

bibliothèque Mozzi Arduino pour la synthèse sonore

le point de vue de Tim Barrass



David Cuartielles (Arduino)

La bibliothèque de sons Mozzi permet de générer de la musique algorithmique sur un Arduino, sans shield supplémentaire ni synthétiseur externe. Vous êtes curieux de savoir comment tout cela a commencé ? Le développeur, Tim Barrass, nous parle du projet, de la communauté Mozzi et de son travail d'artiste.

▲
Tim Barrass a développé la bibliothèque sonore Mozzi pour les cartes Arduino 8 bits.

Tim Barrass est un artiste indépendant qui vit dans la campagne australienne. Il a développé la bibliothèque sonore Mozzi pour les cartes Arduino 8 bits (sensorium.github.io/Mozzi). En 2012, Tim et son frère Stephen ont développé un projet baptisé SweatSonics. Les problèmes qu'il a constatés dans les bibliothèques de synthèse sonore existantes l'ont amené à réfléchir à une manière plus performante de produire du son à partir des premières architectures de processeurs 8 bits incorporées sur des cartes comme Arduino Uno, Nano ou Mega. Le mode de fonctionnement de la bibliothèque a fait l'objet d'un article, un an plus tard, dans le carnet d'idées de l'ICMC. Il s'agit d'un tampon sonore de taille relativement modeste, appelé au moyen d'interruptions logicielles pour produire un son PCM mono sur l'une des broches de la carte. Le dispositif expose un certain nombre de méthodes que vous devrez contourner si vous souhaitez créer votre propre instrument de musique ou installation sonore. L'une des méthodes est dédiée à la lecture des commandes, l'entrée du système, tandis que l'autre est destinée à déterminer comment ces entrées vont affecter le rendu de la sortie audio. J'ai suivi le développement de la bibliothèque Mozzi au fil des ans car j'ai toujours été très intéressé par

les possibilités des microcontrôleurs pour la production de sons. Tim a passé plus de 30 ans à se produire en tant que musicien et à réaliser des installations sonores. Il a créé Mozzi pour faciliter son travail dans le domaine de l'art sonore. Mozzi est probablement l'outil le plus ancien pour créer facilement des oscillateurs, des boîtes à rythmes, des œuvres d'art sonores interactives et des instruments de musique novateurs. S'il a d'abord fonctionné sur des cartes Arduino 8 bits, l'outil a récemment été porté sur d'autres architectures. Il est toujours agréable de voir comment un code peut avoir une vie propre et être diffusé sur différentes machines. Dans cette interview, vous apprendrez à connaître Tim, son travail, le projet Mozzi et la communauté qui l'entoure.

David Cuartielles : Après avoir découvert l'origine de la bibliothèque Mozzi, je me suis mis à penser que vous deviez avoir une formation musicale et technologique. Suis-je loin de la vérité ? Pouvez-vous vous présenter brièvement ?

Tim Barrass : Oui, je suis un artiste formé dans le domaine de la technologie. Je vis à la campagne depuis quelques années maintenant, où je passe l'essentiel de mon temps à réparer des clôtures et cultiver des

plantes. Je ne fais plus autant d'électronique, bien que je fasse quelques expositions d'art de temps en temps. Je joue dans un groupe, mais nous ne nous produisons pas très souvent. J'ai travaillé pendant les 30 dernières années avec un cirque, un théâtre physique. Je jouais de la musique pour eux. Cela m'a servi pour Mozzi, parce que les possibilités sonores des récepteurs audio/vidéo (AVR) sont comparables au type de son que je faisais pour le cirque.

Cuartielles : Avez-vous une formation officielle dans ce domaine ?

Barrass : J'ai étudié le design industriel quand j'étais jeune, et pendant une année, l'animation par ordinateur et la technologie musicale. J'ai fait des recherches sur les processus parallèles massifs, par exemple sur la façon de programmer des populations de fourmis, ce qui était ma recherche de maîtrise.

Cuartielles : J'ai lu cet article que vous avez publié - je suppose - avec votre frère et dans lequel vous présentez Mozzi pour la première fois. Vous y mentionnez que ce travail est issu d'un projet appelé Sweatsonics. Pouvez-vous nous expliquer en quoi consiste ce projet ?

Barrass : C'est un sujet sur lequel mon frère faisait des recherches. Il s'intéressait à la façon dont les gens utilisaient le son dans les activités physiques. Ces travaux ont été réalisés en 2009 avec des iPod et des accéléromètres. C'était une façon pour Steve de contextualiser son travail. Rien de sérieux n'en est sorti, ni une entreprise, ni rien de tel.

Cuartielles : Pourquoi Mozzi ? Aviez-vous besoin d'une bibliothèque pour faciliter votre propre développement, essayiez-vous de simplifier la création de sons à l'aide de microcontrôleurs ?

Barrass : Lors de la conférence NIME (*New Interfaces for Musical Expression*) en 2010, nous avons réalisé quelques petits objets grâce à l'environnement Arduino. Nous avons expérimenté l'orientation et le son. Nous avons aussi préparé quelques petits objets qui ont nécessité des corrections tard dans la nuit. Il était environ 5 heures du matin et nous étions encore en train de programmer. Nous avons testé un code d'Adrian Freed, qui a réalisé l'une des premières générations d'ondes sonores avec un Arduino. Suite à cette expérience, j'étais déçu et j'ai senti que je devais faire quelque chose. Mon frère avait tenté de créer sa propre bibliothèque, mais il avait perdu le code. Je suis parti de zéro, car c'était la première chose que j'avais faite avec Arduino, et je dois dire que c'était très amusant. J'ai expérimenté cet univers pendant quelques années, en maintenant la bibliothèque, avec l'espoir de pouvoir faire des trucs géniaux. Je ne l'ai jamais fait pour faire des trucs, mais pour créer la bibliothèque.

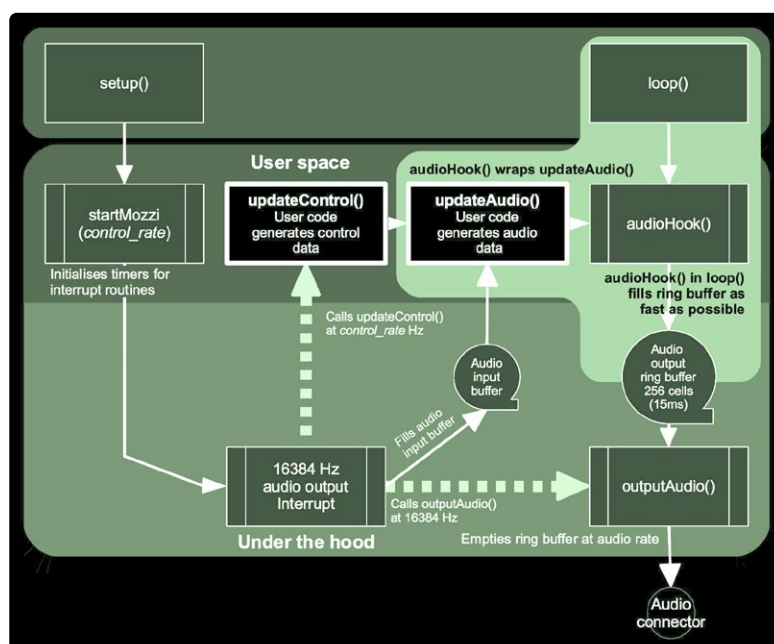
Cuartielles : Cela donne une meilleure perspective de l'histoire. En lisant l'article, j'avais une idée différente de la façon dont vous avez créé Mozzi. C'est génial d'avoir la possibilité de dialoguer pour comprendre comment les choses ont été faites concrètement et pourquoi. On dirait que vous l'avez fabriqué pour vous-même, mais que beaucoup d'autres personnes l'ont utilisé, non ?

Barrass : Depuis le début, je voulais partager la bibliothèque. Quel est l'intérêt d'écrire tout ce code et de le garder pour soi ? Cependant, publier une bibliothèque représente un énorme travail. Certains me disaient que j'allais devoir la maintenir éternellement et que je devais simplement faire mon truc et passer à autre chose. C'est ce que j'avais en tête, et au bout d'un moment, c'est ce qui s'est passé. Le projet a demandé beaucoup de travail, et j'en ai eu assez après quelques années. J'ai appris le C et le C++ en créant Mozzi. Lorsque je l'ai rendu public, j'étais stressé en pensant que ce n'était pas assez bon. Et au bout d'un moment, j'ai senti que je ne pouvais plus travailler dessus.

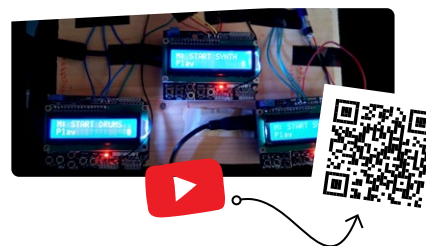
Cuartielles : J'ai vu Mozzi un peu partout, dans des universités et des salons. Cette bibliothèque revient tout le temps. Si quelqu'un veut faire un projet lié au son, il y a un moment où il va nécessairement télécharger et essayer Mozzi.

Barrass : Je suis moi-même surpris car le framework est une chose très simple. Au cours des deux dernières années, de très bons programmeurs l'ont porté sur d'autres plateformes. Ils sont probablement beaucoup plus expérimentés que moi ; c'est assez intimidant.

L'interface entre la bibliothèque Mozzi et l'environnement Arduino se compose de quatre fonctions principales. (Source : <https://sensorium.github.io/Mozzi>)



Vidéo : OscPocketo, an Arduino pocket synth and drum machine.



```
#include <MozziGuts.h>
#include <Oscil.h>
#include <tables/sin2048_int8.h>

#define CONTROL_RATE 128
Oscil <2048, AUDIO_RATE> aSin(SIN2048_DATA);

void setup(){
    aSin.setFreq(440);
    startMozzi(CONTROL_RATE);
}

void updateControl(){
}

int updateAudio(){
    return aSin.next();
}

void loop(){
    audioHook();
}
```

Un croquis Mozzi minimal. (Source : <https://sensorium.github.io/Mozzi>)

Je suis surpris qu'ils ne l'aient pas écrit de manière beaucoup plus claire en partant de zéro.

Cuartielles : Je pense que ce qui fonctionne avec Mozzi, c'est le paradigme que vous proposez. Pour que nos lecteurs comprennent mieux, vous exposez deux méthodes à remplacer, baptisées « updateControl » et « updateAudio ». L'une traite les interfaces physiques, l'autre le rendu sonore. Et c'est là que réside plus ou moins la magie. Le dispositif produit des signaux sonores mono. Pour de nombreuses personnes - programmeurs ou instrumentistes - qui ne sont pas nécessairement des luthiers et ne comprennent pas comment produire une sortie sonore, il y a là de grandes possibilités de création. Je peux reconnaître la beauté qu'apporte la simplicité de cette architecture. Il y a quelques points forts sur lesquels j'aimerais vous interroger.

Barras : Une routine d'interruption lente se produit 64 fois par seconde et s'interface avec les capteurs ou les signaux MIDI pour vous permettre de définir des variables. Il y a également une interruption du débit audio où vous pouvez utiliser ces variables pour affecter les sons au cours de la phase de synthèse. Dans le contexte du framework, je m'attendais à ce que les utilisateurs l'étendent par le biais de classes. Il y a un certain nombre de classes de synthèse, mais pas de classes de contrôle. Je l'ai fait pour que d'autres puissent l'étendre avec leur propre code. Dans un certain sens, c'est un peu comme le Forth, le premier langage de programmation que j'ai appris ; il y a une

entrée et une sortie, et c'est tout.

Les modules de synthèse sont faits pour que les utilisateurs puissent créer les leurs. Je n'ai proposé que quelques exemples sur la manière de faire une synthèse. J'avais espéré qu'il y ait des développements à ce sujet, mais ce n'est pas vraiment ce qui s'est produit. Il y a la place pour aller plus loin et produire des sons nouveaux. Stenduno a été parmi les premiers à travailler avec le dispositif. Ils ont fabriqué des instruments de musique assez tôt - 2013 ou 2014. Ce qui les intéressait, c'était la création de leurs propres sons, et pas seulement la modification des paramètres des sons existants. J'ai aimé ce qu'ils ont fait avec le design des circuits imprimés, etc. Leurs œuvres sont maintenant des objets de collection et elles sont même devenues assez chères.

Cuartielles : Si j'ai bien compris, le projet d'origine était très limité en raison de la conception des cartes. Vous poussiez le son à 16 kHz, sur des cartes fonctionnant à 16 MHz. Mais savez-vous jusqu'où les utilisateurs ont poussé les performances ?

Barras : Certains ont produit des cartes qui peuvent fonctionner à des fréquences plus élevées. Je n'ai pas été en mesure de les essayer, car je n'ai pas le matériel nécessaire. Mais ce qui me surprend, c'est que des utilisateurs continuent à utiliser mon code, qui a été optimisé pour fonctionner dans ces contraintes originales, et écrit de manière non intuitive, avec de l'arithmétique à virgule fixe... ces nouveaux processeurs peuvent faire beaucoup mieux. L'un de ces gars m'a dit qu'il aimait le fait de pouvoir écrire le code pour une plateforme et continuer à travailler sur une autre. Pour moi, le défi, dans ce cas, est d'attendre de ces petites cartes qu'elles fassent des choses qui sont conçues pour des processeurs plus grands.

Cuartielles : À un moment donné, j'ai animé un atelier au cours duquel j'ai conçu une boîte à rythmes permettant d'enregistrer une seconde de votre voix dans la mémoire du microprocesseur ATmega328 (celui de l'Arduino Uno d'origine) et de l'utiliser comme ensemble de sons de base pour créer un looper ou une batterie. Parmi les participants de l'atelier, il y avait un couple d'artistes français qui avait porté libpd (moteur sonore du logiciel PureData) pour le faire fonctionner sur l'Arduino Uno, ce qui m'a époustoufflé. Je savais qu'au sein du MTG de Barcelone (le

master en production numérique de l'université Pompeu Fabra), PureData avait été porté pour qu'il fonctionne sur des machines 8 bits, mais je ne l'avais jamais vu fonctionner sur une carte Arduino aux ressources très limitées. Nous avons déjà parlé de ce qui vous a poussé à créer Mozzi, mais étiez-vous au courant des autres solutions existantes et celles-ci vous ont-elles influencé vos décisions d'une manière ou d'une autre concernant la création de Mozzi ?

Barrass : Je n'étais pas au courant du portage de libpd. À notre connaissance, il existait peu de choses utilisant le concept d'un temporisateur à interruptions et des tables d'ondes. Je n'ai pas vraiment cherché, mais je n'ai rien trouvé, y compris l'interruption de contrôle, qui aurait permis de faire quelque chose d'intéressant avec le son. Il n'y a eu que quelques tentatives qui n'étaient ni terminées ni performantes. Je ne pense pas que les exemples que j'ai trouvés avaient un tampon audio, et c'est ce qui a rendu Mozzi plus flexible au final, un petit volume d'audio en tampon qui vous permet de faire autre chose. Cela me rappelle un des problèmes de Mozzi : certains capteurs bloquaient le processeur. Quelqu'un a écrit un code non bloquant pour travailler avec les capteurs, et il est maintenant incorporé dans Mozzi. J'ai fait des essais incluant jusqu'à cinq capteurs différents. Certains d'entre eux sont difficiles à utiliser en raison du volume modeste du tampon sonore.

Cuartielles : Pour les lecteurs peu versés dans le son, je dois expliquer qu'un tampon sonore est une technique qui permet de préparer un certain nombre de millisecondes de données que vous allez émettre à l'aide d'une interruption temporisée. Cela entraîne une certaine latence, c'est-à-dire une différence de temps entre le moment où vous envoyez une commande et celui où le son est modifié. Dans le cas de Mozzi, elle est de 15 millisecondes. Quel est l'effet perçu de cette latence ?

Barrass : C'est comme un effet de « chorus » (chœur) ou de « delay flanger » (mélangeur retardateur). Il est très discret et s'entend surtout dans les percussions. Mais Mozzi n'a jamais vraiment été conçu pour créer des instruments de musique ; il était plutôt destiné à créer des installations. Je suis surpris de voir combien de personnes ont créé des synthés avec des boutons, des câbles et du MIDI, et des capteurs pas si fous que ça. Mon frère et moi avions une expérience de la sonorisation et nous pensions à des choses comme une soufflerie, dotée de nombreux dispositifs Arduino qui pourraient modifier la perception du son. Ce n'est pas la fabrication d'instruments qui nous intéressait moi et mon frère.

Cuartielles : Avez-vous des exemples de ces installations et de ces productions sonores ?

Barrass : Il y en a un certain nombre sur la page Galerie de la bibliothèque Mozzi : <https://sensorium.github.io/Mozzi/gallery/>.

Cuartielles : J'aimerais avant tout savoir ce que vous avez en tête. Cette interview porte sur ce qui inspire les gens à créer des choses pour aider les autres, dans l'esprit de la démarche open source. Mais j'ai aussi quelques questions plus amusantes. Commençons par celles-ci. D'où vient le terme *Mozzi* ? J'ai lu que le terme d'argot *mozzi* désigne le moustique en Australie et en Nouvelle-Zélande. Est-ce bien le cas ?

Barrass : Au départ, nous voulions l'appeler *Project Cuttlefish* (projet Seiche), car l'un des objets fabriqués était taillé dans un os de seiche. Et à la dernière minute, nous avons opté pour Mozzi, parce que cet animal émet de petits sons et que la bibliothèque ainsi baptisée allait vraiment être utilisée avec de petits haut-parleurs, sans amplification ou enregistrement. C'est ce qui convenait vraiment - un petit son. Je suis assez surpris de la façon dont les choses se sont passées. Si vous avez de la place (dans votre instrument de musique), pourquoi ne pas expérimenter quelque chose qui soit beaucoup plus facile à utiliser ?

Cuartielles : Il existe d'autres plateformes plus récentes que Mozzi et qui ont fini par disparaître, comme Axolotl. Il s'agit (ou s'agissait) d'une très belle carte dotée d'un logiciel qui vous permettait de patcher des blocs pour modifier le son. Malheureusement, sa production était très coûteuse et il ne semble pas que le projet ait été poursuivi. Il existe quelque chose de nouveau appelé Bale, qui possède un hub USB, et des entrées de capteurs...

Barrass : Il y a quelques semaines, j'en ai vu un appelé Daisy Seed.

Cuartielles : Oui, c'est exact. Depuis l'arrivée du Cube-X (une ancienne plateforme de capteurs programmable avec Max/MSP), il y en a eu bien d'autres. Cependant, dans certaines situations, vous avez simplement besoin d'un morceau de code à ajouter à votre projet existant, et c'est là qu'un dispositif comme Mozzi peut jouer un rôle important. Si les interruptions n'entrent pas en collision avec le reste du code, vous pouvez le faire fonctionner. Il y a une différence avec ces systèmes dédiés et autonomes.

Vidéo : Turbulence Wind Sound Installation.



Barrass : Je n'ai jamais pensé que Mozzi était facile à comprendre, et je n'en suis toujours pas sûr. J'ai eu l'impression de devoir beaucoup l'expliquer aux autres, pendant longtemps. Je suis très heureux si l'outil est facile à comprendre pour certaines personnes.

Cuartielles : En lien avec ce concept d'explication de la bibliothèque à d'autres personnes, j'allais également poser une question sur la diffusion. Votre page de documentation comporte 12 catégories d'exemples. Je les lis dans l'ordre, en ajoutant de la complexité au fur et à mesure. Aviez-vous une intention pédagogique lorsque vous les avez créés, ou vous contentiez-vous de décrire les capacités du système ?

Barrass : L'un et l'autre. J'ai probablement fait les exemples dans l'ordre où j'ai écrit la bibliothèque. Et, progressivement, je suis passé à des exemples plus intéressants.

Cuartielles : Il est très courant dans ce monde de la technologie et de l'art, qui est également lié au monde universitaire, de finir par enseigner au cours d'ateliers sur la technologie. Avez-vous, à un moment donné, organisé des ateliers utilisant Mozzi avec des musiciens et des artistes d'installation pour expliquer les possibilités de la bibliothèque ?

Barrass : [rires] Mon frère travaillait en tant qu'universitaire. C'était très bien pour lui de continuer ses voyages autour du monde et d'enseigner dans le cadre

de quelques ateliers. Mais je n'ai pas suivi cette voie, même si j'aurais aimé le faire.

Cuartielles : Qu'en est-il des collaborations et des contributions ? En accédant au référentiel Github, j'ai pu voir qu'il y a des pics pour la production du code, mais qu'en est-il des collaborations, de l'ajout d'autres noyaux, des demandes de retrait, de la gestion des problèmes, etc. ?

Barrass : Il y a probablement trois ans, Thomas Friedrichsmeier m'a contacté alors qu'il effectuait un portage sur sa plateforme. Jusque-là, nous communiquions par un rapport de bogue occasionnel que j'essayais de suivre. Au bout de quelques mois, j'ai demandé à Thomas s'il voulait être un collaborateur, puisque je n'avais ni son matériel ni le temps de m'y intéresser. Je n'avais pas envie de gérer sa base de code, et il a commencé à s'occuper de l'essentiel du développement, surtout du portage, plutôt que d'étendre le côté créatif. Il est génial.

Un autre développeur, Thomas Combriat, est arrivé plus récemment et a apporté de nombreuses contributions. L'autre Thomas semble être heureux de faire le gros du travail. Les choses décollent. J'apprécie que le projet soit devenu autonome par rapport à moi. Comme sur le forum où les utilisateurs s'entraident (Thomas Friedrichsmeier a réalisé le portage de Mozzi pour le STM32, les processeurs ESP et le RP2040. Thomas Combriat, quant à lui, s'est occupé de Teensy 4).

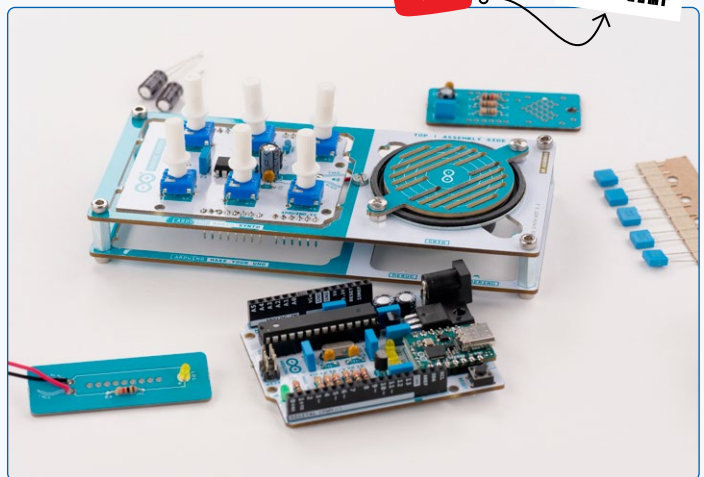
Le kit Arduino Uno DIY

Si vous voulez effectuer des expérimentations avec la bibliothèque Mozzi, vous pouvez utiliser quelque chose de tout petit comme l'Arduino Uno, et un simple élément piézo connecté à la broche numéro 9 de votre carte. Vous pouvez également construire une machine sonore dédiée avec le kit Arduino Uno DIY. Vous devrez assembler votre propre Arduino Uno et le shield de synthétiseur qui l'accompagne, qui contient six potentiomètres (cinq connectés aux entrées analogiques et un pour contrôler le volume), un amplificateur audio et un haut-parleur, petit mais puissant. En fonction de votre niveau en matière de soudage, il vous faudra entre 60 et 90 minutes pour tout assembler et commencer à jouer. Tous les composants du kit sont à trou traversant. Nous avons inclus un module USB-C pour effectuer des chargements vers le processeur principal, alimenter la carte et avoir une communication série en retour vers votre ordinateur.

Avec le kit Arduino Uno DIY, vous pouvez produire des airs simples en utilisant la bibliothèque de sons par défaut dans l'IDE, mais vous pouvez aussi créer des sons synthétiques complexes en chargeant la bibliothèque Mozzi sur votre carte. Démarrez avec le kit DIY Uno et Mozzi en trois étapes :


1. Assemblez le kit Arduino Uno DIY et le shield synth (~60 min).
2. Installez la bibliothèque Mozzi pour Arduino dans votre IDE (~5 min).
3. Essayez différents exemples.

Vidéo : Exemple de la bibliothèque Mozzi avec ce shield



Cuartielles : Ce qui nous amène à la question encore sans réponse : Quel est votre principal centre d'intérêt en ce moment ?

Barrass : J'ai emménagé ici (dans la campagne australienne) en pensant que j'allais faire pousser des légumes, mais les kangourous et les wallabies ne cessent de les manger. Et puis nous avons eu des feux de brousse, qui ont complètement ruiné nos projets. Nous sommes encore dans la phase de récupération. Depuis, mon travail créatif est surtout orienté vers mon groupe punk de protestation politique. Le groupe est installé sur une remorque et il joue pendant qu'elle se déplace. Il est vraiment dirigé par des acrobates de cirque avec lesquels j'ai joué de la musique pendant près de trois décennies - je les soutiens en jouant de la guitare pour les chansons qu'ils écrivent. Nous n'avons pas encore donné de véritables concerts. Je ne pense pas que nous irons loin ! Les chansons sont longues et ennuyeuses si vous ne connaissez pas les abominables hommes politiques dont elles parlent. Les wombats que l'on voit au milieu de cette vidéo - <https://youtu.be/E4qUPzUWxxA> - sont des animaux dont je me suis occupé comme soignant pour la faune

sauvage. L'une de mes principales activités consiste maintenant à aider une organisation de sauvetage de la faune à élever et à relâcher des animaux après les incendies dévastateurs qui ont eu lieu ici en 2019 et 2020. 

220438-04 — VF : Pascal Godart

À propos de l'auteur

David Cuartielles est co-fondateur d'Arduino. Titulaire d'un doctorat en design d'interaction et d'un master en génie des télécommunications, il enseigne à l'université de Malmö.

Des questions, des commentaires ?

Avez-vous des questions ou des commentaires relatifs à cet article ? N'hésitez pas à contacter Elektor à l'adresse redaction@elektor.fr.

LIENS

- [1] T. Barrass, « Greenwash », 2013 : <https://vimeo.com/69398645>
- [2] J. Deladrière, MozMo: Arduino Mozzi Synth in Eurorack Hardware, 2021: <https://bit.ly/github-Mozmo>
- [3] A. Freed, « Arduino Sketch for High Frequency Precision Sine Wave Tone Sound Synthesis », 2009: <https://bit.ly/freed-sketch>
- [4] Y. Nakanishi, Powderbox, The University of Tokyo: <https://yoshihito-nakanishi.com/projects/powderbox/>



Produits

Vous recherchez les principaux produits mentionnés dans cet article ? Arduino et Elektor s'occupent de vous !

- > **Arduino Uno Rev3**
www.elektormagazine.fr/arduino-uno
- > **Arduino Uno DIY + Synth Shield Kit**
www.elektormagazine.fr/arduino-synth-shield-kit



L'édition spéciale bonus d'Arduino

Chaque semaine, de nouveaux articles déclassifiés !

Obtenez-le ici GRATUITEMENT !

www.elektormagazine.fr/arduino-bonus

