

N° 09 - FEV 2013

Commandez des copies
des N° 1 à 8 plus une
reliure sur
themagpi.com



The **MagPi**™

Le Magazine des utilisateurs de Raspberry Pi

Découvrez Ladyada

Une interview de
la fondatrice
d'Adafruit
Industries

Dans ce numéro...

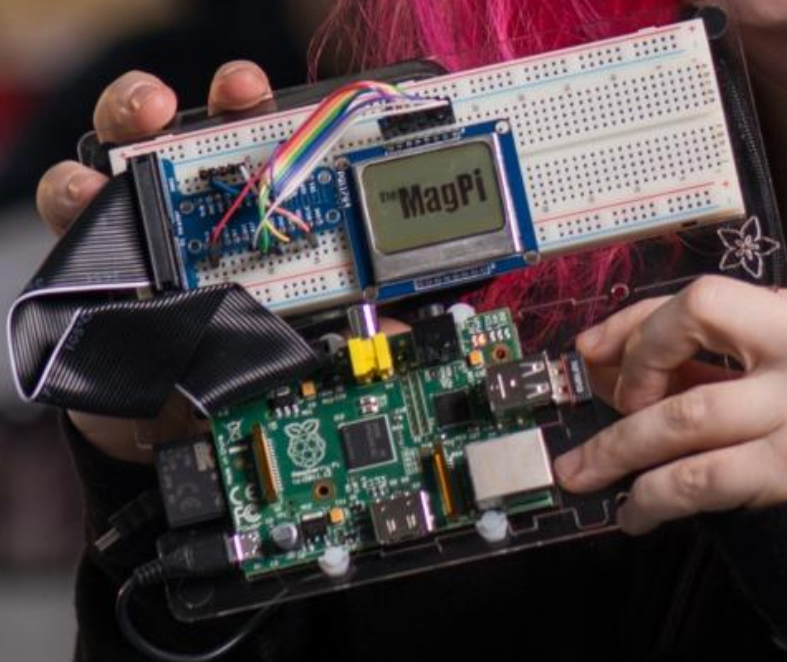
Sauvegardez votre carte SD

Le framework WebIOPi

Vala et LedBorg

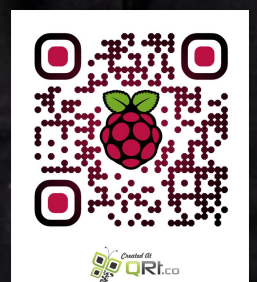
GPIO en Scratch

RISCOS



Raspberry Pi est une marque déposée de la Fondation Raspberry Pi.
Ce magazine a été créé avec un Raspberry Pi.

The **MagPi**™



<http://www.themagpi.com>



The MagPi™

Bienvenue dans le Numéro Neuf,

Un grand merci à tous ceux d'entre vous qui ont généreusement supporté notre projet kickstarter. Le kickstarter a dépassé de plus de quatre fois les résultats espérés, avec 616 souscripteurs. Après un petit retard, nous nous préparons pour l'impression. La conception de la reliure du Volume 1 (Numéros 1 à 8) est terminée. Si vous avez raté le Volume 1, vous pouvez toujours le commander sur notre boutique en ligne www.themagpi.com/shop/

Ce mois-ci, nous sommes très heureux de vous présenter une interview de l'ingénieur et fondatrice d'Adafruit Industries, Limor Fried. Elle a été un encouragement pour beaucoup de personnes, créant des tutoriels, répondant aux questions et fournissant toute sorte de matériel pour la construction de projets intéressants.

Ce numéro contient également des articles sur le matériel, des instructions d'installation de RISCOS et Arch Linux, et notre série régulière d'articles sur Scratch, Python et C.

Ash Stone

Rédacteur en Chef de The MagPi

MagPi team

Ash Stone - *Rédacteur en Chef / Administrateur*

Tim 'meltwater' Cox - *Auteur / Mise en page / Admin.*

Chris 'tzj' Stagg - *Auteur / Photographe / Mise en page*

Colin Deady - *Auteur / Mise en page*

Jason 'Jaseman' Davies - *Site Web / Mise en page*

Matt 'Othe0judge0' - *Site Web / Administrateur*

Aaron Shaw - *Auteur / Mise en page / Graphisme*

Ian McAlpine - *Auteur / Mise en page / Graphisme*

Sam Marshall - *Mise en page / Graphisme*

W. H. Bell - *Auteur / Mise en page*

Bryan Butler - *Mise en page / Graphisme*

Colin Norris - *Graphisme*

Mark Robson - *Relecture*

Alex Baker - *Relecture*

Richard Wenner - *Relecture*

Steve Drew - *Relecture*

Chosp - *Relecture*

Benjamin Donald-Wilson - *Relecture*

Mike Richards - *Relecture*

Guest writers

Alex Eames

Eric PTAK

Norman Dunbar

Pete Nowosad

Ross Taylor

Simon Walters

Couverture : Limor Fried, Ingénieur et fondatrice d'Adafruit



Sommaire

04 Une interview avec Limor Fried d'Adafruit

Limor Fried, fondatrice et ingénieur d'Adafruit Industries, répond au MagPi.

08 WebIOPi - infrastructure REST pour Raspberry Pi

Comment contrôler l'interface GPIO du Raspberry Pi depuis un navigateur Web.

12 Sauvegarder votre Raspberry Pi

Sauvegardez votre carte SD avec compression facultative et archivage sur DVD.

15 Gagnez des accessoires pour Raspberry Pi

Il est possible de remporter une carte Gertboard ce mois-ci.

16 Carte d'interface d'E/S de Quick2Wire pour le Raspberry Pi

Une description du kit et de la carte assemblée.

18 Une introduction à RISCOS

Une présentation basique de RISCOS, de l'installation sur carte SD jusqu'au bureau.

20 Installer et configurer Arch Linux

Comment installer Arch Linux, une distribution idéale pour mini-PC, sur le Raspberry Pi.

22 Une introduction à la programmation Vala

Écriture de code en Vala, un langage de haut niveau genre C#.

24 Les évènements du mois

Trouvez ce qui se passe ce mois-ci.

26 La Caverne du C - structures, histogrammes et analyse de données

Comment construire des programmes et des structures de données complexes.

32 Le Patch en Scratch - contrôler l'interface GPIO en Scratch

Les premiers pas pour contrôler le GPIO, pour permettre des interfaçages plus élaborés.

34 Le Repaire du Python - piloter le Raspberry Pi avec un téléphone portable

Une présentation de webpy, pour établir des connexions à des portables depuis des projets Python.

36 Questions et réactions

Commentaires et réactions des lecteurs.

Elle est une partisane du matériel libre, fondatrice d'Adafruit, et a été élue "Entrepreneur 2012". Qui est Limor Fried ?

[MagPi] Tout d'abord, félicitations pour le titre d'"Entrepreneur 2012" qui vous a été attribué par Entrepreneur Magazine. Pensez-vous que ce soit un signe avant-coureur d'une acceptation du "hacking", dans sa vraie définition et du mouvement "constructeur" en général ?

[Limor] Merci beaucoup pour vos paroles aimables. La communauté Raspberry Pi mérite un grand merci également, tous les votes se faisaient par Internet et la communauté Pi s'est vraiment mobilisée pour nous! Je crois que le mouvement "constructeur" a dépassé le stade "est-ce quelque chose de réel?". Il y a 6 ans j'ai été invitée à une conférence au sujet du nouveau mouvement "constructeur" qui venait juste d'apparaître et une société importante a prétendu qu'Adafruit n'était pas une vraie société. Ça a été un défi quotidien de prouver qu'une grande entreprise pouvait soutenir une grande cause comme le libre. Ma nomination comme "Entrepreneur de l'année" par Entrepreneur Magazine signifie qu'il y a moins de barrières pour quelqu'un qui démarre aujourd'hui. Ils n'ont pas besoin d'entendre que quelque chose est impossible ou irréel, ils peuvent voir qu'il y a des opportunités illimitées de création et de partage - et l'exploitation d'une entreprise prospère.

[MagPi] Nous arrivons un peu en avance nous-mêmes. Prenons un peu de recul. Vous êtes la fondatrice et ingénieure de Adafruit, la société basée à New-York que vous avez créée en 2005 après avoir obtenu une maîtrise en génie électrique et informatique du Massachusetts Institute of Technology (MIT). Qu'est-ce qui vous

a inspiré pour commencer votre propre entreprise plutôt que de "vous faire les dents" chez un employeur de l'industrie ?

[Limor] Exploiter sa propre entreprise n'est pas pour tout le monde, d'abord je n'étais même pas sûre que c'était pour moi, mais avoir la liberté et la souplesse nécessaire pour poursuivre son but et travailler sur des choses importantes est séduisant et enrichissant. Il y a une tonne de



risques bien sûr, mais le plus grand risque est de regretter plus tard de n'avoir pas essayé un tant soit peu. Il n'y a jamais eu un meilleur moment pour développer une société qui célèbre des gens intelligents, des collectivités intelligentes et l'apprentissage. La demande pour des initiatives telles que Raspberry Pi a totalement changé Adafruit. N'importe qui peut apprendre à concevoir un circuit électronique, écrire du code et avoir de multiples façons d'amener les produits dans les mains des clients. L'une des possibilités, quand vous gérez une entreprise, est que vous pouvez accepter certains projets qui ne semblent pas de prime abord avoir d'impact sur votre résultat financier, mais vous pouvez prendre le

risque. La plupart des projets que nous lançons chez Adafruit ne seraient jamais approuvés par une grande entreprise uniquement concentrée sur quelques produits. Nous avons plus de 1200 produits et certains d'entre eux sont purement expérimentaux.

[MagPi] Votre surnom de "Ladyada" a, je pense, une relation avec Lady Ada Lovelace, la première personne au monde qui programma un ordinateur ? Bien sûr, le lien vers Adafruit est plus évident,

mais quelle était la source d'inspiration derrière ce nom ? Peut-être que vous avez eu la prémonition que vous travailleriez avec des raspberries un jour?

[Limor] Quand j'étais plus jeune je n'ai fait que jouer avec Linux, l'installer sur tout ce que j'ai pu trouver et explorer toutes les choses qui le font fonctionner. A mon époque de hacker, en fait je suis encore dans cette période :)..., mon surnom était Ada. J'ai toujours programmé, construit, fait de la rétro-ingénierie aussi le nom de Ladyada me colle à la peau depuis le début. Si je considère mes bidouilles sur Linux, c'est comme si j'étais en formation pour travailler sur le Raspberry Pi. Pour les plus jeunes qui aiment passer du temps à bidouiller, ça peut vraiment finir par être une aventure fantastique et une carrière!

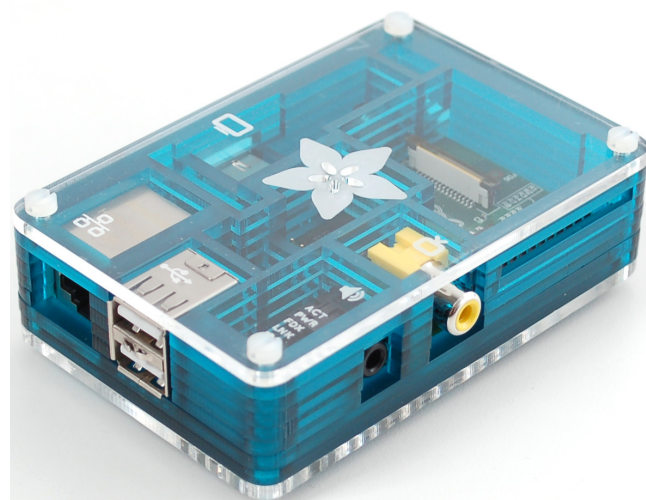
[MagPi] Nous avons tous entendu parler de logiciel open source et il est utilisé chaque jour par des millions de personnes, dont plus d'un million d'utilisateurs de Raspberry Pi. Mais vous êtes fortement impliquée dans la communauté du matériel open source. En effet, tous les produits Adafruit sont disponibles sous la forme de matériel open source avec téléchargement gratuit des schémas, des typons des circuits imprimés, du micrologiciel et du logiciel. Arduino est un exemple populaire de matériel open source, mais pouvez-vous nous en dire plus sur ce que vous pensez de l'avenir du matériel open source?

[Limor] J'aime aller à la fin de l'histoire, tout matériel est copiable. Il y a des marques de commerce pour les logos, les noms et les brevets pour certaines choses, mais si c'est fait à partir de composants et que c'est intéressant, quelqu'un va le copier. Aussi j'ai toujours travaillé à partir de cela, si quelqu'un doit copier quelque chose venant de moi, je dois faire de mon mieux pour le rendre éducatif, amusant et utile pour la société. Dans un monde où des entreprises comme Apple et Samsung se poursuivent les unes les autres, c'est assez clair que le progrès s'arrête lorsque vous pensez que vous pouvez arrêter la copie. Une façon de voir cela, ce sont les recettes : nous pouvons tous faire tous les plats que nous aimons à la maison, mais nous allons au restaurant pour une expérience. C'est comme ça que je regarde le matériel, vous n'achetez pas seulement les composants de Adafruit, vous obtenez le service, le support et la communauté des constructeurs. À l'avenir, toutes les entreprises de matériel devront être une cause et une source de profit, Raspberry Pi et Arduino sont d'excellents exemples.

[MagPi] Qu'est-ce qui vous a intéressé dans le

Raspberry Pi? En regardant www.adafruit.com je peux voir que la rubrique du Raspberry Pi est l'une des plus grandes.

[Limor] Chez Adafruit nous avons un but important et une mission ; pour enseigner la programmation et la construction aux enfants... et c'est maintenant enseigner à tous, et non seulement aux enfants. Nous pensons que tout le monde devrait être en mesure d'utiliser un ordinateur pédagogique à faible coût pour apprendre l'électronique et bien sûr apprendre un langage informatique. Nous avons galéré pour savoir de quelle façon nous serions en mesure de commencer cette entreprise, et c'est alors que le Raspberry Pi a été annoncé. Il est devenu si populaire si rapidement qu'il nous a vraiment amenés à répondre à la demande. Je teste personnellement tout ce que nous concevons ou distribuons dans la boutique. Pour le Raspberry Pi, nous savions que ce serait important d'avoir les meilleures ressources éducatives en plus du meilleur support. Six mois plus tard, la rubrique Pi est l'une de nos plus importantes, et les tutoriels



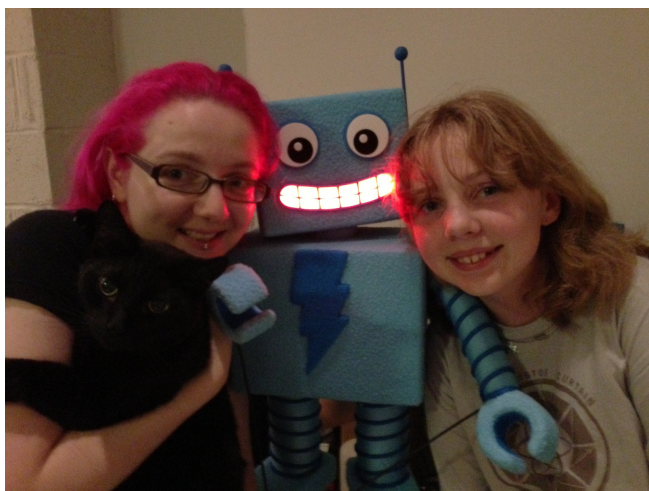
Le boîtier Pibow en exclusivité pour Adafruit

Pi sont les plus consultés sur le système d'apprentissage Adafruit <http://learn.adafruit.com/category/raspberry-pi>.

Mais attendez, il y a plus! Nous savions également que nous aurions besoin d'une façon d'enseigner basée sur le Web, alors nous avons investi beaucoup de temps et de ressources à construire notre propre environnement de développement intégré (IDE), incluant débogage et visualisation. Le WebIDE du Raspberry Pi est de loin le moyen le plus simple d'exécuter du code sur votre Raspberry Pi. Il suffit de connecter votre Pi à votre réseau local. Connectez-vous ensuite au WebIDE dans votre navigateur Web pour modifier Python, Ruby, JavaScript ou autre, et les envoyer facilement sur votre Pi. En plus, vos versions de code seront gérées dans un dépôt git local, et chargées sur Bitbucket afin que vous puissiez y accéder de n'importe où, et

n'importe quand. Regardez la vidéo sur <http://www.youtube.com/watch?v=8NoiBBgaKCI> Enfin, il y a notre ligne de produits Pi. Nous prenons le meilleur du design Adafruit et l'utilisons pour aider les gens à tirer le meilleur parti de leur Pi. Tout est entièrement documenté et supporté pour les constructeurs de tous les âges ... et encore, nous nous échauffons comme on dit!

[MagPi] Les femmes ont toujours été peu représentées dans le secteur de la technologie et malheureusement, cela peut durer tant que les préjugés sociaux et l'enseignement continuent à influencer les jeunes filles. Sur les quinze finalistes de «Entrepreneur 2012», vous étiez la seule femme. Mais l'Arduino a permis l'accès à l'électronique et à l'interfaçage aux artistes et aux créateurs, alors que le Raspberry Pi va permettre aux pédagogues de faire participer les garçons et les filles de façon amusante et intéressante. Mes deux jeunes filles ont chacune leur propre Raspberry Pi. Elles créent leurs propres programmes en utilisant Scratch ou recopient les programmes Scratch que nous publions régulièrement dans MagPi, puis les modifient. Vous êtes un excellent modèle pour les jeunes femmes d'aujourd'hui et j'ai récemment découvert une "super géniale Sylvia", âgée de 11 ans (www.sylviaishow.com), mais qu'est ce qui peut



Limor, Mosfet, AdaBot et Sylvia

encore être fait pour encourager les filles à étudier des sujets techniques?

[Limor] Une de mes citations favorites est: "Nous sommes ce que nous célébrons" de Dean Kamen. Il y a beaucoup de femmes en technologie, mais elles ne sont pas autant célébrées ou pas de la même façon que leurs homologues masculins. Les journalistes et les organisateurs de conférences peuvent mettre un coup de projecteur mérité sur de nombreuses femmes étonnantes; elles existent. Si une jeune fille qui grandit ne voit pas de femmes dans un domaine

particulier, ou n'est pas en mesure de dire "je veux faire ça, je veux être comme elle", alors nous ne verrons jamais plus de femmes dans ce domaine. Il appartient à chacun de nous de trouver des gens de la communauté et de les promouvoir. Il y a eu beaucoup de progrès, et il y a encore beaucoup de travail à faire. Avec un Pi à 35 \$ sur le marché, chacun de nous peut faire don de l'ingénierie à un jeune enfant. Vous ne savez jamais ce qui va amener quelqu'un sur la voie de l'ingénierie, mais je pense que nous avons plus de possibilités, en fait je pense que nous en avons un million de plus que ce que nous avons fait l'an dernier.

[MagPi] Nos jeunes lecteurs pourraient être intéressés par Mosfet. Que pouvez-vous nous dire sur Mosfet? Peut-être que Mosfet et Mooncake de Liz devrait s'appeler sur Skype?

[Limor] Mosfet est mon chat noir qui a environ 10 ans. Il a grandi chez Adafruit et il est dans notre émission hebdomadaire d'ingénierie en direct "Demandez à un Ingénieur" sur <http://www.adafruit.com/ask>. Il y a quelque chose de spécial au sujet des chats et des ingénieurs, nous ne savons pas ce que c'est exactement, mais les meilleurs ingénieurs que nous connaissons ont des chats - si bien que nous avons commencé "Les chats de l'ingénierie" sur <http://www.adafruit.com/catsofengineering>. En janvier, MAKE Magazine a tenu ses deuxièmes rencontres annuelles mondiales de constructeurs sur Google+ Hangouts. Le chat d'Eben et Liz a pu voir Mosfet pendant que nous parlions de la vente d'un million de Pi et plus. Je ne suis pas sûre de ce qui va se passer avec Internet dans 10, 50 et 100 ans, mais pour autant que je peux le dire, il a été conçu pour que les chats communiquent entre eux :)

[MagPi] Le site Adafruit présente une très large gamme de produits fascinants. Je suis particulièrement intéressé par l'afficheur négatif RGB 2x16 pour le Raspberry Pi et le MintyBoost pourrait facilement alimenter un modèle A de Raspberry Pi. Et il y a aussi la gamme complète d'appareils électroniques portables FLORA. Quels sont actuellement les produits qui suscitent le plus grand intérêt auprès de vos clients et quels sont ceux qui vous excitent actuellement?

[Limor] Les deux grandes lignes de produits sont tout ce qui concerne le Raspberry Pi et notre nouvelle plateforme portable FLORA. Les gens veulent apprendre à programmer, à construire des projets intéressants et aussi accéder à l'ère de l'électronique portable. En ce moment les produits qui sont vraiment excitants pour moi sont ceux qui inspirent la communauté pour créer des

projets cool et les partager. J'aime les radios Wi-Fi basées sur le Pi et j'ai vraiment plaisir à voir arriver les projets qui utilisent FLORA d'une manière que nous n'avons pas prévue. Je travaille sur un contrôleur de jeu Pi, un écran tactile et pour FLORA, sur une version encore plus petite pour des applications portables super-miniatures.

[MagPi] Je suis constamment étonné par les projets que les gens créent avec leur Raspberry Pi. Je les découvre sur www.raspberrypi.org ou je les vois sur vos vidéos "Show & Tell" ou nous les publions dans MagPi. Quels projets avez-vous vu construire avec des produits que vous vendez et qui ont provoqué un "Wow"?

[Limor] Un de mes projets préférés est en cours. Un fabricant très prolifique dans notre communauté, nommé Kris, construit un robot déneigeur à base de Raspberry Pi. Il fabrique toutes les pièces du robot, les servos de contrôle, la vision et plus, tout ça avec un Pi. C'est un robot fonctionnel qui repose sur l'open-source et nous apprenons et partageons tous ensemble. Penser que quelqu'un pourrait faire un projet aussi complexe chez lui et pendant son temps libre, et à un coût raisonnable c'est juste "Wow!". [Ed: J'ai vu Kris dans l'épisode de "Show & Tell" du 19 janvier. Son travail est impressionnant.]

[MagPi] Vous avez parlé plus tôt du Système d'Apprentissage Adafruit, qui contient plus de 100 didacticiels et des vidéos, y compris une série pour le Raspberry Pi. Personnellement, je n'avais aucune idée de l'ampleur de cette ressource pédagogique et j'y retournerai pour en savoir plus et aussi télécharger le WebIDE Raspberry Pi. Quels sont vos projets pour développer encore le Système d'Apprentissage Adafruit?

[Limor] Merci, l'équipe a fait un travail extraordinaire pour créer le meilleur système d'apprentissage en ligne. Justin, Tyler, Daigo et mon équipe ont vraiment porté les tutoriels à un niveau supérieur. Le Système d'Apprentissage Adafruit n'est qu'une partie de la tâche gigantesque d'éducation pour tous. Nous commençons avec la programmation et

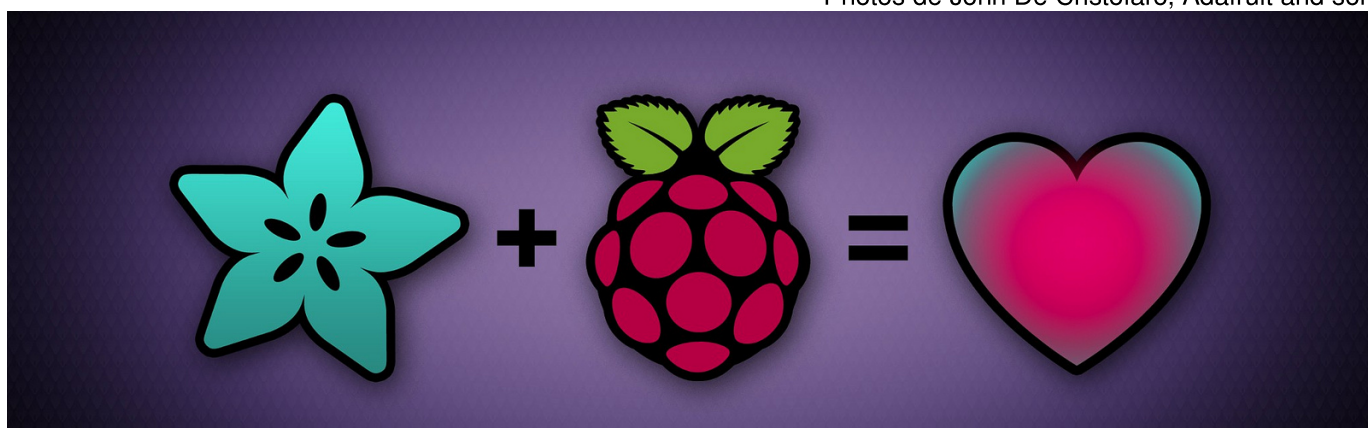
l'ingénierie, mais nous n'allons pas nous arrêter là. Nous pensons qu'il n'y a pas de limite à l'accroissement de notre potentiel à travers le partage. Nous allons ajouter notre système de marquage, plus de possibilités pour l'utilisateur et plus de facilité à utiliser et partager les connaissances. Nous avons des fichiers PDF que vous pouvez imprimer et de partager, le système fonctionne très bien pour les composants et nous allons ajouter plus de vidéo et plus d'interactivité (enregistrement de capteur et plus) très bientôt.

[MagPi] Je lis que 2013 est l'année de l'entrepreneur. Peut-être cela est-il influencé par l'économie mondiale, mais nous vivons certainement dans une époque intéressante. Les produits tels que le Raspberry Pi et Arduino initient une nouvelle génération et permettent de nouvelles fonctionnalités ainsi que les imprimantes 3D, espaces hacker et foires aux constructeurs rendent tous la technologie plus accessible à tous dans un esprit communautaire, d'éducation et de partage. Je ne vous demanderai pas quelle est la prochaine grande chose (!) mais ayant été l'un des moteurs majeurs de cette communauté et l'ayant vu évoluer au cours des 7 dernières années quelles tendances voyez-vous apparaître si vous vous projetez dans le futur ?

[Limor] Vous pouvez faire qu'une quelconque prédiction se réalise si vous êtes prêt à faire le travail difficile, ainsi chez Adafruit nous essayons de prédire l'avenir en le construisant. Dans les 7 prochaines années plus de personnes seront capables de faire tout ce qu'ils peuvent imaginer, alors qu'est-ce que ça veut dire ? Je pense que nous avons de gros problèmes à résoudre. Si un ordinateur à 35\$ peut aider à enseigner, pourrait-il aussi diagnostiquer et aider à guérir ? Pourrait-il aider la démocratie et la liberté dans le monde ? Est-ce que ce type de technologie et l'accès pour tous pourrait nous faire progresser tous ensemble ? Je pense que oui, et c'est à ça que je vais passer mes jours et mes nuits.

Article de Ian Mc Alpine

Photos de John De Cristofaro, Adafruit and son





WebIOPi - framework REST pour le Raspberry Pi

Didacticiel : Robot-caméra télécommandé

WebIOPi est un framework REST qui vous permet de contrôler le GPIO du Raspberry Pi à partir d'un navigateur. Il est écrit en Javascript pour le client et en Python pour le serveur.

Vous pouvez entièrement personnaliser et créer facilement votre application web. Vous pouvez même utiliser toute la puissance de WebIOPi directement dans votre propre script Python et enregistrer les fonctions pour pouvoir les appeler depuis l'application web. WebIOPi inclut aussi d'autres fonctionnalités comme le PWM logiciel pour tout le GPIO.

Installation

L'installation sur le Raspberry est vraiment facile, puisqu'elle nécessite seulement Python. Sur Raspbian Wheezy, vous pouvez utiliser la Pistore pour télécharger et installer WebIOPi. Vous pouvez également l'installer à partir d'un terminal ou d'une connexion SSH. Vérifiez la dernière version sur la page du projet, puis entrez :

```
$ wget
http://webiopi.googlecode.com/files/WebI
OPi-0.5.3.tar.gz
$ tar xvzf WebIOPi-0.5.3.tar.gz
$ cd WebIOPi-0.5.3
$ sudo ./setup.sh
```

Vous devriez voir se dérouler la compilation et l'installation, l'affichage des instructions d'utilisation confirmant le succès de l'opération :

```
WebIOPi installé avec succès
* Pour démarrer WebIOPi avec python:
sudo python -m webiopi
* Pour démarrer WebIOPi avec python3:
sudo python3 -m webiopi
* Pour démarrer WebIOPi au boot:
sudo update-rc.d webiopi defaults
* Pour démarrer le service WebIOPi:
sudo /etc/init.d/webiopi start
* Regardez dans
```

```
/home/pi/webiopi/examples
pour des exemples d'utilisation de la
bibliothèque Python.
```

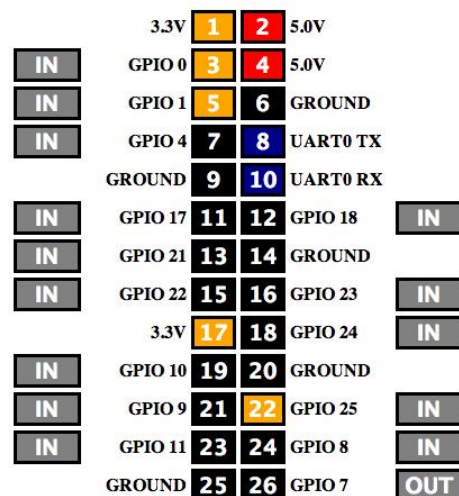
Vous aurez une ligne pour chaque version installée de Python que vous pouvez utiliser pour lancer WebIOPi.

Il est temps de démarrer WebIOPi, par exemple avec Python 2.X:

```
$ sudo python -m webiopi
WebIOPi/Python2/0.5.3 a démarré sur
http://[IP]:8000/webiopi/
```

Première utilisation

Ouvrez un navigateur sur une machine de votre réseau, tapez l'URL `http://[IP]:8000/webiopi/`, ou utilisez localhost sur votre Pi si vous avez un clavier et un écran connectés dessus. Vous serez alors invité à vous connecter, **l'utilisateur par défaut est webiopi et le mot de passe est raspberry**. Vous devriez voir l'écran par défaut de l'application :

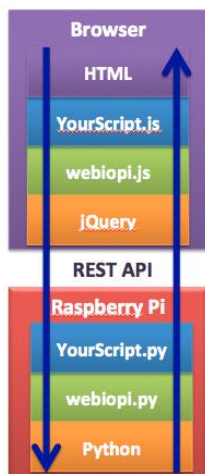


Avec cet écran par défaut, vous pouvez modifier les fonctions du GPIO d'entrée à sortie, et changer l'état des broches. Il suffit de cliquer sur les boutons IN/OUT, et sur chaque broche pour changer son état

lorsqu'elle est réglée en sortie.

Toutes les broches du GPIO peuvent être directement utilisées avec l'API REST. Par exemple, pour configurer GPIO 23 en sortie, il suffit de faire une requête HTTP POST sur /GPIO/23/function/out puis de sortir un 1, faites un POST sur /GPIO/23/value/1. Pour relire les états, faites HTTP GET sur /GPIO/23/function et /GPIO/23/value.

La bibliothèque Javascript incluse permet de modifier les GPIO sans se soucier des appels REST.

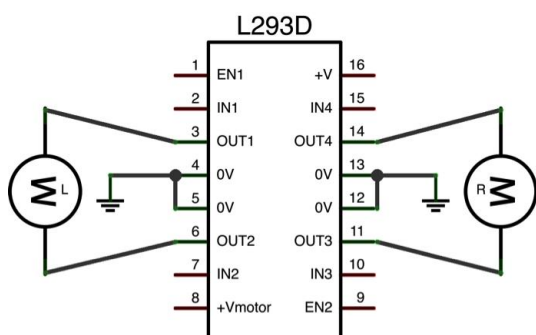


Caméra robot

Les parties suivantes de cet article présentent une webcam robotisée que vous pouvez contrôler à partir de n'importe quel navigateur. Vous aurez besoin de:

- Un châssis
- Un Raspberry Pi avec WebIOPi
- Une Webcam USB opérationnelle
- Un adaptateur USB-Wifi opérationnel
- Un pont en H L293
- 2 ensembles (moteur + réducteurs + roues)
- Des piles et un régulateur de puissance
- Diverses pièces détachées électroniques

Du point de vue électronique, le L293 contient un circuit électronique similaire au pont en H du Skutter décrit dans le MagPi de décembre. Le L293 ajoute une entrée de validation qui peut être utilisée avec un signal PWM pour limiter la vitesse. Il possède



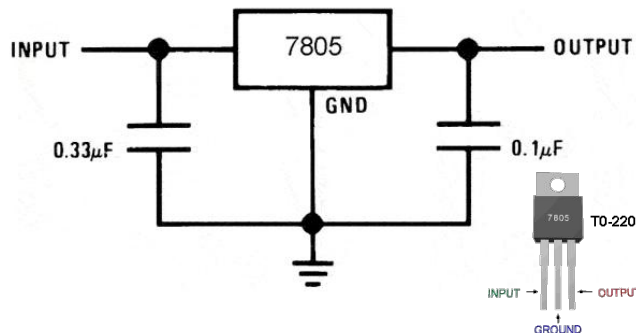
également deux entrées d'alimentation, une pour la logique (+V=5V), et une pour le moteur (+Vmotor<36V).

La rotation du moteur est commandée par IN* et EN* :

| IN1 | IN2 | EN1 | MOTOR L |
|------|------|------|----------|
| LOW | LOW | X | STOP |
| X | X | LOW | STOP |
| HIGH | LOW | HIGH | FORWARD |
| LOW | HIGH | HIGH | BACKWARD |

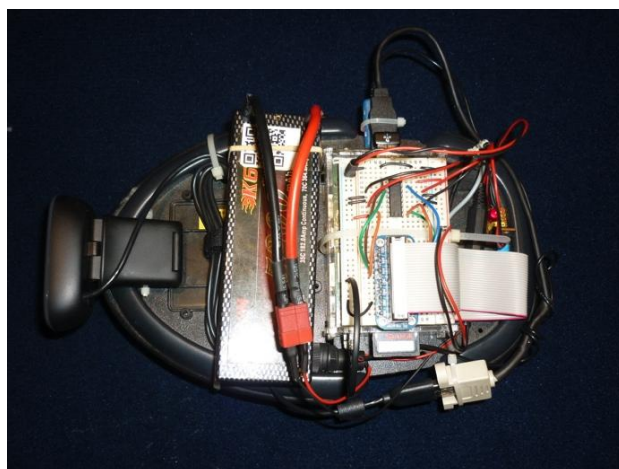
| IN3 | IN4 | EN2 | MOTOR R |
|------|------|------|----------|
| LOW | LOW | X | STOP |
| X | X | LOW | STOP |
| HIGH | LOW | HIGH | FORWARD |
| LOW | HIGH | HIGH | BACKWARD |

Nous pouvons relier +V au +5v du Pi et IN*/EN* aux broches du GPIO. +Vmotor sera relié aux piles ou à un régulateur dédié. Vous aurez besoin d'au moins un régulateur 5V pour alimenter le Pi avec une pile, via la prise micro USB. Vous pouvez utiliser un 7805 avec deux condensateurs comme ceci :



Avec une tension d'entrée supérieure à 7V, vous obtiendrez une tension de sortie de 5V régulée pour le Pi.

Pour contrôler le pont en H avec WebIOPi et créer une interface, nous devons écrire quelques lignes en Python côté serveur, et un peu de Javascript pour le client.



Écriture du script Python

Commencez par créer un nouveau fichier Python avec votre éditeur de texte favori. Vous aurez à importer webiopi puis à

instancier un serveur. Un paramètre est obligatoire: le port auquel le serveur va se connecter. Vous pouvez également modifier les login et mot de passe par défaut. Le serveur va démarrer et fonctionner dans son propre fil d'exécution (thread) jusqu'à la fin du script. Il faut ajouter une boucle pour maintenir le serveur en fonctionnement. Nous utilisons la fonction `webiopi.runLoop()` pour cela. Elle met en sommeil le fil d'exécution principal jusqu'à un CTRL-C. Nous pouvons également passer une fonction à la boucle.

```
import webiopi
# Instancier le serveur WebIOPi
# Il démarre immédiatement
server=webiopi.Server(
    port = 8000,
    login="cambot",
    password="Cambot")

# Exécuter la boucle par défaut
webiopi.runLoop()

# Arrêter proprement le serveur
serveur.stop ()
```

Le script précédent démarre simplement le serveur WebIOPi. Nous pouvons utiliser l'application Web par défaut pour interagir avec le GPIO, l'API REST ou la bibliothèque Javascript. Nous pourrions contrôler directement les entrées du pont en H en Javascript, mais nous allons ajouter les macros REST `go_forward` et `stop` pour diminuer la latence.

Pour continuer, nous avons besoin d'une bibliothèque GPIO. Nous pouvons utiliser `RPi.GPIO` ou la bibliothèque intégrée `GPIO`, qui est un dérivé de `RPi.GPIO`. Une bibliothèque intégrée permet d'éliminer de nombreuses vérifications pour l'accès depuis le serveur et offre plus de fonctionnalités.

Juste après la section d'importation:

```
# Librairie GPIO intégrée
GPIO=webiopi.GPIO
```

Nous ajoutons des variables pour faciliter le contrôle du pont en H :

```
# GPIOs moteur gauche
L1=9 # L293 IN1 sur GPIO 9
L2=10 # L293 IN2 sur GPIO 10
LS=11 # L293 EN1 sur GPIO 11

# GPIOs moteur droit
R1=23 # L293 IN3 sur GPIO 23
R2=24 # L293 IN4 sur GPIO 24
RS=25 # L293 EN2 sur GPIO 25
```

Avant l'appel du serveur, nous écrivons les

fonctions pour les moteurs gauche et droit, puis nous les insérons dans les macros `go_forward` et `stop` :

```
# Fonctions du moteur gauche
def left_stop():
    GPIO.output(L1, GPIO.LOW)
    GPIO.output(L2, GPIO.LOW)

def left_forward():
    GPIO.output(L1, GPIO.HIGH)
    GPIO.output(L2, GPIO.LOW)

# Fonctions du moteur droit
def right_stop():
    GPIO.output(R1, GPIO.LOW)
    GPIO.output(R2, GPIO.LOW)

def right_forward():
    GPIO.output(R1, GPIO.HIGH)
    GPIO.output(R2, GPIO.LOW)

# Réglage de la vitesse du moteur
def set_speed(speed):
    GPIO.pulseRatio(LS, speed)
    GPIO.pulseRatio(RS, speed)

# Fonctions de mouvement
def go_forward():
    left_forward()
    right_forward()

def stop():
    left_stop()
    right_stop()
```

Ensuite et toujours avant l'appel au serveur, nous initialisons le GPIO :

```
# Initialisation des broches GPIOs
GPIO.setFunction(LS, GPIO.PWM)
GPIO.setFunction(L1, GPIO.OUT)
GPIO.setFunction(L2, GPIO.OUT)

GPIO.setFunction(RS, GPIO.PWM)
GPIO.setFunction(R1, GPIO.OUT)
GPIO.setFunction(R2, GPIO.OUT)

set_speed(0.5)
stop()
```

Finalement, nous devons enregistrer les macros sur le serveur pour les ajouter à l'API REST. Ceci permettra de les appeler depuis l'appli web :

```
server.addMacro(go_forward)
server.addMacro(stop)
```

Article d'Eric PTAK

Nous verrons la suite dans le prochain numéro, mais si vous voulez prendre une longueur d'avance, visitez

<http://files.trouch.com/webiopi/cambot.zip>

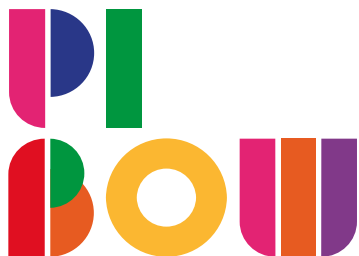
Pirate Store

from PIMORONI



<http://shop.pimoroni.com/>

Pibow * Electronics * GPIO * Raspberry Pi®



The neat little layer case
for your Raspberry Pi®

Ninja



Crystal



Toxic



Adafruit



Rainbow



Available From:
<http://pibow.com/>



SAUVEZ VOTRE RASPBERRY PI



Apprenez à sauvegarder votre carte SD avec ou sans compression plus découpage pour gravure sur DVD.

Dans cet article en 2 parties nous verrons comment facilement sauvegarder votre carte SD sur un autre ordinateur, une clé USB ou des CD/DVD puis la restaurer. Tout cela est simple sur un ordinateur Linux. Les utilisateurs Windows souhaitant faire des sauvegardes sur CD ou DVD feraient mieux d'utiliser une distribution Live comme l'excellent Linux Mint.

Pourquoi faire une sauvegarde ?

Vous devriez toujours avoir une ou plusieurs sauvegardes de votre carte SD. Prenez l'habitude d'en faire une chaque jour avant qu'il ne soit trop tard... surtout si vous éditez des fichiers ou ajoutez souvent des logiciels.

Les cartes SD sont beaucoup plus robustes que des disques durs mais des pertes totales de données peuvent arriver. J'avais deux cartes devenues corrompues qui ont bloqué mon Pi. Le seul recours a été de le débrancher et dans les deux cas je me suis retrouvé avec une partition "/" illisible, donc sans système d'exploitation.

Par chance j'avais une sauvegarde. C'était une copie d'une carte 4Go et celle endommagée faisait 8Go. Heureusement une image 4Go peut être restaurée sur une carte de plus grande capacité comme nous le verrons plus loin dans cet article.

Utilisateurs Windows

Ces instructions ne concernent que Linux. Les utilisateurs Windows peuvent créer une image de leur carte avec le programme Win32DiskImager qu'ils ont utilisé pour installer Raspbian. Insérez la carte dans le lecteur de cartes de votre ordinateur et lancez Win32DiskImager. Entrez un nom pour l'image et cliquez sur Read. Une fois la sauvegarde achevée, la taille de l'image

devrait être quasiment identique à celle de la carte SD.

Un mot d'avertissement

Comme Windows est le système le plus répandu, beaucoup de vendeurs fournissent des disques durs USB formatés en FAT32. Ce système de fichiers ne gère pas les fichiers de plus de 4 Go. Il faudra donc découper la sauvegarde en blocs de 4 Go si votre carte est plus grande.

Vous pouvez reformater votre support en NTFS, mais si vous ne l'utiliserez jamais sous Windows, vous pouvez choisir n'importe quel système de fichiers Linux comme EXT4.

Que ce soit sous Windows ou Linux, s'il s'agit d'une version 32 bits vous rencontrerez la même contrainte sur la taille de fichiers. Le maximum pour un fichier unique sur un système 32 bits est de 4 Go. Si votre copie de sauvegarde est plus grande, vous devrez la scinder en morceaux de 4 Go. La compression et le découpage sont abordés en détails plus loin.

Obtenir l'accès administrateur

Nous supposons dans le reste de l'article que vous avez un accès administrateur à l'ordinateur sur lequel vous ferez la sauvegarde. Pour cela, vous pouvez soit préfixer chaque commande avec sudo, soit lancer la commande suivante pour démarrer un terminal administrateur :

```
$ sudo sh
```

Identifiez votre carte SD

Insérez votre carte dans l'ordinateur Linux, pas dans le Raspberry Pi. S'il a un lecteur intégré, elle sera sans doute sur /dev/mmcblk0. Avec un adaptateur USB, ce

Disk /dev/mmcbk0: 7948 MB, 7948206080 bytes
4 heads, 16 sectors/track, 242560 cylinders, total 15523840 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x000dbfc6

| Device Boot | Start | End | Blocks | Id System |
|---------------|--------|----------|---------|-------------------|
| /dev/mmcbk0p1 | 8192 | 122879 | 57344 | c W95 FAT32 (LBA) |
| /dev/mmcbk0p2 | 122880 | 15523839 | 7700480 | 83 Linux |

sera plutôt /dev/sdx où "x" est le n° représentant le périphérique le plus élevé de ce type.

Lancez la commande fdisk -l. La fin de l'affichage est reproduite au haut de cette page. Je peux reconnaître la carte grâce à sa taille, 8 Go (ou 7948 Mo ci-dessus), car c'est le seul périphérique à avoir cette taille. Sont aussi indiquées les deux partitions, une petite en FAT32 (LBA) W95 et une Linux, ce qui confirme que c'est bien ma carte SD.

Plus haut, les suffixes "p1" et "p2" dans la colonne Device Boot désignent les partitions. La carte elle-même est nommée /dev/mmcbk0 et c'est le nom dont nous avons besoin, pas celui des partitions.

Faire une sauvegarde

Allez dans votre répertoire de sauvegarde, où vous voulez placer la copie de votre carte, et tapez la commande suivante sur une seule ligne :

```
$ dd if=/dev/mmcbk0  
of=Rpi_8gb_backup.img bs=2M
```

Le résultat ressemblera à ceci :

```
3790+0 records in  
3790+0 records out  
7948206080 bytes (7.9 GB) copied,  
1369.42 s, 5.8 MB/s
```

Voilà. La carte SD 8 Go a été (lentement) copiée en entier sur mon ordinateur. Comment savoir si ça a marché ? Tapez ls -l -h et vous verrez le nom du nouveau fichier et sa taille.

Compression de la sauvegarde

Une fois créée et vérifiée, l'image peut être compressée pour gagner de la place. Il suffit d'un simple appel à la commande gzip :

```
$ gzip -9 Rpi_8gb_backup.img
```

Celle-ci réalisera une compression maximale. Le CPU sera très sollicité mais le fichier généré sera le plus compact possible. Vous pouvez trouver un compromis entre CPU et taille de fichier en changeant l'option "-9" par un nombre inférieur. Une compression rapide sera obtenue avec "-1" au détriment d'une taille de fichier plus importante.

Utiliser gzip de cette façon oblige à avoir assez d'espace pour stocker à la fois l'image complète et l'image compressée pendant l'exécution de la commande. Une fois la compression terminée, le fichier d'origine sera supprimé pour ne laisser que la version compressée avec le suffixe ".gz". Dans cet exemple, mon image compressée s'appelle Rpi_8gb_backup.img.gz.

Compression à la volée

Si vous ne voulez faire que la sauvegarde ou si vous n'avez pas assez d'espace libre pour les deux fichiers, alors vous pourriez être intéressés par la compression à la volée.

Sauf indication contraire, dd envoie par défaut sa sortie sur la console. Vous pouvez utiliser cette caractéristique pour faire un tube vers gzip et créer un fichier compressé en une seule étape. Vous n'aurez pas besoin d'autant d'espace disque car l'image de taille complète ne sera jamais créée, il n'y aura que la plus petite compressée. La commande est :

```
$ dd if=/dev/mmcbk0 bs=2M | \  
gzip -9 - > Rpi_8gb_backup.img.gz
```

Le "|" final doit être en fin de ligne, juste avant l'appui sur Entrée. Il indique au shell que la commande n'est pas finie et qu'elle se poursuit ligne suivante.

Suite sur la page suivante...

Puisque vous redirigez la sortie de gzip vers un fichier, c'est à vous d'ajouter l'extension ".gz". Il faut aussi inclure un tiret après l'option "-9" dans la commande gzip afin qu'elle envoie son résultat sur la console, qui, dans ce cas, est redirigée vers Rpi_8gb_backup.img.gz.

Découper la sauvegarde

Comme indiqué auparavant, certains systèmes de fichiers imposent des limites à la taille maximale des fichiers. Par exemple, 4 Go en FAT32, et les systèmes 32 bits sont aussi limités à 4 Go pour chaque fichier. Si vous prévoyez de créer ou copier vos sauvegardes sur ce type de systèmes ou de périphériques, vous devrez alors découper vos fichiers en blocs de taille appropriée.

Une fois encore, un tube va venir à notre secours. L'outil split est conçu pour diviser un gros fichier en plusieurs fichiers plus petits, chacun de taille donnée. Pour créer une sauvegarde compressée d'une carte SD, prête à être gravée sur un ou plusieurs CD ou DVD, la commande suivante va faire la sauvegarde, compresser l'image à la volée puis découper cette dernière en blocs de 2 Go prêts pour la gravure sur DVD.

```
$ dd if=/dev/mmcblk0 bs=2M | \
gzip -9 - | \
split --bytes=2G - \
Rpi_8gb_backup.img.gz.part_
```

Rappelez-vous d'appuyer sur Entrée juste après avoir tapé le "\" sur chaque ligne.

La commande split, comme gzip, doit avoir un tiret pour nom de fichier en entrée. Ci-dessus, il n'y a pas de fichier car nous lisons depuis un tube, le tiret dit à split de lire depuis la console qui est dans notre cas la sortie redirigée depuis gzip.

Le dernier paramètre définit la partie principale des noms de fichiers créés. Chacun sera appelé "Rpi_8gb_backup.img.gz.part_xx". La partie "xx" sera "aa", "ab", "ac" etc.

Il est possible que la compression de l'image SD permette au fichier d'être suffisamment petit pour ne pas dépasser les limites du système de fichiers du périphérique choisi. Dans ce cas, vous pouvez redonner au fichier

unique son nom d'origine.

Restaurer les fichiers de sauvegarde

Avoir une sauvegarde compressée ou scindée en plusieurs morceaux implique davantage de travail pour la restauration. Une fois encore le processus est relativement simple et fait appel à des tubes pour constituer une chaîne d'outils dont le résultat final sera envoyé à la commande dd. Cette dernière restaurera les données sur la carte SD.

Le cas le plus simple est celui d'une compression sans découpage. Pour restaurer la carte SD sans avoir à recréer l'image décompressée, tapez la commande suivante :

```
$ gunzip Rpi_8gb_backup.img.gz -c | \
dd of=/dev/mmcblk0 bs=2M
```

Le "-c" dans la commande gunzip est nécessaire pour que les données décompressées soient envoyées sur la console. La commande dd lit son entrée depuis celle-ci, sauf indication contraire, si bien que la sortie de gunzip va directement vers dd puis sur la carte SD.

Si la sauvegarde est découpée en morceaux, il faut les recoller dans le bon ordre et les envoyer via un tube vers la commande chaînée, comme ceci :

```
$ cat Rpi_8gb_backup.img.gz.part_* | \
gunzip -c | \
dd of=/dev/mmcblk0 bs=2M
```

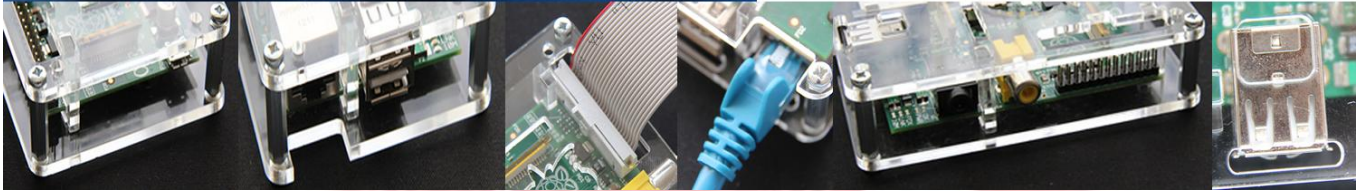
La liste de fichiers à réunir doit être spécifiée dans l'ordre correct. Cependant, l'utilisation d'un joker pour la commande cat fait qu'ils sont lus dans l'ordre alphabétique, ce qui est exactement ce que nous voulons.

À suivre dans la 2^e partie

Dans la 2^{de} partie, nous verrons comment vérifier que la sauvegarde a réussi et comment utiliser le fichier de sauvegarde comme pseudo-disque dur. Je vous montrerai aussi comment modifier des fichiers à l'intérieur - sans avoir besoin de restaurer l'image sur une carte SD.

Article de Norman Dunbar

CONCOURS DE FÉVRIER



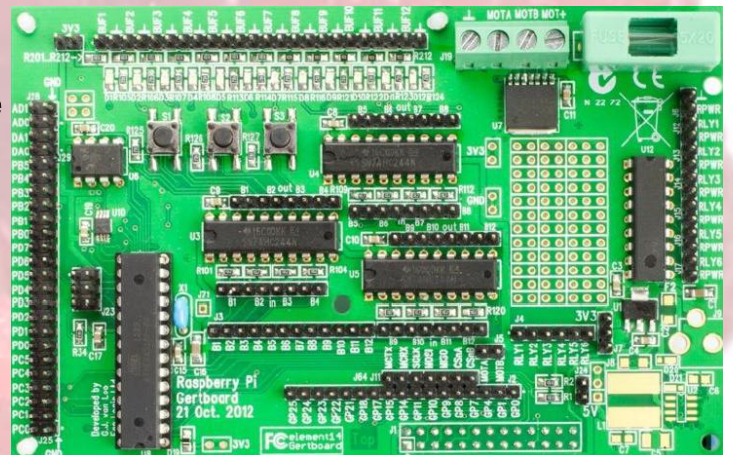
Une fois de plus The MagPi et PC Supplies Limited (PCSL) sont fiers d'annoncer une autre chance de gagner de fantastiques accessoires pour le Raspberry Pi !

Ce mois-ci, il y a trois prix à gagner !

Le gagnant du premier prix recevra une des nouvelles Gerboards pré-assemblés ! Les gagnants des deuxième et troisième prix recevront un boîtier PCSL couleur framboise.

Pour participer au concours du mois visitez :
<http://www.pcslshop.com/info/magpi>

La date de clôture est le 20 Février 2013. Les gagnants seront informés dans le magazine du prochain mois et par courriel. Bonne chance !



Pour retrouver la large gamme des accessoires de PCSL pour le Raspberry Pi, visitez <http://www.pcslshop.com>

Les gagnants de décembre !

Il y a eu près de 540 participants ! Le choix a été difficile avec autant de réponses intéressantes. La gagnante du Raspberry Pi 512 Mo Modèle B avec alimentation 5V/1A et boîtier PCSL est **Fiona Parker (Ilkley, GB)**. Les gagnants des 2^e et 3^e prix, un boîtier pour Raspberry Pi de PCSL sont **Damien Plunkett (Flagstaff, USA)**, et **Nico van der Dussen (Pretoria, Afrique du Sud)**. Bravo. Nous vous indiquerons prochainement par courriel comment récupérer vos lots !



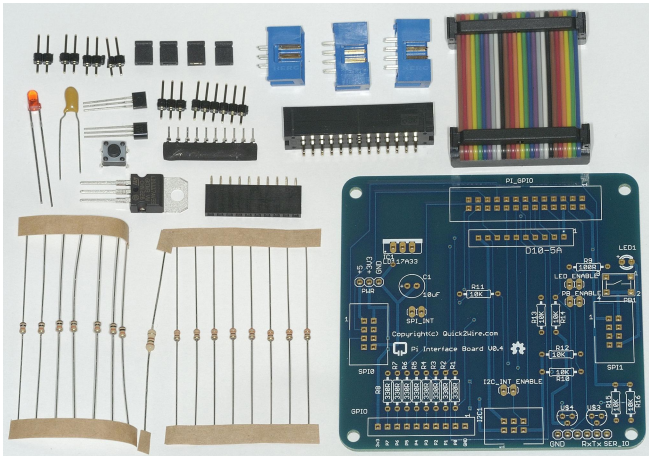


Pi Interface Board Review Quick2Wire

La carte d'interface Quick2Wire est une nouvelle carte d'interface pour le Raspberry Pi qui sera mise en vente en février 2013.

J'ai réussi à mettre la main sur l'un des kits bêta pour le tester et j'ai également eu une conversation avec Romilly Cocking, directeur de Quick2Wire.

Alors, Qu'est ce qu'il y a dans le kit?



- 1 x circuit imprimé (PCB)
- 1 x câble plat multicolore
- 4 x cavaliers
- 11 x connecteurs assortis
- 2 x transistors (FET)
- 1 x condensateur au tantale
- 1 x diode électroluminescente (DEL)
- 1 x bouton-poussoir
- 1 x régulateur de tension (3V3)
- 1 x pont de diodes
- 16 x résistances

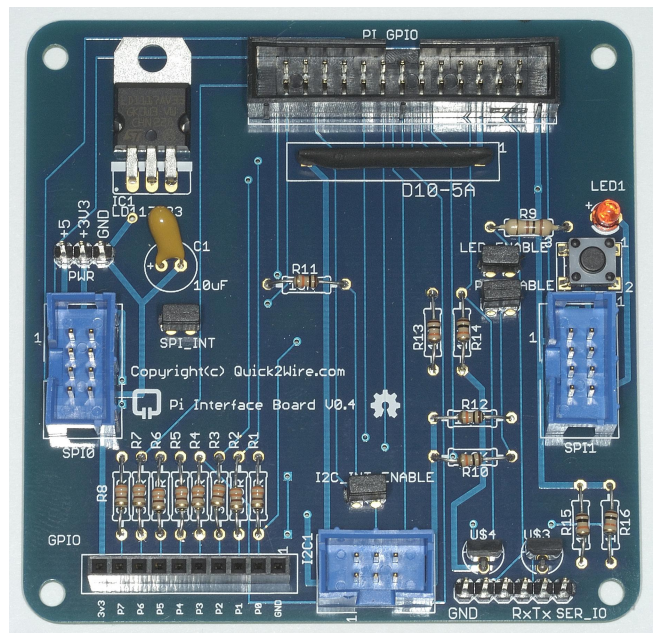
Construction de l'interface

La construction a été un plaisir. J'avais montré à mon fils comment souder la veille, et laissé la soudure au plomb et le fer à souder sortis. J'ai décidé de les utiliser pour ce montage.

Les instructions sont claires et la construction est assez facile. Vous pourriez probablement construire sans les instructions car l'excellente sérigraphie, imprimée sur le circuit, vous montre où vont les composants. Les seuls pièges éventuels sont l'orientation

du pont de diodes et celle du condensateur au tantale. Ils doivent tous deux être montés dans le bon sens. Les instructions expliquent comment y arriver.

Il m'a fallu environ 20 minutes pour monter toutes les résistances (j'étais pointilleux pour les aligner). En tout, il a fallu 53 minutes du début à la fin et ce fut une expérience agréable. Voilà à quoi ça ressemble une fois fini.



Installation du logiciel

L'étape suivante a été la mise à jour Raspbian et l'installation du logiciel pour tester l'interface. Les instructions étaient excellentes et l'installation logicielle a parfaitement fonctionné. Le plus long a été la mise à jour des paquets de Raspbian (en utilisant `sudo apt-get update` et `sudo apt-get upgrade`).

“La seule chose différente ici, par rapport aux autres pilotes GPIO basés sur Python, du moment que vous êtes dans le groupe "gpio", c'est qu'il n'y a pas besoin d'être

autorisé à exécuter sudo pour lancer les commandes. c'est génial ! "

Procédures d'essai

La section suivante décrit les procédures de test issues du manuel fourni. Les commandes de test sont données pour allumer/éteindre la DEL et lire l'état du bouton-poussoir (appuyé/relâché). Elles ont parfaitement fonctionné (bien que la DEL sur ma carte ne soit pas très brillante).

Ensuite, vous êtes invité à tester votre interface FTDI 5v (un type de câble que vous pouvez utiliser pour vous connecter directement, sans connexion réseau) si votre carte en est équipée. La mienne a fonctionné parfaitement. J'ai été en mesure de me connecter avec un terminal via le port série. Ceci a clôturé la procédure de test. Tout s'est bien passé.

Test de la bibliothèque Python Quick2Wire

Pour aller un peu plus loin que les instructions de test du manuel, j'ai téléchargé la bibliothèque Python pour Quick2Wire de GitHub <https://github.com/quick2wire> et mis en œuvre deux programmes de test que j'y ai trouvé - clignotement de la DEL et utilisation du bouton pour la contrôler. Ils n'ont pas encore été documentés.

J'ajouterai que vous pouvez utiliser cette bibliothèque de logiciels pour contrôler les ports GPIO, quel que soit le matériel dont vous disposez (en dehors du Pi lui-même). Donc, théoriquement, je pourrais écrire une troisième version en Python de la suite de tests pour Gertboard pour aller avec le WiringPi pour Python et les versions RPi.GPIO.

DEL RVB

Donc j'ai écrit un petit script pour faire clignoter une DEL tricolore (Rouge, Vert, Bleu - RVB) suivant différentes combinaisons de couleurs à l'aide des sorties P7, P6 et P5 (numérotation selon Quick2Wire). Cela a également fonctionné à la perfection. Voici un lien vers la vidéo d'une visite guidée de la carte Quick2Wire avec les programmes de tests en action.

<http://www.youtube.com/watch?v=JJuKtsdAerU>

Nous interrompons cet article...

Un autre domaine où la bibliothèque Python Quick2Wire se démarque (que je n'ai pas

exploré encore) le support d'epoll. Cela vous permet de contrôler les choses en utilisant les interruptions au lieu d'une scrutation continue. Cela signifie que votre programme peut répondre à un changement, au lieu d'utiliser la puissance du processeur à contrôler en permanence l'état des ports GPIO. Il s'agit d'un développement important.

Conclusion

L'interface Quick2Wire Pi est un très bel outil, avec beaucoup de promesses et ce qui semble une très bonne Interface de programmation (API) Python, en arrière plan. Ce sera moins cher que si vous vouliez le faire vous-même. Si j'étais pointilleux, je préférerais une DEL plus visible (plus lumineuse). Mais c'est vraiment la seule chose que je changerais, ce qui en dit long !

L'avenir ?

Il y a d'autres cartes interfaces Quick2Wire en projet, y compris un convertisseur analogique-numérique (CAN), des contrôleurs de moteur, des écrans ACL, des contrôleurs de servomoteurs etc. Ça devient un système modulaire, facile à construire et à utiliser. On parle même d'une interface pilotée par Scratch sur toute la ligne.

Matériel et logiciel sont tous totalement libres. L'accent est mis sur les bibliothèques de logiciels et le grand besoin de ressources pédagogiques libres. Il me semble vraiment que Quick2Wire est une entreprise à suivre. Leur site web est Quick2Wire.com.

Article d'Alex Eames

Alex Eames dirige le blog RasPi.tv, où il présente des choses éducatives, ludiques, innovantes ou tout simplement idiotes faites avec un Raspberry Pi. Il a également écrit l'adaptation en Python du logiciel de la Gertboard.

**Le SAVIEZ
VOUS ?**

L'équipe de Quick2Wire se compose de neuf personnes, couvrant une zone de 7 fuseaux horaires. Ils vivent à Chicago, Londres, Bristol et dans les Pyrénées. Ils n'ont jamais été tous réunis au même endroit !



RISC OS Pi

Installation de RISC OS sur le Raspberry Pi

Historique

Le système d'exploitation (OS) à jeu d'instructions réduit (RISC) pour les ordinateurs basés sur les processeurs et émulateurs "Acorn RISC Machine" (ARM) existe depuis 1987 (sous le nom originel d'Arthur), soit presque depuis aussi longtemps que le processeur ARM lui-même. Le premier ordinateur équipé d'un ARM 2 était l'Acorn Archimedes d'après le nom du fameux savant Grec antique. À sa sortie, il a représenté un formidable bond en avant par rapport à l'omniprésent BBC Micro, bâti autour d'un 6502 et vendu pour les usages personnels et l'éducation. À la suite de cela le plus puissant RISC PC est sorti en 1994, basé sur la puce StrongARM cadencée à 300Mhz, bien que la production se soit interrompue il y a une dizaine d'années.

En 1998 Acorn est démantelé, et Castle Technology rachète les droits de RISC OS à Pace Technology et lance l'loynix PC. Depuis 2006 RISC OS Ltd (ROOL) a pris en charge le développement de RISC OS en code source partagé (Shared Source Initiative = SSI) et quelques variantes existent maintenant qui s'exécutent sur l'émulateur du RISC PC sous Windows et Unix (RPCEmu), la carte Beagle, la carte Panda, ARMini, et pour ce qui nous intéresse plus particulièrement ici, sur le Raspberry Pi.

À bien des égards, le Raspberry Pi et RISC OS sont des partenaires idéaux. Bien entendu le point clé est que le Pi renferme un ARM 11 en son sein. Mais également, grâce à son héritage, RISC OS est peu exigeant en ressources système et fonctionne efficacement même lorsque la puissance du processeur et la mémoire sont sous-dimensionnés. Cela peut être dû en grande partie au fait que la plus grande partie du système d'exploitation est codé directement en assembleur ARM par des programmeurs astucieux. En outre, même si toutes les fonctionnalités essentielles sont fournies, des extensions au système et des bibliothèques sont téléchargeables sous forme de modules

disponibles si besoin.

Amorçage

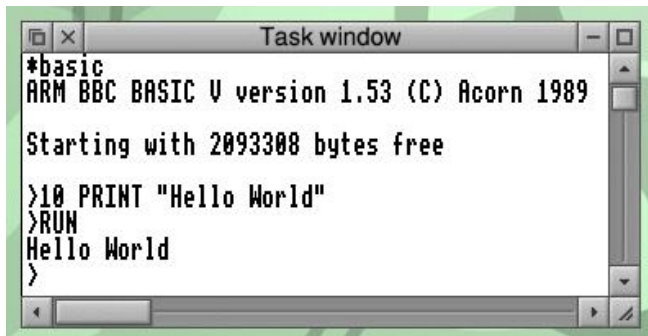
RISC OS démarre directement sur un bureau avec fenêtres, icônes, menus, pointeur (WIMP) et un superbe fond d'écran Raspberry Pi (voir capture d'écran), à partir duquel les applications peuvent être lancées. En bas de l'écran, un bandeau appelé barre d'icônes affiche les périphériques en bas à gauche, et les applications en cours d'exécution à droite. Un clic sur l'icône d'un périphérique (par exemple disque dur, carte SD, RAM) ouvre une fenêtre qui peut être utilisée pour la navigation et le lancement de différents types de fichier comme les programmes BASIC, les modules et les applications. De même un clic gauche sur une application ouvre une fenêtre pour permettre à l'utilisateur d'interagir avec elle, ou un clic sur le bouton du milieu fait apparaître un menu à partir duquel les options de configuration peuvent être définies et des actions exécutées. Les fenêtres de tâches donnent accès à une interface en ligne de commande (CLI) à partir de laquelle de nombreuses tâches courantes peuvent être exécutées et celles-ci peuvent être regroupées et empaquetées dans des fichiers Obey pour plus de commodité.

Les applications basées sur WIMP coopèrent par le biais de l'API (Interface de Programmation des Applications) basée sur l'interruption logicielle (SWI) qui est documentée dans les Manuels de Référence du Programmeur (PMRs). Ils sont disponibles sur le site Web foundation.riscos.com et représentent cinq gros volumes constituant l'équivalent de la Bible pour les développeurs d'application RISC OS.

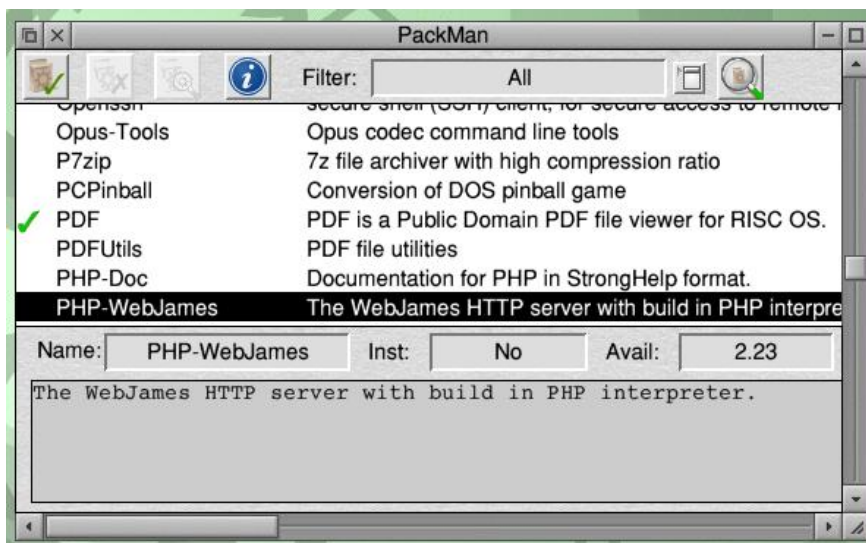
Les fichiers peuvent être épinglés sur le bureau pour en faciliter l'accès et des fonds d'écran et économiseurs d'écran peuvent être facilement configurés, ainsi n'importe qui un peu familier avec Windows ou Unix se sentira vite chez lui.

RISC OS est livré avec un navigateur Internet

appelé !NetSurf, bien qu'aujourd'hui l'accès au réseau soit assuré par un câble ethernet. Le BASIC ARM BBC peut être démarré à partir d'une Fenêtre de tâches (appuyez sur ctrl F12), à partir du prompt * tapez Basic.



Les applications fournies incluent un éditeur de texte !Edit, un logiciel de dessin !Draw et un autre de peinture !Paint, mais une quantité d'applications tierces sont disponibles, y compris des jeux, de la musique, PAO et des applications artistiques. Une majorité des plus utilisées d'entre elles sont librement installables en utilisant le gestionnaire de paquets fourni (!PackMan), qui ressemble à Linux Update Manager.



RISC OS Open a uni ses forces avec certains des développeurs de logiciels de premier plan dans la communauté RISC OS et vend avec une remise importante Nut Pi, un ensemble de logiciels spécifiques à RISC OS Pi.

Une fenêtre de tâches permet de répartir la mémoire entre les différentes parties du système et les applications utilisateur, et un certain nombre de modes d'écran de différentes résolutions et nombres de couleurs (jusqu'à 24 bits de profondeur de couleur) sont pris en charge.

Installation



Pour installer RISC OS sur Raspberry Pi, allez sur la page www.raspberrypi.org/downloads et suivez les

instructions de téléchargement. Le zip de téléchargement contient une image du disque pouvant être écrite sur une carte SD en utilisant le logiciel gratuit sous Windows Win32DiskImage ou l'utilitaire Unix dd de la même manière que pour la distribution Raspbian Wheezy.

Mon intérêt particulier pour RISC OS sur le Raspberry Pi est d'en faire un hôte pour les outils de développement et de démonstration Charm, pour l'éducation et le secteur amateur, dont je suis l'auteur. Une version sous licence GPL est fournie avec la distribution, mais la dernière version est disponible sur www.charm.qu-bit.co.uk qui est éventuellement compilable pour utiliser le coprocesseur VFP pour les opérations en virgule flottante. Lisez l'article sur le Charm dans la prochaine édition du MagPi pour plus d'informations.

En résumé RISC OS sur le Raspberry Pi est facile à installer, démarre en moins de 20 secondes, rapide, intuitif et facile à prendre en main alors pourquoi ne pas lui donner sa chance ?

Article de Pete Nowosad

Installer & configurer



Apprenez à installer Arch Linux, une distribution LINUX qui tourne sur des configuration minimalistes.

Beaucoup de gens pensent que Linux est un système d'exploitation, mais en fait, ce n'est qu'un noyau, la base. Pour le transformer en système d'exploitation proprement dit, vous avez besoin d'en rajouter un peu plus. Comme Linux est gratuit et partagé, de nombreuses personnes l'ont fait, et chacun d'eux a fait un peu différemment, créant de nombreuses différentes 'distributions', ou 'distros' pour faire court.

La principale distribution pour le Raspberry Pi proposée sur <http://www.raspberrypi.org/downloads> est Raspbian, une version de Debian. Toutefois, si vous descendez un peu sur la page, vous en verrez quelques autres, dont une appelée ArchLinux.

Alors, quelle est la différence entre Raspbian et Arch ? La principale différence est le gestionnaire de paquets et le mode de gestion des mises à jour.

Debian, et donc Raspbian, sont très strictes sur les paquets de mises à jour. Ils doivent passer par des tests, afin que les mainteneurs soient sûrs qu'ils sont stables et fonctionnent avant de les mettre à disposition des utilisateurs. C'est une bonne chose car cela signifie que le logiciel est presque assuré de fonctionner, le mauvais côté c'est que les mises à jour mettent du temps à atteindre les utilisateurs.

Arch Linux est différent à cet égard, libérant les mises à jour dès que possible. Pour cette raison, il est appelé une distro "à version glissante", puisque vous n'avez qu'à l'installer une fois, puis à chaque fois qu'un paquet reçoit une mise à jour, vous pouvez le mettre à niveau. Cela permet aux utilisateurs d'obtenir les mises à jour plus rapidement, même si cela signifie que le logiciel est plus instable. En cas de problème, vous pouvez simplement régénérer l'image sur la carte SD.

L'autre différence majeure entre les deux est que Raspbian est complètement prêt, tandis que Arch arrive avec le strict minimum, ce qui permet aux utilisateurs de choisir ce qu'ils veulent installer. Cela rend l'installation plus difficile pour les débutants, mais ce guide devrait aider à faciliter le processus.

Donc, si ça vous convient d'utiliser votre Pi avec des mises à jour continues, pourquoi ne pas essayer d'installer Arch ? Cependant, vous aurez besoin d'une connexion Internet pour votre Pi, donc si votre connexion internet est particulièrement lente ou si vous avez une limite de téléchargement faible, il peut être préférable de vous en tenir à Raspbian.

Commencez par télécharger la dernière image de <http://downloads.raspberrypi.org/download.php?file=/images/archlinuxarm/archlinux-hf-2012-09-18/archlinux-hf-2012-09-18.zip>. Puis enregistrez l'image sur la carte SD, en utilisant Win32DiskImager ou dd. Vous trouverez plus d'informations dans le numéro 2 de MagPi. Une fois que c'est fait, nous

pouvons passer à la configuration.

Premier démarrage

Le premier démarrage peut durer un certain temps, attendez jusqu'à ce qu'il soit fini. Une fois que vous arrivez à l'écran de connexion, utilisez comme nom d'utilisateur root et comme mot de passe root.

Vous aurez alors accès à un terminal. Vous remarquerez que si vous essayez startx, ça ne fonctionne pas. Cela vous donne une idée du peu qui est livré avec Arch. Ne vous inquiétez pas, nous aborderons l'installation d'une interface graphique.

Avant de commencer à faire quoi que ce soit, vous voudrez peut-être adapter la disposition du clavier pour votre pays. Faites-le en tapant :

```
loadkeys code_du_pays
```

Je suis en Angleterre, donc mon code pays serait uk (fr pour la France). La liste complète est ici: https://wiki.archlinux.org/index.php/KEYMAP#Keyboard_layouts. Ce n'est que temporaire, nous le réglerons de façon permanente par la suite.

Modification des paramètres de langue

Le nom d'hôte du système par défaut est alarmpi. Personnellement je n'aime pas ça, et je préférerais autre chose. Si vous êtes comme moi, changez le nom d'hôte du système en tapant:

```
echo nom_d_hote > /etc/hostname
```

où nom_d_hôte est le nouveau nom d'hôte que vous souhaitez. J'ai utilisé raspberrypi. Il ne sera effectif qu'après un redémarrage.

Ensuite, nous allons changer la langue et le fuseau horaire. Pour voir les fuseaux horaires disponibles, tapez:

```
ls /usr/share/zoneinfo
```

Choisissez le domaine qui vous convient le mieux (pour moi, ce serait Europe) et tapez:

```
ls /usr/share/zoneinfo/area
```

pour voir les fuseaux horaires spécifiques. Choisissez-en un, puis tapez:

```
ln -s /usr/share/zoneinfo/area/timezone /etc/localtime
```

le tout sur une seule ligne. Mon choix s'est porté sur Paris, donc area pourrait être l'Europe et le fuseau horaire de Paris. Si vous obtenez une erreur disant: "Le fichier existe", tapez:

```
rm /etc/localtime
```

Ensuite, tapez la commande précédente à nouveau.

Maintenant, modifions les paramètres régionaux. Ils sont utilisés pour déterminer le pays où vous vivez pour que des choses comme les dates soient affichées correctement. Pour ce faire, nous avons besoin de modifier certains fichiers. Tapez ce qui suit :

```
nano /etc/locale.gen
```

trouvez votre pays, et supprimez le symbole '#' en face de celui-ci. Par exemple, le mien serait fr_FR. Lorsque c'est fait, utilisez Ctrl+O pour enregistrer et Ctrl+X pour quitter. Ensuite, tapez :

```
locale-gen
```

Maintenant, nous avons besoin de créer un autre fichier, tapez :

```
nano /etc/locale.conf
```

et modifiez-le avec les mêmes langue et code de pays que précédemment.

Enfin, nous avons besoin de configurer la console pour qu'elle utilise le clavier du pays en permanence. Pour cela tapez :

```
nano /etc/vconsole.conf
```

et changer KEYMAP pour le code du pays que vous avez utilisé avec la commande loadkeys précédemment.

Tous les paramètres de langue sont maintenant réglés, donc si vous voulez, vous pouvez redémarrer pour voir les changements, en tapant :

```
reboot
```

Utilisation de pacman

Le gestionnaire de paquets d'Arch Linux est appelé pacman, et nous l'utilisons pour installer des paquets. Pour installer un paquet, tapez :

```
pacman -S <package name>
```

Essayez-le avec le paquet sudo, car nous en aurons besoin plus tard.

Comme Arch est à «mise à jour glissante», beaucoup de mises à jour sont sorties depuis la parution de l'image, donc pour mettre à niveau tous vos paquets :

```
pacman -Syu
```

Cela devrait très bien fonctionner avec l'image la plus récente, mais vous avez besoin d'une connexion Internet. En raison de la rapidité avec laquelle les mises à jour sortent, il est recommandé d'exécuter une mise à niveau complète régulièrement, une fois par semaine ou peut-être même une fois par jour pour rester à la pointe.

Si vous souhaitez supprimer un progiciel, vous pouvez le faire avec :

```
pacman -R <package name>
```

Et pour voir une liste des paquets installés, tapez :

```
pacman -Q
```

Ajout d'un nouvel utilisateur

Il est d'une importance vitale de créer un nouvel utilisateur pour notre système, car se connecter en tant que root pose des problèmes de sécurité. Pour ajouter un nouvel utilisateur, entrez :

```
adduser
```

et suivez les instructions pour ajouter un nouvel utilisateur.

Ensuite, nous avons besoin de les ajouter à la liste des **sudoers** afin qu'ils puissent toujours installer des programmes, mais d'une manière plus sûre. Si vous ne l'avez pas déjà fait, installez sudo. Pour ajouter l'utilisateur au fichier sudoers, tapez :

```
export EDITOR = nano && visudo
```

Cela vous permettra de modifier le fichier **sudoers** avec l'éditeur habituel **nano**. Trouvez la ligne qui indique **root ALL=(ALL) ALL** et copiez-la sur une autre ligne, remplacez root par le nom d'utilisateur du nouvel utilisateur. Si vous voulez que sudo ne demande pas votre mot de passe, comme dans l'installation par défaut de Raspbian, ajoutez **NOPASSWD** : avant le ALL final.

Enfin, nous pouvons changer le mot de passe du compte root, en utilisant la commande :

```
passwd root
```

Donc veillez à choisir un mot de passe suffisamment sûr.

Après cela, nous en avons fini avec la configuration de base ! Tapez :

```
logout
```

pour vous déconnecter de root et connectez-vous avec le nouvel utilisateur que vous venez de configurer.

Mise en place d'une interface graphique

Cette dernière partie est facultative, mais si vous préférez plus qu'une simple ligne de commande, vous devriez le faire. Pour installer une interface graphique, il suffit de taper :

```
pacman -S gamin dbus xorg-server xorg-xinit xorg-server-utils mesa xf86-video-fbdev xf86-video-vesa xfce4
```

le tout sur une seule ligne. Une fois l'installation terminée, tapez :

```
cp /etc/skel/.xinitrc ~/.xinitrc
```

puis :

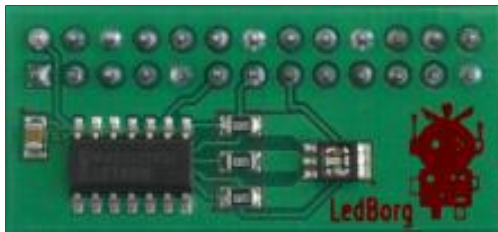
```
echo "exec startxfce4" >> ~/.xinitrc
```

et enfin :

```
startx
```

Votre environnement graphique devrait alors démarrer. Félicitations, vous avez maintenant un système Arch Linux opérationnel sur votre Raspberry Pi ! Vous voudrez peut-être modifier le fichier config.txt, mais ce processus est le même que pour Raspbian. Amusez-vous bien !

Article d'Alex Kerr



Introduction à Vala

Écriture d'un contrôleur web simple pour la carte RPI LedBorg

Le langage Vala est, parmi les langages de programmation, un petit nouveau et il est encore en observation chez de nombreux programmeurs. Nous allons utiliser Vala pour communiquer avec LedBorg de www.piborg.com/ledborg via Internet. [NDLR: dans ce numéro, nous verrons comment faire la même chose avec Python.]

Vala est un langage de style C# construit sur le système objet GLib fournissant un accès facile aux bibliothèques et sous-systèmes de base de GNOME. Le compilateur, valac, transforme le code Vala en C, et à son tour déclenche le compilateur C du système pour produire du code natif. Cela signifie que, contrairement à C#, il n'y a pas de framework .NET ou de machine virtuelle. En effet, c'est une façon d'écrire des applications C de plus haut niveau. La page d'accueil du projet est <https://live.gnome.org/Vala>.

Bien que Vala soit basé sur la GLib et les bibliothèques GNOME, il ne sert pas qu'au développement d'applications de bureau GNOME et convient également pour écrire des applications et services pour la console.

LedBorg

La carte LedBorg est pré-assemblée, elle se connecte sur les broches GPIO et comprend une DEL RVB ultra-lumineuse. Chaque canal - rouge, vert et bleu, a trois intensités : éteint, demi luminosité, et pleine luminosité. Le pilote LedBorg crée le fichier de périphérique `/dev/ledborg`. Il s'agit d'un excellent exemple de la philosophie UNIX au travail avec les périphériques vus comme des fichiers au lieu de mystérieuses API. Nous pouvons lire et écrire dans `/dev/ledborg` comme nous le ferions avec n'importe quel autre fichier, par exemple :

```
echo "202" > /dev/ledborg
```

Les trois chiffres contrôlent R, V et B, chacun valant 0, 1 ou 2, et correspondent aux trois intensités, par exemple : '202' fait émettre un violet brillant à la LedBorg (rouge + bleu).

Contrôle par le réseau

J'ai décidé qu'un moyen facile de contrôler LedBorg à distance était d'utiliser le protocole bien connu HTTP. En utilisant LibSoup dans Vala, il est facile de mettre en place un serveur HTTP léger qui peut répondre aux

requêtes. Les tests devraient être simples depuis n'importe quel navigateur Web sur le réseau.

Le serveur accepte les requêtes GET au format d'URL suivant :
`/?action=SetColour&red=x&green=y&blue=z`
où x, y et z sont des nombres entiers compris entre 0 et 2.

Pour faciliter l'utilisation, le programme répond aussi à toutes les requêtes à partir d'un formulaire HTML minimum, contenant une liste déroulante par couleur, et un bouton submit.

Le Code

Pour essayer ce code, les paquets suivants seront installés avec leurs dépendances. En supposant que Raspbian/Debian est l'OS que vous utilisez :

```
$ sudo apt-get install valac \
  libsoup2.4-dev
```

Après avoir entré le code ci-dessous, et l'avoir enregistré dans `LedBorgSimpleServer.vala` vous pouvez alors le compiler avec la commande suivante :

```
$ valac --pkg libsoup-2.4 --pkg \
  gio-2.0 --pkg posix --thread -v \
  LedBorgSimpleServer.vala
```

Vous pouvez entrer cette ligne de commande dans un fichier texte et l'enregistrer sous `compile.sh` - puis rendre ce fichier exécutable avec :

```
$ chmod +x compile.sh
```

Vous pouvez recompiler avec

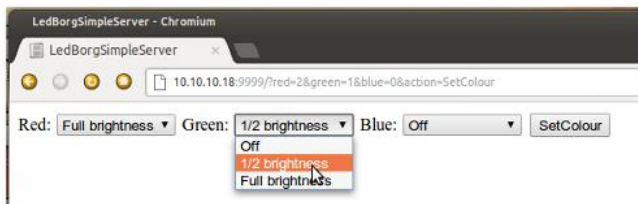
```
$ ./compile.sh
```

L'option `-v` génère une sortie verbeuse, donnant une idée de ce qui se passe. Si vous voulez voir le code C généré, ajoutez l'option `-C` pour obtenir `LedBorgSimpleServer.c`

Exécutez le programme avec `./LedBorgSimpleServer` et entrez l'adresse de votre Pi dans un navigateur, en ajoutant `:9999` pour spécifier le numéro de port, par exemple : `http://192.168.1.69:9999`.

Le code manque de robustesse : nous ne vérifions pas la présence des paramètres rouge, vert et bleu du GET, ni ne validons leurs valeurs. Il n'y a pas de retour dans le HTML envoyé au client, pour dire si l'opération a réussi, quelle couleur a été définie, ou si la LedBorg n'est pas connectée.

Ces ajouts peuvent être un exercice pour le lecteur!



```
// LedBorgSimpleServer.vala

// Les espaces de nom que nous utiliserons
using GLib;
using Soup;

// Notre classe principale
public class LedBorgSimpleServer :
GLib.Object {
    // définit le numéro de port à écouter
    static const int LISTEN_PORT = 9999;

    // Définit le fichier correspondant à la carte
    static const string DEVICE = "/dev/ledborg";

    // la méthode utilisée au démarrage
    public static int main (string[] args)
    {
        // crée une instance du serveur
        var server = new
Soup.Server(Soup.SERVER_PORT, LISTEN_PORT);

        // gère les requêtes du client
        server.add_handler("/", default_handler);

        // démarre le serveur
        server.run();

        return 0;
    }

    // gestionnaire http par défaut
    public static void
default_handler(Soup.Server server,
Soup.Message msg, string path,
GLib.HashTable<string, string>? query,
Soup.ClientContext client)
    {
        // traite une requête
        if(query != null)
        {
            // vérifie les paramètres
            if(query["action"] == "SetColour")
            {
                // récupère les valeurs RVB dans les
paramètres
                string red = query["red"];
                string green = query["green"];
                string blue = query["blue"];

```

```
        /* construire notre chaine couleur RVB
chaque chiffre valant 0, 1 or 2:
éteint, demi ou pleine brillance */
        string colour = red + green + blue;

        // modifier la couleur
        do_colour_change(colour);
    }
}

// construire la chaine html pour le client
string html = ""
<html>
<head>
    <title>LedBorg Serveur Simple</title>
</head>
<body>
    <form method="get" action="/">
        Rouge:<select name="red">
            <option value="0">Éteint</option>
            <option value="1">Faible</option>
            <option value="2">BrillantFull</option>
        </select>
        Vert:<select name="green">
            <option value="0">Éteint</option>
            <option value="1">Faible</option>
            <option value="2">BrillantFull</option>
        </select>
        Bleu:<select name="blue">
            <option value="0">Éteint</option>
            <option value="1">Faible</option>
            <option value="2">BrillantFull</option>
        </select>
        <input type="submit" name="action"
value="SetColour" />
    </form>
</body>
</html>
    """;

    // renvoie le html au client
    msg.set_status_full(
        Soup.KnownStatusCode.OK, "OK");
    msg.set_response("text/html",
        Soup.MemoryUse.COPY, html.data);
}

// changer la couleur
public static void
do_colour_change(string colour)
{
    /* Ici nous utilisons la gestion de fichier
POSIX
pour écrire dans le fichier, plutôt que la
gestion
GIO de vala, puisque nous n'avons pas besoin
de la
sécurité de GIO pour accéder à /dev */
    // ouvre le fichier pour écrire
    Posix.FILE f = Posix.FILE.open(DEVICE, "w");

    // écrit la chaine de couleur dans le fichier
    f.puts(colour);
}
}

```

Article de Ross Taylor



Le Guide des Événements du MagPi

Vous voulez être tenu au courant de tout ce qui concerne le Raspberry Pi dans votre région ?

Alors cette nouvelle rubrique est pour vous ! Nous avons pour objectif de lister tous les événements Raspberry Jam dans votre région en vous fournissant un calendrier RPi pour le mois à venir.

Organisez-vous un événement Raspberry Pi ? Voulez-vous le promouvoir ?
Contactez-nous par courriel à : editor@themagpi.com

Preston Raspberry Jam

Quand: Samedi 9 février 2013 à 10h00
Où: [Accrington Academy, Queens Road West, BB5 4FF, UK](#)

Cette conférence se tient de 10h00 à 17h00 et est hébergée chez TechWizZ
Billets et informations sont disponibles sur <http://techwizz.eventbrite.co.uk>

New York City Raspberry Jam

Quand: Jeudi 21 février 2013 à 19h00
Où: [Two Sigma, 16th Floor, 100 Avenue of the Americas, New York, NY, USA](#)

Apportez vos projets, vos idées et voyez ce que font les autres. Il y aura même une démo du WebIDE d'Adafruit. Plus d'informations sur <http://www.meetup.com/NYC-Raspberry-Jam>

Sheffield Raspberry Jam

Quand: Dernier mercredi du mois à 18h30
Où: [1st Floor, The Workstation, Grinders Hill / Brown St., Sheffield, S1 2BX, UK](#)

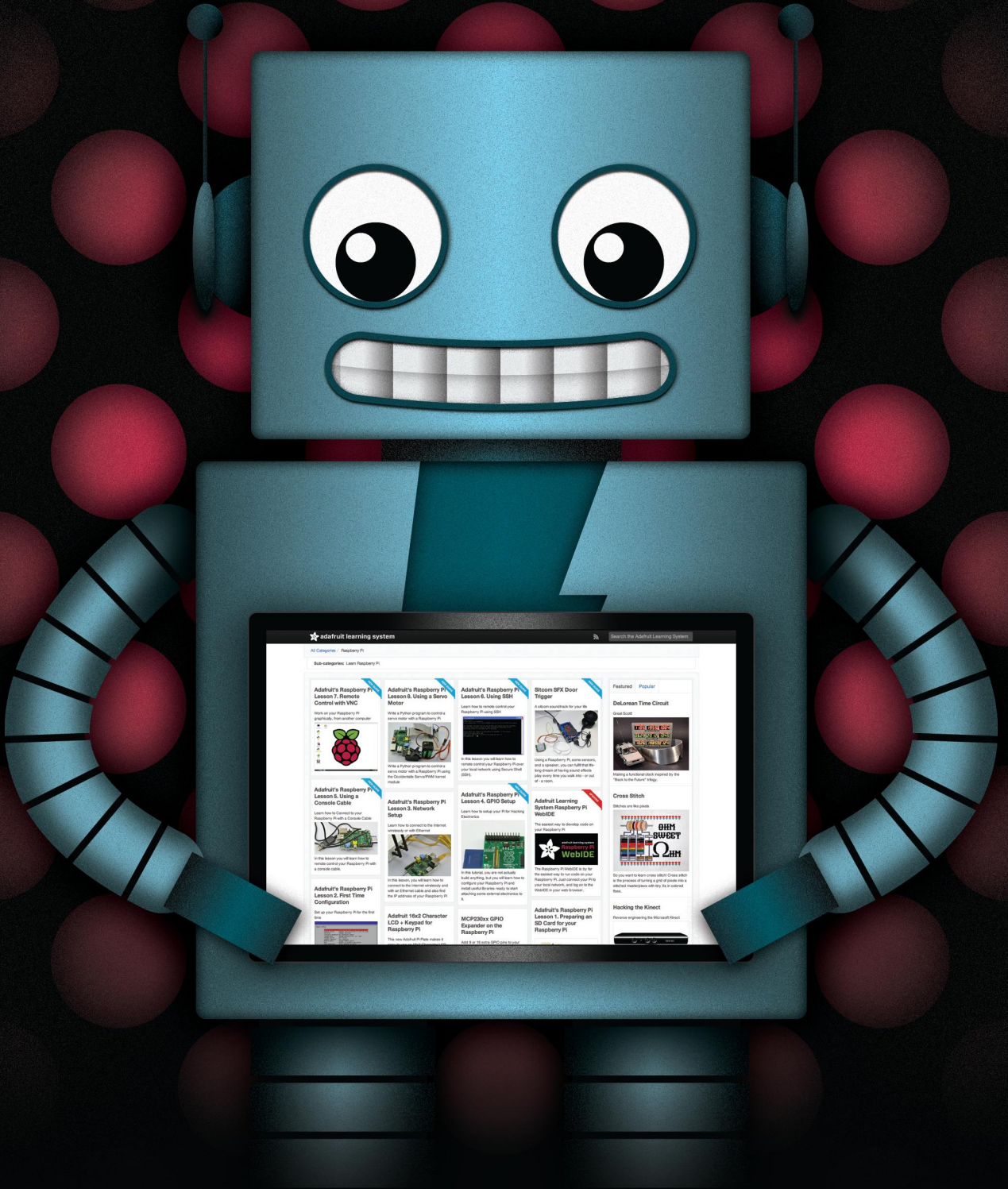
Les réunions ont lieu chez GISThub. Les portes ouvrent à 18h20 et la rencontre dure de 18h30 à 20h30. Plus d'informations sur <http://sheffieldraspi1210.eventbrite.com>

CAS York Hub Meeting

Quand: Mercredi 13 février 2013 à 16h30
Où: [National STEM Centre, University of York, YO10 5DD, UK](#)

L'événement se tiendra de 16h30 à 17h30 et s'intitule "Engaging Pupils with Raspberry Pi".
Billets et informations disponibles sur <http://rcasstem.eventbrite.co.uk>

BUILD AMAZING THINGS & LEARN HOW TO PROGRAM WITH THE RASPBERRY PI



Tutoriel 5 - Structures, fichiers d'entête et analyse de données.

Bon retour à la caverne du C. Ce tutoriel donne un exemple de programme construit à partir de plusieurs fichiers sources C et compilé avec un Makefile. Avant de continuer, comment vous en êtes-vous sortis avec le défi précédent ?

Solution du défi

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/sysinfo.h>
int main() {
    int i = 0;
    float ramUsed;
    char gpCmdOne[250], gpCmdTwo[250], systemCmd[1000];
    FILE *cmdPtr = 0, *outPtr = 0;
    char c, fileName[100], *strPtr = 0;
    struct sysinfo info; /* Structure sysinfo pour stocker l'état. */
    cmdPtr = popen("hostname","r"); /* Lance la commande hostname. */
    if(!cmdPtr) return 1;
    strPtr = &fileName[0]; /* Pointeur sur la chaîne. */
    while((c=fgetc(cmdPtr)) != EOF) { /* Accès à chaque caractère. */
        *strPtr = c; /* Fixe la valeur du caractère. */
        strPtr++; /* Va à la position suivante dans le tableau. */
    }
    pclose(cmdPtr); /* Ferme le fichier hostname. */
    strPtr--; /* Va un élément en arrière pour ignorer la fin de ligne. */
    sprintf(strPtr,"-data.txt"); /* Ajoute le suffixe. */
    printf("%s\n",fileName);
    outPtr = fopen(fileName,"w"); /* Ouvre le fichier de sortie. */
    if(!outPtr) return 1; /* Retourne une erreur s'il est impossible d'ouvrir
le fichier */
    for(i=0;i<60;i++) {
        sysinfo(&info); /* Lit les infos système */
        ramUsed = info.totalram - info.freeram;
        ramUsed /= 10240.0;
        fprintf(outPtr,"%d %f %d\n", i, ramUsed, info.loads[0]); /* Écrit la ram
utilisée. */
        usleep(500000); /* Pause d'1/2 seconde. */
    }
    fclose(outPtr); /* Ferme le fichier de sortie. */
    /* Maintenant, affichage des données */
    sprintf(gpCmdOne, "plot \'%s\' using 1:2 title \'%s\' ", fileName, "Ram
utilisée");
    sprintf(gpCmdTwo, ", \'%s\' using 1:3 title \'%s\'\n", fileName, "Charge");
    /* Crée la commande complète, avec le tube vers gnuplot */
    sprintf(systemCmd, "echo \"%s%s\" | gnuplot -persist", gpCmdOne, gpCmdTwo);
    system(systemCmd); /* Exécute la commande système. */
    return 0; /* Retour réussi vers le système. */
}
```

La solution met en œuvre des fonctions et des techniques vues précédemment. Des façons plus simples existent pour créer le nom de fichier à partir de l'hôte. Le C propose un entête string.h avec des fonctions

utiles pour gérer les chaînes. La liste complète est visible avec `man string`. `strcat` peut concaténer des chaînes :

```
char fileName[100]="monHôte", suffix[10]="-data.txt"
strcat(fileName, suffix); /* Ajoute le suffixe, résultat dans fileName. */
```

Le nom d'hôte peut être lu avec `fgets` plutôt qu'avec `fgetc` :

```
fgets(fileName,100,cmdPtr); /* Lit jusqu'à EOF, fin de ligne ou 99 caractères. */
```

où 100 est la taille du tableau de caractères `fileName`. Enfin, la fonction `gethostname` de `unistd.h` permet aussi de le lire :

```
#include <string.h>
#include <stdio.h>
#include <unistd.h>

int main() {
    char fileName[100], suffix[10]="-data.txt";
    gethostname(fileName,100);
    strcat(fileName,suffix); /* Ajoute le suffixe au nom de fichier */
    printf("%s\n",fileName);
    return 0;
}
```

Structures

Elles ont été survolées dans le numéro 6 pour lire les infos système. Elles utilisent un bloc continu de mémoire comme les blocs en FORTRAN. Leur syntaxe est similaire aux classes C++ avec les données-membres publiques. Une struct est définie par un nom et un bloc de définition de variables qui peuvent aussi être des struct. Une structure simple :

```
struct dataPoint {
    unsigned int timeSeconds;
    float value;
};
```

`timeSeconds` est défini en premier en mémoire, puis `float value`. La taille du struct en mémoire est la somme des tailles des variables. Elle doit être définie avant d'être utilisée et se trouve en principe dans le fichier d'entête, mais peut être écrite dans le même fichier avant son utilisation. Pour tester cette struct :

```
int main() {
    /* Déclare une variable de type "struct dataPoint". */
    struct dataPoints;
    /* Assigne à la structure s des valeurs initiales. */
    s.timeSeconds = 60;
    s.value = 100.0;
    /* Affiche les emplacements en mémoire et les tailles */
    printf("sizeof(s) = %ld\n", sizeof(s));
    printf("&(s.timeSeconds) = %p\n", &(s.timeSeconds));
    printf("&(s.value) = %p\n", &(s.value));
    printf("sizeof(unsigned int) = %ld\n", sizeof(unsigned int));
    printf("sizeof(float) = %ld\n", sizeof(float));
    return 0;
}
```

où le programme définit des valeurs et affiche leurs adresses pour illustrer la structure en mémoire.

Quand elles sont passées en paramètres, ce sont par défaut des copies locales à la fonction qui sont créées. Au retour de la fonction appelante, la valeur de la structure est donc inchangée. Le comportement est le même que pour une variable classique passée à une fonction :

Suite sur la page suivante...


```
void printDataPoint(struct dataPoint dp) {
    printf("timeSeconds = %d, value = %f\n", dp.timeSeconds, dp.value);
}
```

Pour modifier les valeurs dans une fonction et les conserver, il faut se servir des pointeurs :

```
void clearDataPoint(struct dataPoint *dp) {
    dp->timeSeconds = 0;
    dp->value = 0.;
}
```

où la syntaxe `dp->timeSeconds` est équivalente à `(*dp).timeSeconds`. Différent du raccourci `"->"`, le comportement est exactement le même que pour les variables abordées dans le n°5.

Fichiers d'entête

Pour illustrer l'utilisation des en-têtes, l'exemple suivant définit des fonctions et une structure de données d'histogramme. Ceux-ci sont très utiles pour surveiller des résultats sur la durée, fournir des schémas récapitulatifs ou pour le stockage des données elles-mêmes.

Les en-têtes doivent être inclus pour pouvoir utiliser les fonctions des bibliothèques standards ou celles implémentées dans d'autres sources C. Ils contiennent des structures de données et les déclarations des fonctions, mais pas leur implémentation. Celle-ci est faite dans les fichiers `.c` qui sont compilés pour produire des bibliothèques statiques ou dynamiques, ou pour les lier avec. D'autres en-têtes peuvent être écrits pour contenir des structures de données ou des définitions de fonctions de manière identique aux en-têtes standards.

```
#ifndef HISTOGRAM_H
#define HISTOGRAM_H
#define MAX_BINS 1000
/* Définit une structure pour stocker les données de l'histogramme. */
struct histogram {
    unsigned int nBins;
    float xMin ;
    float xMax;
    float binContents[MAX_BINS];
};
/* Définit la structure en tant que nouveau type, comme un raccourci. */
typedef struct histogram Histogram;
/* Remplit un histogramme. */
int fillHist(Histogram *, float value, float weight);
/* Enregistre un histogramme dans un fichier. */
int saveHist(Histogram *, FILE *);
#endif
```

est un fichier d'en-tête appelé `histogram.h` qui définit une structure et déclare des fonctions, mais n'implémente pas les fonctions qu'il définit. Si le fichier est inclus dans un autre fichier d'en-tête, il pourrait être inclus plusieurs fois dans un programme. Pour empêcher cette double déclaration, la directive de précompilation `ifndef` est employée. La condition est remplie lors de la première inclusion, et fausse pour les suivantes. La directive `define` définit des valeurs qui seront remplacées à l'exécution du précompilateur, juste avant la compilation du code. Comme l'allocation dynamique de mémoire n'a pas encore été vue, une taille de tableau fixe est utilisée pour `binContents`. Enfin, `typedef` sert à simplifier la déclaration de la variable `Histogram`. Le fichier d'en-tête doit être inclus dans le programme avant d'utiliser les fonctions ou les structures :

```
#include "histogram.h"
#include <stdio.h>
#include <stdlib.h>
int main() {
    unsigned int i;
    Histogram h ; /* Crée une structure d'histogramme */
    init Hist(&h,10,0.,10.); /* Initialise l'histogramme */
    for(i =0;i<1000;i++) { /* Génère 1000 points aléatoires */
        fillHist(&h,10*(float)rand()/RAND_MAX,1.0); /* Chaque valeur de l'histogramme. */
    }
}
```

```

}
saveHist(&h,"Hist.txt"); /* Sauve l'histogramme. */
return 0;
}

```

Le programme crée l'histogramme à partir de nombres aléatoires compris entre 0 et 1. Il ne fonctionnera pas sans l'implémentation des fonctions définies dans histogram.h. Celle-ci se trouve dans un fichier .c :

```

#include "histogram.h"
#include <stdio.h>

int initHist(Histogram *hist, unsigned int nBins, float xMin, float xMax) {
    unsigned int i;
    if((hist->nBins+2) >= MAX_BINS) {
        printf("Erreur: trop de classes demandées.\n");
        return 1; /* Une erreur s'est produite. */
    }
    hist->nBins = nBins;
    hist->xMin = xMin;
    hist->xMax = xMax;
    for(i=0;i<(hist->nBins+2);i++) hist->binContents[i] = 0.;
}

int fillHist(Histogram *hist, float value, float weight) {
    unsigned int ibin;
    float binSize;
    if(value < hist->xMin) ibin = 0; /* Dépassement par le bas */
    else if(value >= hist->xMax) ibin = hist->nBins+1; /* Dépassement par le haut */
    else { /* Cherche la classe appropriée. */
        ibin = 1;
        binSize = (hist->xMax - hist->xMin)/hist->nBins;
        while(value >= (ibin*binSize + hist->xMin) &&
            ibin < hist->nBins && ibin < MAX_BINS) {
            ibin++;
        }
    }
    if(ibin >= MAX_BINS) { /* Rester dans les limites du tableau */
        printf("Erreur: ibin = %u est hors limite\n",ibin);
        return 1; /* Une erreur s'est produite. */
    }
    hist->binContents[ibin] += weight; /* Ajoute le poids */
    return 0;
}

int saveHist(Histogram *hist, const char *fileName) {
    FILE *outputFile = 0;
    unsigned int ibin;
    float binSize;
    outputFile = fopen(fileName, "w"); /* Ouvre le fichier de sortie. */
    if(!outputFile) { /* Si le fichier n'est pas ouvert.*/
        printf ("Erreur : impossible d'ouvrir %s en écriture\n",fileName);
        return 1; /* Une erreur s'est produite. */
    }
    binSize = (hist->xMax - hist->xMin)/hist->nBins;
    /* Ecrit les centres des classes et leur contenu dans le fichier. */
    ibin=0;
    while (ibin < (hist->nBins+2) && ibin < MAX_BINS) {
        fprintf(outputFile,"%lf %lf\n",
            binSize*((double)ibin+0.5) + hist->xMin,
            hist->binContents[ibin]);
        ibin++;
    }
    fclose(outputFile); /* Ferme le fichier de sortie. */
    return 0;
}

```

Suite sur la page suivante...

Au lieu d'appeler gcc plusieurs fois, un Makefile permettra de construire les fichiers sources et de produire l'exécutable :

```
CC = gcc
TARGET = hist
OBJECTS = $(patsubst %.c,%.o, $(wildcard *.c))

$(TARGET): $(OBJECTS)
    @echo "*** Liaison de l'Executable"
    $(CC) $(OBJECTS) -o $(TARGET)

clean:
    @rm -f *.o *~

veryclean: clean
    @rm -f $(TARGET)

%.o: %.c
    @echo "*** Compilation du source C"
    $(CC) -c $<
```

Plus d'informations sur make se trouvent dans le n°7. Placez les deux fichiers .c dans le même répertoire, avec histogram.h et Makefile. Tapez ensuite make pour construire l'exécutable. Le programme lancé va produire un fichier contenant les sommes des poids dans les classes, en-dessous et au-dessus. Le format est choisi pour permettre de tracer l'histogramme avec gnuplot :

```
gnuplot
plot 'Hist.txt' with boxes
```

Comme dans le précédent tutoriel, la commande pour tracer peut être intégrée au programme. Pour enregistrer le graphique en png, tapez :

```
set term png
set output 'Hist.png'
replot
```

après la commande de traçage d'origine.

Le défi

Utilisez cet article et le précédent pour faire l'histogramme de la charge système sur 30 minutes. Ajoutez des fonctions à histogram.h et histogram.c pour calculer la moyenne et l'écart-type de la distribution. La moyenne d'un histogramme se calcule à partir de :

```
xMean = 0.;
for (i=1;i<=nBins;i++) { /* Ignore débordements */
    xMean += binContents[i]/nBins;
}
```

L'écart-type est donné par :

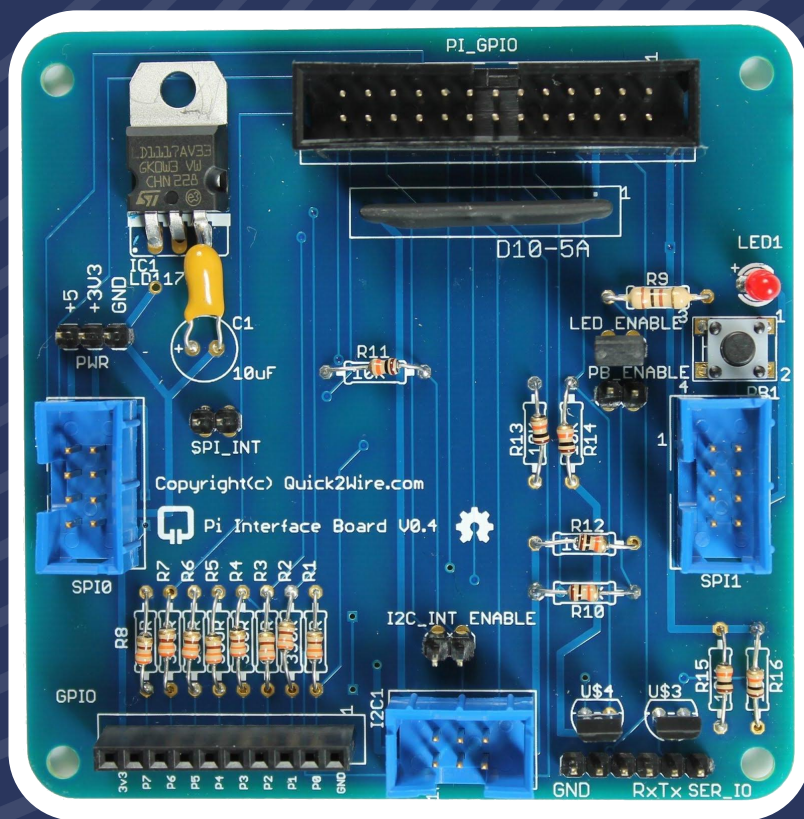
```
xStdDev = 0.;
for(i=1;i<=nBins;i++) { /* Ignore débordements */
    xStdDev += pow(xMean - binContents[i],2)/(nBins-1);
}
if(xStdDev > 0.) xStdDev = sqrt(xStdDev);
```

La solution du problème sera donnée la prochaine fois.

Article de W. H. Bell

Quick2Wire

SAFE AND SIMPLE
CONNECTION TO YOUR
RASPBERRY PI



HARDWARE

- Interface board
- I²C Port Extender
- Analogue board

SOFTWARE

- For GPIO, I²C, SPI
- Device Libraries
- Examples

SUPPORT

- Articles
- Tutorials
- Forum

Interface board: £13.86 | Port Extender: £9.80 | Combo: £22.66 (save £1.00)

Prices include UK VAT but exclude Postage and Packaging: from £2.70

Find out more at quick2wire.com

THE SCRATCH PATCH



Contrôle GPIO avec Scratch

Cet article a pour but de rendre SIMPLE COMME PI l'accès et l'emploi du GPIO en Scratch pour contrôler des lumières et réagir à des interrupteurs et des capteurs.

Tandis que le Raspberry Pi est un outil formidable pour créer des logiciels grâce à des langages comme Scratch, Python, C etc., le meilleur moyen de le rendre vivant et d'ajouter plus de possibilités à ce petit ordinateur à bas prix consiste à jouer en bidouillant le matériel et à traiter des données physiques. Cela implique d'utiliser le Pi pour contrôler différentes choses comme des DEL, réagir à des interrupteurs et des capteurs. Plus que jamais, cela inclut aussi la connaissance et l'apprentissage à la fois du logiciel et du matériel dans un environnement concret - pas seulement coder pour le plaisir de coder, mais, par exemple, créer des robots et les programmer pour qu'ils fassent des choses amusantes !

Cet article est basé sur un message du blog Cymplecy de Simon Walters (<http://wp.me/p2C0q1-27>), un assistant d'enseignant TICE en école primaire et gourou Scratch en général !

Éléments nécessaires - un Raspberry Pi avec Raspbian et une connexion Internet, une platine Labdec, quelques diodes émettrices de lumière (DEL), quelques résistances et des bouts de fils électriques. Cout total 5-10 £ (Pi non inclus).

Comment contrôler les broches GPIO avec le Raspberry Pi en Scratch

Votre Raspberry Pi doit être connecté à Internet pour installer le logiciel mais la connexion n'est pas nécessaire pour lancer ScratchGPIO. Copiez le texte ci-dessous (depuis sudo jusqu'à gpio.sh) et collez-le dans une fenêtre LX Terminal, puis lancez-le pour télécharger l'installateur :

```
sudo wget http://db.tt/jCIVXBJE -O /boot/install_scratch_gpio.sh
```

Ensuite, tapez et lancez : `sudo /boot/install_scratch_gpio.sh`

Cela va installer tous les logiciels supplémentaires requis et quelques exemples simples. (Si vous n'avez pas Internet sur votre Pi, placez votre carte SD dans un lecteur puis, dans un navigateur, faites un clic-droit et enregistrez le script directement sur la carte, remettez-la dans le Pi et lancez la seconde instruction)

Mise en place des composants

Soyez extrêmement attentifs quand vous connectez des éléments matériels sur les broches du GPIO car cela peut endommager le Pi - ne le faites que si vous êtes sûrs de pouvoir suivre ces instructions correctement. Vous devez au moins avoir la platine et des raccords mâle-femelle 0,1 (disponibles chez RS/CPC ou votre vendeur local). Vérifiez le câblage des broches GPIO sur la platine pour être sûr de ce qu'elles font.

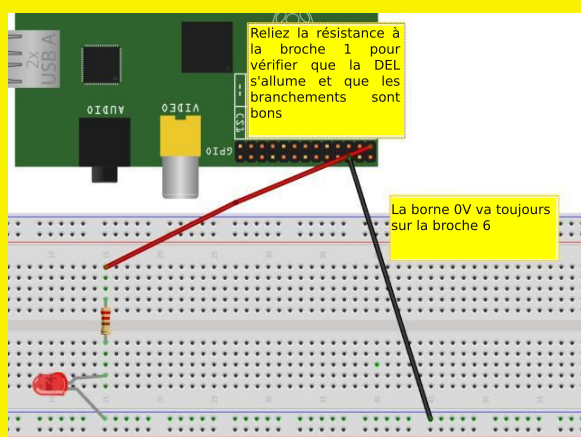


Figure 1 - Test de DEL

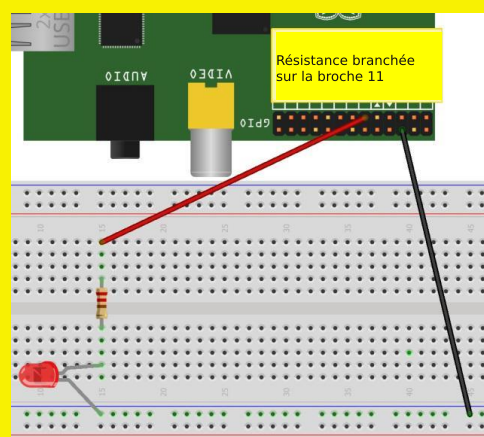


Figure 2 - Test du GPIO

Figure 1 ci-dessus, câblez la broche 1 (3,3V) sur une résistance de (au moins) 330 ohm - connectez-la à la longue patte (qui est la patte positive) d'une DEL puis connectez l'autre patte à la broche 6 (masse). Ça devrait l'allumer. Sinon, changez le sens de la DEL, car les polarités sont sans doute inversées. Une fois que ça marche, vous pouvez déplacer le fil rouge (ou la couleur que vous avez utilisée) de la broche 1 à la broche 11, comme sur la figure 2.

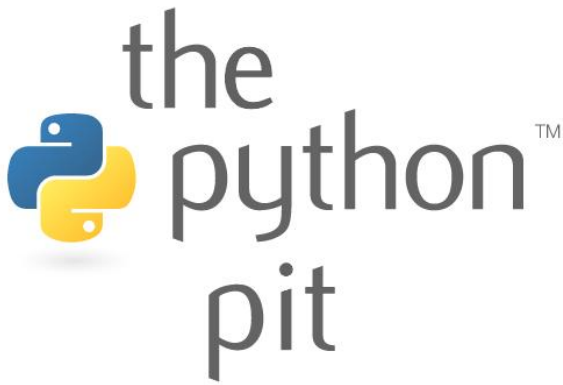
Vous pouvez maintenant lancer l'icône Scratch spéciale (Scratch GPIO) de votre bureau. C'est en fait une version normale de Scratch, elle lance juste un petit programme Python en arrière-plan pour communiquer avec le GPIO. Si d'autres programmes Python accédant au GPIO tournent déjà, l'ouverture de Scratch GPIO peut planter votre Pi. Pour éviter cela, ouvrez une fenêtre LX Terminal et lancez : `sudo killall python`

Pour tester l'accès au GPIO dans Scratch, cliquez sur Fichier>Ouvrir et sélectionnez blink11 dans /home/pi/Scratch. Une fois le programme chargé, cliquez sur le drapeau vert et votre DEL devrait clignoter : allumée 1 seconde, éteinte 2 secondes - voir dépannage sur le blog Cymplecy si ça ne marche pas.

Que puis-je faire d'autre avec Scratch et le GPIO ?

La prochaine fois, nous verrons d'autres façons passionnantes d'utiliser le GPIO avec Scratch.

Article de Simon Walters et Aaron Shaw



Dans le repaire du Python de ce mois, nous vous montrerons comment contrôler votre Raspberry Pi avec votre téléphone intelligent grâce au cadre web.py de www.webpy.org.

Cet article propose une alternative à l'utilisation de Vala, également abordé dans ce numéro.

Web.py vous permet de présenter des formulaires web sur des clients (Chrome sur Android et Safari sur iOS) et de récupérer le contenu via HTTP POST sur le serveur web (votre Pi). Les valeurs retournées sont fournies directement à votre script Python qui peut alors exécuter toute commande de votre choix. Installer Apache ou Lighttpd est inutile car web.py embarque un serveur web.

Comme exemple, nous utiliserons le navigateur de votre téléphone connecté à votre réseau local en WiFi et votre Pi effectuera un contrôle distant basique de LedBorg (www.piborg.com). LedBorg est une matrice de DEL préassemblée capable d'afficher 27 couleurs. Cependant, pour cette démonstration, nous n'utiliserons que rouge, vert et "noir" (tout éteint). La technique décrite peut aussi bien fonctionner pour tout autre projet de contrôle distant du Pi. Le code spécifique à LedBorg doit juste être remplacé pour correspondre à vos besoins.

Web.py peut être installé depuis la ligne de commande via PIP (voir comment installer PIP dans le numéro 8) :

```
sudo pip install web.py
```

Créez ensuite sur votre Pi un répertoire (pyserver dans cet exemple) qui servira de racine au serveur web que web.py va lancer. Créez deux sous-dossiers et enregistrez

dans ceux-ci les fichiers suivants :

```
pyserver/ledborg.py
pyserver/templates/index.html
pyserver/static/styles.css
```

Concentrons-nous sur les deux premiers fichiers : ledborg.py contient notre code Python et index.html le modèle de page web qui sera appelé lors de l'exécution du programme. La feuille de styles, styles.css, est optionnelle et remplace les boutons traditionnels gris terne des formulaires web par ceux de la capture d'écran, plus grands et colorés. Les styles ont été générés sur www.cssbuttongenerator.com.

Lancez le programme sur votre Pi et naviguez sur son adresse IP depuis votre téléphone, en ajoutant le numéro de port 8080. Cela vous présente le formulaire, permettant l'envoi de commandes à votre script Python par l'appui sur les boutons. Sur cette capture d'exemple, le téléphone se connecte au Pi sur

<http://192.168.1.69:8080> - l'adresse IP peut être obtenue en exécutant ifconfig en ligne de commande.



```
ledborg.py
# web.py : contrôlez votre Pi depuis un téléphone intelligent
# avec LedBorg (www.piborg.com) comme exemple
# ColinD 27/12/2012

import web
from web import form
```



VERSION PYTHON : 2.7.3rc2
VERSION PYGAME : 1.9.2a0
O.S. : Debian 7

TESTED!

```

# Définit les pages (index) du site
urls = ('/', 'index')
render = web.template.render('templates')

app = web.application(urls, globals())

# Définit les boutons à afficher sur le formulaire
my_form = form.Form(
    form.Button("btn", id="btnR", value="R", html="Rouge", class_="btnRed"),
    form.Button("btn", id="btnG", value="G", html="Vert", class_="btnGreen"),
    form.Button("btn", id="btnO", value="0", html="-Eteint-", class_="btnOff")
)

# Définit ce qui se passe quand la page d'index est appelée
class index:
    # GET est utilisée quand la page est demandée
    def GET(self):
        form = my_form()
        return render.index(form, "Contrôle distant du Raspberry Pi en
Python")

    # POST est appelée quand un formulaire est envoyé
    def POST(self):
        # lit les données envoyées depuis le formulaire
        userData = web.input()

        # Détermine quelle couleur LedBorg doit afficher
        if userData.btn == "R":
            print "ROUGE"
            lbColour = "200" #Rgb
        elif userData.btn == "G":
            print "VERT"
            lbColour = "020" # rGb
        elif userData.btn == "0":
            lbColour = "000"
            print "Eteint LedBorg"
        else:
            print "Ne fait rien - il doit se passer quelque chose..."

        # Écrit la couleur dans LedBorg (voir www.piborg.com)
        LedBorg = open('/dev/ledborg', 'w')
        LedBorg.write(lbColour)
        print lbColour
        del LedBorg

        # Recharge le formulaire pour la prochaine saisie
        raise web.seeother('/')

# run
if __name__ == '__main__':
    app.run()

```

templates/index.html

```

$def with (form, title)
<!doctype html>
<html>
  <head>
    <title>${title}</title>
    <link rel="stylesheet" type="text/css" href="/static/styles.css">
  </head>
  <body>
    <br />
    <form class="form" method="post">
      ${form.render()}
    </form>
  </body>
</html>

```

Essayez d'ajouter un bouton Hasard pour afficher n'importe laquelle des 27 couleurs représentées sur l'image en bas de la page précédente. Web.py gère plusieurs autres éléments de formulaires comme les listes déroulantes ou les cases à cocher. Les détails complets sont sur www.webpy.org.

L'heure des questions et des réactions

Vous faites de l'excellent travail en rendant possible le magazine le plus utile que j'aie jamais lu ! Je n'ai que 14 ans et votre mag m'a inspiré pour démarrer la programmation. Merci!

J Forster

Super travail de vous tous pour les numéros de 2012. Je suis un de vos 600+ supporters avec le Kickstarter pour en obtenir une copie physique. Je me demandais si vous pensez faire le même type de levée de fonds en 2013 - ou offrir un service d'abonnement par l'intermédiaire de votre site Web?

T Giles

[**Ed:** Oui, nous prévoyons un service d'abonnement. Nous publierons bientôt une enquête sur les différentes options]

Félicitations pour le magazine ! En tant qu'utilisateur RPi d'un certain "âge" c'est bien de "bricoler" à nouveau avec un matériel pas cher, relativement simple, relativement puissant, et open source !

P Welsh

Votre publication est remarquable! Les articles variés et bien composés que chaque numéro de MagPi contient sont extrêmement utiles. S'il vous plaît continuez ce bon travail !

T Gomez

Je suis heureux que vous sortiez une copie papier. J'ai encore à trouver un système sur lequel lire les pdf, cela vous rend encore plus impressionnant. Vous serez mis en valeur sur l'étagère à côté de mes copies papiers de Linux Format.

Yakko TDI

Merci pour cet incroyable magazine.

O Bellés

Je suis quelqu'un qui s'est fait les dents en informatique sur un Sinclair MK14, et je pense que le Pi est incroyable. J'ai lu la plupart des numéros de MagPi au moins deux fois et je suis très impressionné par l'ampleur et la profondeur des articles que vous avez publiés. En tant que parent, je suis très impressionné de voir que la plupart des articles publiés ne sont pas relatifs aux jeux. S'il vous plaît continuez votre bon travail les gars !

J Ainhirn-Williams

Je suis un de ceux qui peuvent se souvenir des premiers magazines d'informatique personnelle et à quel point ils nourrissaient nos compétences en programmation tout en agissant comme un catalyseur essentiel de l'enthousiasme créatif émergent dans les écoles. Je me revois tapant, ligne après ligne, du code sur un ZX81 pour lire à la fin "Ram Pack Déconnecté" et tout recommencer. Ensuite, il y a eu les listings informatiques corrigés automatiquement et justifiés ! Mais les magazines informatiques comme le MagPi sont vraiment un ingrédient essentiel pour permettre aux jeunes programmeurs de démarrer avant de parvenir à réaliser de plus grandes choses. Alors continuez à dispenser la bonne parole.

D Lockwood



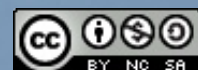
The **MagPi**™

editor@themagpi.com

The MagPi est une marque déposée de The MagPi Ltd. Raspberry Pi est une marque déposée de la fondation Raspberry Pi. Le magazine MagPi est réalisé collaborativement par un groupe indépendant de propriétaires de Raspberry Pi, et n'est en aucun cas affilié à la fondation Raspberry Pi. Il est interdit de reproduire ce magazine dans un but commercial sans l'autorisation de The MagPi Ltd. L'impression dans un objectif non commercial est autorisée conformément à la licence Creative Commons ci-dessous. Le MagPi n'est ni propriétaire ni responsable des contenus ou opinions exprimés dans les articles de cette édition. Tous les articles ont été vérifiés et testés avant la date de sortie mais des erreurs peuvent subsister. Le lecteur est responsable de toutes les conséquences, tant logicielles que matérielles, pouvant survenir suite à la mise en pratique de conseils ou de code imprimé. Le MagPi ne prétend pas détenir de droits d'auteur et tout le contenu des articles est publié sous la responsabilité de l'auteur de l'article.

Ce travail est placé sous les termes de la licence Creative Commons Attribution - Pas d'Utilisation Commerciale - Partage dans les Mêmes Conditions 3.0. Une copie de cette licence est visible sur :

<http://creativecommons.org/licenses/by-nc-sa/3.0/>



ou envoyez un courrier à Creative Commons, 444 Castro Street, Suite 900, Mountain View, California, 94041, USA.