Command’ergo

emile@landerretche.net

Par Emile Landerretche

Table des matières

❖ Présentation

➢ *Qu'est-ce que la Command'ergo*

Je vais vous présenter la Command'ergo. Tout d'abord il faut savoir que j'ai un frère en situation de handicap. Tout son côté gauche est paralysé..

Comme il n'a qu'un bras valide c'est difficile de jouer aux jeux vidéo. Il y a bien la Wii qui ne s'utilise qu'avec une seule main mais les jeux commencent à se faire vieux et la nôtre ne fonctionne plus. Tous les autres jeux requièrent maintenant l'utilisation de deux mains (que ce soit pour PC avec le clavier ou console avec les manettes) ce qui est impossible pour lui.

L'objectif de la Command'ergo est donc de proposer une commande qui émule les touches d'un clavier ou les déplacements d'une souris mais sous une forme utilisable par mon frère et donc ergonomique. Ma commande se veut aussi accessible et open-source pour la refaire chez soit facilement. J'ai donc prévu plus tard de déposer ce projet en open-source ce qui permettra à n'importe qui de réutiliser la manette en la refabriquant chez soi mais aussi de l'améliorer avec tous le codes sources disponibles. L'objectif étant de faire évoluer le projet et de le rapprocher de la perfection le plus possible avec l'aide des utilisateurs et de passionnés.

➢ *Comment je l'ai fabriqué*

Il y a 4 ans j'avais déjà fait un premier prototype entièrement imprimé en 3D (c'était un peu comme une souris mais avec un joystick et des boutons) malheureusement cette solution demandait trop de souplesse des doigts et j'ai vite abandonné l'idée. Quelque mois après j'ai vu qu'il était possible de faire communiquer une nunchuk (commande secondaire de la Wii) avec un Arduino c'est là que je me suis dit qu'il fallait que j'en utilise une pour mon projet. En plus de cela, utiliser la nunchuk permet de la recycler car la Wii n'est plus renouvelée et beaucoup arrivent en fin de vie, utiliser la nunchuk permet donc de ne pas la jeter. Malheureusement j'ai été confronté a beaucoup de difficultés pour fabriquer ce projet et j'ai vite arrêté. J'avais notamment vu que Microsoft faisait quelque chose d'assez semblable et cela m'a paru inutile d'en refaire une. Mais cette année je me suis dit que finalement la manette proposée par Microsoft n'était pas si accessible qu'elle ne le voulait et que ma « Command 'ergo » avait son utilité. Certes la manette de Microsoft est bien mais elle émule une manette ainsi il est impossible de jouer aux jeux qui ne la reconnaissent pas (et il y en a beaucoup), elle coûte aussi très cher car il faut acheter le boitier principal puis chaque périphérique indépendamment.

➢ *L'objectif final*

Ma commande se veut donc plus accessibles et simple d'utilisation. Elle émule les touches d'un clavier, les déplacements et les boutons d'une souris ce qui permet d'être compatible avec la grande majorité des jeux d'ordinateur. Chaque touche est personnalisable à souhait grâce à une application qui se charge de communiquer par port série avec la commande. Il est aussi possible de jouer à plusieurs, un deuxième port est prévu. L'objectif étant de permettre une rivalité entre deux personnes en situation de handicap mais aussi pourquoi pas avec une personne valide pour jouer dans la même situation.

❖ Fonctionnement

➢ *Introduction*

Pour l'utiliser c'est très simple. Il suffit de brancher le boitier par un câble USB standard a l'ordinateur, prendre dans une main la nunchuk, poser dans l'autre les boutons et commencer à jouer. Mais évidemment chaque jeu ne nécessite pas forcément les mêmes touches. C'est pour cela que j'ai créé une application. Elle permet de redéfinir chaque bouton ou axe de joystick de la manette en touches ou déplacements de clavier et de souris. L'application va ensuite communiquer avec le boitier et le « reprogrammer ».

Le logiciel de la Teensy s'appuie sur la librairie « nintendo extension ctrl » par dmadison https://github.com/dmadison/NintendoExtensionCtrl

➢ *Les 4 parties*

Concrètement la commande est constituée de 4 parties :

- Le boitier principal
- 2 nunchuk (commande secondaire de la Wii)
- 2 modules de bouton
- L'application de paramétrage

➢ *Le boitier*

Le boitier est imprimé en 3D. J'ai utilisé le logiciel FreeCAD pour modéliser le boitier sur mesure. Ce logiciel est libre et open source ce qui pourra permettre à n'importe qui, une fois déposé en open-source d'être capable de le modifier gratuitement.

L'objectif était de faire quelque chose de très compact. A l'arrière, il y a un trou pour permettre de brancher le câble USB et à l'avant les trous nécessaires pour

chaque commande. Sur le « sol » du boitier il y a des petites « entretoises » qui permettent de visser la veroboard (plaque sur l'électronique est soudé) sur le boitier. Il y a eu beaucoup d'essais pour réussir à faire correspondre exactement chaque trou au bon endroit mais maintenant tout est fonctionnel.

➢ *L'électronique*

Sur la veroboard il y a le « cerveau » du projet. C'est une carte Teensy LC on peut la souder directement, moi j'ai d'abord soudé des pins sur laquelle j'emboite la Teensy qui me permet d'avoir une carte amovible et donc réutilisable. La Teensy c'est une dérivée de la carte Arduino. Je l'ai choisi car elle petite, est équipée d'un processeur nativement équipé de la fonction HID (permet de communiquer avec l'ordinateur), et a 2 ports i2c ce qui permet donc d'utiliser deux nunchuk sans difficultés.

Les boutons secondaires se branchent grâce à un câble Jack. Les ports femelles sont reliés à la carte sur les pins 4 et 5 pour le premier et 2 et 3 pour le deuxième. Ils sont ensuite tous deux reliés à la masse. Il n'y a pas besoin de rajouter des résistances de PULLUP (mise au potentiel pour supprimer l'effet rebond) car elles sont incluses dans la Teensy, il suffit de l'indiquer dans le code. Les nunchuk sont reliés grâce à un petit module appelé « wiichuck » ce qui permet de brancher et débrancher les manettes à souhait.

Mais il est aussi possible de les souder directement à la carte pour éviter d'acheter le wiichuck (les nunchuks ne seront pas amovibles). Le premier wiichuck est connecté aux ports 22 et 23 et le deuxième aux ports 18 et 19. Ces ports ne sont pas choisis au hasard, ce sont des ports i2c : le protocole de communication de la nunchuk. Ces pins n'ont pas de résistances de PULLUP intégré il a donc fallu en rajouter. Ensuite, chaque wiichuck est relié à la masse et au +3.3V.

➢ *Les boutons secondaires*

Comme je le disais ci-dessus mon frère a une légère mobilité sur sa main gauche. Il était donc possible de lui ajouter des boutons très sensibles. J'ai donc choisi d'ajouter un boitier avec à l'intérieur deux micro-switch.

(C'est comme des boutons mais beaucoup plus petits et donc sensible) au-dessus il y a d'autres pièces imprimées en 3D tenues par un axe. Ces pièces prennent le rôle d'un levier qui appuie sur les micro-switch. Les micro-switch sont, eux, reliés au boitier par un port jack. Le tout reste donc toujours aussi pratique et amovible. Sur le port jack il y a trois parties : une pour le micro-switch de gauche, une autre pour le micro-switch de droite et la dernière est reliée à la masse.

L'avantage d'utiliser un port jack est qu'il est universel. Il est possible que certaines personnes n'aient pas de mobilité sur leur deuxième main. Les boutons pourront donc être remplacés par d'autres capteurs comme un capteur à ultrasons (de distance), qui

au bout d'un certain seuil émule une touche de clavier ou un accéléromètre qui détecte les mouvements de la personne etc…

Ce second port jack a donc une grande capacité d'adaptation puisqu'on pourrait y ajouter n'importe quel capteur finalement. Ce sont donc des pistes d'amélioration de la commande.

➢ *L'application*

L'objectif de l'application était de rendre la commande utilisable en autonomie par l'utilisateur le plus possible. Elle permet de réassigner chaque bouton ou joystick de la commande avec la touche, le clic ou le déplacement de souris correspondant au jeu avec lequel on veut jouer.

Je l'ai créé avec le logiciel Visual studio. L'installation ne nécessite aucune autorisation particulière et s'exécute directement sur une clé USB (comme un logiciel portable). L'interface est divisée en 4 parties principales. En haut il y a quelques consignes simples pour expliquer rapidement comment s'en servir et le titre. Le milieu est divisé en deux parties, à gauche il y a une liste avec toutes les touches que l'on peut assigner (a, b, c, …, clic gauche, ctrl, alt, f1, f2, flèche haut, etc..) et à droite il y a la représentation de chaque bouton et axe de joystick (2x car il y a les deux joueurs). Pour l'utiliser il suffit de sélectionner à gauche quelle touche on veut assigner et, ensuite recliquer, à droite, à l'endroit on l'on veut qu'elle soit. Si on veut utiliser les déplacements de souris sur un joystick il y a une case à cocher au milieu, un menu déroulant s'affiche ensuite et permet de sélectionner la sensibilité de la souris (varie de 1 à 9).

La partie du bas se charge, elle, de communiquer avec la commande par port série. A gauche un menu déroulant permet de sélectionner sur quel port série est branché la Command'ergo. Une fois fait, il suffit d'appuyer sur « envoyer » et l'application va créer une suite de nombres par rapport aux touches assignées et l'envoyer à la commande. La carte Teensy se chargera ensuite de déchiffrer ces nombres et vérifier si c'est bien l'application qui l'envoie et non un autre logiciel qui tenterait de communiquer avec elle. Elle le vérifie grâce à un checksum (l'application va envoyer à la fin de la suite de nombres les deux derniers chiffres de la somme des nombres qu'elle aura envoyés, La carte va elle aussi faire le calcul et vérifier si le checksum correspond) Si le checksum ne correspond pas, la carte va l'ignorer mais si il correspond elle va réassigner chaque bouton et joystick et les écrire dans la mémoire EEPROM (petit « disque dur » de la carte ce qui permet de pas recommencer à chaque fois en gardant tout en mémoire).

Ce n'est encore que la première version de l'application elle marche mais n'est pas parfaite il reste encore plusieurs pistes d'améliorations (que je détaillerai plus bas).

❖ Bilan, Difficultés et Améliorations

➢ *Difficultés*

Comme je le disais plus haut j'ai commencé ce projet il y a 4 ans par un « repose main » ressemblant à une souris que j'avais d'abord confectionner en pate a sel puis ensuite imprimer en 3D sur laquelle il y avait un joystick et des boutons mais c'était très compliqué il y a eu plusieurs prototypes, il fallait à chaque fois réimprimer le tout parce qu'à chaque fois il y avait un problème.

Finalement cette solution n'était pas si utile car même si elle avait marché, il n'y aurait que 3 boutons (1 de moins) et elle serait plus compliquée à utiliser. J'ai donc abandonné cette manière de faire. Et heureusement quelques mois plus tard j'ai découvert comment utiliser la nunchuk avec un Arduino ce qui est finalement tout bénéfice car cela permet d'avoir quelque chose de simple à utiliser et permet de réutiliser la nunchuk qui aurait été jetée sinon.

Pendant le programme de cette nouvelle version j'ai été confronté à d'autres difficultés. Tout d'abord pendant cette version, J'ai appris à me servir de la nunchuk et de sa bibliothèque. Ensuite, j'ai mis beaucoup de temps à comprendre la bibliothèque permettant d'émuler le clavier et à chaque fois que je m'en servais le jeu avançait par à-coup ce qui n'est pas le résultat attendu…en fait une fonction était prévue permettant de rester appuyé puis de relâcher quelque temps après. Ce « petit » problème a failli entrainer l'arrêt du projet car ce n'est qu'après 4 mois que j'ai trouvé la solution. Dans cette même version les touches sur lesquelles j'appuyais n'étaient jamais les bonnes, j'ai vite compris que le clavier que j'émulais était en qwerty il fallait donc aller dans l'ordinateur et changer le paramétrage du clavier ce qui n'était pas très pratique… J'ai donc préféré changer chaque touche tour à tour dans la bibliothèque pour qu'elles soient utilisables.

Une fois que cette version était à peu près complète j'avais donc une nunchuk qui émulait des touches déjà prévues. C'était un bon début, mais pas suffisant. Il n'y avait pas assez de boutons. Dans la deuxième version j'ai donc ajouté le module de boutons secondaires ce qui rajoutait des touches qu'il utilise avec sa deuxième main (mon frère peut légèrement bouger les doigts de sa main gauche) ces boutons devaient donc etre très sensibles pour qu'il puisse appuyer dessus sans effort. Puis j'ai aussi rajouté dans cette version la réassignation des touches pour chaque bouton ou axe de joystick par l'utilisateur. Pour ça j'utilisais la communication série avec la ligne de commande. Il fallait donc un logiciel qui s'occupe de ça : soit directement le logiciel Arduino, soit PuTTY par exemple.

Mais cette solution était limitée, on pouvait assigner uniquement des lettres minuscules (pas de caractères spéciaux, de déplacements de souris, de clic etc…) et ce n'était absolument pas ergonomique car il y avait un temps imparti obligatoire pour valider, cela voulait dire taper vite au clavier (pas très accessible pour mon frère). J'ai donc décidé dans cette 4eme version de m'initier à la création de logiciel qui sera donc ergonomique et permettra de se servir de toutes les touches du clavier et de la souris. Mais cette tâche était compliquée. Il m'a tout d'abord fallu choisir comment j'allais la

créer. Plusieurs solutions s'offraient à moi : j'ai découvert WinDev il y a une version gratuite j'ai donc commencé à développer dessus. Mais cette version était trop limitée et finalement impossible de faire tout ce que je voulais. J'ai ensuite découvert Qt. J'ai donc commencé à suivre des tutoriels pour l'utiliser mais après 12h de tutoriels vidéo je ne savais toujours pas faire une page vierge. Cette solution m'aurait pris beaucoup trop de temps rien qu'à apprendre à me servir du logiciel j'ai donc abandonné. Et c'est là que j'ai découvert Visual studio que je ne connaissais pas du tout. J'ai donc enfin réussi à créer mon application avec ce logiciel mais il m'a fallu tout de même beaucoup de temps pour apprendre à programmer en c# (langage que je ne connaissais pas avant).

Une fois que tout était fait je commençais à avoir quelque chose d'assez correct mais il manquait encore un élément important : de la rivalité J'ai donc fait une sorte de « symétrie » et refait le programme une deuxième fois. Mais la carte Arduino que j'utilisais n'était pas compatible pour 2 nunchuk il m'a donc du fallu en retrouver une nouvelle : la Teensy avec ces 2 ports i2c.

Maintenant il fallait que j'emboite tout ça « proprement », tout d'abord j'ai appris à souder sur la veroboard chaque composant. Ensuite pour le boitier, j'avais plusieurs solutions J'ai choisi l'impression 3D car nous venions d'avoir une imprimante 3D et c'est beaucoup plus simple de faire quelque chose sur mesure. J'ai mis quelques temps avant d'avoir un boitier correct car, plusieurs fois je me trompais dans les dimensions. Après plusieurs impressions, le boitier était prêt ! La première version a été faite avec le logiciel fusion mais ce logiciel n'était qu'une évaluation gratuite. Si je veux le publier il me fallait trouver une application entièrement libre et gratuite : FreeCAD. J'ai donc du tout refaire sur ce logiciel.

La dernière chose qu'il me restait à faire (et pas des moindres) était de rendre mon code propre et lisible par n'importe qui. Tout d'abord il a fallu réduire le nombre de lignes grâce aux boucles, un vrai casse-tête, car il y'avait énormément de répétitions. Ensuite il a fallu documenter chaque ligne de code sur les deux programmes : l'application et le code Arduino.

➢ *Améliorations*

Il reste encore beaucoup de pistes d'améliorations. J'aimerais par exemple refaire entièrement l'application. Elle est fonctionnelle, mais loin d'etre parfaite. Il faudrait par exemple ajouter le redimensionnement de la fenêtre car actuellement elle a une taille prédéfinie et est inchangeable. Cela peut paraitre un détail mais il faudrait modifier presque entièrement le code. Cela permettra ensuite donc de s'adapter à chaque taille d'écran. Ensuite j'aimerais ajouter la possibilité d'enregistrer des profils de jeu. Au lieu de ressaisir à chaque fois la même chose pour un jeu, il suffira de le faire une fois puis de l'enregistrer sous le nom que l'on veut. Les prochaines fois il n'y aura plus qu'à double cliquer sur ce profil pour l'envoyer. Pas besoin de tout ressaisir. Ce changement nécessitera aussi une refonte de l'application car pour enregistrer des profils il lui faudra l'accès aux fichiers de l'ordinateur. J'ajouterais aussi une fonction qui inverse la communication. Au lieu d'envoyer, on demande les touches qui y sont déjà et on les affiche. Ce pourrait être utile dans le cas où ce n'est pas nous qui avons configuré la

commande et que ce sont des touches utiles il n'y aurait qu'à récupérer puis l'enregistrer dans un nouveau profil. Et pour finir avec l'application j'aimerais rajouter un menu de paramètres complets avec des configurations plus poussées.

Au niveau de la commande il faudrait que je développe des nouveaux capteurs à la place des boutons pour ceux qui ne pourraient pas les utiliser. Je pense par exemple à un capteur à ultrason qui a un certain seuil de distance émule une touche. Il pourrait y avoir aussi un accéléromètre qui capte les mouvements. L'avantage est qu'on peut le mettre partout comme sur la tête de l'utilisateur qui pourrait déclencher un bouton en un mouvement de tête.

Plus tard j'ai le projet de rendre la commande compatible Bluetooth. Ce qui sera plus simple à utiliser en autonomie et il y aurait moins de câbles. C'est un nouveau défi car il faudra donc ajouter une batterie interne, un système de recharge, refaire le code de l'Arduino et de l'application etc…

➢ *Open source*

Pour que le projet soit entièrement open source il doit être entièrement et facilement re-faisable et même modifiable. Pour cela j'ai dû utiliser des logiciels gratuits pour le créer. Les modèles 3D ont été faits sur FreeCAD, la programmation de l'Arduino sur le logiciel basique d'Arduino et le Code de l'application sur Visual studio.

Au niveau du matériel il y a besoin de :

- Impression 3D (boitier et bouton secondaire) peut se faire en fablab ou dans un lycée/collège (3€)
- Processeur : Teensy LC (10€)
- Nunchuk (peut etre récupéré pour recycler à des voisins, amis… ou bien acheter sur internet)
- Port jack mâle/femelle (pour brancher les boutons secondaires)
- Wiichuck (pour brancher les nunchuk, pas très cher sur internet)
- Fils, PCB, visserie, résistance.

Finalement le projet couterait environ 40€ à produire et environ 2h à assembler et tout programmer.

❖ Photos, Schéma

Résultat final

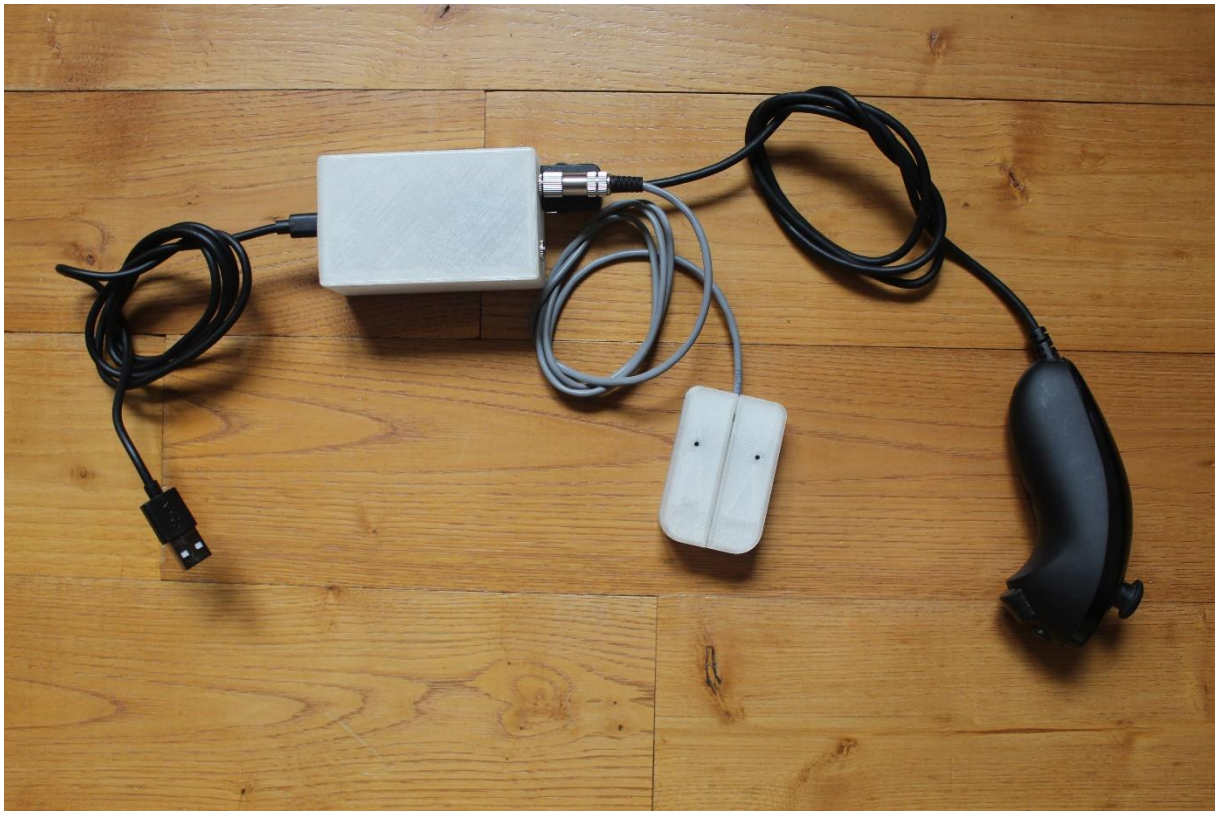
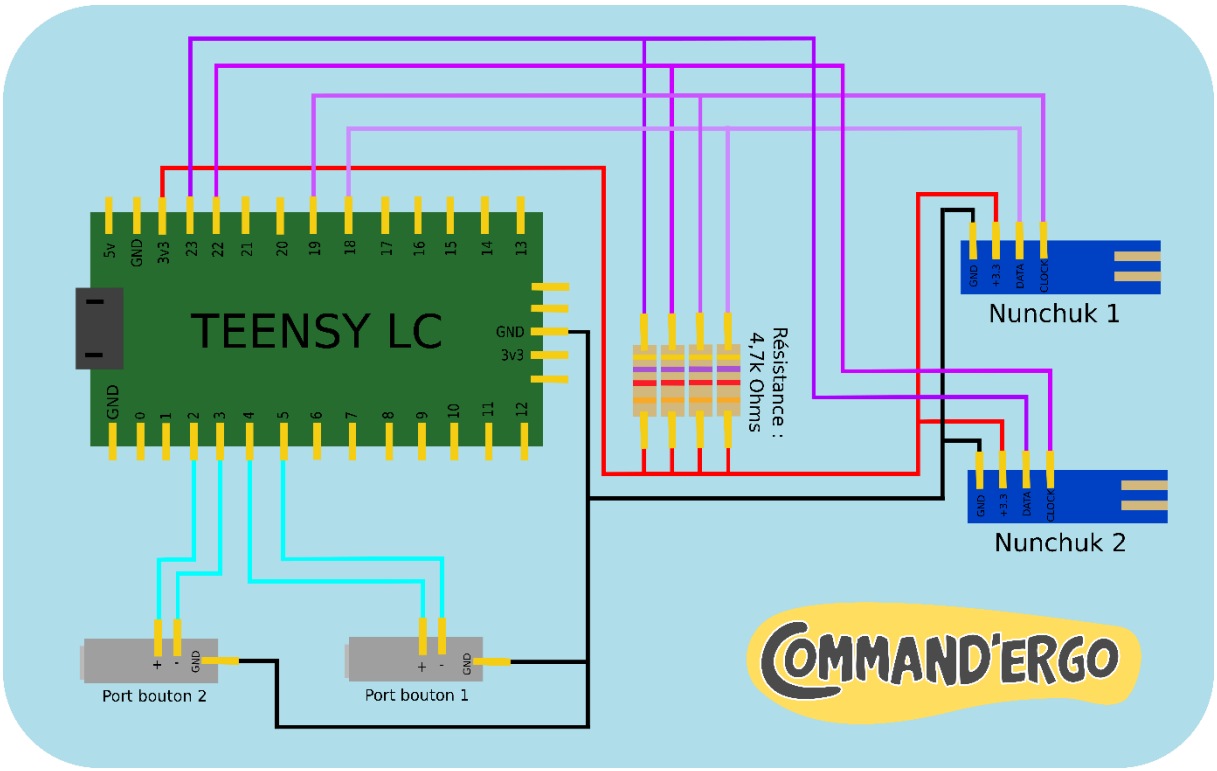
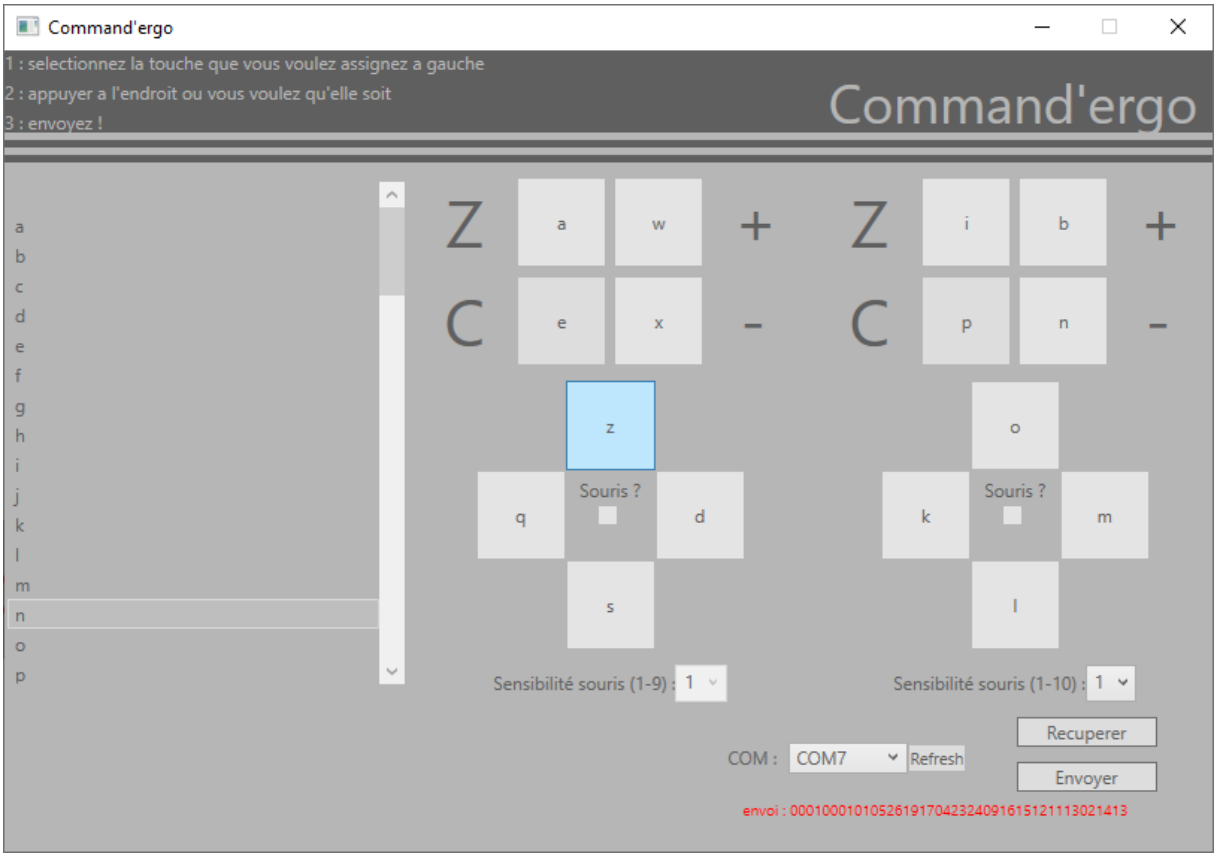


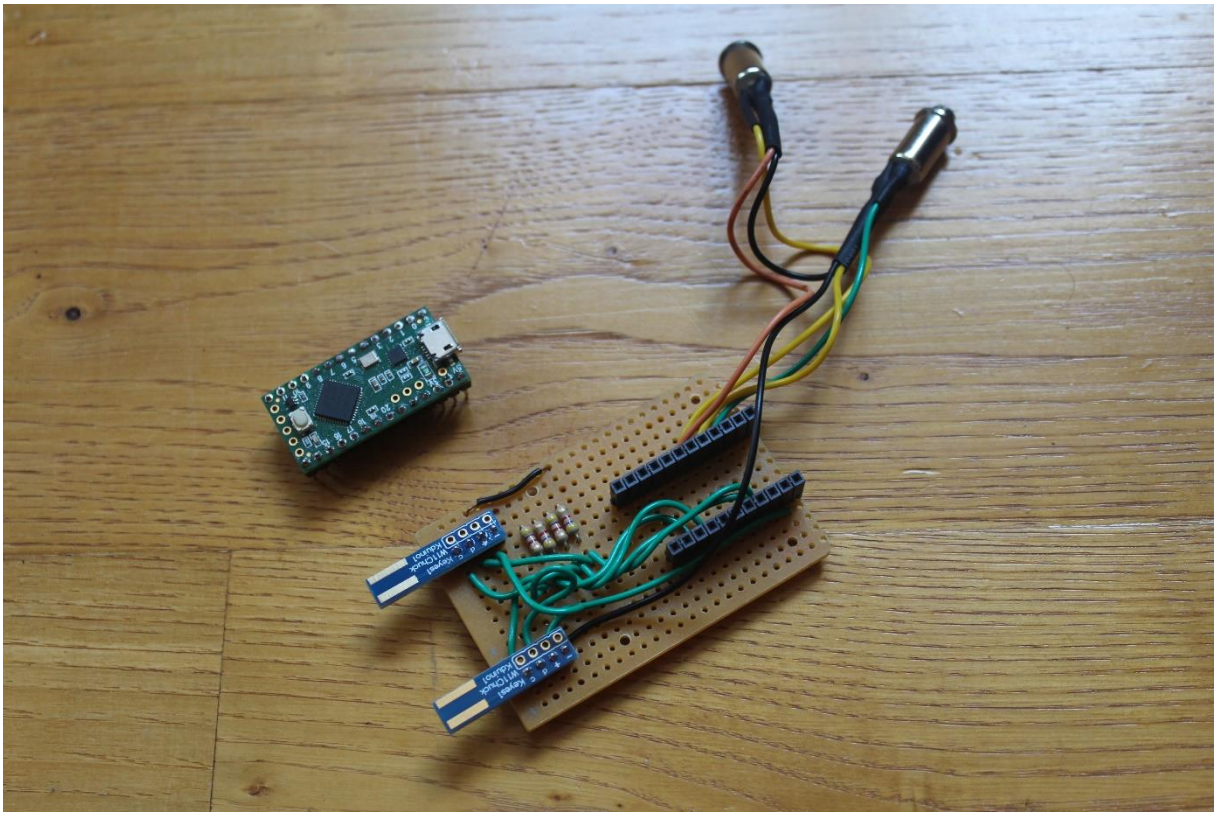
Schéma câblage Command’ergo + logo



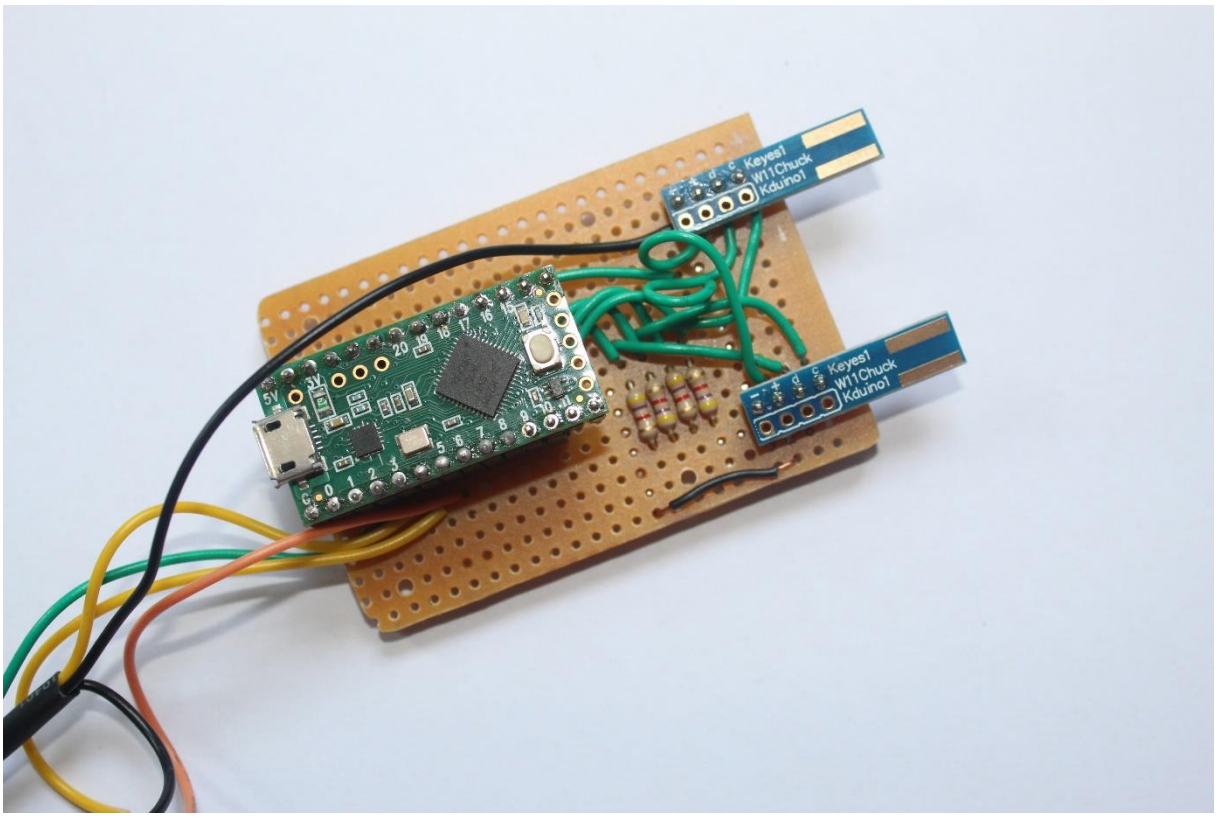
Application Command’ergo



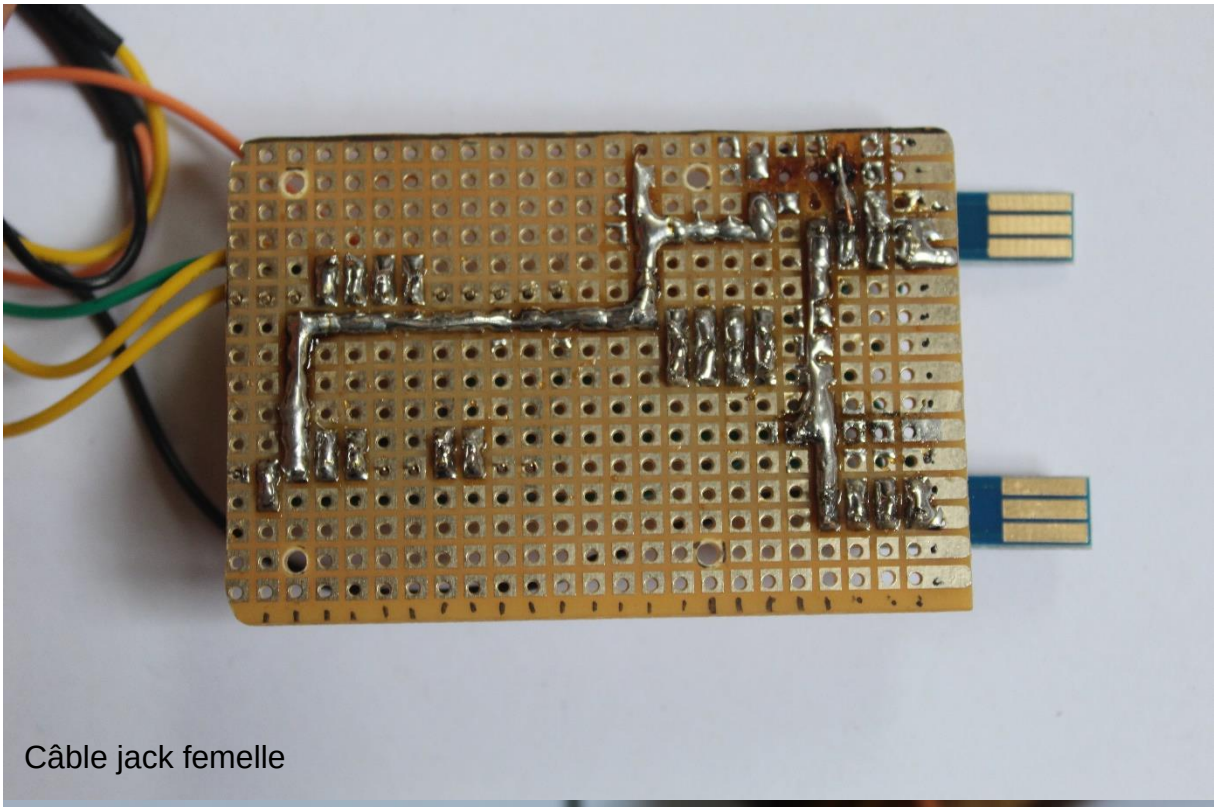
Veroboard et Teensy désassemblé



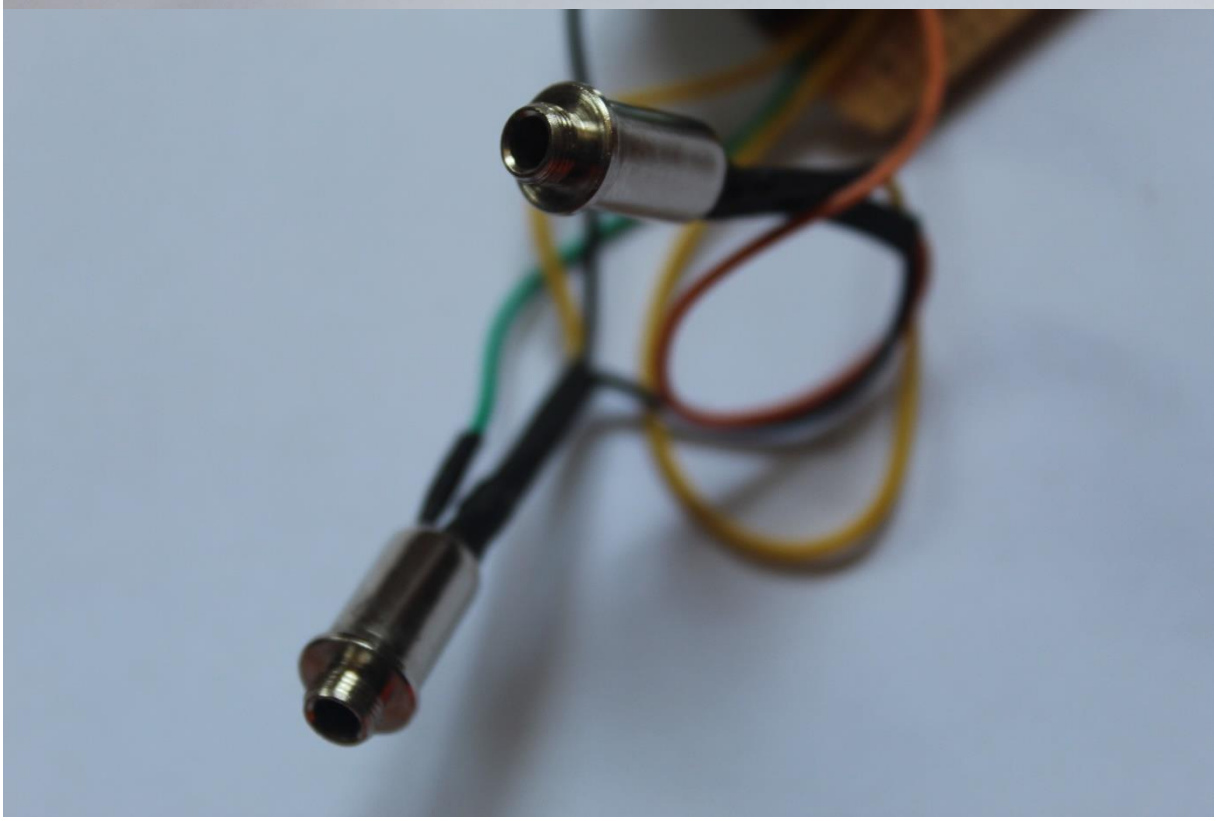
Veroboard et Teensy assemblés



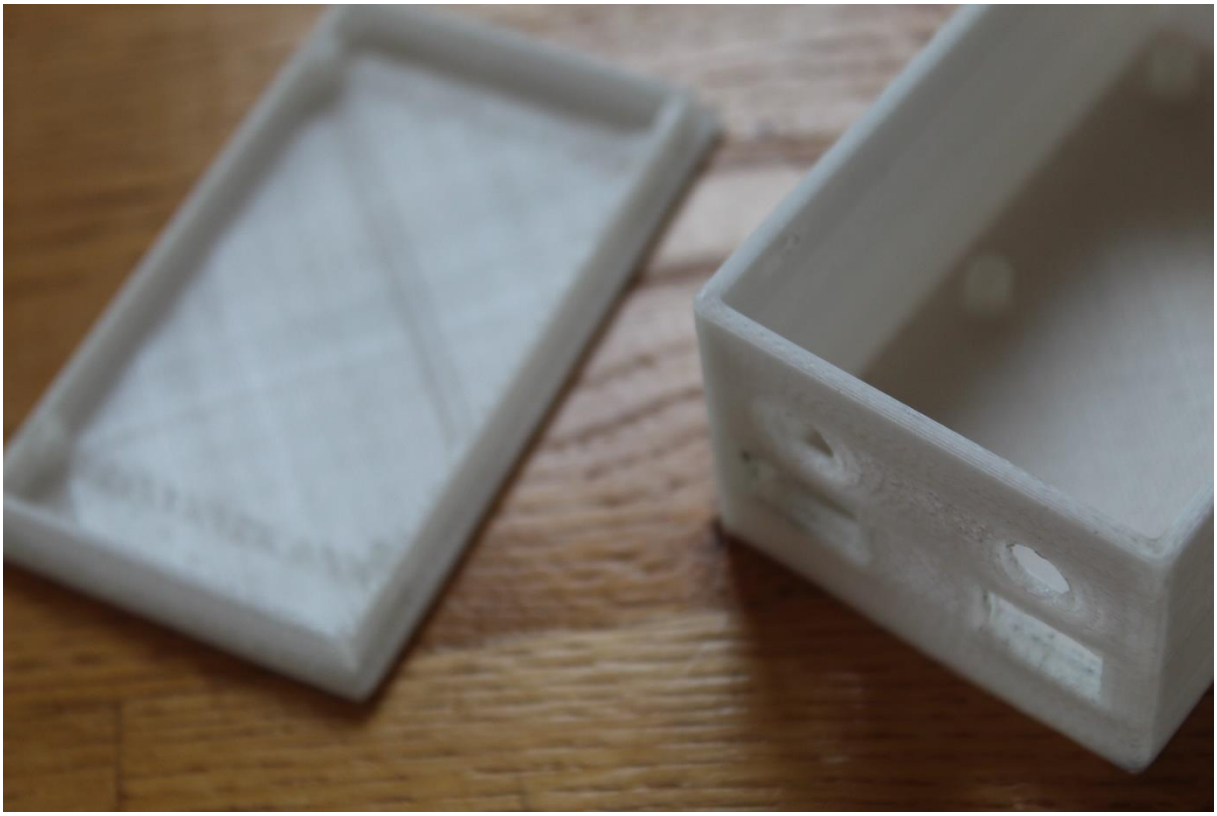
Dos de la veroboard



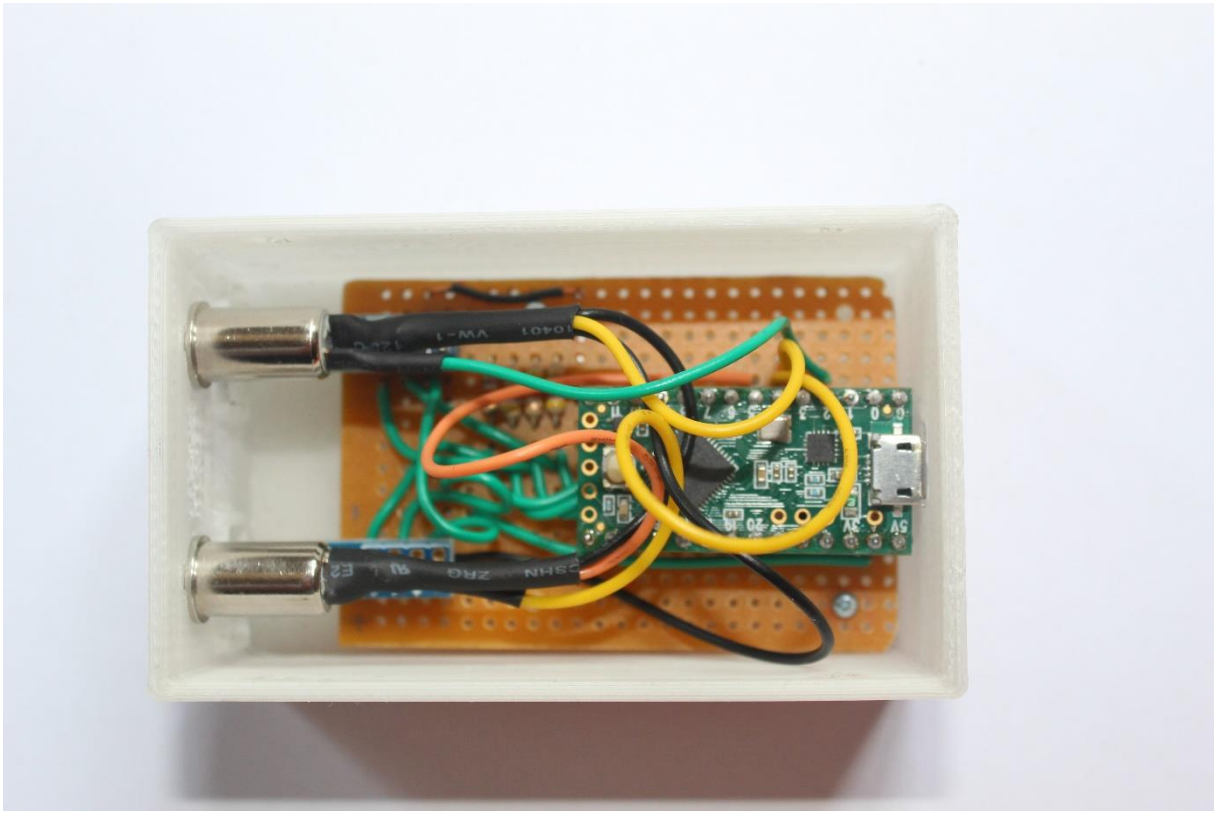
Câble jack femelle



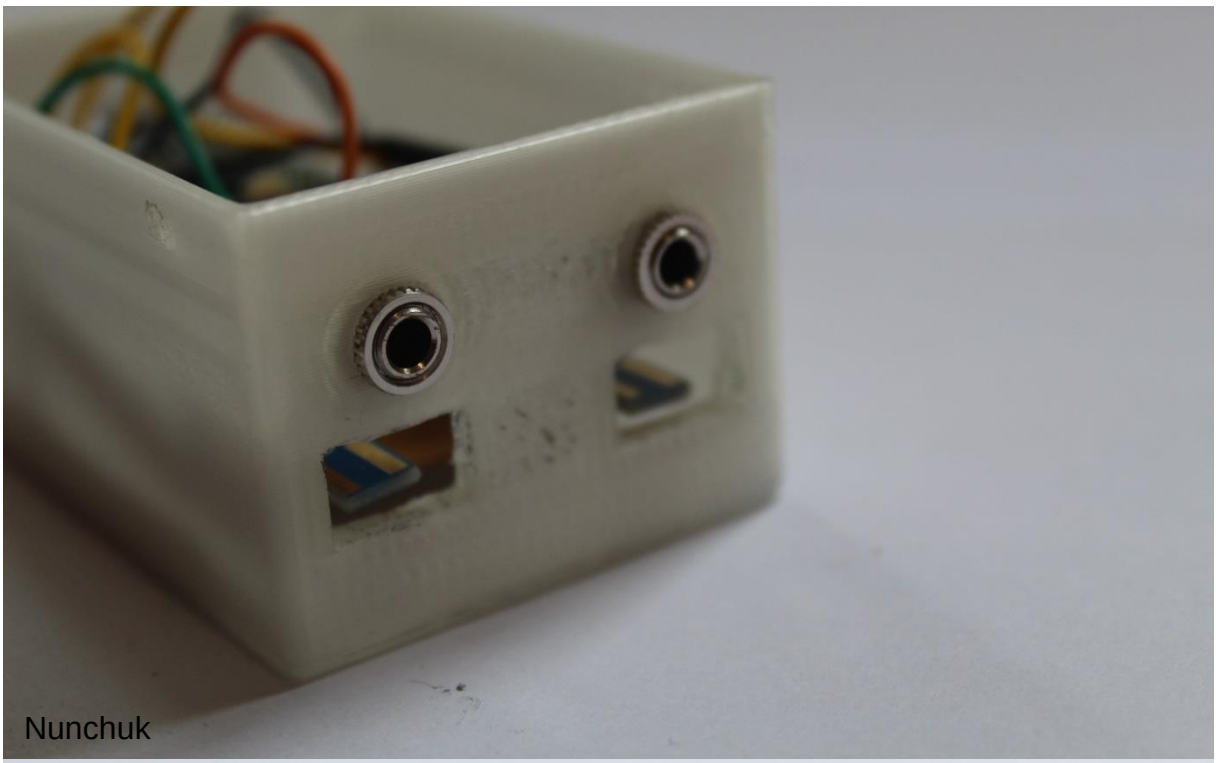
Boitier imprimé en 3D



Boitier, Veroboard et Teensy assemblés



Boitier, Veroboard et Teensy assemblés



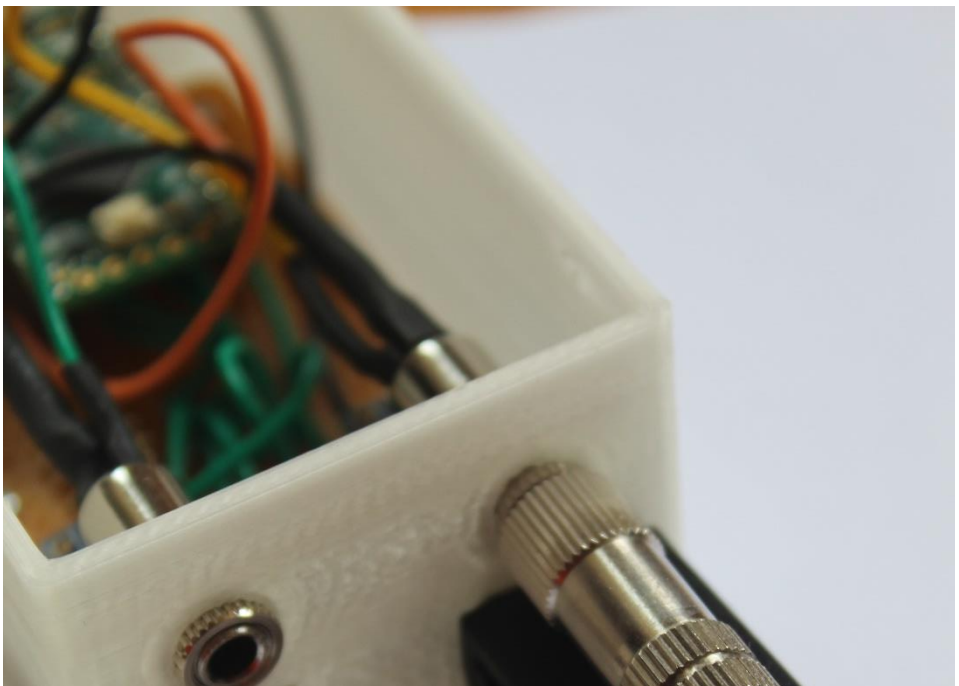
Nunchuk

Cable Nunchuk

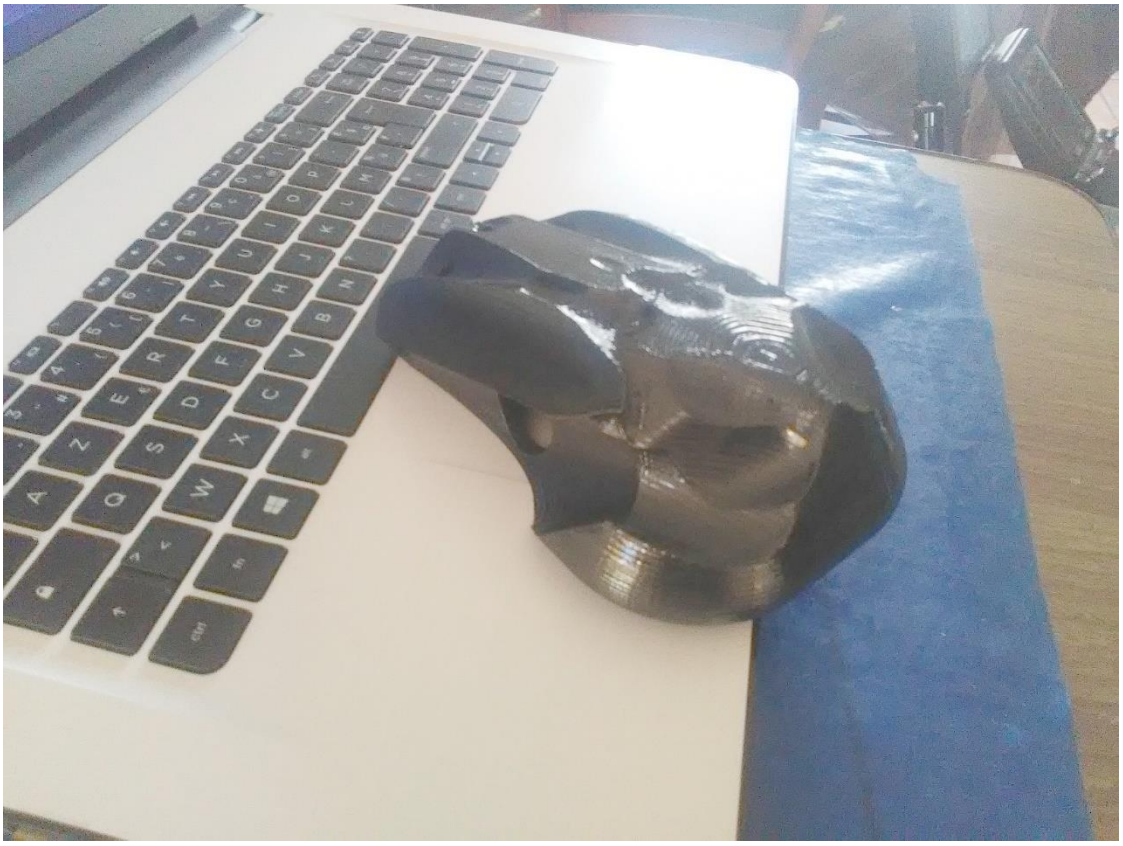
Intérieur commande secondaire

Commande secondaire

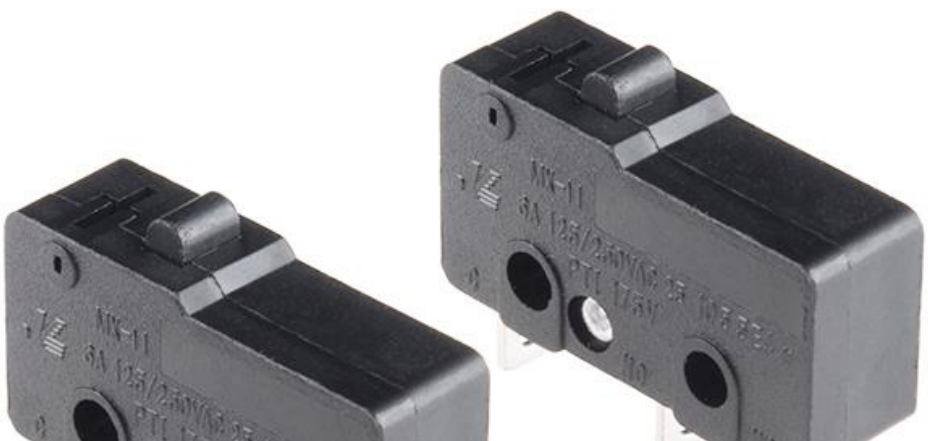
Boitier, Nunchuk et commande secondaire branchés



Le premier prototype de la manette (main imprimer en 3D)



Exemple de microswitch « sensibles » utiliser dans les boutons secondaires



FIN

Merci de votre lecture

Emile L.