

BINOREILLE

Présentation du projet: Développer une aide auditive à bas coût pour malentendant d'une oreille qui puisse être reproductible partout dans le monde.

Principe du prototype: Amplifier le signal d'un micro posé sur l'oreille malentendante pour le transmettre à l'oreille entendante par un casque avec un traitement du signal entre les deux.

Prototype Binoreille v1 réalisé aux Abilympics

Matériel :

- Basé sur une teensy 3.2 et shield audio
- Un micro
- Un écouteur

Code:

- Traitement du signal (filtre biquad)
- Contrôle par potentiomètre des paramètres suivants : Volume, freq, width

Interface hardware de réglage 3 potentiomètres + micro + casque courant.

Source code Arduino: Voir en Annexe

Avantage du proto basé sur Teensy : petit, modulaire et permettant une amélioration rapide du proto avec ajouts de nouvelles fonctionnalités par de simples lignes de code (traitement digital du son) et/ou modules. Accessible aux débutants.

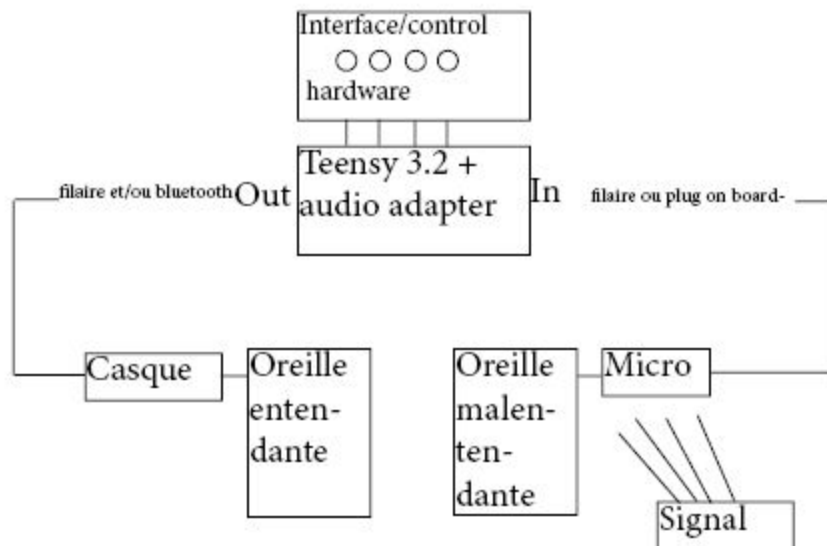
Prototype Binoreille v2 projet pour Fabrikarium

Objectifs généraux:

- Objet qui peut être visible/esthétique/assumé
- Basse consommation énergie
- Mobile
- Réglable facilement avec une interface simple proposant des pré-réglages (presets) pour différentes situations.
- Interface de réglage ajustable en temps réel en hardware et /ou en version android
- Objet fiable et robuste (voir imperméable)
- Confort d'écoute
- Permettre à l'utilisateur de faire lui-même ses réglages, ou de choisir des pré-réglages et/ou de sauvegarder ses propres réglages.

- Open source
- **Objectifs techniques** : Fabriquer le prototype basé sur [Teensy 3.2+audio shield](#) en deux versions en s'appuyant sur la v1.
- **Binoreille v2.1** : Une version avec interface de réglage hardware (potentiomètres directement fixée sur la boîte où se trouve la Teensy audio shield) pour que le dispositif soit totalement autonome avec une alimentation mobile (piles ou chargeur de batterie mobile)
- **Binoreille v2.2** : Une version “high tech” avec casque bluetooth (non prioritaire) et interface réglable sur Android.
- **Ressources techniques (datasheet+code)** :Audio chip datasheet : <https://www.pjrc.com/teensy/SGTL5000.pdf>

DIAGRAM BINOREILLE V2.1

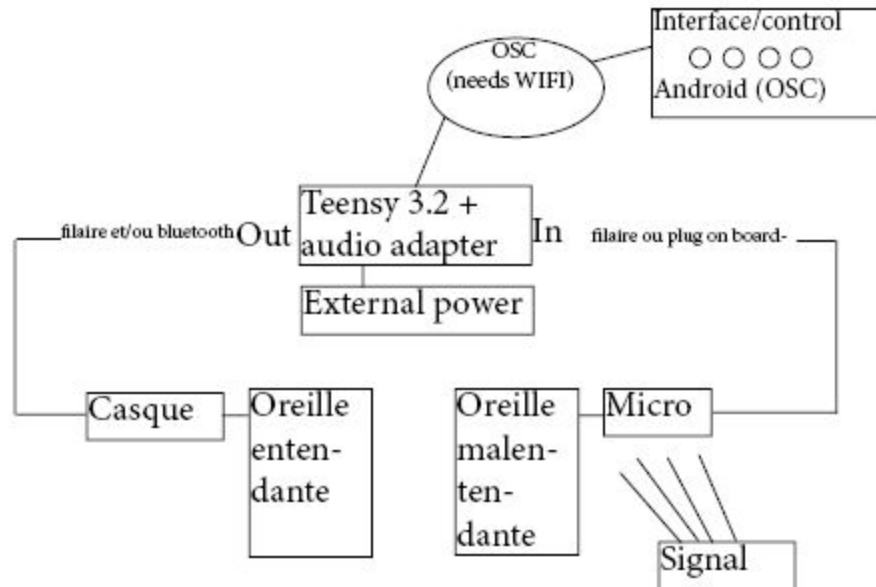


Contraintes plug on board:

Le micro est fixé dans le boîtier sur l'audio adapter: trouver un design adapté au port de cette boîte.

Penser à la taille de l'alimentation et la protection de l'humidité

DIAGRAM BINOREILLE V2.2



Contraintes plug on board:

Le micro est fixé dans le boîtier sur l'audio adapter: trouver un design adapté au port de cette boîte.

Penser à la taille de l'alimentation et la protection de l'humidité

OBJECTIFS TECHNIQUES

Interface Hardware

Version avec interface de réglage hardware
(potentiomètres directement fixée sur la boîte où se trouve la Teensy audio shield) pour que le dispositif soit totalement autonome

Interface Android

Version "high tech" avec interface réglable sur Android (et casque bluetooth non prioritaire)

PHASE 1: Développement commun aux deux prototypes

FABRICATION:

- Souder chaque [teensy 3.2](#) avec audio shield (avec pin header) en pensant que sur l'un des proto il faudra y ajouter la carte WIFI et la carte Bluetooth, et sur l'autre laisser des entrées accessibles pour les potentiomètres

Exemples pour monter plusieurs boards: https://www.pjrc.com/store/wiz820_sd_adaptor.html

CODE/TRAITEMENT DU SIGNAL:

Check : [7-Band Parametric EQ / 5-Band Graphic EQ / ToneControl](#) (page 28)+ check v1 code Arduino

- Limiteur,
- Compresseur
- Gate
- Équalisation
- Presets
- Noise cancellation (inverseur de phase?)
- Volume
- Power on/off
- Reset (pour recharger le prog à l'état initial)

Tutorial audio+teensy:

https://www.pjrc.com/store/audio_tutorial_kit.html

https://www.pjrc.com/store/teensy3_audio.html

DESIGN:

- Package du micro (embout de pince croco, stylo, barrette, etc)
- Design du casque
- Présentation finale

COMMUNICATION:

- Restitution percutante en anglais d'une durée de 3mn: Objectifs, Difficultés, Résultat, Retour expérience utilisateur.

TESTS ET RETOUR UTILISATEUR:

- Différents micros (electrets..)
- Différents casque audios (conduction osseuse, standard, bluetooth)
- tester les différents micros dans plusieurs situations (calme, bruyante, détection des voix de près, de loin)
- Tester différents emplacements du micro

- Casque à conduction osseuse et casque normal :comparer les designs, confort auditifs, esthétique
- possibilité de tester un casque à conduction osseuse DIY: le piezzo+amp+preamp

DOCUMENTATION:

PHASE 2: Développement distincts pour chacun des prototypes

Interface Hardware	Interface Android
FABRICATION	FABRICATION
<p>Electronique: Connecter les potentiomètres à la Teensy pour contrôler l'équalisation, le volume, et un switch pour le power on/off.</p> <p>Prévoir commutateur pour les préréglages pour situations données (conversation, musique, milieu bruyant, etc) qui seront fixes</p>	<p>Assembler les modules sur la carte et audio shield avec pin headers</p> <ul style="list-style-type: none"> - WIFI - Bluetooth
CODE	CODE
<p>Equalisation 5 ou 7 bandes ou plus simple (biquad)</p> <p>Volume</p> <p>Presets: Prévoir des préréglages fixes et la possibilité d'en ajouter des perso à l'utilisateur</p> <p>Compresseur/Limiteur/Gate : déclencher le son à partir d'un certain seuil et limiter le gain quand il dépasse un certain seuil</p> <p>Préréglages sélectionnable avec commutateur</p>	<p>Equalisation 5 ou 7 bandes</p> <p>Volume</p> <p>Presets: Prévoir des préréglages fixes (et la possibilité d'en ajouter des perso par l'utilisateur si possible, non prioritaire)</p> <p>Compresseur/Limiteur/Gate : Déclencher le son à partir d'un certain seuil et limiter le gain quand il dépasse un certain seuil</p> <p>Wifi: Ajout d'une couche WIFI pour communiquer en OSC avec l'interface sur Android avec le module ESP8266 http://snootlab.com/seedstudio/903-module-wifi-esp8266-antenne-fr.html https://forum.pjrc.com/threads/27850-A-Guide-To-Using-6-With-TEENSY-3</p> <p>OSC: Coder l'envoi et la réception OSC pour communiquer avec l'interface Android les paramètres</p>

	<p>suivants:</p> <ul style="list-style-type: none"> - Volume - Equalisation 7 bandes - Presets <p>Programmes : ControlOSC ou TouchOSC http://hexler.net/software/touchosc (5 euros) http://charlie-roberts.com/Control/ (gratuit)</p> <p>Tutorial OSC+arduino/teensy: http://cnmat.berkeley.edu/oscuino https://github.com/CNMAT/OSC http://www.deadpixel.ca/arduino-osc/</p> <p>Bluetooth pour casque (Non prioritaire): Implémentation d'une couche bluetooth pour envoyer l'audio au casque bluetooth ou en filaire (choix ajustable sur interface OSC) - Prévoir un bouton (push button?) pour activer ou non le bluetooth</p> <p>Modules en stock : https://hackspark.fr/fr/hc-05-bluetooth-v2-0-serial-transceiver-module-3-3v.html https://hackspark.fr/en/spk-b-bluetooth-audio-module.html https://hackspark.fr/en/serial-port-ble-module-master-slave.html</p>
<p>DESIGN</p> <p>3D Boîte : contenant les potentiomètres et le dispositif électronique (nombre de potentiomètre à définir pendant le brainstorming): volume, égalisation, pré-réglages avec un avec un potentiomètre à cran ou 2 push button + et - pour naviguer dans la banque de pré-réglages. Prévoir accès au reset de la carte teensy</p>	<p>DESIGN</p> <p>3D : Boîte: contenant toutes les cartes électroniques. Prévoir accès au reset de la carte teensy. Mobile, esthétique, léger, robuste.</p> <p>App Design OSC: Designer l'interface avec OSCControl ou TouchOSC pour avoir :</p> <ul style="list-style-type: none"> - L'équalisation 7 bandes ajustable (7 sliders) - Presets (nb à déterminer) - Volume - S'appuyer sur les remarques de l'utilisateur pour le design - Autres? (cf:brainstorming, conseil accousticien,

	<p>utilisateur,)</p> <p>Programmes : ControlOSC ou TouchOSC http://hexler.net/software/touchosc (5 euros) http://charlie-roberts.com/Control/ (gratuit)</p>
COMMUNICATION	COMMUNICATION
<p>Low Tech:</p> <ul style="list-style-type: none"> - Facilement reproductible dans les pays en voie de développement - Autonomie du prototype (pas besoin d'android) - Prix abordable 	<p>High tech :</p> <ul style="list-style-type: none"> - version permettant un réglage sur Android - Interface "user friendly": simple et facilement accessible - Prix abordable
TESTS UTILISATEUR	TESTS UTILISATEUR
<p>Tests :</p> <ul style="list-style-type: none"> - Tester le prototype - tester les différents préreglages <p>Retour d'expérience:</p> <ul style="list-style-type: none"> - Son, ergonomie, design, esthétique 	<p>Tests :</p> <ul style="list-style-type: none"> - Tester le prototype - tester les différents préreglages <p>Retour d'expérience:</p> <ul style="list-style-type: none"> - Son, ergonomie, design, esthétique
DOCUMENTATION	DOCUMENTATION

MATERIEL:

Adaptateur série/usb

Micro à tester pour développement futur (non disponible en stock @Fabrikarium) :

<https://www.pjrc.com/store/microphone.html>