LCD sans rétro-éclairage.
- Gérer la ventilation de l'imprimante.
- Contrôler la matrice de LEDs UV permettant une cure précise de la résine photosensible
- Contrôler avec précision la hauteur du plateau d'impression à l'aide de commandes envoyées à un moteur pas à pas.

## FAQ

1. Quelle est notre motivation ?

R. Les cookies!

Nous voulons créer un design simple et facilement reproductible, afin de concevoir une imprimante modulaire pouvant être assemblée et utilisée par tous et pour un coût aussi bas que possible.

2. Pourquoi vouloir faire tout en *Open Source* ?

R. Les imprimantes 3D à résine photosensible disponibles actuellement sur le marché utilisent toutes des contrôleurs propriétaires, ce qui augmente le coût de celles-ci ainsi que la dépendance de l'utilisateur vis à vis du fabricant pour l'entretien et la réparation de la machine. L'open source nous permet de garder le coût du projet aussi bas que possible tout en offrant aux utilisateurs la possibilité de comprendre le fonctionnement de leur machine, ce qui facilite grandement son entretien ainsi que le remplacement de pièces lorsque nécessaire.

3. Quelles sont les plus grands défis du projet ?

R. L'un des plus grand défis de ce projet sera probablement de trouver des alternatives aux composants les plus coûteux tout en gardant les fonctionnalités des motherboards déjà existantes sur le marché.

4. Software explications ?

R. La partie Soft du projet est principalement gérée par des logiciels extérieurs déjà existants, en effet il existe plusieurs logiciels nommés "slicer" qui permettent de décomposer les fichiers 3D en multiples couches 2D pouvant être analysés par une imprimante 3D. à l'état actuel du projet le choix du slicer reste encore à déterminer mais nos premières recherches nous orientent vers ChiTuBox ou Slic3r ayant déjà un export destiné aux imprimantes résine.

5. Pourquoi Utiliser L'IDE arduino?

R. L'IDE est déjà utilisé par la plupart des imprimantes et connu des amateurs comme des professionnels.

6. Bonus ?

R. Bien que la carte mère constitue le cœur de notre projet, nous essayerons de monter une imprimante 3D complète pour pouvoir démontrer son fonctionnement.

## **Mode D'emploi<sup>1</sup> de LaBalek:**

1. Vérifier les connecteurs et les branchements.
2. Mettre l'appareil sous tension.
3. Une suite de tests automatiques sera lancée.
4. Des leds indiquent le statut des composants.
  - a. Pour la première initialisation, naviguez avec l'écran tactile jusqu'à menu>réglages>bed level et suivez les instructions à l'écran.
5. Insérer une carte SD contenant un ou plusieurs modèles 3D dans l'emplacement pour carte SD.
6. Sélectionner le modèle 3d à imprimer dans le menu sur l'écran LCD.
7. Pendant le cycle de chargement et vérification du modèle 3d, inspectez le bac pour réapprovisionner/nettoyer si nécessaire
8. L'imprimante<sup>1</sup> attend un cycle d'ouverture/fermeture avant de lancer l'impression

---

<sup>1</sup> chaque imprimante 3D peut avoir une configuration différente, vérifier votre modèle. Le capteur de la porte peut être désactivé dans le menu>réglage>capteurs. Il n'est pas recommandé d'utiliser l'imprimante avec ce capteur désactivé.

# Protocoles de communication

## 1. General Purpose Input/Output (GPIO)

### 1.1. Explications théoriques:

Un connecteur GPIO offre à une [carte électronique](#) la possibilité de communiquer avec d'autres circuits électroniques. Ces connecteurs sont gérés par les pilotes du [noyau](#) du [système d'exploitation](#).

Un connecteur GPIO peut être configuré pour jouer le rôle d'une entrée (réception d'un signal) ou d'une sortie (émission d'un signal). Lorsqu'il est configuré en tant que sortie, on peut écrire dans un [registre](#) interne afin de modifier l'état d'une sortie. Lorsqu'il est configuré en tant qu'entrée, on peut détecter son état en lisant le contenu d'un [registre](#) interne.

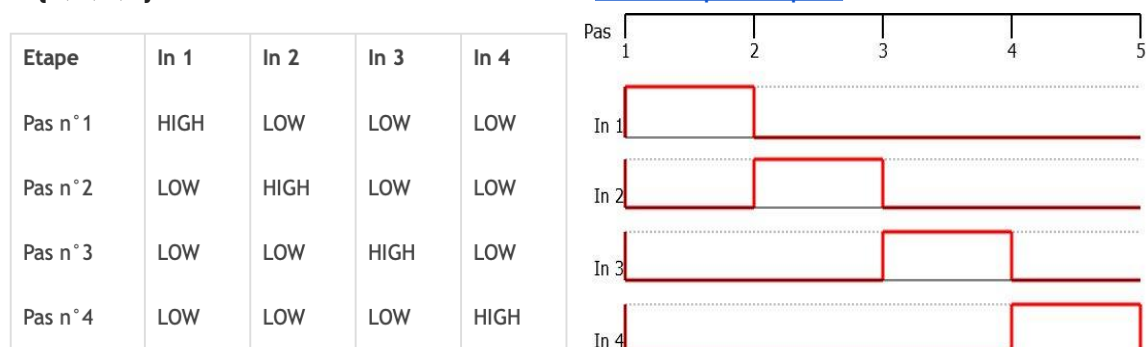
Un connecteur GPIO ne peut traiter que des signaux numériques, 1 ou 0, une connexion sur un support analogique nécessitera un [CAN/CNA](#) comme le MCP3008. Les périphériques GPIO peuvent également produire des interruptions et des événements d'accès direct à la mémoire (EDMA).

Un connecteur GPIO est généralement alimenté en 3.3Vcc et ne peut émettre que des courants de faible intensité, allant de 3 mA à 50 mA.

### 1.2. Implémentation

Le GPIO va être responsable pour la communication avec le moteur pas à pas, en générant des signaux HIGH LOW comme dans l'exemple:

In {1,2,3,4} est les câbles de la bobine du [moteur pas à pas](#)



Le GPIO va également contrôler les différents leds, capteurs et boutons. La possibilité de créer des interruptions va nous permettre de sécuriser la Motherboard des éventuelles dégâts pour mauvaise utilisation, et également protéger l'utilisateur d'une exposition aux rayons UV.

## 2. Display Serial Interface ( MIPI DSI )

### 2.1. Explications théoriques:

L'interface [MIPI DSI](#) (de l'anglais : Mobile Industry Processor Interface Display Serial Interface), créée par le groupe MiPi Alliance, est une interface standardisée de type **bus série** et de protocole de communication entre un **contrôleur d'affichage** d'un système (généralement mobile) et un périphérique d'affichage (typiquement, **écran à cristaux liquides**) ou autre périphérique à destination des données d'affichage.

### 2.2. Implémentation:

Ce protocole est imposé par le constructeur de la dalle LCD, il est similaire au [I<sup>2</sup>C](#) donc ne présente pas trop de complications pour l'application.

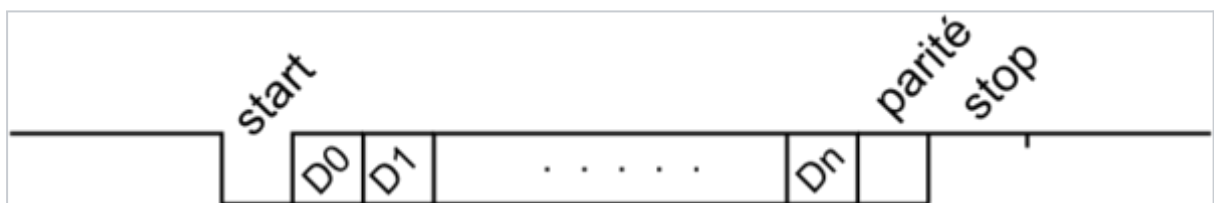
## 3. Universal Asynchronous Receiver Transmitter (UART)

### 3.1. Explications théoriques:

UART est un émetteur-récepteur asynchrone universel. Il a aussi été parfois désigné sous le nom de ACIA, pour Asynchronous Communication Interface Adapter (ce nom étant à présent tombé en désuétude).

En langage courant, c'est le composant utilisé pour faire la liaison entre l'ordinateur et le **port série**. L'ordinateur envoie les données en parallèle (autant de fils que de bits de données). Il faut donc transformer ces données pour les faire passer à travers une **liaison série** qui utilise un seul fil pour faire passer tous les bits de données.

Une trame UART est constituée des bits suivants :



Constitution d'une trame UART

- un bit de *start* toujours à 0 : servant à la synchronisation du récepteur
- Données : la taille des données est comprise entre 5 et 9 bits. Bits envoyés du LSB (bit de poids faible) au MSB (bit de poids fort).
- **Parité** : Paire ou Impaire (optionnel)
- Fin : Un bit de stop, toujours à 1. La durée de celui-ci varie entre 1, 1.5 et 2 (A l'utilisateur de choisir).

Le niveau logique de repos est le 1.

### 3.2. Implémentation:

Le UART va être utilisé pour la communication entre la LaBalek et l'ordinateur, d'abord comme port série, il sera une alternative au connecteur USB.

Il sera responsable aussi pour les sorties des ports de debug du microcontrôleur, où les ports flash et debug seront distincts pour faciliter le debug en cas de pannes d'un composants lors du flash.

## 4. Inter-Integrated Circuit bus (I<sup>2</sup>C):

### 4.1. Explications théoriques:

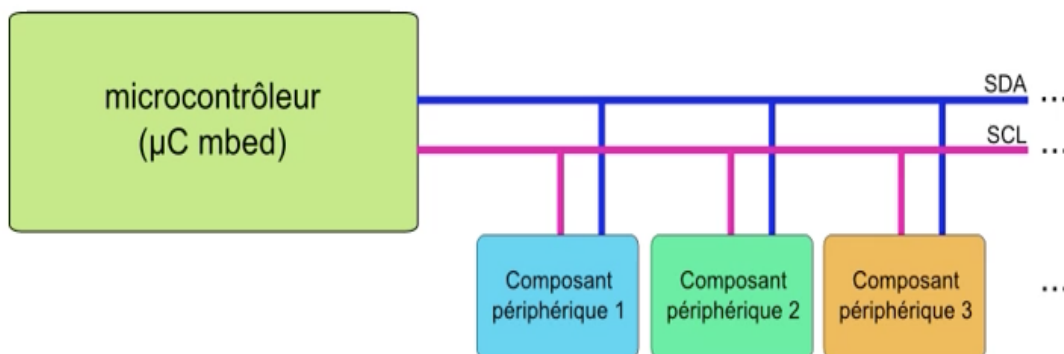
Il s'agit d'un protocole de communication en série synchrone, qui utilise un système hiérarchique type master-slave entre les composants.

C'est un bus multi-master.

Les données sont communiquées à l'aide de deux fils:

- SDA (Serial DAta)
- SCL (Serial CLock)

Dans le cas de notre écran LCD, elles doivent être connectées à une source d'alimentation positive (5V) à l'aide d'une résistance pull-up.



Chaque composant peut changer de mode de fonctionnement, mais généralement on préférera associer un rôle unique à chacun:

- (Rôle) Master ou Slave - (Mode) Transmit ou Receive

Chaque composant possède une adresse codée sur 7 bits (+1 bit Read/Write):

$$\text{Adresse} = A_6 A_5 A_4 A_3 A_2 A_1 A_0 0$$

Nombre maximum de périphériques slaves :  $2^7$  (128)

4 bits seront utilisés pour afficher un caractère, le transfert de donnée ne peut être effectif que lorsque le bus n'est pas occupé.

#### 4.2. Implémentation:

Le ILI9325D (écran LCD TFT utilisé pour contrôler les menus de l'imprimante) supporte les signaux de commande, écriture/lecture de données adressées aux périphériques slaves.

Deux adresses slaves de 7-bit (0111100 à 0111101 + R/W#) sont réservées. Le bit de lecture/écriture est respectivement 1 et 0.

La séquence est démarrée par un signal START (SDA 1->0 / SCL 1) par le composant master, suivi de l'adresse du/des slaves (1 à 3 octets).

Une fois identifié(s), un ou plusieurs mots de commandes sont envoyés afin de définir le statut du composant.

Le bit D/C# détermine le statut de chaque octet de donnée (commande ou donnée RAM). Suivant l'état de ce bit, s'ensuivra une série d'octets de données d'affichage ou de commande. S'il est mis à 1, les octets destinés à la RAM sont stockés à l'adresse du pointeur de donnée. Ce pointeur est automatiquement mis à jour et communique avec le périphérique concerné. S'il est mis à 0, les octets sont interprétés en tant que commandes et modifieront les paramètres du périphérique.

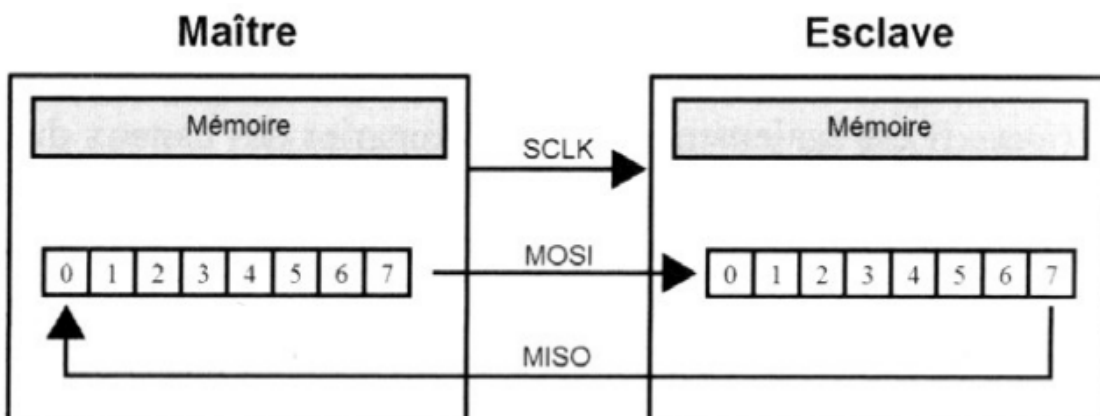
Seul le périphérique slave concerné renvoie après chaque octet le signal ACK.

La fin de transmission I2C est signalée à l'aide d'une condition STOP(SDA 0-1/SCL1).

### 5. Serial Peripheral Interface ( SPI )

#### 5.1. Explications théoriques:

Le bus SPI se distingue du protocole I2C par le fait qu'aucun adressage des esclaves n'est nécessaire : l'esclave devient actif au moyen d'une ligne de sélection de boîtier appelée SS (Slave Select). En cela, le bus SPI nécessite plus de fils que l'I2C mais est beaucoup plus rapide que ce dernier en mode standard.



## 5.2. Implémentation:

La communication entre l'écran et le ATMEGA se décompose en 3 phases :

Début de la communication :

Le ATMEGA met le signal SS à zéro de façon à indiquer à l'esclave que la communication commence (Slave Select actif à l'état bas). Côté CPLD, la détection du front descendant de ce signal va permettre d'initialiser les signaux et registres internes avant le premier coup d'horloge; le bit 7 doit alors imposer son état sur la ligne MISO (Master Input Slave Output).

Acquisition et transfert de bits :

\_\_\_\_\_Le maître fera ensuite basculer huit fois le signal d'horloge.

Sur les fronts montants, l'esclave enregistre les données provenant du fil MOSI et le maître mémorise sur le fil MISO (*Master Input Slave Output*) la valeur de chaque bit provenant de l'esclave : la communication est full duplex, il ne peut donc pas y avoir de collisions lors du transfert, l'arbitrage n'est alors pas nécessaire.

Les données sur les lignes MISO/MOSI sont mises à jour sur les fronts descendants du signal SCLK.

Fin de la communication :

Le maître met le signal SS à l'état haut pour indiquer que la communication est terminée.

## BILL OF MATERIALS<sup>2</sup>

Dans l'optique d'avoir une BOM a la fois peut coûteuse et accessible à tous la majorité des articles seront sourcé sur aliexpress.

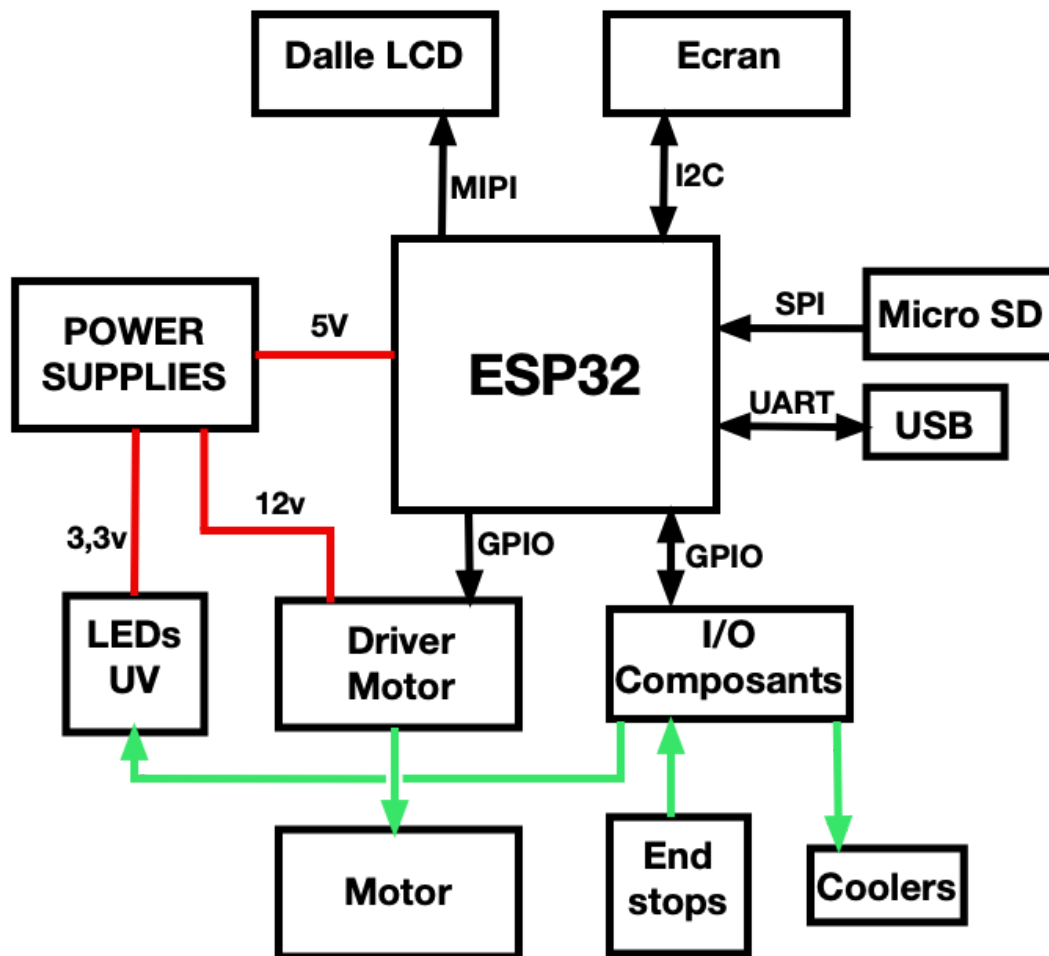
Bisous Jack Ma! 🤪

Composant	qte	Fournisseur	Lien
ESP32 wrover	1	aliexpress	<a href="https://tinyurl.com/yxc6ccfg">https://tinyurl.com/yxc6ccfg</a>
A4988 stepper driver	1	aliexpress	<a href="https://tinyurl.com/y5e4zv9w">https://tinyurl.com/y5e4zv9w</a>
Dual output power supply	1	aliexpress	<a href="https://tinyurl.com/yvojumyg">https://tinyurl.com/yvojumyg</a>
SD01A-01210 socket SD	2	aliexpress	<a href="https://tinyurl.com/y432ypb3">https://tinyurl.com/y432ypb3</a>
LCD TFT de 3.0 pouces	1	aliexpress	<a href="https://tinyurl.com/y3vu5pul">https://tinyurl.com/y3vu5pul</a>
USB Type B female	1	aliexpress	<a href="https://tinyurl.com/y42ttjq4">https://tinyurl.com/y42ttjq4</a>
lot / header pin m-f	1	aliexpress	<a href="https://tinyurl.com/yxsz4jxc">https://tinyurl.com/yxsz4jxc</a>
lot / 16 boutons	1	aliexpress	<a href="https://tinyurl.com/y329uk4d">https://tinyurl.com/y329uk4d</a>
lot / 100 leds	1	aliexpress	<a href="https://tinyurl.com/y2yvqz58">https://tinyurl.com/y2yvqz58</a>
lot / 300 resistances	1	aliexpress	<a href="https://tinyurl.com/y4nlplh2">https://tinyurl.com/y4nlplh2</a>
interrupteur à bascule	1	aliexpress	<a href="https://tinyurl.com/y5v5x5fk">https://tinyurl.com/y5v5x5fk</a>

---

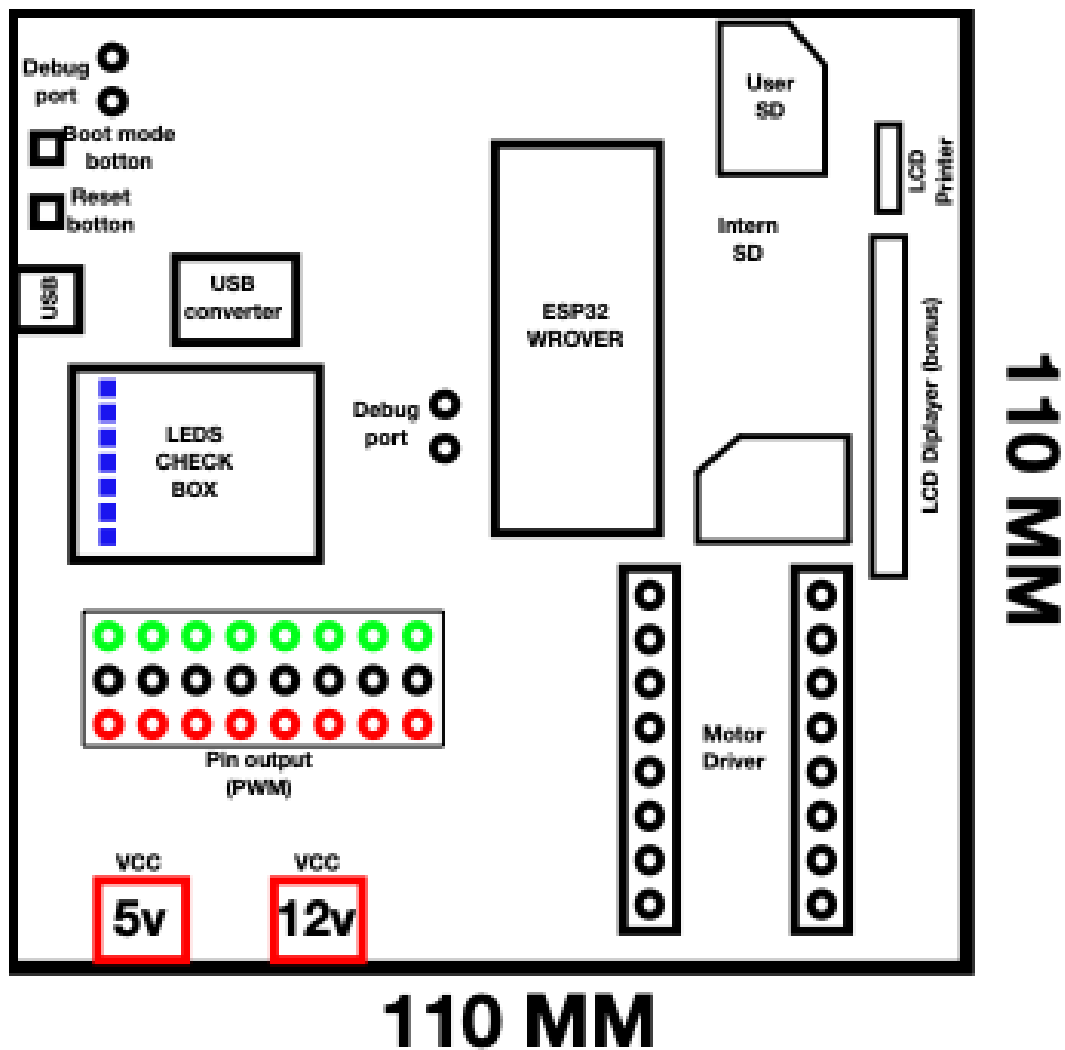
<sup>2</sup> La BOM n'as pas pris en compte les composants auxquelles on ne peut pas encore définir au préalable

## Block Diagram



(Le choix d'un ESP32 est du le but d'avoir une motherboard très peu coûteuse et facile à assembler pour n'importe qui que possède les schémas, sa facilité d'achat nous à aussi incentive à la choisir, pour un v2 software avec du Wifi (maybe ??) )

## Représentation graphique de la pcb:



cette representations est dans un but illustratif

Cette représentation ne prend pas en compte les consommables tels que résistances et capacités, ni les GPIO expander.

## **Bonus<sup>3</sup>**

- 1. Assemble une imprimante 3d capable d'imprimer pièces d'une complexité simple.**
- 2. Ecran LCD pour le menu de contrôle de l'imprimante.**
- 3. Capteurs pour une fonction d'auto-réglage de l'imprimante**
- 4. Optimization du software pour une impression plus précise, visant à imprimé une pièce de complexité moyenne au minimum.**
- 5. Système de Tilt pour le bac à résine, ce tilt vise à éviter la formation de parasites dans le bac.**
  - 5.1. Système d'auto-remplissage et auto-vidage du bac.**

---

<sup>3</sup> L'ordre des bonus liste ci dessous ne représente pas l'ordre à laquelle ils seront réalisés