elektor

LE SEUL MAGAZINE ÉCRIT PAR ET POUR LA COMMUNAUTÉ RPI

Le magazine officiel du RASPBERRY PI

MagPi

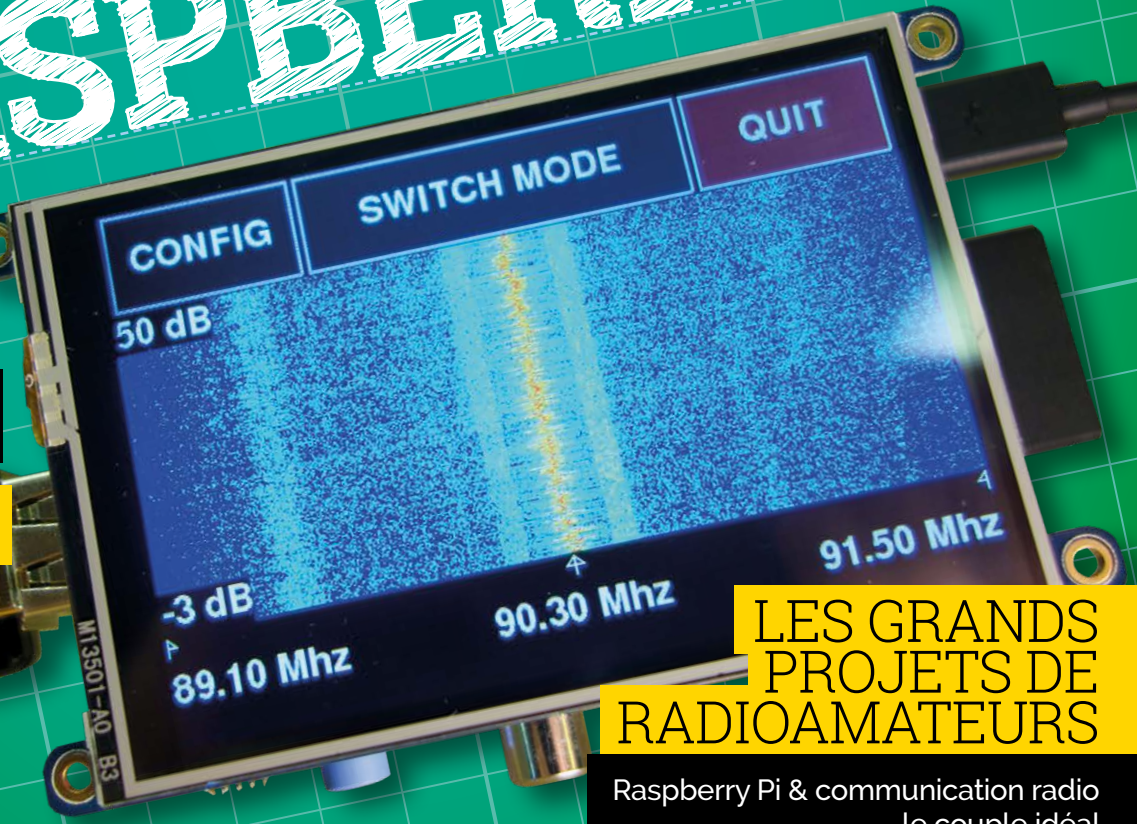
n°9
juillet - août 2019

www.magpi.fr

Le magazine officiel du **Raspberry Pi**

CONSTRUIRE UN PROJET RASPBERRY PI

- ✓ Découvrir les trucs des makers
- ✓ S'inspirer de projets accessibles aux débutants
- ✓ Réaliser son propre projet



CRÉER DES ROMANS GRAPHIQUES

Convertir une vidéo enregistrée avec un HAT TV en roman-photo

Entre autres :

- **Système de surveillance des plantes** avec OpenCV et un module Camera
- **Images de l'ISS par ondes radio** avec une clé RTL-SDR
- **Barista personnel** avec Snips et une machine à café retouchée

LES GRANDS PROJETS DE RADIOAMATEURS

Raspberry Pi & communication radio – le couple idéal

LUX 10,55 € - CH 15,90 FS - DOM/S 10,95 € - CAN 17 $ca

ÉLECTRONIQUE AVEC GPIO ZERO

L 15926 - 9 - F: 9,95 € - RD

productronica
fast forward
the startup platform
powered by Elektor

PARTICIPEZ AU CONCOURS POUR LANCER VOTRE STARTUP

SUR LE SALON **PRODUCTRONICA** 2019

Participez à l'édition 2019!
12 au 15 novembre 2019 à Munich

informations détaillées :
www.elektormagazine.fr/p-ffwd

Platinum Sponsor:
DISTRELEC
Distribution with a difference

Productronica Fast Forward is brought to you by
productronica
elektor
INNOVATION STARTUP TRADE

BIENVENUE
Le Raspberry Pi est partout !

Il y a quelques semaines, j'ai fait un tour sur une brocante organisée par une association locale de radioamateurs. Et là, le Raspberry Pi dans ses différentes variantes trônait fièrement au côté d'appareils d'émission et de réception coûteux et sophistiqués. Les radioamateurs ont adopté le RPi parce que son matériel et son logiciel sont adaptables à nombre de leurs besoins. Nous donnons donc un coup de projecteur sur le vaste monde des communications par ondes radio. Le célèbre François Mocq du site framboise314, lui-même radioamateur, nous donne un panorama de tout ce que le RPi peut apporter dans ce domaine. Nous avons également donné la parole au président du Réseau des Émetteurs Français. Après ces lectures, vous aurez peut-être envie d'obtenir votre « certificat d'opérateur des services d'amateur ».

Le second coup de projecteur est braqué sur des projets qui devraient vous aider à dépasser les limites du logiciel, plus précisément à connecter différents composants à votre RPi. Plusieurs makers vous donnent des conseils sur comment s'équiper, expérimenter… Vous aurez de quoi étoffer vos projets.

Un autre sujet revient de manière récurrente : la commande vocale. Le YogAI, un miroir magique avec yogi virtuel, ainsi que la machine à café qui obéit à la voix reposent tous les deux sur la reconnaissance vocale conçue par Snips, déjà présentée dans de précédents numéros. Vous serez peut-être tenté de vous y mettre avec le kit de Snips (voir le banc d'essai à la page 88).

Bien entendu cette présentation des articles n'est pas exhaustive. Vous serez sûrement surpris, voire inspiré par la guitare intelligente ou le robot de Sean. Les tutoriels vous aideront à concrétiser vos idées.

Mariline Thiebaut-Brodier rédactrice en chef

PS Nous saluons l'arrivée du dernier né de la famille : Raspberry Pi 4 !

RÉDACTRICE EN CHEF

Mariline Thiebaut-Brodier

Rédactrice en chef des versions françaises des magazines Elektor et MagPi, elle est tombée il y a bien longtemps dans la marmite de l'électronique et de l'informatique. Toutefois quand elle en sort, elle expérimente toutes sortes de loisirs créatifs comme la broderie à la machine électronique.

magpi.fr

MagPi

juillet - août
2019 - n°9
ISSN : 2608-1822
Dépôt légal : juillet 2019

Raspberry Pi APPROVED RESELLER

CC BY NC SA

Directeur de la publication : Donatus Akkermans

MagPi est édité par : PUBLITRONIC SARL
c/o Regus Roissy CDG
1 rue de la Haye
BP 12910
FR-95731 Roissy CDG cedex
www.magpi.fr

Éditrice : Mariline Thiebaut-Brodier
redaction@magpi.fr
Abonnement : www.elektor.fr/magpi-abo

Rédacteurs : Lucy Hattersley, Rob Zwetsloot, Phil King, Jem Roberts

Auteurs : Alex Bate, Brian Beuken, Mike Cook, David Crookes, Gordon Horsington, Kent Elchuk, Kyle Goff, Phi King, Sean McManus, Binsen Qian, Matt Richardson, Laura Sachs, Richard Smedley, Dean Swift, Clive Webster, Sai Yamanoor, Srihari Yamanoor

MagPiFR

MagPi_FR

Traducteurs : Thierry Destinobles, Pascal Duchesnes, Pascal Godart, Jean-Louis Mehren, Hervé Moreau, Kévin Petit, Denis Lafourcade
Maquette : Dougal Matthews , Julian van den Berg
Impression : Pijper Media, Amsterdam (Pays-Bas)
Diffusion : MLP (France), AMP (Belgique)

Service aux lecteurs : contact@magpi.fr
Publicité : Margriet Debeij
margriet.debeij@elektor.com

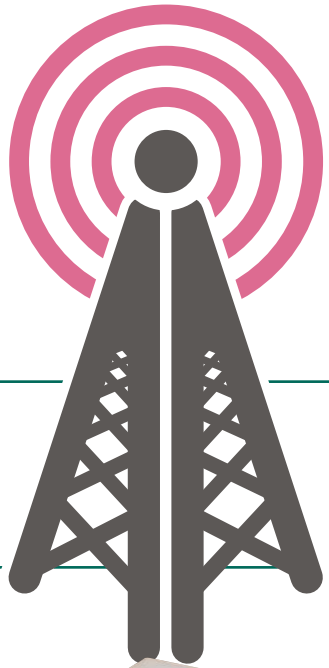
POWERED BY elektor

MagPi – le magazine officiel du Raspberry Pi (version française) est une version sous licence du magazine MagPi publié par Raspberry Pi (Trading) Ltd, 30 Station Road, Cambridge, CB12JH (Royaume Uni). Tous les contenus de ce magazine sont soumis à la licence Creative Commons *Attribution-Non Commercial-ShareAlike 3.0 Unported* (CC BY-NC-SA 3.0).

Raspberry Pi PRESS

Sommaire

> nº 9 > juillet - août 2019



Coup de projecteur

14 Projets radioamateurs étonnants

11 Le Raspberry Pi et les radioamateurs par François Mocq

Le monde du RPi

06 Découvrez Raspberry Pi 4

08 Lancement de la bibliothèque GPIO Zero 1.5

09 Nouveaux dongles AutoPi

10 Un calendrier intelligent grâce au Raspberry Pi

Projets

22 YogAI - miroir magique avec yogi virtuel

24 Clapper Raspberry Pi - commande par claquement de mains

26 Votre barista personnel avec Snips

28 Le robot de Sean

32 Moniteur Commodore

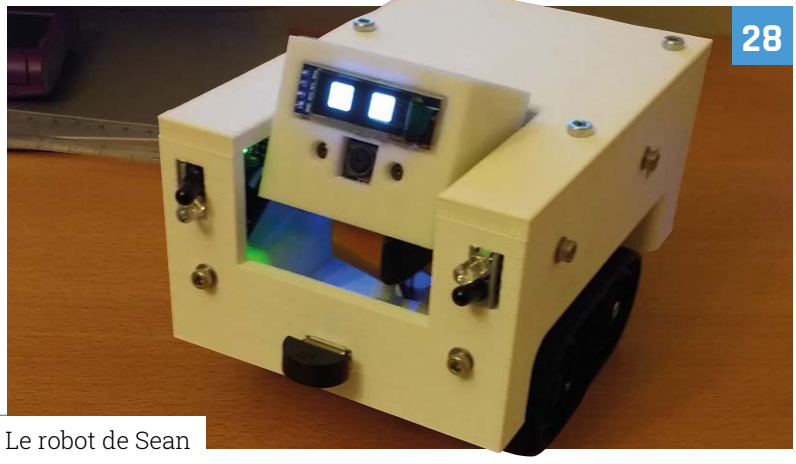
34 Chord Assist - guitare intelligente et accessible

Communauté

90 Jean-Louis TRUQUET, F5DJL - Réseau des Émetteurs Français

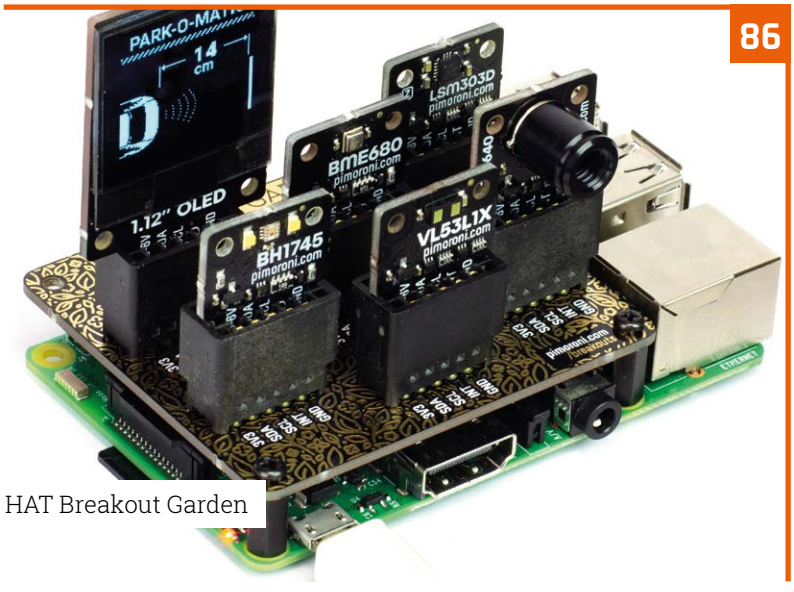
92 Nikole Vaughn - organisatrice de CoderDojo

12



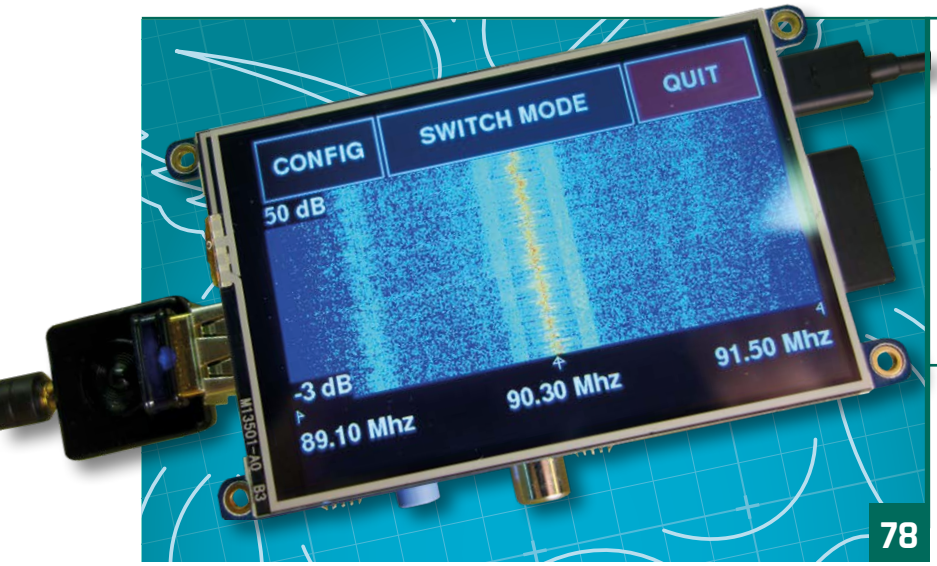
28

Le robot de Sean



86

HAT Breakout Garden



Coup de projecteur

78 Dépasser les limites du logiciel : réaliser un projet matériel avec un Raspberry Pi

Tutoriels

36 Bâtir un serveur de fichiers Samba
40 3e partie - coder un jeu en vue isométrique : AmazeBalls
46 Création de romans graphiques
50 Fabriquer une machine à karaoké
54 Images de l'ISS par ondes radio
60 Électronique avec GPIO Zero 1.5
64 Lanceur de bombe de table piloté par un Raspberry Pi
68 Surveillance de plantes par IA
72 Programmer le RPi en Java 11

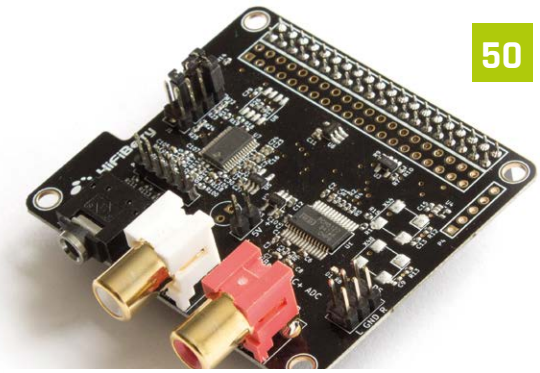
Bancs d'essai

77 Zero Dongle - RPi Zero transformé en clé USB
86 HAT Breakout Garden
88 Snips – reconnaissance vocale pour le Raspberry Pi

Divers

96 Raspberry Pi @Elektor
98 Autres trouvailles intéressantes

Images de l'ISS



Fabriquer une machine à karaoké

GPIO Zero 1.5

Surveillance de plantes

Découvrez Raspberry Pi 4

Raspberry Pi 4 marque une étape majeure dans la réinvention de la gamme Raspberry Pi.

Spécifications

SOC : Broadcom BCM2711B0 quadricœur A72 (ARMv8-A) 64 bits @ 1,5 GHz

PROCESSEUR GRAPHIQUE : Broadcom VideoCore VI

RÉSEAU : réseau local sans fil 802.11b/g/n/ac, de 2,4 GHz à 5 GHz

MÉMOIRE RAM : 1 Go, 2 Go ou 4 Go de mémoire SDRAM LPDDR4

BLUETOOTH : Bluetooth 5.0, Bluetooth Low Energy (BLE)

GPIO : barrette GPIO à 40 broches installée

SUPPORT DE STOCKAGE : carte micro-SD

PORTS : 2× micro-HDMI 2.0, jack audio-vidéo analogique de 3,5 mm, 2× USB 2.0, 2× USB 3.0, Gigabit Ethernet, interface série pour caméra (CSI), interface série pour écran (DSI)

DIMENSIONS : 88 mm × 58 mm × 19,5 mm, 46 g

A | UNITÉ CENTRALE
Le nouveau SoC BCM2711B0 offre un gain de performance exceptionnel par rapport à ses prédécesseurs.

B | ALIMENTATION
Grâce à l'évolution vers le connecteur USB de type C pour l'alimentation, Raspberry Pi 4 accepte des appareils USB consommant davantage de courant.

Guide de démarrage rapide

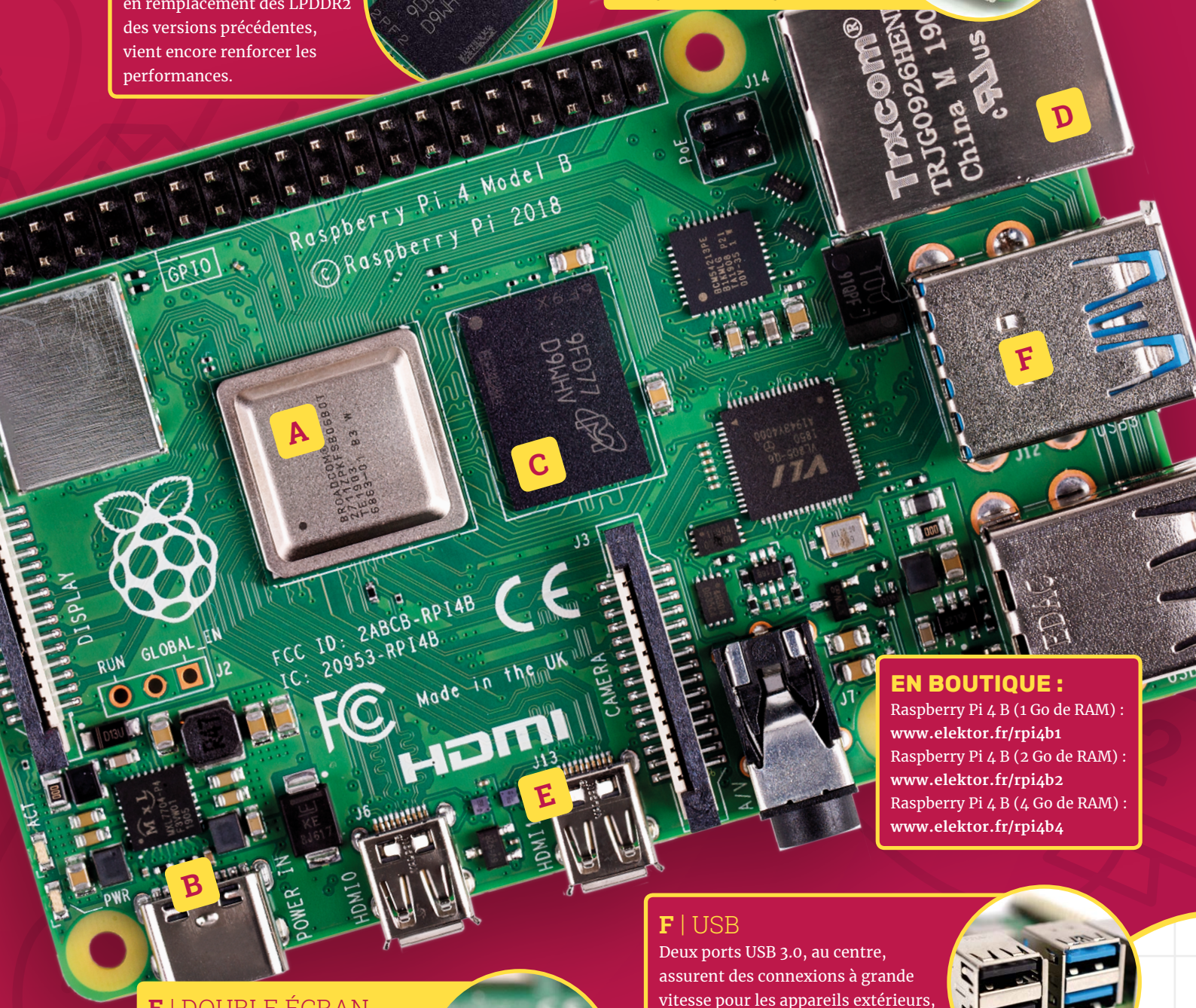
Raspberry Pi 4 est directement compatible avec le modèle 3B+ et toutes les versions précédentes de Raspberry Pi. Toutefois vous devrez mettre à jour votre système d'exploitation pour le rendre compatible avec le nouveau SoC. *La méthode la plus simple pour vous lancer est de consulter notre guide de démarrage rapide qui vous accompagnera pour implanter le tout récent installateur NOOBS sur une carte micro-SD.*
magpi.cc/quickstart

C | MÉMOIRE RAM

L'évolution vers une capacité de 4 Go de mémoire LPDDR4, en remplacement des LPDDR2 des versions précédentes, vient encore renforcer les performances.

D | ETHERNET

Le port Ethernet, déplacé vers la partie supérieure droite de la carte, offre aujourd'hui une connexion réseau à pleine vitesse, sans goulots d'étranglement.



EN BOUTIQUE :

Raspberry Pi 4 B (1 Go de RAM) : **www.elektor.fr/rpi4b1**
Raspberry Pi 4 B (2 Go de RAM) : **www.elektor.fr/rpi4b2**
Raspberry Pi 4 B (4 Go de RAM) : **www.elektor.fr/rpi4b4**

E | DOUBLE ÉCRAN

Grâce à ses deux connecteurs micro-HDMI, Raspberry Pi 4 peut piloter deux écrans 4K, avec une résolution de 4Kp30, ou un seul avec une résolution de 4Kp60.

F | USB

Deux ports USB 3.0, au centre, assurent des connexions à grande vitesse pour les appareils extérieurs, notamment les supports de stockage et les accélérateurs matériels.

(VF : Pascal Godart)

Lancement de la bibliothèque GPIO Zero

Mieux maîtriser vos projets Python avec la nouvelle version de GPIO Zero. **Rosie Hattersley.**

GPIO Zero est une bibliothèque Python de fonctions précodées destinées à faciliter la commande du matériel connecté au RPi : faire clignoter des LED, déclencher un buzzer ou faire tourner des roues devient un jeu d'enfant. Conçue pour simplifier l'accès au RPi, cette bibliothèque est tout particulièrement destinée à l'enseignement.

Devenir un expert de cette bibliothèque est nettement plus facile avec la version 1.5.

C'est une excellente nouvelle si vous souhaitez vous lancer dans des projets informatiques physiques avec votre RPi, par ex. de la robotique ou de la domotique, puisque vous serez mieux accompagné dans vos expérimentations.

De nombreux kits, qui nécessitaient jusqu'ici un volume significatif de codage, sont aujourd'hui intégrés et gérés. Vous y découvrirez des projets RPi (Halloween, sapin de Noël à LED et HAT Jam – extension pour feux de circulation de ModMyPi avec buzzer piézoélectrique et boutons).

Pour vous amuser avec le son, essayez la version améliorée de l'alarme à plusieurs tons de Claire Pollard. Selon Ben Nuttall, créateur de la bibliothèque GPIO Zero, la nouvelle version contient une classe *TonalBuzzer* et un dictionnaire pour faciliter le choix de notes et de fréquences MIDI adaptées.

Appairage de dispositifs

Les fonctions de rappel (*callback*) offrent un bon moyen de connecter des appareils. Avec la version 1.5 de la bibliothèque GPIO Zero, le raccordement de dispositifs est encore plus facile. Pour appairer deux appareils, vous pouvez utiliser la méthode suivante :

« De nombreux kits, qui nécessitaient jusqu'ici un certain volume de codage, sont aujourd'hui intégrés et gérés. »

▼ Les makers les plus avertis peuvent créer leurs propres *pin factories* (« fabriques » de broches) pour commander les broches GPIO de leur carte Raspberry Pi.

```
led.source = button
```

Outre les *pin factories* standard (RPi.GPIO, pigpio, RPIO, fonctions natives), les utilisateurs chevronnés apprécieront la possibilité de *custom factories* dans la bibliothèque.

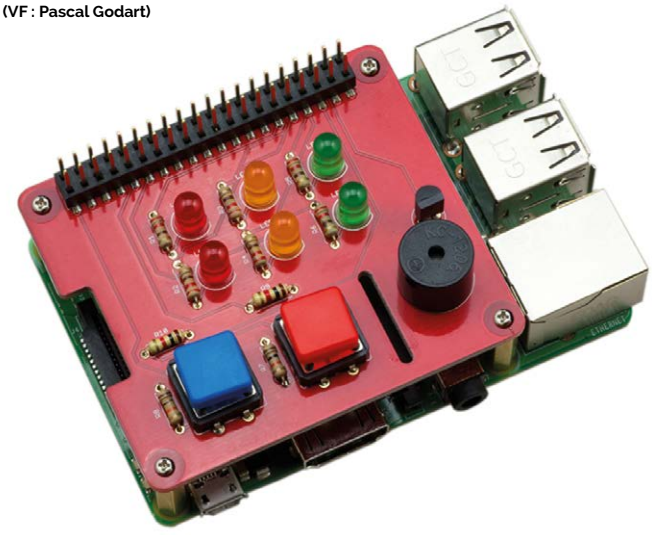
À titre d'exemple, Google a écrit sa propre *pin factory* pour le kit AIY Vision - vision pour intelligence artificielle « maison » (**magpi.cc/imfAQZ**).

La bibliothèque pigpio permet de commander à distance des broches et des capteurs spécifiques. Si vous vous êtes débattu dans le passé avec la classe *DistanceSensor*, n'hésitez pas à passer à la version 1.5 et à utiliser la *pin factory* pigpio. La précision de détection et de mesure n'en sera que meilleure.

Pour mettre à jour votre bibliothèque GPIO Zero, entrez les commandes suivantes sur le Terminal :

```
sudo apt update
sudo apt install python3-gpiozero python-gpiozero
```

Pour une liste complète des nouveautés de la bibliothèque GPIO Zero 1.5, voir le journal des modifications (*changelog*) : **magpi.cc/vgTmJj**. **(VF : Pascal Godart)**



La bibliothèque GPIO Zero prend directement en charge de nombreux kits de matériel largement diffusés, comme le nouveau HAT Jam de ModMyPi.

Nouveaux dongles AutoPi

De nouveaux dongles AutoPi, l'interface OBD-II pour automobiles basée sur le RPi, sont disponibles. Cette deuxième génération est plus rapide et possède une horloge en temps réel.

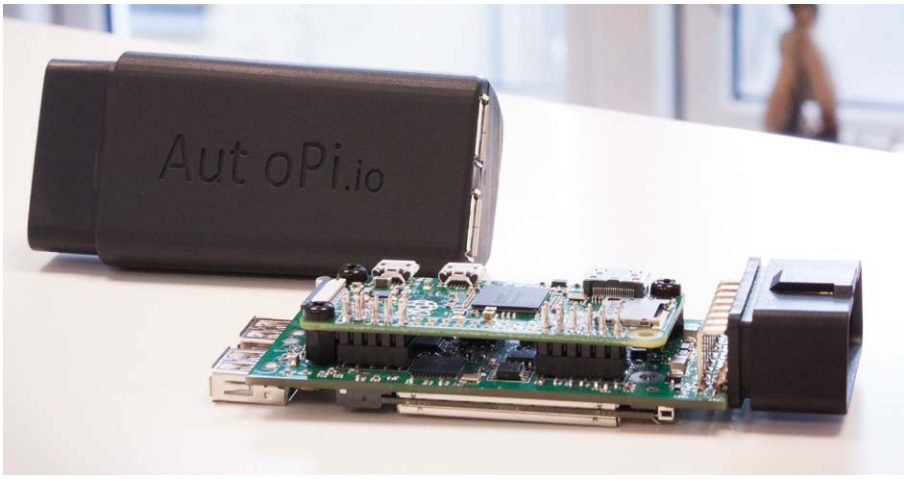
Le dongle AutoPi (**www.autopi.io**) que l'on branche sur le port OBD-II d'une voiture, contient un Raspberry Pi Zero W, un modem 4G, un GPS et un accéléromètre. Le dongle peut ainsi collecter diverses données en temps réel et les transmettre.

La firme danoise qui a mis AutoPi sur le marché a sorti ses dongles de deuxième génération il y a peu. Le modem 4G/LTE est désormais un module de catégorie 4 ; il permet des vitesses de téléchargement jusqu'à 150 Mbps, et de téléversement jusqu'à 50 Mbps. C'est suffisant pour faire de l'AutoPi un point d'accès Wi-Fi.

Les poids lourds avec une tension d'entrée de 35 V sont aussi supportés, et la consommation au repos a été réduite. L'horloge en temps réel permet de disposer de données temporelles correctes hors ligne. Le connecteur pour cartes SIM accepte désormais les cartes au format 4FF (nano SIM).

Le prix des dongles est quant à lui resté le même. Ils sont disponibles sur **shop.autopi.io** : de 105 € pour le modèle DIY sans RPi à 275 € pour celui équipé de 4G/LTE. **(VF : Jean-Louis Mehren)**

Les nouveaux dongles AutoPi sont plus rapides et possèdent une horloge en temps réel.



Un calendrier intelligent grâce au Raspberry Pi

Vitec GmbH mise sur une large diffusion de son produit cadencé par un RPi, comme l'a découvert **Lucy Hattersley**.

Les calendriers intelligents sont des projets RPi extrêmement appréciés, même si tout le monde n'a pas envie de fabriquer le sien. Certains préféreront acheter un produit clé en main plutôt que de le réaliser.

C'est toute l'ambition de Vitec GmbH : proposer au plus grand nombre son calendrier à base de RPi, le *Smart Calendar Thync* (**vitec-visual.com**).

« Pour la plupart des familles, gérer les activités quotidiennes, avec d'innombrables tâches et rendez-vous, est un cauchemar », écrit Marc Roset, PDG et fondateur de Vitec GmbH. « La majorité des hommes d'affaires et des professionnels considèrent qu'il est impossible de concilier leurs obligations professionnelles et leurs activités familiales ».

▶ Le calendrier intelligent contient un circuit imprimé sur-mesure qui accueille un RPi Zero.

▼ Le *Smart Calendar Thync* utilise un RPi pour mettre le produit à la portée du plus grand nombre.

C'était sans compter sur le *Smart Calendar Thync*, qui permet de synchroniser et de connecter jusqu'à cinq plannings de rendez-vous différents. L'appareil avec écran TFT de 10,1 pouces contient un RPi Zero W alimenté par batteries.

Plus remarquable encore, le *Smart Calendar Thync* dispose d'une interface à reconnaissance gestuelle. « D'un simple geste de la main, vous pouvez ainsi aller et venir dans l'interface de manière rapide et facile. Un mouvement discret vers le haut déclenche l'affichage d'informations supplémentaires sur la fonction active », précise Marc.

L'application de Vitec, proposée pour les systèmes iOS et Android, est le moyen de plus simple pour configurer le produit. « Nous n'avons pas opté pour une commande vocale, car de nombreux utilisateurs ne veulent pas être écoutés ».

Détection de lumière ambiante

Le *Smart Calendar Thync* dispose également d'un capteur de lumière ambiante qui mesure la luminosité de l'environnement proche, régule la sienne et optimise la consommation d'énergie.

Le *Smart Calendar Thync* peut être installé sur un mur ou une table (avec le support fourni). Une campagne Kickstarter a été lancée le 7 avril 2019. M

(VF : Pascal Godart)

Le Raspberry Pi et les radioamateurs

De tout temps les radioamateurs ont été à la pointe de la technique et de l'expérimentation. Avec l'avènement des récepteurs SDR et l'arrivée du numérique dans les transmissions radio, le Raspberry Pi a tout naturellement trouvé sa place. Il est devenu un incontournable du fait de son prix abordable et de sa formidable flexibilité. Voici quelques possibilités offertes par ce nano-ordinateur.

Une quantité de programmes OM présents dans Raspbian

Le lien **[1]** référence plus de 120 programmes à destination des radioamateurs, directement accessibles depuis Raspbian. Un simple *apt-get install* suffit pour les installer et en profiter rapidement. De nombreux autres programmes développés par des radioamateurs ne sont pas dans cette liste, mais seront accessibles via des plateformes comme SourceForge ou Github.

Note : l'acronyme « OM » (abréviation radiotélégraphique de *Old Man*) est souvent utilisé pour remplacer le terme « radioamateur ». On trouve maintenant l'acronyme « YL » pour *Young Lady*, sans distinction d'âge. C'est une femme qui utilise une radio (autrefois c'était réservé aux hommes).

Expédition scientifique

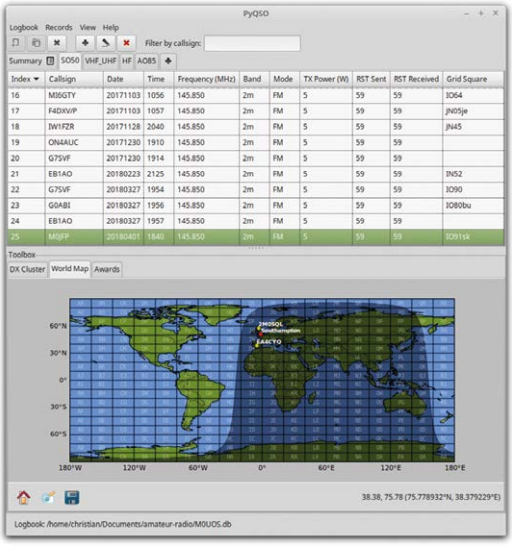
▲ **Figure 1.** Le ballon sort des nuages. (© F4HHV)

L'équipe de Fabrice (F4HHV) **[2]** a lancé un ballon. L'objectif était de tester le comportement d'un écran OLED en altitude, dans des conditions extrêmes (altitude de 27 km, pression de 200 mbar à -60 °C). Le suivi de la nacelle était assuré par APRS. Il s'agit d'un système de radiocommunication numérique utilisé par les radioamateurs : le « système de transmission automatique par paquets » (*Automatic Packet Reporting System*) permet de visualiser automatiquement et en temps réel les positions de stations radio et de divers objets sur des cartes affichées sur un PC.

En deux heures de vol, l'engin a pu tester la résistance de l'écran, filmé en permanence par une caméra RPi. Il a également fourni des images de très bonne qualité grâce à cette caméra de 8 Mp (**figures 1** et **2**).

▲ **Figure 2.** Delta du Rhône. (© F4HHV)

Gestion des *logs*



▲ **Figure 3.** Capture d'écran de PyQSO. (© christianjacobs.uk)

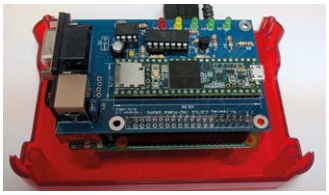
Une des premières utilisations du Raspberry Pi dans la station d'un radioamateur est la gestion des *logs* (**figure 3**). Le *logbook* ou carnet de trafic sert à enregistrer les QSO (en clair les contacts). Autrefois il s'agissait d'un simple cahier tenu à la main,

MAKER

François Mocq

Électronicien, radioamateur (indicatif F1GYT), formateur informatique à la retraite. Il partage son temps entre le blog framboise314.fr, l'écriture de livres sur le Raspberry Pi et des conférences.

framboise314.fr



▲ **Figure 4.** Carte TNC-PigK6, un TNC 1200 de 2400 à 9600 bauds. (© https://tnc-x.com/)

▲ **Figure 6**. Carte relais MMDVM. (© F8BSY Xavier)

▲ **Figure 9.** Récepteur SDRPlay et Raspberry Pi. (© Andrew Back)

aujourd'hui il est dématérialisé. Des programmes comme PyQSO **[3]** peuvent aider à gérer les contacts en récupérant automatiquement les données sur **qrz.com** ou **hamqth.com**. En plus du tri des contacts, ils proposent l'impression ou la création du journal en PDF, le suivi du DXCC, la gestion des doublons, etc.

Apprendre/recevoir/envoyer la CW

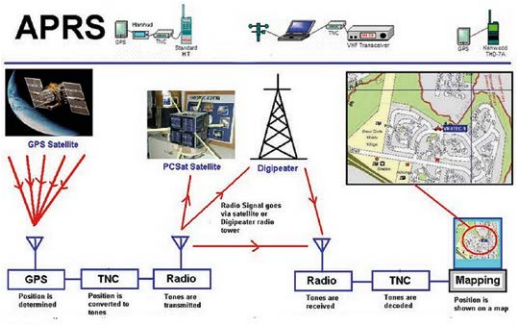
La télégraphie (*Continuous Waves*) est également au menu du Raspberry Pi. Celui-ci peut produire des suites de caractères pour l'apprentissage ou traduire les frappes sur le clavier en code morse pouvant être émis directement (mais où est le plaisir). Certains programmes sont capables de décoder la CW, mais il faut bien reconnaître que si l'opérateur est expérimenté et utilise un double contact… le Raspberry Pi aura beaucoup de mal à rivaliser avec une oreille bien entraînée !

Trafiquer en AX.25

Le mode de transmission AX.25 développé par les radioamateurs a connu ses heures de gloire dans les années 80. Il est peu à peu tombé dans l'oubli, remplacé par d'autres solutions numériques. Il est à la base des réseaux Packet et APRS, dans les bandes VHF et UHF, avec des débits compris entre 300 et 9600 bauds. Des cartes d'extension à 1200 et 9600 bauds ont été conçues pour le Raspberry Pi (**figure 4** et **[4]**).

Réseau APRS

L'APRS [5] ou système de transmission automatique par paquets permet le partage d'informations entre les stations OM (**figure 5**). Il permet l'affichage, sur une carte, de données comme l'indicatif, la localisation des véhicules et des personnes, des rapports météo et/ou des messages en temps réel.



▲ **Figure 5.** Principe du réseau APRS. (© nwclimate.org)

MMDVM, *Multimode Digital Voice Modem*

C'est dans le domaine du MMDVM (la radio numérique) que le Raspberry Pi montre toute sa puissance et sa flexibilité. Il existe différents modèles de cartes, de la plus élémentaire, compatible avec le Raspberry Pi Zero, qui permet de créer un hotspot simple pour des tests en numérique, à la carte de la **figure 6**. Elle est compatible PiStar et permet de réaliser un véritable relais numérique.

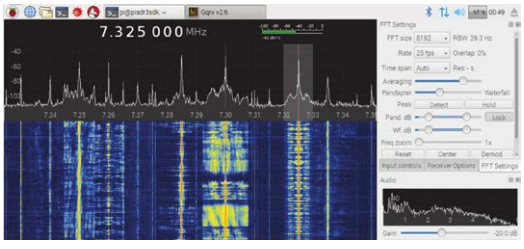
Paramétrage de sa station

CHIRP **[7]** est un logiciel gratuit et à code source ouvert. Il permet de programmer les appareils utilisés par les radioamateurs. CHIRP prend en charge des dizaines de fabricants et de modèles ; il gère de nombreux types de fichiers de données (**figure 7**).

▲ **Figure 7.** Copie d'écran de CHIRP.

Réception des bandes amateurs

L'arrivée sur le marché de clés USB SDR a permis de développer des récepteurs numériques. Les premières versions fonctionnaient sur PC, mais l'arrivée des nano-ordinateurs a facilité le portage vers ces cartes peu coûteuses, comme le Raspberry Pi **[8]** (**figure 8**).



▲ **Figure 8.** Réception sur la bande des 40 m (7 MHz) sur le Raspberry Pi. (© gqrx.dk)

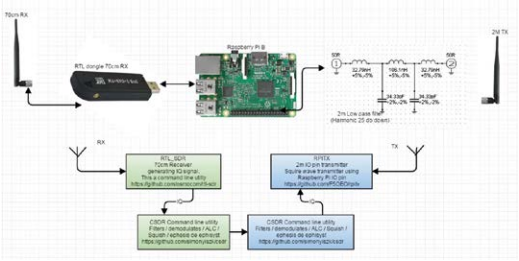
Il existe des systèmes plus puissants comme le SDRPlay **[9]** qui a ouvert la voie à la réalisation de récepteurs performants et peu chers (**figure 9**). Ce matériel couvre de 1 kHz à 2 GHz et possède deux entrées d'antennes coaxiales (0-30 MHz et VHF-UHF). Il dispose d'un LNA (*Low Noise Amplifier*, ampli à faible bruit). Une entrée d'antenne à haute impédance est également disponible pour la HF.

La mise en service est facilitée par des applications spécialisées. La réception de l'ADS-B (système de surveillance coopératif pour le contrôle

du trafic aérien et autres applications connexes) ou la réalisation d'un serveur SDR (*Software-Defined Radio*, radio logicielle) sont à la portée de nombreux OM **[10]**.

Mini-transpondeur avec un Raspberry Pi

Le logiciel rpiTx développé par F5OEO [11] permet d'émettre directement depuis un des ports GPIO du Raspberry Pi. Couplé avec un récepteur SDR et un filtre de bande, rpiTx permet de réaliser sans gros investissement un transpondeur de faible puissance **[12]** (**figure 10**).



▲ **Figure 10**. Un transpondeur pour les bandes de 70 cm à 2 m avec un Raspberry Pi. (© ZR6AIC)

Balise WSPR

Le WSPR (*Weak Signal Propagation Reporter*) transmet avec une faible puissance et un débit très lent des informations sur la station (indicatif, QRA locator et puissance). Les données sont disponibles sur le site wsprnet **[13]**.

La balise WSPR de HB9FGK fonctionne en 30 m (10,14 MHz) avec une puissance de 10 mW (**figure 11**). Elle est équipée d'un filtre de bande sur la sortie GPIO et a été reçue à… 2850 km de distance !

Équiper un relais radioamateur

Le Raspberry Pi est susceptible d'équiper un relais en offrant des possibilités étendues : gestion de la logique du relais, pilotage du relais à distance via Internet ou par DTMF, synthèse vocale (accueil / indicatif / valeur Smètre / température / humidité / pression…), surveillance de l'alimentation, alertes…

Ensemble d'émission/réception complet RadioBerry

Pour terminer ce tour d'horizon (non exhaustif) des possibilités du Raspberry Pi, voici le système RadioBerry **[14]** développé par PA3GSB.

C'est un émetteur/récepteur autonome, basé sur un Raspberry Pi et une carte fille conçue par PA3GSB autour d'un FPGA Intel Cyclone dont le micrologiciel est chargé via le RPi (**figure 12**). Il utilise un modem à large bande, avec convertisseur analogique/numérique sur 12 bits, d'Analog Devices (AD9866) et couvre de 0 à 30 MHz.

Distribution prête à l'emploi

Il en existe de nombreuses versions. Par exemple piSDR **[15]** de Luigi PU2SPY qui intègre gQRX, SDRAngel, GNURadio Companion, LimeUtil, LimeVNA et Soapy. Un grand nombre de périphériques est déjà géré : RTL-SDR, LimeSDR Mini, LimeSDR USB, AirSpy Mini, AirSpy R2. Le développement de cette distribution continue et vous pouvez y participer si vous êtes intéressé(e).

Conclusion

Au travers de ces exemples qui présentent une partie des applications possibles du Raspberry Pi dans le monde radioamateur, vous avez pu découvrir cette fabuleuse petite carte. La standardisation apportée par le Raspberry Pi et la disponibilité de distributions prêtes à l'emploi facilitent la mise en œuvre de ces technologies dans la station. C'est tout un monde qui s'ouvre à l'OM qui voudra s'investir dans ce domaine. M

▲ **Figure 11.** Balise WSPR de HB9FGK. (© hb9fgk)

▲ **Figure 12**. Émetteur/récepteur RadioBerry. (© PA3GSB)

Webographie

[1] Programmes pour les radioamateurs pour Raspbian : **http://www.raspberryconnect.com/raspbian-packages-list/item/71-raspbian-hamradio**

[2] Ballon dans la stratosphère : **http://fab4space.000webhostapp.com/language/fr/**

[3] Programme de gestion des logs : **https://christianjacobs.uk/pyqso/**

[4] Carte pour AX.25 : **https://tnc-x.com/**

[5] APRS : **http://www.nwclimate.org/aprs/introduction-to-aprs/**

[6] Carte de radio numérique : **http://www.ref66.fr/2018/11/05/pi-star-configurer-en-un-clin-doeil-votre-hotspot_duplex/**

[7] Logiciel de programmation CHIRP : **https://chirp.danplanet.com/projects/chirp/wiki/Home**

[8] Logiciel Gqrx SDR : **http://gqrx.dk/download/gqrx-sdr-for-the-raspberry-pi**

[9] Récepteur SDRPlay RSP2 PRO : **https://www.passion-radio.fr/recepteurs-sdr/rsp2-463.html**

[10] Serveur SDR : **https://www.rs-online.com/designspark/building-a-remote-sdr-with-the-pi-3-model-b-and-sdrplay-rsp2**

[11] Logiciel rpiTx : **https://github.com/F5OEO/rpitx**

[12] Transpondeur de faible puissance : **http://zr6aic.blogspot.com/2016/11/creating-2m-fm-repeater-with-raspberry.html**

[13] Site wsprnet pour les balises WSPR : **http://wsprnet.org/drupal/**

[14] Système complet d'émission/réception : **https://github.com/pa3gsb/RadioBerry**

[15] Distribution prête à l'emploi : **https://github.com/luigifreitas/pisdr-image**

PROJETS RADIO-AMATEURS ÉTONNANTS

Raspberry Pi et radioamateurisme, un mariage réussi célébré par **Rob Zwetsloot**.

Peut-être imaginez-vous les radioamateurs comme des reclus dont le passe-temps favori consiste à tourner un bouton dans l'espoir de tomber sur un autre reclus. Oubliez ce cliché, les radioamateurs sont avant tout des explorateurs des ondes, doublés d'inlassables expérimentateurs. Les plus avertis d'entre eux construisent même leurs propres satellites de communication ! Pas étonnant donc de retrouver le RPi au cœur de nombreux projets radioamateurs, dont ceux présentés ici.

PAS SI AMATEURS QUE ÇA

L'histoire du radioamateurisme est indissociable de celle de la radiodiffusion.

Dès leur regroupement en clubs dans les années 1910, les radioamateurs ont contribué à l'essor des techniques et des applications de la radiodiffusion. L'exemple le plus remarquable de cet apport est l'établissement, en 1923, de la première liaison transatlantique en ondes courtes. L'usage de ces ondes devint dès lors si frénétique et désordonné qu'après la Grande Guerre, l'U.I.T. (Union Internationale des Télécommunications) décida de réglementer l'utilisation des fréquences radioélectriques.

Aujourd'hui un radioamateur doit être titulaire d'une licence pour émettre, et ne le peut le faire que sur les bandes de fréquence allouées au « service d'amateur ».

Époque moderne

Le déploiement de l'électronique de loisir, des micro-ordinateurs et des smartphones n'a fait qu'accroître les possibilités offertes aux radioamateurs. La baisse du prix des composants et l'augmentation de la puissance de calcul ont ainsi permis à bon nombre d'entre eux de mettre le RPi au cœur de leurs projets. Ceux que nous vous présentons ici ont à nos yeux le double mérite d'être à la fois étonnants et source d'inspiration.

Ham radio ?

Les anglophones surnomment « ham radio » la radio amateur et « hams » ses opérateurs. Le qualificatif *ham* désignait à l'origine les télégraphistes incompétents et maladroits (*ham-fisted*) dans la manipulation du code Morse. L'usage a retenu ce sobriquet, pourtant péjoratif.

▲ Après la guerre de 39-45, le matériel utilisé par les radioamateurs provenait souvent des surplus militaires.

▲ Des ingénieurs de la Poste britannique vérifiant un poste émetteur-récepteur en 1897. *Credit : Cardiff Council Flat Holm Project.*

◀ Un radioamateur allemand en randonnée avec son équipement « portable ».

RADIOAMATEURISME AVEC LE RASPBERRY PI

Quelques préalables matériels, logiciels et… administratifs.

Il existe d'innombrables ressources matérielles et logicielles pour s'essayer aux techniques propres au radioamateurisme. La communauté des radioamateurs s'étant emparée du Raspberry Pi dès son lancement, bon nombre de paquets Linux existants ont été portés sur Raspbian.

Prérequis

UN RASPBERRY PI – Un modèle récent est préférable, mais de nombreuses applications fonctionneront aussi bien avec un RPi 3B+ qu'avec un RPi Zero.

RASPBIAN – La plupart des logiciels concernés sont compatibles avec Raspbian, donc privilégiez ce système d'exploitation sur la carte SD de votre RPi.

LICENSE DE RADIO-AMATEUR – La plupart des pays, dont la France, exige de tout radioamateur qu'il soit titulaire d'une licence.

Principe de base

De la même façon que l'on peut transmettre des données sous forme de motifs lumineux dans une fibre optique, des informations peuvent être transmises par voie aérienne en modulant de façon spécifique des ondes radio. Le codage du signal se fait en modulant (en fréquence ou en amplitude selon l'application en jeu) une onde dite porteuse. L'opération inverse, la démodulation, est le décodage du signal reçu.

Dans le contexte qui nous intéresse, c'est le RPi, souvent couplé à un autre dispositif spécialisé, qui sert de matériel de codage et décodage. Un ordinateur est plus facilement adaptable à la lecture, à l'envoi et au traitement de données – un autoradio serait incapable de visualiser des trajectoires d'avion.

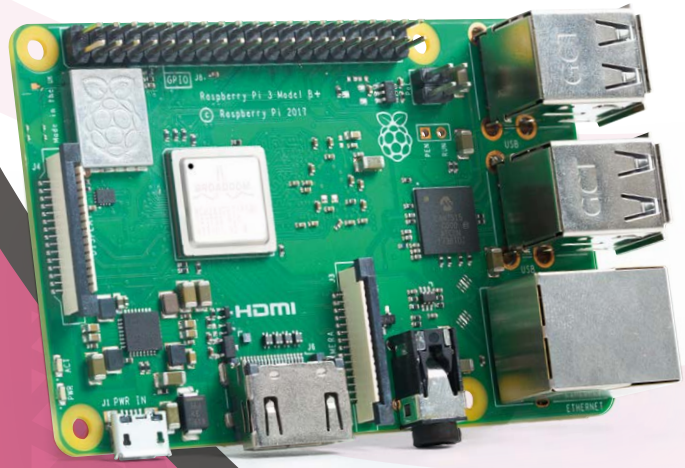
Fait pour la radio

La portabilité du RPi, sa faible consommation et son adaptabilité sont des atouts indéniables pour le radioamateurisme. Un logiciel approprié saura facilement lire et émettre des données via ses broches GPIO, et quelques périphériques auxiliaires feront du RPi un dispositif opérationnel à la fois léger et durable une fois relié à une alimentation portable.

Les projets suivants montrent quelques réalisations de radioamateurs où le RPi occupe une place de choix.

RAPPEL !

La loi française exige de tout radioamateur qu'il détienne un « certificat d'opérateur des services d'amateur » (une licence) dont la délivrance est conditionnée par le passage d'un examen. Plus d'informations sur le site **https://www.anfr.fr**.



▼ Un RPi 3 modèle B+ est recommandé pour les applications radio, mais dans bien des cas un RPi Zero suffit.

PROJETS RADIOAMATEURS ÉTONNANTS

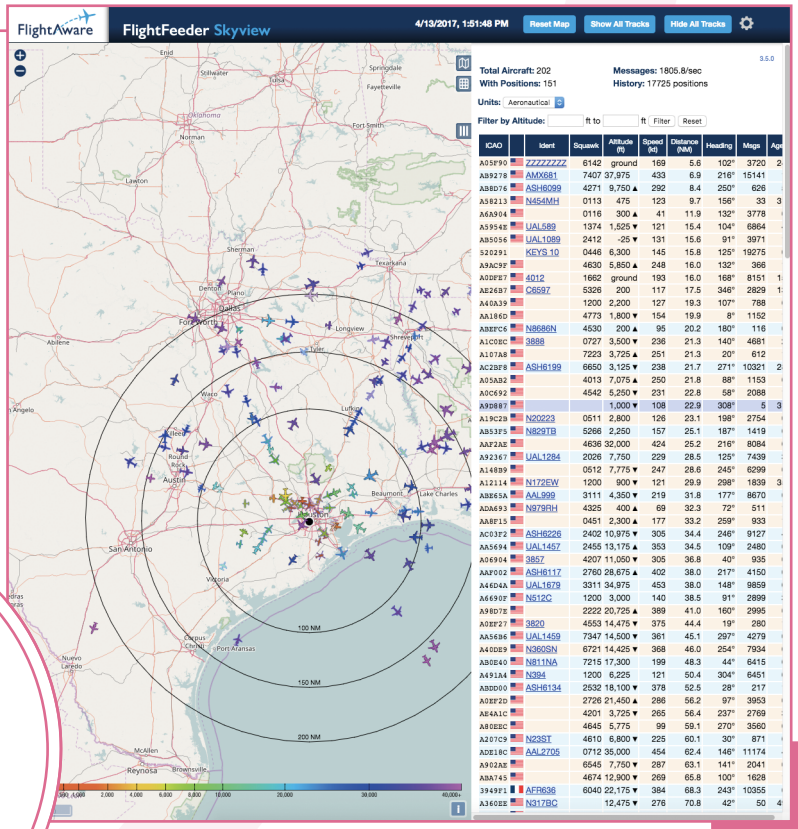
Puissent ces projets émettre aussi des ondes positives jusque chez vous !

Suivi de vol

Un grand classique : transformez votre RPi en station au sol ADS-B (« surveillance dépendante automatique en mode diffusion ») et recevez en temps réel les données de vol des avions dans un rayon de 160 à 480 km (la portée dépend de l'antenne utilisée). La même technologie sert au suivi des ballons-sondes de haute altitude.

Le logiciel PiAware de FlightAware permet de tracer l'itinéraire d'un avion à partir des données radio reçues.

magpi.cc/phwYXr



Le logiciel PiAware permet de suivre et visualiser en temps réel des itinéraires d'avion sur un rayon allant jusqu'à 300 miles (env. 480 km).

> « Utile pour savoir si l'avion d'un proche est en retard. »

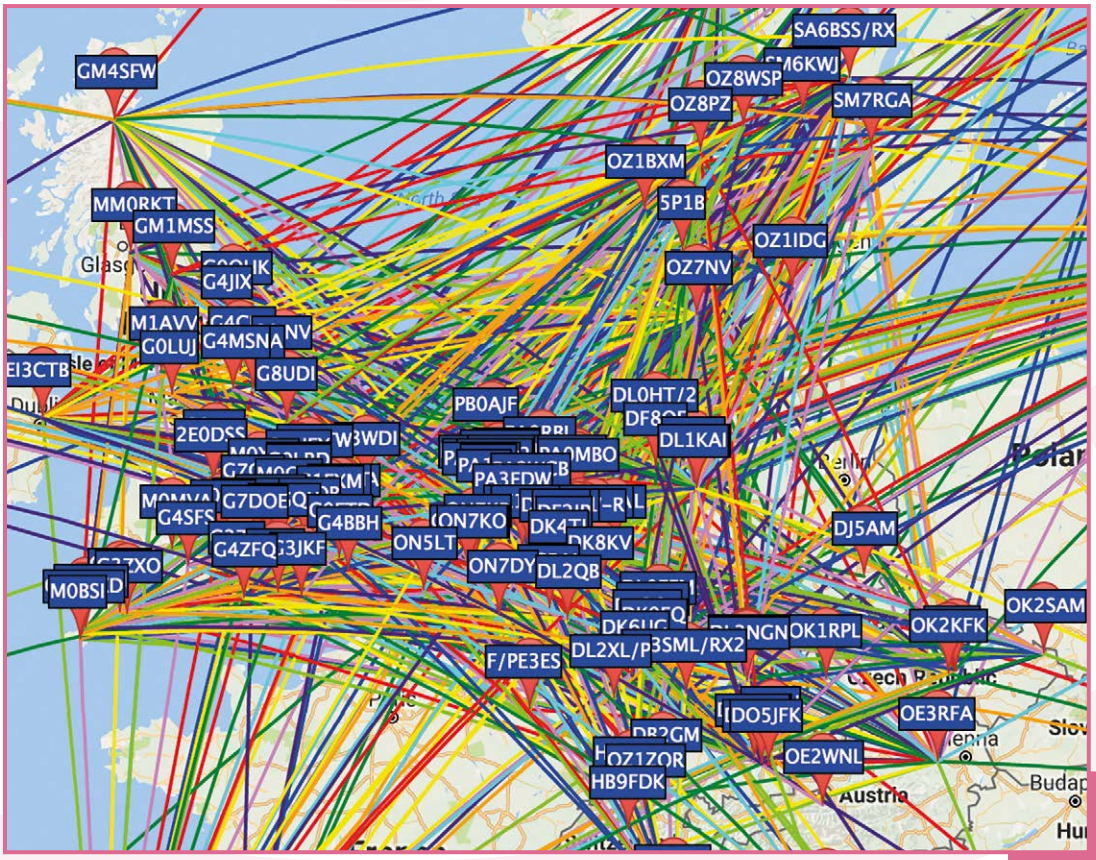
Émetteur WSPR

La transmission WSPR (prononcez *whisper*), soit *Weak Signal Propagation Reporter* ou *reporter de propagation de signal faible* en français, permet de tester la distance maximale pouvant être atteinte avec une puissance minimale. De nombreux radioamateurs réalisent des balises WSPR avec un RPi couplé à une antenne, p. ex. avec le logiciel WsprryPi

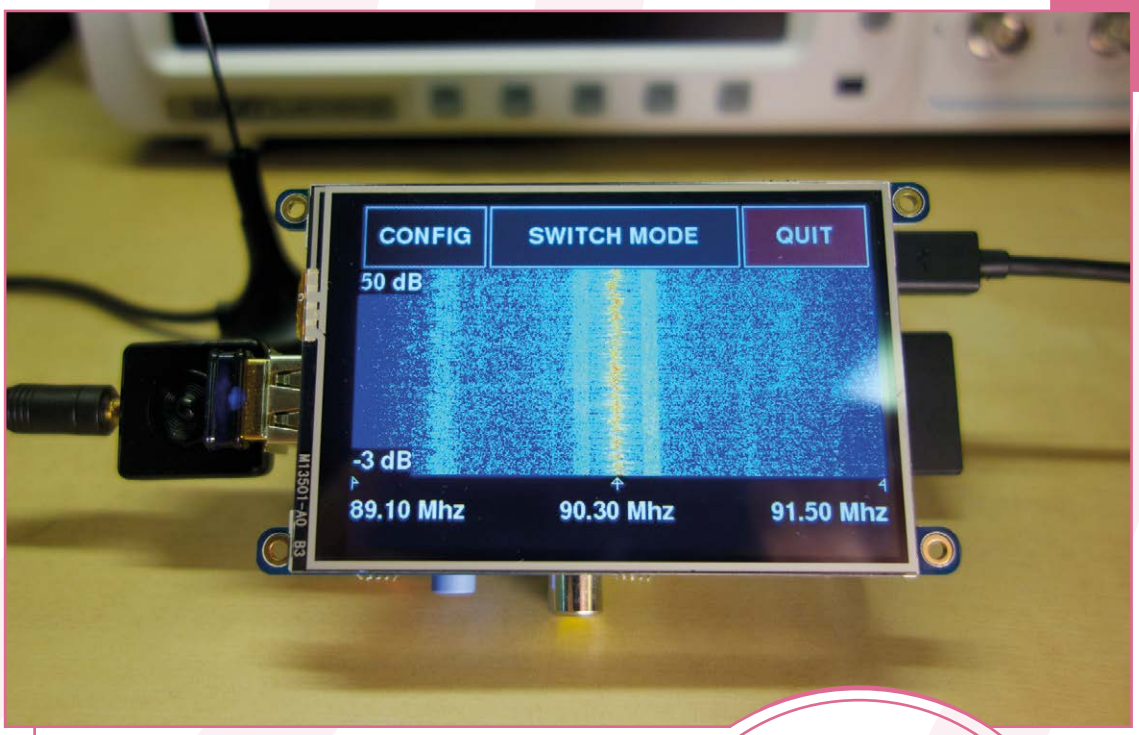
Comme de nombreuses stations actives communiquent les rapports de réception WSPR, les radioamateurs du site WSPRnet ont pu créer une carte affichant la localisation des balises reçues, leur indicatif d'appel, puissance d'émission, etc. Visitez WSPRnet pour plus de détails.

magpi.cc/EjQpbd

▼ Reliez les broches GPIO4 et GND de votre RPi à une antenne via un filtre passe-bas, et vous aussi pourrez apparaître sur WSPRnet.



Et aussi, page 54 :
comment recevoir des images de l'espace par ondes radio !



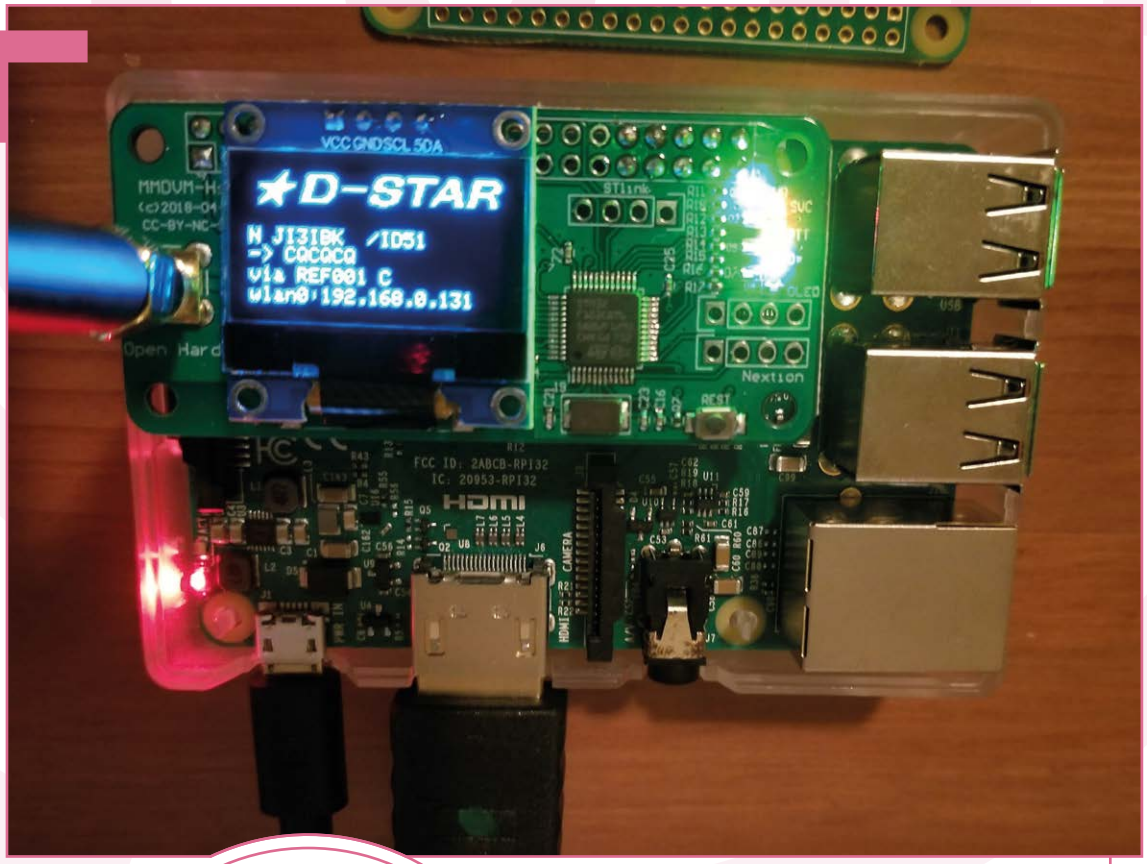
Radio logicielle

Les récepteurs radio dont le décodage du signal est effectué pour l'essentiel par logiciel sont dits SDR (*software-defined radio*). Une radio logicielle ajoute une couche de flexibilité à la réception du signal, et peut bien sûr être assemblée à partir d'un RPi.

Dans ce projet d'Adafruit, la partie matérielle du récepteur est une clé USB RTL-SDR à puce RTL2832U. Le signal reçu est ensuite traité par un programme dérivé du logiciel populaire SDRSharp. Suivez le guide ci-dessous pour construire votre propre « scanner SDR à RPi ».

magpi.cc/jawZRE

« Une radio logicielle ajoute de la flexibilité au traitement du signal. »



« Ce point d'accès peut aussi servir de relais numérique. »

Point d'accès DV

Tout comme en télévision, le passage de l'analogique au numérique en radio permet d'améliorer la qualité du signal et de réduire la bande passante utilisée. Les radioamateurs disposent à cet effet de quelques systèmes et protocoles de codage/décodage à mode numérique, dont D-STAR, le protocole de transmission de la voix (DV, *Digital Voice*) et des données utilisé par ce *hotspot* (point d'accès) qui peut aussi servir de relais numérique.

Le projet repose sur Pi-Star, une surcouche de Raspbian contenant de nombreux outils et logiciels pour radioamateurs.

magpi.cc/VjYZcp

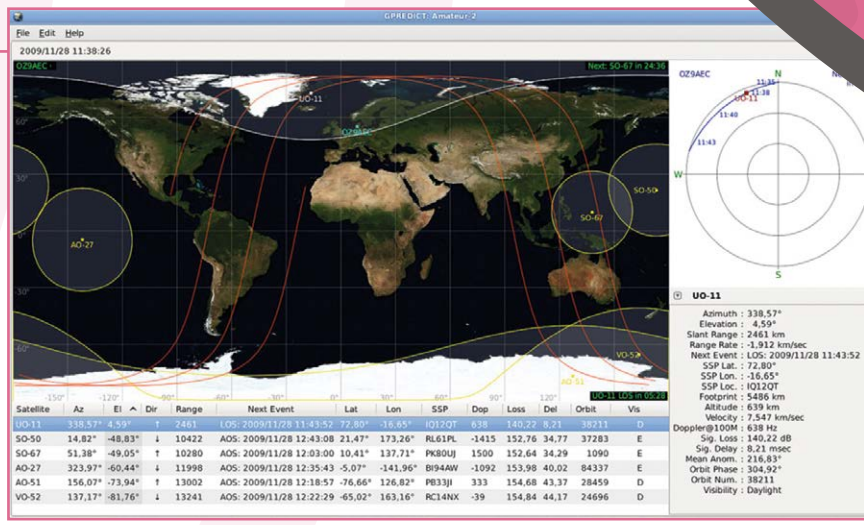
Vous avez réalisé un projet radioamateur étonnant ?
Twittez-nous vos photos ! **@MagPi_FR**

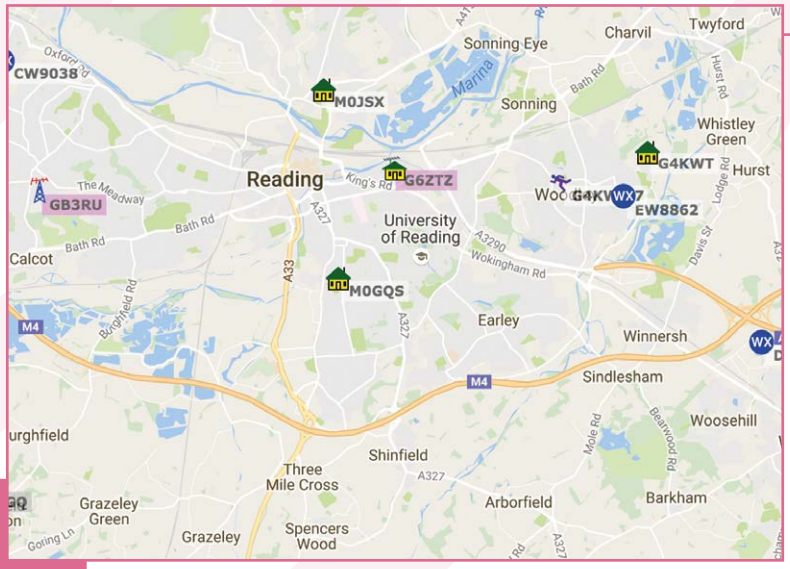
Suivi de satellite

Ce projet ressemble au suivi de vol précédent, mais ici le ciel devient l'espace, ce qui est plutôt cool ! Le suivi est là aussi en temps réel et provient de Gpredict, un logiciel libre offrant de nombreuses options de configuration pour l'affichage des orbites et des données associées aux satellites dont, coucou là-haut, la Station spatiale internationale.

Gpredict peut aussi commander l'orientation d'une antenne mobile. Notez qu'il vous faudra compiler vous-même Gpredict sur le RPi, ce qui en fait sans doute un projet pour utilisateur expérimenté.

magpi.cc/BqPWLm





▲ Pour construire un répéteur APRS IGate, il vous faudra un RPi et une clé RTL-SDR.

APRS IGate

Le matériel radioamateur n'offre, par nature, que peu de puissance d'émission. Les ondes radio sont en outre dégradées par diverses interactions avec l'environnement. D'où l'intérêt des répéteurs numériques à APRS (« système de transmission automatique par paquets »). Celui-ci effectue une répétition numérique des paquets via une passerelle Internet (IGate), d'où une plus grande portée. Il sert aussi à suivre ses propres émissions et celles des radioamateurs locaux.

Le matériel et les logiciels sont identiques à ceux nécessaires à la construction d'une radio logicielle, mais le traitement des données diffère. **(VF : Hervé Moreau)**

magpi.cc/yppPRB

YogAI

Ce miroir magique met en vedette un yogi virtuel qui vous guidera dans la pratique des postures. **Phil King** s'échauffe.

▲ YogAI est capable d'en détecter 14, notamment les trois postures du guerrier.

MAKERS

Salma Mayorquin et Terry Rodriguez

Spécialistes de l'apprentissage automatique, développeurs, bidouilleurs, passionnés de sécurité, intéressés par l'exploration d'applications concrètes de l'IA.

smellslikeml.com

Ces dernières années, nous avons vu défiler de nombreux « miroirs magiques » comme celui qui abrite un écran TV pour afficher de précieuses infos. Avec son concept de coach virtuel, le miroir YogAI (magpi.cc/nQTDgX) propose néanmoins un niveau d'interactivité sans précédent pour guider et corriger vos postures.

Salma Mayorquin et Terry Rodriguez, mordus de yoga, ont pensé que la présence d'un entraîneur personnel leur permettrait de suivre leurs progrès. Alors, pourquoi ne pas inventer un entraîneur virtuel ? « Nous avons imaginé que des modèles [d'apprentissage profond] comme l'estimation de la posture nous permettraient de créer un programme susceptible de suivre notre façon de pratiquer. Voilà notre point de départ », nous confie Terry.

YogAI conduit le pratiquant au travers d'une séquence de postures de yoga en lui proposant des conseils lorsque la caméra détecte qu'il n'est pas dans la bonne position. YogAI se sert de l'estimation de la posture pour trouver des points clés sur le corps. Ceci permet de saisir et de classer les postures de yoga communes.

> « Un RPi 3 interprète en temps réel les images capturées par la caméra, en détectant les points clés du corps pour afficher et catégoriser la posture. »

L'utilisateur interagit avec YogAI de deux façons : visuellement grâce à l'écran miroir et vocalement grâce à l'assistant vocal Snips AIR (**snips.ai**). Ce dernier permet de recevoir des commandes vocales pour lancer, arrêter, interrompre et redémarrer une séance de yoga. YogAI s'exprime au travers du synthétiseur vocal Flite pour guider l'élève vers la bonne posture.

Comme nos makers sont expérimentés, une semaine leur a suffi pour fabriquer un prototype de miroir magique. Par contre, l'entraînement de l'IA à reconnaître les postures de yoga en temps réel a été plus complexe. « Nos modèles de vision par ordinateur doivent tourner rapidement pour avoir une résolution temporelle suffisante pour identifier le mouvement », indique Terry.

Prendre la posture

Un RPi 3 interprète en temps réel les images capturées par la caméra, en détectant les points clés du corps pour afficher la posture sur le miroir et la classer à l'aide d'un modèle d'apprentissage profond entraîné avec un jeu d'environ 35 000 échantillons.

Cependant, le couple a constaté qu'avec le RPi, l'inférence à partir d'images ne pouvait dépasser une image toutes les 4 à 5 s, ce qui se traduisait par un décalage temporel. Ils ont rapidement contourné le problème : « En réduisant nos modèles d'estimation de posture à l'aide de TensorFlow Lite, la fréquence d'images est passée de 0,2 à 2,5 ips », explique Salma. « Pour accélérer l'inférence, nous devrons réduire davantage le modèle. Nous pensons également que le passage au CM3 augmentera les performances de manière significative. »

Le niveau de précision est d'environ 80 % pour une douzaine de postures courantes. Il n'est pas surprenant de constater que les confusions proviennent de variantes de postures similaires, par ex. celles du guerrier. La classification est moins bonne lorsque la tête/le visage est caché, comme c'est le cas dans la posture du chien tête en bas.

La caméra fournit les images au modèle IA pour la détection des postures.

La commande vocale permet à l'utilisateur de conduire la séance de yoga.

Le miroir intelligent affiche alors une « carte thermique » de la posture ainsi que le nom de cette dernière.

En bref

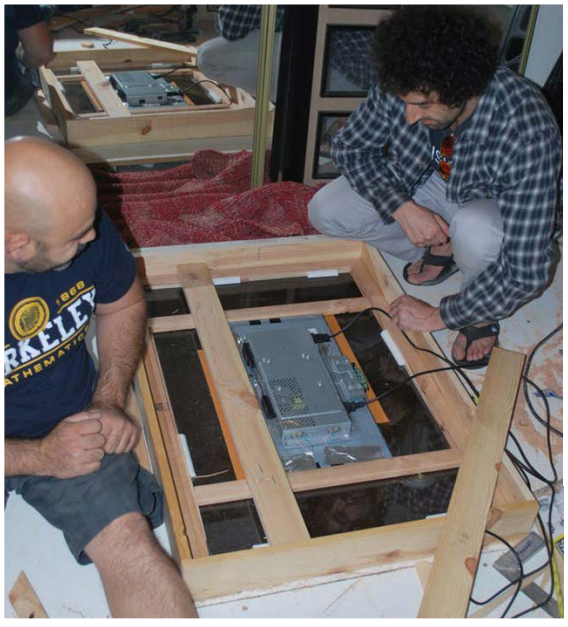
- Actuellement, YogAI reconnaît 14 postures de yoga.
- La partie entraînement aux postures de l'IA est assurée par TensorFlow Lite.
- Le logiciel est disponible sur **GitHub : magpi.cc/emKnDd**.
- Terry et Salma expérimentent l'ajout d'un avatar réaliste.
- Autres projets RPi : dispositif portable susceptible de prévenir l'arrivée de convulsions (**magpi.cc/OsMEve**).

Exercices plus intenses

En plus du yoga, Salma et Terry prévoient d'adapter YogAI pour surveiller des séances d'entraînement plus énergiques. « La musculation nous intéresse, certains nous suggèrent la danse, les katas de karaté », précise Terry. « YogAI peut accomplir des tâches plus générales en matière de santé et de bien-être personnel. »

« Nous visons l'intégration aux appareils de soins de santé portés sur soi », ajoute Salma. « Une montre intelligente dotée d'un accéléromètre et d'un moniteur de fréquence cardiaque rapprocherait YogAI de notre vision d'un miroir yogi intelligent et d'une plateforme de bien-être personnel. » M

(VF : Pascal Duchesnes)



Ils se sont servis d'un miroir sans tain de 60×90 cm auquel ils ont ajouté un cadre de bois.

Clapper Raspberry Pi

Vos souhaits réalisés d'un claquement de mains ? **Nicola King** applaudit l'ingénieuse créatrice.

MAKER

Ash Puckett

Ash est rédactrice technique indépendante, originaire des Appalaches (région est des États-Unis). Quand elle ne s'occupe pas d'illustrations, Ash est plongée dans un projet RPi impressionnant qu'elle ne manquera pas de partager.

ashpuckett.com

Ash Puckett se souvient du jour où elle s'est dit qu'il était temps de se simplifier la vie : « Un matin, je suis sortie du lit en claquant des mains pour allumer la lumière. Mais voilà, ça n'a pas marché. C'est alors qu'est née l'idée de créer un *Clapper Raspberry Pi.* »

Comme on n'est jamais si bien servi que par soi-même, Ash s'est procuré un RPi et un microphone USB pour créer quelque chose de simple, de chic et de pratique à la fois. Ce qui est surprenant, c'est que le projet a été conçu et réalisé en une journée avant d'être converti en tutoriel en ligne (**magpi.cc/CcMzdP**).

Le défi consistait à réaliser un guide flexible. Elle voulait créer un point de départ pour tous ceux qui ont un projet similaire en tête. Qu'il s'agisse de claquer des mains pour lancer une musique, ouvrir ou fermer une porte de garage, le principe reste le même. Son intention était donc de trouver cette fameuse base commune.

Vos claquements de mains sont des ordres

Alors, comment ça marche exactement ? « Le RPi est programmé pour capter vos claquements au travers d'un microphone USB », explique Ash. « Ces claquements permettent de déclencher des événements personnalisés. En principe, vos claquements de mains peuvent commander presque tout ce qui peut être piloté grâce au langage Python. »

> « Je n'avais aucun doute quant aux capacités du RPi. L'expérience montre que les broches GPIO permettent de créer des projets géniaux. »

Les possibilités sont quasiment illimitées. « Mes claquements m'ont permis de piloter une LED, de lancer des mélodies *funky* et quelques scripts personnalisés. La faisabilité ayant été démontrée, le reste repose sur l'imagination de chacun », ajoute Ash. « Si le pilotage est possible au travers d'un RPi, alors rien ne peut arrêter vos mains. Ceci englobe la domotique qui se sert de relais, la commande d'appareils électroménagers intelligents et (bien sûr) l'éclairage. »

Ash s'est servi du RPi parce qu'elle « n'avait aucun doute quant aux capacités du RPi. L'expérience montre que les broches GPIO permettent de créer des projets géniaux. Je voulais utiliser quelque chose qui soit capable de gérer un projet complexe, si besoin est. […] Le RPi est parfait pour ce genre d'entreprise créative. »

Une idée aurait-elle germé ? Depuis, la communauté RPi ne cesse de lui envoyer des commentaires très positifs : « Quel plaisir ! Certains me confient leurs projets de claquements de mains. D'autres me proposent leurs propres modifications, me recommandent du matériel spécifique, m'offrent de nouvelles pistes. »

Et Ash d'ajouter : « Beaucoup ont à l'esprit la technologie activée par le son. Ce n'est pas surprenant puisque presque tout le monde a un assistant à commande vocale dans sa poche. Comme les gens réalisent qu'ils peuvent utiliser la commande par le son dans leurs projets RPi, on devrait assister à une montée en flèche de ce genre d'idées. *Clapper* n'est qu'un maillon de cette chaîne sans fin. »

L'avenir est entre vos mains

Les projets d'Ash sont légion. « J'ai toute une liste de projets RPi prêts à démarrer : du système de sécurité pour habitation à la serre automatisée. Tous ces projets se serviront du RPi comme système de commande centralisé, avec accès et pilotage à distance. Je vous invite donc à me suivre sur howchoo (**magpi.cc/Demoez**) pour découvrir mes derniers projets RPi. Et vous pouvez être sûr que si je peux utiliser mes claquements de main, je ne m'en priverai pas. »

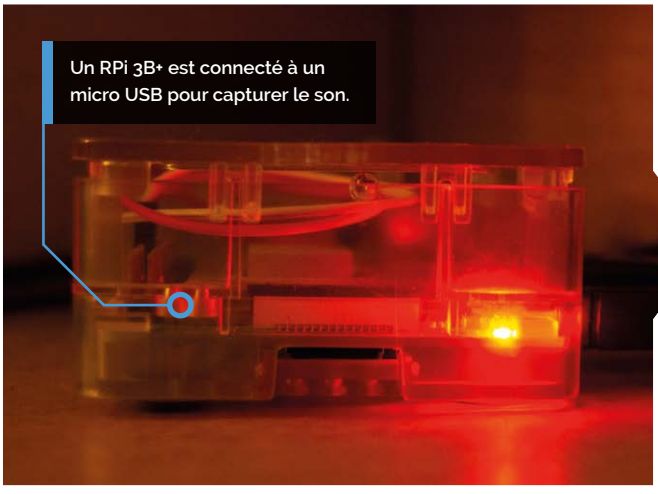
Il est clair que la simplicité du Clapper invite à l'action. Il n'est pas trop tard pour vous y mettre. Découvrez la magie de vos claquements de mains !

(VF : Pascal Duchesnes)

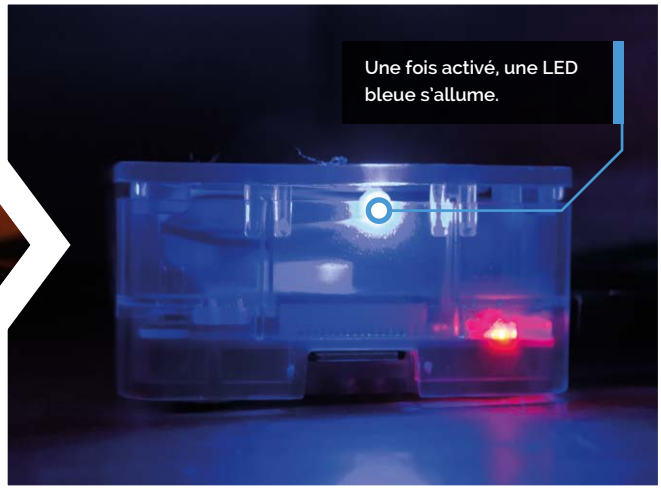
L'appareil se déclenche sur un simple claquement de mains.

En **bref**

- Les claquements sont détectés par le script pi-clap de Nikz Jon - **magpi.cc/GDpKhv**.
- Le tutoriel d'Ash est disponible sur **magpi.cc/CcMzdP**.
- La bibliothèque PyAudio permet de détecter le son capté par le micro USB.
- Donnez libre cours à votre imagination pour que vos claquements de mains deviennent des ordres !
- Parmi les autres projets RPi conçus par Ash, citons un réveil intelligent et un dispositif de diffusion de musique en continu.



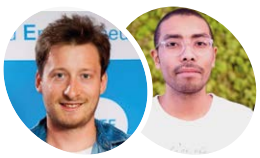
Un RPi 3B+ est connecté à un micro USB pour capturer le son.



Une fois activé, une LED bleue s'allume.

Votre barista personnel

Avez-vous déjà rêvé d'avoir votre propre barista, qui vous serve votre café favori à la demande ? Avec Snips, c'est possible. Demandez la recette à **Koen Vervloesem**.



MAKERS

Valentin et Bjay

Valentin est chef de produit chez Snips et maker pendant son temps libre. Quant à Bjay, il est développeur de logiciel et hacker en résidence. Ils sont passionnés par les interfaces vocales, et ils hackent divers appareils domestiques pour en faire des objets commandés par la voix. C'est la combinaison de leurs talents qui a permis la réalisation de ce projet.

snips.ai

Lorsque l'on achète un nouvel appareil, il faut d'abord se farcir le manuel de l'utilisateur, souvent volumineux, avant de pouvoir s'en servir. Avec une commande vocale, on n'en a plus besoin, la technologie passe à l'arrière-plan. Comme le dit si bien Valentin, chef de produit dans l'entreprise française : « C'est en fait le but que nous poursuivons chez Snips, camoufler la technologie. Grâce aux appareils commandés par la voix, on peut à nouveau se permettre de passer de vrais moments de qualité avec des êtres chers ».

Pour illustrer cette technologie, Valentin et Bjay, hacker en résidence chez Snips, ont conçu une interface vocale pour une machine à café. Vous souhaitez un double expresso, un *flat white* ou de l'eau chaude pour votre thé ? Il suffit de le demander à la machine.

Transformation de la machine à café

Valentin et Bjay ont démonté une De'Longhi ECAM 22.140.B pour mener à bien leur projet. Ils ont foré cinq petits trous dans le circuit imprimé avec les boutons-poussoirs qui est derrière le panneau avant, et ils y ont raccordé des fils pour remplacer la commande manuelle par une commande logicielle. Pour le réglage de l'intensité du café, c'était plus difficile, il se fait à l'aide d'un codeur rotatif. Ils ont dû monter un moteur pas à pas sur le codeur à l'aide d'une pièce conçue par eux et réalisée avec une imprimante 3D.

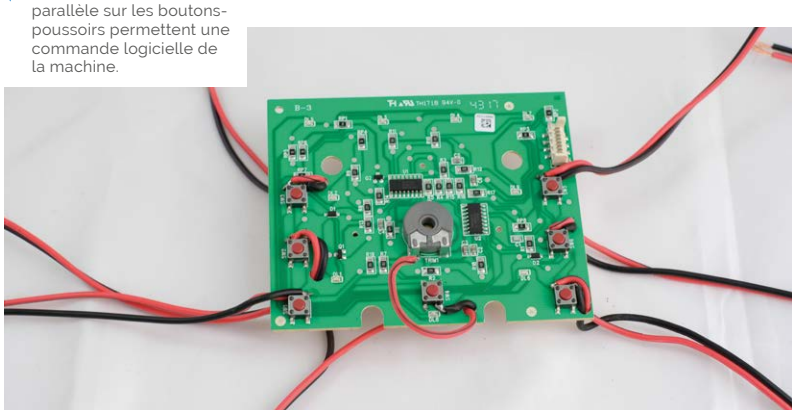
« Machine, deux cappuccinos bien forts, s'il te plait ! »

Le logiciel *Snips Flow* tourne sur un Raspberry Pi ; via une carte d'extension ReSpeaker 2-Mics, il comprend votre voix, interprète votre demande, et la code en Python pour commander un Arduino Nano. Pour l'obtention du breuvage désiré, ce dernier commande le moteur pas à pas via une carte de commande spécifique, et les fonctions normalement activées par les boutons-poussoirs via une carte équipée de relais.

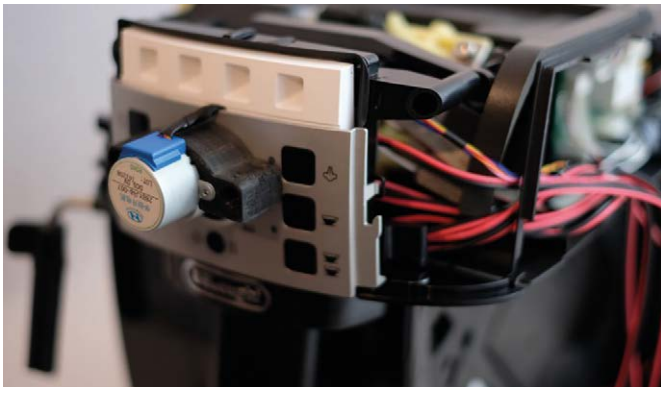
Ce projet de machine à café à commande vocale était un véritable challenge, nous confie Valentin : « D'un point de vue matériel, le plus dur était de trouver les bons endroits où raccorder des fils sur le circuit imprimé de la machine, et ce sans causer de court-circuit. Il n'y avait pas beaucoup de place et c'était délicat ».

L'écriture du logiciel n'était pas une sinécure

Les fils soudés en parallèle sur les boutons-poussoirs permettent une commande logicielle de la machine.



Valentin et Bjay ont monté un moteur pas à pas sur le panneau avant pour pouvoir régler l'intensité du café.



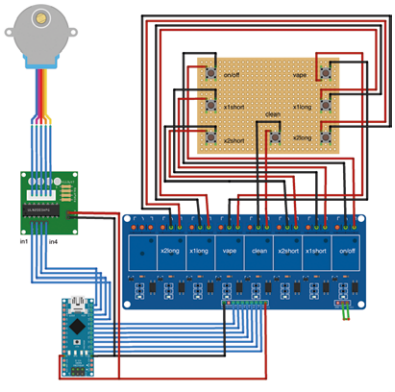
▲ « Machine, deux cappuccinos bien forts, s'il te plait ! ». C'est tout de même plus sympa que de devoir chercher sur quels boutons appuyer, non ?

◀ Avec le kit de base d'interaction vocale Snips, on peut doter sa machine à café d'une interface vocale.

non plus, car au début du projet la bibliothèque Hermes-Python, qui permet l'interface avec la plateforme Snips, était toujours en cours de développement. Valentin explique : « La manière dont on utilise la bibliothèque a évolué, ce qui a causé des crashs de notre code. Hermes-Python est désormais stable, et notre De'Longhi aussi ! ».

Dans le monde entier

Les représentants de Snips exposent leur machine à café à commande vocale lors de salons ou de rencontres avec des clients, et Valentin de renchérir : « Elle a fait le tour du monde, du CES de Las Vegas à l'IFA de Berlin, en passant par Barcelone pour le Mobile World Congress. On pourrait aussi l'utiliser dans nos bureaux, mais il y a quatre-vingts personnes, et il faudrait hacker une plus grosse machine ».



▲ Le schéma de l'électronique de commande de la machine par l'Arduino Nano.

> « Avec cet appareil, personne en dehors de chez vous ne connaîtra l'étendue de votre addiction à la caféine. »

Valentin et Bjay continuent d'améliorer la machine. Pour le matériel, il n'y a que la carte du microphone qui a changé : « Nous utilisons maintenant un microcontrôleur NXP RT1050 ou un micro sans fil de DSP Group ». Les possibilités de l'assistant vocal ont quant à elles été étendues : « On peut demander à l'appareil de procéder à son entretien, de produire de la vapeur, ou d'arrêter l'opération en cours ».

Ce qui n'a pas changé, c'est la détermination de Snips à respecter la vie privée : en dehors de chez vous, personne ne connaîtra l'étendue de votre addiction à la caféine… M

(VF : Jean-Louis Mehren)

En bref

- Il s'agit d'une machine à café commandée par la voix, qui comprend quel café vous souhaitez et comment vous le voulez.
- Plus besoin d'appuyer sur des boutons ou de regarder un écran.
- Demandez une ou plusieurs tasses, petites ou grandes, d'un café fort ou allongé ; tout comme vous le feriez avec un barista.
- Les éléments principaux de l'appareil sont un Raspberry Pi avec Snips Flow et un Arduino Nano.
- Il n'aura fallu qu'un peu moins de trois semaines à Valentin et Bjay pour concrétiser leur projet de machine à café à commande vocale.
- Il s'agissait de transformer un appareil domestique ordinaire en un objet commandé facilement par la voix, tout en insistant sur le respect de la vie privée.

Hackez votre machine à café !

Si vous voulez hacker votre machine à café pour la commander à la voix, lisez d'abord notre test du kit de base d'interaction vocale Snips à la page 88, puis suivez les explications de Valentin et Bjay sur **bit.ly/2MWqAvo.**

MAKER

Sean Glendinning

Ce lycéen de 16 ans, passionné de physique, mathématique et programmation vit à Aberdeenshire (nord-est de l'Écosse)

Le robot de Sean

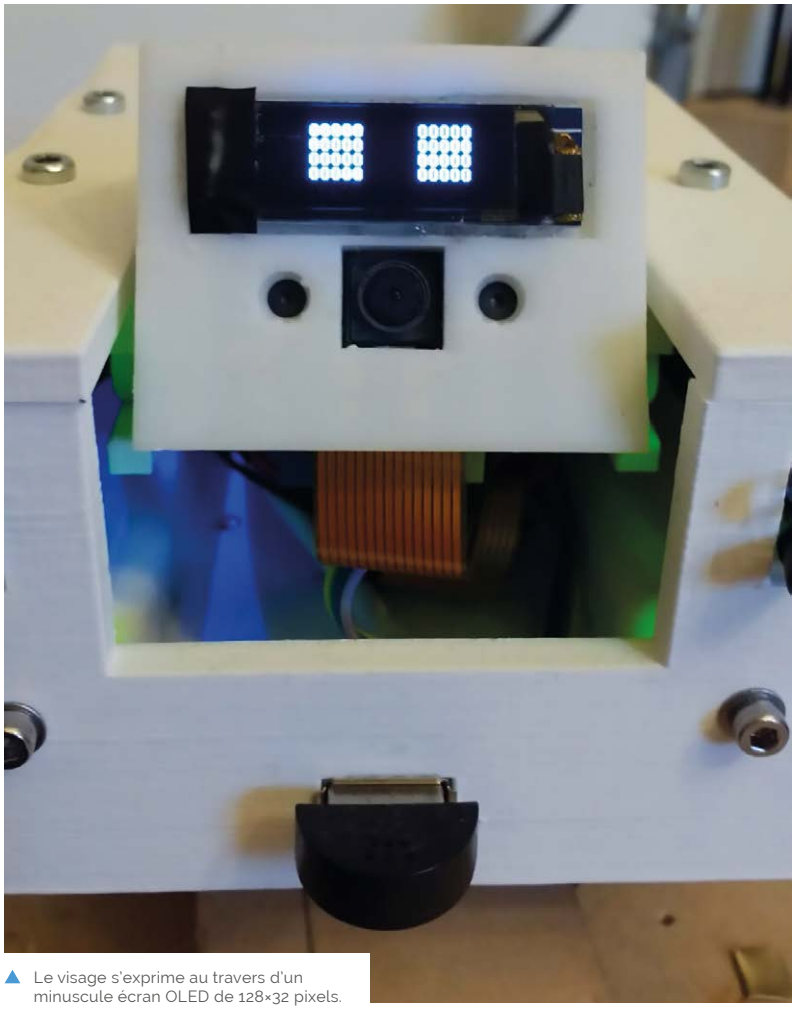
Ce joyeux robot à commande vocale est capable de détecter les bords de table ainsi que les visages. **Rob Zwetsloot** le regarde droit dans les yeux.

Les robots avec un visage sont loin d'être séduisants. Le résultat est parfois terrifiant. Heureusement, le robot de Sean Glendinning est mignon, ce qui est appréciable puisqu'il risque de vous regarder droit dans les yeux pour effectuer une reconnaissance faciale.

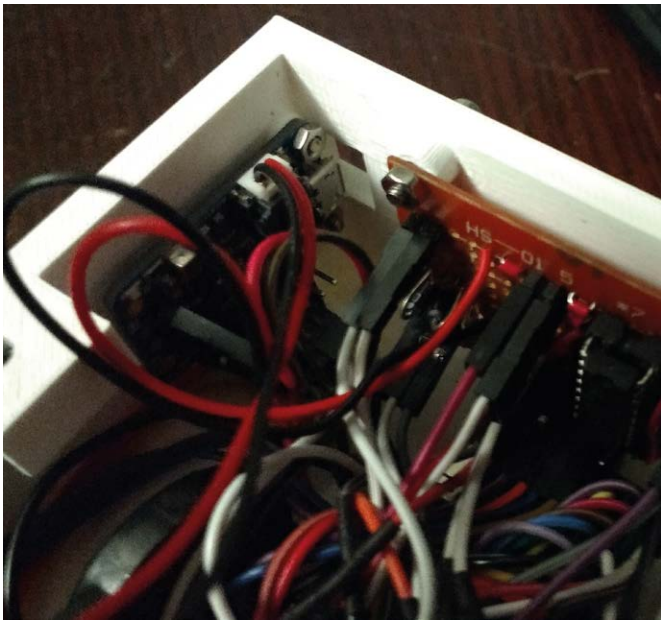
« Pour mon projet Gold CREST Award, j'ai décidé de concevoir et de construire un robot piloté par RPi », raconte Sean. « Mon robot est capable de détecter des bords de table et des visages. Il répond même à certaines commandes vocales. Il se déplace grâce à des chenilles et s'il détecte un visage, il réagit en inclinant la tête. »

Le programme national CREST (**crestawards.org**) a pour objet d'attirer les jeunes vers la science, la technologie, l'ingénierie et les mathématiques (STEM) en leur offrant des projets plus tangibles, plus pratiques qu'à l'école. Sean a choisi un projet Gold qui doit être réalisé en 70 h ou plus.

« Ce n'était pas évident. Néanmoins j'ai pris beaucoup de plaisir à réaliser ce projet et j'ai appris énormément de choses », poursuit Sean. « J'ai conçu moi-même le châssis en 3D, en mettant en pratique les compétences acquises au lycée en matière de graphisme. Je disposais déjà de la plupart des composants électroniques nécessaires. Sinon, j'ai essayé d'improviser et de réduire les coûts du mieux que je pouvais. J'ai dû acheter en ligne certains composants spéciaux, comme un microphone USB et le châssis imprimé en 3D. J'ai essayé de donner au robot ergonomie et personnalité ainsi que les moyens pour qu'il puisse se déplacer sans culbuter. »



▲ Le visage s'exprime au travers d'un minuscule écran OLED de 128×32 pixels.



Le robot lève les regards dès qu'il vous détecte pour établir un contact visuel.

Les capteurs de distance situés sous et devant le robot lui permettent de ne pas heurter de murs ou de culbuter.

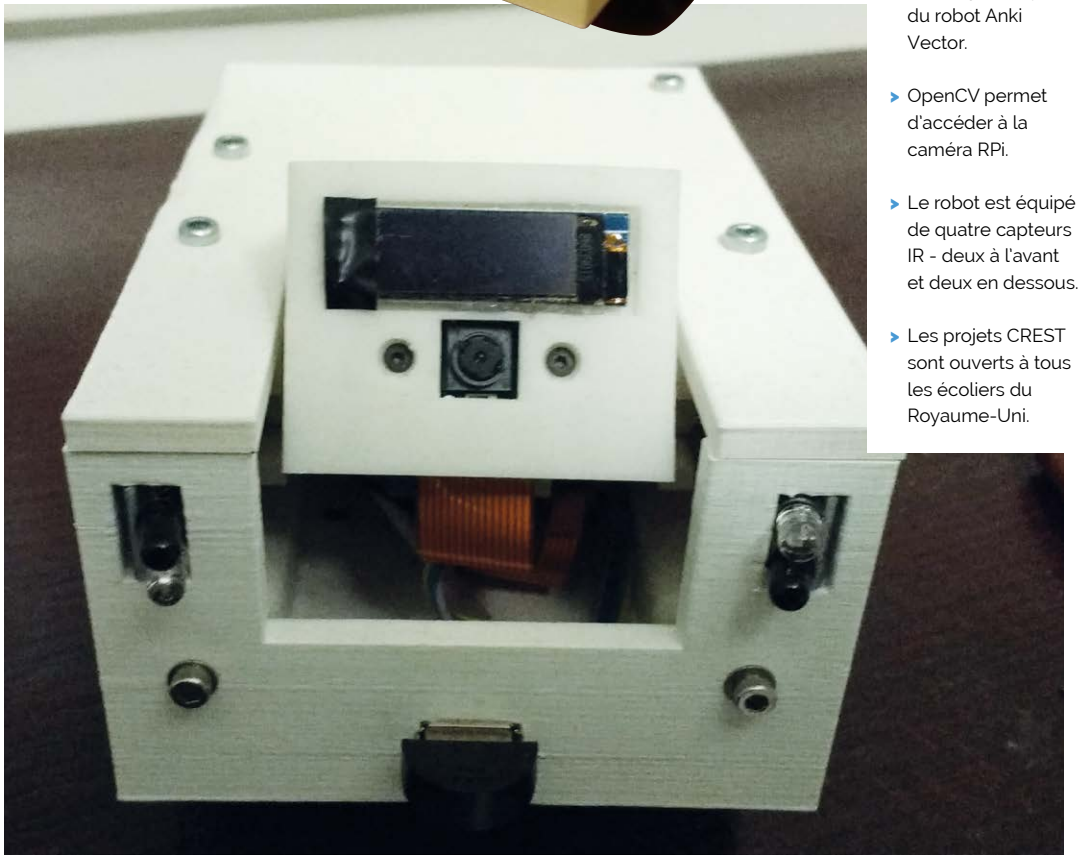
Entièrement personnalisé, le châssis du robot est une impression en 3D.

En bref

- Le RPi Zero W utilisé est celui offert avec l'abonnement au MagPi.
- Le *design* s'inspire du robot Anki Vector.
- OpenCV permet d'accéder à la caméra RPi.
- Le robot est équipé de quatre capteurs IR - deux à l'avant et deux en dessous.
- Les projets CREST sont ouverts à tous les écoliers du Royaume-Uni.

▶ Ce châssis cubique très pratique met en valeur les nombreuses fonctions.

▼ Le matériel est bien agencé à l'intérieur du châssis.



▲ Chaque composant a sa place.

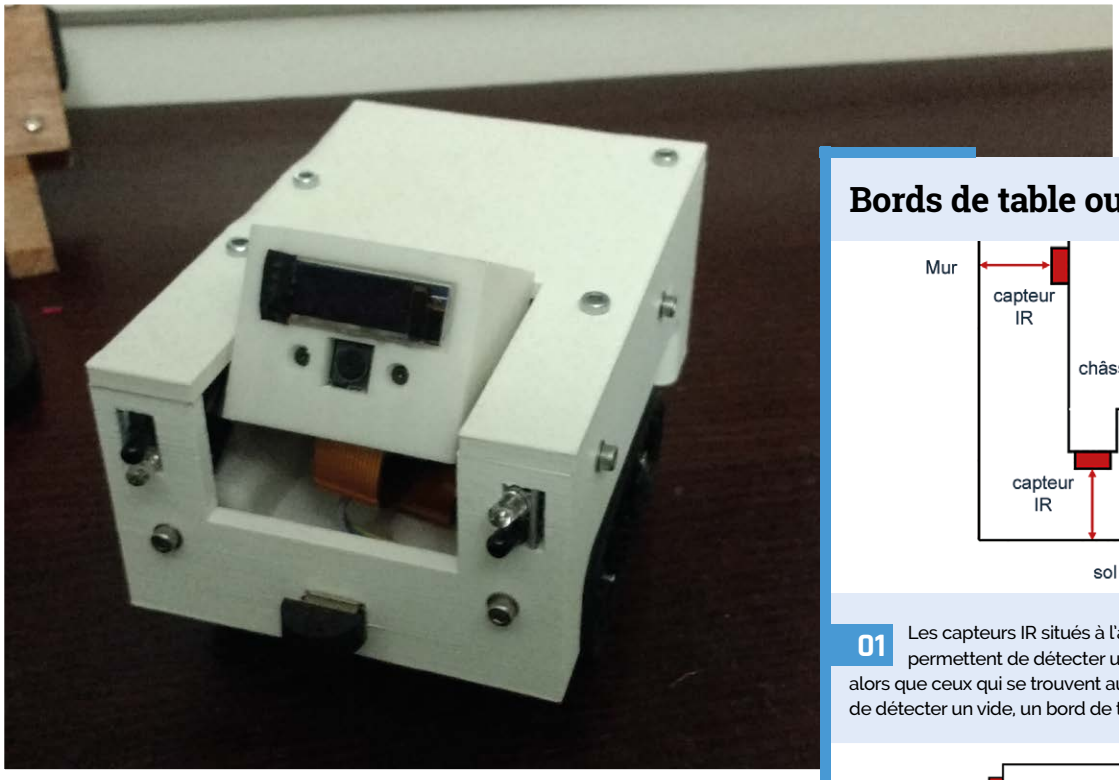
▲ L'habit fait le robot.

Des bras aux chenilles

Voilà des années que Sean construit des robots avec des bras robotiques beaucoup plus simples, comme ceux que l'on trouve dans les usines automobiles ou les kits comme MeArm Pi et PiArm.

« J'ai fabriqué mon premier bras en 2015 avec du contreplaqué et de la colle », se souvient Sean. « Pour le suivant, j'ai utilisé de l'acrylique découpé au laser. Comme ces bras étaient mal conçus, ils n'ont jamais fonctionné sur mes robots. En 2018, j'ai mis trois mois pour concevoir un troisième bras, fait de contreplaqué épais et de renforts en

▼ En plus d'une batterie au lithium, l'alimentation requiert aussi des piles AA.



La table est son habitat naturel.

aluminium. Ce bras a une meilleure saisie que les deux autres. »

Après s'être inspiré d'un robot jouet, Sean a décidé de construire un robot plus mobile en utilisant un RPi.

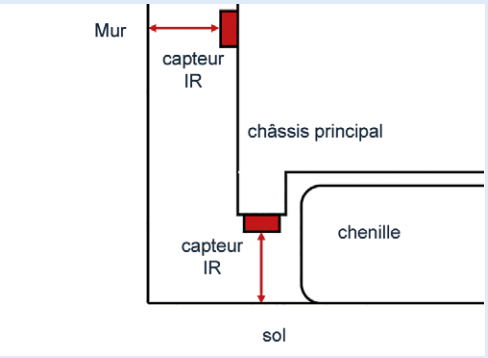
> « Je disposais déjà de la plupart des composants électroniques nécessaires. J'ai essayé d'improviser et de réduire les coûts du mieux que je pouvais. »

« Je me suis servi du RPi dans plusieurs projets. Je savais qu'un Arduino manquerait de puissance pour la reconnaissance vocale ou faciale », explique-t-il. « La taille, le coût et la puissance du processeur RPi contribuent à une gestion efficace des bibliothèques complexes liées à la reconnaissance faciale et vocale. »

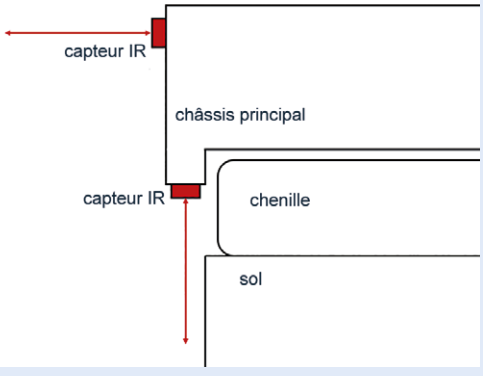
Le robot utilise des capteurs de distance IR disposés autour de son châssis pour détecter les murs et les bords de table ainsi qu'un module de caméra RPi pour reconnaître les visages. Le matériel est logé dans un boîtier imprimé en 3D. Excellente construction ! Nous sommes impatients de voir ce que les robots de Sean feront à l'avenir.

(VF : Pascal Duchesnes)

Bords de table ou mur ?



01 Les capteurs IR situés à l'avant du robot permettent de détecter une surface, un mur, alors que ceux qui se trouvent au-dessous permettent de détecter un vide, un bord de table.



02 Le connecteur GPIO du RPi surveille en permanence les capteurs tandis que le logiciel, lui, gère les résultats.

03 Le programme envoie des signaux au pilote des moteurs pour inverser leur sens de rotation ou les faire pivoter pour que le robot prenne une autre direction.

Moniteur Commodore

Cette petite réplique de moniteur a été créée pour les jeux sur Commodore 64, et la qualité n'est pas une question de taille, comme l'a découvert **David Crookes**.

MAKER

Chris Mills

Chris se passionne pour l'informatique depuis la fin des années 1970. C'est en 1985 qu'il a acquis un Commodore 128, et il apprécie tout autant de se plonger dans la technique que d'utiliser une machine pour son plaisir.

magpi.cc/VKPkzy

Au fil des années, le Raspberry Pi a confirmé son statut de favori parmi les passionnés de rétrogaming par ses aptitudes à émuler des ordinateurs et des consoles classiques, venus d'une époque révolue. Beaucoup de ces enthousiastes utilisent des logiciels comme *RetroPie* pour créer des machines qui font tourner des jeux de différents systèmes. Généralement, ces makers connectent leurs consoles sur des téléviseurs à écran large.

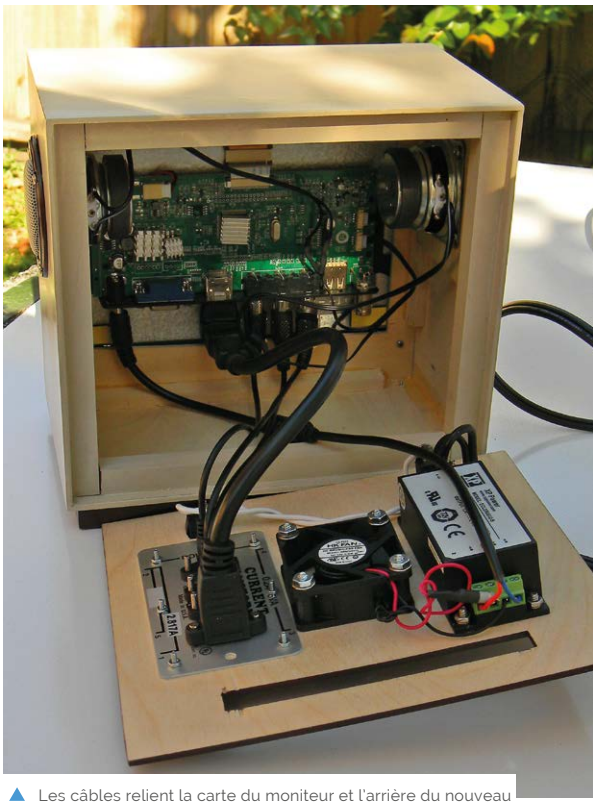
Pourtant, lorsque Chris Mills décide de réaliser une émulation de Commodore 64, ses ambitions sont plus modestes. Inspiré par le lancement récent de la console miniature THEC64 Mini, il s'engage dans la réalisation d'une version amincie du vénérable moniteur 1702 de Commodore. C'est du moins le résultat auquel il est finalement parvenu. « À l'origine, je voulais réaliser un petit caisson pour installer le moniteur et un RPi exclusivement destiné à émuler la console Commodore », précise le maker. « Je n'avais pas vraiment prévu d'en faire une version aussi élaborée ».

Une petite merveille

Chris apprécie tellement le RPi que, malgré l'achat d'une THEC64 Mini pour ses qualités *plug-and-play*, il préfère utiliser le RPi pour ses jeux Commodore 64. Le RPi lui permet de faire tourner beaucoup plus de logiciels, essentiellement grâce à l'émulateur Combian 64, une distribution accessible grâce à une appli baptisée Vice.

« Cette version sert exclusivement à l'émulation Commodore et l'amorçage permet d'obtenir, en quelques secondes, le fameux écran bleu Commodore. J'ai l'impression d'être plus proche des réactions du vrai matériel Commodore avec des commandes moins lentes », ajoute Chris. Connecté au moniteur, la console mini-C64 à ses côtés et une manette de jeu en plus, le système a fière allure, du moins en apparence.

Non, il ne s'agit pas d'une manette de jeu géante : sa taille est normale, mais comparativement, le moniteur est de dimensions modestes. Au premier plan, la console THEC64 Mini.



Les câbles relient la carte du moniteur et l'arrière du nouveau boîtier pour faciliter l'accès aux connecteurs. Les composants sont refroidis par un ventilateur.

> « Vous n'imaginez pas le vacarme que fait un ventilateur de 50 mm dit *silencieux* dans un boîtier de cette taille. »

Moniteur de surveillance

Pour soigner le look, Chris a choisi un petit moniteur de surveillance TFT à cristaux liquides. « J'avais besoin d'un rapport hauteur/largeur de

Le Commodore 64 était connu pour sa puce musicale SID qui méritait une qualité optimale de restitution pour l'écoute. Chris peut aussi ajouter un jack pour casque s'il a besoin de se connecter à un système sonore plus puissant.

Grâce à l'émulateur Combian 64, il n'y a rien de plus à faire que de configurer l'écran et les paramètres d'émulation.

Chris aimerait utiliser les broches GPIO du RPi pour allumer une LED lors des transferts de données, mais il est ravi que son ensemble RPi/moniteur lui permette de prendre du bon temps avec ses jeux préférés.

4:3 puisque l'écran devait accepter une résolution de 320×200 », poursuit Chris. « Je souhaitais aussi pouvoir installer le tout sur mon bureau, sans que cela prenne trop de place. Je n'imaginais pas afficher des jeux Commodore 64 sur un téléviseur de 32". J'ai grandi avec un écran à tube cathodique de 13" pour mes ordinateurs, et pour moi, une diagonale de 8" était un bon compromis entre encombrement et lisibilité ».

Le moniteur HD est installé dans un boîtier en bois, conçu et réalisé par Chris lui-même. Le dos de l'écran retiré, il a connecté le câblage à l'arrière du nouveau moniteur rétro et raccordé deux haut-parleurs de 2", non sans mal.
Jusqu'à ce qu'il essaie de monter les haut-parleurs dans le caisson, il n'avait pas réalisé que leurs trous de fixation et ceux du panneau n'étaient pas alignés. Chris a également inséré une résistance en série pour abaisser de 12 V à 4 V la tension d'alimentation du ventilateur. « Vous n'imaginez pas le vacarme que fait un ventilateur de 50 mm dit *silencieux* dans un boîtier de cette taille », précise-t-il dans un éclat de rire. « Je crois que pendant toutes ces années, mon appréciation du mot *silence* était erronée ».

Pour la finition, le boîtier a été poncé avec beaucoup de soin, puis recouvert d'une laque Krylon en aérosol, avec ponçage à l'eau entre les couches à l'aide de papier abrasif (grain 2000). Chris a ensuite pris une photo d'un insigne Commodore sur l'un de ses moniteurs 1702 originaux. Il a recadré l'image, puis créé un logo Commodore d'aspect métallisé, apposé ensuite sur la face avant du mini-moniteur. « Les réactions des fans de Commodore ont été enthousiastes », conclut-il. **(VF : Pascal Godart)**

En bref

- Le coût total du projet est de 120 $.
- Sa construction a nécessité 25 h.
- Le boîtier est constitué de bois peint.
- Le RPi utilise Combian 64, l'émulateur C64.
- Pour le moment, le RPi est installé à l'extérieur du moniteur.

Quel que soit l'angle de vue, le moniteur a un aspect vraiment professionnel, et une belle finition dans le plus pur style rétro.

MAKER

Joe Birch

Joe est ingénieur, spécialiste du système Android. Il travaille chez Buffer, à Brighton (Royaume-Uni). Il est également *Google Developers Expert* (GDE) sur Android, Google Pay et Flutter.

chordassist.com

Chord Assist

Avec son assistant vocal et son système d'intelligence artificielle intégré, cette guitare intelligente et accessible peut aider tout un chacun à apprendre à jouer. **Phil King** lui-même s'est mis à la gratte…

Apprendre à jouer de la guitare peut être difficile si vous êtes atteint(e) de déficience visuelle ou auditive. La guitare associée à l'appli Chord Assist de Joe Birch apporte justement les moyens de simplifier l'apprentissage de l'instrument pour les personnes muettes, ou atteintes de déficience auditive ou visuelle.

La famille de Joe est porteuse d'une maladie génétique appelée rétinite pigmentaire qui provoque, au fil du temps, un rétrécissement visuel. Sa mère atteinte est aujourd'hui reconnue comme malvoyante. Joe a été sensibilisé par sa proximité avec des proches atteints de cette maladie, en constatant ses effets sur leur vie. Pour lui, la musique n'est pas réellement accessible à tous. L'appli Chord Assist est née de son envie d'améliorer cela.

> « Pratiquer des ouvertures dans la guitare, puis installer toutes les pièces à l'intérieur a été assez fastidieux. »

Outre l'écran LCD, l'afficheur à quatre chiffres et les boutons pour présenter et choisir les accords, la guitare acoustique modifiée de Joe dispose d'un lecteur Braille, dérivé de sa précédente liseuse d'articles BrailleBox (**magpi.cc/LpjrGF**). Il y a même, à proximité du lecteur, un indicateur de progression à vibreur pour signaler une demande.

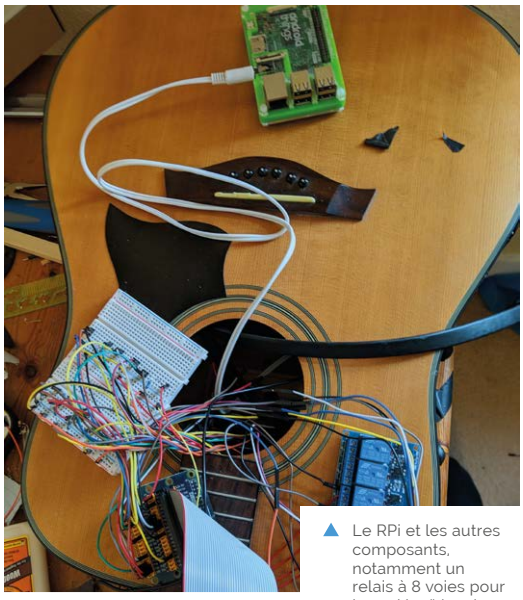
De plus, l'utilisateur peut faire une demande vocale d'accord grâce au micro intégré et à l'assistant vocal Google, et recevoir une réponse sur le haut-parleur. Capable de jouer aussi une note à la demande pour accorder l'instrument, le système utilise l'appli Chord Assist (**magpi.cc/xUAQNo**) depuis la plateforme *Actions on Google*, qui peut aussi servir d'assistant autonome, accessible avec un smartphone ou d'autres appareils.

Un écran LCD permet d'afficher les tablatures ; il y a aussi un lecteur Braille de l'autre côté de l'instrument.

Le musicien en herbe peut parler à son instrument et recevoir des réponses vocales grâce au haut-parleur incorporé.

Le Raspberry Pi et les autres composants électroniques sont cachés à l'intérieur de la guitare, sans affecter le son outre mesure.

Le lecteur Braille situé de l'autre côté de la guitare permet aux malvoyants de lire les instructions nécessaires pour jouer un accord.



Le RPi et les autres composants, notamment un relais à 8 voies pour les solénoïdes du lecteur Braille, sont confinés à l'intérieur de l'instrument.

Composants invisibles

Joe a consacré près de six mois à ce projet. Pratiquer des ouvertures dans la guitare, puis installer toutes les pièces à l'intérieur ont été les phases de construction les plus difficiles. À l'origine, tout était monté sur des cartes de prototypage, mais les composants se débranchaient en permanence. Joe a donc préféré tout souder sur des circuits imprimés, ce qui a bien amélioré les choses.

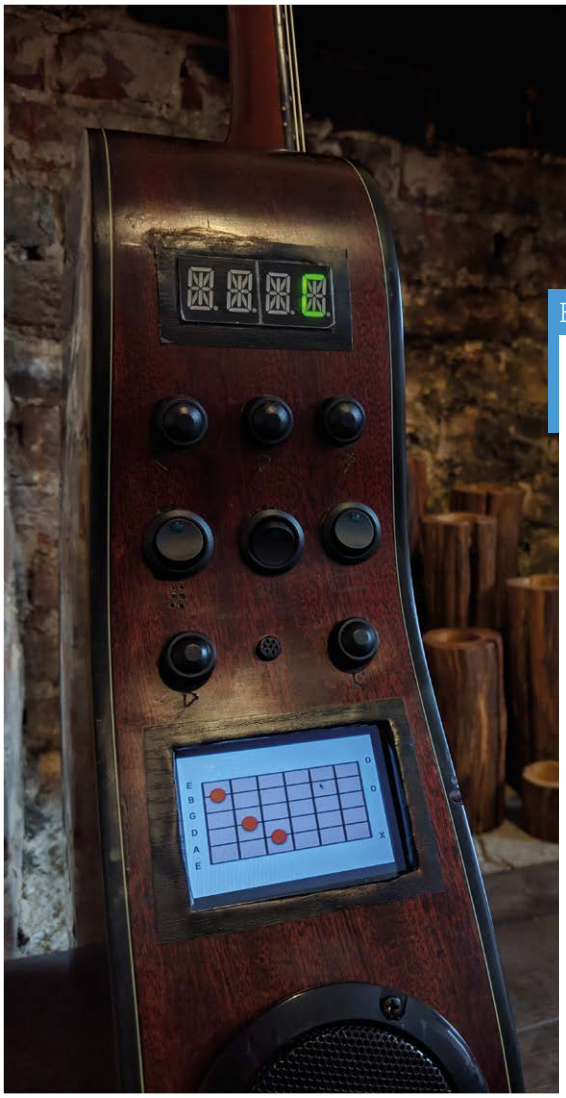
Tous les composants électroniques, y compris le RPi, sont logés dans la guitare. Même s'ils sont nombreux, ils sont légers, à l'exception de la batterie portable. Tout cela ne semble pas avoir d'impact négatif sur la qualité du son : « Au départ, je pensais que ce serait un gros problème, mais en comparant le son avec celui de mon autre guitare, je n'ai pas réellement constaté de différence ».

Monter le son

Joe est satisfait de sa réalisation, même s'il prévoit quelques évolutions. « J'aimerais ajouter un ampli et une commande de volume pour améliorer le fonctionnement du haut-parleur. Je prévois aussi d'incorporer une vraie fonction pour accorder la guitare, ajoutée par ex. dans le logiciel pour analyser la note jouée et indiquer au musicien s'il faut tendre ou relâcher une corde. Ce n'est pas encore possible avec l'assistant Google, mais ce serait très utile pour ceux qui n'ont pas l'habitude d'accorder un instrument. Je pourrais aussi utiliser l'écran pour apporter des indications au musicien et répondre à différents besoins d'accessibilité.

Envie de gratter ? Joe propose l'essentiel des éléments nécessaires en *open source*, sauf le programme qui fonctionne sur le RPi, chargé de piloter l'interface utilisateur et les broches GPIO pour accéder aux composants. Le schéma de câblage et le robot conversationnel sont l'un et l'autre *open source*, donc accessibles à tous. Suite aux commentaires des utilisateurs, Joe prévoit de créer d'autres guitares intelligentes. Ce projet a touché une corde sensible chez nous, et nous ne boudons pas notre plaisir !

(VF : Pascal Godart)



Parmi les nombreuses commandes et fonctions, figurent les boutons de sélection, un micro et un haut-parleur, ainsi qu'un afficheur à segments et un écran LCD.

En bref

- Chord Assist a été sélectionné pour les *Webby Awards* (qui distinguent les applications de qualité sur l'internet).
- Il est possible d'allumer ou d'éteindre séparément l'écran, le lecteur Braille et le haut-parleur.
- L'instrument possède un bouton pour répéter rapidement la dernière réponse vocale reçue.
- Un afficheur à segments à quatre chiffres montre les accords possibles.
- Au lieu de la voix, il est possible d'utiliser des boutons pour choisir un accord.

MAKER

Lucy Hattersley

Lucy est la rédactrice en chef de la version originale du MagPi. Elle aime la technologie (bidouiller et réparer), et construire des montages bizarres. Elle s'est agitée sur Samba tout le week-end !

magpi.cc

Bâtir un serveur de fichiers Samba

Se servir de Samba/CIFS pour partager un répertoire du Raspberry Pi avec d'autres ordinateurs sur un réseau

Il est facile d'utiliser un RPi comme serveur de fichiers Samba pour y enregistrer des sauvegardes et y partager des fichiers avec tous les autres ordinateurs de votre réseau.

Samba est l'implémentation sous Linux du standard de partage de fichiers SMB/CIFS utilisé par les PC Windows et les ordinateurs Apple, et généralement pris en charge par les serveurs multimédias, les consoles de jeu et les applis des mobiles.

Avec Samba activé, vous pouvez rapidement recopier des fichiers d'un ordinateur de votre réseau sur un RPi par Wi-Fi (ou connexion Ethernet directe).

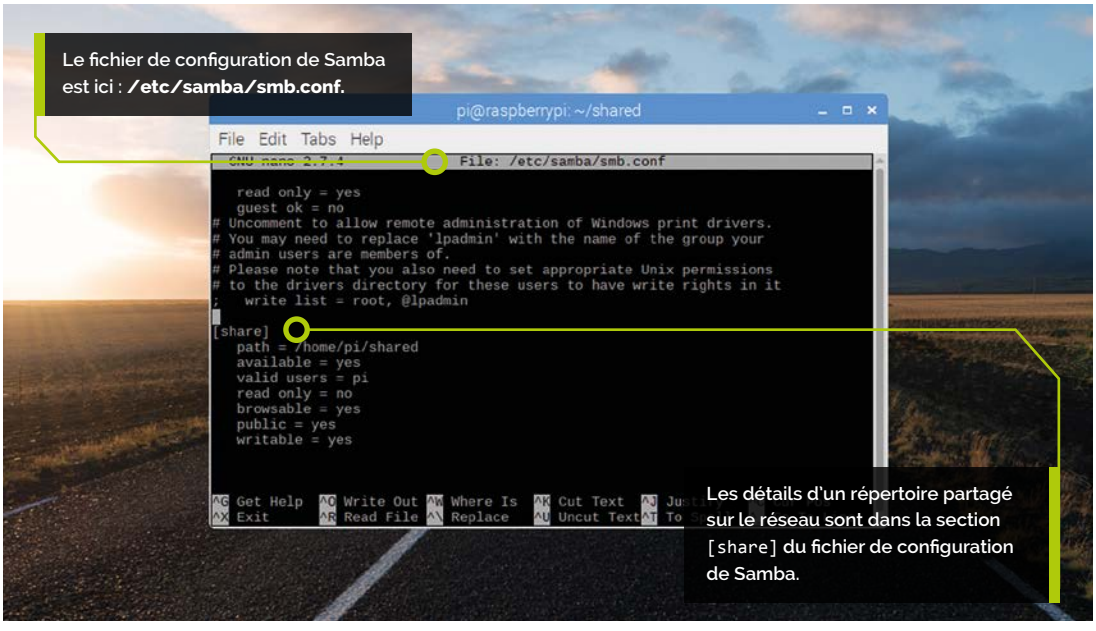
Pour ce tutoriel, vous devez disposer d'un clavier, d'une souris et d'un moniteur pour configurer votre serveur de fichiers ; vous pouvez aussi activer SSH (**magpi.cc/ssh**) et vous y connecter à distance depuis un autre ordinateur de votre réseau local.

Nous supposons aussi que vous avez une carte micro-SD de 32 Go (ou moins) avec un espace de stockage raisonnable sans nécessiter d'étape supplémentaire pour le rendre accessible. Toutefois, s'il vous faut plus d'espace, on peut monter un gros disque USB externe et lui créer une entrée Samba.

Autre solution pour garder un faible encombrement, installez Raspbian sur des cartes micro-SD jusqu'à 256 Go, bien que nous pensions plus sage d'installer Samba sur un disque externe et d'utiliser une carte micro-SD plus petite pour amorcer Raspbian et s'y connecter.

Avant d'acheter, nous vous conseillons de vérifier en ligne les cartes SD non prises en charge (**magpi.cc/2q97aGO**), pour être sûr d'en avoir une parfaitement compatible avec le RPi.

Une fois configuré, vous pouvez monter votre serveur de fichiers personnel sur tous les autres ordinateurs de votre réseau, et l'utiliser

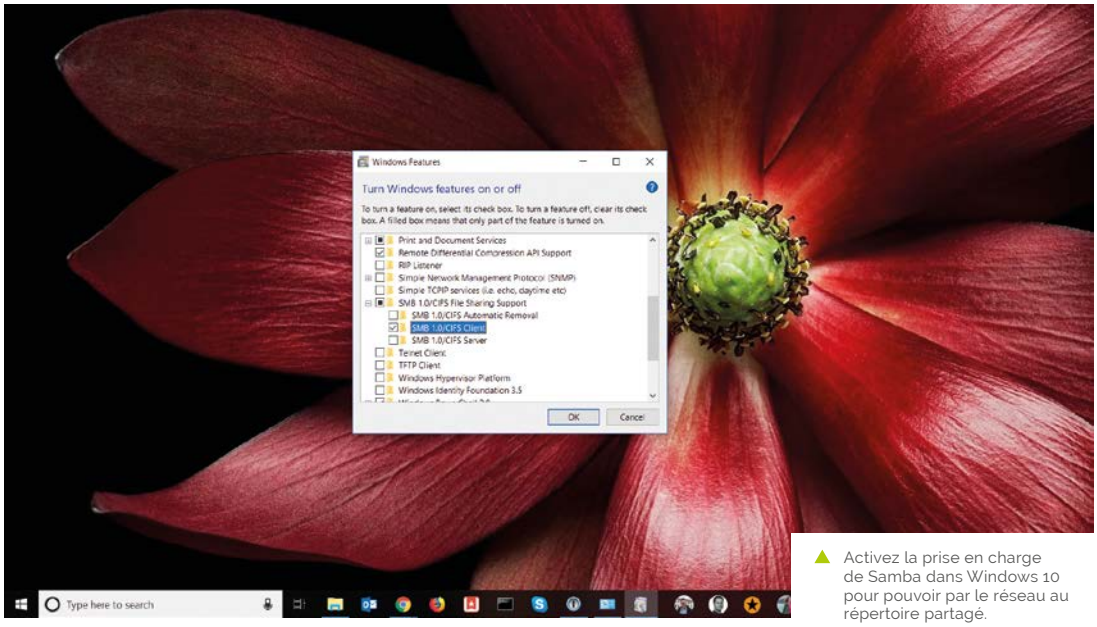


Le fichier de configuration de Samba est ici : **/etc/samba/smb.conf.**

Les détails d'un répertoire partagé sur le réseau sont dans la section `[share]` du fichier de configuration de Samba.

Ingrédients

- Raspberry Pi
- Carte micro-SD
- Clavier et souris pour la configuration
- Réseau et routeur (Ethernet recommandé)



Activez la prise en charge de Samba dans Windows 10 pour pouvoir par le réseau au répertoire partagé.

Astuce

Rapport d'état

`sudo service smdb status` vous permet de voir l'état du disque Samba. Et vous pouvez utiliser `smdb restart` pour arrêter et redémarrer le service.

comme emplacement pour stocker toutes sortes de fichiers : musique disponible pour vos colocataires, sauvegarde de documents importants, fichiers de jeu à partager entre ordinateurs.

Préférez une connexion filaire Ethernet pour sa stabilité et ses vitesses de transfert rapides. Le serveur continuera à fonctionner si vous connectez votre RPi par Wi-Fi, mais cela affectera les performances, en particulier lorsqu'il s'agit de recopier des gros fichiers.

01 Installer Raspbian

Téléchargez la dernière version de NOOBS (ou le fichier image du dernier Raspbian Stretch avec bureau) et installez-le sur une carte micro-SD. Quant à la configuration du RPi, si c'est votre première fois, voyez **magpi.cc/quickstart**. Insérez la carte micro-SD dans le RPi et connectez clavier, souris et câble HDMI. Il est préférable de relier le RPi au modem/routeur avec un câble Ethernet (sinon utilisez l'icône Wi-Fi pour vous connecter à un réseau sans-fil au démarrage). Branchez le câble d'alimentation pour démarrer le RPi.

02 Installer Samba

Samba est disponible dans les dépôts de logiciels standards de Raspbian. Nous sommes en train de mettre à jour notre index des dépôts, vérifiez que votre système d'exploitation est bien mis à jour, et installez Samba avec APT. Ouvrez un Terminal et tapez :

```
sudo apt update
sudo apt upgrade
sudo apt install samba samba-common-bin
```

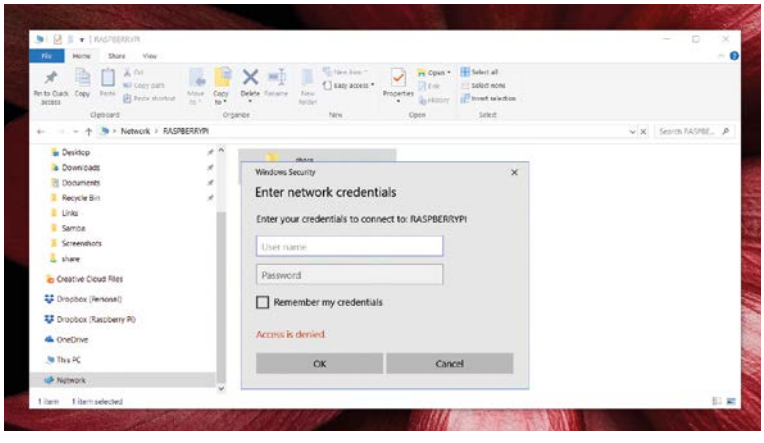
Répondez **O** lorsque demandé.

03 Créer un répertoire partagé

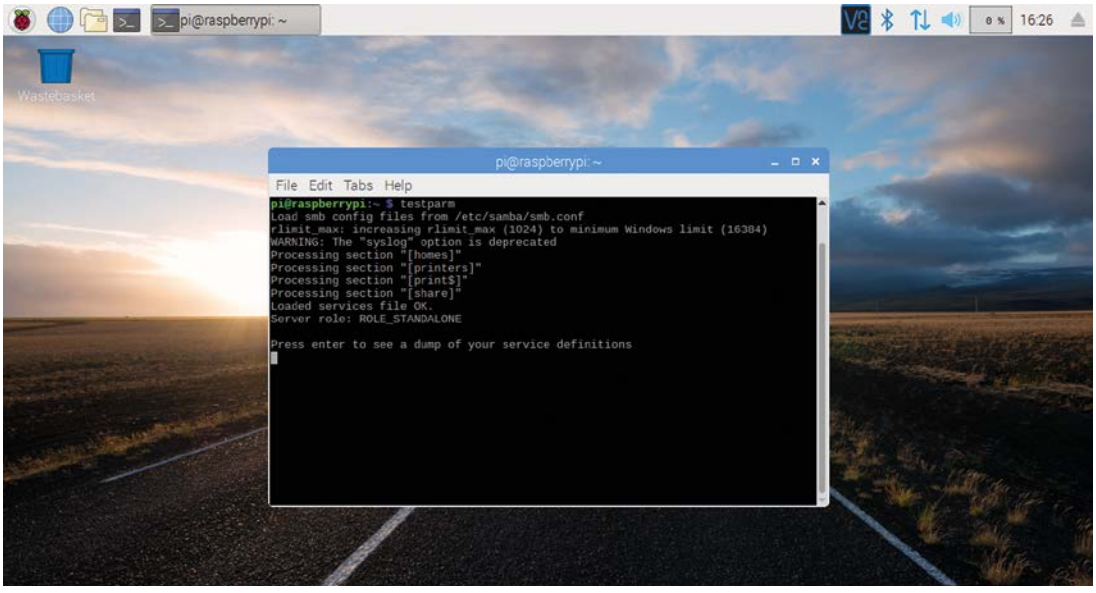
Maintenant créons un répertoire partagé dédié dans notre répertoire d'accueil.

```
sudo mkdir /home/pi/shared
```

Vous pouvez le mettre n'importe où, mais le nôtre est dans le répertoire d'accueil (cela simplifie l'accès sans avoir à modifier les autorisations). Vous pourrez y accéder depuis les autres ordinateurs de votre réseau. Les fichiers que vous y placerez



Entrez les identifiants (nom d'utilisateur et mot de passe) créés pendant la phase de configuration pour accéder au partage Samba.



Testez vos modifications de **smb.conf** avec **testparm**.

seront accessibles par les autres ordinateurs, et les fichiers déposés dans ce répertoire depuis les autres ordinateurs apparaîtront sur votre RPi.

04 Configurer Samba

Ensuite nous modifions le fichier de configuration de Samba pour rendre visible par les autres ordinateurs du réseau le répertoire partagé. Commençons par une sauvegarde au cas où vous voudriez examiner l'original.

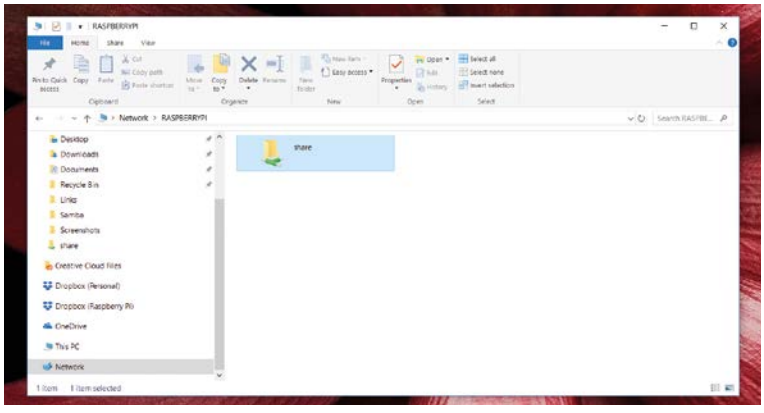
```
sudo cp /etc/samba/smb.conf /etc/samba/smb.conf_backup
```

Éditons maintenant le fichier de configuration de Samba. Entrez :

```
sudo nano /etc/samba/smb.conf
```

Le fichier **smb.conf** est rempli d'informations détaillées sur la configuration de Samba (c'est une bonne idée de les lire toutes). Descendez à la fin du fichier et ajoutez le code de configuration du listage **smb-pi.conf**. Tapez CTRL+O pour enregistrer le fichier puis CTRL+X pour quitter **nano**.

Le partage Samba affiché par l'Explorateur de fichiers dans son réseau.



05 Examiner le fichier de configuration

Le fichier de configuration utilise le même format que les fichiers **.ini** de Windows, avec des sections entourées par des crochets. Les commandes sont insensibles à la casse et ignorent les espaces. Regardons de plus près quelques-unes de ces lignes…

L'emplacement du dossier partagé :

```
path = /home/pi/shared
```

Indiquez « no » ici pour désactiver le service :

```
available = yes
```

Voici la liste des utilisateurs autorisés à se connecter au service :

```
valid users = pi
```

Pour que ce partage soit vu dans la liste des partages disponibles dans une vue du réseau et dans la liste de navigation :

```
browsable = yes
```

Vérifiez que les utilisateurs peuvent écrire dans les fichiers (par défaut, c'est non) :

```
writable = yes
```

Consultez la documentation sur **Samba.org** pour plus d'informations sur les options de configuration disponibles dans *smb.conf* (**magpi.cc/alEJZY**). Testez le fichier de configuration de Samba avec **testparm**. Entrez :

```
testparm
```

… et vous devriez voir une série de tests « *Processing section* ». Cela devrait se terminer par « *Loaded services file OK* ». Sinon, revenez au début de cette étape et vérifiez que vous avez saisi correctement le texte de configuration additionnel.

06 Créer un mot de passe

Avant de démarrer le serveur, vous devriez mettre un mot de passe à Samba. Entrez :

```
sudo smbpasswd -a pi
```

Entrez un mot de passe simple et facile à se rappeler pour les essais (vous pourrez en créer un plus sécurisé plus tard en relançant la commande à volonté).

Maintenant redémarrez votre serveur Samba :

```
sudo service smbd restart
```

Le service Samba va redémarrer et le disque partagé sera maintenant disponible sur le réseau.

07 Activer SMB dans Windows

D'habitude Samba était installé par défaut dans Windows, mais c'est récemment devenu une installation optionnelle.

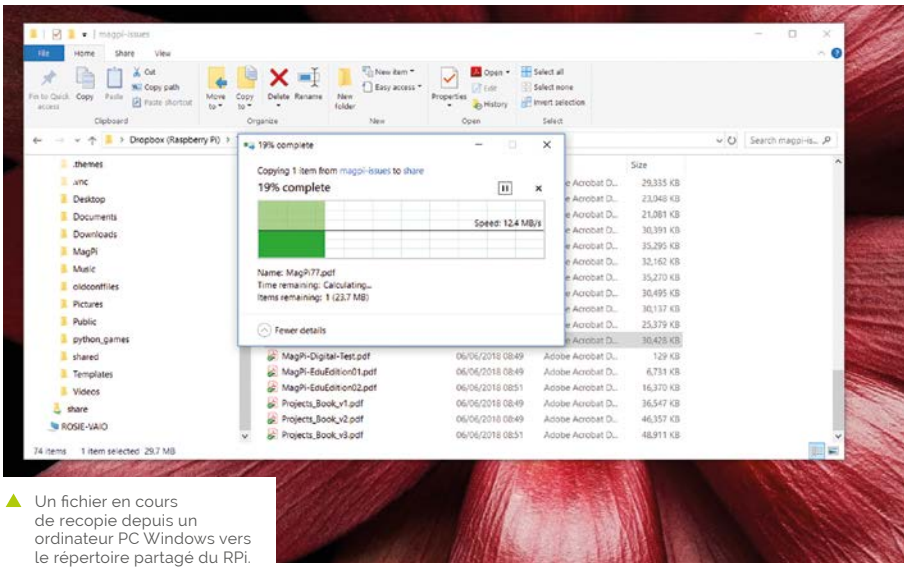
Dans Windows 10, cliquez sur la barre de recherche et tapez « Panneau de configuration ». Après sélection du panneau, cliquez sur « Programmes » et « Activer ou désactiver des fonctionnalités Windows ». Descendez pour trouver « Support de partage de fichiers SMB 1.0/CIFS » et cliquez sur l'icône « + » pour dévoiler ses options. Cochez la case marquée « Client SMB 1.0/CIFS ». Cliquez sur OK. La prise en charge du client Samba sera alors activée sur votre PC Windows pour qu'il puisse accéder au RPi.

08 Trouver votre RPi sur le réseau

Maintenant vous pouvez trouver votre serveur de fichiers RPi (nommé RASPBERRYPI par défaut) depuis n'importe quel appareil sur votre réseau local.

Cliquez sur l'icône de l'Explorateur de fichiers dans Windows et choisissez l'option Réseau dans la barre latérale. Vous verrez RASPBERRYPI dans la liste des appareils. Cliquez dessus et la fenêtre « Entrer les informations d'identification réseau » apparaît. Entrez votre nom d'utilisateur **pi** et le mot de passe créé à l'étape 6. Cliquez sur « Mémoriser mes informations d'identification » si vous voulez enregistrer le mot de passe puis cliquez sur OK.

L'Explorateur de fichiers va maintenant afficher le répertoire partagé. Vous pouvez glisser des fichiers et dossiers depuis Windows vers le dossier partagé et ils seront recopiés sur votre RPi.



▲ Un fichier en cours de recopie depuis un ordinateur PC Windows vers le répertoire partagé du RPi.

smb-pi.conf

TÉLÉCHARGER LE CODE COMPLET : magpi.cc/pFBYLi

> Langage : **ASCII**

```
001.  [share]
002.      path = /home/pi/shared
003.      available = yes
004.      valid users = pi
005.      read only = no
006.      browsable = yes
007.      public = yes
```

09 Trouver le RPi sur un Mac

Samba fonctionne sur de nombreux appareils différents. Il est parfaitement possible d'accéder au dossier partagé depuis un ordinateur Mac, ou même depuis un appareil iOS ou Android (avec l'appli adéquate). Ouvrez une nouvelle fenêtre Finder sur un Mac et choisissez Réseau dans la barre latérale. Cliquez sur « raspberrypi » et « partage » pour accéder au répertoire. Comme avec Windows, vous pouvez glisser des fichiers directement de et vers le RPi. **(VF : Denis Lafourcade)**

Astuce

Tout redémarrer

Si vous avez tout configuré et tout vérifié, essayez de redémarrer votre RPi et votre ordinateur Windows. Quelquefois cela aide Windows à trouver le disque sur le réseau (une fois que le disque partagé apparaît, cela reste stable).

Partie 3

Codez un jeu en vue isométrique : AmazeBalls

Quelques méchants et un peu de dynamite agrémenteront le code de cette dernière partie.

MAKER

Mark Vanstone

Auteur de logiciels éducatifs dans les années 1990, dont la série ArcVenture, perdue corps et biens dans l'océan des logiciels, mais sauvée de l'oubli par le RPi !

magpi.cc/YiZnxL

@mindexplorers

Ingrédients

- Raspbian Jessie ou plus récente
- L'éditeur de tuiles gratuit Tiled (**mapeditor.org**)
- Les images de **magpi.cc/NvafjA** ou les vôtres
- La dernière version de Pygame Zero (1.2)

À ce stade du code, parcourir notre labyrinthe s'avère aussi excitant qu'une ronde de nuit autour d'un château d'eau. Aussi allons-nous y ajouter des « méchants », des balles errantes capables de bloquer le joueur en déplaçant des blocs. Et pour que l'infortuné ne finisse pas emmuré, nous lui permettrons de dégager le passage à coups de dynamite.

01 Recyclage à 2 balles

Nous disposons déjà de 8 images correspondant aux différentes phases du rebond de la balle `player`. Plutôt que d'en recréer 8 nouvelles pour chaque balle ennemie, nous pouvons modifier la couleur de celles dont nous disposons déjà, p. ex. avec l'outil *Colorier* de GIMP. C'est ce que j'ai fait, en enregistrant les images sous les noms *eballo.png*, *eball1.png*, etc. (le préfixe « e » est pour « enemy »). J'ai choisi de garder la même couleur (violet) pour les autres balles, donc ces 8 images suffisent. Vous les trouverez sur GitHub (**magpi.cc/NvafjA**).

02 Ne pas réinventer la balle

Côté code, nous avions représenté le joueur au moyen d'un dictionnaire. Comme les balles ennemies seront elles aussi des balles bondissantes, nous pouvons copier-coller la définition du dictionnaire `player` et le renommer `enemy1` pour définir une première balle. Nous ajouterons d'autres balles plus tard, pour l'instant initialisons les valeurs `x` et `y` de celle-ci à 13 pour la placer au centre du labyrinthe.

03 Animation de l'ennemie

L'animation de la balle ennemie ne différant en rien de celle du joueur, nous pouvons réutiliser la fonction `updateBall()` du code précédent. Là encore nous l'appelons depuis la fonction `update()` de Pygame Zero, mais au lieu de lui passer le paramètre `player`, nous lui passons le dictionnaire `enemy1` en écrivant `updateBall(enemy1)` juste après `updateBall(player)`. Dès que cette balle sera mise en mouvement, toutes les valeurs de ses clés seront donc modifiées comme le sont celles de la balle `player`.

04 Affichage de l'ennemie

Si nous n'avons rien eu à faire pour mettre à jour l'animation de la balle ennemie, son affichage nécessite par contre un traitement spécial dans la fonction `drawMap()` puisque notre balle doit se déplacer à mesure que la carte défile. Pour déterminer ses coordonnées d'écran, nous devons donc utiliser à la fois la position de la carte et les valeurs de ses clés `sx` et `sy`. Comme le montre la nouvelle définition de `drawMap()` (listage **figure1.py**), nous calculons les positions `bx` et `by` du bloc, puis les ajoutons aux valeurs `sx` et `sy` (ligne 020).

05 Rebond miroir

Si vous lancez le programme maintenant, vous verrez une balle violette rebondir au centre du labyrinthe, son rebond étant synchrone avec celui de la balle `player`. Si vous préférez des rebonds désynchronisés, modifiez la valeur d'initialisation de la clé `frame` du dictionnaire `enemy1`. Définissons à présent une fonction appelée `updateEnemy()`

▼ Incorporation de la balle ennemie dans `drawMap()`.

dont le rôle sera de bouger la balle ennemie et de déplacer des blocs.

06 Déplacement aléatoire

Comme nous l'avons déjà fait durant cette série, nous affecterons à la balle ennemie un déplacement aléatoire afin que le jeu ne soit pas trop prévisible. Importons donc le module `random` en début de code avec `from random import randint`, et définissons la fonction `updateEnemy()` en ajoutant la ligne `updateEnemy(e):`. Le paramètre **e** représente le dictionnaire `enemy` que nous passerons à la fonction lors de son appel. Définissons à présent une liste de 4 directions, elles-mêmes représentées par des listes de la forme `[x,y]`. La liste [0,1] signifiera p. ex. qu'il n'y a aucun bloc à déplacer dans la direction x, et un bloc à déplacer en y.

07 Chamboule-tout

Définissons donc notre liste avec `edirs = [[-1,0],[0,1],[1,0],[0,-1]]`. Nous choisirons l'une des 4 directions au moyen d'un entier aléatoire r compris entre 0 et 3 donné par `r = randint(0,3)`. Nous voici prêts à réutiliser `enemy`, mais avant cela nous devons apporter deux modifications à la fonction `doMove()`. La première est de retourner la valeur du bloc vers lequel se déplace la balle : s'il

figure1.py

▶ Langage : **Python 3**

```
001. def drawMap():
002.     psx = OFFSETX
003.     psy = OFFSETY-32
004.     mx = psx - player["sx"]
005.     my = psy - player["sy"]+32
006.
007.     for x in range(player["x"]-12, player["x"]+16):
008.         for y in range(player["y"]-12, player["y"]+16):
009.             if onMap(x,y):
010.                 b = mapData["data"][y][x]
011.                 td = findData(mapData["tiles"], "id", b)
012.                 block = td["image"]
013.                 bheight =  td["imageheight"]-34
014.                 bx = (x*32)-(y*32) + mx
015.                 by = (y*16)+(x*16) + my
016.                 if -32 <= bx < 800 and 100 <= by < 620:
017.                     screen.blit(block, (bx, by - bheight))
018.                 if x == player["x"] and y == player["y"]:
019.                     screen.blit("ball"+str(player["frame"]),
      (psx, psy))
020.                 if x == enemy1["x"] and y == enemy1["y"]:
021.                     screen.blit("eball"+str(enemy1[
      "frame"]),(bx + enemy1["sx"],(by-32)+enemy1["sy"]))
```

figure2.py

Langage : **Python 3**

```
001.  def doMove(p, x, y):
002.      global mazeSolved
003.      if onMap(p["x"]+x, p["y"]+y):
004.          mt = mapData["data"][p["y"]+y][p["x"]+x]
005.          if mt == 1 or mt == 3:
006.              p.update({"queueX":x, "queueY":y,
      "moveDone":False})
007.              if mt == 3 and p == player:
008.                  mazeSolved = True
009.          return mt
```

▲ `doMove()` modifiée afin que la balle ennemie ne déclenche pas la fin de partie.

Astuce

Carte dynamique

Le type d'un bloc peut être modifié en changeant sa clé id dans `mapData`. Animer ainsi les éléments de la carte depuis `update()` est très amusant.

▲ Les images d'un jeu peuvent être créées de diverses façons. Celle-ci l'a été avec le logiciel de modélisation 3D gratuit Blender.

s'agit d'un bloc de valeur 2 (un bloc de type mur), alors nous désorienterons le joueur en chamboulant la position des blocs alentours – sournois, n'est-ce pas ?

08 Sournois mais régulier

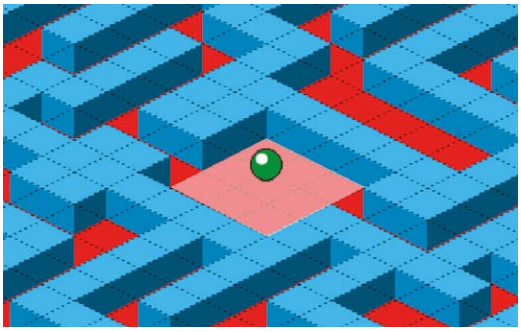
La seconde modification concerne la condition de fin de partie. La fonction `doMove()` précédente vérifiait si la balle `player` était sur la tuile *finish* et, si c'était le cas, mettait la variable `mazeSolved` à `True`. Si nous réutilisions le même code pour la balle ennemie, celle-ci pourrait gagner à la place du joueur. Pour parer à cette éventualité, nous vérifions donc si la balle qui atteint la tuile finish est bien l'acteur `player` (`if mt == 3 and p == player:`, listage **figure2.py**).

09 Blocs dynamiques

Comme les blocs sont représentés par des entiers dans le dictionnaire `mapData`, il nous suffit de changer ces valeurs pour modifier à la volée le tracé du labyrinthe. Un bloc de type mur a une valeur de 2, et un bloc de type sol vaut 1. Si donc nous affectons 1 à un bloc de valeur 2, le joueur verra un pan de mur disparaître et être remplacé par une tuile. C'est de cette façon que nous changerons le labyrinthe à mesure que la balle ennemie se déplacera.

10 Déplacement des murs

Revenons à la fonction `updateEnemy()` (listage **figure3.py**). Nous devons donc d'abord



▲ La dynamite transforme tous les blocs entourant le joueur en blocs de type sol (ici représentés en rose).

générer un entier aléatoire pour déterminer une certaine direction, puis déplacer la balle ennemie dans cette direction. Si la balle est dirigée vers un bloc de type mur, nous le déplaçons dans la même direction que la balle après avoir vérifié qu'il y avait un espace libre derrière lui. Pour effectuer ces actions, nous appelons `doMove()` avec le dictionnaire `enemy1` (passé à `updateEnemy()`) puis, si le bloc est de type mur (sa clé `id` a une valeur de 2), nous appelons une fonction nommée `moveBlock()`.

11 Changement de position

Nous passerons à cette fonction `moveBlock()` (listage **figure3.py**) les coordonnées `x` et `y` du bloc ainsi que les valeurs de la direction utilisée pour déplacer la balle ennemie. La fonction teste d'abord l'appartenance des valeurs du déplacement à la carte du labyrinthe, puis vérifie si le bloc est déplacé vers un bloc de type sol (de clé `id` égale à 1). Si ces deux conditions sont vraies, nous affectons au bloc de type sol la clé `id` du bloc déplacé.

12 Mais plus précisément ?

Le code du listage **figure3.py** ayant de quoi dérouter, explicitons-le. La fonction `updateEnemy()` définit pour l'essentiel les quatre directions possibles, puis dicte : si la balle ennemie n'est pas en train de se déplacer, alors choisir une direction au hasard et passer les valeurs x et y de cette direction à la fonction `doMove()`. Si la clé id du bloc vers lequel se fait le déplacement vaut 2, alors déplacer ce bloc en utilisant les valeurs de la position à atteindre ainsi que celles de sa direction. Mettre ensuite à 0 les coordonnées d'écran `x` et `y` (`sx` et `sy`) du dictionnaire `e` de la balle ennemie.

13 Ajustement des coordonnées

Notez que si la balle ennemie se déplace, alors nous vérifions si la valeur de la clé `frame` vaut 7 (dernière phase des rebonds), et si c'est le cas nous modifions les coordonnées de façon à ce qu'elles soient relatives au nouvel emplacement sur la carte, et non à l'ancien. La fonction `moveBlock()` vérifie directement avec les données de `mapData` que le bloc peut être déplacé, déplace les données de l'emplacement source vers l'emplacement cible, et définit l'emplacement source comme étant un bloc de type sol.

14 Ajout d'ennemies

Ceci fait, il ne nous reste plus qu'à écrire `updateEnemy(enemy1)` juste après les appels à `updateBall()` dans la fonction `update()`. Il est dès lors facile d'ajouter une deuxième balle ennemie afin de rendre le jeu plus intéressant. Pour cela il suffit en effet de dupliquer le dictionnaire enemy1, de le nommer `enemy2`, d'initialiser ses valeurs de départ `x` et `y` à 25 (p. ex.), puis d'ajouter les appels `updateBall(enemy2)` et `updateEnemy(enemy2)` dans `update()`. Vous pouvez ajouter d'autres balles de la même façon, ou les placer toutes dans une liste pour avoir un code plus concis.

15 Équilibrage des forces

L'ajout de balles ennemies rend certes le jeu plus intéressant, mais ne laisse guère de chances au joueur de s'en sortir (littéralement). Un peu de dynamite devrait l'aider ! J'ai donc créé deux images : celle d'une tuile portant un bâton de dynamite, et celle d'un simple bâton de dynamite servant à monter au joueur combien de bâtons il a récolté. Vous les trouverez, de même que les fichiers *.json* et *.tsx* nécessaires au chargement de la carte, sur notre dépôt GitHub : **magpi.cc/NvafjA**.

figure3.py

Langage : **Python 3**

```
001. def updateEnemy(e):
002.     edirs = [[-1,0],[0,1],[1,0],[0,-1]]
003.     if e["moveX"] == 0 and e["moveY"] == 0:
004.         r = randint(0,3)
005.         if doMove(e, edirs[r][0], edirs[r][1]) == 2:
006.             moveBlock(e["x"]+edirs[r][0],e["y"]+edirs[r]
     [1],edirs[r][0],edirs[r][1])
007.         e["sx"] = e["sy"] = 0
008.     else:
009.         if e["frame"] == 7 and e["movingNow"] == True:
010.             if e["sx"] == 12: e["sx"] -= 32
011.             if e["sx"] == -12: e["sx"] += 32
012.             if e["sy"] == 6: e["sy"] -= 16
013.             if e["sy"] == -6: e["sy"] += 16
014.
015. def moveBlock(mx,my,dx,dy):
016.     if onMap(mx+dx,my+dy):
017.         d = mapData["data"][my+dy][mx+dx]
018.         if d == 1:
019.             mapData["data"][my+dy][mx+dx] =
     mapData["data"][my][mx]
020.             mapData["data"][my][mx] = 1
```

▲ Mise à jour de la balle ennemie et déplacement des blocs sur son passage.

▼ Ajout de deux balles ennemies dans la fonction `update()`.

figure4.py

langage : **Python 3**

```
001. def update(): # Pygame Zero update function
002.     global player, timer
003.     mt = 0
004.     if player["moveDone"] == True:
005.         if keyboard.left: mt = doMove(player, -1, 0)
006.         if keyboard.right: mt = doMove(player, 1, 0)
007.         if keyboard.up: mt = doMove(player, 0, -1)
008.         if keyboard.down: mt = doMove(player, 0, 1)
009.     if mt == 4:
010.         mapData["data"][ player["y"] + player["queueY"]][
     player["x"] + player["queueX"]] = 1
011.         player["dynamite"] += 1
012.     updateBall(player)
013.     updateBall(enemy1)
014.     updateBall(enemy2)
015.     updateEnemy(enemy1)
016.     updateEnemy(enemy2)
```

16 Gestion des explosifs

Commençons par ajouter au dictionnaire `player` une paire clé-valeur stockant le nombre de bâtons de dynamite détenus par le joueur : `"dynamite":0`. J'ai déjà ajouté les blocs sol-dynamite à la carte du labyrinthe depuis l'éditeur de tuiles Tiled (cf. le tutoriel précédent pour son utilisation), je supposerai que vous avez fait de même si vous utilisez votre propre tracé. La carte étant prête, nous testons dans la fonction `update()`, juste après la lecture du clavier, si le joueur se trouve sur un bloc de type dynamite, autrement dit si `mt` vaut 4 (cf. listage **figure4.py**). Si oui, nous transformons le bloc dynamite en bloc de type sol, ce qui le fera disparaître de la carte, et incrémentons la valeur de la clé `dynamite`.

17 Affichage des icônes

Comme nous l'avons déjà fait dans certains jeux pour indiquer p. ex. le nombre de vies, nous utiliserons des icônes pour afficher le nombre de bâtons récoltés par le joueur. Nous le faisons dans la fonction `draw()` en écrivant : `for l in range(player["dynamite"]): screen. blit("dmicon", (650+(l*32),80))`. Avec ces valeurs, l'affichage est en haut et à droite de l'écran.

18 Et boum !

Il nous reste à coder l'explosion de la dynamite. La fonction `on_key_down()` de Pygame Zero s'avère ici incontournable. Nous y testons si le joueur a appuyé sur la barre d'espace (SPACE), et si c'est le cas une boucle imbriquée transforme les blocs entourant le joueur en blocs de type sol. Voyez le listage **amazeballs3.py** pour les détails de ce dernier morceau de code. Mais en avons-nous vraiment terminé avec ce jeu ? Est-il trop facile ? Trop difficile ? Faut-il plus de balles ennemies ? D'autres objets à collecter ? Différents tracés ? Cette dernière étape vous appartient !

19 À vous de jouer !

Ce tutoriel conclut notre série consacrée à Pygame Zero qui, je l'espère, vous permettra de concrétiser vos propres idées de jeux. Il est d'ailleurs temps d'avoir une pensée chaleureuse pour Daniel Pope, le créateur de Pygame Zero. Car, vous serez sans doute d'accord avec moi, cette bibliothèque est vraiment un tremplin idéal pour la programmation de jeux vidéo en Python.

(VF : Hervé Moreau)

amazeballs3.py

▸ Langage : **Python**

```
001. import pgzrun
002. import map3d
003. from random import randint
004.
005. player = {"x":3, "y":3, "frame":0, "sx":0, "sy":96,
006.           "moveX":0, "moveY":0, "queueX":0, "queueY":0,
007.           "moveDone":True, "movingNow":False,
     "animCounter":0, "dynamite":0}
008. enemy1 = {"x":13, "y":13, "frame":0, "sx":0, "sy":0,
009.           "moveX":0, "moveY":0, "queueX":0, "queueY":0,
010.           "moveDone":True, "movingNow":False,
     "animCounter":0}
011. enemy2 = {"x":25, "y":25, "frame":0, "sx":0, "sy":0,
012.           "moveX":0, "moveY":0, "queueX":0, "queueY":0,
013.           "moveDone":True, "movingNow":False,
     "animCounter":0}
014. OFFSETX = 368
015. OFFSETY = 300
016. timer = 0
017. mazeSolved = False
018.
019. mapData = map3d.loadmap("maps/map1.json")
020.
021. def draw(): # Pygame Zero draw function
022.     screen.fill((0, 0, 0))
023.     drawMap()
024.     screen.blit('title', (0, 0))
025.     screen.draw.text("TIME: "+str(timer) , topleft=(20,
     80), owidth=0.5, ocolor=(255,255,0), color=(255,0,0) ,
     fontsize=60)
026.     for l in range(player["dynamite"]): screen.blit(
     "dmicon", (650+(l*32),80))
027.     if mazeSolved:
028.         screen.draw.text("MAZE SOLVED in " + str(timer) + "
     seconds!" , center=(400, 450), owidth=0.5, ocolor=(0,0,0),
     color=(0,255,0) , fontsize=60)
029.
030.
031. def update(): # Pygame Zero update function
032.     global player, timer
033.     mt = 0
034.     if player["moveDone"] == True:
035.         if keyboard.left: mt = doMove(player, -1, 0)
036.         if keyboard.right: mt = doMove(player, 1, 0)
037.         if keyboard.up: mt = doMove(player, 0, -1)
038.         if keyboard.down: mt = doMove(player, 0, 1)
039.     if mt == 4:
040.         mapData["data"][ player["y"] + player["queueY"]][
     player["x"] + player["queueX"]] = 1
041.         player["dynamite"] += 1
042.     updateBall(player)
043.     updateBall(enemy1)
044.     updateBall(enemy2)
045.     updateEnemy(enemy1)
046.     updateEnemy(enemy2)
047.
048. def on_key_down(key):
049.     if player["dynamite"] > 0 and key.name == "SPACE":
050.         player["dynamite"] -= 1
051.         for x in range(player["x"]-1, player["x"]+2):
```

TÉLÉCHARGEZ LE CODE COMPLET :

```
052.             for y in range(player["y"]-1, player["y"]+2):
053.                 mapData["data"][y][x] = 1
054.
055. def timerTick():
056.     global timer
057.     if not mazeSolved:
058.         timer += 1
059.
060. def drawMap():
061.     psx = OFFSETX
062.     psy = OFFSETY-32
063.     mx = psx - player["sx"]
064.     my = psy - player["sy"]+32
065.
066.     for x in range(player["x"]-12, player["x"]+16):
067.         for y in range(player["y"]-12, player["y"]+16):
068.             if onMap(x,y):
069.                 b = mapData["data"][y][x]
070.                 td = findData(mapData["tiles"], "id", b)
071.                 block = td["image"]
072.                 bheight =  td["imageheight"]-34
073.                 bx = (x*32)-(y*32) + mx
074.                 by = (y*16)+(x*16) + my
075.                 if -32 <= bx < 800 and 100 <= by < 620:
076.                     screen.blit(block, (bx, by -
    bheight))
077.                 if x == player["x"] and y == player["y"]:
078.                     screen.blit("ball"+str(player[
    "frame"]), (psx, psy))
079.                 if x == enemy1["x"] and y ==
    enemy1["y"]:
080.                     screen.blit("eball"+str(enemy1[
    "frame"]), (bx + enemy1["sx"], (by-32)+enemy1["sy"]))
081.                 if x == enemy2["x"] and y ==
    enemy2["y"]:
082.                     screen.blit("eball"+str(enemy2[
    "frame"]), (bx + enemy2["sx"], (by-32)+enemy2["sy"]))
083.
084. def findData(lst, key, value):
085.     for i, dic in enumerate(lst):
086.         if dic[key] == value:
087.             return dic
088.     return -1
089.
090. def onMap(x,y):
091.     if 0 <= x < mapData["width"] and 0 <= y <
    mapData["height"]:
092.         return True
093.     return False
094.
095. def doMove(p, x, y):
096.     global mazeSolved
097.     if onMap(p["x"]+x, p["y"]+y):
098.         mt = mapData["data"][p["y"]+y][p["x"]+x]
099.         if mt == 1 or mt == 3 or mt == 4:
100.             p.update({"queueX":x, "queueY":y,
    "moveDone":False})
101.             if mt == 3 and p == player:
102.                 mazeSolved = True
103.         return mt
104.
105. def updateEnemy(e):
106.     edirs = [[-1,0],[0,1],[1,0],[0,-1]]
107.     if e["moveX"] == 0 and e["moveY"] == 0:
108.         r = randint(0,3)
109.         if doMove(e, edirs[r][0], edirs[r][1]) == 2:
110.             moveBlock(e["x"]+edirs[r][0],e["y"]+
    edirs[r][1],edirs[r][0],edirs[r][1])
111.         e["sx"] = e["sy"] = 0
112.     else:
113.         if e["frame"] == 7 and e["movingNow"] == True:
114.             if e["sx"] == 12: e["sx"] -= 32
115.             if e["sx"] == -12: e["sx"] += 32
116.             if e["sy"] == 6: e["sy"] -= 16
117.             if e["sy"] == -6: e["sy"] += 16
118.
119. def moveBlock(mx,my,dx,dy):
120.     if onMap(mx+dx,my+dy):
121.         d = mapData["data"][my+dy][mx+dx]
122.         if d == 1:
123.             mapData["data"][my+dy][mx+dx] =
    mapData["data"][my][mx]
124.             mapData["data"][my][mx] = 1
125.
126. def updateBall(p):
127.     if p["movingNow"]:
128.         if p["moveX"] == -1: moveP(p,-1,-0.5)
129.         if p["moveX"] == 1: moveP(p,1,0.5)
130.         if p["moveY"] == -1: moveP(p,1,-0.5)
131.         if p["moveY"] == 1: moveP(p,-1,0.5)
132.     p["animCounter"] += 1
133.     if p["animCounter"] == 4:
134.         p["animCounter"] = 0
135.         p["frame"] += 1
136.         if p["frame"] > 7:
137.             p["frame"] = 0
138.         if p["frame"] == 4:
139.             if p["moveDone"] == False:
140.                 if p["queueX"] != 0 or p["queueY"] !=0:
141.                     p.update({"moveX":p["queueX"],
    "moveY":p["queueY"], "queueX":0, "queueY":0,
    "movingNow": True})
142.             else:
143.                 p.update({"moveDone":True, "moveX":0,
    "moveY":0, "movingNow":False})
144.
145.         if p["frame"] == 7 and p["moveDone"] == False
    and p["movingNow"] == True:
146.             p["x"] += p["moveX"]
147.             p["y"] += p["moveY"]
148.             p["moveDone"] = True
149.
150. def moveP(p,x,y):
151.     p["sx"] += x
152.     p["sy"] += y
153.
154. clock.schedule_interval(timerTick, 1.0)
155. pgzrun.go()
```

Créez des romans graphiques

Convertissez une vidéo enregistrée avec votre HAT TV en roman-photo, et feuilletez-le sur votre liseuse ou tablette.

MAKER

PJ Evans

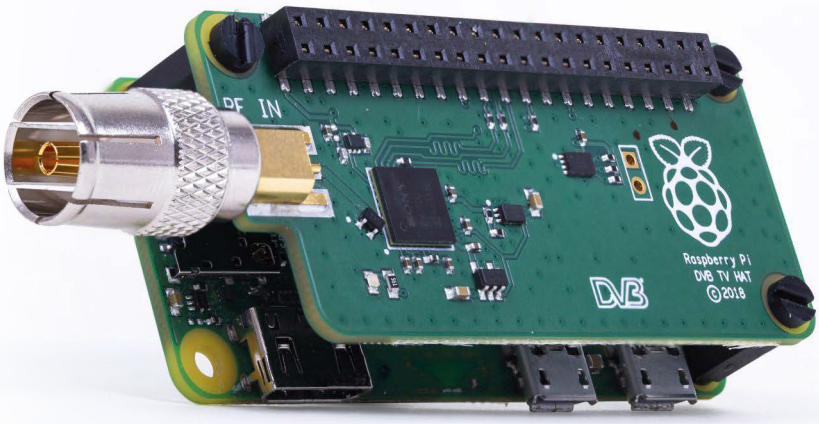
PJ est écrivain, développeur et organisateur de Jam RPi. Ses goûts cinématographiques sont déplorables.

mrpjevans.com

Que diriez-vous de *lire* la télé ? O-k, cela paraît idiot, alors essayons autrement : que diriez-vous de transformer un épisode de votre série préférée en roman-photo, puis de le feuilleter tranquillement sur une liseuse ou sur une tablette ? Voilà qui semble déjà plus sympa. Le vrai plaisir ici sera toutefois de découvrir de façon ludique la puissance de Python en tant que langage de script. Nous verrons comment utiliser divers outils de traitement d'images, de reconnaissance optique des caractères et de création de PDF pour extraire les images et les sous-titres d'une vidéo, puis comment les mettre en page dans un PDF.

01 Enregistrement d'une vidéo

Je supposerai le logiciel Tvheadend ainsi que l'extension HAT TV déjà installés sur votre RPi (**cf. magpi.cc/QCkFdt**). Nous aurons besoin d'un enregistrement vidéo (quelconque, mais avec des sous-titres). Sous Tvheadend, les flux enregistrés se trouvent dans l'onglet *Enregistrements terminés* du menu *Magnétoscope numérique*. Vous pouvez télécharger le fichier enregistré depuis cet onglet, ou le récupérer sous **/var/lib/hts**. Tvheadend enregistre les flux dans leur format original de diffusion MPEG-2 TS (*Transport Stream*).



▼ Le branchement d'une antenne numérique au HAT TV permet de capter plus de 80 chaînes de télévision et de radio.

02 Installation des dépendances

La conversion finale en PDF passera par différentes étapes, dont l'extraction des données de la vidéo, la reconnaissance optique des caractères (OCR) des sous-titres, et la création proprement dite du PDF. Chacune sera effectuée par un script Python appelant divers utilitaires. Leur rôle deviendra clair au fur et à mesure de leur utilisation. Pour l'instant, téléchargeons tout ce dont Python aura besoin pour diriger notre chaîne d'assemblage :

```
sudo apt update && sudo apt -y upgrade
sudo apt install git python3-pip ffmpeg
imagemagick
pip3 install fpdf arrow
```

03 Installation de ccextractor

L'utilitaire ccextractor peut extraire les sous-titres de vidéos respectant la norme DVB (*Digital Video Broadcasting*). N'étant hélas pas disponible dans les dépôts APT, il nous faut le compiler. Entrez donc les instructions suivantes pour télécharger son code source depuis GitHub, installer les dépendances nécessaires, compiler le code, puis installer le programme :

```
cd
git clone https://github.com/CCExtractor/
ccextractor.git
sudo apt install -y libglfw3-dev cmake
```

TÉLÉCHARGEZ LE CODE COMPLET :
magpi.cc/swtRoH

Les sous-titres sont extraits, convertis en texte, puis affichés avec une police de style BD.

Les images sont extraites en fonction des changements de scène et de l'apparition des sous-titres.

```
gcc libcurl4-gnutls-dev tesseract-ocr
tesseract-ocr-dev libleptonica-dev
 cd ccextractor/linux
 ./build
 sudo mv ./ccextractor /usr/local/bin/
```

04 Installation du script

La création du PDF pouvant impliquer des centaines, voire des milliers de fichiers, j'ai automatisé leur manipulation au moyen d'un script Python. Comme il est un peu long à copier, je l'ai placé sur mon dépôt GitHub. Utilisons donc à nouveau Git pour le récupérer :

```
 cd
 git clone https://github.com/mrpjevans/
comical.git
```

Le dossier **comical** créé contient le script **comical.py** et quelques fichiers nécessaires.

05 Extraction des sous-titres

Plutôt que d'exécuter le script en une seule passe, nous le ferons par étapes afin de mieux saisir son fonctionnement. Vous devez connaître le chemin d'accès de votre fichier vidéo. Comme exemple, j'ai choisi le kitschissime *Plan 9 from Outer Space*, aujourd'hui dans le domaine public. Commençons par en extraire les sous-titres, ils serviront plus tard de légende aux images :

> « Je supposerai le logiciel Tvheadend ainsi que l'extension HAT TV déjà installés sur votre Raspberry Pi. »

```
 cd ~/comical
 python3 comical.py -i plan9.ts --extract
```

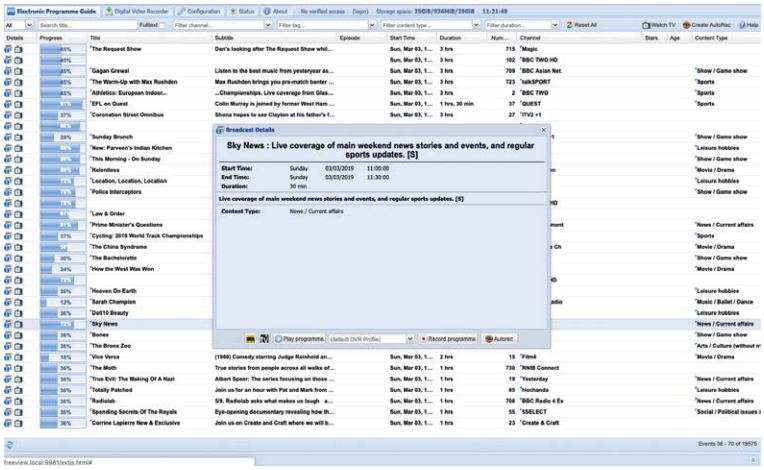
Le dossier **plan9.d** créé par cette commande contient l'image PNG de chaque sous-titre, ainsi qu'un fichier XML **plan9.xml** indiquant leur horodatage.

06 Redimensionnement

Mais pourquoi des sous-titres sous forme d'images ? Parce qu'ainsi le veut la norme de diffusion numérique européenne DVB. Pour les utiliser, nous n'avons donc d'autre choix que d'opérer une reconnaissance des caractères sur les PNG créés par ccextractor. Ces caractères étant trop petits pour être reconnus par l'application Tesseract, nous les agrandissons et les convertissons en noir et blanc avec l'option

Ingrédients

- Extension RPi TV HAT **www.elektor.fr/rpi-tv-hat**
- Tvheadend (installation : **magpi.cc/QCkFdt**)
- Liseuse ou tablette



La richesse des fonctions offertes par Tvheadend compense largement l'étrangeté de son interface.

clean, qui appelle en sous-main l'utilitaire *mogrify* d'ImageMagick :

```
python3 comical.py -i plan9.ts --clean
```

Les sous-titres sont maintenant à la bonne échelle et monochromes.

07 OCR avec Tesseract

Conçu à l'origine par Hewlett-Packard et aujourd'hui *open source*, Tesseract est un logiciel d'OCR remarquable. Un exemple ? Pour convertir quelque chose en texte, lancez simplement :

```
tesseract <fichier image> <fichier de
sortie>
```

Notre script envoie quant à lui chaque image extraite à Tesseract. Le redimensionnement précédent des caractères devrait donner de bons résultats.

```
python3 comical.py -i plan9.ts --ocr
```

À chaque image de sous-titre est maintenant associé un fichier .txt correspondant.

08 Extraction d'images fixes

La partie suivante du script extrait une image fixe de la vidéo en se basant sur l'horodatage des sous-titres. Ces horodatages sont d'abord récupérés avec les modules d'analyse XML de Python. Pour chacun d'entre eux, nous demandons ensuite à **ffmpeg** (le couteau suisse du traitement vidéo) d'extraire une image fixe au format JPEG, sauvegardée dans un dossier appelé (ici) **plan9_process**. Le nom du fichier représente le code temporel de l'image. Nous (re)copions également dans ce dossier le fichier du sous-titre avec un nom correspondant à son code temporel.

```
python3 comical.py -i plan9.ts --images
```

09 Détection de scène

Le dossier **plan9_process** contient tout ce qu'il faut pour assembler le PDF… à condition qu'il n'y ait aucune pause dans les dialogues, ce qui semble bien improbable. Et il y aussi les scènes dépourvues de sous-titres. Là encore **ffmpeg** vient à la rescousse grâce à un filtre perfectionné pouvant détecter une différence significative entre deux images, donc un changement de scène. Grâce à ce filtre, le script extrait d'autres images JPEG là où il y a changement de scène, ignorant celles qui sont à moins d'une seconde d'une image avec sous-titre.

```
python3 comical.py -i plan9.ts
--detectscenes
python3 comical.py -i plan9.ts
--extractscenes
```

Votre **dossier _process** contient maintenant les images utiles.

10 Assemblage

La dernière partie du script assemble un PDF à partir de tous les fichiers image et texte précédents :

```
python3 comical.py -i plan9.ts -o plan9.
pdf --build
```

Le code fait appel à la bibliothèque Python fpdf pour aligner six images par page selon une grille de 2 × 3, les sous-titres étant placés sous les images. Pour donner un cachet « roman graphique » au résultat final, le script utilise une police TTF de style BD (je l'ai ajoutée au dépôt GitHub du projet).

Astuce

Autres formats

Le script fonctionnera aussi avec les conteneurs vidéo (formats) pouvant stocker des sous-titres.

« Tesseract est un utilitaire remarquable de reconnaissance optique des caractères. »

11 Options typographiques

Un sous-titre un peu long peut faire déborder la légende, et une police trop petite laisser trop de blanc. Ce genre de problème peut se régler en ajustant les arguments des trois options suivantes :

```
python3 comical.py -i plan9.ts -o plan9.pdf --build --fontsize 8 --lineheight 5 --offset 68
```

L'option `fontsize` définit la taille de la police, `lineheigth` l'espacement vertical entre les lignes, et `offset` la position de la première ligne de texte sous l'image. La commande ci-dessus utilise les valeurs par défaut.

12 Options de commande

D'autres options permettent de contrôler le script. On peut ainsi exécuter l'ensemble des étapes en une seule passe avec l'option `full` :

```
python3 comical.py -i plan9.ts -o plan9.pdf --full
```

L'option `pre-build` effectue un pré-assemblage :

```
python3 comical.py -i plan9.ts -o plan9.pdf --prebuild
```

Cette commande exécute chaque étape précédant l'assemblage du PDF, ce qui permet de supprimer certaines images et certains sous-titres. Une fois votre dossier `_process` élagué de tout ce qui ne vous intéressait pas, lancez :

```
python3 comical.py -i plan9.ts -o plan9.pdf --build
```

13 Scénarios possibles

Ce projet vous semblera peut-être un brin frivole, mais encore une fois il était surtout prétexte à vous faire découvrir quelques techniques plutôt cool de traitement d'images. Examinez le code pour comprendre comment j'ai utilisé Python pour relier les différents utilitaires et guider les flux de données qu'ils s'échangent. Et que diriez-vous d'améliorer le script ? Vous pourriez automatiser son exécution pour chaque nouvelle vidéo enregistrée, appliquer divers filtres aux images, mixer différentes sources, ou encore vous essayer au « glitch art ». M

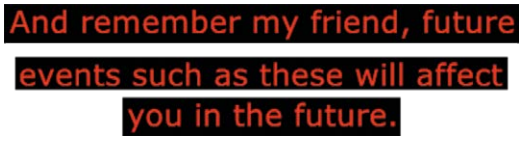
(VF : Hervé Moreau)

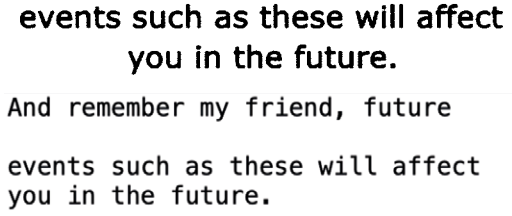
Astuce

Jouez à l'artiste

Essayez OpenCV pour donner à vos images un côté BD :

magpi.cc/RNVeo





Voici une image (non retouchée !) au format PNG extraite du flux des sous-titres de *Plan 9 from Outer Space*.

La même image PNG convertie en noir et blanc et redimensionnée à 400% avec *mogrify* pour améliorer la précision de l'OCR.

Tesseract analyse l'image et produit une sortie texte. La précision est excellente dès lors que la source est « propre ».

Pour finir, le script utilise une police apportant une touche de fantaisie aux légendes.

Fabriquer une machine à karaoké

Avec quelques logiciels astucieux et un peu de matériel, vous pouvez transformer un RPi en un système de karaoké complet.

MAKER

K.G. Orphanides

K.G est une écrivaine, développeuse et parfois musicienne, spécialisée dans la production de sons étranges à partir de cartes RPi. On a chanté beaucoup de *heavy metal* au cours de cette réalisation.

@KGOrphanides

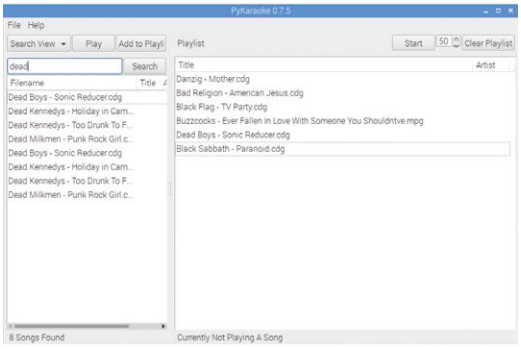
Ingrédients

- CN/A + CA/N HiFiBerry **www.elektor.fr/18948**
- Microphone dynamique (par ex. Shure SM-58 ou QTX 173.853)
- Adaptateur XLR ou TRS vers 3,5 mm
- Enceintes avec connecteurs stéréo RCA
- Moniteur, TV ou projecteur HDMI
- Pistes de karaoké MP3 + CGD

Le karaoké peut être le clou des animations d'une fête, une excellente activité pour les enfants par temps de pluie, ou un entraînement pour vos talents vocaux. Il y a deux logiciels de karaoké qui fonctionnent bien sur le RPi. Nous nous concentrerons sur le plus accessible des deux, PyKaraoke, car peu gourmand en ressources et facile à mettre en œuvre. Toutefois, si vous préférez une interface plus soignée, reportez-vous à l'encadré « Devenez pro avec OpenKJ ».

Comme le RPi n'a pas d'entrée micro, nous nous servons d'un HAT CN/A + CA/N HiFiBerry, qui dispose d'une entrée stéréo de 3,5 mm avec un pré-ampli en option et deux sorties RCA que l'on peut connecter à une paire d'enceintes amplifiées. Avec cela, il faut un microphone dynamique, plus quelques câbles appropriés pour connecter à une fiche mâle de 3,5 mm la prise XLR ou TRS ¼" de votre micro, une paire d'enceintes convenables et un grand écran, téléviseur ou projecteur pour y afficher les paroles.

Conseil : commencez ce projet avec une nouvelle installation de Raspbian, entièrement mise à jour et configurée avec `sudo raspi-config` pour étendre son système de fichiers et répondre aux exigences de surbalayage de votre écran.



▲ PyKaraoke offre une interface de consultation simple pour trouver et lire votre collection de karaoké.

▲ Si vous ne voulez pas réunir vos propres morceaux, vous pouvez toujours rechercher des pistes de karaoké sur YouTube et chanter en même temps avec un retour micro par PulseAudio.

01 Connecter le HAT CN/A + CA/N

Le CN/A + CA/N HiFiBerry se monte sur le connecteur GPIO du RPi. La carte-fille est livrée avec un jeu de quatre entretoises en plastique optionnelles pour maintenir solidement les deux cartes en place, mais que vous devrez omettre par manque de place si vous prévoyez d'utiliser un des boîtiers universels de HiFiBerry. Un boîtier spécifique CN/A + CA/N doit sortir dans les prochains mois.

02 Configurer le gain de l'entrée micro

L'entrée stéréo de 3,5 mm du CN/A + CA/N est configurée par défaut pour accepter de l'audio de niveau « ligne » – un son déjà amplifié, comme sur les connecteurs de sortie « ligne » de la plupart des appareils audio. Pour utiliser un microphone dynamique avec, activez le gain d'entrée avec un cavalier sur la barrette J1. Le cavalier doit relier la seconde paire de broches en haut de J1 pour activer le gain optionnel de 32 dB du CN/A + CA/N.

Le HAT se place sur votre connecteur GPIO, et utilise les GPIO 2 et 3 pour la configuration et les GPIO 18 à 21 comme interface audio.

Le CN/A + CA/N a deux sorties stéréo RCA pour les enceintes et une entrée stéréo de 3,5 mm utilisable pour la connexion d'un micro dynamique.

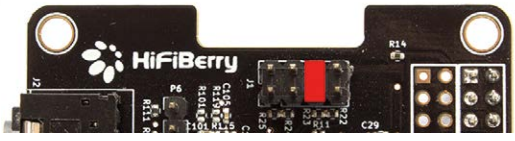
« PyKaraoke est un lecteur de karaoké efficace, fiable et sans fioritures dans le dépôt des logiciels standards de Raspbian. »

03 Mettre à jour votre noyau

Le noyau des versions actuelles de Raspbian Stretch (début 2019) ne prend pas en charge certaines fonctions du CN/A + CA/N. Heureusement, HiFiBerry a produit un noyau personnalisé version 4.19.y pour remplacer le noyau Raspbian 4.14 et mettre correctement à jour **/boot/config.txt** pour charger les pilotes adaptés pour le dispositif audio. Dans une fenêtre de Terminal, entrez `bash <(curl https://raw.githubusercontent.com/hifiberry/dacadckernel/master/update-kernel)` pour commencer le processus, qui peut durer longtemps. Si vous n'êtes pas sûr de votre version actuelle de noyau, vérifiez-la en tapant `uname -a`.

04 Connecter et tester votre matériel

Branchez votre microphone, avec les adaptateurs nécessaires, dans le port de 3,5 mm du CN/A + CA/N et connectez vos enceintes à ses ports RCA. Pour éviter les conflits, l'outil de mise à jour HiFiBerry désactive le processeur audio intégré par défaut Broadcom BCM2837 du RPi, donc seul le CN/A + CA/N devrait apparaître pendant ces essais. Tapez `aplay -l` sur la ligne de commande pour lister les périphériques de sortie, et `arecord -l` pour voir les entrées. Tapez `speaker-test -c2` pour lancer un test de bruit blanc sur les canaux gauche et droit de vos enceintes, et tapez `alsaloop` pour confirmer que votre entrée micro peut être capturée et transmise aux enceintes.



Le CN/A + CA/N HifiBerry a un pré-ampli pour micro intégré, dont vous pouvez activer le gain via un cavalier (32 dB ci-contre).

05 Installer PyKaraoke

PyKaraoke est un lecteur de karaoké efficace, fiable et sans fioritures et, par-dessus tout, on le trouve dans le dépôt des logiciels standards de Raspbian. Tapez juste `sudo apt install pykaraoke` pour installer le programme et toutes ses dépendances. Un raccourci vers l'IHM de PyKaraoke sera placé dans votre menu Son et Vidéo. Pour la prise en charge des polices dans les fichiers karaoké MIDI et dans l'ultraléger mini-lecteur PyKaraoke, tapez `sudo apt install ttf-dejavu.`

Astuce

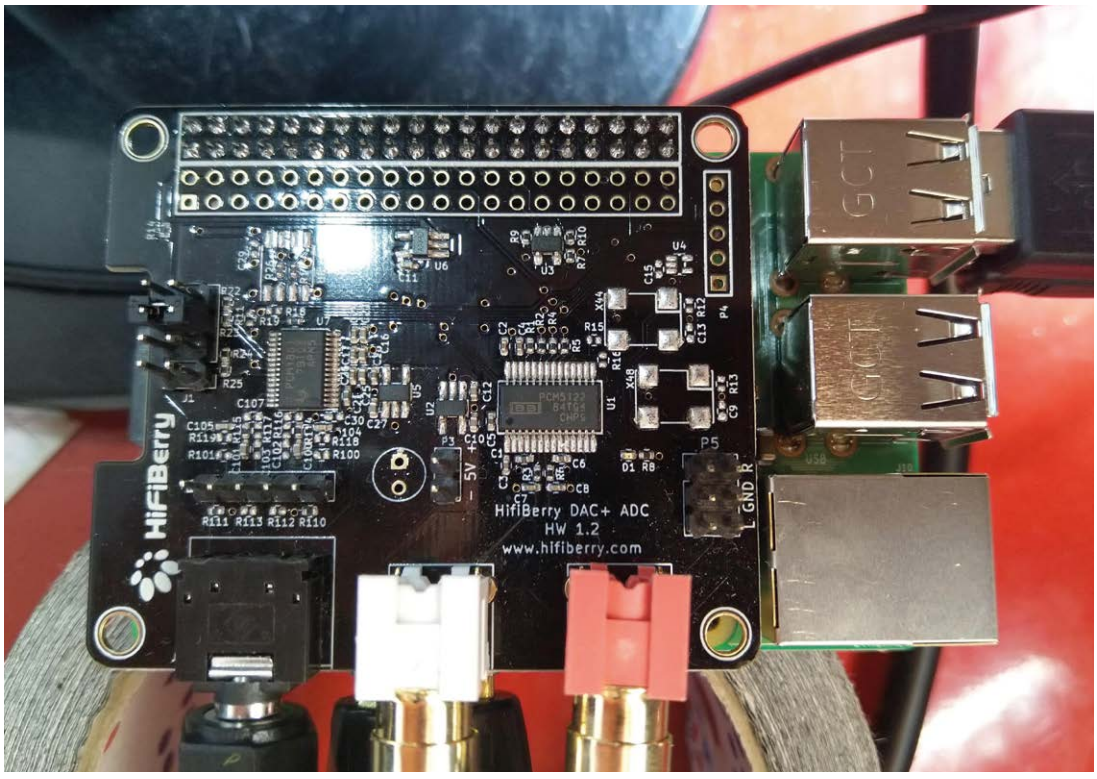
Commande du son

Installez une application d'égaliseur graphique pratique pour commander vos niveaux audio avec `sudo apt install qasmixer`.

Astuce

Double audio

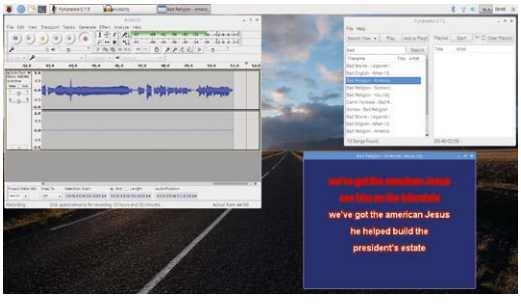
Si vous souhaitez utiliser l'audio intégré au RPi avec une liaison HDMI vers une télévision ou un récepteur AV et utiliser simultanément le HAT audio avec une seconde paire d'enceintes pour le micro, ôtez le commentaire de `dtparam=audio=on` dans **/boot/config.txt**.



Un RPi 3 avec le HAT CN/A + CA/N, connecté à des enceintes amplifiées, un micro XLR via un connecteur de 3.5 mm et un câble XLR vers TRS, un moniteur HDMI, une souris et un clavier.

06 Configurer votre retour micro

Comme beaucoup de lecteurs de karaoké, PyKaraoke n'a pas sa propre gestion de l'entrée micro, mais comme il repose sur le sous-système PulseAudio, nous allons au-devant d'un conflit de périphérique si nous essayons d'utiliser le retour ALSA de notre essai. Pour éviter cela, nous pouvons créer un retour en utilisant PulseAudio à la place. Ouvrez un Terminal et tapez `sudo apt install pulseaudio`. Puis tapez `pactl load-module module-loopback latency_msec=1`. Vérifiez que tout fonctionne normalement. N'exécutez cette commande qu'une fois par session, sinon vous créerez de multiples retours micro.



Si vous voulez enregistrez les prestations de vos amis pour la postérité, chargez Audacity pour capturer l'entrée micro pendant qu'ils chantent.

07 Trouver des chansons douces

Avant de pouvoir organiser votre grande soirée karaoké, il vous faut quelque chose à chanter. Vous pouvez acheter des pistes d'accompagnement en ligne, mais à moins d'aimer lire une partition, assurez-vous qu'elles soient accompagnées d'un fichier CDG des paroles associées sur l'écran. Des sites commerciaux comme **karaoke-version.com** ont des sélections gratuites réduites. De nombreux musiciens font don de pistes d'accompagnement karaoké, en les mettant en ligne (services de fichiers partagés, groupes de réseaux sociaux) ou en les téléchargeant sur YouTube pour que les gens les chantent en direct ou les téléchargent avec des outils comme *YouTube-downloader*. Vous pouvez même en trouver quelques-unes sur *Internet Archive*.

Où trouver des pistes d'accompagnement karaoké gratuites

Sites spécialisés :
magpi.cc/zikqUb
magpi.cc/zWYkfW

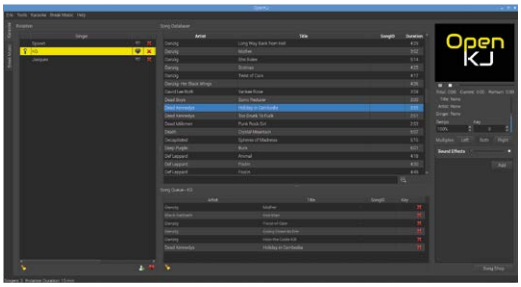
YouTube :
magpi.cc/uHMQaC
magpi.cc/frkEhY

Internet Archive :
magpi.cc/mtztmt

Facebook :
facebook.com/bostonmetalkaraoke

08 Copier vos fichiers de karaoké

Le format de karaoké le plus puissant et le plus cohérent pour travailler utilise des fichiers d'accompagnement MP3 spécialement créés avec le fichier graphique CDG correspondant pour les paroles. *PyKaraoke* prend aussi en charge des formats qui incluent des vidéos MPEG, des fichiers MIDI avec paroles intégrées, et des fichiers KAR basés sur MIDI. Créez un répertoire Karaoke dans



▲ OpenKJ fournit une interface bien plus sophistiquée, mais il charge beaucoup le système et peut provoquer des problèmes de performance pour certains fichiers si vous ne surfréquencez pas, ni ne refroidissez votre RPi.

votre dossier d'accueil pour contenir vos fichiers – vous pouvez en outre les ranger par format, artiste ou alphabétiquement.

09 Configurer PyKaraoke

Ouvrez PyKaraoke, allez à *File > Add new songs to database*. Ajoutez le dossier Karaoke créé à l'étape précédente, cliquez sur *Scan Now* et, lorsque c'est fini, cliquez sur *Save* et *Close*. Il y a aussi une option *Configure* dans le menu *File* pour ajuster vos autres réglages. Vous pouvez y régler la fenêtre des paroles pour qu'elle s'ouvre en plein écran, définir les fréquences d'échantillonnage par défaut et choisir un lecteur vidéo externe pour traiter les vidéos de karaoké en MPEG et AVI. Comme vous pourrez changer les réglages par défaut plus tard, vous pouvez tranquillement les laisser pour commencer.

10 À vos marques, prêts, chantez !

Après l'ajout de tous vos fichiers, tapez juste un mot ou une lettre dans la barre de recherche de PyKaraoke pour voir toutes les correspondances. Faites glisser les pistes vers le cadre de la liste de lecture à droite pour les mettre en file d'attente avant d'appuyer sur *Start*, ou cliquez juste sur *Play* pour les jouer immédiatement. Votre micro est déjà activé et prêt à fonctionner, grâce à notre retour par PulseAudio. Si vous avez du mal à trouver une chanson, cliquez sur le bouton *Search View* et basculez sur *Folder View*, où vous pouvez naviguer dans tout votre système de fichiers à la recherche de fichiers compatibles.

11 Retour micro dès le chargement

Enfin, partisans du moindre effort, nous créons un script Bash pour configurer automatiquement notre retour micro et ouvrir PyKaraoke. Avec un éditeur de texte, créez un nouveau fichier nommé *karaoke* dans votre répertoire d'accueil. Notre petit encadré de code vous montre comment formater les commandes. Sur la ligne de commande, tapez `chmod u+x karaoke` pour le rendre exécutable, et utilisez l'éditeur du Menu Principal pour l'ajouter par commodité comme nouvel item à *Son et Vidéo*. M

(VF : Denis Lafourcade)

Devenez pro avec OpenKJ

Même si PyKaraoke est plus que suffisant pour chanter tout seul à la maison, si vous voulez organiser une soirée karaoké plus sophistiquée, OpenKJ a une meilleure interface, des listes de lecture pour plusieurs chanteurs et un large éventail de fonctions supplémentaires.

Il fonctionne, mais atteint parfois les limites du processeur du RPi 3B+. Si vous souhaitez tester ses fonctions avancées, suivez ces instructions pour le compiler et l'installer à partir des sources.

Installez les dépendances audio :

```
sudo apt install pulseaudio
gstreamer1.0-pulseaudio
gstreamer1.0-tools gstreamer1.0-
plugins-good gstreamer1.0-
plugins-ugly gstreamer1.0-
plugins-bad libgstreamer-
plugins-base1.0-dev
```

Installez QT5 :

```
sudo apt install qt5-default
qtcreator qtmultimedia5-
dev libqt5multimediawidgets5
libqt5multimedia5-plugins
libqt5multimedia5 libqt5svg5-dev
qt5-qmake
```

Installez libtag :

```
sudo apt install libtag1-dev
libtag-extras1 libtag-extras-dev

git clone https://github.com/
OpenKJ/OpenKJ.git/.
cd OpenKJ/OpenKJ
/usr/lib/arm-linux-gnueabihf/
qt5/bin/qmake
make

sudo apt install checkinstall
sudo checkinstall
```

Checkinstall installera par défaut OpenKJ et créera un package .deb que vous pourrez utiliser ultérieurement.

« Le format karaoké le plus puissant et le plus cohérent pour travailler utilise des fichiers d'accompagnement MP3 spécialement créés avec le fichier graphique CDG correspondant pour les paroles. »

karaoke

▸ Langage : **Bash**

```
001. #!/bin/bash
002.
003. pactl load-module module-loopback latency_msec=1
004. pykaraoke
```

▲ Ce petit script active votre retour micro avant de lancer le lecteur de karaoké.

Images de l'ISS par ondes radio

Que diriez-vous de recevoir une carte postale de la Station spatiale internationale par radio ? Plutôt excitant, non ?

MAKER

Dave Honess

Dave vit aux Pays-Bas et travaille au Centre européen de technologie spatiale, où il coordonne les programmes éducatifs réalisés par l'équipage européen de l'ISS.

@dave_spice

Un de mes passe-temps consiste à capter les images que la Station spatiale internationale (ISS) transmet par télévision à balayage lent (SSTV, *Slow Scan Television*) depuis la station radioamateur installée dans le cadre du programme ARISS (*Amateur Radio on the ISS*). Cette station n'est pas toujours active, mais lorsqu'elle l'est les émissions durent de quelques heures à quelques jours. Mais quel est donc le rapport entre une station radioamateur et la télévision à balayage lent ? Il vient du principe même de la SSTV, qui consiste à convertir une image en son afin de pouvoir la transmettre par ondes radio. La bande passante utilisée étant de l'ordre de 3 kHz, les signaux reçus sont audibles et ressemblent aux tonalités d'un fax ou d'un (vieux) modem « 56k ».

Ingrédients

- RPi 2B ou plus récent
- Haut-parleurs ou casque (jack 3.5 mm)
- Clé RTL-SDR (cherchez RTL2832U) avec une bonne antenne fouet ou dipôle (ou utilisez ce kit : **www.elektor.fr/18949**).
- Mètre ruban d'au moins 1 m.

01 Préparatifs

Je supposerai que vous disposez d'une version à jour de Raspbian Stretch avec environnement de bureau.

Commençons par basculer la sortie audio sur la prise casque de 3,5 mm. Cette étape est nécessaire en raison de la façon dont PulseAudio traite l'audio du système en tant que périphérique d'entrée. Faites donc un clic droit sur l'icône représentant un haut-parleur à droite de la barre de menu, et sélectionnez *Analog*.

Ouvrez ensuite un terminal et installez les logiciels nécessaires au projet :

```
sudo apt-get update
sudo apt-get install rtl-sdr sox
pulseaudio qsstv ntpdate -y
```

Ceci fait, assurez-vous que votre fuseau horaire et votre localisation sont corrects depuis l'onglet *Localisation* de *Menu > Préférences > Configuration*.

▲ Voici l'une des images les plus nettes que j'ai obtenues avec le matériel de ce tutoriel (reçue le 15 février 2019 près d'Amsterdam avec une élévation de l'ISS de 83°).

Revenez au terminal, et synchronisez l'heure de votre système avec un serveur NTP :

```
sudo ntpdate pool.ntp.org
```

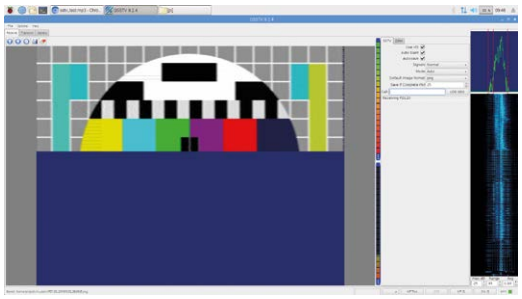
Notez que l'horloge de la barre de menu ne se met à jour que toutes les minutes.

02 Test de la clé RTL-SDR

Insérez la clé RTL-SDR dans votre RPi, et reliez l'antenne à la clé (à ce stade, peu importe son orientation). Vérifiez le bon fonctionnement de la clé en lançant le programme `rtl_test` depuis un terminal :

```
rtl_test
```

Si le message *failed to open rtlsdr device* apparaît, retirez la clé et relancez rtl_test. Laissez-le tourner une trentaine de secondes, et voyez si un message signale une perte d'échantillons. Une perte d'environ 100 octets est acceptable. Arrêtez le test avec **Ctrl-C**.



▲ Le décodage d'une image prend environ deux minutes. C'est aussi la durée typique d'une transmission d'image depuis l'ISS, et également l'intervalle séparant une transmission de la suivante.

03 Test avec une station FM

Testons maintenant à la fois la clé et l'antenne avec une radio FM. La commande ci-dessous envoie les données brutes de la sortie de `rtl_fm` à l'entrée de `play`, qui produit la sortie audio. Modifiez l'argument de l'option `-f` en fonction de la fréquence de la radio servant à votre essai ; n'oubliez pas le `M` majuscule, il spécifie le multiple des hertz (87.8 MHz correspond à France Inter à Paris) :

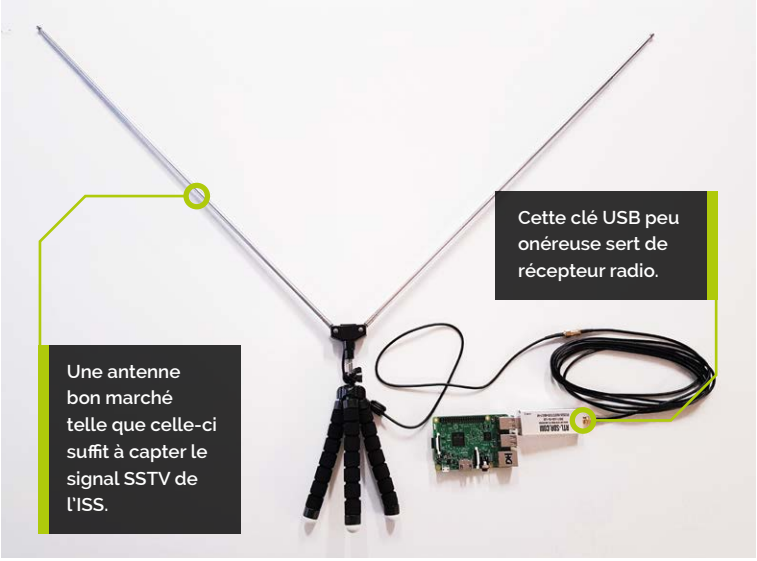
```
rtl_fm -M wbfm -f 98.8M | play -r 32k -t raw -e s -b 16 -c 1 -V1 -
```

Notez que rtl_fm s'accorde sur une fréquence différente pour éviter les « pointes CC ». Ce décalage à dessein vient de l'étage matériel des radios logicielles et peut être ignoré. La station FM devrait être audible via votre casque ou haut-parleur. Bougez l'antenne s'il y a des parasites, ou choisissez une station dont vous savez l'émetteur plus proche. Quittez rtl_fm avec **Ctrl-C**.

04 Configuration de QSSTV

Nous utiliserons le logiciel QSSTV. Son rôle sera d'écouter la sortie audio du RPi et de décoder le son des images lorsque la station ISS sera à portée d'antenne. Assurez-vous d'abord que QSSTV est bien configuré. Depuis le menu principal, sélectionnez *Internet > QSSTV*, puis *Options > Configuration*. Dans l'onglet *Sound*, vérifiez que les options ci-dessous sont activées comme indiqué :

Audio Interface – PulseAudio
Input and Output Audio Device – default -- Playback/recording through the PulseAudio sound server
Sound Input – From sound card
Sound Output – To sound card



Validez avec *OK*, puis ouvrez *Options > Calibrate*. Soyez patient, le calibrage prend plusieurs minutes. Cliquez sur OK lorsque les deux barres de progression ont atteint 99 %.

05 Test de décodage d'une image

QSSTV étant configuré, téléchargeons un fichier audio test pour vérifier qu'il se transforme bien (lentement) en image. Ouvrez un terminal et entrez :

```
wget https://raw.githubusercontent.com/davidhoness/sstv_decoder/master/sstv_test.mp3
```

Dans l'onglet *Receive* de QSSTV, activez les paramètres ci-dessous comme indiqué :

Use VIS – on
Auto Slant – on
Autosave – on
Signals – Normal
Mode – Auto

Même s'il ne se passera rien, cliquez sur le bouton *play* ▶ pour activer le récepteur.

Ouvrez le fichier **sstv_test.mp3** dans le navigateur Chromium, soit par glisser-déposer depuis le gestionnaire de fichiers, soit en entrant **file:///home/pi/sstv_test.mp3** dans la barre d'adresse.

Sa lecture s'accompagne de l'apparition d'une image dans QSSTV. Notez, à droite de l'écran, l'évolution du signal et du spectrogramme. On trouve en ligne d'autres fichiers audio d'essai, p. ex. ici : **magpi.cc/dXpjRr**.

Astuce

SSTV sur mobile

Ces applis gratuites permettent un test rapide : Robot36 pour Android, et CQ SSTV pour iOS. Lisez ce MP3 (**magpi.cc/ruANWC**) et placez le mobile près du haut-parleur.

Astuce

Kit RTL-SDR

Je recommande ce kit de démarrage contenant une clé USB RTL-SDR et une antenne en V de bonne qualité : **magpi.cc/uBMKQg**

06 Informations nécessaires

Trois informations sont nécessaires pour recevoir des images de l'ISS :

- les dates d'activités SSTV (l'émission radio n'est pas permanente).
- La fréquence prévue du signal SSTV (généralement 145,8 MHz).
- Les heures de passage de l'ISS au-dessus de chez vous.

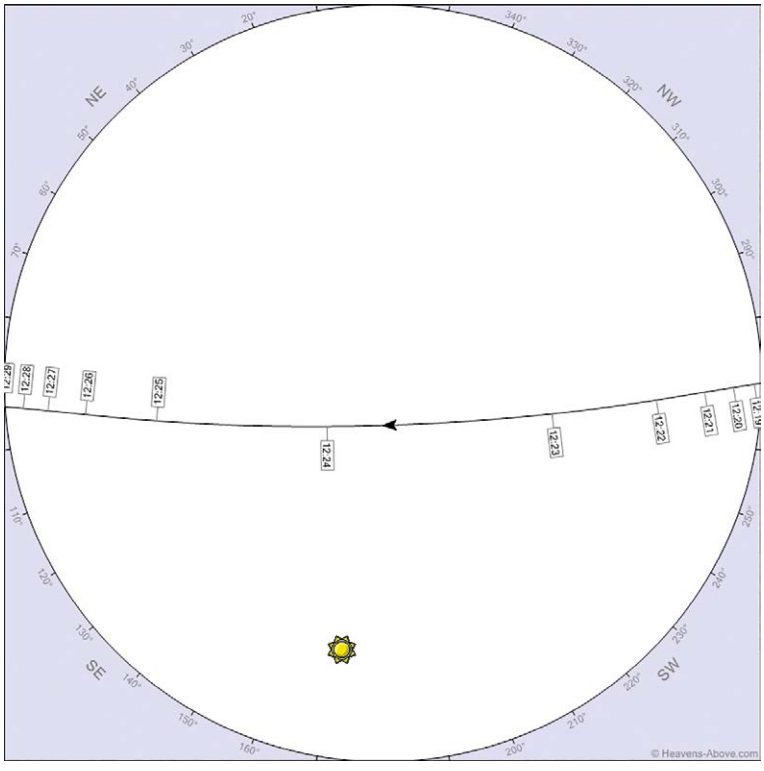
À moins que vous n'habitiez en Arctique ou en Antarctique, il y a de fortes chances que la station ISS soit visible depuis votre lieu d'habitation plusieurs fois par jour, traversant le ciel à 28000 km/h à une altitude moyenne de 400 km. Les dates des prochaines émissions SSTV sont annoncées sur **ariss-sstv.blogspot.com** (en anglais) ainsi que sur **www.ariss-f.org** (en français).

07 Calendrier des passages de l'ISS

De nombreux sites fournissent les heures de passage de l'ISS en fonction de vos coordonnées géographiques. Mon préféré est **heavens-above.com** (en français).

En haut et à droite de la page d'accueil, cliquez sur le lien *Unspecified* du paramètre *Localisation*. Entrez le nom de votre commune suivi de son code postal, et cliquez sur *Recherche*. Cliquez sur *Actualiser* pour revenir à la page d'accueil. Sous *Satellites*, cliquez sur *ISS (Station Spatiale Internationale)*. Pour le paramètre *Inclure les passages:*, cochez « *tous* ».

Le tableau indique les prochaines dates et heures de passage de l'ISS en fonction de votre localisation. La valeur Elev de la colonne Culmination indique l'élévation maximale au-dessus de l'horizon. Une hauteur de 90° correspond à un passage directement au-dessus de chez vous. Plus cette élévation est grande, meilleure est la réception. Cliquez sur une rangée du tableau pour voir le trajet de l'ISS sur une carte du ciel. Si vous teniez cette carte au-dessus de votre tête et aligniez son Nord avec le vôtre, la ligne indiquerait la trajectoire de l'ISS dans votre ciel, une information utile à la bonne orientation de l'antenne. Le lien *Trajectoire au sol* (en haut à droite) visualise cette trajectoire de façon peut-être plus intuitive, avec en plus une indication horaire.



Imaginez-vous tenir cette carte au-dessus de votre tête en l'alignant selon les directions d'une boussole. La ligne indiquerait la trajectoire de l'ISS.

08 Réglage de l'antenne

Si vous utilisez le kit RTL-SDR avec clé et antenne, lisez le guide de démarrage : **magpi.cc/qzUKtZ**.

L'ajustement de la longueur des bras de l'antenne à la fréquence de résonance de 145,8 MHz permet de mieux capter les signaux faibles lorsque l'élévation est basse.

J'utilise pour cela le calculateur en ligne **magpi.cc/uGNNxP**. Pour *Desired Frequency*, entrez 145.8. Pour *Select Antenna Calculation*, sélectionnez *One Side*. Cliquez sur *Calculate*, la réponse devrait être 48.9 cm.

Comme il y a environ 2 cm de métal de chaque côté de la base de l'antenne, il faut en fait prendre 46,9 cm (48,9 - 2) comme longueur de bras (sans tenir compte du filetage des vis). Ce n'est pas au dixième de millimètre près, mais essayez d'approcher au mieux cette valeur avant de visser les bras.

09 Déploiement de l'antenne

Essayez de placer votre antenne aussi haut que possible et à un endroit offrant une bonne visibilité du ciel dans toutes les directions. Une antenne extérieure est préférable, mais elle captera aussi le signal SSTV depuis l'intérieur

d'une maison – il y aura juste plus de bruit dans les images. Posez ses bras à plat, alignez-les avec la trajectoire de l'ISS, et donnez-leur la forme d'un V vertical dont l'angle fait 90°. Vous obtiendrez de meilleurs résultats si vous réalignez votre antenne à chaque passage de l'ISS, mais ce n'est pas nécessaire.

Pour trouver la meilleure accroche, jouez avec l'orientation du V à chaque passage de l'ISS, en mettant p. ex. un bras horizontal et l'autre vertical. Trouver la bonne configuration relève parfois plus de l'art que de la science.

10 Resynchronisation de l'horloge

QSSTV enregistre les images décodées avec un nom de fichier indiquant l'heure UTC, une information utile pour retrouver l'origine d'une image. Avant chaque session, entrez donc à nouveau :

```
sudo ntpdate pool.ntp.org
```

11 Écoute de la fréquence ISS

La station ISS utilisant une bande FM étroite pour ses émissions SSTV, nous devons lancer la commande `rtl_fm` avec des paramètres différents de ceux utilisés ci-dessus pour la bande large des radios FM :

```
rtl_fm -M fm -f 145.8M -s 48k | play -r
48k -t raw -e s -b 16 -c 1 -V1 -
```

Vous devriez percevoir un bruit blanc, qui bien sûr se transformera en tonalités caractéristiques lorsque vous capterez le signal SSTV. Laissez ouvert votre terminal. La durée de transmission d'une image est d'environ deux minutes, une pause de deux minutes étant observée entre deux transmissions.

12 Lancement de QSSTV

Lancez QSSTV en cliquant sur *Menu > Internet > QSSTV*. Sélectionnez l'onglet *Receive* du programme et paramétrez-le comme suit :

Use VIS – on
Auto Slant – on
Autosave – on
Mode – Auto
Save if Complete (%) – 10

Cliquez sur le bouton de lecture ►. La forme d'onde et le spectrogramme du bruit provenant de `rtl_fm` devraient s'afficher. Laissez QSSTV ouvert.

▲ L'antenne VHF ARISS est installée dans le bas du module Columbus. Crédit : NASA/ESA.

> « Trouver la configuration idéale d'une antenne relève parfois plus de l'art que de la science. »

13 Décalage Doppler

Lorsqu'un camion de pompier s'approche de vous, la fréquence perçue de sa sirène augmente à mesure qu'il se rapproche, est identique à celle de la fréquence émise lorsqu'il passe devant vous, et diminue lorsqu'il s'éloigne. Ce phénomène, valable aussi bien pour les ondes sonores qu'électromagnétiques, est appelé décalage Doppler.

L'ISS se déplaçant à 7 km/s, ce décalage doit être compensé. C'est ce que font les récepteurs des stations au sol en s'accordant à ~3,5 kHz au-dessus de 145,8 MHz à l'approche de l'ISS, sur 145,8 MHz lorsqu'elle passe au zénith de la station, et sur ~3,5 kHz au-dessous de 145,8 MHz lorsqu'elle s'éloigne. Heureusement pour nous, QSSTV compense automatiquement ce décalage en suivant la fréquence centrale du signal audio.

Astuce

Antennes à éviter

Évitez les petites antennes à montage magnétique de 30 cm dont sont souvent équipées les clés RTL-SDR, elles ne capteront probablement pas l'ISS.

▲ Un certificat peut vous être décerné même si vous ne décodez que quelques lignes d'une image.

14 Compensation du décalage Doppler (optionnel)

Cela ne changera pas grand-chose, mais j'ai écrit, juste pour le plaisir, un script Python (**magpi.cc/zNtRDB**) compensant le décalage Doppler. Il utilise le module `ephem` pour le suivi de l'ISS, calcule la fréquence corrigée en fonction de son passage, et réajuste `rtl_fm` au moyen d'un socket UDP. Pour l'utiliser, installez d'abord ephem avec `sudo pip3 install ephem`, puis exécutez le script depuis un autre (onglet de) terminal. Notez qu'il vous faudra adapter les valeurs des constantes LATITUDE, LONGITUDE et ALTITUDE du code à vos coordonnées géographiques (fournies par **heavens-above.com** ; estimez votre altitude).

15 Elle arrive !

L'arrivée en direct de l'ISS et d'une image s'accompagne toujours d'un petit moment d'excitation, mais ses horaires de passage ne sont pas forcément compatibles avec nos activités (le sommeil p. ex.) Notre présence n'est heureusement pas indispensable puisque QSSTV enregistre les images reçues dans l'onglet *Gallery*.

16 Recevez un prix SSTV !

Sur votre RPi, les images décodées sont sous */home/pi/qsstv/rx_sstv*. Si vous souhaitez participer à l'amélioration de la station radioamateur de l'ISS, vous pouvez les soumettre ici (cf. la FAQ) : **magpi.cc/Kdjnmy**. Le site récompense certains contributeurs en leur envoyant par courriel un « ARISS SSTV Award », une image PNG à numéro unique. Bonne chance !

(VF : Hervé Moreau)

doppler.py

▶ Langage : **Python**

```
001. #!/usr/bin/python3
002. import urllib.request
003. import math
004. import socket
005. import ephem
006. import time
007. import sys
008. import ssl
009.
010.
011. C = 300000000.0
012. F0 = 145800000.0
013.
014. LATITUDE = "52.219308"
015. LONGITUDE = "4.419926"
016. ALTITUDE = 20
017.
018.
019. class tle_reader(object):
020.     """
021.     For keeping ephem two line element sets up to date
022.     """
023.     def __init__(self,
024.                  tle_name="ISS (ZARYA)",
025.                  tle_file=
     "https://celestrak.com/NORAD/elements/stations.txt",
026.                  tle_max_age=3600):
027.         self._tle_name = tle_name
028.         self._tle_file = tle_file
029.         self._tle_max_age = tle_max_age
030.         self._tle = None
031.         self.reload()
032.
033.     def build_index(self, tle_lines):
034.         index = {}
035.         for i in range(0, len(tle_lines), 3):
036.             index[tle_lines[i].strip()] = (tle_lines[i +
     1], tle_lines[i + 2])
037.         return index
038.
039.     def reload(self):
040.         print("Loading: %s" % self._tle_file)
041.
042.         try:
043.             ctx = ssl.create_default_context()
044.             ctx.check_hostname = False
045.             ctx.verify_mode = ssl.CERT_NONE
046.             with urllib.request.urlopen(self._tle_file,
     context=ctx) as response:
```

TÉLÉCHARGEZ LE CODE COMPLET :

magpi.cc/zNtRDB

```
                tle_lines = response.read().decode("utf-8").splitlines()
            index = self.build_index(tle_lines)
            tle_data = index[self._tle_name]
            self._tle = ephem.readtle(self._tle_name, tle_data[0], tle_data[1])
        except Exception as e:
            print(e)

        self._tle_age = time.time()

    @property
    def tle(self):
        return self._tle

    @property
    def tle_expired(self):
        return time.time() - self._tle_age > self._tle_max_age


class rtl_fm_remote(object):
    """
    For remote control of rtl_fm command line program
    """
    def __init__(self,
                 host="localhost",
                 port=6020):
        self._host = host
        self._port = port
        self._s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
        self._s.connect((self._host, self._port))

    def set_freq(self, freq):
        self.send_cmd(0, freq)

    def set_mode(self, mode):
        self.send_cmd(1, mode)

    def set_squelch(self, squelch):
        self.send_cmd(2, squelch)

    def set_gain(self, gain):
        self.send_cmd(3, gain)

    def send_cmd(self, cmd, param):
        cmd_bytes = (cmd).to_bytes(1, "little")
        param_bytes = (param).to_bytes(32, "little")
        self._s.send(cmd_bytes + param_bytes)

    def __del__(self):
        self._s.close()


rtl = rtl_fm_remote()
iss = tle_reader(tle_name="ISS (ZARYA)", tle_max_age=5520)

if iss.tle is None:
    sys.exit(0)

myloc = ephem.Observer()
myloc.lon = LONGITUDE
myloc.lat = LATITUDE
myloc.elevation = ALTITUDE

freq = F0
running = True

try:
    while running:
        myloc.date = time.strftime('%Y/%m/%d %H:%M:%S', time.gmtime())
        iss.tle.compute(myloc)
        alt = math.degrees(iss.tle.alt)

        if alt > 0:  # iss is flying over location
            new_freq = int(F0 - iss.tle.range_velocity * F0 / C)  # doppler
            if new_freq != freq:
                print(new_freq, round(alt, 2), myloc.date)
                rtl.set_freq(new_freq)  # set new frequency in rtl_fm
            freq = new_freq
        elif iss.tle_expired:
            iss.reload()  # we could be running for days / weeks
        else:
            time.sleep(10)  # do nothing, wait for iss to arrive
            freq = F0
except KeyboardInterrupt:
    running = False

print("Bye")
```

Électronique avec GPIO Zero 1.5

Prenez un carillon de porte sans fil du commerce, alimenté par pile, et modifiez-le pour lui ajouter de nouvelles fonctions, une caméra et un buzzer à plusieurs tons pour jouer votre propre jingle.

MAKER

Ben Nuttall

Créateur de GPIO Zero et de *piwheels*, Ben est l'expert du Python à la Fondation Raspberry Pi.

@ben_nuttall

Dorénavant la bibliothèque GPIO Zero pour Python prend en charge la production de mélodies avec des buzzers, c'est donc l'occasion parfaite pour créer votre nouveau jingle de sonnette.

Ingrédients

- Raspberry Pi Zero ou 3A+ **www.elektor.fr/rpi**
- Module Camera Raspberry Pi **www.elektor.fr/pi-camera-module-2**
- HAT Jam, ou buzzer et transistor **magpi.cc/jMxZCU**
- Carillon de porte ordinaire sans fil (marque Wilko) **magpi.cc/pfUhqo**

01 Mettre à jour GPIO Zero

La bibliothèque GPIO Zero v1.5 vient juste de sortir ! Elle amène quelques nouvelles fonctions utilisées dans ce tutoriel, alors pensez à la mettre à jour ! Vérifiez que votre RPi est connecté, ouvrez un Terminal et tapez :

```
sudo apt update
sudo apt install python3-gpiozero
```

Ces commandes récupèrent la dernière version de GPIO Zero, et installent aussi python3-colorzero, car c'est une nouvelle dépendance de GPIO Zero.

Vérifiez votre la version en tapant :

```
apt-cache policy python3-gpiozero
```

Vous serez aussi informé de la disponibilité d'une nouvelle version.

02 Buzzer à plusieurs tons

`TonalBuzzer` est une des nouvelles classes de périphérique dans GPIO Zero, qui permet de jouer des tonalités particulières en variant la fréquence MLI. Vous pouvez jouer une séquence de notes pour produire une mélodie, ou bien créer d'amusants effets sonores, comme une sirène de police, en parcourant des gammes de fréquences à différentes vitesses. Rien à voir avec la qualité que vous obtiendriez avec un haut-parleur, mais il est très possible de réaliser des sons reconnaissables et des jingles. Il y a un buzzer tonal sur le nouveau HAT Jam de ModMyPi, ou vous pouvez utiliser un buzzer ordinaire comme celui du kit CamJam, mais vous aurez de meilleurs résultats si vous prenez un transistor pour appliquer le 5 V au buzzer, comme sur le HAT Jam.

La sonnette Wilko à 12 € est livrée avec un carillon déporté alimenté par pile.

03 Jouer une mélodie

Ouvrez d'abord un *shell* Python ou le REPL dans votre éditeur Python préféré, importez tout ce qu'il vous faut, puis créez un objet `TonalBuzzer` associé à la broche GPIO à laquelle il est connecté :

```
from gpiozero import TonalBuzzer
from gpiozero.tones import Tone
from gpiozero.tools import sin_values
from time import sleep

tb = TonalBuzzer(20)
```

Maintenant essayez de jouer une seule note :

```
tb.play(60)
```

Le buzzer joue la note MIDI 60 (Do3). Vous aurez une alerte « *ambiguous tone* », mais ne vous inquiétez pas ; c'est juste parce que l'interface `Tone` autorise à utiliser les notes MIDI, les fréquences ou la notation musicale. Pour être plus explicite, vous devriez généralement utiliser `Tone(midi=60)` plutôt que juste 60 ou `Tone(60)`. Tapez `tb.stop()` pour interrompre la lecture. Tapez `Tone(60)` dans le REPL pour voir ses trois représentations :

```
<Tone note='C4' midi=60 frequency=261.63Hz>
```

Examinez l'intérieur.

Essayez de jouer une gamme :

```
for note in 'C4 D4 E4 F4 G4 A4 B4 C5'.
split():
    tone = Tone(note)
    print(repr(tone))
    tb.play(tone)
    sleep(0.3)
tb.stop()
```

Enfin, pour l'effet, essayons ce bruit de sirène dont nous vous avons parlé :

```
tb.source = sin_values()
```

En passant simplement une série de valeurs de l'onde sinusoïdale, le buzzer va varier continuellement sa fréquence d'une octave du bas vers le haut, produisant un effet de sirène. Pour l'accélérer ou le ralentir, modifiez le `source_delay` du buzzer (par défaut à 0,1) :

```
tb.source_delay = 0.01
```

...ou

```
tb.source_delay = 0.5
```

04 Commande par le clavier

Installez maintenant la bibliothèque d'entrées (`inputs`), c'est un utilitaire précieux pour lire (entre autres choses) les pressions de touches en temps réel. Ouvrez un Terminal et tapez :

```
sudo pip3 install inputs
```

Revenez au shell Python, importez `get_key`, et créez une table de correspondance de chacun des caractères de la rangée du milieu de votre clavier avec les notes :

Câblage entre la sonnette et les broches GPIO du RPi.

> « Créez une table de correspondance entre chacun des caractères de la rangée du milieu de votre clavier et les notes. »

```
from inputs import get_key
keys = {'Q': 'C4', 'S': 'D4', 'D': 'E4',
'F': 'F4', 'G': 'G4', 'H': 'A4', 'J': 'B4',
'K': 'C5'}
```

Ajoutez une boucle pour scruter les pressions de touches et jouer les notes correspondantes :

```
while True:
events = get_key()
for event in events:
if event.ev_type == 'Key' and event.code[-1]
in 'QSDFGHJK':
if event.state:
tb.play(keys[event.code[-1]])
else:
tb.stop()
```

Pressez maintenant n'importe quelle touche de **Q** à **K**, le buzzer jouera une note. Vous avez transformé le clavier de votre ordinateur en un clavier musical ! Essayez de jouer une mélodie.

05 Jingle de sonnette personnalisé

Nous avons créé la mélodie du thème de la Panthère Rose. Utilisez-la, ou profitez de l'occasion pour créer la vôtre, en notes MIDI ou notation musicale. Vous pouvez opter pour l'effet sirène de police, mais cela pourrait être plus inquiétant que nécessaire lorsqu'il s'agit juste d'une livraison de commande d'Amazon.

Pour la Panthère rose, voici une fonction pour prendre une liste de notes et de durées et les jouer l'une après l'autre.

Astuce

Paradigmes de programmation

Essayez différents styles de programmation – GPIO Zero fournit différentes approches, comprenant le procédural, le mode bloqué et les fonctions de rappel. Lisez la documentation pour une meilleure compréhension.

Astuce

Soudage

Soyez prudent en soudant – n'allez pas accidentellement dessouder d'autres composants utiles du circuit imprimé de la sonnette. Prenez votre temps.

```
def play(tune):
      for note, duration in tune:
          print(note)
          tb.play(note)
          sleep(float(duration))
      tb.stop()

tune = [('C#4', 0.2), ('D4', 0.2), (None, 0.2),
    ('Eb4', 0.2), ('E4', 0.2), (None, 0.6),
    ('F#4', 0.2), ('G4', 0.2), (None, 0.6),
    ('Eb4', 0.2), ('E4', 0.2), (None, 0.2),
    ('F#4', 0.2), ('G4', 0.2), (None, 0.2),
    ('C4', 0.2), ('B4', 0.2), (None, 0.2),
    ('F#4', 0.2), ('G4', 0.2), (None, 0.2),
    ('B4', 0.2), ('Bb4', 0.5), (None, 0.6),
    ('A4', 0.2), ('G4', 0.2), ('E4', 0.2),
    ('D4', 0.2), ('E4', 0.2)]

play(tune)
```

06 Test de la caméra

Au début, le plus simple pour tester la caméra est d'ouvrir un Terminal et de taper :

```
raspistill -k
```

Cette commande ouvre la prévisualisation de la caméra jusqu'à ce que vous la tuiez avec **CTRL+C**. Revenons à Python ; écrivez un bout de code simple pour déclencher une prise de vue dans une boucle :

```
from picamera import PiCamera
from datetime import datetime

camera = PiCamera()

while True:
    input("Press Enter to capture")
    dt = datetime.now().isoformat()
    camera.capture('/home/pi/{}.jpg'.format(dt))
```

Ce code prend une photo lorsqu'on appuie sur **Entrée**, et l'enregistre avec l'horodatage courant..

07 Modifier la sonnette

Note : ces instructions ne concernent que les sonnettes alimentées par piles. N'UTILISEZ PAS une sonnette alimentée par le secteur.

Ouvrez le carillon de la sonnette en dévissant le boîtier pour découvrir le circuit imprimé à l'intérieur. Vous remarquerez un fil blanc avec une extrémité libre (c'est l'antenne radio), des fils rouge et noir qui vont au bloc-piles, d'autres fils blancs qui vont au haut-parleur et une autre paire de fils rouge et noir qui vont à la LED sur le devant du boîtier. Lorsqu'on appuie sur la sonnette et que le signal est reçu, la LED s'allume et le carillon sonne.

Vous pouvez aussi choisir de recâbler l'alimentation du récepteur pour qu'elle provienne du RPi au lieu des piles. Comme il y a deux piles AA (1,5 V chacune), il faut environ 3 V. Vous pouvez l'alimenter avec le RPi en connectant le fil noir à GND et le rouge au 3,3 V (nous avons soudé les sorties directement sur un trou 3,3 V d'un RPi Zero). Et bien sûr, si vous préférez votre propre jingle, vous pouvez déconnecter le haut-parleur.

Ensuite, dessoudez les fils qui vont à la LED et connectez-les au RPi à la place : le rouge sur une broche GPIO et le noir à GND (vous pouvez couper des fils de liaison en deux et les souder aux fils, et connecter l'extrémité femelle du fil de liaison aux broches du RPi, ou même souder les extrémités directement sur les trous GPIO du RPi Zero, comme nous l'avons fait). Conseil : testez tout avant de vous engager dans le soudage de l'ensemble, alors allez-y doucement.

08 Tester la sonnette

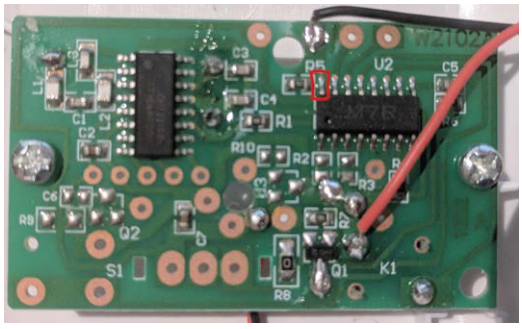
Est-ce que ça a marché ? Voyons voir. En théorie, vous devriez pouvoir détecter un signal sur la broche GPIO connectée à la sonnette, de la même façon que vous détecteriez la pression d'un bouton. Donc, retour à Python et GPIO Zero :

```
from gpiozero import Button

doorbell = Button(21)

button.wait_for_press()
print("Pressed")
button.wait_for_release()
print("Released")
```

Appuyez maintenant sur la sonnette, le message « *pressed* » devrait apparaître, puis « *released* » lorsque le carillon arrête de sonner. Si ça ne fonctionne pas, vérifiez votre câblage. Vous devrez peut-être ajouter une résistance pour garantir la détection d'un changement de tension quand le bouton est pressé. Nous avons constaté qu'en câblant la nôtre directement sur le circuit intégré (carré rouge sur la photo), la résistance intermédiaire agit.

09 Assemblage de l'ensemble

Assemblons maintenant tous les composants : la sonnette, le buzzer et la caméra.



▲ Câblage entre la sonnette et les broches GPIO du RPi.

Si un RPi doit être monopolisé pour ce projet, prenez un RPi Zero W ou 3A+. Avec un RPi Zero, vous pouvez souder les composants directement sur les trous GPIO. L'accès aux broches GPIO sera plus facile avec le 3A+ ou un autre RPi (plus) plein format, mais cela prendra plus de place, ce qui peut être déterminant si vous essayer de tout mettre dans un petit volume (en particulier le boîtier d'origine du récepteur de sonnette).

Connectez votre HAT Jam ou votre buzzer, et soudez les fils de votre récepteur de sonnette aux broches GPIO du RPi. Connectez la caméra et trouvez un moyen de la placer. Si vous avez une vitre sur votre porte d'entrée, vous pouvez coller le RPi derrière ou à côté de la porte, et tourner la caméra vers l'extérieur. Vérifiez que toutes les connexions sont solides et que le boîtier maintient tout en place, puis connectez l'alimentation du RPi, en vous assurant qu'on ne peut pas l'arracher. Fixez la sonnette à la porte d'entrée et rassemblez votre code pour le faire tourner ! Voir le code complet dans le listage **doorbell.py**.

10 Exécution au démarrage

La touche finale est importante : exécuter le programme au démarrage. Il y a quelques bonnes façons de faire, et si voulez en trouver d'autres, demandez juste à quelqu'un comment il ferait – il y a des chances que vous ayez à chaque fois des réponses différentes. Pour lancer des scripts aussi élémentaires que celui-ci, l'auteur préfère **cron**, avec **crontab** si vous savez comment faire, sinon nous recommandons d'installer **GNOME Schedule**, une interface graphique pour cron :

```
sudo apt install gnome-schedule
```

Ajoutez juste une nouvelle tâche récurrente lancée au démarrage. La tâche devrait ressembler à ceci :

```
python3 /home/pi/doorbell.py &
```

doorbell.py

> Langage : **Python 3**

TÉLÉCHARGEZ LE CODE COMPLET :
magpi.cc/github81

```
001. from gpiozero import TonalBuzzer,
002. from picamera import PiCamera
003. from datetime import datetime
004.
005. buzzer = TonalBuzzer(20)
006. doorbell = Button(21)
007. camera = PiCamera()
008.
009. def play(tune):
010.     for note, duration in tune:
011.         buzzer.play(note)
012.         sleep(float(duration))
013.     buzzer.stop()
014.
015. pink_panther = [
016.     ('C#4', 0.2), ('D4', 0.2),  (None, 0.2),
017.     ('Eb4', 0.2), ('E4', 0.2),  (None, 0.6),
018.     ('F#4', 0.2), ('G4', 0.2),  (None, 0.6),
019.     ('Eb4', 0.2), ('E4', 0.2),  (None, 0.2),
020.     ('F#4', 0.2), ('G4', 0.2),  (None, 0.2),
021.     ('C4', 0.2),  ('B4', 0.2),  (None, 0.2),
022.     ('F#4', 0.2), ('G4', 0.2),  (None, 0.2),
023.     ('B4', 0.2),  ('Bb4', 0.5), (None, 0.6),
024.     ('A4', 0.2),  ('G4', 0.2),  ('E4', 0.2),
025.     ('D4', 0.2),  ('E4', 0.2)
026. ]
027.
028. while True:
029.     doorbell.wait_for_press()
030.     dt = datetime.now().isoformat()
031.     camera.capture('/home/pi/{}.jpg'.format(dt))
032.     play(pink_panther)
```

11 C'est à votre tour

Jusqu'ici nous avons ajouté quelques fonctions rudimentaires – un jingle personnalisé et une caméra – mais vous pouvez en faire bien plus. Si vous avez des lampes intelligentes, faites clignoter l'éclairage de la maison pour donner une indication visuelle quand on sonne, même lorsque vous écoutez de la musique. Vous pouvez vous faire envoyer des courriels, des textos ou des tweets. Pourquoi pas une notification sur votre ordiphone ? Si vous vous sentez ambitieux, vous pourriez essayer d'identifier celui qui sonne à la porte et décider si vous le laissez entrer ! Le monde vous appartient. **(VF : Denis Lafourcade)**

Lanceur de bombe de table piloté par un Raspberry Pi

Vous devez allumer une bombe de table et vous voulez tirer le cordon avec du code ? Ce qu'il vous faut c'est un lanceur de bombe de table automatisé animé par Raspberry Pi.

MAKER

Mark Vanstone

Auteur de logiciels éducatifs dans les années 1990, dont la série ArcVenture, perdue corps et bien dans l'océan des logiciels, mais sauvée de l'oubli par le RPi !

magpi.cc/YiZnxL

@mindexplorers

L'idée de la bombe de table a d'abord été présentée lors d'une journée d'atelier sur les feux d'artifice de la Jam Raspberry de Milton Keynes (Royaume-Uni), puis développée pour la Journée des Filles au Musée National de l'Informatique de Bletchey Park (cf. Magpi anglais, n°77, magp.cc/77). La machine permet de déclencher à distance une bombe de table avec un peu de code Python sur un RPi. D'abord nous concevons un support pour notre bombe, par ex. dans une appli de modélisation 3D, puis nous le fabriquons avec une imprimante 3D. Ensuite nous procéderons à la mise à feu au moyen d'un servo bon marché.

01 Commençons par la fin

D'habitude, dans un tutoriel, on commence par les bases et on développe l'idée, mais ici nous devons voir ce vers quoi nous allons, puis revenir en arrière et rassembler tous les morceaux.

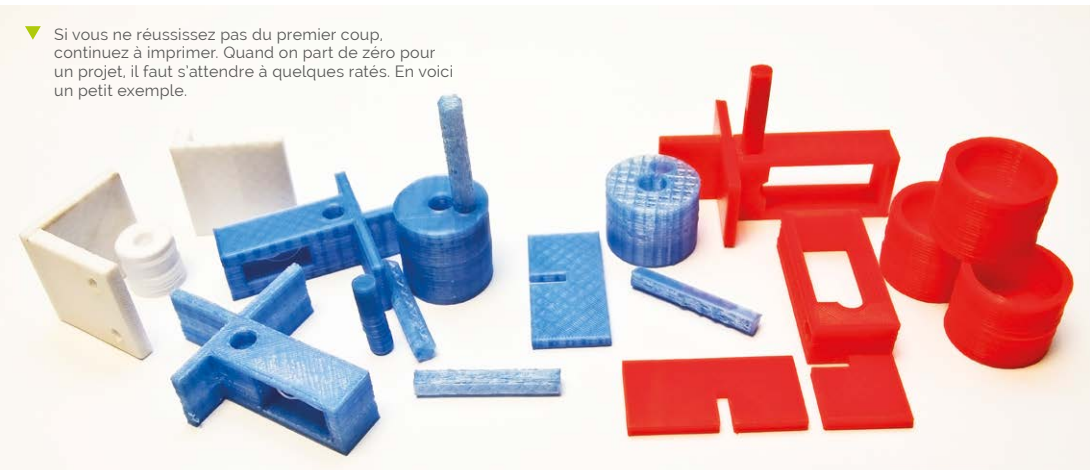
Telle a été notre approche pour la conception du lanceur : nous avons décidé de ce qui devait se passer puis nous avons travaillé sur chaque partie jusqu'à satisfaire les exigences. Nous nous sentons comme des singes qui écrivent « Guerre et Paix », mais nous passerons sous silence ce qui n'a pas marché. Découvrez sur l'image principale le lanceur entièrement assemblé, et passons en revue les éléments dont il est constitué.

02 Le dispositif de soutien

Le lanceur a été conçu avec Blender puis imprimé en 3D. Les fichiers source sont tous disponibles sur GitHub (**magpi.cc/YiZQxd**) si vous voulez jouer avec. D'abord comment soutenir une bombe de table ? Elle doit être maintenue droite avec un accès au cordon de mise à feu. Nous avons un morceau de tube, dont la dimension intérieure reproduit la forme de la bombe, avec un trou pour faire passer la « poignée ». La base du tube est

Ingrédients

- Bombe de table
- Appli de modélisation 3D **blender.org**
- Imprimante 3D
- Servo numérique puissant **magpi.cc/cVxSzg**
- 3× fil avec embouts mâle-femelle **www.elektor.fr/mini-breadboards-jumper-wires**



Si vous ne réussissez pas du premier coup, continuez à imprimer. Quand on part de zéro pour un projet, il faut s'attendre à quelques ratés. En voici un petit exemple.

Le lanceur de bombe de table imprimé avec du filament PLA. Le lanceur est imprimé en quatre parties : deux pour la base, une pour le support, et une tige pour les relier.

Le servo est relié directement aux broches GPIO du RPi. Une connexion à la masse, une à la broche de 5 V et le fil de signal à l'une des broches utilisables pour la MLI.

assez épaisse pour permettre d'y fixer une tige et avoir une fixation solide.

03 La base

La base pourrait être imprimée en un seul morceau, mais nous avons constaté que placer la partie principale sur la tranche permettait une meilleure insertion du servo. On peut alors imprimer une traverse à emboîter pour stabiliser la base et éviter qu'elle ne se renverse. Lors de l'impression de ces pièces, il faudra peut-être adapter l'épaisseur des cloisons pour qu'elles soient bien ajustées, sans être trop serrées. Il y a aussi un trou dans la base, comme dans le support, pour fixer une tige.

04 La tige

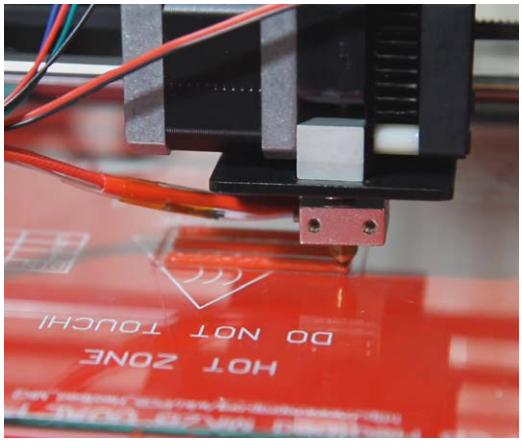
C'est elle qui réunit le tout. La tige est juste un cylindre qui relie le support à la base. Elle a un méplat sur toute sa longueur, mais c'est juste pour qu'elle adhère bien au plateau de l'imprimante 3D pendant l'impression. Avec cette conception, on peut imprimer les quatre parties en une seule fois puis les ranger dans une petite boîte, comme un porte-cartes de visite en plastique, avec les autres composants pour la construction, ce qui est parfait pour les ateliers. Cela donne aussi une dimension supplémentaire pour la fabrication par l'atelier si aucune instruction de montage n'est fournie.

05 Conception pour impression 3D

Les pièces sont imprimées en 3D, mais comment sont-elles conçues ? Si vous êtes novice en la matière, ce n'est pas difficile. Il faut une appli de modélisation 3D ; il y a plusieurs options gratuites et payantes. Notre préférée du moment est Blender, parce qu'elle est gratuite et que nous avons suivi son développement depuis des années ; la version 2.8 bêta vient de sortir, et elle déchire ! N'importe quelle appli qui exporte des fichiers *.obj* ou *.3ds* ou similaire convient, du moment que le format est compatible avec votre logiciel d'impression 3D.

06 Réalisation des formes

Si vous débutez dans la modélisation 3D,



Les pièces 3D peuvent être dessinées et redessinées dans une appli de modélisation 3D puis exportées vers un logiciel d'impression 3D. L'imprimante utilise alors un filament pour fabriquer les pièces par couches.

Astuce

Blender

La modélisation 3D est un peu limitée sur le RPi en raison de la prise en charge du graphique par le matériel. Vous pouvez installer Blender avec `sudo apt-get install blender`. Il y a aussi quelques autres choix, comme 3D Slash.

Astuce

Servos analogiques

Vous pouvez utiliser ici des servos analogiques, mais comme le GPIO est numérique, vous verrez que le calcul des angles est plus difficile, car la sortie donnera des résultats bizarres.

limitez-vous à des objets simples pour commencer. Cette construction est un bon exemple qui n'utilise que des formes élémentaires. Nous avons quelques rectangles, un long cylindre et un cylindre court et large. Pour obtenir des trous dans ces formes , on crée une forme pleine, puis une forme pour la découpe. Utilisez un outil booléen pour soustraire ou découper la seconde forme de la première. Une fois votre conception terminée, exportez le modèle dans un format adapté. Essayez-en plusieurs, car les résultats peuvent varier lors de l'importation dans votre logiciel d'impression 3D.

07 Impression 3D

Étant donnée la disposition des pièces, il n'y a pas de surplomb significatif, à part les petits trous pour la tige, ce qui vous évite d'avoir à détacher des supports ou autre matière ajoutée. Nous avons utilisé du filament PLA sur une imprimante très bon marché à assembler soi-même. Au passage, pour tout savoir sur le fonctionnement des imprimantes 3D, prenez une version en kit et si vous vous sentez courageux, une vraiment pas chère avec des instructions en langue étrangère. Lorsqu'elle finira par fonctionner, vous saurez tout sur chacun des composants !

08 Servos puissants

Au début, il semblait évident de sortir des tiroirs tous les gadgets susceptibles de tirer un cordon de bombe de table. Nous avons essayé un moteur, mais cela posait des problèmes parce qu'il fallait des pignons, un module de pilotage fiable et une fixation solide. Nous avons alors trouvé un de ces servos bleus de 9 g. On a essayé, vraiment, mais à la fin nous avons dû accepter que ça n'allait pas le faire. Nous avons trouvé un servo HS-422 abandonné dans un tiroir, retravaillé le socle et obtenu de meilleurs résultats. Depuis, nous avons trouvé le servo MG996R, très proche en taille et résultat pour cette application.

▼ Comme l'imprimante 3D accumule de la matière du bas vers le haut, il vaut mieux concevoir des pièces sans partie en surplomb à remplir, sinon l'impression risque de mal tourner.

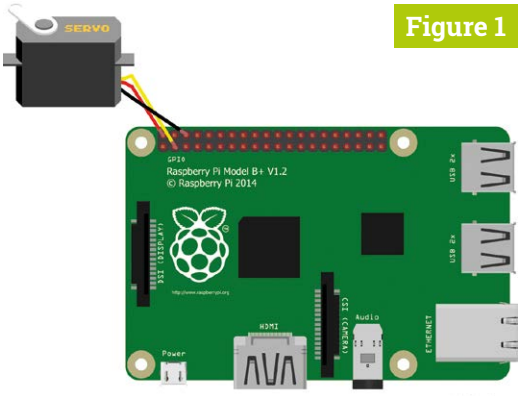


Figure 1

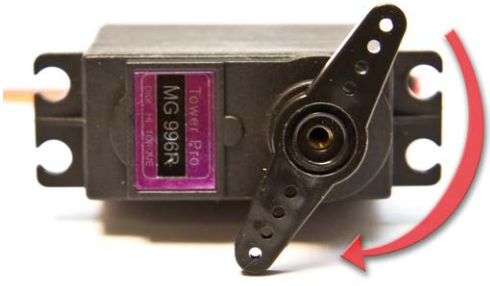
▲ **Figure 1** Câblage des fils de liaison : le rouge au 5 V, le jaune/orange à la broche GPIO 2 et le marron/noir à GND.

09 Branchement

On relie le servo directement aux broches GPIO du RPi avec trois fils à embouts mâle-femelle (câblage sur la **figure 1**). Attention : même si connecter le servo sur ces broches alors que le RPi fonctionne ne devrait pas causer de dommage, si vous vous trompez, le résultat pourrait être très différent, donc éteignez le RPi avant de câbler les broches GPIO, et vérifier le branchement avant de le rallumer. Une fois que tout est connecté, nous pouvons tester le servo et le faire bouger.

10 Tester 1, 2, 3

Nous écrivons un programme simple en Python pour faire tourner le servo. Voir le listage **party_popper.py**. Vous pouvez le saisir avec IDLE 3 ou, si vous préférez, un autre éditeur fourni avec Raspbian. Nous configurons la broche GPIO 2 (= broche physique 3) en sortie à MLI (Modulation



▲ La fonction `SetAngle()` déplace le palonnier du servo de sa position actuelle vers le nouvel angle.

par Largeur d'Impulsion) et nous la démarrons. Nous avons alors une fonction pour changer l'angle du bras du servo. Nous calculons la valeur MLI nécessaire (`pwm`) à partir de l'angle et changeons le rapport cyclique du servo, nous attendons qu'il bouge, puis nous remettons cette valeur à zéro. En changeant la valeur dans l'appel à `SetAngle()` au-dessous de cette fonction, nous verrons bouger le bras du servo quand nous exécutons le programme.

11 À propos des angles

La définition des angles nécessaires pour que le servo bouge et tire suffisamment le cordon de la bombe pour l'amorcer dépendra du réglage du palonnier du servo, et probablement de quelques autres facteurs. Si vous fixez le servo dans la base pour que le cordon puisse être attaché à l'extrémité du palonnier et que lorsqu'il tourne, il tire le cordon tout droit vers le bas (ou aussi droit que possible), alors il faut juste trouver l'angle avec lequel la longueur du cordon est la plus courte, puis l'angle pour lequel le palonnier est au plus bas.

12 Mise à feu !

Tout est correctement branché, le servo inséré dans la base du support est en position de départ, le cordon est fermement attaché au palonnier : vous êtes prêts pour la mise à feu de la bombe de table. Exécutez le code avec le paramètre de `SetAngle()` réglé sur l'angle final désiré, et en quelques secondes, le servo va passer à l'action, en tirant sur le cordon avec une force de plusieurs kilogrammes. Avant de vous en rendre compte, il devrait y avoir une forte explosion et une jolie pile de confettis à ramasser ! **(VF : Denis Lafourcade)**

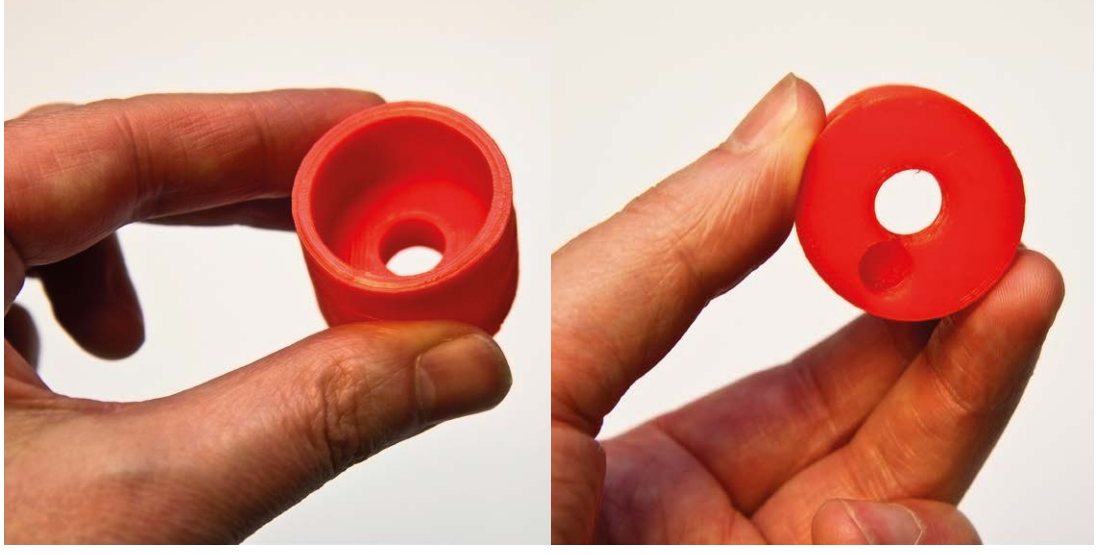
> « En quelques secondes, le servo va passer à l'action, en tirant sur le cordon avec une force de plusieurs kilogrammes »

party_popper.py

TÉLÉCHARGEZ LE CODE COMPLET : **magpi.cc/YiZQxd**

Langage : **Python**

```
001. import RPi.GPIO as GPIO
002. from time import sleep
003.
004. GPIO.setmode(GPIO.BOARD)
005. GPIO.setup(3, GPIO.OUT)
006. pwm=GPIO.PWM(3,50)
007. pwm.start(0)
008.
009. def setAngle(angle):
010.     duty=angle/18+2
011.     GPIO.output(3, True)
012.     pwm.ChangeDutyCycle(duty)
013.     sleep(1)
014.     GPIO.output(3, False)
015.     pwm.ChangeDutyCycle(0)
016.
017. setAngle(0)
018.
019. pwm.stop()
020. GPIO.cleanup()
```



Le support de la bombe de table est un cylindre avec le fond percé et, au-dessous, un trou pour la tige de fixation à la base.

Astuce

Présentation en ligne

Vous pouvez trouver la présentation de l'atelier initial sur le site de TechnoVisual Education : **magpi.cc/yKUSQg**

Surveillance de plantes par IA

Mesurez la hauteur et la largeur de toutes sortes d'objets en utilisant le module Camera et OpenCV.

MAKER

PJ Evans

PJ est écrivain, développeur et organisateur de Jam RPi. Il aime sentir les fleurs en Lego.

mrpjevans.com

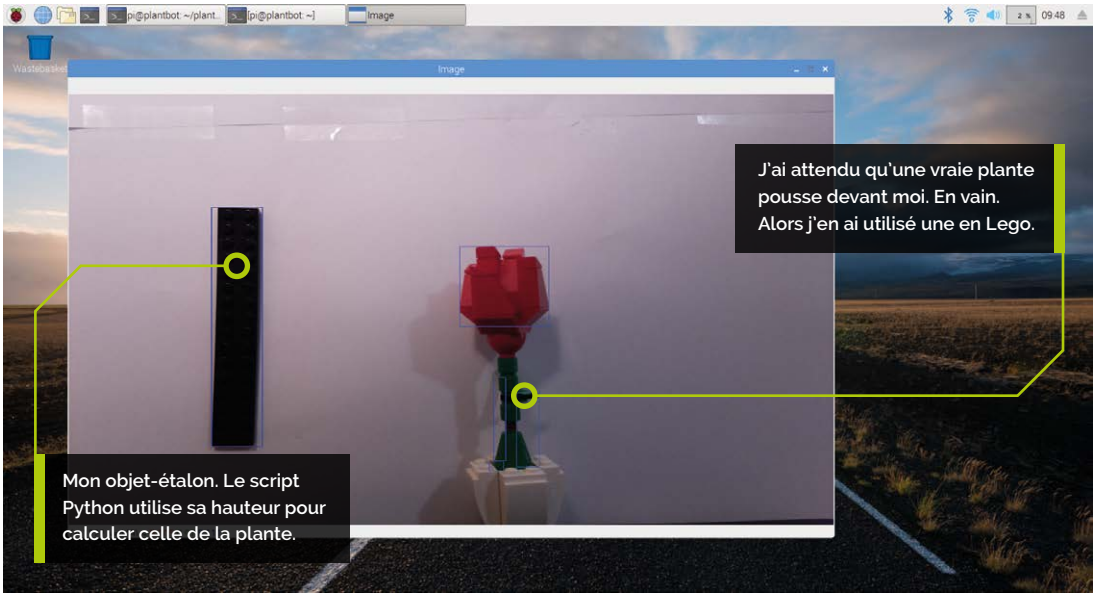
Vous êtes-vous déjà demandé à quelle vitesse poussaient vos plantes ? Ou même quelle était leur taille ? Une simple règle suffirait bien sûr, mais passer comme ici une plante à la toise de l'IA nous fera découvrir OpenCV (version 2), un outil puissant d'analyse et de reconnaissance d'objet. En comparant un objet à un autre de dimensions connues, OpenCV peut en effet évaluer sa taille. Nous n'aurons besoin que du module Camera, mais l'ajout de capteurs de température ou d'humidité vous permettra de passer d'une surveillance passive à un contrôle actif, pour ne pas dire intelligent, de la croissance d'une plante.

01 Planter le décor

La précision des mesures dépendra de plusieurs facteurs. D'abord, l'arrière-plan devra être uni et offrir un bon contraste ; un fond blanc ou noir est idéal. Ensuite, l'emplacement devra être bien éclairé au moment de la prise de vue. Enfin, l'objectif devra être aligné verticalement avec la plante et lui faire face directement. Tout écart d'angle ou d'alignement dégradera la qualité des mesures. Un téléobjectif peut s'avérer nécessaire si la caméra est placée très loin de la plante.

02 Étalon de mesure

Puisque mesurer revient à comparer, nous aurons besoin d'un objet de largeur et hauteur connues. L'image prise devra inclure l'étalon et la plante. Choisissez donc un objet dont la taille est comparable à celle de votre plante. J'ai utilisé une longue brique de Lego, mais une règle convient. Placez votre étalon face à l'objectif de façon ce qu'il apparaisse à gauche sur l'image et soit à la même distance de la caméra que l'est la plante.

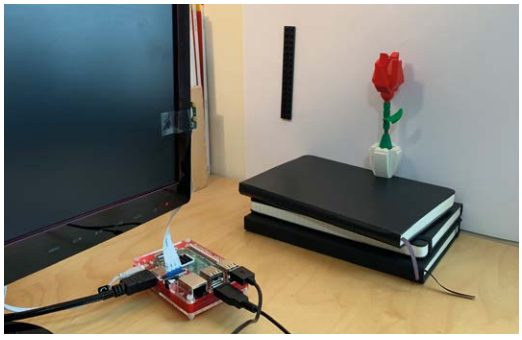


J'ai attendu qu'une vraie plante pousse devant moi. En vain. Alors j'en ai utilisé une en Lego.

Mon objet-étalon. Le script Python utilise sa hauteur pour calculer celle de la plante.

Ingrédients

- Module Camera Pi **www.elektor.fr/pi-camera-module-2**
- Plante d'intérieur
- Règle ou objet rectangulaire
- Fond uni contrasté



Mon banc d'essai. J'ai aligné verticalement le module Camera et la plante du mieux que j'ai pu.

03 Préparation du RPi

Les images prises par le module Camera seront analysées par la bibliothèque OpenCV et ses liaisons Python (*bindings*). N'importe quel RPi actuel conviendra, mais le traitement des images sera plus rapide avec un modèle 3B/3B+. OpenCV nécessitant un système de fenêtrage X Window, une version Raspbian Stretch avec environnement de bureau s'impose. Assurez-vous aussi que votre RPi est à jour en lançant :

```
sudo apt update && sudo apt -y upgrade
```

Installez votre caméra si ce n'est déjà fait, activez-la depuis **Préférences > Configuration du Raspberry Pi > Interfaces**, et redémarrez.

04 Installation des dépendances

L'installation de la bibliothèque OpenCV est moins directe que celle de la plupart des programmes, car elle dépend de nombreux paquets. Nous aurons en outre besoin d'outils supplémentaires pour qu'OpenCV puisse dialoguer avec le module Camera. Ouvrez donc un terminal et entrez :

```
sudo apt install python3-pip libatlas-base-dev libhdf5-100 libjasper1 libqtcore4 libqt4-test libqtgui4

pip3 install imutils opencv-contrib-python picamera[array]
```

Même si piwheels accélère considérablement l'installation, notez que l'exécution de la dernière commande prend un certain temps.

plantbot.py

TÉLÉCHARGEZ LE CODE COMPLET : magpi.cc/omacNA

Langage : **Python 3**

```
001. import argparse
002. import imutils.contours
003. import cv2
004. from picamera.array import PiRGBArray
005. from picamera import PiCamera
006. from time import sleep
007.
008. # Get our options
009. parser = argparse.ArgumentParser(description='Object height measurement')
010. parser.add_argument("-w", "--width", type=float, required=True,
011.                     help="width of the left-most object in the image")
012. args = vars(parser.parse_args())
013.
014. # Take a photo
015. camera = PiCamera()
016. rawCapture = PiRGBArray(camera)
017. sleep(0.1)
018. camera.capture(rawCapture, format="bgr")
019. image = rawCapture.array
020.
021. # Cover to grayscale and blur
022. greyscale = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
023. greyscale = cv2.GaussianBlur(greyscale, (7, 7), 0)
024.
025. # Detect edges and close gaps
026. canny_output = cv2.Canny(greyscale, 50, 100)
027. canny_output = cv2.dilate(canny_output, None, iterations=1)
028. canny_output = cv2.erode(canny_output, None, iterations=1)
029.
030. # Get the contours of the shapes, sort l-to-r and create boxes
031. _, contours, _ = cv2.findContours(canny_output, cv2.RETR_EXTERNAL,
032.                                   cv2.CHAIN_APPROX_SIMPLE)
033. if len(contours) < 2:
034.     print("Couldn't detect two or more objects")
035.     exit(0)
036.
037. (contours, _) = imutils.contours.sort_contours(contours)
038. contours_poly = [None]*len(contours)
```

Astuce

Restez sobre

Évitez tout papier peint psychédélique ou à motifs, OpenCV échouera sans un fond uni et bien éclairé.

05 Test de calibrage

Vérifiez votre installation avant d'exécuter le code. Dans le terminal, entrez :

```
raspistill -t 0
```

Cette commande met la caméra en mode « preview » (qui nécessite un paramétrage particulier dans le cas d'une connexion par VNC, donc utilisez une connexion directe). Étudiez attentivement l'image. La caméra est-elle bien orientée ? N'est-elle pas trop basse ou trop haute ? Le contraste est-il suffisant ? Faut-il plus de lumière ? Appuyez sur **Crtl-C** lorsque vous aurez terminé votre vérification.

06 Script plantbot.py

Saisissez le code du listage **plantbot.py** dans un éditeur de texte et enregistrez-le (ou téléchargez-le depuis **magpi.cc/akmZsW**). Le script demande au module Camera de prendre une photo sous forme de flux et l'envoie à OpenCV. L'image est soumise à divers filtres de nettoyage, puis les formes présentes y sont détectées. Nous savons que la forme la plus proche de l'origine (le point (0,0) situé tout en haut et à gauche) est l'objet-étalon. La connaissance de sa largeur permet d'évaluer la taille des autres objets.

07 Exécution du script

Mesurez donc soigneusement la largeur de votre étalon (16 mm pour le mien), puis lancez le code avec la commande suivante, en remplaçant 16 par la largeur que vous aurez mesurée :

```
python3 plantbot.py -w 16
```

L'image de sortie s'affiche au bout de quelques secondes. Le contour de votre objet-étalon devrait être délimité par un rectangle à bord bleu, tandis qu'une ou plusieurs boîtes englobantes devraient représenter la silhouette de votre plante. Le contour d'une plante étant par nature irrégulier, OpenCV aura probablement détecté plusieurs de ces « boîtes d'encombrement ». Le code se sert de la plus haute et de la plus basse pour calculer la hauteur.

08 Enregistrement des données

Plutôt que de lire le résultat dans le terminal, nous pouvons enregistrer les mesures dans un fichier. Ajoutez d'abord la ligne d'importation suivante au début du code :

```
import datetime
```

Supprimez ensuite les six dernières lignes du code (la première commence par *print*) et remplacez-les par :

```
print("\"" + str(datetime.datetime.now()) +
"\"," + str(plantHeight))
```

Cette version du code vous permettra de créer un fichier CSV à partir duquel vous pourrez représenter graphiquement la croissance de votre plante.

09 Automatisation

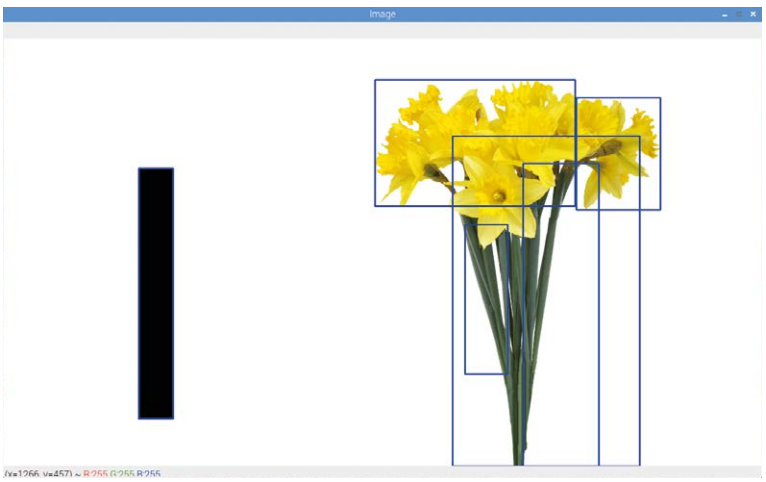
Que diriez-vous de surveiller quotidiennement la croissance de votre plante ? Rien de plus facile avec l'utilitaire cron du RPi. Une tâche « cron » se planifie dans un fichier dit crontab. Pour ouvrir celui de l'utilisateur par défaut, lancez :

```
sudo nano /etc/crontab
```

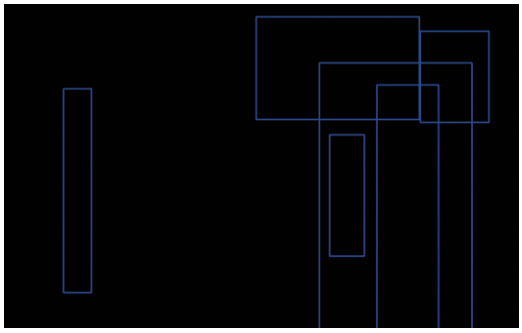
Ajoutez à la fin du fichier :

```
0 14    * * *   pi      python3 /home/
pi/plantbot/plantbot.py -w 16 >> /home/pi/
plantbot.csv
```

Cette ligne demande à cron d'exécuter le script tous les jours à 14 h (codé par « 0 14 ») et d'ajouter sa sortie à **plantbot.csv**.



Superposition de l'image de test et de l'image découpée par OpenCV. La précision pourrait être meilleure, mais la méthode rapide utilisée ici suffit.



Un objet-étalon (à gauche) et un bouquet de jonquilles (à droite) tels que vus par OpenCV.

« Vous savez maintenant mesurer toutes sortes de choses avec OpenCV, mais pourquoi s'arrêter là ? »

10 Des données comme engrais

Vous connaissez la hauteur de votre plante, pourquoi ne pas l'associer à d'autres données, p. ex. l'humidité de sa terre ? Ce type de capteur est bon marché et répandu (**magpi.cc/seyhTc**). Essayez de modifier le script de façon à ce qu'il lise votre capteur et ajoute ses mesures à la sortie CSV. Vous pourriez même envoyer une alerte en cas de manque d'eau, ou importer les données d'une API publique telle que **openweathermap.org/api** pour étudier l'influence de la météo sur votre plante !

11 Ne prenez pas racine

Vous pouvez maintenant mesurer toutes sortes de choses avec OpenCV, y compris la croissance de bambins, mais pourquoi s'arrêter là ? Je ne vous ai offert qu'un avant-goût de cette bibliothèque et n'ai exploité qu'une fraction de sa puissance. Elle fournit des fonctions de vision artificielle et d'apprentissage automatique qui permettent d'entraîner un RPi à reconnaître des objets ou encore des visages. Les roboticiens l'utilisent souvent, elle joue d'ailleurs un grand rôle dans les défis Pi Wars. Pour finir, je vous recommande le site **pyimagesearch.com**, qui a inspiré ce tutoriel.

(VF : Hervé Moreau)

Astuce

Pas de plante chez vous ?

Si vous n'avez pas de plante ou pas de conditions de mesure adaptées, testez le projet avec les images de **magpi.cc/omacNA**.

plantbot.py (suite)

Langage : **Python 3**

```
boundRect = [None]*len(contours)
for i, c in enumerate(contours):
    contours_poly[i] = cv2.approxPolyDP(c, 3, True)
    boundRect[i] = cv2.boundingRect(contours_poly[i])

output_image = image.copy()
mmPerPixel = args["width"] / boundRect[0][2]
highestRect = 1000
lowestRect = 0

for i in range(1, len(contours)):

    # Too smol?
    if boundRect[i][2] < 50 or boundRect[i][3] < 50:
        continue

    # The first rectangle is our control, so set the ratio
    if highestRect > boundRect[i][1]:
        highestRect = boundRect[i][1]
    if lowestRect < (boundRect[i][1] + boundRect[i][3]):
        lowestRect = (boundRect[i][1] + boundRect[i][3])

    # Create a boundary box
    cv2.rectangle(output_image, (int(boundRect[i][0]), int(boundRect[i][1])),
                  (int(boundRect[i][0] + boundRect[i][2]),
                  int(boundRect[i][1] + boundRect[i][3])), (255, 0, 0), 2)

# Calculate the size of our plant
plantHeight = (lowestRect - highestRect) * mmPerPixel
print("Plant height is {0:.0f}mm".format(plantHeight))

# Resize and display the image (press key to exit)
resized_image = cv2.resize(output_image, (1280, 720))
cv2.imshow("Image", resized_image)
cv2.waitKey(0)
```

Programmer le RPi en Java 11

Vous n'êtes pas partisan de l'utilisation de Python pour programmer votre Raspberry Pi ? Pas de souci : vous pouvez aussi utiliser Java.

AUTEUR

Frank Delporte

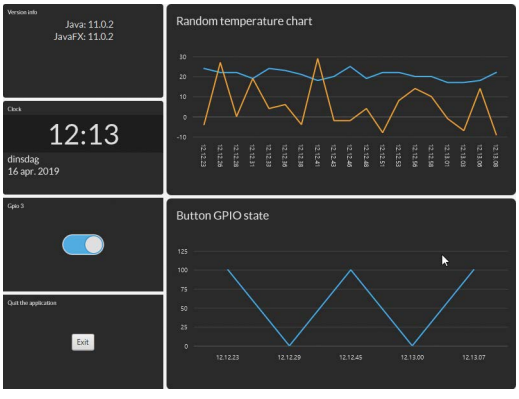
Développeur Java de profession, fan de technologie, partageur de connaissances en tant que blogueur, enfin coach du CoderDojo d'Ypres en Belgique. Il programme depuis ses 11 ans.

www.webtechie.be

@FrankDelporte

On peut bien entendu commander des LED avec un programme en Python sur le Raspberry Pi sous Raspbian, mais Python est un peu limité en ce qui concerne les interfaces graphiques. Sur le blog de Franck, il y a un jeu de Ping-Pong écrit en Python, mais nous voulions tout de même voir ce qu'on pouvait faire avec Java.

La **figure 1** montre le résultat du projet de cet article.



▲ **Figure 1.** Capture d'écran de l'application en fonctionnement.

Ingrédients

- Raspberry Pi 3B+
- Carte micro-SD de 8 Go avec Raspbian
- Écran, clavier et souris
- LED, résistance, bouton-poussoir, fils
- Code et scripts de **github.com/FDelporte/PiJava**

01 Préparation

Veillez à avoir tout ce qu'il faut pour commencer : un RPi 3B+ avec Raspbian, un clavier et un écran. Une liaison internet sera aussi nécessaire pour télécharger diverses bibliothèques.

Une certaine connaissance de Java est bien entendu utile.

02 Installation de Java

Il existe de nombreux environnements d'exécution Java, nous avons choisi Liberica JDK de BellSoft. Ils proposent une version spécifique pour le RPi, qui inclut JavaFX (nous reviendrons sur ce sujet).

Il n'y a qu'un fichier à télécharger, à décompacter et à ranger à un endroit où il sera accessible pour exécuter notre code et notre application :

```
$ cd /home/pi
$ wget https://download.bell-sw.com/
java/11.0.2/bellsoft-jdk11.0.2-linux-
arm32-vfp-hflt.tar.gz
$ sudo mv bellsoft-jdk11.0.2-linux-arm32-
vfp-hflt.tar.gz /opt
$ cd /opt
$ sudo tar -xvzf bellsoft-jdk11.0.2-
linux-arm32-vfp-hflt.tar.gz
$ sudo rm bellsoft-jdk11.0.2-linux-arm32-
vfp-hflt.tar.gz
```

Astuce

Si le lien vers le fichier tar.gz de Liberica JDK ne fonctionne plus, on peut le retrouver sur **bell-sw.com/pages/java-11.0.2**.

03 Test de l'installation Java

Le plus simple pour vérifier si on a la bonne version de Java est d'exécuter ce qui suit :

```
$ /opt/jdk-11.0.2/bin/java -version
openjdk version "11.0.2-BellSoft 2019-
01-15
OpenJDK Runtime Environment (build
11.0.2-BellSoft+7)
OpenJDK Server VM (build
11.0.2-BellSoft+7, mixed mode)
```

On lance Java avec le paramètre `-version` et on reçoit en retour les informations sur la version qui tourne sur notre RPi.

Lors de nos essais initiaux, nous avons utilisé un RPi B+ v1.2, mais nous avons obtenu le message « Illegal argument ». La cause en est le processeur du RPi : Liberica JDK a été développé pour un ARMv7/8, alors que les RPi moins récents utilisent un ARMv6, sur lequel Java 11 ne peut pas tourner. Un problème identique se pose avec le RPi Zero. Nous avons dès lors opté pour un RPi 3B+ pour nos expérimentations.

D'un autre côté, cela renforce nos convictions sur la puissance du RPi : il suffit de passer la carte micro-SD d'un modèle à l'autre pour vérifier si ça marche ou pas.

04 Notre premier code Java

Depuis la version 11, il est possible d'exécuter du code Java sans devoir d'abord le compiler, comme on peut aussi le faire avec Python ou un script Bash.

Nous l'avons bien entendu testé avec une application « Hello World » :

```
$ cd /home/pi
$ nano HelloWorld.java
```

Nous plaçons le code suivant dans ce nouveau fichier, ou nous copions **HelloWorld.java** de notre répertoire GitHub :

```
public class HelloWorld {
    public static void main (String[] args)
{
        System.out.println("Hello World");
    }
}
```

Pour exécuter le code, on lance Java depuis l'emplacement où il a été téléchargé, avec comme argument le fichier qui vient d'être créé :

```
$ /opt/jdk-11.0.2/bin/java /home/pi/
HelloWorld.java
Hello World
```

Voilà notre premier code Java fonctionnel sur un RPi !

05 Connexion du matériel

Un peu de matériel supplémentaire est nécessaire pour profiter de la pleine puissance du RPi (**figure 2**), on connecte donc une LED et un bouton-poussoir comme suit :

- GPIO22 (broche physique du RPi n°15, WiringPi n°3) > résistance > LED > GND (masse)
- 3,3 V > bouton > GPIO24 (broche physique du RPi n°18, WiringPi n°5)

Pour vérifier que la LED est bien dans le bon sens, il suffit de la relier provisoirement avec un fil entre le 3,3 V et la broche GPIO22. Si elle ne s'allume pas, il faut la retourner.

HelloGpio.java

TÉLÉCHARGEZ LE CODE COMPLET : github.com/FDelporte/PiJava

Langage : **Java**

```
001. public class HelloGpio {
002.     public static void main (String[] args) {
003.         System.out.println("Hello Gpio");
004.
005.         try {
006.             Runtime.getRuntime().exec("gpio mode 3 out");
007.
008.             var loopCounter = 0;
009.             var on = true;
010.
011.             while (loopCounter < 10) {
012.                 System.out.println("Changing LED to " +
013. (on ? "on" : "off"));
014.                 Runtime.getRuntime().exec("gpio write 3 "
015. + (on ? "1" : "0"));
016.
017.                 on = !on;
018.
019.                 Thread.sleep(500);
020.
021.                 loopCounter++;
022.             }
023.         } catch (IOException ex) {
024.             System.err.println("IOException from Runtime:
025. " + ex.getMessage());
026.         } catch (InterruptedException ex) {
027.             System.err.println("InterruptedException from
028. Thread: " + ex.getMessage());
029.         }
030.     }
031. }
```

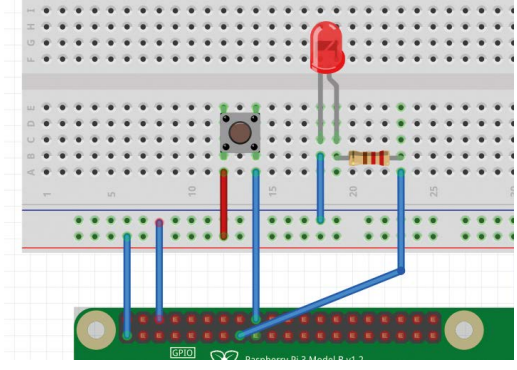


Figure 2. Les connexions sur la platine d'expérimentation.

06 Test du matériel

Les connexions du matériel peuvent être testées avec les commandes gpio depuis un terminal.

Allumage (1) et extinction (0) de la LED sur GPIO22 :

```
$ gpio mode 3 out
$ gpio write 3 1
$ gpio write 3 0
```

Vérification de l'état du bouton-poussoir (1 = pression, 0 = repos) :

```
$ gpio mode 5 in
$ gpio read 5
1
```

07 Et maintenant avec Java

On peut effectuer les mêmes opérations en Java en recopiant le code de **HelloGpio.java** (ou en récupérant le fichier d'exemple de GitHub).

Dans cet exemple, nous faisons varier l'état de la broche 3 dix fois toutes les 500 ms.
Java nous oblige à traiter les exceptions possibles, d'où le code *try/catch*.

On peut aussi se passer de la compilation, avec le même résultat :

```
 $ /opt/jdk-11.0.2/bin/java /home/pi/
HelloGpio.java
 Hello Gpio
 Changing LED to on
 Changing LED to off
 Changing LED to on
```

08 En avant pour une application !

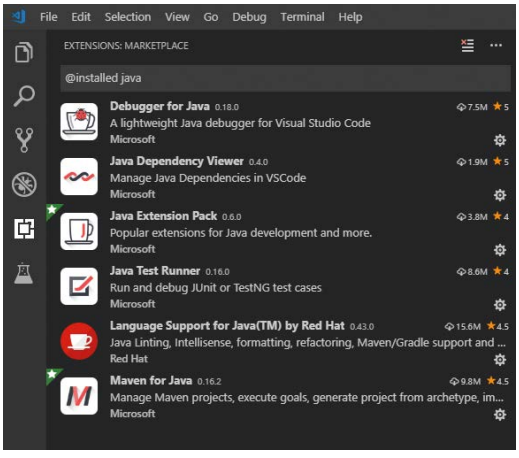
Exécuter du code Java est amusant et permet de tester l'une ou l'autre chose rapidement, mais ce n'est qu'un premier pas. Une application Java demande un peu plus d'efforts.

Notre application, que l'on peut télécharger dans notre dépôt GitHub, contient tout ce qu'il faut pour créer un fichier compilé .jar avec Maven.

Il faut veiller à ce que le kit de développement Java 11 correct (JDK, *Java Development Kit*) soit aussi installé sur votre ordinateur. Nous utilisons le *jdk-11.0.3+7* pour Windows x64 de **adoptopenjdk.net**, qui est gratuit ; ce site propose également des versions pour d'autres plateformes.

Après avoir téléchargé le code, ouvrez-le dans un environnement de développement intégré (EDI), par ex. IntelliJ ; Eclipse reconnaîtra aussi le fichier **pom.xml** (*Project Object Model*).

Vous pouvez aussi utiliser la version gratuite de Visual Studio Code ; le programme suggérera



▲ **Figure 3 :** extensions Java-dans Visual Studio Code.

les extensions à installer pour la gestion optimale du code Java (voir sur la **figure 3** les extensions que nous utilisons). Bien que ces outils aient été créés pour le développement d'applications sur la toile, ils sont désormais capables de bien d'autres choses, dont la programmation en Java (voir **code.visualstudio.com/docs/languages/java** pour plus d'informations). Le cheminement vers l'endroit où vous avez copié votre JDK devra être indiqué dans **.vscode\settings.json**.

Prenez une vue d'ensemble du code de l'exemple, nous allons nous y plonger progressivement.

Qu'est-ce que Maven ?

Maven aide à configurer l'EDI Java et à créer l'application. Les fonctions utilisées par l'application (bibliothèques supplémentaires pour des fonctions spécifiques) doivent être indiquées dans le fichier **pom.xml**.

09 Le boss, c'est pom.xml

L'application utilise des bibliothèques pour

créer un tableau de bord. On trouve donc JavaFX et TilesFX dans **pom.xml**. L'EDI téléchargera les versions indiquées à partir de la bibliothèque de Maven lors du développement, de telle manière que vous receviez des infobulles lors de l'ajout de code et que vous puissiez mener à bien le travail.

Qu'est-ce que JavaFX ?

JavaFX est une couche « au-dessus » de Java pour le développement d'applications graphiques. On trouve dans les bibliothèques des boutons, des canevas (*layout*), des chargeurs d'images, etc. Il y a également des dizaines de projets à code source ouvert pour bâtir des structures encore plus complexes.
Exemples : **github.com/HanSolo/tilesfx** (la bibliothèque utilisée dans cet article) et **www.pixelduke.com/fxribbon** (une bibliothèque qui permet l'ajout de barres de tâches semblables à celles d'une application Office).

10 Aide pour le GPIO

Avec **HelloGpio.java**, nous pouvions allumer et éteindre une LED avec une seule ligne de code, mais si nous voulons pouvoir intervenir en cas d'erreur ou connaître l'état des broches du port GPIO, alors il faut utiliser **Gpio.java**.

Trois fonctions sont disponibles :

```
initiatePin(final int pin, String mode)
```

Il faut appeler cette fonction pour chaque broche que nous voulons assigner en entrée ou sortie.

```
setPinState(final int pin, boolean on)
```

Cette fonction sert à allumer (`true`) ou éteindre (`false`) une LED.

```
getPinState(final int pin)
```

Cette fonction permet de connaître l'état d'une broche, haut (`1 = true`) ou bas (`0 = false`).

11 Démarrage des fonctions

La fonction qui est appelée au démarrage est enregistré dans **pom.xml : Launcher.java** pour lancer correctement JavaFX, qui appelle à son tour **Main.java**. Le code peut être testé facilement dans l'EDI avec ces deux fonctions.

12 Tableau de bord avec TilesFX

TilesFX permet la création rapide d'un tableau de bord dans FxScreen.java.

Prenons l'exemple d'une case avec un interrupteur à glissière pour allumer ou éteindre la LED :

```
var ledSwitchTile = TileBuilder.create()
                .skinType(SkinType.SWITCH)
                .prefSize(250, 200)
                .title("Gpio " + PIN_LED)
                .roundedCorners(false)
                .build();

ledSwitchTile.setOnSwitchReleased(e ->
    Gpio.setPinState(
        PIN_LED,
        ledSwitchTile.isActive()
    )
);
```

13 Création de l'application

Avec Maven et **pom.xml**, on peut créer une application avec deux fonctions, de telle manière qu'elle puisse être exécutée sur notre ordinateur ou sur le RPi.

Avec `compiler:compile`, les fichiers Java sont transformés en fichiers `.class` (*bytecode*). Le fichier `.jar`, qui contient le programme entier (dont la taille n'est que de 9 Ko…), est ensuite produit par `jar:jar`.

Avec Visual Studio Code, c'est encore plus simple : si on maintient le curseur de la souris sur la fonction `main`, l'option **Run|Debug** apparaît ; elle permet de lancer directement l'application (**figure 4**).

```
14  public class Launcher {
        Run | Debug
15      public static void main(String[] args) {
16          Main.main(args);
17      }
18  }
```

▲ **Figure 4.** Lancement de l'application avec Visual Studio Code.

Truc
Un fichier **.jar** peut être ouvert comme un fichier **.zip**. On y retrouve la structure du code, mais avec des fichiers **.class**, qui peuvent être exécutés par Java.

14 Module additionnel pour le RPi

Le module TilesFX manque encore sur notre RPi, pour ajouter un tableau de bord à l'application. Il faut le télécharger et l'enregistrer dans le répertoire */opt*, à côté du JDK téléchargé auparavant.

L'opération est plus rapide, puisqu'il n'y a pas de fichiers à décompacter :

```
$ cd /home/pi
$ wget -O tilesfx-11.1.jar http://
central.maven.org/maven2/eu/hansolo/
tilesfx/11.1/tilesfx-11.1.jar
$ sudo mkdir /opt/tilesfx11
$ sudo mv tilesfx-11.1.jar /opt/tilesfx11
```

15 Lancement de l'appli sur le RPi

Dernière étape : enregistrement du fichier **pijava-0.1-SNAPSHOT.jar** sur le RPi ; celui-ci a été téléchargé de GitHub ou créé sur l'ordinateur. Nous préférons le téléchargement de GitHub, mais on peut le faire via SSH en se connectant au RPi.

On peut ensuite lancer l'application, avec quelques options (la fonction est aussi dans les scripts de GitHub) :

```
$ sudo /opt/jdk-11.0.2/bin/java \
    --module-path=/opt/tilesfx11 \
    --add-modules=eu.hansolo.tilesfx \
    -jar pijava-0.1-SNAPSHOT.jar \
    -Dprism.verbose=true \
    -Dembedded=monocle \
    -Dglass.platform=Monocle
```

Diverses informations de connexion apparaissent d'abord à l'écran, puis l'écran FX est ouvert.

La température est enregistrée toutes les 2,5 s, au hasard à l'extérieur ou à l'intérieur, et représentée sous forme de graphique.

Un interrupteur à glissière, situé dans une des cases de TilesFX, permet d'allumer ou éteindre la LED.

Toutes les ½ s l'état du bouton est lu et ajouté au graphique du bas s'il a été modifié.

16 Conclusion

Quand vous savez quelle version de Java utiliser sur le RPi – merci à GluonHQ et BellSoft pour JavaFX – vous pouvez vous lancer en deux temps trois mouvements avec des fichiers de test simples.

La création d'une application demande un peu plus de travail pour tout mettre en place dans l'EDI et faire des tests, mais le développement peut se faire sur un ordinateur avant de porter l'application sur le RPi.

À vous d'imaginer la suite : pourquoi pas le tableau de commande d'une centrale de domotique avec les divers composants de TilesFX ? Tenez-nous au courant ! **(VF : Jean-Louis Mehren)**

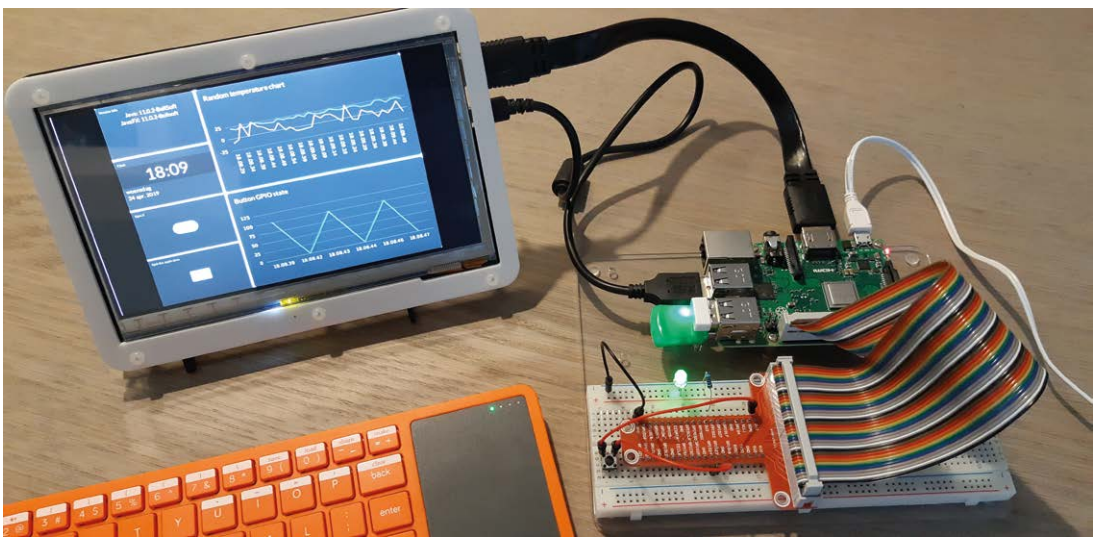
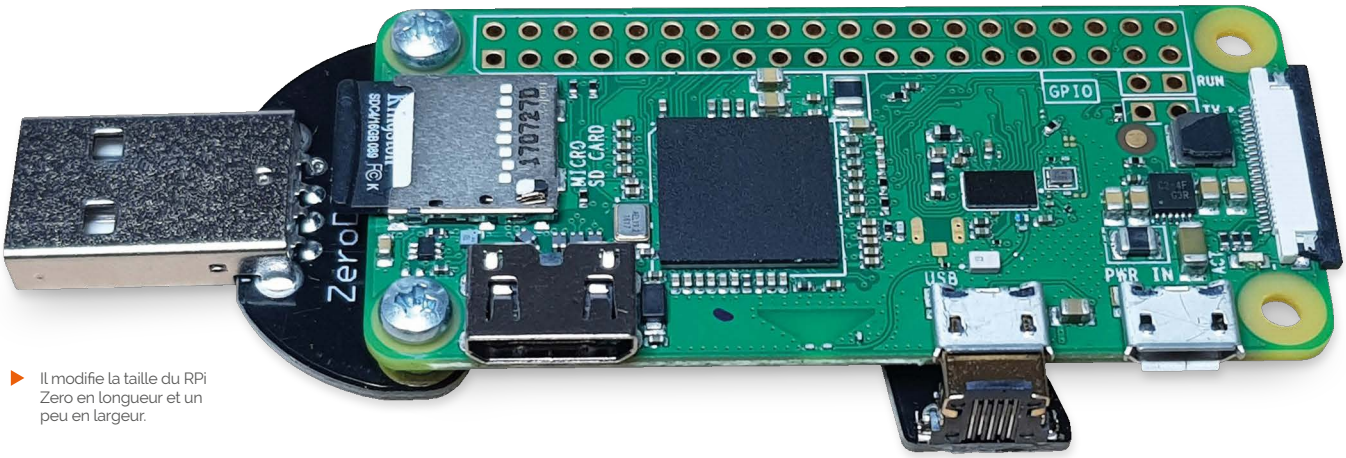


Figure 5. Le banc de test avec l'appli en marche.



Il modifie la taille du RPi Zero en longueur et un peu en largeur.

Zero Dongle

8086.net | **magpi.cc/KHZHya** | 5,60 €

Transformez votre Pi Zero en clé USB, en toute simplicité, grâce à ce dispositif bon marché. **Rob Zwetsloot** a désormais la clé des champs.

SPÉCIFICATIONS

PORTS USB : 1 × USB A, 1 × USB micro

MONTAGE : 2 vis

ENCOMBREMENT : 70×35 mm

On oublie trop souvent qu'une fois connecté au port USB d'un ordinateur, le RPi Zero peut fonctionner comme une clé USB prête à l'emploi. Il peut s'agir de l'alimenter, de s'en servir de mémoire auxiliaire USB, ou même de partager la connexion internet d'un ordinateur.

La principale difficulté consiste à doter le RPi Zero d'une bonne connexion USB. En effet, les câbles mâle-mâle de micro-USB à USB A ne courent pas les rues et ceux que l'on trouve ne sont pas toujours très soignés. Zero Dongle lui est différent. Cette solide carte permet de relier un connecteur USB A mâle à un connecteur micro-USB mâle.

Zero Dongle se tient droit

Zero Dongle se visse sur n'importe quel RPi Zero, du côté du lecteur de carte SD. Comme il est plutôt robuste, Zero Dongle ne risque pas de se plier. Il modifie inévitablement la taille du RPi Zero : la petite saillie latérale permet le raccord au port de données micro-USB du RPi Zero tandis que le connecteur USB A assure la liaison avec le PC.

Le fin circuit imprimé se distingue par sa robustesse.

Comme il n'a pas de puce spécifique, le dongle ne configurera pas automatiquement votre RPi en clé USB. Vous devrez procéder aux modifications habituelles sur votre carte SD. Vous aurez probablement des difficultés à trouver un écrin qui convient.

Mais à ce prix-là et vu le résultat obtenu, on ne peut pas tout avoir.

(VF : Pascal Duchesnes)

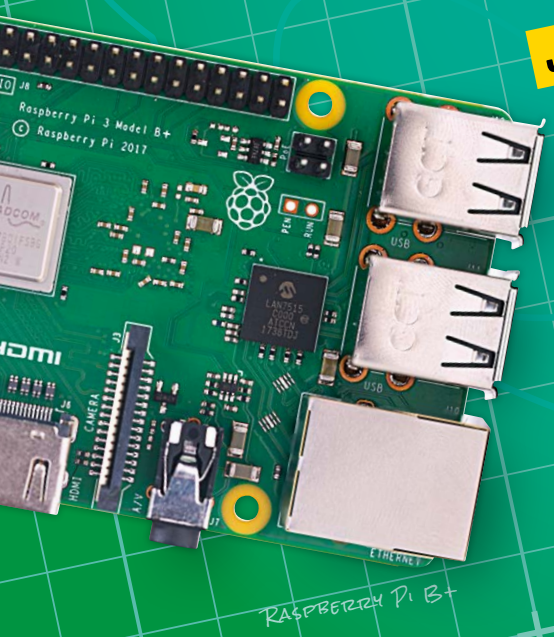
Notre avis

Excellent dispositif bon marché qui permet de transformer le RPi Zero en clé USB.

9/10

DÉPASSER LES LIMITES DU LOGICIEL : RÉALISER UN PROJET MATÉRIEL AVEC UN RASPBERRY PI

Vous avez surfé sur le web et bricolé avec Raspbian. Le moment est venu pour **Jenny List** de vous montrer comment réaliser quelque chose de concret avec votre RPi !



58 MM

Félicitations, vous avez acheté un Raspberry Pi. Vous l'avez branché et essayé, mais vous avez atteint la limite des choses intéressantes à faire assis derrière un bureau. Vous voilà prêt à ajouter du matériel pour que votre RPi fasse quelque chose d'intéressant !

Le RPi est un bon point de départ pour vos projets matériels, grâce à une parfaite association d'interfaces, épaulée par la meilleure documentation et communauté d'aide du domaine. La plupart des ordinateurs ne trônent pas sur un bureau ; au lieu de cela, ils effectuent des mesures et commandent des machines. Ainsi, en intégrant votre RPi dans vos réalisations, vous faites le premier pas dans le monde passionnant de l'informatique embarquée.

Vous avez déjà dû voir quelques projets matériels avec un RPi vraiment sympas, mais à y regarder de plus près ils semblent incroyablement complexes. C'est plutôt intimidant, mais n'ayez crainte : c'est plus simple qu'il n'y paraît.

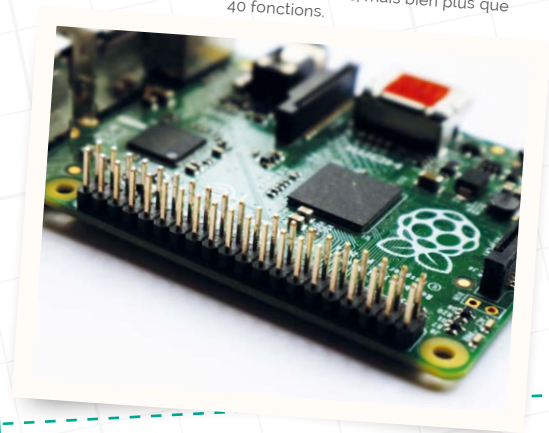
On peut diviser les fonctions de tout projet informatique en trois parties. L'ordinateur acquiert des signaux d'entrée (mesures), utilise du logiciel pour réfléchir à ce qu'il va faire avec ces mesures, puis fait quelque chose avec le résultat. Si vous êtes tenté par un projet, découpez-le ainsi en trois, il sera beaucoup plus facile à comprendre.

Comment un ordinateur mesure-t-il quelque chose simplement en examinant ses entrées ? La réponse est qu'il interroge un capteur connecté à l'une de ses entrées et reçoit la valeur lue. Ce capteur peut être aussi simple qu'un bouton-poussoir, un peu plus évolué comme un capteur de température, ou très complexe comme le module Camera RPi. Il ne s'agit pas forcément de matériel physique – par ex. un flux de l'internet avec la température à Miami ou l'heure de l'horloge du RPi peuvent tous deux être des entrées. Les informations qu'elles renvoient peuvent ensuite alimenter le logiciel pour l'étage intelligent.

Le logiciel, le code que vous écrivez, peut être extrêmement simple ou très élaboré, mais sa fonction est toujours de prendre les informations des entrées et de leur appliquer un ensemble de conditions pour décider quoi faire. Par exemple, un bout de logiciel pourrait envoyer un signal de sortie vers une LED seulement lorsqu'on a pressé deux boutons, ou décider de déplacer un robot en fonction de la détection d'obstacles par un capteur de proximité.

Enfin, en fonction des signaux d'entrée lus et des décisions prises par le logiciel, l'ordinateur doit faire quelque chose. Pour cela, il lui faut des sorties pour passer du logiciel au monde réel. Cela peut être aussi simple qu'une LED, un peu plus compliqué comme un servo de robot, voire très sophistiqué comme le mécanisme d'une imprimante 3D. Encore une fois, les sorties n'ont pas besoin d'être physiques – par ex. un bout de logiciel peut envoyer un Tweet en guise de sortie.

Le type de projet que vous créez dépend de vous, que ce soit pour le plaisir ou pour résoudre un problème pratique. On trouve beaucoup de tutoriels sur des projets complets, dans les pages du MagPi ou sur le web, nous ne freinerons donc pas ici votre créativité en vous expliquant quoi faire et comment le faire. Au lieu de cela, nous allons essayer de vous aider à concrétiser vos idées, en vous montrant quelques-unes des briques d'électronique et de logiciel dont vous aurez besoin.



Le connecteur d'extension GPIO du RPi a 40 broches, mais bien plus que 40 fonctions.

CONSEIL DE MAKER

Lorraine Underwood

« Ne vous méprenez pas en pensant qu'il vous faut des plaques d'expérimentation et de délicates résistances pour utiliser le RPi comme un pro. Le déclic m'est venu lorsque j'ai appris qu'on pouvait relier directement des lampes RVB au RPi. J'ai fabriqué et codé ces éclairages d'escalier qui dépendent de la météo en moins d'une heure. Chaque couleur représente 5°C de la température extérieure. »

magpi.cc/cwenYY

Connecteur d'extension GPIO

Le connecteur d'extension à 40 broches du RPi offre beaucoup plus de possibilités qu'il n'a de broches. Pour vous aider à l'utiliser dans vos projets, nous allons vous expliquer ce qu'il fait.

01 Port GPIO, tout ou rien

Le moyen d'accès le plus simple à votre RPi sont ce que l'on appelle le port GPIO, ce qui signifie « general-purpose input/output » (entrées/sorties d'usage général). Ce sont des broches tout ou rien que vous pouvez utiliser comme sorties ou entrées, en envoyant ou recevant des niveaux de tension qui correspondent à des 0 ou 1 logiques. Utilisez une broche GPIO partout où une simple condition de marche ou d'arrêt est requise, par ex. pour allumer ou éteindre une LED, ou pour détecter si on a appuyé sur un bouton.

02 La vie au-delà du GPIO

On a parfois besoin de plus qu'une simple broche GPIO ; le connecteur a des fonctions secondaires qui peuvent aider. SPI, I²C et 1-wire sont des interfaces à haut débit pour certaines puces de périphériques qui étendent les capacités du RPi. Vous pouvez les utiliser avec un écran LCD, un capteur de température ou un convertisseur analogique-numérique. Le HAT Sense est un exemple de connexion simultanée de plusieurs puces de cette façon.

03 Prendre soin du RPi

Toutes ces broches GPIO actives conduisent directement à l'UC de votre RPi. Donc si vous leur appliquez une tension ou un courant trop élevés, vous pouvez endommager le RPi. Ces broches fonctionnent généralement sous 3,3 V et 10 mA, donc il faut prendre des précautions, comme ajouter une résistance avec une LED. Pour utiliser votre RPi avec un dispositif sous 5 V comme un Arduino, il existe des HAT spéciaux qui protègent votre RPi avec un jeu de GPIO tamponnées pour travailler en toute sécurité.

LES FONDAMENTAUX D'UN PROJET

Avant de débuter votre périple dans le matériel RPi, vous aurez besoin de quelques outils et techniques.

CONSEIL DE MAKER

Alex Eames

« Commencez petit et facile, vous serez surpris de votre rapidité à faire croître un projet en lui ajoutant de plus en plus d'éléments, en essayant et combinant de nouvelles idées, en apprenant et recherchant sur Google au passage. En général, vous n'avez pas à réinventer complètement la roue pour concrétiser votre idée. »

raspi.tv

Pour connecter votre RPi à des composants externes, vous devez acquérir certaines compétences de base et avoir quelques outils.

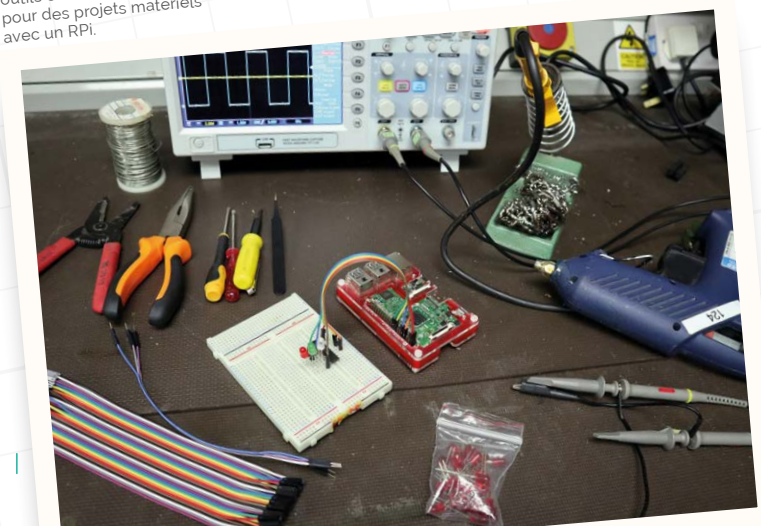
Ne vous inquiétez pas : il n'y a rien de difficile ni de coûteux, et une fois que vous les avez, vous êtes bien équipé pour vos travaux ultérieurs. Vous êtes au début du chemin pour devenir ingénieur électronicien, et les techniques que vous devrez utiliser pour connecter quelque chose à un RPi sont exactement les mêmes que celles utilisées par les professionnels, simplement à plus petite échelle.

01 Commençons avec votre établi

Votre espace de travail (ou votre établi) est l'endroit où vous faites vos expérimentations avec votre RPi et d'autres travaux électroniques. Un établi idéal est bien éclairé par une lampe puissante et dispose d'un espace dégagé entouré d'étagères faciles d'accès pour les composants et appareils, avec de nombreuses prises de courant.

Il est important d'organiser votre établi : il y aura de minuscules vis et pièces qui peuvent facilement être perdues. La soudure peut brûler les surfaces et les outils peuvent les marquer quand on ne les utilise pas avec soin. Cherchez une protection pour le plan de travail, idéalement une en silicone antistatique destinée à un usage électronique. Trouvez une solution de rangement pour les petites pièces.

L'oscilloscope est optionnel, mais devant lui se trouve les outils essentiels sur un établi pour des projets matériels avec un RPi.



Un simple fer à souder, un support, un nettoyeur de panne et de la soudure. Tout ce qu'il vous faut pour pérenniser vos projets.

02 Outils de paillasse

La panoplie standard démarre à un niveau étonnamment simple avec des outils à main. Un jeu de tournevis de précision, une petite paire de pinces, une pince à dénuder, un cutter et des « troisièmes mains » sont des choses qui vous serviront bien au-delà de votre travail avec le RPi.

Quiconque travaille avec du matériel électronique a besoin, à un moment donné, de vérifier une tension ou l'absence de celle-ci. Il faut un multimètre pour cette tâche, vous pouvez en acquérir un pour un prix étonnamment bas. Un multimètre bon marché typique comprendra un écran numérique et un sélecteur rotatif pour ses différentes gammes de tension, courant et résistance.

03 Plaque d'expérimentation

Au début, vos circuits ne seront pas permanents – vous explorerez les nombreuses possibilités offertes par votre RPi. La plaque d'expérimentation est pratique pour réaliser des circuits faciles à démonter ou reconfigurer. Disponible en plusieurs tailles, c'est un bloc en plastique avec un ensemble de lignes de contacts à ressort fixés en dessous d'une grille à trous. On réalise les circuits en insérant les pattes des composants dans les trous, les connexions se faisant via les lignes de contacts à ressort en dessous.

Outre celles entre les composants, une plaque d'expérimentation nécessite d'autres connexions, donc vous aurez aussi besoin d'un jeu de fils de liaison. Prenez-en quelques-uns avec des extrémités mâle et femelle : vous pouvez utiliser le côté femelle pour la connexion à votre RPi.

04 Soudage pour une connexion permanente

Pour expérimenter, les plaques sont parfaites, mais pas pour le long terme. Le soudage relie de façon permanente les conducteurs avec du métal en fusion pour une bonne connexion. Cela demande du soin, mais vous sera bien utile dans tout travail d'électronique.

Le fer à souder est comme un crayon dont l'extrémité atteint 350°C pour fondre la soudure. Un fer à souder peut causer de vilaines brûlures, alors faites attention. Achetez un support avec votre fer, et de quoi nettoyer la panne, comme une poignée de copeaux de cuivre ou une éponge humide.

Un fer à souder bon marché suffira, mais achetez-en un à contrôle de température si vous pouvez vous le permettre.

CONSEIL DE MAKER

Archie Roques

« Prenez soin d'examiner toutes les interfaces disponibles sur le RPi lorsque vous en utilisez une dans un projet matériel. Souvent, la première qui vient à l'esprit n'est pas la plus simple à long terme ! »

roques.xyz

« Pour connecter votre RPi à des composants externes, vous devez acquérir certaines compétences de base et avoir quelques outils. »

05 Pas besoin de tout faire vous-même

Il faut savoir que certaines choses sont difficiles à faire et qu'il n'y a pas de honte à les obtenir de quelqu'un d'autre. S'il vous faut une impression en 3D par exemple, votre fablab local peut vous aider. Un HAT de commande de moteur du commerce sera beaucoup plus fiable qu'un fait maison, et un châssis de robot prêt à l'emploi vous économisera d'énormes quantités de travail. Beaucoup de capteurs et composants semi-conducteurs sont disponibles sur ce que l'on appelle des cartes de liaison ou d'extension : le composant est monté sur un petit circuit imprimé dont les connexions sont ramenées vers des broches compatibles avec les plaques d'expérimentation.

Ce n'est pas tricher que d'acheter quelque chose dans le commerce, si cela vous aide à faire marcher votre projet !

CONSEIL DE MAKER

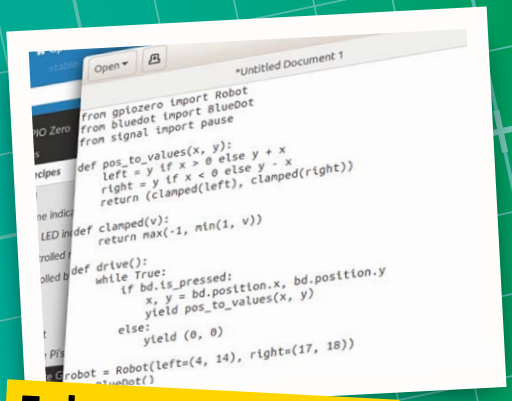
Tanya Fish

« Rappelez-vous que la communauté partage largement des exemples de code – n'ayez pas peur de modifier du code existant, avec référence à l'auteur d'origine bien sûr ! »

tanyafish.com

06 Où trouver tout ce matériel ?

Les boutiques physiques pour le matériel électronique ont presque totalement disparu. Heureusement il y a pléthore de fournisseurs en ligne pour vous expédier ce dont vous avez besoin (entre autres **www.elektor.fr**). Explorez les sites web des revendeurs de Raspberry Pi, visitez les stands lors des rassemblements de makers. Vous trouverez également des offres sous forme de kits d'outils ou de matériel de soudage.



Faire marcher tout ça avec GPIO Zero

Nous avons vu la partie physique de la construction d'un projet matériel RPi, mais qu'en est-il du logiciel ? Heureusement, c'est aussi simple que possible avec Python, le langage de programmation point de départ pour les codeurs RPi. La programmation du matériel s'effectue à l'aide d'une bibliothèque Python nommée GPIO Zero, fournie par défaut avec la distribution Raspbian. Son interface simple vous conduit directement au matériel et en plus ne nécessite qu'un minimum de code ; elle prend en charge nativement une énorme variété de capteurs et périphériques de sortie. Pour plus d'informations sur GPIO Zero, lisez le tutoriel à la page 60.

PROJETS RPI CONCRETS

Inspirez-vous de ces projets destinés aux débutants

Ordinateur portable

Raspberry Pi

Cet ordinateur portable RPi est joli et a été réalisé avec des outils ordinaires

magpi.cc/bbGxDF

Il y a plein de projets d'ordinateur portable RPi sur le web, mais beaucoup sont confectionnés avec des techniques chères comme la découpe laser, auxquelles tous les lecteurs n'ont pas accès. Nous en avons donc choisi un qui utilise des matériaux courants et des outils à main et électriques du quotidien. L'ordinateur portable de Chitlange Sahas est semblable à beaucoup d'autres par l'utilisation d'un écran d'ordinateur portable avec une carte pour la conversion HDMI. Deux batteries externes alimentent respectivement l'écran et le RPi. Un petit clavier USB complète l'ensemble en lui donnant une forme conventionnelle en clapet.

CONSEIL DE MAKER

Brian Corteil

« Même le plus grand des projets peut être décomposé en éléments simples pour y travailler. Méfiez-vous de la dérive des fonctions, et soyez prêt à laisser tomber des fonctions pour atteindre votre objectif. Ce n'est pas mal d'échouer : l'échec fait de vous un meilleur créateur quand vous apprenez de vos erreurs ».

core-tec.co.uk

Pilotage de Neopixels avec un Raspberry Pi

De jolis éclairages, un incontournable de beaucoup de projets de makers.

magpi.cc/siviiT
www.elektor.fr/neopixel-led-strip

WS2812, parfois appelé NeoPixel, est une puce à LED adressable, que l'on peut connecter en cascade et qui peut afficher toutes les couleurs de l'arc-en-ciel par commande logicielle. Vous les verrez dans énormément de projets, et savoir les piloter est utile dans l'arsenal de tout maker. Un tutoriel et du logiciel d'Adafruit permettent de piloter des NeoPixels avec un RPi, avec beaucoup de configurations matérielles différentes pour diverses applications. Les puces WS2812 sont proposées en rouleaux ou sur des circuits imprimés de diverses formes (bagues, cartes à coudre…).

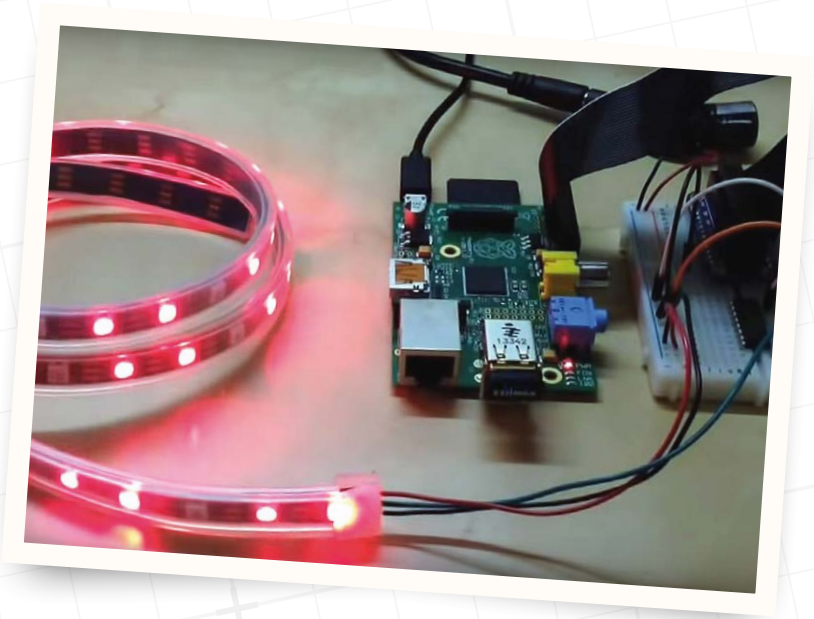
« Toute l'électronique de commande du moteur est prise en charge par le fabricant du servo. »

Pilotage de servos avec un RPi : le MeArm

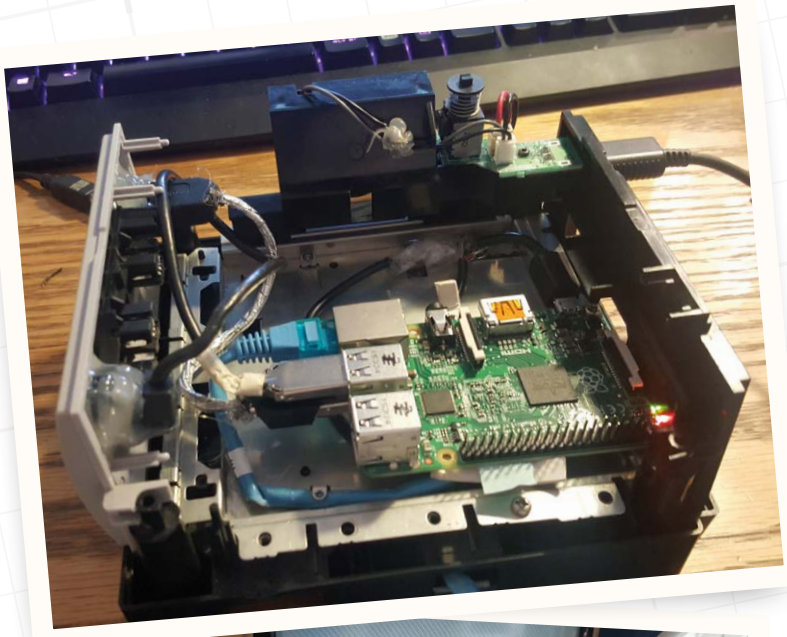
Des servos donnent vie à ce bras robotisé.

magpi.cc/QXDsfQ
www.elektor.fr/mearm-pi

Vous voulez que vos projets aient des parties mobiles, un modeste servo offre un moyen simple et bon marché pour y arriver. Le MeArm est un projet robotique célèbre que vous pouvez acheter en kit ou essayer de réaliser à partir de zéro. Toute l'électronique de commande du moteur est prise en charge par le fabricant du servo, donc il ne vous reste plus qu'à l'alimenter et à gérer la MLI depuis votre RPi. Le monde passionnant de la robotique vous attend !



RASPBERRY PI CAMERA MODULE

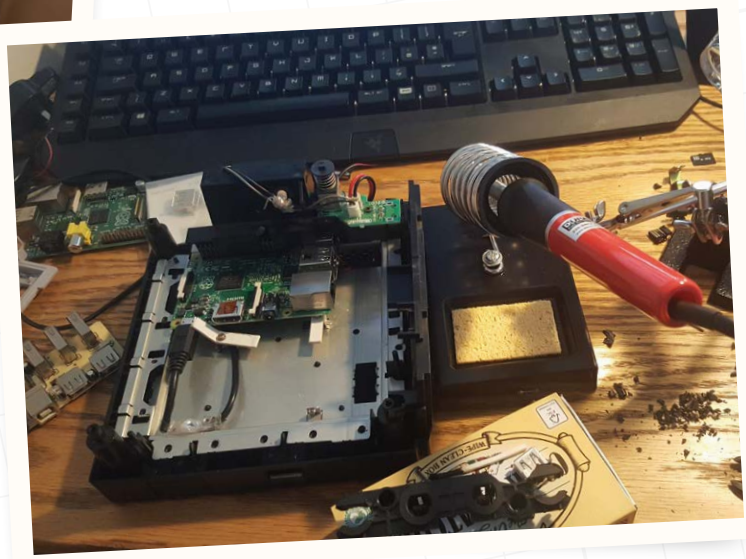


Renaissance d'une console rétro avec un RPi

Cette GameCube Nintendo devait finir à la décharge, sauf qu'un RPi l'a sauvée.

magpi.cc/tztAHx

RetroPie est un projet bien soutenu qui transforme votre RPi en une puissante machine de jeu rétro multiplateforme. Liam McLoughlin avait une GameCube Nintendo qui ne fonctionnait plus, alors avec un RPi, un outil multi-usage de type Dremel, et une bonne dose de colle thermofusible, il en a fait la parfaite console RetroPie. Comble du raffinement, il a pu réutiliser l'alimentation d'origine de la GameCube. Les makers logent des machines RetroPie dans toutes sortes d'enveloppes, depuis les consoles portables avec écrans LCD, en passant par les consoles rétro comme celle-ci, jusqu'aux bornes d'arcade grandeur nature avec des écrans cathodiques.



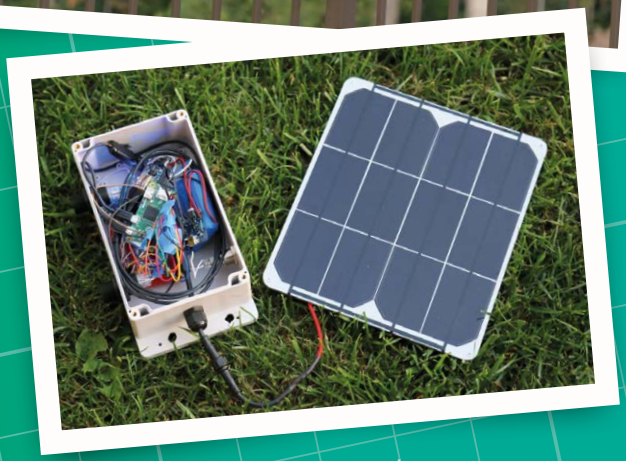
ø26MM

CONSEIL DE MAKER

Rachel 'Konichiwakitty' Wong

« Rejoignez la communauté des makers sur Twitter. C'est plus simple pour demander de l'aide, en particulier si vous êtes autodidacte et ne connaissez pas les bons termes, pour prendre une photo du problème et la partager avec le hashtag #makerhelp ou bien déclarer un incident sur un service client comme **@Raspberry_Pi** ».

magpi.cc/YxPqFw



Une station météo à énergie solaire

Suivez la météo avec un Raspberry Pi.

magpi.cc/xawnkY

Brian Masney a transformé un RPi en une station météo enregistreuse complète avec capteurs de vent, de pluie, de température, d'humidité et de pression. Le RPi Zero est installé dans un boîtier étanche avec un jeu de batteries au lithium-ion, et surmonté d'un panneau solaire. Les jours d'hiver, il n'y a pas assez de soleil pour qu'elle fonctionne toute la nuit, alors il prévoit d'installer une plus grosse batterie. Son propre logiciel YADL complète le projet, en fournissant une interface de navigation avec de nombreux graphiques pour montrer les tendances de la météo.

« Une station météo complète avec capteurs de vent, de pluie, de température, d'humidité et de pression. »

Un radiateur qui suit vos mouvements

Ce radiateur à Raspberry Pi piste tous vos mouvements.

magpi.cc/uObdem

Le module Camera Raspberry Pi peut faire beaucoup plus que capturer des instantanés ou des vidéos. Avec le sien et un logiciel de détection de mouvement, Matthias Wandel se maintient au chaud dans son atelier en gardant un radiateur à infrarouge toujours pointé sur lui. Le radiateur est placé sur une plateforme rotative, pilotée par un moteur pas-à-pas depuis le RPi, via une carte contrôleur, et tourne pour pointer automatiquement sur lui. Le logiciel est son propre package Imgcomp, bien qu'un résultat similaire puisse être obtenu avec le célèbre OpenCV. **M** (VF : Denis Lafourcade)

HAT Breakout Garden

SPÉCIFICATIONS

CONNECTIQUE : 6× connecteur encartable à 5 broches au pas de 0,1"

COMMUNICATION : bus I2C partagé

EN BONUS : barrette de liaison à 20 broches

▶ Pimoroni ▶ **www.elektor.fr/18951** ▶ 19,95 €

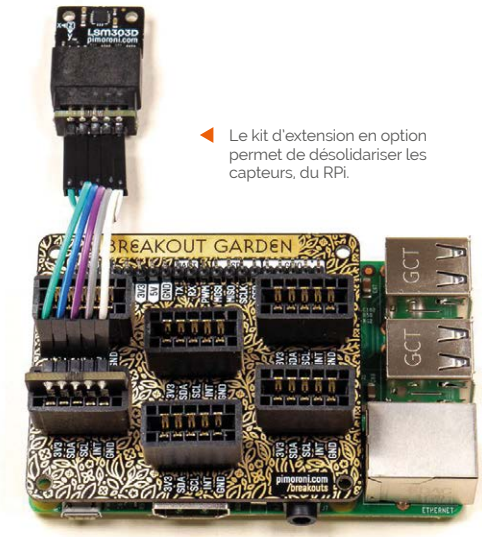
Comment développer ses projets tout en s'épargnant les tracas du soudage et du câblage. **Phil King** est allé piocher dans le jardin Pimoroni.

Le *HAT Breakout Garden* est une véritable aubaine pour les makers qui veulent se consacrer au codage et à la création tout en s'épargnant les tracas du soudage et des câblages complexes.

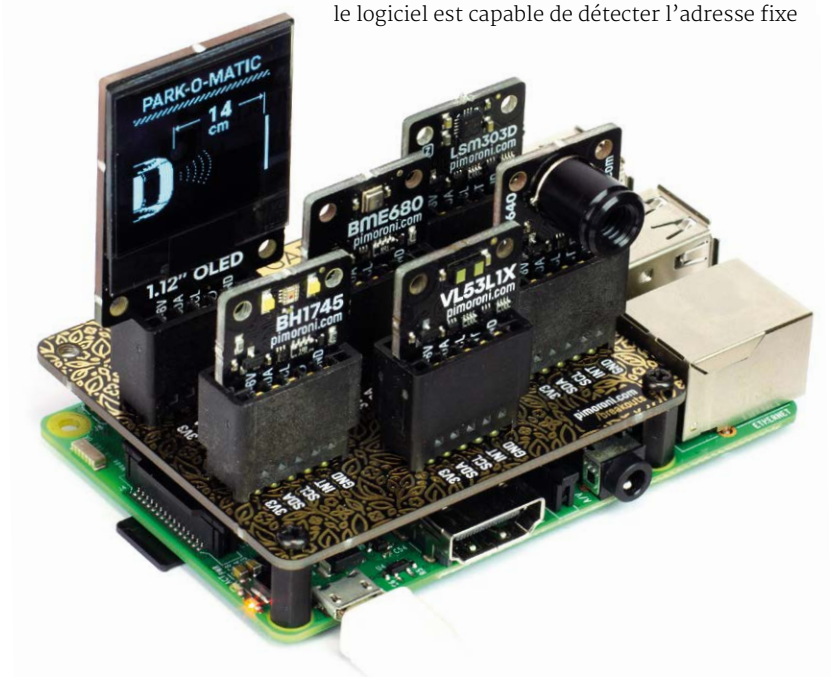
En effet, à l'instar d'autres systèmes de « connecteurs » prêts à l'emploi tels que Grove, ce HAT permet de brancher diverses petites cartes de liaison (capteurs, petits écrans et même caméra thermique). Actuellement, Pimoroni en propose une vingtaine (cf. gamme sur **magpi.cc/TsRuUJ**).

Le *HAT Breakout Garden* peut accueillir jusqu'à six cartes. Insérez celles dont vous avez besoin pour partir à l'aventure. Bien enfoncées dans les connecteurs, ces cartes ne risquent pas de tomber, même si elles basculent légèrement quand on les touche, surtout quand elles ont la taille d'un écran OLED.

Aucun risque d'inversion de polarité. De plus, le logiciel est capable de détecter l'adresse fixe I²C de chaque carte, de sorte que vous pouvez les changer d'emplacement sans aucun risque. Les broches GPIO ne manquent pas puisque les cartes se partagent le bus I²C. Le connecteur mâle fourni avec le kit permet d'en récupérer 20. Le léger décalage par rapport à la barrette de liaison assure le maintien en place (vous pourrez le souder si vous avez des doutes).



◀ Le kit d'extension en option permet de désolidariser les capteurs, du RPi.



▼ HAT bourré de cartes de liaison et notamment d'un écran OLED illustré avec l'exemple du capteur de marche arrière de voiture.

Installation du logiciel

Le logiciel s'installe sans difficulté. Même si l'installateur d'une ligne de commande Curl de Pimoroni ne fonctionnait pas au moment d'écrire ces lignes, nous avons pu installer le logiciel après avoir téléchargé et décompressé le dépôt GitHub Breakout Garden (**magpi.cc/MvzTSO**). Il suffit de taper *sudo ./install.sh* après avoir inséré les cartes dans les connecteurs du HAT. Par contre, il nous a fallu installer manuellement la bibliothèque de la matrice RVB 5×5 pour qu'elle puisse fonctionner.

▲ Pas besoin de les souder, les cartes s'enfichent tout simplement dans les six connecteurs du HAT.

▶ Matrice RVB de 5×5 et écran mono OLED.

Le dépôt GitHub propose plusieurs exemples écrits en Python pour vous mettre dans le bain. Nous avons commencé par le projet météo en nous servant du capteur BME680 et du petit écran mono OLED de 128×128 pour afficher les résultats, la température ainsi que la pression barométrique. Ensuite, nous avons utilisé le capteur de mouvement LSM3030D 6DoF et l'écran OLED pour servir de sismographe afin de capter l'arrivée d'un « dinosaure », bref la nôtre ! Parmi les autres exemples disponibles, citons celui du capteur de temps de vol VL53L1X piloté par laser qui sert de capteur de marche arrière d'une voiture en affichant la distance entre un objet et le véhicule. Nous avons également adapté cet exemple pour représenter la distance avec des carrés colorés sur l'écran matriciel RVB de 5×5.

« Corbeilles suspendues »

Même si ce *HAT* est bien pratique pour le prototypage logiciel avec des mini-cartes de liaison, certains projets requièrent néanmoins qu'un capteur de distance ou de température s'éloigne du RPi pour assurer une lecture plus précise.

Dans ce cas, il sera préférable de souder les broches (fournies avec des connecteurs à angle droit) aux mini-cartes de liaison et de les connecter au RPi soit directement soit via une plaque d'essai pour pérenniser le projet. Sinon, Pimoroni propose un « Garden Extender Kit » (**magpi.cc/xFojdG**) pour étendre la portée de vos cartes de liaison à l'instar des corbeilles suspendues. Il existe aussi d'autres extensions et des câbles volants qui vous permettront d'utiliser n'importe quel autre connecteur I²C avec le HAT. La polyvalence de ce dernier n'est plus à démontrer.

Si vous préférez un facteur de forme plus petit, vous pourrez également vous procurer un *pHAT Breakout Garden* à la taille du RPi Zero, mais ce dernier n'aura que trois connecteurs. M

(VF : Pascal Duchesnes)

> « Le HAT Breakout Garden peut accueillir jusqu'à six cartes. Insérez celles dont vous avez besoin pour partir à l'aventure. »

Notre avis

En offrant un prototypage prêt à l'emploi, ce HAT chic et pratique favorisera votre créativité en vous épargnant les tracas du soudage et du câblage.

9/10

Snips Reconnaissance vocale pour le Raspberry Pi

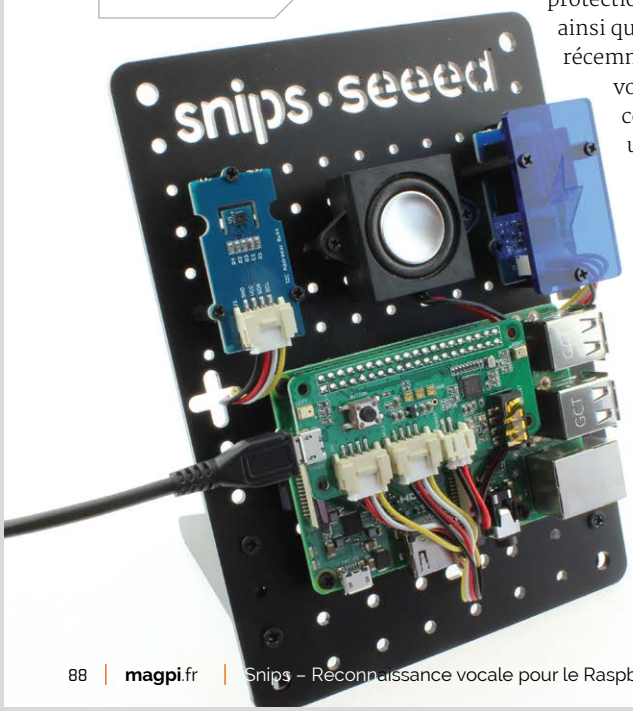
SPÉCIFICATIONS

- 1× Raspberry Pi 3B+
- 1× HAT Re-Speaker 2-Mics
- 1× relais Grove relais
- 1× capteur de température et d'humidité Grove
- 1× haut-parleur 6 Ω/2 W
- 1× carte micro-SD
- 1× bloc d'alimentation avec connecteur micro-SD
- 2× cable Grove
- 1× plaque de montage avec quincaillerie

▶ Seeed ▶ **www.elektor.fr/snips-base-kit** ▶ 119,95 €

Les assistants vocaux comme Alexa, Google Home ou Siri sont aujourd'hui devenus monnaie courante. **Mathias Claussen** parle maintenant à son horloge.

« Thé, Earl Grey – chaud » – il y a 10 ans, la possibilité pour un ordinateur d'apporter une réponse appropriée à ce type de demande vocale tenait de la pure science-fiction… Mais c'était avant l'apparition d'Alexa, Google Home et Siri. Une fois l'assistant activé, tout ce qui est collecté est directement envoyé aux serveurs du fournisseur respectif. La reconnaissance vocale n'est pas traitée localement, mais quelque part dans le nuage, chez le fournisseur. En tant qu'utilisateurs ou développeurs d'applications, nous ne savons rien de ce que deviennent ces informations. Nous ne pouvons que faire confiance aux fournisseurs pour qu'ils les traitent dans le respect des réglementations applicables en matière de protection des données. C'est ainsi qu'un juge américain a récemment cité un assistant vocal Alexa d'Amazon comme « témoin » dans une affaire de meurtre. Mais il existe des alternatives, comme Snips (**snips.ai**) – logiciel de reconnaissance vocale fonctionnant intégralement hors connexion, qui veille à ce que vos données restent confinées dans les limites de votre réseau domestique. Il ne nécessite ni connexion Internet permanente, ni batteries d'ordinateurs pour assurer la reconnaissance vocale. Le kit d'interaction vocale Snips basé sur RPi 3B+ permet de construire son propre assistant vocal local.



◀ Le kit de base d'interaction vocale Snips contient tout ce dont vous avez besoin pour créer votre propre assistant vocal.

Contenu du kit

Le kit est livré dans une boîte en carton bleu uni. À l'intérieur, vous trouvez tous les composants, avec le RPi 3B+, un adaptateur secteur 5 V/3 A, un relais, un capteur de température et d'humidité, un haut-parleur, un HAT ReSpeaker 2-Mics pour RPi, une plaque de montage, une carte SD avec le système préinstallé et un guide de démarrage rapide.

Vous avez donc à portée de main tous les éléments nécessaires pour construire votre propre petit assistant vocal. Pour l'assemblage, utilisez les entretoises et les vis appropriées. Malheureusement, le manuel a omis un certain nombre de détails, et il nous arrivera de nous demander à quoi servent certaines vis. Après quelques tâtonnements, nous avons quand même réussi à nous en sortir, même si la carte RPi n'est pas idéalement positionnée pour accéder facilement à la trappe d'accès à la carte SD.

Le kit contient une plaque de montage en acrylique sur laquelle vous pouvez fixer les différents modules à l'aide de vis autotaraudeuses. La matrice des trous de fixation autorise un certain degré de liberté pour l'implantation des modules.

HAT ReSpeaker 2-Mics pour RPi

Le cœur du système — hors RPi — est le HAT ReSpeaker 2-Mics pour RPi, construit autour d'un codec audio WM8960. Ce codec comporte un

▲ Snips est un assistant vocal qui fonctionne complètement hors ligne.

amplificateur de classe D, capable de fournir 1 W sous 8 Ω par canal, et permet de raccorder des microphones. La carte possède donc deux micros ; les connexions nécessaires pour un haut-parleur ou un casque sont également disponibles.

La carte comporte également trois LED RVB (APA102), connectées au RPi par l'interface SPI (*Serial Peripheral Interface*). Ces témoins indiquent le moment où l'assistant vocal commence à écouter (ou n'a pas compris). Un bouton-poussoir permet également une interaction avec le système. Vous disposez aussi de connexions pour le relais, et le capteur de température et d'humidité. Malheureusement, la connexion de la barrette n'est pas traversante, ce qui empêche l'ajout d'un autre HAT. Il est donc impossible d'utiliser un afficheur SPI-TFT comme interface utilisateur.

Parler à son RPi

Après assemblage des différents composants, le système est prêt pour un premier essai. Prononcez ceci : « Hey Snips - What's the temperature? ». Le système devrait vous répondre en vous indiquant la température. Ensuite, dites : « Hey Snips – Turn the relay on. » pour activer le relais.

En principe, ces commandes devraient fonctionner directement sans avoir à modifier la configuration. Nous allons maintenant aborder la partie la plus intéressante en découvrant comment réaliser nos propres applications et commandes

> « Grâce à Snips, nos données ne partent pas on ne sait où dans le nuage. »

à reconnaissance vocale. À titre de petite démonstration, nous demanderons à l'assistant vocal de nous donner l'heure.

Quelle heure est-il ?

Pour que l'application comprenne une commande vocale, nous devons la soumettre à un apprentissage pour définir une intention (ou *intent* en abrégé). Nous souhaitons que le système écoute (et entende) le mot « Time », ainsi qu'un certain nombre de synonymes.

Quand l'apprentissage est terminé, nous nous occupons de l'action que notre commande vocale doit déclencher. Avec quelques lignes de code Python, nous définissons dans la console Snips l'action de notre assistant lorsqu'il reconnaît « Time » ou un des synonymes. Pour la réponse, nous créons une chaîne, qui est ensuite adressée au module logiciel de conversion texte-voix, chargé de donner l'heure vocalement.

Pour tester notre première application, il suffit de la transférer sur le RPi. Ensuite il ne reste plus qu'à demander l'heure : « Hey Snips, what's the time? »… M

Notre avis

Ce kit contient tout ce dont vous avez besoin pour découvrir la commande. Malheureusement, les signaux produits par les deux microphones intégrés sont de mauvaise qualité et entachés de bruit. Toutefois la reconnaissance vocale n'en est pas affectée. Soyez très attentif(ve) pour l'assemblage du matériel. Une ou deux pages supplémentaires dans le manuel avec des illustrations représentant la mise en place des entretoises auraient été très utiles. Une fois familiarisé(e) avec l'interface web Snips.ai, c'est un plaisir de travailler avec le système et d'explorer toutes ses possibilités.

7/10

La vaste communauté des radioamateurs

Jean-Louis Truquet (indicatif F5DJL), président de l'association « Réseau des Émetteurs Français » nous parle des passionnés des ondes radioélectriques ainsi que de leurs multiples activités.

Qui sont donc les radioamateurs aujourd'hui ?

Depuis un siècle maintenant, c'est une communauté de passionnés des ondes électromagnétiques et d'électronique, mais aujourd'hui également d'informatique et de traitement numérique du signal. Les radioamateurs construisent leurs récepteurs, leurs émetteurs et les autres éléments de leur station. Ils expérimentent de nouvelles antennes ou de nouveaux modes de communication. Ils échangent via les ondes directement, via des satellites amateurs et même par réflexion des ondes sur la Lune. Ils se livrent à toutes ces activités soit à titre individuel depuis leur domicile, soit à titre collectif dans des radio-clubs (des installations assemblées bénévolement par leurs soins).

Cette communauté regroupe trois millions de pratiquants dans le monde et environ 14.000 en France. Lorsqu'il a réussi son examen, un radioamateur reçoit un indicatif personnel délivré par l'administration ; en France, c'est un indicatif du type FXXXX. C'est à ce titre qu'il peut alors utiliser les fréquences allouées au niveau international au service amateur.

Outre les activités expérimentales et éducatives, les radioamateurs « valident » leurs installations en échangeant sur les ondes des messages dont la teneur ne peut être ni de nature privée, ni commerciale, politique ou religieuse. Les radio-clubs sont aussi identifiés par leur indicatif.

Que fait un radioamateur ?

Les facettes du radio amateurisme sont très nombreuses et il serait illusoire de toutes les décrire en quelques lignes. Certains radioamateurs construisent et expérimentent en utilisant les techniques les plus modernes (SDR, DSP, FPGA…) alors que d'autres restaurent de vieux appareils. D'aucuns échangent des images en utilisant les dernières technologies de télévision numérique.

Certains radioamateurs participent à des activités éducatives auprès des jeunes dans le cadre des liaisons radio établies entre les écoles et la station spatiale internationale (ISS). Ces démonstrations peuvent avoir lieu lors de manifestations scolaires : Fête de la science, introduction à la radioélectricité…

Les radioamateurs participent aussi à des compétitions amicales permettant de comparer les performances de leurs installations. Enfin les radioamateurs ont un rôle citoyen, car comme nous le constatons régulièrement lors des évènements dramatiques, lorsque les infrastructures de communication sont détruites, ils

▲ Ensemble émetteur-récepteur de télévision numérique amateur. Récepteur : REF-Minitiouner F6DZP/F1TE reçoit des images du satellite amateur QO-100. Émetteur de télévision numérique DVB : un Raspberry Pi 3B+ pilote une platine SDR LimeSDR.

sont capables de mettre en place rapidement des communications de secours au service de tous.

Pourquoi être radioamateur aujourd'hui ?

Certes aujourd'hui un smartphone permet d'échanger facilement avec tous les points de la planète, donc la magie d'entendre la voix ou de voir une image d'un correspondant lointain a un peu diminué. Mais les objectifs du radioamateurisme demeurent, à savoir expérimenter et éduquer, ce sont les fondements mêmes de l'existence des radioamateurs telle que définie internationalement. La communauté de radioamateurs continue à développer le goût de la technologie et des sciences, à en faire découvrir de nouveaux aspects, les améliorer ou les rendre plus accessibles avec des moyens simples : un exemple proche de votre magazine est la télévision numérique en utilisant notre cher Raspberry Pi. De multiples autres applications du Raspberry Pi existent d'ailleurs dans la communauté des radioamateurs.

Même si l'activité peut être pratiquée individuellement, c'est aussi une grande chaîne de l'amitié entre ses pratiquants. Grâce à ses radio-clubs, dont certains sont devenus de véritables Fab Labs des ondes, et aux associations, les radioamateurs sont presque une forme de réseau social de passionnés qui échangent et partagent leurs expériences. Certains disent même que c'est le premier réseau social de l'histoire. La synergie avec la communauté des makers est d'ailleurs évidente et de nombreuses activités ou évènements sont communs : des makers sont devenus radioamateurs et des radioamateurs deviennent makers.

▲ HAMEXPO, le grand rendez-vous national des radioamateurs français. Allez les rencontrer !

Comment devient-on radioamateur ?

On trouve en ligne des livres de formation et des cours, mais surtout il existe des radio-clubs dans toutes les régions françaises. Ils dispensent des cours de formation à l'examen et ils sont avant tout des lieux d'échange, d'expérimentation et de rencontre, et de mise en commun de matériel où toute personne intéressée pourra découvrir le monde radioamateur. Ce ne sera peut-être pas sous toutes ses facettes, mais cela permettra de prendre un premier contact. La formation n'est pas si difficile et surtout ce sera le début d'un voyage passionnant à la découverte des ondes et d'une communauté. M

Jean-Louis TRUQUET, F5DJL, est le président de l'association du Réseau des Émetteurs Français (REF). L'association, reconnue d'utilité publique, regroupe près de 5.000 radioamateurs français et existe depuis 1925. Elle est membre fondateur de l'IARU (Union internationale des radioamateurs). Le site de l'association est **www.r-e-f.org**. Le paysage radioamateur français comprend de multiples associations actives, dont certaines sont plus spécialisées dans un domaine d'activité particulier : les communications par satellite, la télévision, le rôle citoyen et l'urgence.... Le REF organise le samedi 12 octobre 2019 au Parc des Expositions du Mans le grand salon radioamateur français : HAMEXPO. Lors de ce salon seront présentées de multiples activités en présence de nombreux acteurs associatifs et commerciaux du domaine, une excellente occasion de visiter ce monde de passionnés des ondes.

Nikole Vaughn

Est-il possible d'être dirigeant et de gérer également un CoderDojo pour des filles scoutes ? Pour Nikole, la réponse est oui !

- Catégorie Organisatrice de CoderDojo
- Profession Vice-présidente
- Site web coderdojocollaborative.org

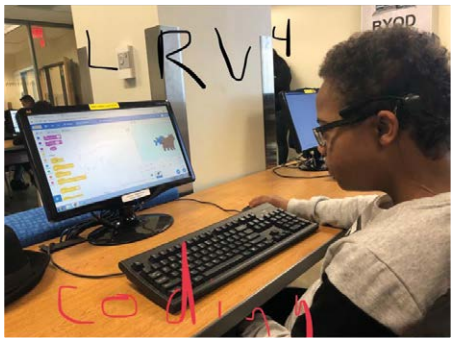
À la différence de beaucoup d'organisateurs d'événements dans la communauté éducative/Raspberry Pi, Nikole n'a jamais vraiment prévu d'en devenir un.

« Un groupe local sans but lucratif en cybersécurité militaire a lancé les premiers Dojos connus à San Antonio (Texas, États-Unis) », explique Nikole. « Mes enfants et moi avons rejoint (sur invitation) un Dojo pour apprendre le JavaScript. Après deux rendez-vous, nous avons appris que le lieu de rencontre allait fermer et que le Dojo cherchait un nouvel endroit. J'enseigne à temps partiel pour l'université de Phoenix (UOP) à San Antonio, ainsi le Dojo a été déplacé là-bas pour les sessions déjà programmées. En étant à l'UOP et en accueillant les Ninjas, j'ai réalisé que je gérais le Dojo. »

Elle gère maintenant son propre CoderDojo avec Malene Lawrence : le *CoderDojo Collaborative* installé depuis deux ans au *San Antonio College*.

« Nous nous focalisons sur la collaboration et la communauté », dit Nikole. « Chacun amène ce qu'il est et offre ses talents au groupe pour que tous progressent. »

Des centaines de ninjas sont passés par le *CoderDojo Collaborative*.



▲ Scratch est un langage de programmation populaire chez beaucoup d'élèves.

▶ Il y a environ une répartition de 60/40 entre les garçons et les filles du *CoderDojo Collaborative*.

À quoi ressemble votre CoderDojo ?
Le *CoderDojo Collaborative* compte 141 Ninjas inscrits, 6 organisateurs bénévoles (Dee Arnold, Tracie Hervey, Sheri Brown, Shiketa Gresham, Malene Lawrence, et moi-même), 35 mentors professionnels bénévoles et 40 bénévoles non techniques. Nous nous réunissons gratuitement dans neuf espaces dédiés avec des ordinateurs au San Antonio College. Nous avons commencé avec vingt Ninjas, deux organisateurs, un mentor et aucun volontaire non technique. Les Ninjas choisissent une matière à couvrir pendant une période de huit à neuf semaines, puis ils peuvent s'inscrire à une matière différente après une cérémonie de promotion.

Comment les enfants ont-ils accueilli le CoderDojo ?
En trois ans, 480 Ninjas se sont inscrits. Les enfants arrivent prêts à s'amuser. Nos Ninjas affrontent souvent le dilemme du samedi matin : participer à un événement sportif avec une équipe, ou venir au Dojo. Les parents racontent que les Ninjas choisissent plus souvent le Dojo. Ils apprécient qu'il n'y ait que deux à trois séances par mois, ainsi les enfants peuvent garder leur position dans leur équipe de sport.

> « Chacun amène ce qu'il est et offre ses talents au groupe pour que tous progressent. »

Citez-nous quelques-uns de vos projets favoris réalisés au Dojo ?
Réponse de Steven Bernstein, mentor RPi au club : ce sont des projets qui ouvrent les yeux des enfants aux possibilités. Voir la LED sur la platine s'éclairer pour la première fois, grâce au code qu'ils ont créé, est grisant.

Certains Ninjas ont composé et joué de la musique en utilisant Python sur le RPi.

Maintenant les Ninjas apprennent comment créer des projets RPi portables et réactionnels en utilisant des LED. Pour les Ninjas intéressés, un mentor enseigne comment créer un hub portable Wi-Fi avec le RPi.

Les sites web *All About Me* semblent simples, mais démontrent vraiment la puissance du code. Les Ninjas y présentent leurs intérêts dans le monde numérique en utilisant le HTML, le CSS et le Javascript. Un Ninja qui voulait ouvrir une boutique de *cupcakes* a créé son propre site avec des vermicelles animés tombant du ciel sur des *cupcakes*, alors qu'un autre Ninja a créé un site expliquant son amour des jeux comme Minecraft. Les Ninjas voient que les idées de notre monde normal ont une vie dans le monde numérique.

D'autres Ninjas ont créé 70 programmes en Scratch, allant d'un livre d'histoires animé à un jeu ressemblant au ping-pong en passant par un jeu mathématique pour apprendre des notions préalgébriques. M

(VF : Thierry Destinobles)

On recherche des champions !

Vous pensez avoir des compétences pour aider des jeunes à apprendre à coder et à réaliser des projets numériques ? Alors vous pourriez devenir un bénévole CoderDojo et aider à former la prochaine génération d'utilisateurs d'ordinateurs. Pour plus de détails, voir **coderdojo.com/fr-FR**

elektor start-up challenge

Une rampe de lancement internationale pour les start-ups françaises

N'attendez pas qu'Elon Musk vous invite à bord de SpaceX !

Après l'Allemagne et les Pays-Bas, cette année Elektor fait escale en France avec un nouveau concours pour les jeunes entreprises dont l'activité est liée à l'électronique embarquée. Cela se passera à Paris dans le cadre du nouveau salon Forum de l'électronique en septembre 2019.

Ce concours baptisé *elektor start-up challenge* vise à stimuler la croissance de jeunes pousses après leurs débuts prometteurs.

Le choix des gagnants sera guidé par la viabilité industrielle des projets, leur utilité et bien sûr leur originalité.

L'*elektor start-up challenge* est le fruit de notre coopération étroite avec Cap'tronic, le programme d'aide aux PME en matière de systèmes électroniques et de logiciel embarqué, et GL Events Exhibitions, l'organisateur du salon Forum de l'électronique.

Qui peut embarquer ?

Vous par exemple, qui développez un produit, un logiciel ou un service lié à l'électronique. *elektor start-up challenge* vous donne l'occasion de venir présenter votre projet aux professionnels de la branche.

Les prédécents concours organisés par Elektor, notamment dans le cadre d'*electronica* à Munich, le plus grand salon d'électronique au monde, ont récompensé des projets remarquables dans différents domaines :

Pourquoi monter à bord de la fusée ?

Si vous vous inscrivez à l'*elektor start-up challenge* de Paris 2019 et que votre candidature est retenue, vous serez accueilli avec les autres concurrents sur l'aire de l'*elektor start-up challenge*, un grand stand au beau milieu du salon Forum de l'électronique (Paris Expo Porte de Versailles, du 24 au 26.09.2019).
Vous pourrez :

- y présenter votre projet sur votre propre pupitre
- concourir au côté des autres finalistes
- attirer l'attention de partenaires potentiels

Avant, pendant et après le salon, vous bénéficierez d'une large visibilité :

- vous et votre start-up serez présents dans les supports du Forum de l'électronique : catalogue des exposants, liste des exposants, site officiel…
- vous et votre start-up serez présentés à l'ensemble des lecteurs du magazine Elektor via tous nos supports (papier, web, réseaux sociaux, chaîne YouTube…) en France et dans les pays francophones
- vous et votre start-up serez présentés aux adhérents du programme Cap'tronic

Pour vous et pour votre start-up, le summum ce serait d'emporter le gros lot. En effet, le gagnant de l'elektor start-up challenge à Paris recevra son ticket d'entrée pour le prochain salon *electronica* à Munich : vous serez sélectionné d'office pour y participer au concours international Fast Forward Award au cours duquel vous présenterez votre projet sur le stand et bénéficierez de la très large publicité donnée à cet évènement.

WE WANT YOU!

Le rendez-vous de votre start-up avec son avenir

Où ? Paris Expo Porte de Versailles

Quand ? Du 24 au 26 septembre 2019

Comment ? Sur inscription au plus tard le mercredi 31 juillet 2019.

Rendez-vous sur :
www.elektormagazine.fr/escparis2019

▲ Participants et vainqueurs du concours *Fast Forward electronica 2018*.

▲ Le gagnant participera d'office au concours *electronica* à Munich l'an prochain.

▼ Le jury au travail, présidé par Clemens Valens d'Elektor Labs.

▲ La réunion préparatoire entre Elektor et ses partenaires.

Attention :

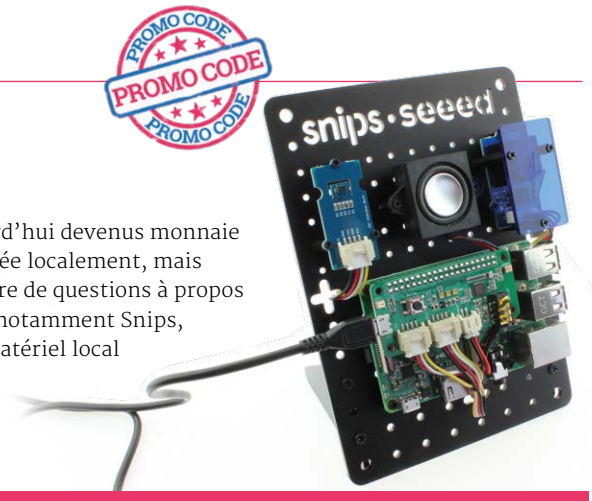
la date limite d'inscription est fixée au mercredi 31 juillet 2019.

En cas de forte affluence, il est possible que le nombre limité de places nous contraigne à avancer la date de clôture des inscriptions. Ne tardez pas à vous rendre sur la page **www.elektormagazine.fr/escparis2019** pour vous inscrire.

Les conditions générales de participation au elektor start-up challenge de Paris sont disponibles en ligne.

BANC D'ESSAI

Snips – Reconnaissance vocale pour le Raspberry Pi



Les assistants vocaux comme Alexa, Google Home ou Siri sont aujourd'hui devenus monnaie courante. Pour autant, la reconnaissance de la parole n'est pas traitée localement, mais dans le nuage, chez le fournisseur – ce qui soulève un certain nombre de questions à propos de la confidentialité des données. Il existe cependant d'autres solutions, notamment Snips, système de reconnaissance vocale qui fonctionne intégralement sur un matériel local comme le RPi.

Remise de 10 € avec le code snips10 (offre valable jusqu'au 31 août 2019 !)

Lisez le banc d'essai sur www.elektormagazine.fr/review-snips

Notre sélection :

1. Raspberry Pi 3B+	2. Raspberry Pi Zero WH	3. Raspberry Pi 3A+	4. HAT PoE pour Raspberry Pi 3B+	5. JoyPi
www.elektor.fr/rpi-3b-plus	www.elektor.fr/rpi-zero-wh	www.elektor.fr/rpi-3a-plus	www.elektor.fr/poe-hat-rpi-3-plus	www.elektor.fr/joypi

Livres

Camera Projects Book

Ce livre (en anglais) qui repose sur des projets testés et fonctionnels explique en termes simples comment configurer et utiliser une caméra Raspberry Pi et une webcam USB dans des montages animés par un RPi.

www.elektor.fr/camera-projects-book

Raspberry Pi

Avec le Raspberry Pi, pour quelques dizaines d'euros, vous disposez d'un ordinateur complet auquel vous pouvez connecter simplement toutes sortes de montages électroniques.

www.elektor.fr/rpi

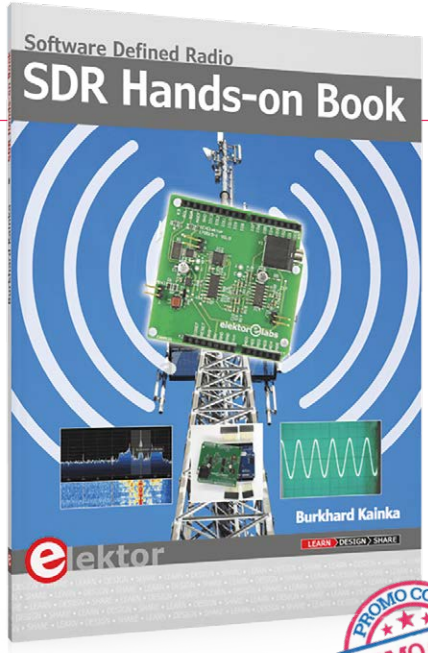
Raspberry Pi 3

Ce livre (en anglais) sur le Raspberry Pi 3 rassemble de nombreux projets programmés en Python, basés sur différents modules (Sense HAT, Swiss Pi, MotoPi, module Camera et autres capteurs analogiques et numériques). Les communications en Bluetooth et Wi-Fi sont également abordées.

www.elektor.fr/18412

Spotlight

SDR Hands-on Book



Les ondes courtes sont particulièrement attrayantes : elles permettent de franchir facilement de longues distances. Comme les signaux en ondes courtes sont réfléchis par les couches conductrices de l'ionosphère, ils peuvent être reçus à des endroits au-delà de l'horizon et peuvent donc atteindre n'importe quel point de la Terre. Aujourd'hui on court après les fréquences les plus élevées et l'on écoute la radio en FM, RNT (radio numérique), par satellite ou sur l'internet. Toutefois les moyens de transmission modernes exigent une infrastructure étendue et sont extrêmement vulnérables. En cas de panne de courant mondiale, rien ne sera plus important que les ondes courtes. Les transmissions des radioamateurs ne sont pas seulement un hobby, c'est aussi un système radio d'urgence !

Le *shield* SDR d'Elektor (réf. 18515) permet de construire un récepteur polyvalent à ondes courtes jusqu'à 30 MHz. Avec une simple carte Arduino et le logiciel approprié, il est possible de recevoir des stations radio, des signaux en morse, des stations SSB et des signaux numériques.

Dans ce livre (en anglais), l'auteur à succès et radioamateur enthousiaste, Burkhard Kainka décrit comment pratiquer la radio logicielle en utilisant le shield SDR d'Elektor. Il ne se contente pas de donner une formation théorique, il explique également comment utiliser de nombreux outils logiciels libres.

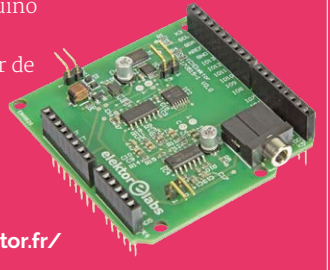
Remise de 5 € avec le code sdrbook5 (offre valable jusqu'au 31 août 2019 !)

www.elektor.fr/sdr-hands-on-book

Kits & Modules

Shield SDR Elektor SDR 2.0

Cette nouvelle version dispose de deux sorties à PLL et de deux sorties à LF, accessibles via des connecteurs supplémentaires sur la carte (non inclus dans le kit). L'utilisateur peut utiliser ce *shield* Arduino comme générateur de signal, émetteur en ondes courtes…



www.elektor.fr/18515

Kit de démarrage GrovePi+

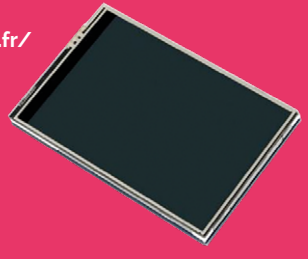
La carte GrovePi+ permet d'utiliser les modules Grove sur un RPi. Ce kit de démarrage contient cette carte qui assure l'interface (I^2C) avec le RPi ainsi que de nombreux capteurs qui compléteront à merveille votre RPi. Tous les modules Grove sont directement connectés à la carte GrovePi+.

www.elektor.fr/18749

Écran tactile LCD de 3,5" pour RPi (480×320)

Cet écran LCD de 480×320 pixels est équipé d'une commande tactile résistive et supporte toutes les versions du Raspberry Pi. Vous pouvez l'utiliser pour regarder des vidéos, jouer à des jeux vidéo, prendre des photos en le touchant, etc.

www.elektor.fr/18936



À ne pas manquer

Voici quelques autres trouvailles intéressantes.

(VF : Pascal Duchesnes)

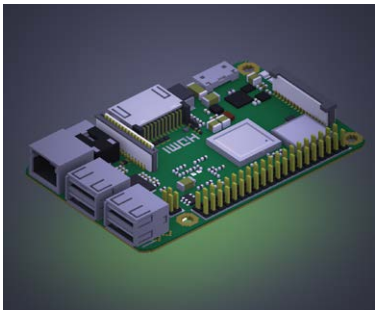
RÉVEIL NES

Quoi de plus adorable qu'un réveil NES personnalisé ! Le contrôleur n'est pas là pour la frime. En plus de servir de réveil, il permet de faire tourner RetroPie. Vous ne pourrez plus dormir sans lui !

▶ magpi.cc/wkWiac

VOXEL RASPBERRY PI

Cette reconstitution du RPi 3B+ en voxels (pixels en 3D) est fabuleuse. Et apparemment, tout le monde est invité à l'utiliser ! N'oubliez pas cependant de rendre à (l'utilisateur reddit) Cal_Macc ce qui est à Cal_Macc.



▶ magpi.cc/GQMgry

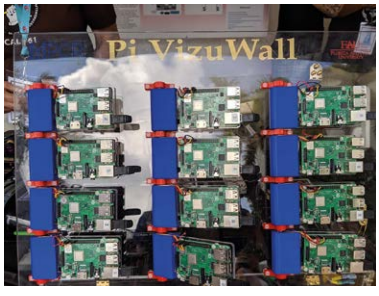
PSPI

PSP est une console de poche classique qui n'a jamais manqué d'inspirer les bidouilleurs depuis son lancement. Ce n'était plus qu'une question de temps avant qu'elle n'abrite un RPi Zero pour ranimer son écran. Voilà qui est fait. Génial !

▶ magpi.cc/RhZZTB

PI VIZUWALL

Fan de MagPi, *Estefannie Explains It All*, nous a présenté ce projet bien sympa lors de la Maker Faire Miami. Cette grappe de serveurs est là pour donner des ailes à chacun de ses RPi. Découvrez la puissance partagée dans un prochain numéro.



▶ magpi.cc/jWMpPo

COMMISSAIRE FRIGO

Avec un RPi et la vision par ordinateur via OpenCV, plus rien ne vous échappe. Prenez-les en flagrant délit, ces collègues ou colocataires qui viennent se servir à votre insu. Adrian a construit ce dispositif pour parer au vol de son précieux houmous.

▶ magpi.cc/ixrRyK

FREEPLAY CM3+

Cette version rétro de console de jeux Game Boy Advance est l'une des premières à disposer d'une commande analogique double discrète, élégante. La croix directionnelle, elle aussi, a fière allure.

▶ magpi.cc/qEVGYz

LEROY MERLIN PRÉSENTE

Maker Faire® Paris

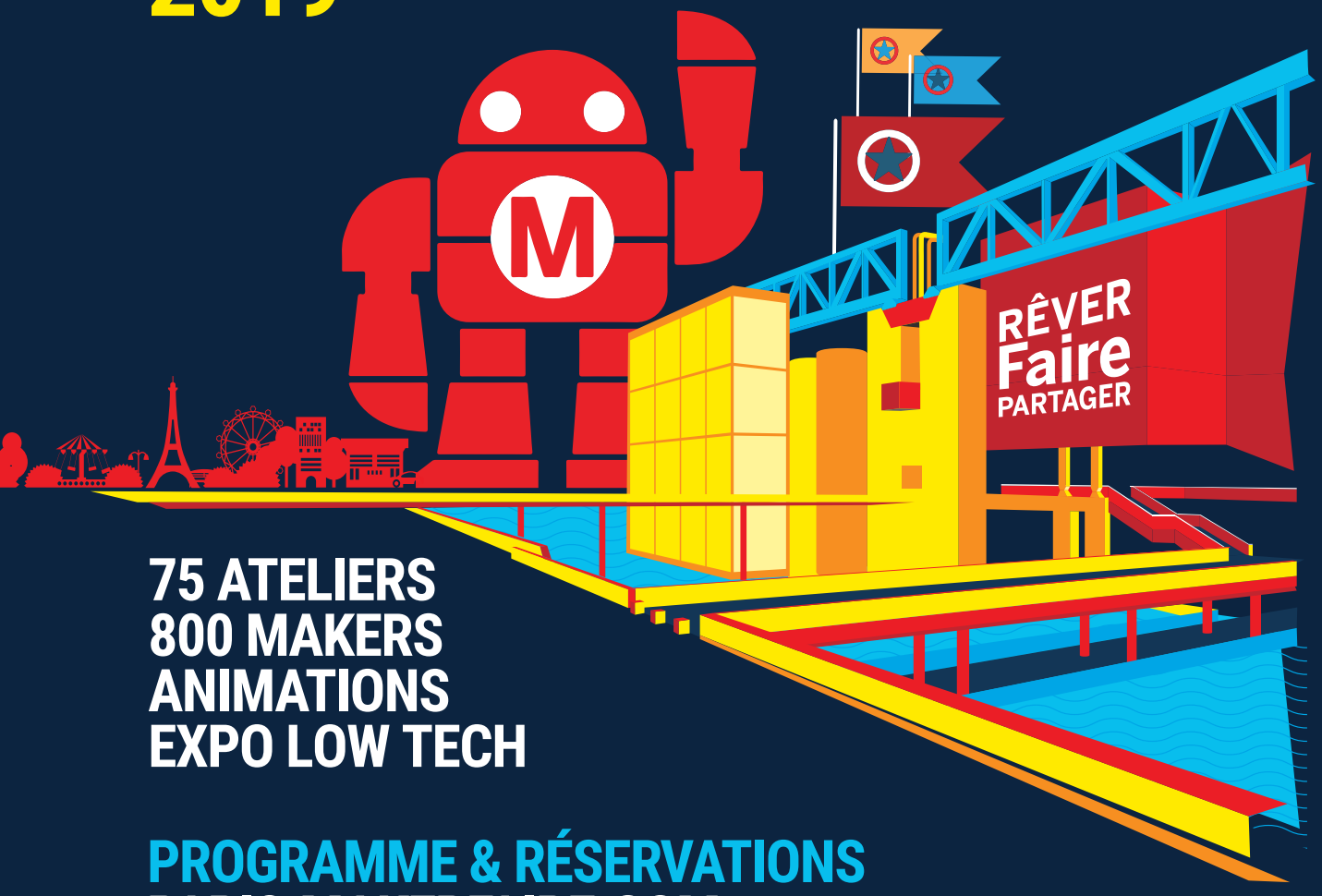
★ LE FESTIVAL DE L'INNOVATION ET DE LA CRÉATIVITÉ ★

À LA CITÉ DES SCIENCES ET DE L'INDUSTRIE

22.23.24 NOVEMBRE 2019

EN PARTENARIAT AVEC

cité sciences et industrie



75 ATELIERS
800 MAKERS
ANIMATIONS
EXPO LOW TECH

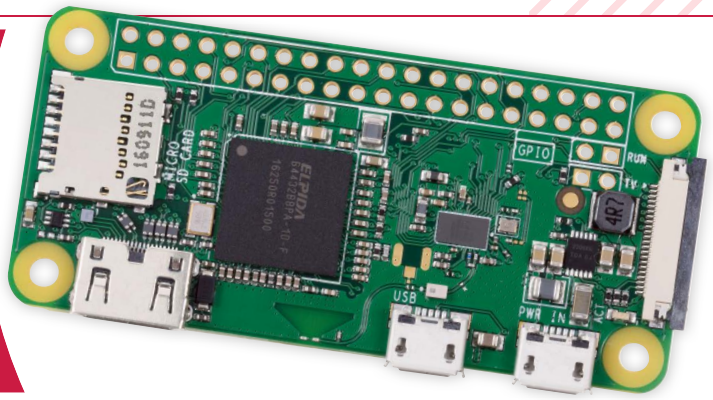
PROGRAMME & RÉSERVATIONS
PARIS.MAKERFAIRE.COM

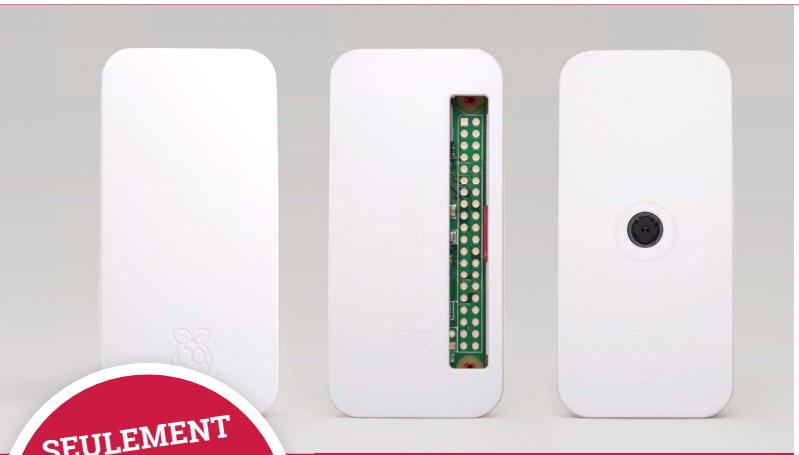
LES ATELIERS LEROY MERLIN RCS 811 687 961 LILLE

Make: makezine.com · LEROY MERLIN · RS Components · ulule · HOP

ABONNEZ-VOUS ET RECEVEZ

RPI ZERO W GRATUIT





SEULEMENT **54,95 €** PAR AN (6 NUMÉROS)

TOUS LES 2 MOIS, LES DERNIÈRES NOUVELLES DU RASPBERRY PI ET LES MEILLEURS PROJETS !

Souscrivez dès maintenant un abonnement d'un an au magazine MagPi, nous vous offrons :

- Six numéros du magazine MagPi
- Une carte Raspberry Pi Zero W
- Un boîtier avec trois couvercles différents
- Un connecteur pour module de caméra
- Un câble HDMI/mini-HDMI et un câble micro-USB/USB OTG

Vos avantages :

- Prix au numéro réduit
- Chaque numéro directement dans votre boîte aux lettres
- Tous les numéros disponibles sous forme numérique (PDF)
- Cadeau de bienvenue d'une valeur de 22,95 €
- Découverte de chaque nouveau numéro avant sa sortie en kiosque

ABONNEZ-VOUS : WWW.MAGPI.FR