

## **Projet ElecBoard :**



**Photo du prototype fonctionnel de l'ElecBoard**

« **ElecBoard** » est le nom que j'ai donné à mon projet de skateboard électrique. Plus précisément un « cruiserboard » électrique : un skateboard de petite taille spécialisé pour le déplacement. Le choix de ce projet vient du fait que je devais marcher 30 minutes à pieds pour aller l'école, en plus du trajet du bus. Le choix d'un cruiserboard s'est fait naturellement, en effet je devais pouvoir le transporter facilement pour les cours et le bus. Il existe déjà des cruiserboards en vente, comme le fait la marque BoostedBoard. Mais, le prix de ces cruiserboards est élevé (~1500€). D'où l'idée de construire l'ElecBoard. De plus je ne souhaitais pas transporter une manette de pilotage (dont sont équipés ces produits), c'est pour cela que j'ai décidé de développer une application pour piloter l'ElecBoard à partir d'un téléphone. Il pourra aussi servir de terminal, pour afficher la vitesse, l'autonomie et faire les réglages de l'ElecBoard. Toutes ces contraintes m'ont permis de définir un cahier des charges, celui de créer un cruiserboard à moindre coût, tout en ayant à minima les mêmes performances qu'un cruiserboard de BoostedBoard.

**Mon projet se divise en trois parties principales** que nous pouvons nommer par leur domaine : Mécanique, Electronique et Informatique. « Mécanique » désigne tout ce qui est conception du cache pour l'électronique et des choix des pièces. « Electronique » est la partie qui comporte le microcontrôleur, le moteur, l'ESC (Electronic Speed Controller) et le module BLE (Bluetooth Low Energy). Enfin la partie « Informatique » fait référence au développement de l'application sous Android.

**Le début de mon projet** a consisté à faire des recherches sur ce qui se fait déjà dans ce domaine. La plupart des projets de ce type consistait simplement à assembler des pièces conçus pour les skateboards électriques ; ce qui induit des coûts relativement élevés du produit fini. C'est pour ça que j'ai décidé de continuer mes recherches pour approfondir ma compréhension sur le fonctionnement d'un skateboard électrique : la batterie, l'ESC, le BMS (Battery Management System), le fonctionnement du moteur et le BLE. Après avoir rassemblé une documentation importante sur ces éléments, j'ai commencé la réalisation et les tests.

**L'électronique, le cœur du projet.** Je suis parti d'un composant connu mondialement, l'Arduino Nano équipé d'un microcontrôleur AtmegaA328P. Grâce à la généricité de l'environnement Arduino, j'ai pu développer beaucoup de modules autour de celui-ci. J'ai aussi trouvé facilement un module de communication BLE - le HC-08, avec des explications sur son fonctionnement sur internet. Il a été simple de faire communiquer l'Arduino et le module BLE. La partie entre le module BLE et l'application, que je détaille plus loin, a été plus délicate à réaliser. Le programme de l'Arduino au début est simple, il agit en fonction des données reçues du module BLE et agit sur l'ESC. Avant de commencer à créer mon propre ESC, j'ai choisi d'en prendre un issu du monde des voitures radiocommandées. En soit mon projet est une voiture radiocommandée en forme de cruiserboard. Les ESC possèdent un microcontrôleur pour interpréter les commandes qu'on lui envoie et agir sur les transistors MOSFETs

qui contrôlent le moteur brushless. Mais j'ai souhaité construire mon propre ESC pour me permettre de gagner en place et en consommation d'énergie. En effet, je voulais que sur un seul PCB (Printed Circuit Board) nous ayons tous les modules (ESC, BMS, BLE, microcontrôleur) de l'ElecBoard.

L'ESC de voiture radiocommandée fonctionne aussi comme abaisseur de tension. En effet, j'ai orienté mon choix de batterie sur la technologie LiPo pour sa densité énergétique et son utilisation très présente dans le monde RC (RadioCommande). Mais les tensions du moteur et du microcontrôleur sont différentes ; d'où l'intérêt de l'abaisseur de tension. J'ai choisi une batterie de 6S 50C de 5000mAh. Ce qui donne une batterie avec six cellules, de 3,7V en moyenne, en série qui peuvent délivrer cinquante fois leur capacité en « burst mode ». Ce qui signifie que nous pouvons demander à la batterie de fournir 250A pendant un laps de temps assez court. Nous obtenons une batterie de 22,2V en moyenne sur laquelle on peut tirer une grande quantité d'ampérage, parfait pour notre moteur. Pour le choix du moteur je me suis orienté sur un constructeur qui s'est spécialisé dans la construction de moteur de skateboard électrique : la réalisation de l'ElecBoard nécessite un moteur avec un fort couple et un faible KV (motor velocity constant). Le KV d'un moteur signifie le nombre de tour que celui-ci fera pour une tension d'un volt. On détermine le KV par le nombre de tours de son bobinage, la puissance des aimants et sa géométrie. Il existe que très peu de BMS, je n'ai donc pas trouvé de BMS qui convenait à mes exigences. J'ai dû pour l'instant à ne pas l'utiliser car il est présent dans les chargeurs de batteries LiPo.

**Impression 3D pour le cache.** J'ai utilisé l'impression 3D pour réaliser un cache sur mesure. Pour le modéliser j'ai utilisé le logiciel Fusion360, pour ensuite l'imprimer à l'aide du Slicer (« trancheur » en français) Cura. J'ai dû me reprendre à plusieurs fois pour obtenir les dimensions exactes ; en plus de la complexité d'impression de la pièce principale du fait de sa taille (300x100x40mm).

**Programmation sur Android Studio.** La partie programmation du projet s'est faite sur Android Studio avec mon téléphone qui me servait comme support de tests. Elle a été constituée de recherches sur le fonctionnement d'Android et de ses bibliothèques. La communication entre mon téléphone et le module fut la partie la plus compliquée. En effet, Android a connaissance du fait que nous pouvons utiliser le Bluetooth de quelqu'un pour connaître sa position. C'est pour ça qu'ils ont mis des sécurités sur les applications utilisant l'identification d'appareils Bluetooth aux alentours. Ces sécurités impliquent que l'application doit demander la permission de localiser grossièrement l'utilisateur en plus de celle d'utiliser le Bluetooth. La demande de ces permissions m'ont pris du temps car la découverte d'appareil Bluetooth est rare dans les applications et par conséquent j'ai pris du temps avant de comprendre mon erreur. J'ai réussi à résoudre cette erreur à l'aide de recherche sur internet des projets qui demandaient la même permission. Il fallait simplement que l'application à son premier lancement demande la permission d'accéder à la localisation bien qu'elle n'utilise jamais le GPS.

Le contrôle de vitesse se faisait à l'aide d'une slidebar, allant de 0 à 1. Pour obtenir un meilleur contrôle sur les faibles vitesses, j'ai préféré que la slidebar suive la fonction  $f(x) = x^2$  au lieu d'une fonction linéaire basique  $f(x) = x$ .

**En conclusion,** aujourd'hui, j'ai un prototype d'ElecBoard qui fonctionne (voir photo en haut de la page 1) et qui m'est revenu à 180 € (coût des pièces). En conclusion, j'ai apprécié réaliser ce projet qui m'a demandé beaucoup de recherches sur les fonctionnements des ESC, des BMS et du protocole BLE. J'ai eu plusieurs problèmes qui m'ont demandé patience et concentration pour les analyser afin de comprendre et de trouver des solutions. De plus ce projet offre encore de grandes possibilités d'amélioration, comme réduire la taille de l'électronique en créant son propre PCB, l'amélioration de l'application pour rajouter les configurations de l'utilisateur. J'ai aussi décrit l'avancement de ce projet sur mon blog. ( <https://erwan.dufour.io/> ).