N°13 | JUILLET - AOÛT 2016

HACKABLE MAGAZINE

DÉMONTEZ | COMPRENEZ | ADAPTEZ | PARTAGEZ

France MÉTRO. : 7,90 € – CH : 13 CHF – BEL/LUX/PORT.CONT : 8,90 € – DOM/TOM : 8,50 € – CAN : 14 $ CAD

ACTUALITÉ

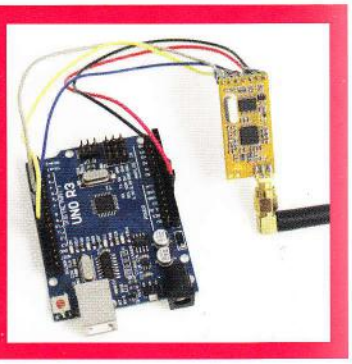
Découvrez la nouvelle Raspberry Pi 3 : plus puissante, plus connectée, même prix !

p. 10

RADIO

Connectez Arduino et Raspberry Pi avec le module de communication radio 433Mhz APC220

p. 18



ÉVÈNEMENT

Retour en images sur la troisième édition de la Maker Faire Paris

p. 04

REPÈRES

Apprenez à déchiffrer et comprendre une documentation technique ou datasheet

p. 86

Découvrez et utilisez la...

CAMÉRA RASPBERRY PI !

p. 32

- Configurez la caméra et votre Pi
- Capturez des images fixes
- Streamez la vidéo
- Créez des time-lapses
- Exploitez les fonctions de vision nocturne (IR)

LUMIÈRES

Créez une barre à leds multicolore animée à partir d'une réglette lumineuse à leds

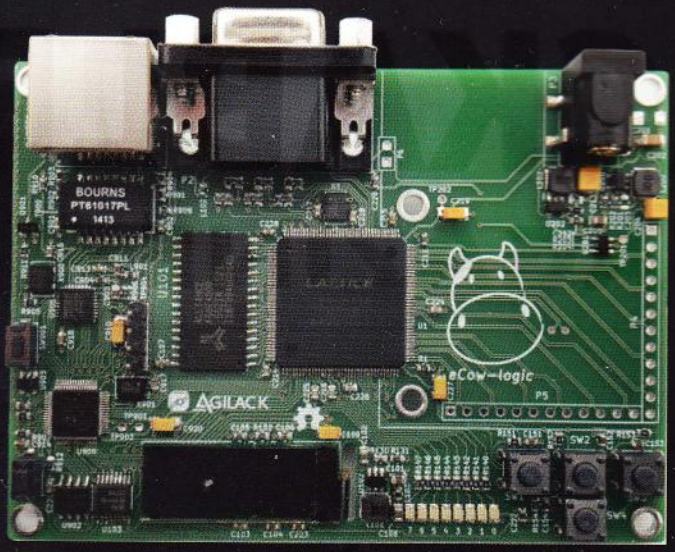
p. 74

IMPRESSION

Utilisez une mini imprimante thermique avec votre Raspberry Pi grâce à Python

p. 64

L 19338 - 13 - F: 7,90 € - RD



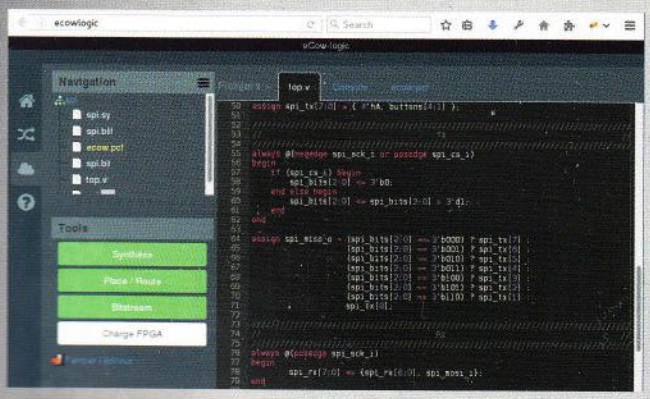
eCowLogic

Développer sur FPGA
n'a jamais été aussi simple

Simple ?

eCowLogic ne nécessite ni sonde de programmation, ni driver, ni même l'installation d'un logiciel, un navigateur web suffit !
A l'heure où tous les objets communiquent osons les outils connectés

Serveur web embarqué



Chargement du FPGA depuis l'interface embarquée,
Communication SPI accessible depuis un formulaire,
Mise à jour du firmware ... depuis l'interface web :)

Un environnement de développement intégré est proposé. Celui-ci s'appuie sur des outils Saas pour la synthèse et le routage du FPGA.

Spécifications

- FPGA iCE40-HX-1K (compatible Yosys/Arachne-pnr)
- Contrôleur Ethernet avec MCU et stack TCP/IP intégrés
- Écran OLED 128x32 pixels
- Mémoire flash 16 Mbits (2 Mo)
- Mémoire SRAM 4 Mbits (512 ko)
- 8 LEDs et 4 boutons poussoirs
- Un port VGA (piloté directement par le FPGA)
- Un port d'extension avec 22 IOs disponibles
- Alimentation supportant de 5V à 15V en entrée

Libre !

- Schémas et routage de la carte réalisés avec Kicad,
- Sources du firmware GPL,
- Tous les exemples fournis testés avec Yosys,
- Documentation sous Creative Commons,

github
Agilack/eCow-logic

irc
freenode
#cow-lab

AGILACK

www.ecowlogic.fr

ÉDITO

Quelque chose ne tourne pas rond...

Alors même que le projet de loi américain sur le droit à obtenir des informations techniques concernant la réparation des équipements électroniques (le *Fair Repair Act* ou *Right to Repair bill*) est sur le point de passer à la trappe (après un an d'enlisement), de mon côté je me penche sur le Sony Xperia Z de ma fille, dont l'écran a subi une déplorable et destructrice rencontre avec le manche d'une masse (oui, oui).

L'achat de la pièce de remplacement, toute la façade avant du smartphone avec écran et une partie de la connectique, n'était pas un problème pour peu que l'on sache identifier le modèle avec exactitude. Ce qui l'était en revanche est la réparation à proprement dite, s'étalant sur deux bonnes heures. La raison ? Une absence d'informations fiables et surtout... de la colle partout.

En effet, tout est littéralement collé à l'adhésif double face dans l'équipement : la coque, l'écran, les câbles, l'accu, les caches... Il n'y a que 4 malheureuses vis dans le smartphone et le risque d'endommager le matériel en tentant une réparation est très important, même en étant très minutieux et bien équipé (pistolet à air chaud, outils de précision, ventouse, etc.).

Peut-être ne l'avez-vous pas remarqué, mais cela fait quelque temps déjà que les fabricants de smartphones ou d'ordinateurs portables semblent utiliser différentes techniques limitant les tentatives de réparation : batterie dans l'équipement et non amovible, utilisation de colles, de résines, vis non standards, connecteurs soudés... À croire qu'ils préféreraient que vous achetiez un nouvel appareil plutôt que de le réparer ou le faire réparer par quelqu'un (*sic*).

Il n'en a pourtant pas toujours été ainsi et les plus vieux s'en souviennent. Fut un temps un ordinateur ou même n'importe quel équipement était accompagné d'une notice incluant le schéma de fonctionnement, un exposé du produit, la liste des pièces détachées et une procédure d'analyse de problèmes et de réparation (ceci existe encore, tantôt, pour les équipements d'atelier type perceuse à colonne).

Certes, cela augmenterait le coût de production des produits, mais entre ces deux extrêmes il faut une juste mesure. Lorsqu'on achète un produit, en principe, il nous appartient et nous devons donc être en mesure de pouvoir y faire des modifications et des réparations sans avoir à subir de restrictions ou de blocages. Je regrette amèrement l'achat de ce smartphone presque irréparable malgré un dénouement heureux après beaucoup d'efforts et de patience.

Je pensais craquer prochainement pour un CAT S60 avec caméra thermique FLIR intégrée, mais la mention « Lithium Ion, non amovible, 3800mAh » va, je pense, me faire passer mon tour et rester avec mon Samsung E1200 à 14,90€. Désolé les gars, acheter un téléphone endurci à 650€ intégrant un élément difficilement remplaçable dont le vieillissement est connu, prévu et avéré ne m'intéresse pas vraiment...

Denis Bodor

Hackable Magazine
est édité par Les Éditions Diamond

LES ÉDITIONS DIAMOND
STIMULANT DE MATIÈRE GRISE

10, Place de la Cathédrale - 68000 Colmar
Tél. : 03 67 10 00 20 – Fax : 03 67 10 00 21
E-mail : lecteurs@hackable.fr
Service commercial : cial@ed-diamond.com
Sites : www.ed-diamond.com
Directeur de publication : Arnaud Metzler
Rédacteur en chef : Denis Bodor
Réalisation graphique : Kathrin Scali
Responsable publicité : Valérie Fréchard,
Tél. : 03 67 10 00 27 v.frechard@ed-diamond.com
Service abonnement : Tél. : 03 67 10 00 20

Impression : pva, Landau, Allemagne
Distribution France : (uniquement pour les dépositaires de presse)
MLP Réassort : Plate-forme de Saint-Barthélemy-d'Anjou. Tél. : 02 41 27 53 12
Plate-forme de Saint-Quentin-Fallavier.
Tél. : 04 74 82 63 04
Service des ventes : Abomarque : 09 53 15 21 77
IMPRIMÉ en Allemagne - PRINTED in Germany
Dépôt légal : À parution,
N° ISSN : 2427-4631
Commission paritaire : K92470
Périodicité : bimestriel
Prix de vente : 7,90 €

MIXTE
Papier issu de sources responsables
FSC® C015136

La rédaction n'est pas responsable des textes, illustrations et photos qui lui sont communiqués par leurs auteurs. La reproduction totale ou partielle des articles publiés dans Hackable Magazine est interdite sans accord écrit de la société Les Éditions Diamond. Sauf accord particulier, les manuscrits, photos et dessins adressés à Hackable Magazine, publiés ou non, ne sont ni rendus, ni renvoyés. Les indications de prix et d'adresses figurant dans les pages rédactionnelles sont données à titre d'information, sans aucun but publicitaire. Toutes les marques citées dans ce numéro sont déposées par leur propriétaire respectif. Tous les logos représentés dans le magazine sont la propriété de leur ayant droit respectif.

Suivez-nous sur Twitter
@hackablemag

À PROPOS DE HACKABLE...

HACKS, HACKERS & HACKABLE

Ce magazine ne traite pas de piratage. Un **hack** est une solution rapide et bricolée pour régler un problème, tantôt élégante, tantôt brouillonne, mais systématiquement créative. Les personnes utilisant ce type de techniques sont appelées **hackers**, quel que soit le domaine technologique. C'est un abus de langage médiatisé que de confondre « pirate informatique » et « hacker ». Le nom de ce magazine a été choisi pour refléter cette notion de **bidouillage créatif** sur la base d'un terme utilisé dans sa définition légitime, véritable et historique.

SOMMAIRE

ACTUALITÉS

04
Retour sur la Maker Faire Paris 2016 : ça mature

10
Raspberry Pi 3 : la petite nouvelle Pi pleine de ressources

ARDU'N'CO

18
Utilisez des modules radio APC220 pour faire communiquer votre Arduino

EN COUVERTURE

32
Installez et utilisez le module caméra V2 avec votre Raspberry Pi

50
Module caméra Raspberry Pi : un mot sur l'infrarouge

EMBARQUÉ & INFORMATIQUE

64
Utilisez une imprimante thermique avec votre Raspberry Pi

DÉMONTAGE, HACKS & RÉCUP

74
Convertissez une réglette lumineuse en afficheur multicolore

REPÈRES & SCIENCE

86
Apprenez à déchiffrer une datasheet

ABONNEMENT

41/42
Abonnements tous supports

63
Offres spéciales professionnels

RETOUR SUR LA MAKER FAIRE PARIS 2016 : ÇA MATURE

Denis Bodor

Comme pour le bon vin, le fromage et bien d'autres choses, au fil des années, les évènements, salons et expositions comme les Maker Faire évoluent et se bonifient avec le temps. Du 30 avril au 1er mai dernier a eu lieu la Maker Faire Paris, au cœur de la Foire de Paris, porte de Versailles. Si vous n'avez pas pu y venir, voici un petit retour en images sur ce que vous avez raté...

Cette troisième édition de la Maker Faire Paris s'est déroulée, comme l'année précédente, dans un pavillon de la Foire de Paris, ce qui est, je trouve, on ne peut plus adapté. En effet, la Foire de Paris accueille depuis 1929 le Concours Lépine, une épreuve mettant en compétition des inventeurs venant présenter leurs créations. Ce concours a primé des inventions devenues depuis des objets du quotidien comme le stylo à bille, l'aspirateur, le fer à repasser à vapeur ou encore les lentilles de contact.

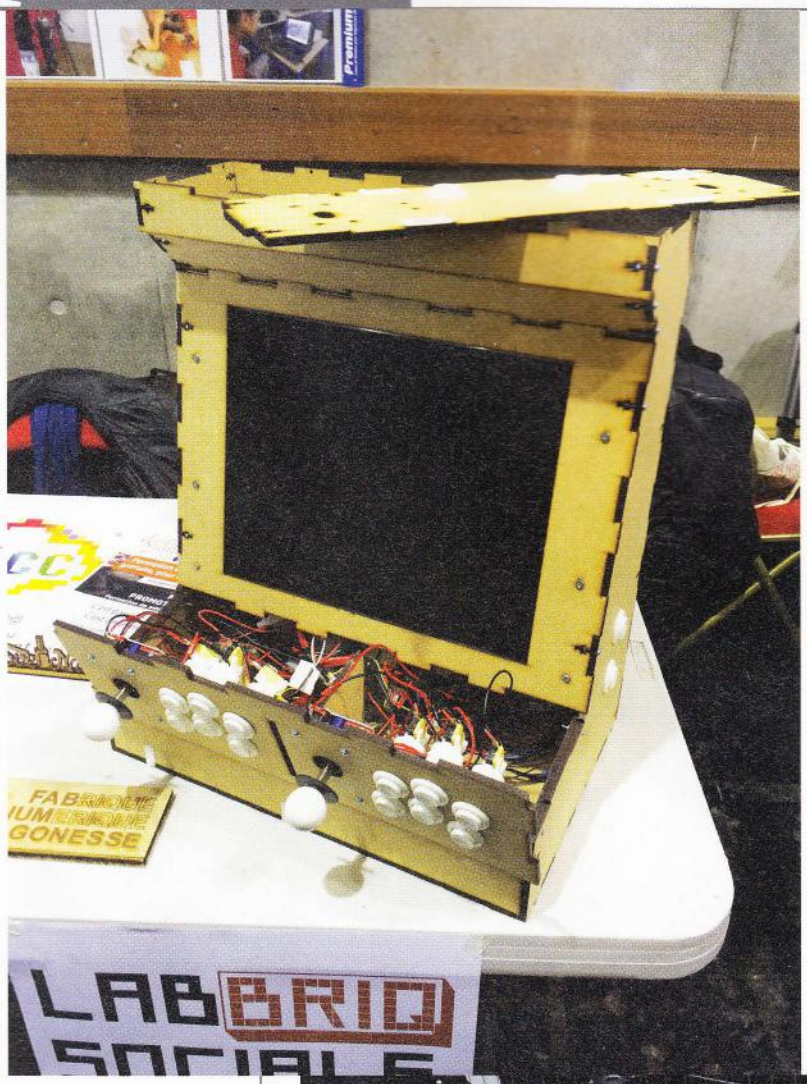
Bien sûr, le Concours Lépine ne s'est pas tenu pour autant au sein de la Maker Faire, mais on ne peut que relever le fait que, des inventeurs, il y en avait plein des stands et plein des allées !

Cette édition s'est déroulée sur deux jours et a connu un nombre record de visiteurs avec un chiffre annoncé de 65000 personnes, soit 30000 visiteurs de plus que l'année précédente. Un chiffre tout autant incroyable que très rassurant, car c'est là un clair indicateur que l'envie d'innover, de créer, de découvrir et d'expérimenter est bel et bien là, et ne cesse de s'étendre.

Du point de vue d'un « tenancier de stand », il était intéressant de remarquer que les visiteurs des deux journées étaient très différents. Le samedi était principalement le jour des connaisseurs, des personnes ayant déjà des affinités avérées avec la thématique de la manifestation.

Il s'agissait davantage de « confrères *makers* » que de « visiteurs » au sens strict du terme. Venus pour trouver des idées, échanger des informations ou simplement « prendre la température », ils ont pu trouver dans les allées et sur les stands des personnes à la sensibilité similaire pour partager des connaissances.

Le dimanche, en revanche, était bien plus teinté de découverte et d'émerveillement. Les personnes s'étant rendues à la Foire de Paris, exposition massive couvrant bien des domaines, ont pu découvrir toute l'étendue



du mouvement *maker* et des créations de gens qui, comme eux, pensent, imaginent et réfléchissent, mais ont choisi d'oser sauter le pas et se lancer dans la mise en pratique de leurs idées.

Nul doute que parmi ces visiteurs curieux, certains ont alors découvert qu'il n'était pas si difficile, coûteux ou risqué de passer de l'autre côté du miroir et, à leur tour, de devenir créateur et non plus simplement utilisateur. Petit bémol cependant, comme l'année dernière, bon nombre de curieux auraient craqué immédiatement pour une mise en pratique et l'acquisition d'une carte comme Arduino ou Raspberry Pi... mais personne n'en proposait à la vente. Dommage.

Mais c'est pourtant bien là tout l'intérêt d'une telle manifestation : titiller la curiosité, susciter l'envie de créer et rassurer quant à la facilité de sauter le pas. La plupart des stands étaient tenus par des personnes venues présenter leurs créations, non tant dans l'optique d'une démonstration de compétences, mais par envie de partager leur passion et leurs motivations : fourmi mécanique des « Machines de l'île », bobine de Tesla d'Arc'Ethic, expériences impressionnantes du youtubeur ExperimentBoy, fantastiques costumes de Angel Light,

tables Tetris d'Arbalet project... La liste n'en finit pas et il est tout bonnement impossible d'être exhaustif.

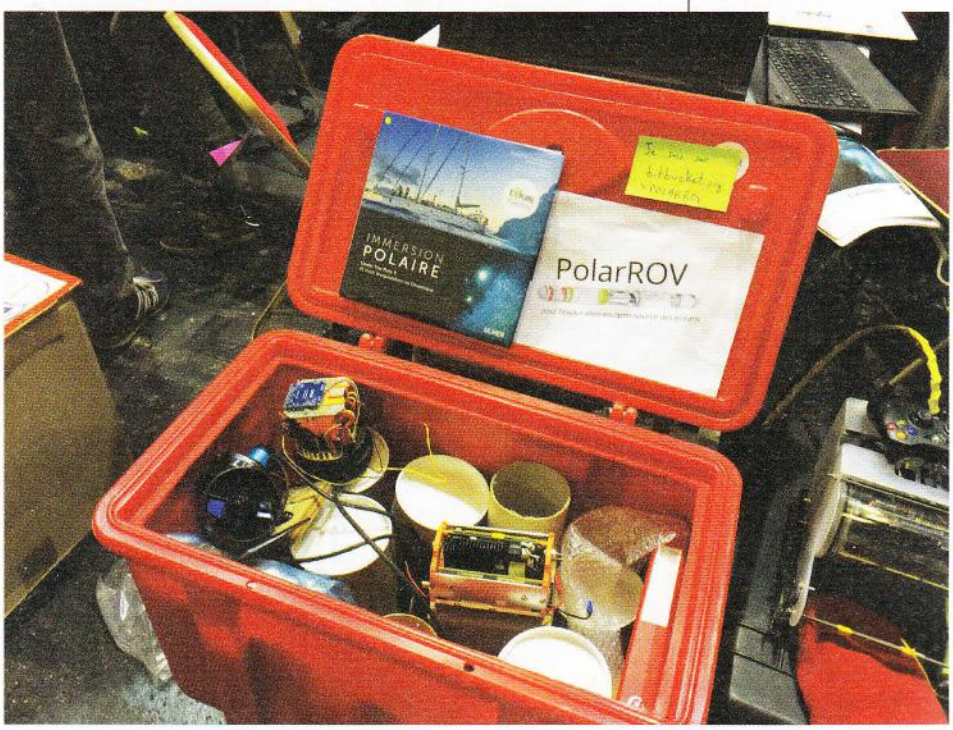
J'ai été rédacteur en chef de *GNU/Linux Magazine* pendant plus de 15 ans et surtout depuis sa création en 1998, quelques années à peine après la création de GNU/Linux. Cette énergie qui baignait la troisième Maker Faire parisienne, comme les précédentes, je la connais bien et je l'ai immédiatement reconnue. C'est un mélange d'envie de savoir, de partage et d'ouverture, un cocktail

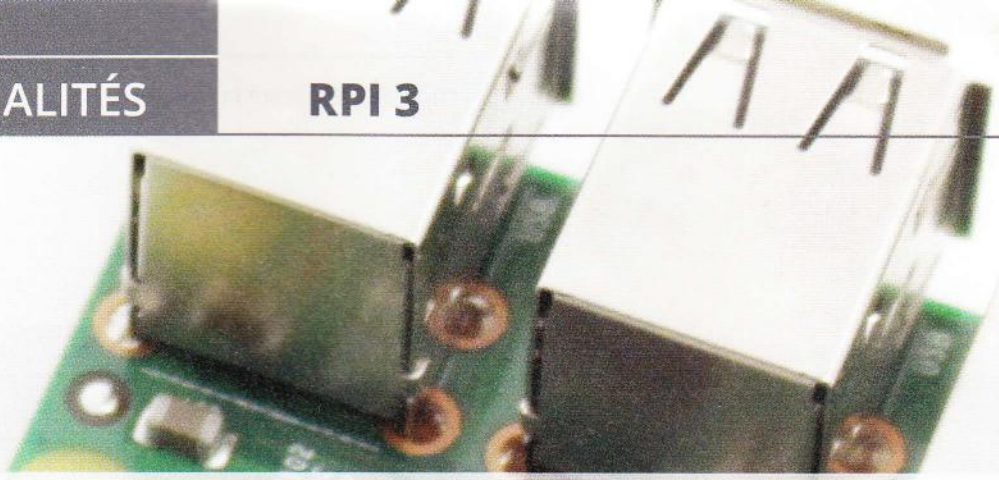
vitaminé poussant à la création et au désir de maîtriser son environnement. À mon sens, l'un des moteurs de la civilisation, ni plus, ni moins.

Au risque de prêcher des convertis, je dirai qu'il est très rassurant de voir que cette énergie est toujours là et gagne en puissance d'année en année. Dans un monde massivement tourné vers le « prêt à consommer » et le « prêt à jeter » l'idée tenace, et sans doute déplaisante pour certains, qu'il soit possible de créer ce qui nous est utile ou plaisant, semble agréablement inaliénable. Mieux encore, c'est aussi la volonté de modifier ce qui nous est vendu

« clé en main » pour l'adapter à nos besoins et non nous conformer à un usage prédéterminé et borné. La créativité est loin d'être morte, la Maker Faire peut être littéralement vue comme sa célébration et c'est avec autant de plaisir que de fierté qu'on se trouve être partie prenante d'un tel élan.

Un grand merci aux organisateurs de cette édition, aux Makers présents et aux visiteurs, et que dire, si ce n'est vivement l'année prochaine pour une édition encore plus sensationnelle et motivante ! **DB**

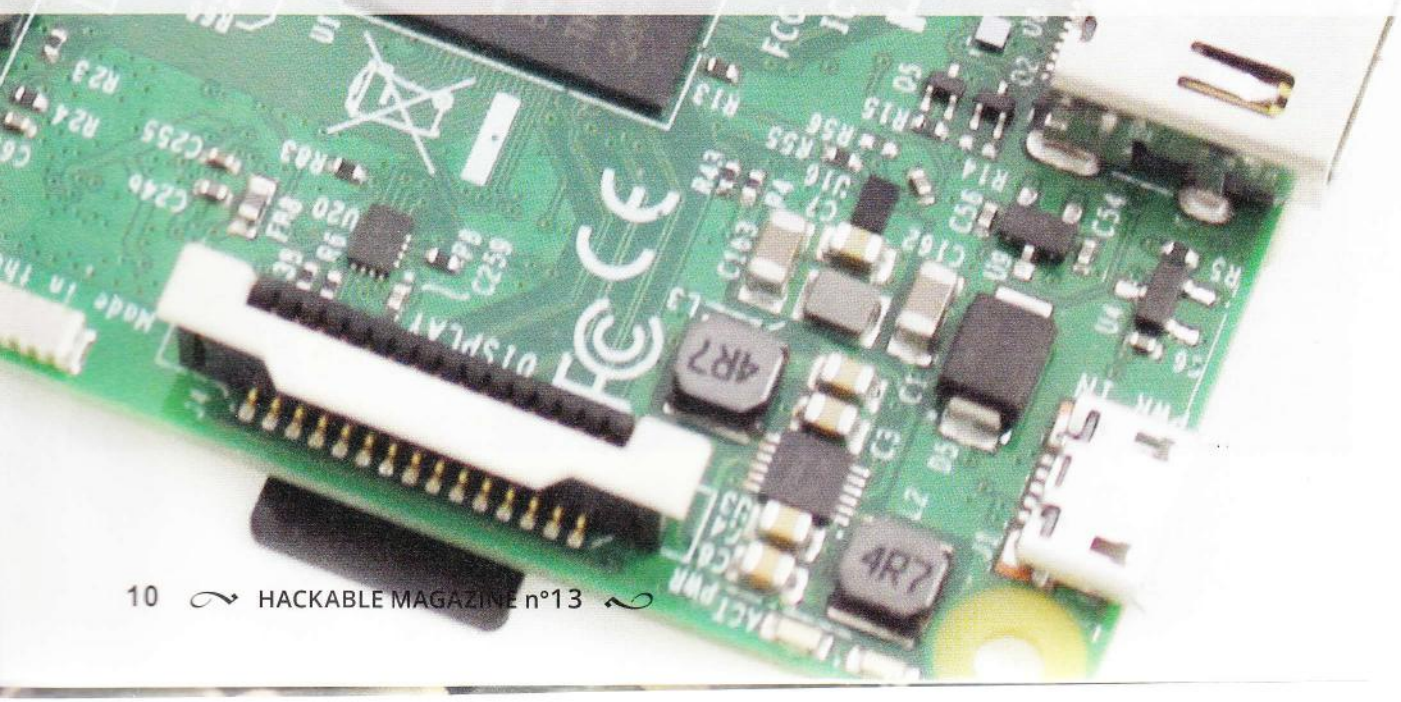


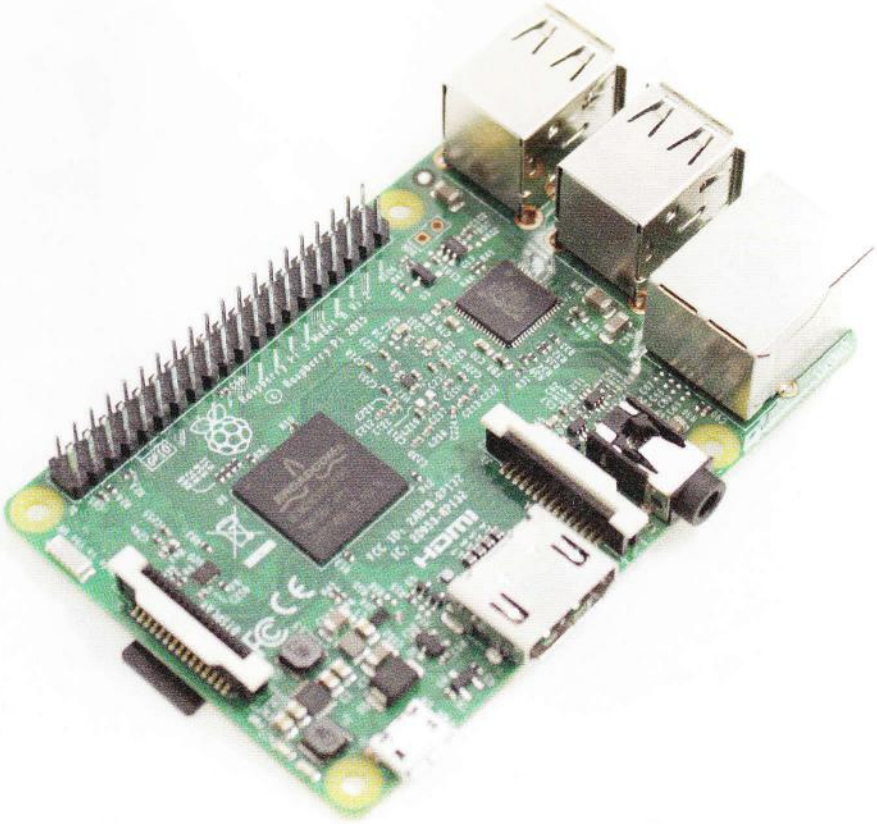


RASPBERRY PI 3 : LA PETITE NOUVELLE PI PLEINE DE RESSOURCES

Denis Bodor

Fin février dernier, la fondation Raspberry Pi annonçait la disponibilité d'une nouvelle carte, par l'intermédiaire d'un billet d'Eben Upton sur le blog officiel. Cette plateforme, disponible au même prix que la Raspberry Pi 2, intègre un grand nombre d'évolutions et de changements, à commencer par des performances annoncées 10 fois supérieures à celles d'une RPi 1. Faisons ensemble un point sur ce qui change et ce qui reste identique...





Voici la petite nouvelle Raspberry Pi 3 qui visuellement est très similaire aux modèles précédents. Mais ne vous y trompez pas, c'est un gros bond en avant tant en termes de puissance qu'en ce qui concerne les options de connectivité.

Je ne vous cache pas que ma première réaction face à cette annonce était pour le moins mitigée et se résumait à quelque chose comme « Quoi ? Encore une nouvelle Pi ? ». Il est vrai qu'on peut s'interroger sur la stratégie mise en œuvre par la fondation, un constat partagé entre la course à la puissance et une tentative de saturer le marché de cartes parfum framboise, face à d'autres « fruits » (oranges, bananes, etc.). Ceci n'est pas sans rappeler une autre plateforme, la CubieBoard, dont l'aventure a débuté par une unique carte et dont les versions et évolutions successives se sont ensuite enchaînées à une cadence infernale, jusqu'à ne plus savoir où donner de la tête...

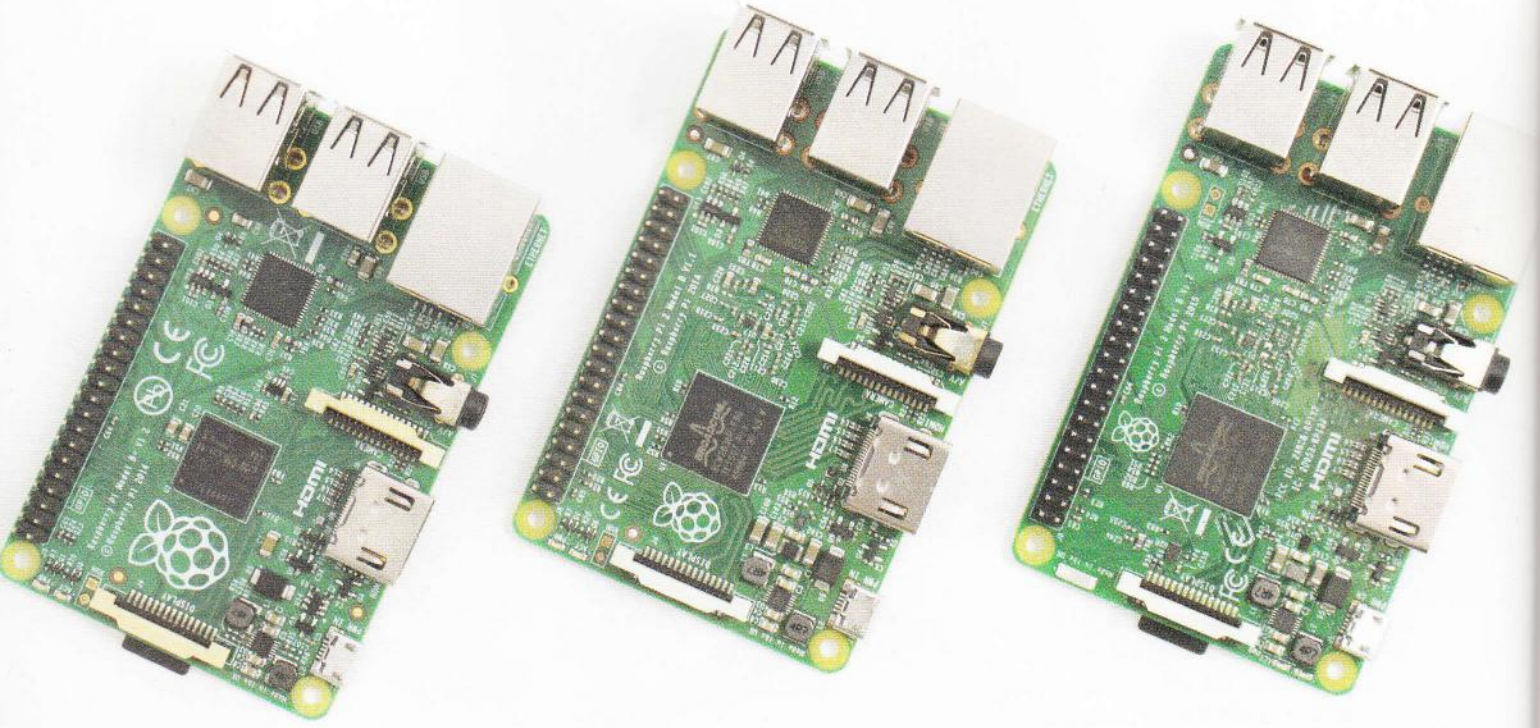
À cette date, la famille/galaxie Raspberry Pi se décline donc en quelques 7 modèles (sans compter les *Compute Modules*) :

- Raspberry Pi modèle B (2012),
- Raspberry Pi modèle A (2013),
- Raspberry Pi modèle A+ (2013),
- Raspberry Pi modèle B+ (2014),
- Raspberry Pi Zero (2015),
- Raspberry Pi 2 modèle B (2015),
- Raspberry Pi 3 modèle B (2016).

Bien sûr, les modèles à privilégier pour de nouveaux projets se résument aux modèles 1B+, 2B et maintenant 3B, ou encore Zero pour des besoins spécifiques en termes d'encombrement limité ou de budget réduit.

La Raspberry Pi 3 est physiquement très similaire aux précédents modèles B/B+ avec un certain nombre de choses qui reste inchangées :

- même connecteur HDMI et jack que le modèle B+ et 2B,
- 4 connecteurs USB,
- 1 port Ethernet 10/100 (dommage, pas de Gigabit Ethernet),
- connecteurs pour écran (DSI) et caméra (CSI) sur la carte (mêmes emplacements),
- mêmes dimensions et trous de fixation (et donc mêmes boîtiers compatibles)
- même connecteur 40 broches pour les GPIO,
- même connecteur d'alimentation micro USB,
- même emplacement pour carte microSD (ou presque, puisqu'il s'agit d'un simple emplacement enfichable, sans le « petit clic »),



Nous avons là presque trois années d'évolution de la Raspberry Pi avec à gauche le modèle B+, au centre la 2 et à droite la petite nouvelle. Depuis l'arrivée du modèle B+, la forme reste identique, mais la puissance ne cesse d'augmenter... alors que la framboise sérigraphiée sur la carte ne cesse de se réduire...

- mêmes principales distributions GNU/Linux utilisables (Raspbian, etc.).

Mais comme vous allez pouvoir le constater, la carte est cependant une nouveauté qui sort du lot...

1. RASPBERRY PI 3 : PROCESSEUR

Le principal changement avec cette nouvelle carte concerne sans l'ombre d'un doute le passage au SoC (*System-on-Chip*) Broadcom BCM2837. La première Pi utilisait un BCM2835 (ARM1176JZF à 700 Mhz) et la 2 un BCM2836 construit autour d'un processeur ARM Cortex-A7 quadricœur à 900 MHz. Nous avions d'ailleurs consacré un article aux intérêts de cette évolution « cardiaque » dans le numéro 7 du magazine, mettant en avant un gain modeste en cas d'utilisation d'applications incapables de tirer parti du multicœur (*multithreading*).

Le BCM2837 est également un SoC utilisant un processeur ARM à 4 cœurs, mais celui-ci est cadencé à 1,2 Ghz et, surtout, il est construit autour d'un ARM Cortex-A53 pouvant fonctionner en 64 bits. Attention, il est ici important de relativiser cette information, car bien que le processeur puisse fonctionner en mode 64 bits, c'est en réalité en 32 bits (tout comme avec les autres modèles) que le système fonctionnera encore un certain temps. À l'heure actuelle, les performances sont d'ores et déjà bien supérieures aux modèles précédents en raison de l'architecture même sur SoC et de la fréquence utilisée.

Il ne faut en aucun cas voir le mode 64 bits comme une panacée ou une solution miracle pour gagner en puissance. Techniquement et dans les grandes lignes, un processeur fonctionnant en 64 bits traite simplement les valeurs de plus de 32 bits (`int32_t`) en une fois (`int64_t`) alors qu'un processeur 32 bits le fera en deux étapes. Le second

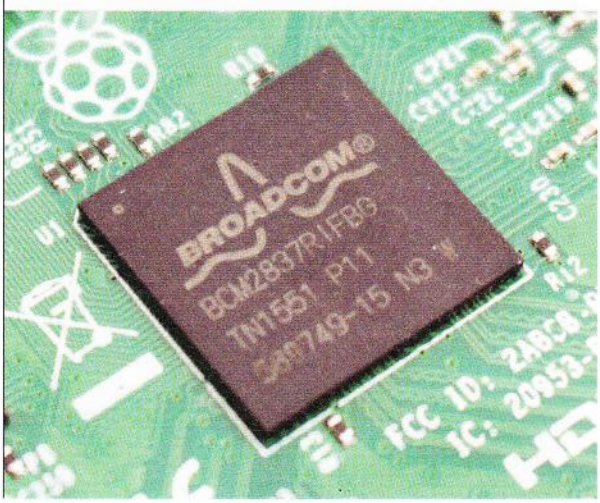
Le dessous de la carte accueille la mémoire d'1 Go exactement comme la RPi 2 alors que le modèle B+ voyait ce composant placé sur le SoC. Notez également l'emplacement à souder près du trou de fixation en haut à gauche correspondant au signal « RUN » permettant de réveiller le système. Les précédents modèles de Raspberry Pi voyaient cet emplacement placé à côté du connecteur pour l'écran DSI.

avantage du 64 bits est de pouvoir adresser, en un seul bloc, 4 Go de mémoire, chose totalement inutile sur une Pi qui n'en possède que 1 Go.

Mais les choses ne s'arrêtent pas là. Pour utiliser le mode 64 bits, il faut un noyau Linux 64 bits pour architecture ARM, ainsi que des applications également compilées en 64 bits. Ceci amène au même dilemme que sur PC avec des systèmes 32 ou 64 bits : on gagne certes en rapidité dans certains cas particuliers, mais en contrepartie il devient nécessaire de maintenir une version supplémentaire du système et de ses applications. En d'autres termes, dans le cas de la Raspberry Pi 3, il faudrait maintenir en parallèle une distribution Raspbian en 32 et une autre en 64 bits pour des bénéfices qui restent à démontrer. La fondation n'a pour l'instant pas encore annoncé de plan pour une telle démarche étant donné qu'un seul système, 32 bits, fonctionne parfaitement sur toutes les Raspberry Pi existantes.

Ne vous laissez donc pas berner par des annonces de certains sites d'actualités technologiques n'hésitant pas à clamer haut et fort des fausses vérités du type « Le Raspberry Pi 3 passe au 64 bits ! ».

Au centre de la Raspberry Pi 3, le SoC Broadcom BCM2837 cadencé à 1,2 Ghz intègre un processeur ARM Cortex-A53 à quatre cœurs pouvant fonctionner en 64 bits. Pour l'heure, le système est toujours en 32 bits, mais le gain en puissance n'en est pas moins important...



Il n'en reste pas moins que le processeur de cette nouvelle framboise est effectivement bien plus rapide comme le démontre un rapide test avec l'outil **sysbench** :

```
$ sysbench --test=cpu --num-threads=4 \
> --cpu-max-prime=2000 run
sysbench 0.4.12:  multi-threaded system evaluation benchmark
Running the test with following options:
Number of threads: 4
[...]
Test execution summary:
    total time:                          4.9187s
    total number of events:              10000
    total time taken by event execution: 19.6593
```

Cette même commande sur une Raspberry Pi 2 affiche un résultat de 11,9082 secondes, contre 54,4539 secondes sur une Raspberry Pi B+ (mono-cœur).

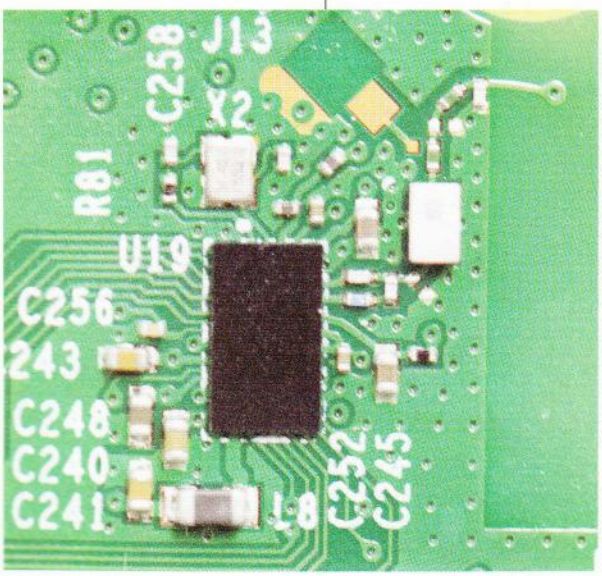
Mais cette montée en puissance a également un coût sous la forme d'une consommation énergétique plus importante et d'une montée en température du SoC. Différentes mesures effectuées à la fois par la fondation et différents internautes montrent clairement qu'il est fortement recommandé de :

- 1. Fournir à la carte une alimentation 5V de bonne qualité capable de délivrer 2,5 ampères de façon stable ;
- 2. Équiper le SoC Broadcom d'un dissipateur thermique (radiateur) afin de faciliter le refroidissement en cas de forte utilisation des ressources.

2. LE GROS PLUS : WIRELESS

Si l'on considère que le gain de puissance est une évolution naturelle de la plateforme au fil du temps (comme pour les PC), il est une nouveauté qui elle reste une avancée considérable pour les utilisateurs : l'intégration d'une interface Wifi et d'une connectivité Bluetooth.

En effet, cette nouvelle carte ne demande plus d'insertion d'une clé USB pour profiter du Wifi, mais intègre un contrôleur Broadcom BCM43438 interfacé avec le SoC de façon transparente pour l'utilisateur (il ne s'agit pas d'un périphérique USB comme l'interface Ethernet). Ce composant, qui se trouve sous la carte assure les fonctions de connectivité Wifi 802.11n et Bluetooth 4.1 (avec *Bluetooth Low Energy* alias BLE). On notera qu'il semble intégrer également un récepteur FM qui est désactivé, mais qui d'après le descriptif du composant est contrôlable par l'interface HCI Bluetooth (peut-être un hack possible).

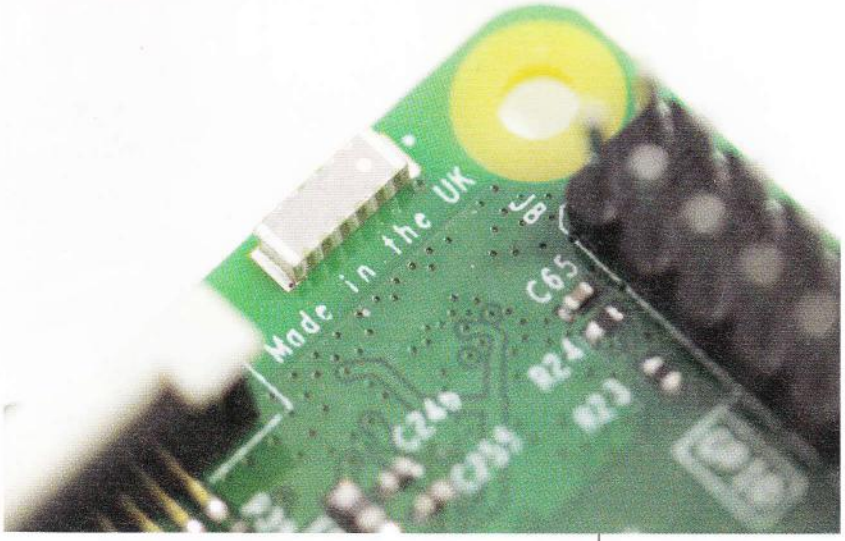


La grande nouveauté de cette Raspberry Pi 3 est le contrôleur BCM43438 fournissant une interface Wifi 802.11n intégrée ainsi qu'une connectivité Bluetooth 4.1 avec BLE (Bluetooth Low Energy). Notez qu'il ne s'agit pas d'un périphérique USB comme c'est le cas pour l'Ethernet, mais d'une connexion directe au SoC (SDIO et série).

Notez que ce composant est interfacé en SDIO pour la partie Wifi et en connexion série pour l'interface Bluetooth. Ceci peut poser des problèmes dans certains cas particuliers comme le choix d'un pilote différent pour le contrôleur SD/MMC ou une tentative d'utilisation du second port série (**ttyS0**/**uart1** est maintenant la console série et **ttyAMA0**/**uart0** est l'interface Bluetooth).

Ce changement de ports provoque d'ailleurs un problème en cas de tentative d'utilisation de la console série. C'est une vieille habitude pour ma part d'utiliser cette solution pour initialement configurer toutes mes cartes et systèmes embarqués, et ainsi ne pas avoir à jouer avec des connexions d'écrans HDMI et de claviers USB. Plusieurs utilisateurs ont ainsi pu remarquer que, par défaut, la connexion à la console série n'est pas en 115200 bps comme normalement spécifiée dans la configuration (**cmdline.txt**), mais en 72000 bps, qui est loin d'être une valeur standard.

Une tentative de configuration en 9600 bps fournit une connectivité en... 6002 bps ! Ce problème est dû à la fois à l'utilisation du second port série pour la console (en lieu et place du premier avec une Pi 2 par exemple), mais également d'une des fréquences contrôlant le fonctionnement du SoC. La vitesse de ce port série est, en effet, calculée en fonction d'une *core frequency* réglée à 400 Mhz (contre 250 Mhz avec les précédentes cartes). Pire encore, cette fréquence peut varier en fonction de celle du système au gré des demandes en ressources et de la température du processeur.



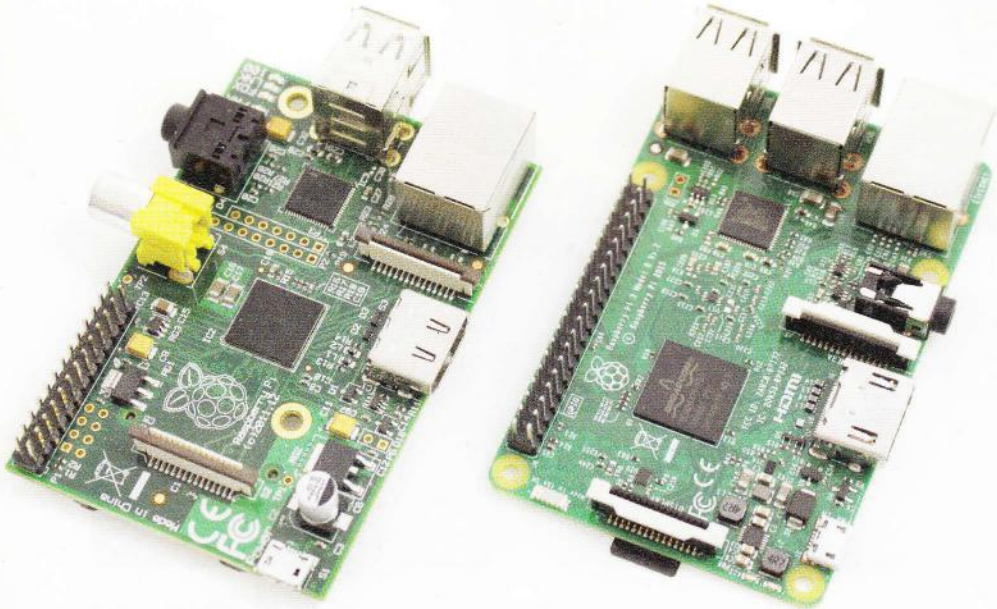
Pour assurer la connectivité Wifi et Bluetooth, la carte est équipée d'une minuscule antenne céramique. Seul regret ici, l'absence de connecteur supplémentaire pour une antenne externe offrant un gain plus important...

À l'heure actuelle, la solution consiste à éditer le fichier **config.txt** sur la partition FAT de la carte micro SD et à ajouter les lignes :

```
force_turbo=1
core_freq=250
arm_freq=1200
```

Ceci aura pour effet de fixer les différentes fréquences et de faire fonctionner la console série en 115200 bps tout en conservant les 1,2 Ghz pour le processeur. Le contrecoup de l'opération est que votre Raspberry Pi 3 sera alors à sa puissance maximum en permanence ce qui impacte sa consommation en énergie. Bien sûr, si contrairement à moi vous ne faites pas usage de la console série, vous pouvez vous passer de cette modification. Ceci peut également être une solution temporaire, le temps de procéder à la configuration du système avant d'y accéder via le réseau (SSH) et donc abandonner la console série. Il est également possible que le problème soit réglé par ailleurs au moment où vous lirez cet article et qu'une simple mise à jour du système suffise.

Quoi qu'il en soit, l'utilisation de l'interface Wifi intégrée se fera exactement de la même manière qu'avec un adaptateur USB. Il en va de même pour les fonctionnalités Bluetooth. Le système Raspbian étant une distribution GNU/Linux Debian presque classique, les mêmes outils seront utilisés sans le moindre problème.



À gauche le tout premier modèle commercialisé de Raspberry Pi et à droite le dernier. Près de 5 ans séparent ces deux cartes et l'évolution est clairement visible.

Notez cependant deux questions revenant régulièrement sur les forums concernant cet adaptateur Wifi intégré :

- Peut-on utiliser la bande des 5 Ghz en plus des 2,4 Ghz ? La réponse est **non**, c'est du 802.11n en 2,4 Ghz uniquement d'après les informations de la fondation (les spécifications précises du BCM43438 ne semblent pas publiques).
- Peut-on utiliser cette interface en mode moniteur pour tester la sécurité des points d'accès Wifi ? **Non**. Les outils `airmon-ng` et `airodump-ng` retournent une erreur, le pilote n'implémente pas RFMON (mais c'est une simple question de temps et de développement de pilotes).

Parmi les autres points négatifs concernant cette intégration, on regrettera l'absence de connecteurs pour une antenne externe (connecteur Hirose U.FL/IPX/UMMCC), même s'il est parfaitement compréhensible de chercher à réduire les coûts de production (la carte coûte seulement 45€).

3. EN RÉSUMÉ

Au final, la carte Raspberry Pi 3 B est la version la plus « musclée » de la famille à ce jour tout en offrant davantage de connectivité de base et donc sans avoir à investir dans des périphériques USB supplémentaires. Ceci rapproche la plateforme d'un environnement de travail de bureau classique tel qu'on en trouve sur PC, mais avec un encombrement fortement réduit.

À cette date, un certain nombre de problèmes restent à corriger (port série, fréquences, pilotes Wifi, refroidissement, etc.), mais à choisir, pour un prix identique, j'aurai tendance à davantage opter pour une RPi 3 plutôt qu'une RPi 2, pour un usage générique. Je pense d'ailleurs que de nombreux utilisateurs feront de même une fois que cette carte sera tout aussi facile d'approvisionnement que les autres modèles (ce qui pourrait bien être le cas au moment où vous lirez ceci).

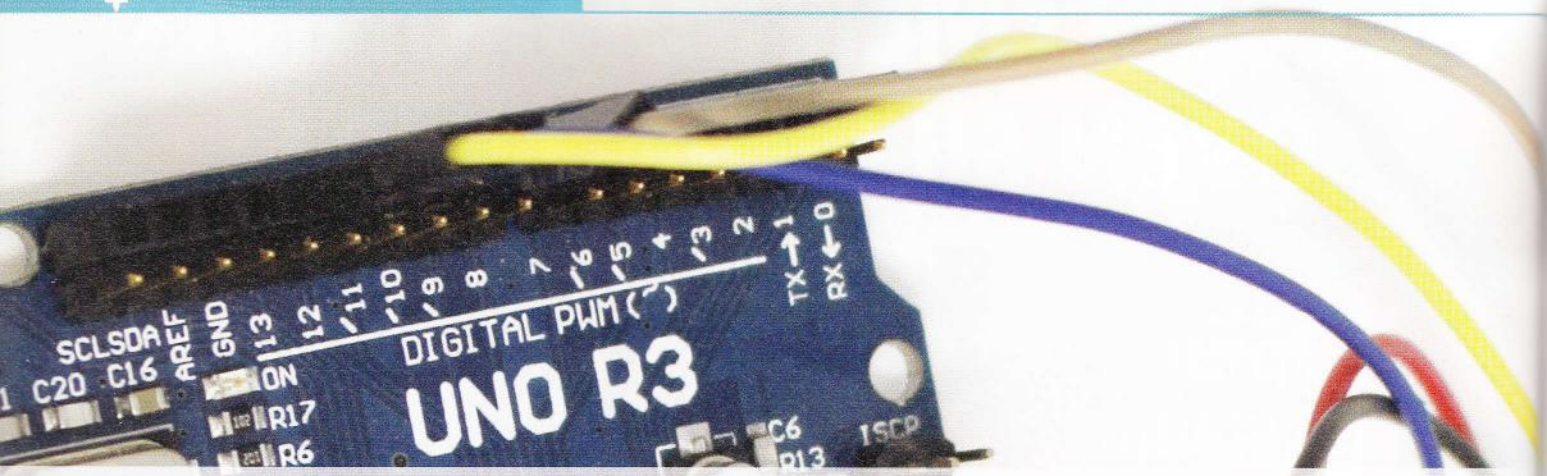
On en arrive donc à se demander quel sera l'avenir d'une partie de cette collection de modèles qui s'étoffe peut-être plus vite que les besoins des utilisateurs. Autant les modèles A+ et Zero ont un intérêt évident pour certains projets, autant la « concurrence » entre RPi 2 et RPi 3 semble inutile (le modèle B+ simple est moins cher et son existence s'explique donc ainsi).

Personnellement donc, je considère qu'en cas de nouvel achat, la Raspberry Pi 3 est le choix par défaut. Je crois que c'est une simple question de temps avant que le précédent modèle ne devienne tout simplement obsolète, car ne disposant pas d'avantages particuliers... **DB**

ACTUELLEMENT DISPONIBLE
OPEN SILICIUM N°19 !

N'ÉCRIVEZ PLUS DE PILOTE LINUX !
DÉCOUVREZ LES MÉTHODES ET SOLUTIONS POUR SUPPORTER VOTRE MATÉRIEL SANS TOUCHER AU NOYAU !

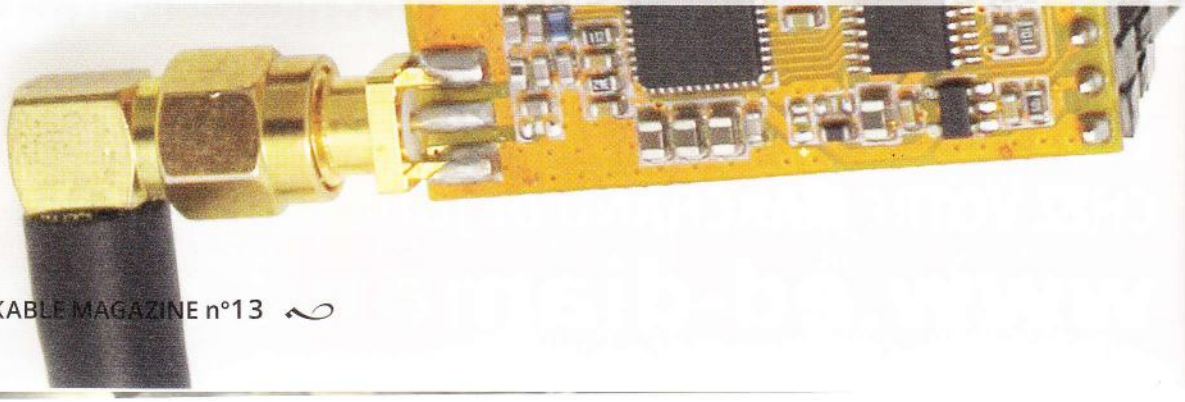
NE LE MANQUEZ PAS
CHEZ VOTRE MARCHAND DE JOURNAUX ET SUR :
www.ed-diamond.com



UTILISEZ DES MODULES RADIO APC220 POUR FAIRE COMMUNIQUER VOTRE ARDUINO

Denis Bodor

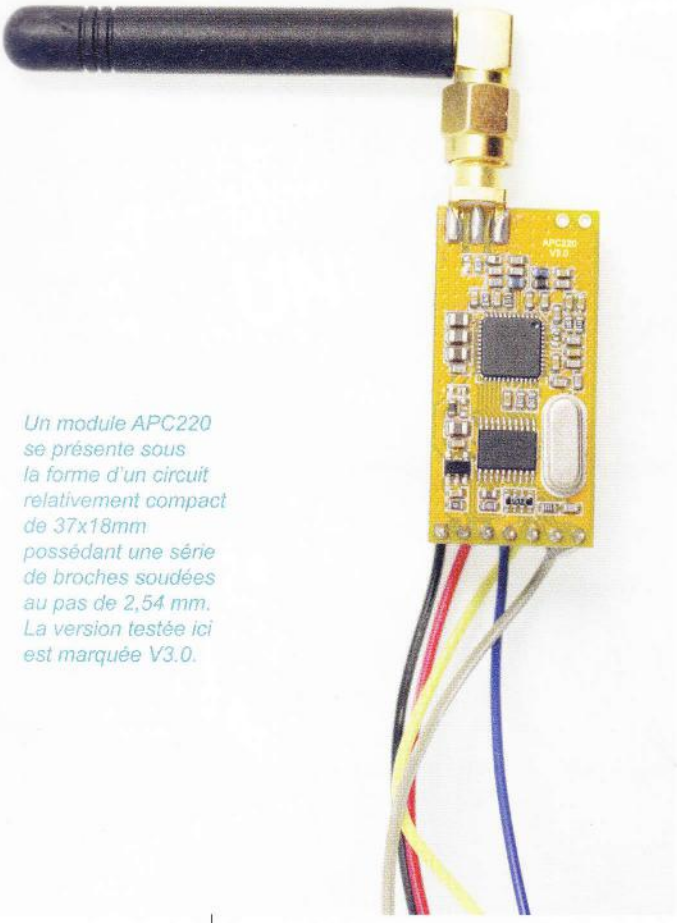
Certains projets demandent parfois de mettre en œuvre une communication sans fil sur de grandes distances (> 500m). Ceci écarte généralement des choses comme le Wifi, ZigBee ou Bluetooth. On est alors tenté de se diriger vers certains modules chinois peu coûteux promettant une méthode simple pour faire communiquer un montage se trouvant à presque un kilomètre. Parfois cela fonctionne et parfois on se retrouve un peu démuni devant le manque de documentation intelligible.



Les modules APC220 sont généralement vendus en paire, équipés d'une antenne avec un connecteur SMA (non RPSMA comme souvent pour le Wifi par exemple) et accompagnés d'un convertisseur USR/série.

Les modules dont nous allons parler ici sont très courants. Il s'agit du modèle APC220 qui est vendu, par exemple par DFRobot, mais également tout un tas d'autres revendeurs via des boutiques et les espaces d'enchères en ligne comme eBay. L'ensemble se présente généralement sous la forme d'une paire de modules assortis d'antennes rigides et d'un adaptateur USB permettant, en principe, la connectivité PC et la configuration des modules. Existant dans plusieurs formes, la plus courante est celle présentée ici en photo, vendue entre 30 et 40 euros le tout, selon les canaux de vente et/ou le port offert ou en supplément.

L'utilisation basique de ce type de modules est extrêmement aisée. Proposant par défaut un mode de communication série à 4800 bps ou 9600 bps, il suffit, en principe, d'utiliser un des modules via l'adaptateur USB d'un côté et un module seul connecté à une carte Arduino par exemple, de l'autre. Cependant, le premier problème qui se pose est d'ordre légal. En effet, ces modules peuvent être configurés pour utiliser une fréquence entre 433 Mhz et 478 Mhz, avec une puissance jusqu'à 20 mW réglable sous la forme de 10 niveaux distincts. Or, en France, la bande ISM ne nécessitant pas de licence d'utilisation, s'étend de 433,05 Mhz à 434,79 et ce avec une puissance apparente rayonnée (PAR) de 10 mW. La configuration par défaut des modules est 431 Mhz et 20 mW (même s'il ne s'agit pas de la PAR) les interdisant donc directement d'utilisation sur notre territoire (ceci n'est pas systématique, je suis également tombé sur des modules préconfigurés en 434 Mhz). Et là, vous n'avez que trois solutions :



Un module APC220 se présente sous la forme d'un circuit relativement compact de 37x18mm possédant une série de broches soudées au pas de 2,54 mm. La version testée ici est marquée V3.0.

- les mettre dans un tiroir en attendant que la règlementation change un jour ou l'autre,
- commencer à potasser vos cours pour passer votre licence radioamateur (cela prend du temps, mais ouvre la porte à bien des opportunités),
- reconfigurer les modules pour les faire entrer bien gentiment dans le rang.

Il est cependant important de relever ici que ces recommandations n'ont absolument aucun caractère juridique valable. Vous seul êtes responsable des matériels que vous utilisez selon la législation en vigueur dans votre pays. C'est à vous de vous conformer à la législation locale. De plus, et même si cela peut paraître évident pour certains, on parle ici de matériels et de modules bas de gamme et non de produits testés, vérifiés et certifiés. Ces petites choses sont fabriquées à la chaîne en Asie par centaines de milliers et revendues sous bien des marques via bien des canaux. Les spécifications données par le détaillant ou le soi-disant fabricant sont à prendre pour ce qu'elles sont : des informations théoriques. Strictement rien ne vous assure à 100% du fonctionnement borné des modules et donc du respect des fréquences et de la puissance utilisée. Souvent, les composants utilisés sur ce type de produits n'ont pas une source clairement identifiée, il peut donc s'agir de composants recyclés n'étant absolument plus dans les spécifications d'origine.

1. RECONFIGURATION DES MODULES

Chaque module présente une interface de 7 broches permettant à la fois l'utilisation et la configuration. L'adaptateur USB livré n'est qu'un convertisseur USB/série dit TTL. En effet, les niveaux de tension utilisés sont 0 et 5 volts (éventuellement 3,3 volts avec un adaptateur différent) alors que dans le cas de ce qu'on appelle un « port série » (RS-232) il s'agit de +/- 12 volts. Ce type de ports de 9 ou 25 broches n'existe plus sur les PC actuels. Ainsi, l'adaptateur USB livré avec les modules peut à la fois être remplacé par un autre d'un modèle différent, mais peut également servir à bien d'autres choses (comme connecter le port série, logiciel ou non, d'une carte Arduino à un PC).

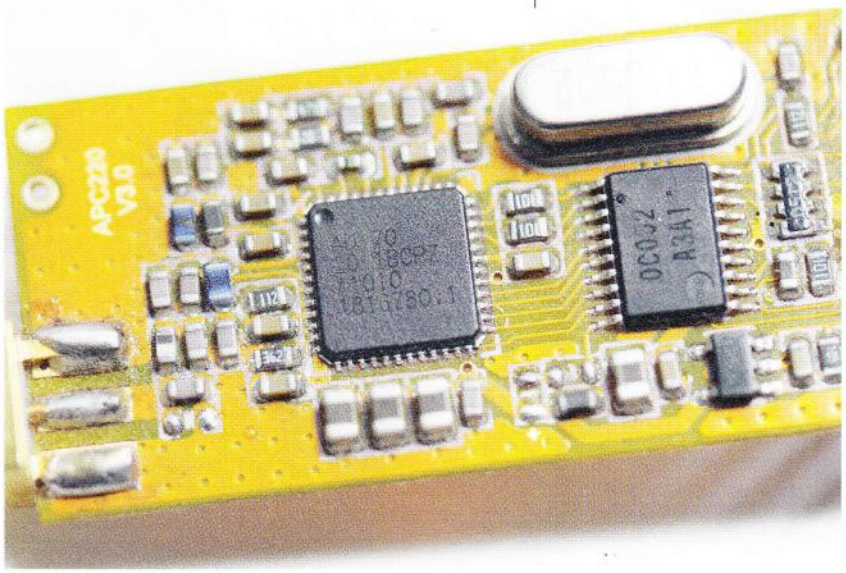
Chaque module utilise le brochage suivant qui est généralement sérigraphié sur le circuit :

- 1, GND : la masse ;
- 2, Vcc : la broche d'alimentation, généralement +5V, mais peut être +3,3V ;

- 3, EN : la broche d'activation qui fonctionne à l'inverse de ce qui se fait généralement. Si cette broche est laissée en l'air ou reliée à une tension de plus de 1,6 volts, le module est actif. Si elle est reliée à une tension inférieure à 0,5 volts (typiquement à la masse) le module passe en sommeil et devient inactif, n'émettant plus ;
- 4, RxD : la ligne de réception des données servant au module pour collecter les données à transmettre ;
- 5, TxD : la ligne d'émission où apparaissent les données reçues par les airs ;
- 6, Aux : cette broche n'est pas clairement décrite dans la pseudo-documentation des modules. Elle est donnée comme étant à la masse en cas de réception et à la tension d'alimentation (Vcc) en cas de transmission. Dans d'autres documentations, elle est donnée comme non utilisée ou réservée pour de futurs usages ;
- 7, Set : la ligne qui, si elle est connectée à la masse (ou laissée en l'air) passe le module en mode « paramétrage » et non en « communication ». On la placera souvent à la tension d'alimentation (Vcc) pour une utilisation courante du module. En reliant cette broche à une sortie d'un Arduino, il est possible de faire en sorte que ce dernier puisse configurer le module et l'utiliser à souhait.

Pour configurer le module ou simplement vérifier la configuration, la documentation recommande d'utiliser l'adaptateur USB accompagnant la paire de transmetteurs et un utilitaire Windows appelé *Rf-Magic*. Le nom est déjà pour le moins étrange, mais en constatant que l'archive téléchargeable s'appelle `rfmagic.rar` et contient un simple fichier `APC22X_V12A.exe`, c'est plus qu'il n'en faut pour me dissuader totalement de le lancer (mais j'ai peut-être de vieux préjugés concernant les programmes fournis en archives RAR). Windows est un système suffisamment vulnérable et prompt à accueillir des malwares de toutes sortes sans qu'on ait besoin de tenter le diable en exécutant des programmes sans documentation, sans sources, sans origine traçable et sans même un simple fichier `README.TXT`.

L'outil proposé utilise la communication série avec le module à configurer via l'adaptateur USB tout en utilisant sa broche SET pour activer le mode « configuration ». Il est donc parfaitement possible de faire de même sans avoir à reposer sur un programme douteux, il suffit de décrypter la documentation...



Au cœur du module se trouve le circuit intégré Analog Device ADF7020, un transceiver (émetteur/récepteur) pour la bande ISM utilisant les modulations FSK et ASK. De fait, ce circuit permet de faire bien plus que ce que propose le module qui l'utilise, mais l'ensemble des fonctionnalités n'est malheureusement pas accessible. La simplicité a un coût...

2. CONFIGURATION DU MODULE AVEC UN CROQUIS ARDUINO

En mode configuration, un transmetteur peut prendre en compte des commandes permettant soit de lire la configuration soit de la définir. Celle-ci restera enregistrée même si le module n'est plus alimenté. Ces commandes utilisent des caractères ASCII et sont au nombre de deux : **"RD"** pour lire la configuration et **"RW"** pour en enregistrer une nouvelle.

La communication en mode configuration se passe toujours en 9600 bps, quelle que soit la vitesse définie pour le fonctionnement en transmission. Pour lire la configuration, tout ce que nous avons à faire est d'envoyer les caractères **"R"**, **"D"**, 13 et 10. Ces deux derniers correspondent respectivement au retour chariot CR (*Carriage Return*) et au saut de ligne LF (*Line Feed*). Ces deux caractères spéciaux sont un héritage ancestral venant du fond des âges de l'informatique et permettaient initialement de faire revenir le chariot d'une imprimante en buté (à gauche) et de faire défiler le papier d'une ligne (comme sur une machine à écrire mécanique).

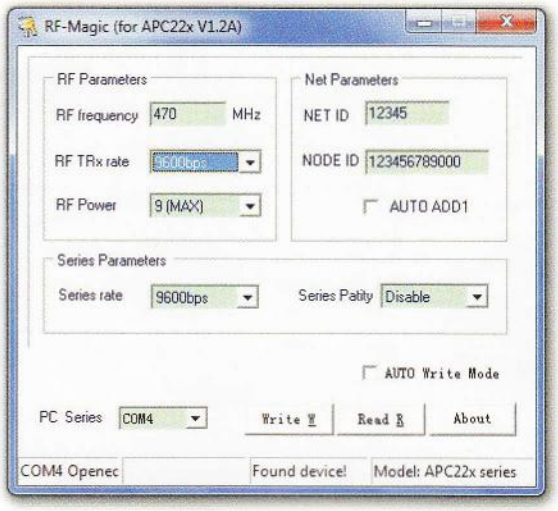
La réponse du module sera composée d'une ligne également terminée par les caractères CR et LF, sous la forme : **"PARA"**, espace, fréquence en hertz (6 caractères), espace, vitesse radio (1 caractère), espace, puissance (1 caractère), espace, vitesse série (1 caractère), espace, parité (1 caractère).

Exemple : **"PARA 434000 2 5 1 0"**, décrivant l'utilisation d'une fréquence de 434 Mhz en 4800 bps, 5/9 de la puissance, et une communication série filaire en 2400 bps sans parité.

La documentation détaille la signification des 4 valeurs codées sur un caractère :

- débit radio : entre 1 et 4 correspondant à 2400, 4800, 9600 et 19200 bps,
- puissance : entre 0 et 9, 9 correspondant théoriquement à 13 dBm ou 20 mW,
- débit série : entre 0 et 6 correspondant à 1200, 2400, 4800, 9600, 19200, 38400 et 57600 bps,
- parité : 0 pour aucune, 1 pour parité paire (*odd*) et 2 pour parité impaire (*even*). Le bit de parité est une façon de vérifier l'intégrité des données transmises en ajoutant un bit de contrôle pouvant prendre une valeur de 1 ou 0 en fonction de la valeur des bits qui le précèdent. Les bits à 1 sont comptés et si le résultat est pair ou impair, le bit de contrôle sera à 1 ou 0. Une parité paire ou impaire définit le sens donné au bit calculé.

À présent que nous connaissons la signification de chaque valeur, il nous est possible de rapidement créer un croquis Arduino pour lire et décoder la configuration actuelle des modules :



Une application Windows est mise à disposition sur le Web afin de permettre de configurer les modules APC220. Disponible sans documentation, sans source et sans origine vraiment claire... Le plus sage est de tout simplement de s'en passer et de préférer une configuration manuelle via un croquis Arduino maison.

```
#include <SoftwareSerial.h>

#define brocheRX 12
#define brocheTX 11
#define brocheSET 8

// nouveau port série
SoftwareSerial apc220(brocheRX,brocheTX);

// fonction de décodage des vitesses
void printspeed(unsigned char data) {
  switch(data) {
    case '0':
      Serial.println("1200 bps"); break;
    case '1':
      Serial.println("2400 bps"); break;
    case '2':
      Serial.println("4800 bps"); break;
    case '3':
      Serial.println("9600 bps"); break;
    case '4':
      Serial.println("19200 bps"); break;
    case '5':
      Serial.println("38400 bps"); break;
    case '6':
      Serial.println("57600 bps"); break;
    default:
      Serial.println("???? bps");
  }
}

// configuration
void setup() {
  // objet pour stocker la réponse du module
  String chaine;

  pinMode(brocheSET, OUTPUT);
  apc220.begin(9600);
  Serial.begin(115200);

  // activation du mode configuration
  digitalWrite(brocheSET, LOW);
  delay(10);
  // envoi de la commande
  apc220.println("RD");
  delay(10);
```

```
  // lecture de la réponse
  if(apc220.available()) {
    // on lit jusqu'au caractère CR
    // (on ignore LF pour ne pas stocker CR dans la chaîne)
    chaine = apc220.readStringUntil('\r');
  }

  // affichage des informations
  Serial.print("Configuration : ");
  Serial.println(chaine);

  Serial.print("Vitesse radio: ");
  printspeed(chaine.charAt(12));

  Serial.print("Vitesse UART: ");
  printspeed(chaine.charAt(16));

  Serial.print("Parite: ");
  switch(chaine.charAt(18)) {
    case '0':
      Serial.println("aucune"); break;
    case '1':
      Serial.println("impaire"); break;
    case '2':
      Serial.println("pair"); break;
    default:
      Serial.println("????"); break;
  }
}

void loop() {
}
```

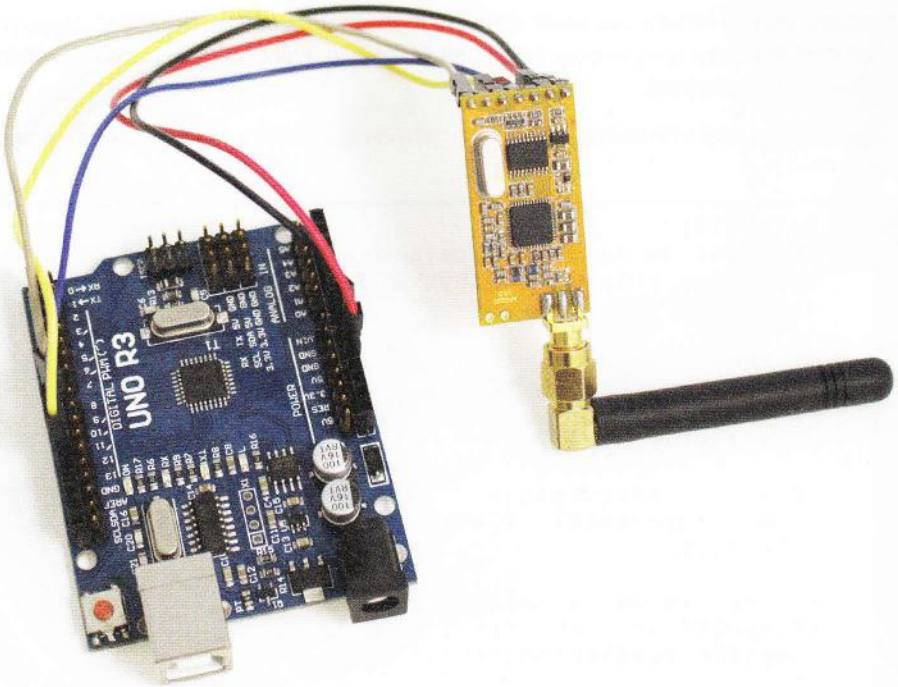
Arduino

La connexion Arduino/module est relativement simple : la broche TX du module sur 12, RX sur 11, SET sur 8, les masses connectées entre elles et VCC sur +5V. Nous utilisons ici la bibliothèque SoftwareSerial nous permettant, sur une carte UNO par exemple, d'ajouter un port série entièrement logiciel et ainsi conserver l'accès au moniteur série dans l'IDE Arduino.

Notre objet représentant ce port série logiciel est appelé `apc220` et nous pouvons l'utiliser en 9600 bps pour envoyer et recevoir des données du module radio.

Après avoir passé la broche SET à la masse pour activer le monde configuration, nous envoyons simplement la chaîne « RD » avec `println()` qui se chargera d'ajouter les caractères CR et LF en fin de ligne. Dès lors, si le module fonctionne et est bien dans le bon mode, celui-ci va répondre en nous fournissant la série de caractères attendue.

Nous les utilisons alors pour confectionner une chaîne de caractères (type `String`) appelée `chaine`



La connexion entre le module APC220 et une carte Arduino est très simple et se limite à trois broches (deux après configuration). Nous choisissons ici d'utiliser un port série logiciel sur la carte, mais il est bien entendu possible de faire usage de Serial si l'on peut se passer de moniteur série.

et pouvons alors utiliser les méthodes propres à ce type d'objet pour repérer les caractères qui nous intéressent et afficher des informations en fonction de leur valeur. La chaîne retournée par le module a une taille fixe et la position de chaque caractère est constante. L'utilisation de `String` n'est pas la seule solution, nous aurions tout aussi bien plus utiliser un tableau de caractères de la même façon et avec le même résultat. Le type `String` cependant offre des facilités et la mémoire occupée pour cela n'est pas réellement un problème : c'est un croquis à usage unique pour ainsi dire...

Sur les modules en ma possession, la configuration par défaut ainsi affichée sur le moniteur série est la suivante :

```
Configuration : PARA 434000 2 9 1 0
Vitesse radio: 4800 bps
Vitesse UART: 2400 bps
Parite: aucune
```

Notez la différence de vitesse entre la communication radio et la liaison filaire. La documentation précise qu'une vitesse identique ou une situation comme celle-ci est à préférer. En effet, si la communication filaire est plus rapide que l'émission de données par radio, vous allez remplir rapidement la mémoire tampon du module qui ne fait que 256 octets. Vous devrez donc prendre en compte, dans votre projet final, une gestion de délais afin de vous assurer de ne pas faire entrer plus de données dans le tampon que celles qui en sortent et sont transmises. Dans le cas contraire, le tampon débordera et un comportement aléatoire en découlera : les données seront corrompues.

Ici, nous constatons que le module utilise une puissance qui n'est pas adaptée. `9` correspond théoriquement à 20mW et nous allons donc corriger cela.

Pour envoyer une nouvelle configuration, il nous suffit d'utiliser la même syntaxe que celle que nous

recevons pour la lecture des paramètres, tout en ignorant la chaîne "PARA". Nous pouvons donc modifier notre croquis pour configurer le module avant de relire la configuration pour vérifier l'enregistrement.

Ainsi, juste après l'initialisation avec `apc220.begin(9600)`, nous ajoutons les lignes suivantes :

```
delay(10);
// envoi de la configuration
apc220.println("WR 434000 2 5 1 0");
delay(10);

// mode normal
pinMode(brocheSET, LOW);
// pause
delay(50);

// mode configuration
pinMode(brocheSET, HIGH);
delay(10);

// lecture de la réponse du module
if(apc220.available()) {
  apc220.readStringUntil('\n');
}
```

Le fait d'envoyer une configuration provoque une réponse de la part du module qui nous répète les paramètres comme si nous avions utilisé la commande « RD ». Nous choisissons ici d'ignorer cette réponse en lisant les caractères reçus, mais en ne les stockant nulle part.

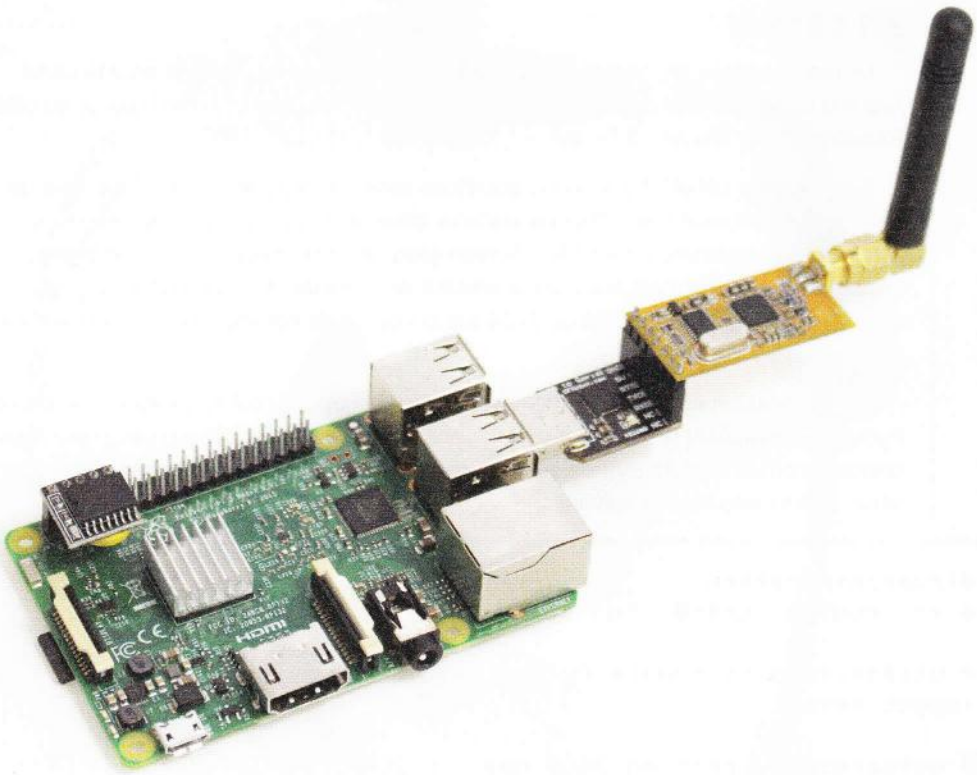
Notez également qu'après l'envoi de la commande, nous basculons en mode normal puis revenons au mode configuration. Il semblerait qu'en l'absence de basculement, le module ne réponde plus correctement, alors que la documentation n'en fait pas mention.

Avec l'intégration de ces quelques lignes, nous obtenons donc un croquis de configuration nous permettant de régler les modules selon nos besoins. Nous pouvons ensuite les utiliser pour établir une connexion entre, par exemple la carte Arduino et une Raspberry Pi.

3. COMMUNICATIONS ENTRE ARDUINO ET RASPBERRY PI

Notre assemblage pour établir un lien entre l'Arduino et la Raspberry Pi consistera à utiliser le module APC220 sur Arduino exactement comme nous l'avons fait précédemment, avec la bibliothèque SoftwareSerial et en utilisant les mêmes broches. Bien entendu, SET, connecté à la broche 8 ne nous est plus utile et nous pouvons tout simplement laisser cela « en l'air » ou le relier à la tension d'alimentation +5V.

Côté Pi, nous pourrions faire usage des broches, mais ceci implique une alimentation du module en +3,3V qui, semble-t-il, selon certains utilisateurs réduit drastiquement la portée du module. En alimentant le module en +5V, nous rencontrons alors un problème de compatibilité puisque les broches (GPIO) d'une Raspberry Pi ne sont pas tolérantes au 5V. Il faudrait donc en principe utiliser un convertisseur de tensions ou éventuellement utiliser des diodes pour



L'utilisation la plus simple de l'APC220 avec une Raspberry Pi consiste à le connecter via l'adaptateur USB/Série qui l'accompagne. Ceci nous évitera à la fois de nous priver de console série pour l'utilisation de la ligne de commandes et de devoir jouer avec les niveaux de tension dans le cas de la connexion via les GPIO.

réduire la tension sur la broche TX du module. Le tout, bien sûr, en se privant par la même occasion de la console série qui pourrait nous être utile.

La solution la plus aisée est alors de tout simplement utiliser l'adaptateur USB livré avec les modules APC220. Celui-ci n'est, en effet, pas uniquement destiné à la configuration des modules et consiste en un adaptateur USR/Série FTDI FT232RL tout à fait standard si ce n'est que la ligne RTS (le *Request To Send* du contrôle de flux matériel RTS/CTS) de la liaison série est utilisée pour contrôler la broche SET du module. Ceci nous permet de disposer d'un port USB supplémentaire sur la Pi et la connexion du duo adaptateur+module est directement détectée et prise en charge par le système :

```
$ dmesg
[...]
ftdi_sio 1-1.4:1.0: FTDI USB Serial Device converter detected
usb 1-1.4: Detected FT232RL
usb 1-1.4: FTDI USB Serial Device converter now attached to ttyUSB0
[...]
```

Nous avons donc ainsi accès au port `/dev/ttyUSB0` pour communiquer via les ondes.

3.1 Côté Pi

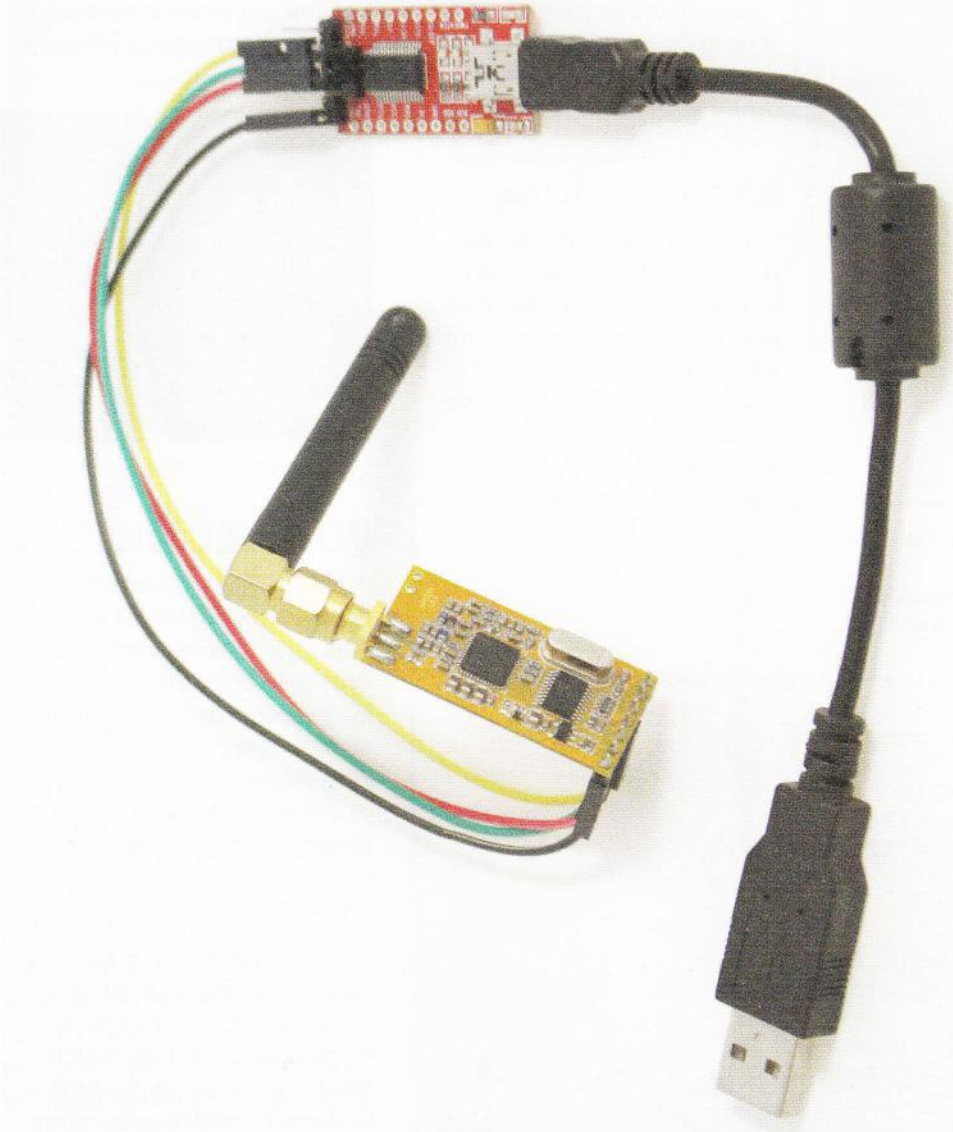
Le test que nous allons faire est relativement simple puisqu'il s'agit pour la carte Arduino d'envoyer une chaîne de caractères. Celle-ci sera alors réceptionnée du côté Raspberry Pi et affichée à l'écran.

Pour réaliser cet affichage, nous pourrions utiliser n'importe quel outil en ligne de commandes pouvant faire office de terminal série. **minicom** est un exemple, mais j'ai une nette préférence pour GNU Screen (paquet **screen**) qui est un multiplexeur de terminal/console, mais aussi un émulateur de terminal série s'il est lancé ainsi : **screen /dev/ttyUSB0 2400** où 2400 est la vitesse de communication à utiliser sur le port **/dev/ttyUSB0**.

Mais l'occasion est trop belle de faire un peu de code et mettre en œuvre un peu de Python. En installant le paquet **python-serial** sur votre distribution Raspbian, vous disposerez d'un module Python vous permettant de très facilement accéder aux ports série. Notre code sera le suivant :

```
#!/usr/bin/python
# -*- coding: utf-8 -*-

# utilisation du module Python Serial
import serial

# ouverture du port en 2400 bps
ser = serial.Serial('/dev/ttyUSB0', 2400, timeout=4)

# Astuce pour intercepter un Ctrl+C
try:
  # boucle infinie
  while 1:
    # lecture des données ligne par ligne
    line = ser.readline()
    # suppression du CRLF
    line = line.replace('\r\n', '')
    # affichage de la ligne reçue
    print line
# capture du Ctrl+c et sortie de boucle
except KeyboardInterrupt:
  print "CTRL+C !"

# fermeture du port série
print "Fermeture"
ser.close()
```

Ce code, enregistré par exemple dans un fichier **reception.py** pourra se lancer avec **python reception.py** ou voir ces attributs changés avec **chmod +x reception.py** pour être lancé, dans le répertoire courant, avec **./reception.py**. Dès lors, il ouvrira le port **/dev/ttyUSB0** et se mettra en attente de données arrivant sur ce port.



Le convertisseur USB/série livré avec le couple de modules APC220 n'a rien de spécial si ce n'est la présence d'une broche RTS utilisée pour la configuration. Il sera donc possible d'utiliser n'importe quel adaptateur capable de dialoguer en utilisant des signaux 0/5V.

L'utilisation de `try`/`except` permet de gérer proprement une interruption avec le raccourci Ctrl+C en fermant le port avant de quitter. Cela se lit en « on essaie la boucle et on fait une exception en cas de Ctrl+C ». Le `while 1` est équivalent à un `while(1){}` d'un croquis Arduino, tout simplement une boucle infinie. Notez l'utilisation de la méthode `replace()` permettant de remplacer les caractères CR (`\r`) et LF (`\n`) par « rien » et donc de se débarrasser d'un saut de ligne inutile lors de l'affichage avec `print`.

Une fois ce script en place et en route sur la Raspberry Pi, nous pouvons nous tourner vers l'Arduino...

3.2 Côté Arduino

Notre croquis sera bien plus concis que le précédent, car la configuration des modules n'est plus à faire. En l'absence de mise à la masse de la broche SET, un module se comporte comme un canal de communication transparent :

```
#include <SoftwareSerial.h>

#define brocheRX 12
#define brocheTX 11

SoftwareSerial apc220(brocheRX,brocheTX);

unsigned int compteur;

void setup() {
  apc220.begin(2400);
  Serial.begin(115200);
}

void loop() {
  apc220.print("coucou ");
  apc220.println(compteur);

  Serial.print("coucou ");
  Serial.println(compteur);

  compteur++;

  delay(1500);
}
```

Notre boucle `loop()` consiste simplement à afficher une chaîne de caractères accompagnée d'un compteur dont la valeur s'incrémente et à envoyer la même chose via le module APC220 piloté avec SoftwareSerial via les broches 11 et 12.

L'enregistrement et l'exécution du croquis provoquera l'affichage, côté Pi, de :

```
[...]
coucou 58
coucou 59
coucou 60
coucou 61
coucou 62
coucou 63
coucou 64
^CCTRL+C !
Fermeture
```

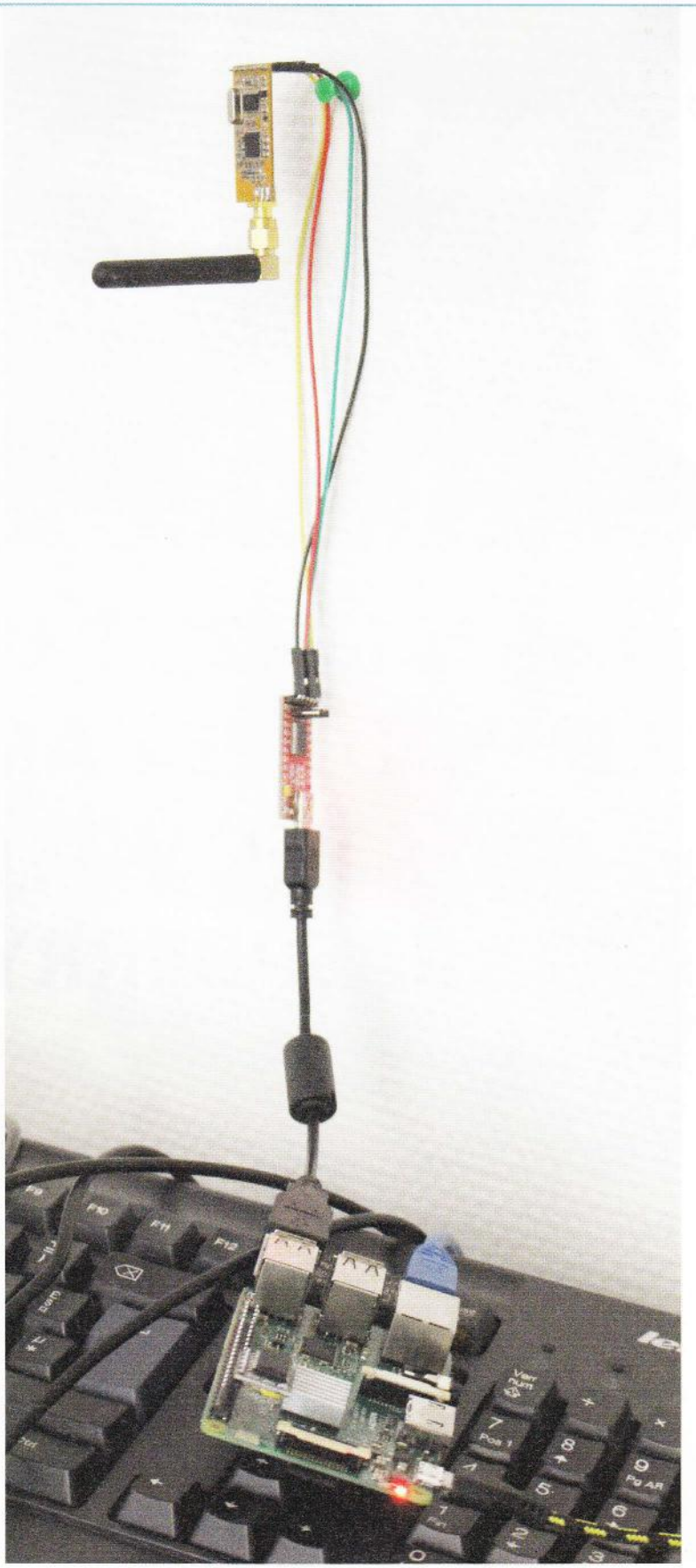
POUR CONCLURE

Une fois les modules correctement configurés, la liaison s'utilisera exactement comme une connexion série et de ce fait permettra de très facilement établir une communication entre différents montages reposant sur plusieurs plateformes (Arduino, Raspberry Pi, BeagleBone Black, Launchpad, etc.).

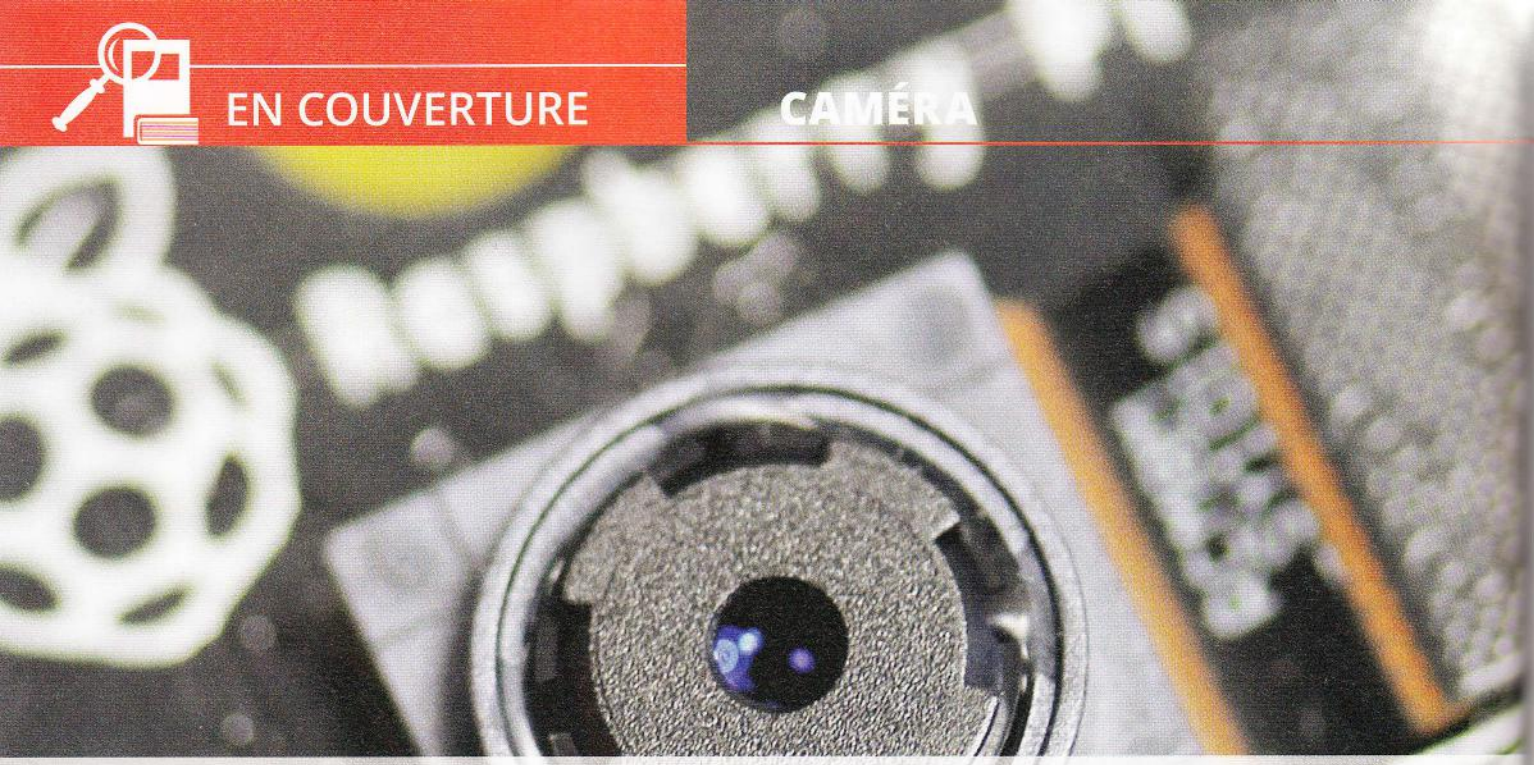
La question qui vient à l'esprit est bien entendu l'utilité de cette solution par rapport à son coût. Quelques 40€ pour une paire de modules de ce type est une somme conséquente qui n'est finalement rentable qu'en raison de la facilité de mise en œuvre. Certes, la communication sur plus de 600m en terrain découvert, en 2400 bps, est très séduisante (et vérifiée par mes soins). Ajoutons à cela qu'un module peut être mis en veille via sa broche EN et de ce fait réduire grandement la consommation électrique d'un montage, en particulier alimenté par un accumulateur ou des piles.

Enfin, j'ajouterai que le fait que ces modules communiquent de façon transparente vous permettra également de vous débarrasser de la carte Arduino. Certains modules et capteurs utilisent une liaison série pour communiquer. Coupler un APC220, judicieusement configuré, avec un tel module vous permettra de vous passer d'Arduino pour les faire communiquer via les ondes. Chose impossible avec des technologies plus évoluées qui nécessitent une prise en charge avec un croquis.

En conclusion, certes, c'est une solution séduisante, mais non tant par ses fonctionnalités techniques que par la facilité de mise en œuvre. Il existe d'autres modules, plus coûteux, utilisant une technologie appelée LoRa, bien plus configurables, plus modernes et avec une plus grande portée, mais aussi plus difficiles à utiliser. Je pense que le choix d'une technologie ou d'une autre est une question de temps de mise en pratique et de niveau technique. Si vous êtes pressé et souhaitez très rapidement et très simplement faire communiquer vos montages, l'APC220, est un bon choix. **DB**



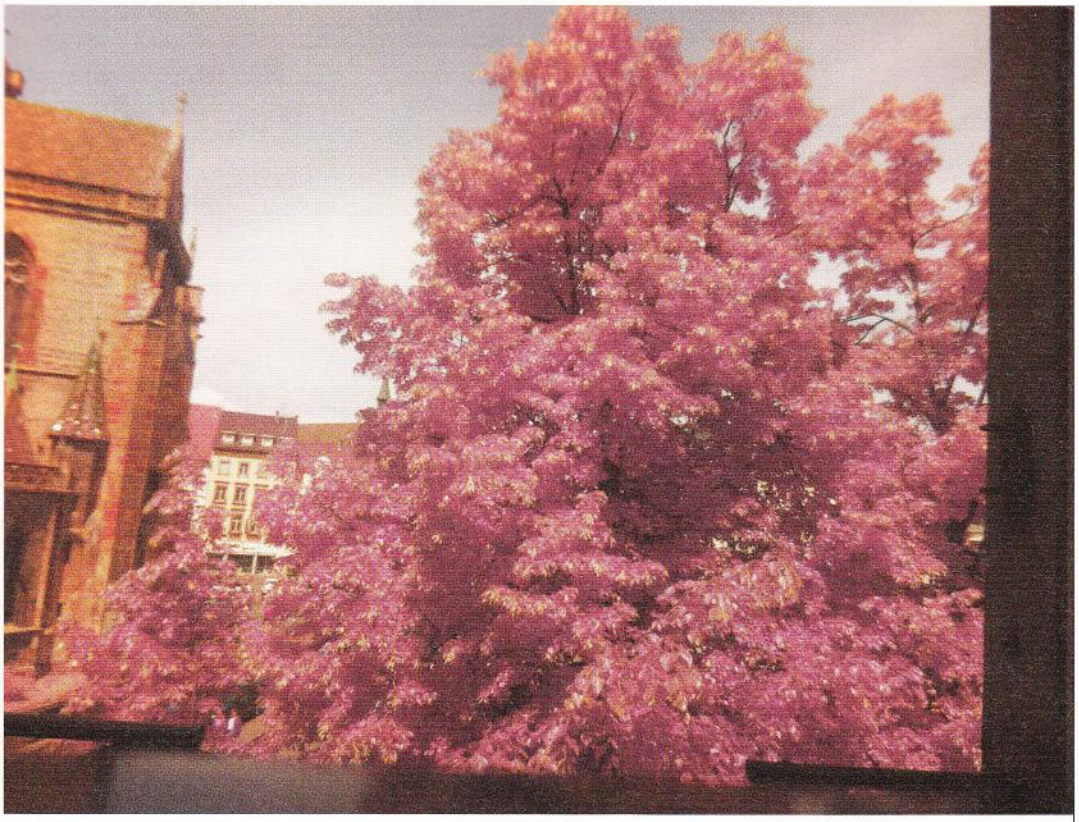
Voici l'une de nos installations hautement technologiques nous ayant permis de tester la communication au travers d'un bâtiment ancien. En 2400 bps le signal est passé sans problème au travers de 4 murs en pierre sur une distance de quelques 12m. D'autres tests en terrain découvert ont permis d'atteindre une distance de quelques 607m avant les premiers signes de pertes de données.



INSTALLEZ ET UTILISEZ LE MODULE CAMÉRA V2 AVEC VOTRE RASPBERRY PI

Denis Bodor

Deux étranges connecteurs peuplaient déjà les premières Raspberry Pi sans que, à l'époque, on ne sache vraiment quoi en faire. Depuis, l'écran officiel et le module caméra ont fait leur apparition. En avril dernier, la fondation annonçait l'arrivée d'une nouvelle version de ce module, bien plus performante et fonctionnant parfaitement avec toutes les Pi disponibles. Voyons ensemble de quoi il en retourne et comment mettre en œuvre ce nouveau jouet...



Le module caméra V2 NoIR ne dispose pas de filtre permettant de bloquer les rayonnements infrarouges. Les couleurs des images obtenues ne sont donc pas les mêmes que celles perçues par nos yeux.

Le contrôleur USB des SoC Broadcom équipant les Raspberry Pi ne jouit pas d'une réputation des plus brillantes, bien au contraire. On se souvient du message de Greg Kroah-Hartman, actuel responsable des branches stables du noyau Linux, décrivant il y a quelques années que le fait que l'USB fonctionne sur ces cartes était un miracle en soi tant le contrôleur et la connexion avec le processeur sont mauvais. Ajoutez à cela que la connexion au réseau filaire Ethernet passe également par le bus USB et vous comprendrez sans peine les problèmes rencontrés par les utilisateurs ayant tenté de transmettre un flux vidéo d'une webcam USB au travers un réseau (streaming).

L'arrivée de la caméra officielle se branchant directement au connecteur CSI (*Camera Serial Interface*), sans pour autant régler tous les problèmes, facilite grandement l'acquisition d'images fixes et de vidéos, puisque la connexion ne passe pas par l'USB, mais est prise en charge directement par le VideoCore IV intégré dans le SoC Broadcom. La création d'une nouvelle version du module, troquant le capteur OmniVision OV5647 contre un Sony IMX219, et par la même occasion une définition de 5 mégapixels (capteur 2592 x 1944) contre 8 mégapixels (capteur 3280 x 2464), apporte un lot très intéressant de bénéfices. Non seulement la résolution des images est plus importante, mais le contraste est grandement amélioré ainsi que la qualité d'image, la sensibilité ou encore le rendu des couleurs.

Ce nouveau module se décline en deux versions utilisant le même capteur et offrant donc les mêmes performances : une version standard et une version sans

Les caméras officielles arrivent dans de jolies boîtes en carton accompagnées d'un guide de sécurité en 16 langues.

filtre infrarouge, appelée NoIR. Également désigné sous le terme « miroir chaud », ce filtre est destiné à bloquer les rayonnements infrarouges pouvant atteindre le capteur. S'il est absent, la caméra est en mesure de capter un rayonnement imperceptible pour nos yeux en plus de la lumière visible. Il en découle un rendu des couleurs qui n'est plus authentique, mais surtout, la caméra est en mesure de capter un rayonnement infrarouge. En d'autres termes, il devient possible d'obtenir un système de vision nocturne dès lors qu'on éclaire la « cible » avec, par exemple, une ou plusieurs LEDs infrarouges dans la bonne longueur d'onde (entre 850 et 940 nm d'après mes tests). Notez que la notion de « vision nocturne » (ou *night vision*) est trompeuse ici puisqu'il ne s'agit en aucun cas d'amplification de lumière, mais simplement d'une sensibilité à un rayonnement invisible normalement pour les humains (ou les appareils photo numériques et webcams classiques).

L'une et l'autre version coûtent le même prix, environ 30€, et il conviendra de choisir en fonction de vos besoins. Personnellement, le rendu des couleurs n'est pas d'une importance capitale pour moi face à la possibilité d'obtenir des images dans l'infrarouge. Il n'en sera pas forcément de même pour vous. L'autre point intéressant dans le fait d'opter pour la version NoIR et la possibilité d'ajouter un filtre de son choix devant l'objectif : un filtre passe-haut pour ne laisser passer que les infrarouges et bloquer la lumière visible, ou un filtre passe-bas pour au contraire bloquer les infrarouges et transformer la caméra en une version « classique ».

1. INSTALLATION ET CONFIGURATION

Électroniquement parlant, les deux versions du module caméra V2 sont identiques et s'installent de la même manière. La première chose dont vous

devrez vous assurer avant toutes opérations est d'avoir votre système à jour (**`sudo apt-get update`** puis **`sudo apt-get dist-upgrade`**) et en mesure de supporter le matériel.

La connexion du module se fait directement sur la Raspberry Pi par l'intermédiaire d'un câble plat ou *flatflex* sur le connecteur « CAMERA » placé à proximité du connecteur HDMI (à l'exception de la nouvelle version de la Pi Zero avec un connecteur en bordure nécessitant un câble particulier vendu séparément). La connexion demande du doigté et se fait avec la partie conductrice orientée vers le connecteur HMDI. Il suffit de lever la partie mobile du connecteur, enfoncer doucement le câble et baisser la partie mobile de façon à bloquer le câble. Bien entendu, ceci doit être fait lorsque la Raspberry Pi n'est **PAS** sous tension.

Après le démarrage, vous pouvez soit utiliser l'outil **`raspi-config`** via **`sudo`** pour vous rendre dans le menu ***Enable Camera*** et choisir ***Enable***, soit éditer le fichier **`/boot/config.txt`** pour ajouter la ligne :

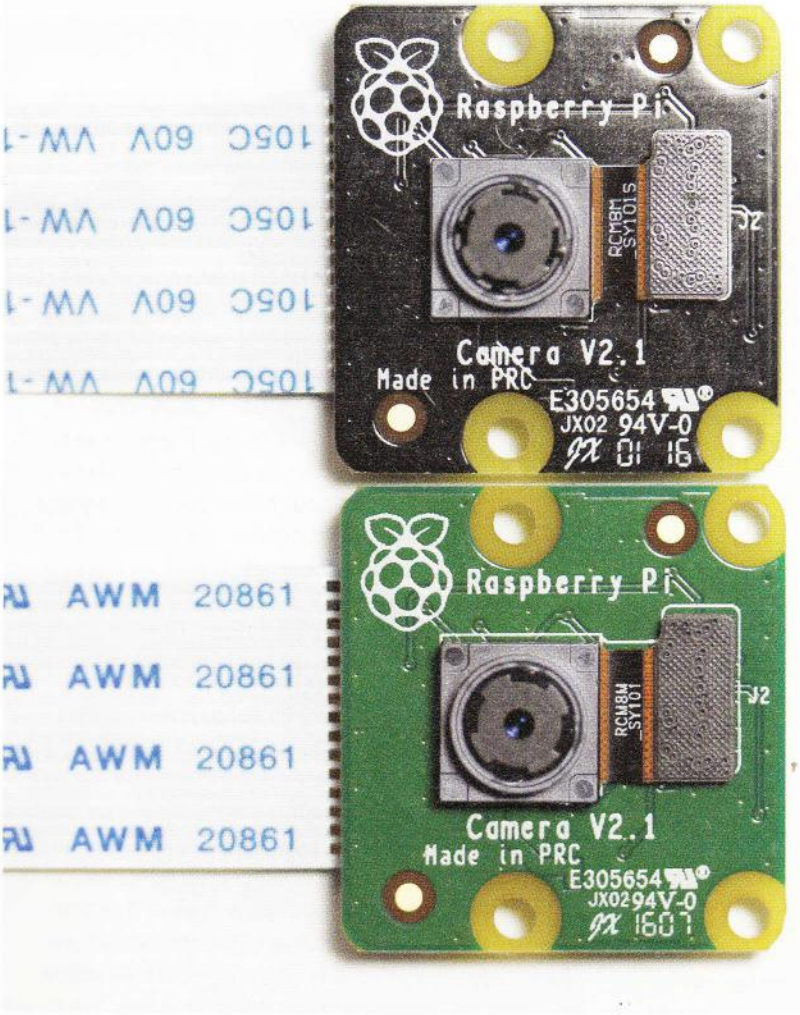
```
start_x=1
```

puis vous assurer que vous avez alloué 128 Mo de mémoire au processeur graphique (GPU) avec une ligne :

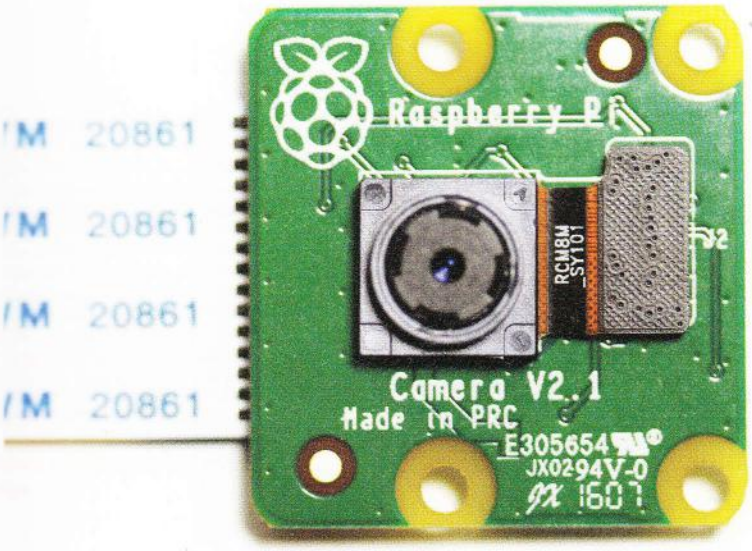
```
gpu_mem=128
```

Le module caméra utilise en effet la connexion CSI pilotée et est contrôlé par le processeur VideoCore IV, celui-là même qui s'occupe de l'initialisation, de la sortie HDMI et de l'écran éventuellement connecté sur DSI.

Dans un cas comme dans l'autre, un redémarrage du système sera nécessaire, après quoi, le module devrait être utilisable et vous pourrez procéder immédiatement à un premier test avec l'utilitaire **`raspistill`** permettant de capturer des images fixes. Si votre système est à jour, cet



Le nouveau module V2 est disponible en deux versions. En haut, celle sans filtre infrarouge appelée NoIR et en bas la version standard capable de capter uniquement la lumière visible.



La caméra standard se présente sous la forme d'un circuit imprimé de 23×25 mm possédant 4 trous de fixation. Notez l'optique se trouvant devant le capteur ne fournissant qu'une focale fixe. Il est cependant possible d'ajuster la mise au point, mais cela suppose de dévisser l'objectif en faisant céder la colle qui le maintient en place (risqué).

outil doit déjà être présent sur votre système (paquet **libraspberrypi-bin**) et la commande suivante vous fournira un fichier JPEG :

```
$ raspistill -o image.jpg
```

Vous noterez un temps relativement important entre la validation de la commande et la fin de son exécution (environ 6 secondes). Ceci est parfaitement normal et provient du fait que le module et la partie du VideoCore IV chargée de la gestion de la caméra (l'ISP pour *Image Sensor Pipeline*) active la caméra et utilise les images détectées pour déterminer les réglages optimaux avant de procéder à la capture. Il est possible de raccourcir ce délai en utilisant l'option **-t** suivie d'une durée en millisecondes (avec 0 étant une durée infinie), mais la qualité de l'image s'en trouvera impactée.

Si votre Raspberry Pi est connectée à un écran HDMI ou un écran LCD via le connecteur DSI, vous remarquerez également que l'image est tout d'abord affichée sur l'écran, et ce quelle que soit l'interface utilisée ou l'application lancée. Cet affichage se place en surimpression et est indépendant de ce qui se trouve à l'écran, car il provient directement du processeur graphique VideoCore IV. Il n'y a pas d'interaction avec la gestion de l'affichage par Linux, n'est pas pris en charge dans une quelconque fenêtre et fonctionne que vous soyez en mode graphique (X) ou en mode console (texte).

Cette fonctionnalité est tantôt appelée *viewfinder* dans la documentation en ligne et fait effectivement office de « viseur ». Notez que le fait que ceci est entièrement pris en charge par le matériel et ne fonctionne donc qu'avec les périphériques d'affichages supportés par le VideoCore IV (sortie HDMI et écran DSI), ce qui n'est pas le cas des écrans LCD TFT SPI comme ceux que nous avons présentés dans un précédent numéro, ni bien sûr d'une connexion réseau SSH ou d'une console série.

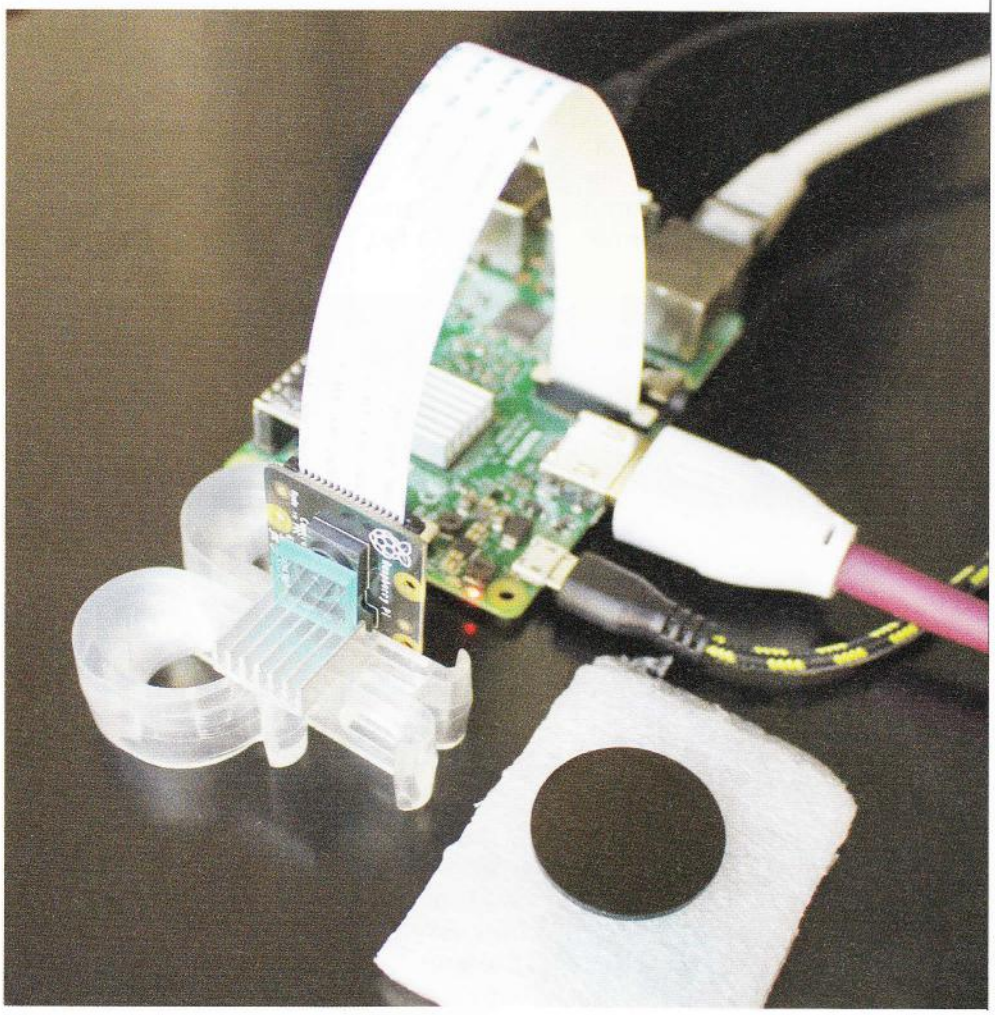
2. UTILISATION AVEC LES UTILITAIRES BROADCOM

Comme vous l'avez remarqué dans ce premier test, l'acquisition d'image peut se résumer à peu de choses. L'utilisation la plus basique de **raspistill** ne nécessite, en argument, qu'un seul paramètre, **-o**, permettant de spécifier le fichier où sera enregistré l'image. Par défaut, ceci se fera au format JPEG à une résolution de 2592 x 1944 pixels avec une qualité maximum (100 %). Selon la complexité de l'image, la taille du fichier pourra atteindre plus de 3 Mo.

Il existe une quantité impressionnante d'options disponibles et nous n'allons, bien entendu,

pas toutes les traiter ici (voir **https://www.raspberrypi.org/documentation/raspbian/applications/camera.md**). `raspistill --help` vous permettra d'avoir une liste complète, mais nous ne nous intéresserons ici qu'aux plus importantes. Les options se déclinent en deux « versions », courtes ou longues, en accord avec la fonction `getopt` de la bibliothèque LibC GNU. `--help`, par exemple pourra être remplacé par `-?` avec le même effet.

La première des options en question concerne l'orientation de la caméra. En effet, en fonction du montage utilisé, vous remarquerez que l'image nécessite un traitement pour correspondre à votre assemblage. `-hf` (`--hflip`) vous permettra un effet de miroir horizontal et `-hf` (`--vflip`) fera de même verticalement (les premières versions du pilote et des outils nécessitent, semble-t-il, systématiquement l'utilisation de `-hf`, ce qui n'est plus le cas à présent). Il vous sera également possible de tourner l'image sur elle-même, ce qui est fort pratique selon la position du module caméra et de la Raspberry Pi : `-rot` (`-rotation`) accepte théoriquement



Vue d'ensemble de l'installation utilisée pour nos tests. Le module caméra V2 NoIR est connecté à une Raspberry Pi 3 et disposé derrière un miroir chaud (filtre IR) récupéré sur un appareil photo numérique hors d'usage. Le disque noir en bas à droite est un filtre passe-haut bloquant la lumière visible et ne laissant passer que les IR avec une longueur d'onde supérieure à 770 nm.

un paramètre de rotation en degrés, mais les limitations matérielles restreignent ces valeurs à `0`, `90`, `180` et `270` : pas de rotation, quart de tour sens horaire, demi-tour, quart de tour sens anti-horaire.

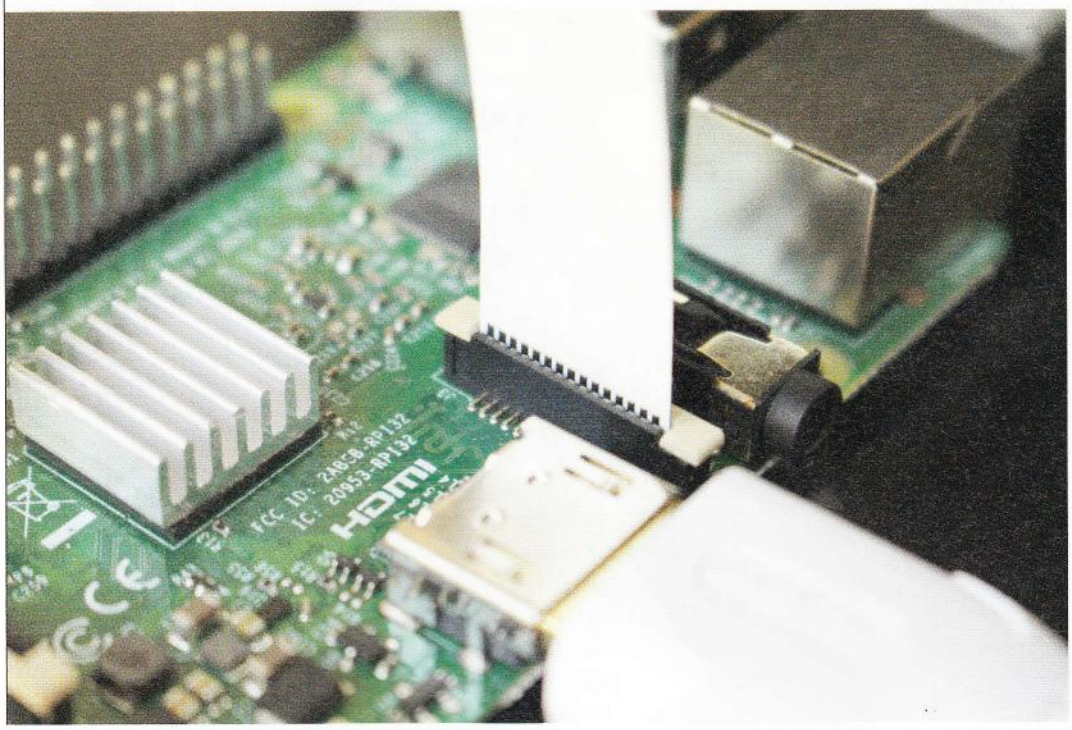
Un autre point très pratique concernant la prise de vue d'images fixes concerne la résolution. En effet, des images de 2592 x 1944 pixels bien que très agréables ne sont pas toujours un bon choix. Il est ainsi possible de spécifier une dimension horizontale avec `-w` (`-width`) et verticale avec `-h` (`--height`) ainsi qu'un niveau de qualité JPEG avec `-q` (`--quality`) entre 0 à 100. Ceci vous permettra de réduire drastiquement la taille des fichiers générés et donc d'économiser un volume conséquent d'espace de stockage sur la carte SD (ou la clé USB connectée à la Pi).

En fonction de la dimension ainsi spécifiée, la résolution du capteur sera automatiquement adaptée. Si toutefois pour une raison ou une autre vous souhaitez forcer cette résolution, vous pouvez utiliser l'option `-md` (`--mode`) en lui passant en paramètre un numéro de mode :

- `0` : automatique,
- `1` : 1920×1080 (1 à 30 fps),
- `2` : 2592×1944 (1 à 15 fps),
- `3` : 2592×1944 (0,166 à 1 fps),
- `4` : 1296×972 (1 à 42 fps),
- `5` : 1296×730 (1 à 49 fps),
- `6` : 640×480 (42,1 à 60 fps),
- `7` : 640×480 (60,1 à 90 fps).

Il vous est également possible d'influer sur les paramètres de prise de vue comme :

- la netteté (*sharpness*) avec `-sh` et une valeur de -100 à 100,
- le contraste avec `-co` et une valeur de -100 à 100,
- la luminosité avec `-br` et une valeur de 0 à 100,

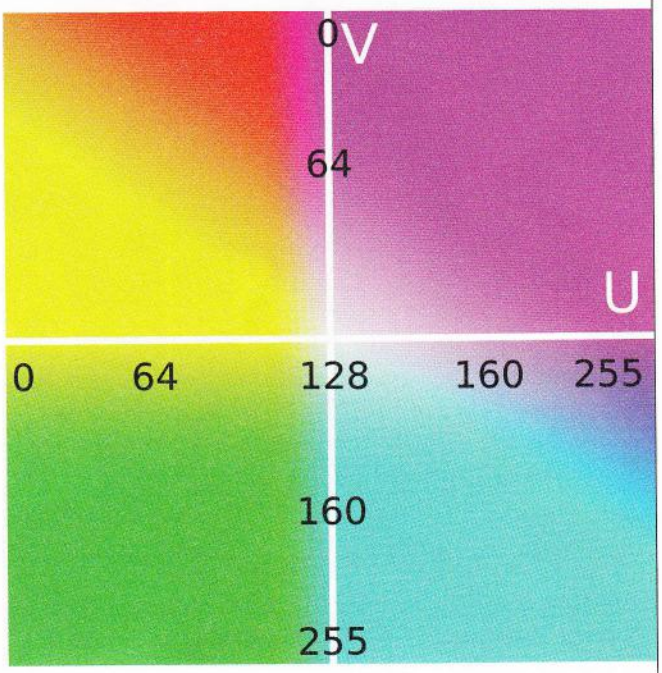


La connexion du module avec la carte Raspberry Pi se fait à l'aide d'un connecteur flatflex. Le module caméra vous est livré avec ce connecteur déjà fixé au module et vous devez simplement l'insérer, dans le bon sens, sur la carte.

- la saturation des couleurs avec **-sa** et une valeur de -100 à 100,
- la vitesse d'obturation avec **-ss** et une valeur en microsecondes avec un maximum de 6000000 µs,
- la sensibilité en ISO avec **-ISO** et une valeur entre 100 et 800,
- l'exposition avec **-ex** et un paramètre étant un profil à choisir entre **auto** (automatique), **night** (nuit), **nightpreview** (prévisualisation nuit), **backlight** (contre-jour), **spotlight** (éclairage direct), **sports** (sport), **snow** (neige), **beach** (plage), **verylong** (très longue), **fixedfps** (FPS fixe), **antishake** (stabilisation) ou **fireworks** (feu d'artifice),
- la balance automatique des blancs avec **-awb** et un profil pouvant être **off** (pas de balance des blancs), **auto** (automatique), **sun** (ensoleillé), **cloud** (nuageux), **shade** (nuancé), **tungsten** (éclairage au tungsten), **fluorescent** (tube fluorescent), **incandescent** (ampoule à filament) ou **flash** (flash), **horizon** (horizon).

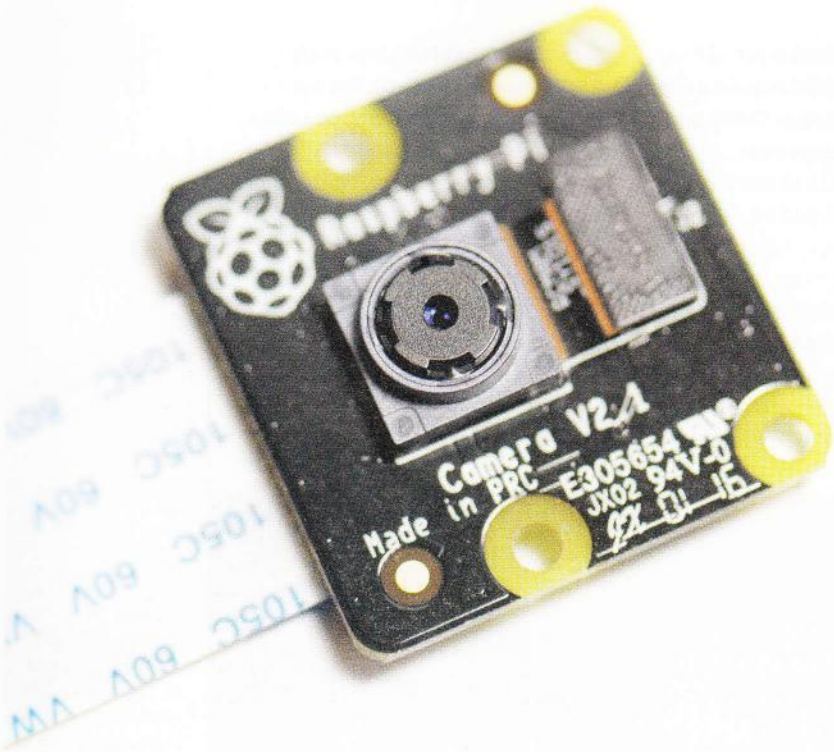
Toutes ces options peuvent s'avérer très utiles en cas de prise de vue spécifique afin d'avoir, de façon constante, des paramètres pour une série de clichés (voir les explications sur le *time-lapse* plus loin dans l'article).

Sauf cas particulier, il est peu probable que vous fassiez usage de ces options. Il n'en va pas de même pour **-ifx** (**--imxfx**), beaucoup plus amusant à utiliser puisqu'il s'agit d'appliquer un filtre d'effet sur l'image. Cette option prend en argument le nom de l'effet à appliquer : **none**, **negative**, **solarise**, **posterise**, **whiteboard**, **blackboard**, **sketch**, **denoise**, **emboss**, **oilpaint**, **hatch**, **gpen**, **pastel**, **watercolour**, **film**, **blur**, **saturation**, **colourswap**, **washedout**, **posterise**, **colourpoint**, **colourbalance** ou **cartoon**. Cette liste provient de la documentation officielle, mais toutes les valeurs ne fonctionnent pas ou ne sont pas pleinement implémentées pour l'instant. Décrire le résultat obtenu est relativement difficile et le plus simple est tout bonnement de les tester par vous-même...



Espace colorimétrique Y'UV utilisé pour la colorisation de certains effets utilisables avec la caméra.

Certains de ces effets utilisent une couleur qu'il est possible de spécifier avec l'option **-cfx** (**--colfx**) prenant en argument deux valeurs séparées par un double-point. Ces valeurs correspondent à la chrominance de l'image dans l'espace colorimétrique YUV très utilisé en vidéo. Comme le montre le diagramme ci-dessus, cet espace colorimétrique désigne une couleur via des coordonnées sur des axes U et V normalement étalonnés de -0,5 à 0,5. Dans le cas de cette option, les axes sont référencés avec des valeurs de 0 à 255, 128 étant donc l'origine (0). Il existe des convertisseurs en ligne permettant d'obtenir U et V à partir de valeurs pour le rouge, ..



La version NoIR du module caméra V2 s'identifie facilement en raison de son circuit imprimé noir (et non vert). Il est amusant d'ailleurs de remarquer que le module NoIR (pas IR, pour pas de filtre infrarouge) est... noir. Pas sûr que l'appellation ait été choisie pour le jeu de mot anglo-français...

le vert et le bleu (espaces RVB ou RGB en anglais). Il est dommage que `raspistill` ne propose pas d'utiliser une telle désignation de couleur en procédant à la conversion en interne.

2.1 Concernant raspistill

Il existe d'autres options pouvant s'appliquer aussi bien à `raspistill` qu'à l'outil de capture de vidéo `raspivid`, mais leur utilité est bien trop spécifique pour encombrer cet article. Certaines options cependant sont propres à `raspistill`, comme `-e` (`--encoding`) permettant de choisir le format du fichier généré entre `gif`, `png`, `bmp` ou `jpg` (par défaut).

Parmi les options spécifiques à `raspistill`, certaines s'avèrent très intéressantes. C'est le cas, par exemple, de `-k` (`--keypress`) permettant d'activer un mode particulier où l'outil ne prendra de photo que si une touche est enfoncée. Cette option s'utilisera de préférence avec `-v` pour un affichage « verbeux » et vous permet de ne pas être dépendant du délai par défaut permettant l'ajustement automatique de la caméra. Ainsi, il suffit de lancer `raspistill -k -v -t 0 -o fichier.jpg`, d'utiliser la prévisualisation à l'écran pour ajuster la prise de vue et d'appuyer sur [Entrée] pour déclencher la capture.

Mieux encore, en utilisant un nom particulier pour le fichier de sortie, comme par exemple `fichier%d.jpg`, chaque pression sur [Entrée] provoquera la création d'un nouveau fichier : `fichier1.jpg`, `fichier2.jpg`, `fichier3.jpg`, etc. Il est également possible de préfixer le numéro de la capture d'un certain nombre de `0`, facilitant ainsi le tri alphabétique des fichiers, avec `%04d` en lieu et place de simplement `%d`, `0` indiquant le préfixe et `4` le nombre de chiffres composant la numérotation (`0001`, `0002`, `0003`, etc.).

Pour quitter `raspistill`, utilisez la touche [X] puis [Entrée]. Une dernière image sera cependant capturée. Vous pouvez aussi, simplement, utiliser [Ctrl+C].

Il faut, pour que cela fonctionne, bien entendu, être en mesure d'envoyer des caractères à l'utilitaire que ce soit dans une fenêtre de terminal, en console ou via une connexion distante SSH. Ceci suppose donc soit un accès physique à la Raspberry Pi, soit le maintien de la connexion SSH.

HACKABLE MAGAZINE

DÉCOUVREZ NOS OFFRES D'ABONNEMENTS !

PRO OU PARTICULIER = CONNECTEZ-VOUS SUR :

www.ed-diamond.com

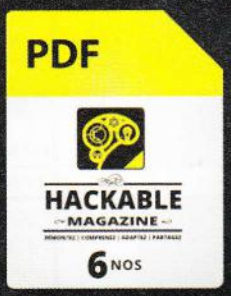
LES COUPLAGES PAR SUPPORT :

VERSION
PAPIER

Retrouvez votre magazine favori en papier dans votre boîte à lettres !

VERSION
PDF

Envie de lire votre magazine sur votre tablette ou votre ordinateur ?



SÉLECTIONNEZ VOTRE OFFRE DANS LA GRILLE AU VERSO ET RENVOYEZ CE DOCUMENT COMPLET À L'ADRESSE CI-DESSOUS !

Voici mes coordonnées postales :	
Société :	
Nom :	
Prénom :	
Adresse :	
Code Postal :	
Ville :	
Pays :	
Téléphone :	
E-mail :	

☐ Je souhaite recevoir les offres promotionnelles et newsletters des Éditions Diamond.

☐ Je souhaite recevoir les offres promotionnelles des partenaires des Éditions Diamond.

En renvoyant ce bon de commande, je reconnais avoir pris connaissance des conditions générales de vente des Éditions Diamond à l'adresse internet suivante : boutique.**ed-diamond.com/content/3-conditions-generales-de-ventes** et reconnais que ces conditions de vente me sont opposables.

HACKABLE MAGAZINE

Les Éditions Diamond
Service des Abonnements
10, Place de la Cathédrale
68000 Colmar – France
Tél. : + 33 (0) 3 67 10 00 20
Fax : + 33 (0) 3 67 10 00 21

Vos remarques :

VOICI TOUTES LES OFFRES COUPLÉES AVEC HACKABLE !
POUR LE PARTICULIER ET LE PROFESSIONNEL ...

Prix TTC en Euros / France Métropolitaine

CHOISISSEZ VOTRE OFFRE !

Prix en Euros / France Métropolitaine

Offre	SUPPORT / ABONNEMENT	PAPIER Réf	PAPIER Tarif TTC	PAPIER + PDF (PDF 1 lecteur) Réf	PAPIER + PDF Tarif TTC	PAPIER + BASE DOCUMENTAIRE (1 connexion BD) Réf	PAPIER + BASE DOCUMENTAIRE Tarif TTC	PAPIER + PDF + BASE DOCUMENTAIRE (PDF 1 lecteur + 1 connexion BD) Réf	PAPIER + PDF + BASE DOCUMENTAIRE Tarif TTC
HK	6n° HK*	☐ HK1	39,-	☐ HK12	58,-				
LES COUPLAGES « EMBARQUÉ »									
D	6n° HK* + 4n° OS	☐ D1	65,-	☐ D12	98,-	☐ D13	85,-*	☐ D123	118,-*
E	6n° HK* + 4n° OS + 6n° MISC	☐ E1	105,-	☐ E12	158,-	☐ E13	179,-*	☐ E123	232,-*
E+	6n° HK* + 4n° OS + 6n° MISC + 2n° HS	☐ E+1	119,-	☐ E+12	179,-	☐ E+13	193,-*	☐ E+123	253,-*
F	6n° HK* + 4n° OS + 11n° GLMF	☐ F1	125,-	☐ F12	188,-	☐ F13	229,-*	☐ F123	292,-*
F+	6n° HK* + 4n° OS + 11n° GLMF + 6n° HS	☐ F+1	183,-	☐ F+12	275,-	☐ F+13	287,-*	☐ F+123	379,-*
G	6n° HK* + 4n° OS + 6n° LP	☐ G1	100,-	☐ G12	150,-	☐ G13	164,- *	☐ G123	214,-*
G+	6n° HK* + 4n° OS + 6n° LP + 3n° HS	☐ G+1	129,-	☐ G+12	194,-	☐ G+13	193,-*	☐ G+123	258,-*
LES COUPLAGES « GÉNÉRAUX »									
H	6n° HK* + 4n° OS + 6n° LP + 6n° MISC + 11n° GLMF	☐ H1	200,-	☐ H12	300,-	☐ H13	402,-*	☐ H123	499,-*
H+	6n° HK* + 4n° OS + 6n° LP + 3n° HS + 6n° MISC + 2n° HS + 11n° GLMF + 6n° HS	☐ H+1	301,-	☐ H+12	452,-	☐ H+13	493,-*	☐ H+123	639,-*

Les abréviations des offres sont les suivantes : LM = GNU/Linux Magazine France | HS = Hors-Série | LP = Linux Pratique | OS = Open Silicium | HC = Hackable

* HK : Attention : La base Documentaire de Hackable n'est pas incluse dans l'offre.

N'hésitez pas à consulter les détails des offres ci-dessus sur : www.ed-diamond.com !

Il existe cependant une autre solution via l'option **-s** (**--signal**) permettant à **raspistill** de rester en attente et de ne prendre des clichés, non pas lorsqu'une touche est utilisée, mais lors de l'arrivée d'un signal. La notion de signal dans un système UNIX comme le GNU/Linux de la distribution Raspbian permet la communication inter-processus (IPC). Les outils et programmes en mémoire peuvent ainsi recevoir et émettre des signaux et réagir en fonction de leur nature. Lorsque vous stoppez l'exécution d'un programme avec [Ctrl+C], par exemple, celui-ci reçoit en réalité le signal SIGINT. Il en existe beaucoup d'autres, SIGHUP, SIGQUIT, SIGKILL, etc. (la commande **kill -l** vous les listera tous), mais deux en particulier peuvent être utilisés sans conséquence : SIGUSR1 et SIGUSR2. Ils n'ont pas d'effet par défaut dans le système et leur interprétation est laissée aux bons soins du programmeur.

Ainsi, **raspistill** lancé avec **-s -t 0 -o fichier%04d.jpg** se placera en mémoire et attendra un signal SIGUSR1 pour prendre une photo et l'enregistrer, comme précédemment, dans un fichier numéroté. Comment envoyer le signal ? Tout simplement avec la commande **kill**, l'option **-USR1** et le numéro du processus visé. Chaque programme en fonction, lors de son exécution, se voit attribuer un numéro (ou plusieurs) de processus dans le système. Ce numéro peut être récupéré avec la commande **pidof** suivie du nom du programme, s'il est en cours d'exécution.

Avec un peu de « sorcellerie » du shell, il est possible de combiner tout cela en une commande : **kill -USR1 'pidof raspistill'** (notez qu'il s'agit d'apostrophes inverses obtenues avec [AltGr+7]). Il est même possible d'aller un cran plus loin en lançant **raspistill** ainsi :

```
$ nohup raspistill -s -v -t 0 -o fichier%04d.jpg &
[1] 22753
```

nohup et **&** permettent de détacher le programme lancé du terminal. En d'autres termes, vous récupérez la main et **raspistill** reste en fonction en mémoire. En cas de connexion SSH, vous pouvez alors vous déconnecter, fermer la fenêtre de terminal ou tout simplement vous déconnecter du système, **raspistill** sera toujours là à attendre un SIGUSR1. Ce que devrait afficher **raspistill** sera automatiquement stocké dans un fichier **nohup.out** :

```
raspistill Camera App v1.3.8

Width 2592, Height 1944, quality 85, filename fichier%04d.jpg
Time delay 0, Raw no
Thumbnail enabled Yes, width 64, height 48, quality 35
Link to latest frame enabled  no
[...]
Waiting for SIGUSR1 to initiate capture
Received SIGUSR1
Opening output file fichier0001.jpg
Enabling encoder output port
Starting capture 1
Finished capture 1
Waiting for SIGUSR1 to initiate capture
```

Pour arrêter **raspistill**, étant donné que vous n'avez plus accès à son exécution en direct, il vous faudra utiliser **kill**, mais avec l'option **-INT** pour lui envoyer un SIGINT, comme avec un [Ctrl+C].

Qu'il s'agisse d'utiliser `-k` ou `-s`, cette « mise en veille » avant prise de vue présente un avantage certain : le réglage automatique est constant et de ce fait la prise de photos quasi immédiate. De plus, c'est une façon très intéressante de « télécommander » la prise de vue.

2.2 Enregistrement de vidéos avec raspivid

L'utilitaire `raspivid`, comme son nom l'indique est l'équivalent de `raspistill` pour la vidéo. Cet outil s'utilise très simplement en spécifiant l'option `-o` suivie d'un nom de fichier où sera enregistrée la vidéo :

```
$ raspivid -o video.h264
```

Ceci aura pour effet d'enregistrer une vidéo de 5 secondes dans un fichier au format MPEG-4 AVC également appelé H.264. Il s'agit d'un format d'encodage normalisé (ISO/CEI 14496-10) très efficace et se prêtant à de nombreux usages. Celui-ci est directement utilisé matériellement et l'outil `raspivid` ne procède pas à l'encodage à proprement parler.

Le délai par défaut de 5 secondes, ou 5000 ms, peut être changé en utilisant l'option `-'t`. Si la valeur passée est `0`, la durée de l'enregistrement n'est pas spécifiée et ne s'arrêtera qu'en utilisant [Ctrl+C]. Les options `-k` et `-s` s'utilisent de la même manière qu'avec `raspistill` à la différence que l'utilitaire basculera de l'enregistrement à la pause et inversement, à chaque utilisation de la touche [Entrée] ou réception du signal SIGUSR1. Il est possible, de plus, de spécifier l'état initial au lancement avec l'option `-i` et un argument pouvant être `record` ou `pause`.

Toujours dans la gestion du temps et de l'enregistrement, `raspivid` peut se comporter comme `raspistill` en générant plusieurs fichiers dont le nom comprend une séquence numérique. Il faut cependant spécifier l'option `-sg` (`--segment`) en précisant une durée en millisecondes pour qu'une mention de `%04d` par exemple ait un effet.

Une commande comme `raspivid -t 0 -sg 60000 -o video%04d.h264` génèrera ainsi une succession de fichiers contenant une minute de vidéo et appelés `video0001.h264`, `video0002.h264`, `video0002.h264`, etc. Chose qui peut être très intéressante pour une application en tant que caméra de sécurité.

Cette segmentation de la vidéo pourra être également très intéressante pour éviter de saturer l'espace de stockage de la Pi. Il suffira, en effet, de déplacer les fichiers régulièrement vers un PC ou une clé USB à mesure que leur nombre augmente.

En parlant de taille justement, une capture standard d'une minute occupera environ, en moyenne, 1 Go ! Bien entendu, cela est fortement dépendant des images capturées, du mouvement sur la vidéo et de la complexité de l'ensemble. Les vidéos capturées par défaut le sont avec une résolution de 1920 par 1080 pixels (full HD donc), mais il vous est possible de réduire cela avec les options `-w` et `-h` jusqu'à 64 pixels. Ceci aura un effet direct sur la taille des fichiers vidéos.

Vous pouvez cependant jouer sur d'autres paramètres parmi lesquels le nombre d'images par seconde (fps) avec l'option `-fps` (`--framerate`) entre 2 et 30 fps. Une autre option importante dans la même idée est le *bitrate* ou le nombre de bits par seconde du flux vidéo, avec `-b` (`-bitrate`). Une vidéo de bonne qualité en 1080 et 30 fps utilise généralement un bitrate de 15 Mb/s, soit l'option `-b 15000000`. Le bitrate maximum est de 25 Mb/s, mais une valeur supérieure à 17 Mb/s n'apporte généralement pas d'améliorations perceptibles.

En jouant sur ces paramètres, vous arriverez à drastiquement réduire la taille de vos vidéos, certes au détriment de la qualité, mais avec une carte microSD

de 8 Go dont la moitié souvent est déjà occupée par le système, il faut savoir faire des sacrifices. Ainsi une vidéo d'une minute enregistrée avec la ligne suivante :

```
$ raspivid -v -t 60000 -b 8000000 -fps 24 -w 960 -h 540 -o video2.h264
```

n'occupera plus que 17 Mo (68 fois moins que la même enregistrée avec les paramètres par défaut) et conservera une qualité tout à fait acceptable dans la plupart des contextes.

3. STREAMING

Être en mesure d'enregistrer une vidéo ou une série de vidéos est une chose, obtenir un flux vidéo et le transférer en temps réel de la Raspberry Pi vers un PC en est une autre. Les outils **raspistill** et **raspivid** disposent d'une fonctionnalité intéressante, existant dans la plupart des outils GNU en ligne de commandes intégrés dans le système : l'utilisation de la sortie standard en lieu et place d'un fichier.

De la même façon qu'il vous est possible de rediriger la sortie d'une commande, comme **dmesg** par exemple, vers un autre utilitaire, comme **grep** ainsi :

```
$ dmesg | grep "rtc"
[ 4.413652] rtc-ds1307 1-0068:
rtc core: registered ds3231 as rtc0
```

afin de demander à **grep** de filtrer et afficher uniquement les lignes contenant le texte **"rtc"**, il est possible de demander à **raspivid** d'envoyer le flux vidéo non pas dans un fichier, mais à l'écran. Pour cela, il suffit de spécifier **-** en tant que nom de fichier en argument de l'option **-o**. Attention cependant, si vous le faites sans rediriger la sortie en question vers quelque chose, les données arriveront à l'écran en masse (et je ne parle pas ici d'un affichage vidéo, mais d'un flot de caractères bizarroïdes).

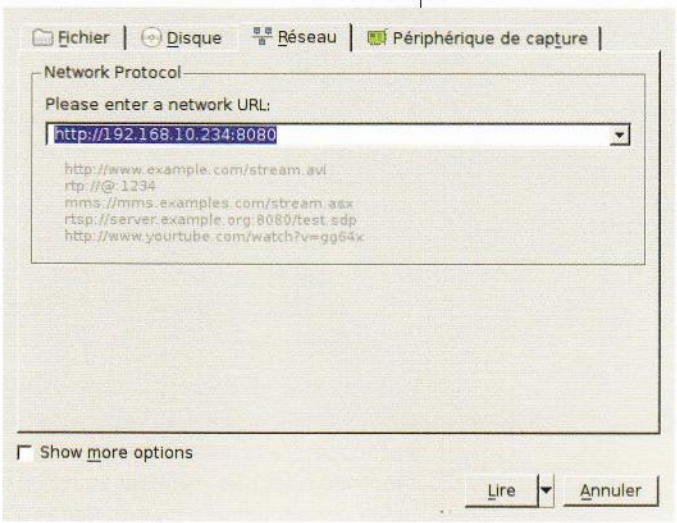
On peut alors utiliser une redirection **>** pour envoyer ces données dans un fichier mais, aussi et surtout, **|** ([AltGr+6]) pour les passer à l'entrée standard d'une autre commande.

Ainsi, selon la commande utilisée, il est possible de traiter le flux vidéo en temps réel et si cette commande est **cvlc**, la version sans interface graphique du media player VLC (paquet **vlc-nox**), nous pouvons créer un serveur de streaming diffusant la vidéo sur le réseau.

Voici la commande utilisée : **raspivid -t 0 -o - | cvlc --demux h264 --sout '#standard{access=http,mux=ts,dst=:8080/video.h264}' stream:///dev/stdin**.

raspivid -t 0 -o - est relativement explicite à présent que nous connaissons les options utilisées ainsi que le principe de redirection avec **|**.

VLC est une application en logiciel libre qu'on ne présente plus. C'est cet outil qui, sur la Raspberry Pi, vous permettra de diffuser la vidéo sur le réseau (filaire de préférence) mais aussi, côté PC/Mac de la récupérer pour l'afficher à l'écran. Prenez le temps de faire le tour des menus de l'application, VLC est bien plus qu'un simple player de vidéos.



En utilisant une taille/résolution, un framerate et un bitrate raisonnables, on trouvera rapidement un juste équilibre entre fluidité et qualité de la vidéo diffusée en streaming par VLC.

Le flux de données est passé à **cvlc** en spécifiant les options **--demux h264** et **--sout** pour *stream out*. La première permet d'activer le module *demuxer* pour le format H.264 permettant ainsi à VLC de traiter correctement les données reçues via l'entrée standard. La seconde détermine la façon de générer un flux à partir de ces données. Enfin, le dernier argument de la ligne précise où VLC doit lire les données, **/dev/stdin** représentant, bien entendu, l'entrée standard.

Ici, nous n'avons pas d'encodage puisque les données sont déjà en H.264. Tout ce que VLC doit faire c'est de prendre ce flux encodé et le mettre à disposition sur le réseau via le protocole HTTP sur le port 8080. Ainsi, une fois cette commande lancée, il suffira de lancer VLC sur un autre ordinateur, utiliser le menu ***Média***, ***Ouvrir un flux réseau*** et préciser **http://192.168.10.234:8080/video.h264** où **192.168.10.234** est l'adresse IP de votre Raspberry Pi sur le réseau local.

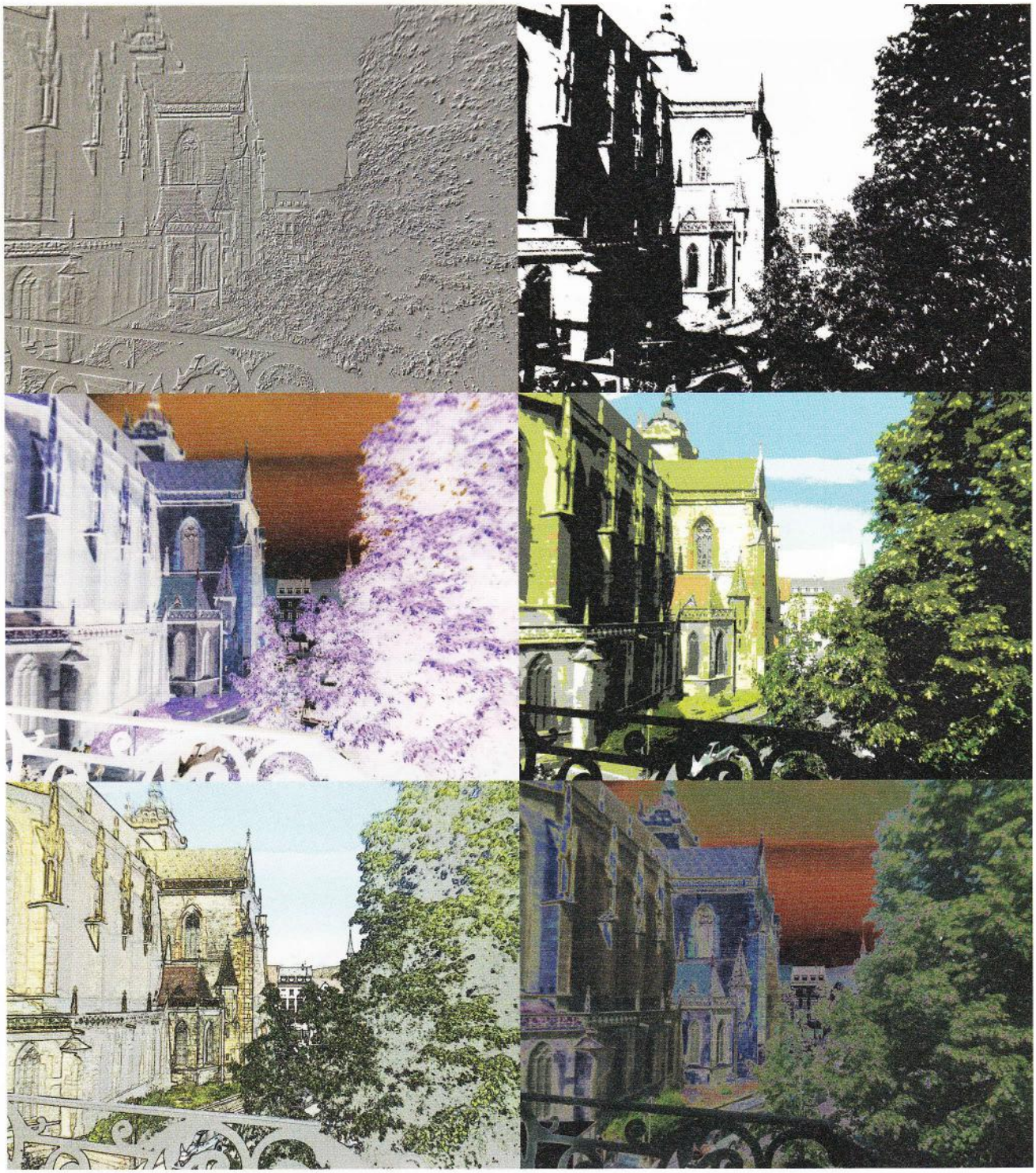
Vous devriez après validation avec le bouton ***Lire*** commencer à voir la vidéo au bout de quelques secondes. Vous remarquerez sans doute que la réactivité n'est pas vraiment au rendez-vous et qu'un certain nombre de problèmes d'affichage apparaîtront. La raison en est simple : la masse de données transférée est tout bonnement énorme et la Pi n'est pas un modèle de puissance de calcul.

Vous pouvez, pour corriger cela, revoir les options utilisées par **raspivid**, en réduisant le nombre d'images par seconde, le bitrate et la taille de la vidéo avec quelque chose comme **raspivid -ih -w 640 -h 480 -b 8000000 -fps 24 -t 0 -o -**.

Le principal avantage d'utiliser VLC pour cette diffusion est la disponibilité du player sur de nombreuses plateformes (Windows, Mac, GNU/Linux, Android, etc.) et sa facilité d'utilisation, en plus de la compatibilité H.264 qui évite tout encodage/décodage sur la Raspberry Pi. Sur le poste client, il vous sera ainsi possible de voir la vidéo en temps réel, mais également de l'enregistrer à distance (menu ***Média***, ***Convertir/Enregistrer***). Ceci pourra donc faire office de système de vidéosurveillance très efficace.

4. TIME-LAPSE

Il y a une option de **raspistill** que je n'ai délibérément pas encore présentée, car elle est tout bonnement fantastique : **-tl** ou **--time-lapse**. Celle-ci permet de créer des *time-lapses* en demandant à l'outil de prendre un cliché à intervalle régulier. Cette option se combine à merveille avec le nommage séquentiel des fichiers générés et prend en argument un délai en millisecondes entre les prises de vue.



Différents effets peuvent être appliqués aux images captées par le module, en temps réel (image fixe et vidéo). Ici, une petite sélection de quelques-uns d'entre eux : emboss, gpen, negative, posterise; sketch et solarise.

L'idée est ici de faire un cliché toutes les *n* secondes pour ensuite combiner l'ensemble des images en une vidéo. Le résultat est tout simplement fantastique puisqu'on obtient alors une version accélérée d'un événement. Avec une prise de vue toutes les secondes (1000 ms), on obtient alors une vision accélérée idéale pour observer, par exemple, la couverture nuageuse en mouvement au-dessus d'un paysage ou un coucher de soleil. Augmentez encore l'intervalle à quelques minutes et choisissez un sujet végétal ou fongique quelconque et vous pourrez observer, après quelques jours sa croissance accélérer...

Tout ce que vous avez à faire pour cela est d'utiliser judicieusement les options **-tl**, **-t** et **-o**, par exemple ainsi : **raspistill -tl 10000 -t 0 -o fichier%08d.jpg**. Deux choses sont cependant à prendre en considération :

- l'espace de stockage utilisé, plus le délai est court entre les clichés plus vous aurez d'espace utilisé pour une durée donnée ;
- et le nombre d'images générées en tout qui, forcément, impactera le nommage des fichiers. Personnellement, j'utilise **%08d** par défaut, me permettant, à coup sûr de pouvoir aller jusqu'au fichier **fichier99999999.jpg**, bien plus que ce que serait capable de contenir une clé USB ou une microSD.

Obtenir les images n'est pas difficile puis qu'il suffit de laisser faire **raspistill** et de ne pas déplacer la Raspberry Pi pour ne pas gâcher le résultat (sauf si c'est délibéré). En revanche, pour regrouper les images en une seule vidéo, il vous faudra utiliser un outil spécifique : **avconv** du paquet **libav-tools** (vous pouvez aussi utiliser **ffmpeg** sur un PC GNU/Linux par exemple, avec la même syntaxe).

Une fois l'outil installé, il vous suffira de vous placer dans le répertoire où se trouve la série d'images JPEG produites par **raspistill** et d'utiliser :

```
$ avconv -r 24 -i fichier%08d.jpg video.mp4
avconv version 11.6-6:11.6-1~deb8u1+rpi1,
  Copyright (c) 2000-2014 the Libav developers
  built on Mar 22 2016 15:53:22 with gcc 4.9.2 (Raspbian 4.9.2-10)
Input #0, image2, from 'CAM/test%08d.jpg':
  Duration: 00:00:39.36, start: 0.000000, bitrate: N/A
    Stream #0.0: Video: mjpeg, yuvj420p, 1280x1024, 25 fps, 25 tbn
[libx264 @ 0x171320] using cpu capabilities: ARMv6 NEON
[libx264 @ 0x171320] profile High, level 3.2
[libx264 @ 0x171320] 264-core 142 r2431a5831aa-H.264/MPEG-4 AVC codec
Output #0, mp4, to 'testVID.mp4':
  Metadata:
    encoder         : Lavf56.1.0
    Stream #0.0: Video: libx264, yuvj420p, 1280x1024, q=-1--1, 24 fps, 24 tbn, 24 tbc
    Metadata:
      encoder         : Lavc56.1.0 libx264
Stream mapping:
  Stream #0:0 -> #0:0 (mjpeg (native) -> h264 (libx264))
Press ctrl-c to stop encoding
```

```
frame=  113 fps=  2 q=28.0 size=     975kB time=2.50 bitrate=3196.4kbits/s     its/s
video:15655kB audio:0kB other streams:0kB global headers:0kB muxing overhead:
0.060816%
[libx264 @ 0x171320] frame I:5     Avg QP:21.75  size: 43345
[libx264 @ 0x171320] frame P:934   Avg QP:24.13  size: 16390
[libx264 @ 0x171320] frame B:45    Avg QP:24.87  size: 11222
[...]
[libx264 @ 0x171320] ref B L0: 76.1% 23.9%
[libx264 @ 0x171320] kb/s:3127.73
```

avconv prend ici en argument l'option **-r** pour spécifier nombre d'images par seconde dans la vidéo et **-i** pour désigner les noms de fichiers à traiter. Notez que la syntaxe est identique à celle de **raspistill** avec le **%08d**. Le dernier argument de la ligne de commandes est le fichier vidéo à produire qui, par défaut sera en MP4 H.264.

Cette commande va alors assembler la série d'images en une vidéo en commençant par **fichier00000001.jpg** (ou en cherchant automatiquement le fichier suivant dans la séquence, jusqu'au cinquième) et en utilisant les JPEGs dans leur ordre de numérotation. La vidéo obtenue, cadencée à 24 fps, sera la version accélérée du temps écoulé lors des prises de vue.

L'exécution de cette commande est relativement lente sur Raspberry Pi en raison du manque de puissance CPU par rapport à un PC, mais permet d'automatiser tout cela en une fois dans un script shell. Les JPEGs ayant servi à la génération pourront ensuite être effacés pour une nouvelle exécution et ainsi produire, par exemple, une vidéo par heure, par jour ou par mois, de façon automatique.

CONCLUSION

En résumé, ces deux nouveaux modules caméra sont fort sympathiques malgré quelques petits problèmes. En effet, même si matériellement et logiciellement, tout cela permet d'obtenir des images fixes et des vidéos de bonne qualité, de nombreuses critiques d'utilisateurs sont dirigées contre l'optique. L'absence de possibilité de procéder à une mise au point est l'un des points faibles de cet accessoire.

Les bordures des images capturées sont généralement floues et de nombreux utilisateurs semblent avoir reçu des modules dont la mise au point n'était pas satisfaisante en sortie d'usine. La lentille placée devant le capteur est vissée sur le support, mais un point de colle « verrouille » l'ensemble. En procédant avec soin, il est cependant possible, à l'aide de pinces, de dévisser de force l'optique. Il faut toutefois procéder avec la plus grande prudence, car l'ensemble capteur+support+lentille est simplement attaché au circuit avec un adhésif double face.

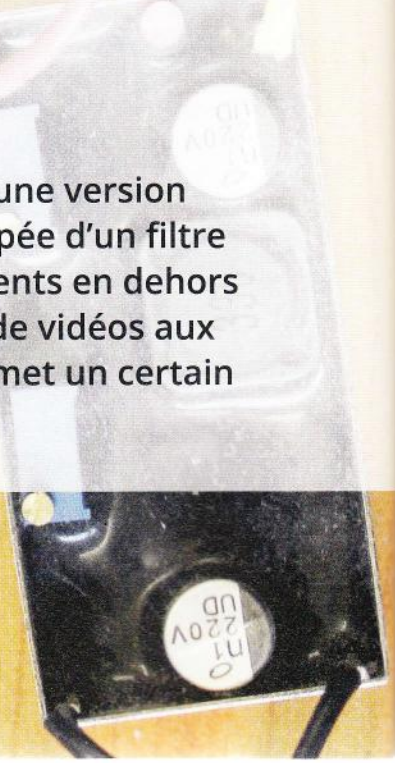
Quoi qu'il en soit, ces modules ouvrent des perspectives très intéressantes dans bien des domaines : caméra réseau, time-lapses, vision nocturne, surveillance, observation de la faune sauvage... Les idées de réalisations pratiques ne manquent pas.

Enfin, bien que relativement sceptique au départ, je dois avouer que j'ai été très agréablement surpris par la qualité des outils **raspivid** et **raspistill** ainsi que des options et fonctionnalités proposées. Pour quelques 30€, et je parle en particulier de la version NoIR, ce sont des heures et des heures de découvertes et d'expérimentations qui vous attendent, ce qui en termes de rapport prix/plaisir est plus que satisfaisant... **DB**

MODULE CAMÉRA RASPBERRY PI : UN MOT SUR L'INFRAROUGE

Denis Bodor

Comme vous l'avez constaté dans l'article précédent, une version spécifique du module caméra est vendue sans être équipée d'un filtre infrarouge, lui permettant ainsi de capter des rayonnements en dehors du spectre visible. En dehors de la capture d'images et de vidéos aux couleurs surprenantes, cette déclinaison du module permet un certain nombre de réalisations captivantes....



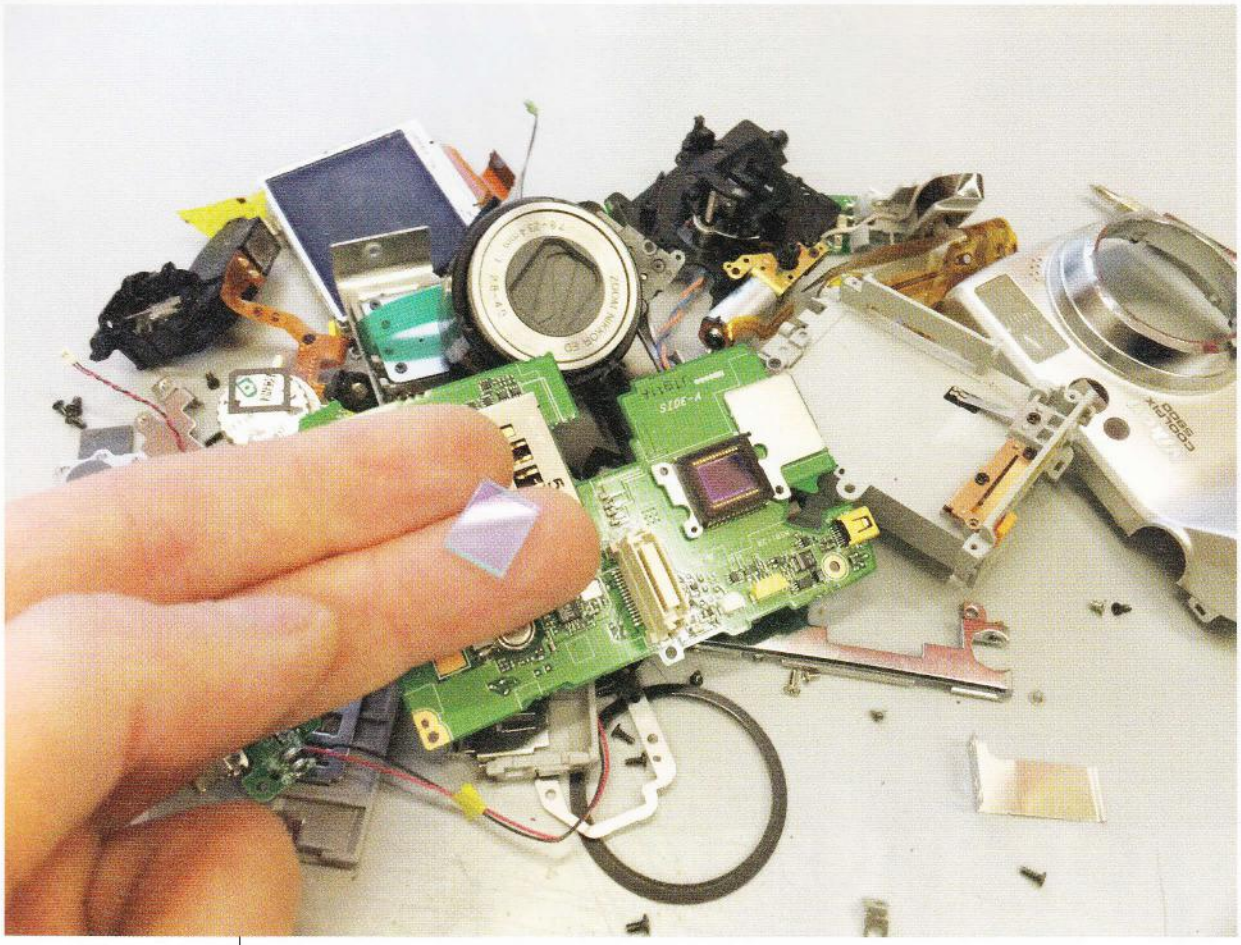
Voici un exemple de cliché pris avec le module caméra V2 en version NoIR équipé d'un filtre passe-haut. Cette image est composée de l'ensemble des réflexions des rayonnements infrarouges sur l'environnement. Notez la couleur des feuilles de l'arbre à droite, vert pour nos yeux, mais blanc pour la caméra, car les infrarouges ne sont pas absorbés par le feuillage, mais totalement réfléchis.

Le fait de pouvoir capter un rayonnement qui nous est normalement imperceptible a quelque chose d'incroyablement excitant puisque cela nous permet d'aller au-delà de nos capacités naturelles à interagir avec le monde qui nous entoure. Il faut savoir, en effet, que les deux choses gélatineuses placées dans votre tête ne permettent de capter qu'une infime portion des rayonnements qui bombardent l'univers et la terre.

Ce que nous pouvons percevoir se limite à la lumière visible, car nos yeux ne sont sensibles qu'à ce rayonnement clairement borné. La lumière est un rayonnement électromagnétique dont la longueur d'onde est comprise entre 400nm (nanomètre) et 700nm. Cette longueur d'onde est la distance séparant deux crêtes successives d'une onde périodique et elle est liée à la fréquence, qui est le nombre de crêtes comptées en une seconde. Plus la fréquence est importante, moins la longueur d'onde l'est et inversement. Mais il existe bien d'autres choses en dehors de ce que nous percevons naturellement...

1. LUMIÈRE, ULTRAVIOLET ET INFRAROUGE

Un rayonnement électromagnétique peut prendre plusieurs formes en fonction de sa fréquence et donc de sa longueur d'onde : rayon gamma, infrarouge, ondes radio, ultraviolets, micro-ondes, rayon X... Tout comme une antenne ne pourra percevoir que les ondes de la fréquence pour laquelle elle aura été conçue, nos yeux sont des capteurs de rayonnement électromagnétique ne fonctionnant que dans une gamme spécifique que nous appelons le spectre visible ou, tout simplement, la lumière.



Pour obtenir un miroir chaud et donc filtrer les infrarouges sur une caméra NoIR, le plus économique consiste à tout simplement démonter un appareil photo numérique hors d'usage et de récupérer celui se trouvant devant le capteur. La technique est simple : si vous voyez une vis, dévissez !

La sensibilité de l'œil humain à détecter différentes longueurs d'ondes et surtout à les différencier est telle que nous avons pris l'habitude, au fil du temps, de donner des noms à certaines plages spécifiques. Ainsi, nous avons le violet (400-446 nm), le bleu (446-500 nm), le vert (500-578 nm), le jaune (578-592 nm), l'orange (592-620 nm) et le rouge (620-700 nm).

Mais avant le violet nous avons un rayonnement invisible, l'ultraviolet et au-delà du rouge, nous avons l'infrarouge. Bien entendu, ce n'est pas parce que nous ne pouvons pas voir ces rayonnements que nous ne sommes pas impactés par eux. Le rayonnement ultraviolet par exemple, tantôt appelé lumière noire, nous permet de synthétiser la vitamine D à faible dose, mais peut également provoquer tumeurs, cancers et maladies à haute dose. Le rayonnement ultraviolet couvre les longueurs d'ondes de 380nm à 10nm et on parle souvent d'ultraviolets, au pluriel, tout en faisant la distinction entre UV-A (400 à 315 nm), UV-B (315 à 280 nm) et UV-C (280 à 100 nm).

Le rayonnement infrarouge à longueur d'onde supérieure à celle de la lumière visible couvre une bande entre 780 nm et 5 mm. Là aussi, on découpe généralement cette gamme en plusieurs morceaux :

- infrarouge proche (PIR) : 780 à 3000 nm,
- infrarouge moyen (MIR) : 3000 à 50 µm,
- infrarouge lointain (LIR) : 50 µm à 5 mm.

Tout comme les rayonnements UV, les infrarouges (IR) existent de façon naturelle avec comme source la plus puissante à proximité : le soleil (qui nous envoie encore bien d'autres ondes électromagnétiques). Mais nous avons également domestiqué ces rayonnements et les infrarouges lointains, par exemple, sont utilisés pour le chauffage sous la forme de panneaux rayonnants. Comme ce rayonnement est lié à l'émission de chaleur, inversement, nous disposons aussi de capteurs sensibles à ces longueurs d'ondes, permettant ainsi la thermographie infrarouge.

Mais plus proche de nous, c'est généralement l'infrarouge proche qui est utilisé, comme par exemple pour les capteurs de mouvements ou encore les télécommandes (voir *Hackable n°3*), mais aussi pour certains systèmes de vision nocturne.

2. C'EST BIEN JOLI TOUT ÇA, MAIS C'EST QUOI LE RAPPORT ?

Nos yeux ne captent pas les rayonnements ultraviolets ou infrarouges, mais les capteurs CCD (*Charge Coupled Device*) et CMOS (*Complementary Metal Oxide Semiconductor*) utilisés dans les équipements électroniques comme les webcams ou les appareils photo numériques sont sensibles à bien plus que la simple lumière visible.

Pour éviter qu'ils ne captent les rayonnements de toutes les longueurs d'onde, ces capteurs sont donc placés derrière des filtres bloquant les rayonnements indésirables et permettant d'obtenir une image avec des couleurs les plus fidèles possible (fidèles à ce que nous voyons, pas à ce qui existe). Les filtres UV sont généralement placés, au besoin, devant l'objectif, mais les filtres infrarouges, également appelés miroirs chauds, sont intégrés au boitier du capteur, directement devant la surface sensible.

Une technique photographique consiste à démonter un appareil photo numérique afin d'accéder au capteur et à remplacer le miroir chaud se trouvant contre le capteur par un verre totalement transparent. Cette manipulation hasardeuse et très risquée, permet alors à l'APN de prendre des photos dans une plage de longueurs d'ondes bien supérieure à celle de la lumière visible et de produire des clichés d'une beauté troublante. C'est une modification que j'avais faite il y a une dizaine d'années sur un Nikon Coolpix 4300 et le jeu en vaut vraiment la chandelle (mais je ne m'y risquerai cependant pas avec mon Canon EOS 700D et encore moins avec celui de la rédaction).

En l'absence de miroir chaud ou de filtre, un capteur optique est donc en mesure de recevoir un rayonnement invisible pour nos yeux et c'est précisément ce que propose la version NoIR du nouveau module caméra Raspberry Pi.

Attention cependant aux raccourcis malheureux. En effet, il est presque systématique de voir ce module désigné comme étant une version « vision nocturne » et l'emballage officiel lui-même est trompeur puisqu'il fait mention de « Low Light Photography » (photographie à faible lumière). L'absence de filtrage des infrarouges n'est en aucun cas synonyme de vision dans l'obscurité totale au sens commun du terme. Certes, au sens strict, la « lumière » est le rayonnement visible et de ce fait l'obscurité est le fait de son absence, mais sans source de rayonnement infrarouge, le module ne captera rien du tout.

Il ne faut donc pas confondre différentes choses souvent regroupées sous les termes « vision nocturne » (vision scotopique) :

- intensification de lumière : peu de lumière visible est nécessaire à la capture d'images grâce à l'utilisation d'un photomultiplicateur ;
- vision thermique : les sources de chaleur, moteurs, radiateurs, corps, etc., émettent des infrarouges lointains captés par un dispositif dédié ;

Ce filtre Edmund Optics est opaque pour tout rayonnement inférieur à 700 nm. Il apparaît donc noir pour nos yeux ou une caméra standard, mais transparent pour le module NoIR. Les couleurs et en particulier celles du filtre découlent de l'ajustement automatique de la caméra. Une fois le filtre couvrant totalement l'optique, le rendu s'équilibrera naturellement et l'image ne sera plus rosâtre.

- infrarouge proche : système utilisé par le module Raspberry Pi ou un APN modifié, mais nécessitant une émission infrarouge dans cette longueur d'onde pour « éclairer » le sujet ou la scène observée.

Les deux premières solutions nécessitent du matériel coûteux et fragile, alors que la troisième ne demande que l'absence d'un dispositif destiné à empêcher ce comportement. C'est ça et uniquement ça que vous obtiendrez avec le module NoIR, et non ce à quoi on pense instinctivement lorsqu'on dit « vision nocturne » (ces jouets-là sont généralement réservés aux militaires, aux chasseurs ou aux psychopathes qui veulent tuer Clarice Starling).

3. LE MODULE NOIR

Comme je l'ai dit, le module en version NoIR est strictement identique à la version standard, à la différence que le filtre IR est absent entre la lentille et le capteur CMOS Sony IMX219.

L'installation et la mise en œuvre du module sont donc identiques avec les deux versions. Bien entendu, ce sont les mêmes outils, **raspistill** et **raspivid**, qui seront utilisés pour les prises de vue et tout ce qui a été dit dans l'article précédent est parfaitement valable ici.

La seule option pouvant être utile ou spécifique concerne l'exposition avec **-ex** et son paramètre **night** ou **nightpreview** qui, tantôt, améliorera très sensiblement la qualité du rendu final.

4. FILTRES

Voilà la partie la plus intéressante de la mise en œuvre du module en version NoIR. Aucun filtre n'est présent dans le module, ce qui signifie que vous pouvez placer le filtre de votre choix devant l'objectif.

Deux types de filtres peuvent ainsi être utilisés : passe-bas ou passe-haut. Le premier laissera passer les rayonnements d'une longueur d'onde inférieure à la valeur spécifiée par le filtre. L'infrarouge proche débutant à 780 mn, en plaçant un filtre ne laissant passer que les rayonnements d'une longueur d'onde inférieure à 700 nm, vous ne capterez que la lumière visible (et un peu d'UV). De ce fait, vous pourrez rapidement transformer votre module NoIR en module standard.

Ces filtres, désignés sous « filtre de suppression IR », « filtre passe-bas IR » ou « miroir chaud IR » sont relativement coûteux (environ 40€) que ce soit sur les sites d'enchères en ligne, les boutiques pour photographes ou encore celles pour les amateurs d'astronomie. La solution la plus économique consiste à faire de la récupération. Comme je l'ai dit plus haut, ces filtres sont présents dans tout ce qui possède un capteur CCD ou CMOS. Tout ce que

vous avez à faire est de mettre la main sur un matériel de ce type, même hors d'usage, et le désosser pour en extraire le filtre.

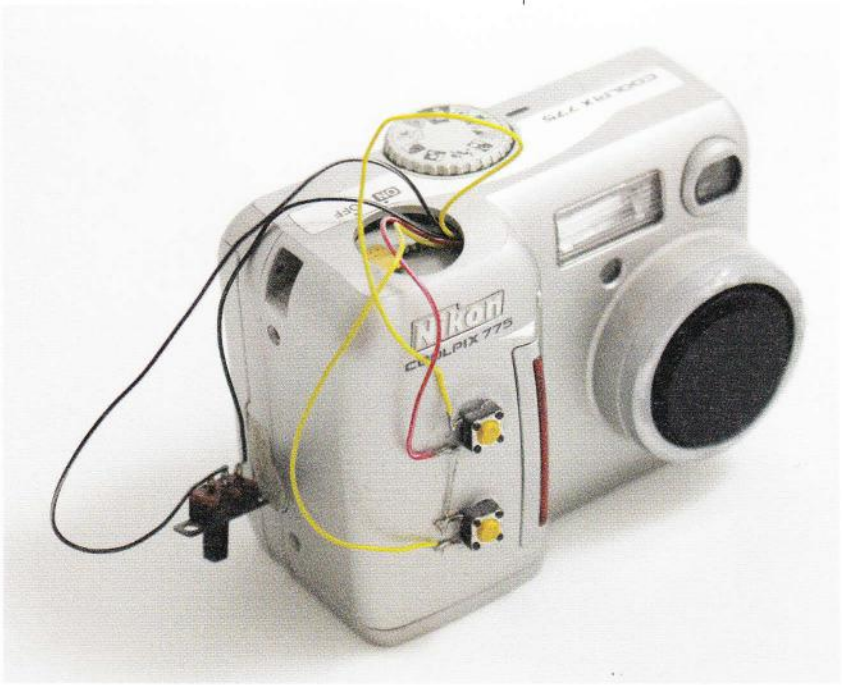
La plupart des appareils numériques utilisent un miroir chaud qui est simplement un petit bloc de verre traité en surface pour laisser passer la lumière visible et réfléchir les rayonnements infrarouges (et éventuellement absorber les UV).

L'optique du module caméra Raspberry Pi officiel est relativement petite, sinon minuscule (et très fragile). Ce désavantage en termes de prise de vue joue ici en notre faveur, car n'importe quelle taille de filtre pourra s'adapter devant l'objectif. Il vous suffit donc de vous rendre sur un site d'enchères en ligne (au hasard eBay) de parcourir la catégorie « *Appareils photo numériques* » et d'affiner les résultats en limitant les annonces à l'état « *Pour pièces détachées/ne fonctionne pas* » et en triant par « *Prix+Livraison : les moins chers* ».

C'est une loterie, car même si la liste d'annonces à moins de 5 € port compris est énorme, vous ne saurez pas dans quelle mesure l'extraction du filtre sera ou non aisée. Parfois, le miroir chaud est simplement coincé ou vissé, mais tantôt vous pouvez tomber sur des éléments collés, plus problématiques. La taille du miroir sera forcément proportionnelle à celle du capteur CCD ou CMOS : plus il y a de mégapixels, plus le capteur a de chances d'être grand et plus le miroir le sera également. De plus, il n'est pas impossible qu'en démontant le matériel vous vous rendiez compte qu'un connecteur est simplement débranché en interne et pourrez donc potentiellement transformer votre achat en appareil photo infrarouge fonctionnel par la même occasion...

Évitez cependant les webcams USB, comprenant des capteurs minuscules et relativement difficiles à démonter. Dans tous les appareils que j'ai eu le grand plaisir à démonter, le filtre IR était soit collé au capteur CCD/CMOS, soit inclus dans l'objectif. Vous aurez peut-être plus de chance que moi, mais je préfère vous avertir : évitez les webcams.

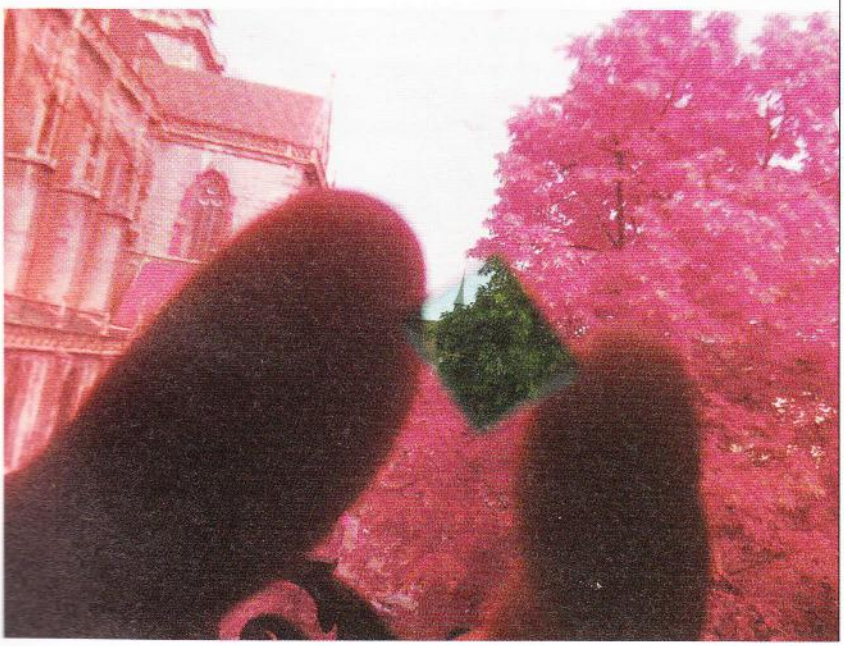
L'utilisation d'un miroir chaud, neuf ou de récupération vous fournira des images « standards » en lumière visible et donc en couleur parfaitement « normales ». Mais il est également possible de faire exactement l'inverse : bloquer la lumière visible et ne laisser que le rayonnement infrarouge atteindre le capteur CMOS. Pour cela, vous devrez utiliser un filtre passe-haut qui,



Voici le résultat de ma première expérience avec l'imagerie infrarouge : un appareil photo numérique Nikon modifié auquel j'ai retiré le miroir chaud et que j'ai équipé d'un filtre passe-haut devant l'objectif. Les boutons en façade découlent d'une destruction malheureuse du déclencheur. Ce hack a quelques 10 ans d'âge, mais est toujours fonctionnel et reproductible actuellement pour quelques euros. En démontant un APN fonctionnel pour retirer le miroir chaud, vous pourrez vous confectionner, par la même occasion, un APN infrarouge.

Le démontage d'un appareil photo numérique permet de récupérer le miroir chaud intégré, mais également quelques lentilles pouvant éventuellement être utilisées avec le module caméra en bricolant un peu (en particulier si, comme moi, vous avez endommagé l'objectif en essayant d'ajuster la mise au point... Bravo Denis ! Clap, clap, clap !).

Le miroir chaud coupant le passage des rayonnements infrarouges, placé devant l'objectif du module NoIR, laisse entrevoir le rendu des couleurs « normales ». Alterner entre ce type de filtre et un filtre passe-haut permet ainsi de choisir la gamme de longueurs d'ondes à utiliser. De là à imaginer un dispositif permettant de changer le filtre avec un servomoteur contrôlé par la Raspberry Pi, il n'y a qu'un pas...



comme son nom l'indique, ne laissera passer que les rayonnements supérieurs à une certaine longueur d'onde tout en bloquant les autres.

Il est ici également possible de faire de la récupération, mais malheureusement pas dans une qualité qu'on pourrait juger propre à une utilisation en optique. En effet, les équipements hifi de type téléviseurs, magnétoscopes ou décodeurs sont généralement contrôlés à l'aide d'une télécommande infrarouge. En façade de ces équipements se trouve donc une diode cachée derrière un support en plastique. Noir pour l'œil humain, ce cache en plastique est généralement totalement transparent pour les rayonnements infrarouges, puisque le capteur doit pouvoir recevoir les signaux de la télécommande.

Il est donc possible de recycler ces caches en plastique pour s'en servir comme filtre. Le type même de matière qui les constitue cependant n'est pas destiné à une utilisation optique, ils sont donc facilement rayés, dépolis ou déformés, réduisant grandement la qualité de l'image. Ce n'est donc pas une bonne piste.

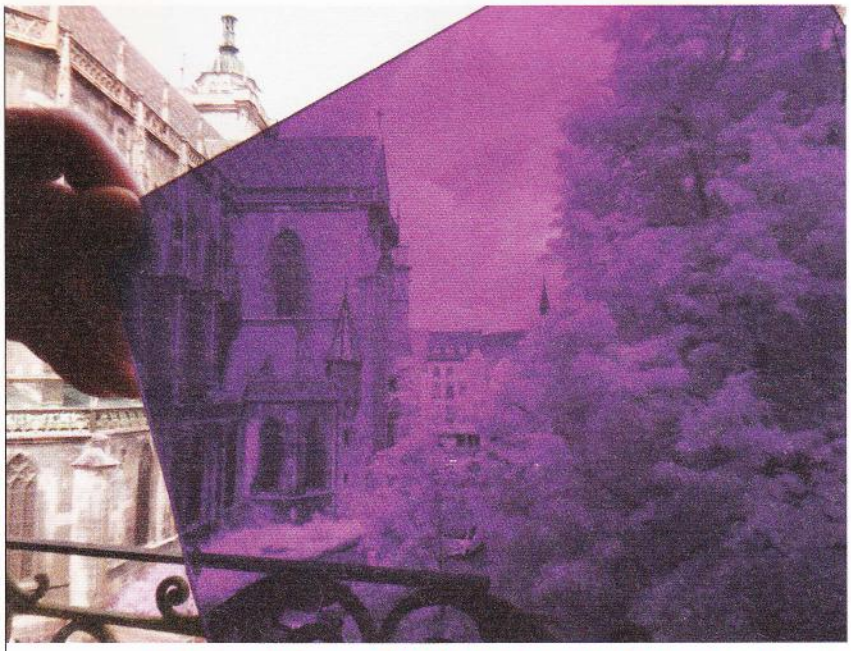
Toutefois, une bonne nouvelle vous attend si vous regardez du côté des fournitures pour photographes et autres fabricants d'éléments optiques, vous vous rendrez rapidement compte qu'un filtre passe-haut n'est pas du tout dans la même gamme de prix que les filtres passe-bas ou passe-bande. Un des leaders dans la distribution de composants optiques, Edmund Optics, propose ainsi

sur sa boutique en ligne (**www.edmundoptics.fr**) des filtres de petite taille (disques de 2,54 cm de diamètre) pour moins de 9€.

Vous pouvez également trouver ce type de filtres dans d'autres boutiques en ligne et sur les sites d'enchères mais, de façon assez étonnante, je n'ai pas trouvé moins cher que ce distributeur réputé, pour une pièce unique. Il y a certes des vendeurs proposant des filtres carrés de grande taille (10 cm de côté) à moins de 30€, mais encore faut-il pouvoir les découper sans les endommager pour arriver à ses fins (en supposant que vous ayez plusieurs modules à équiper bien sûr).

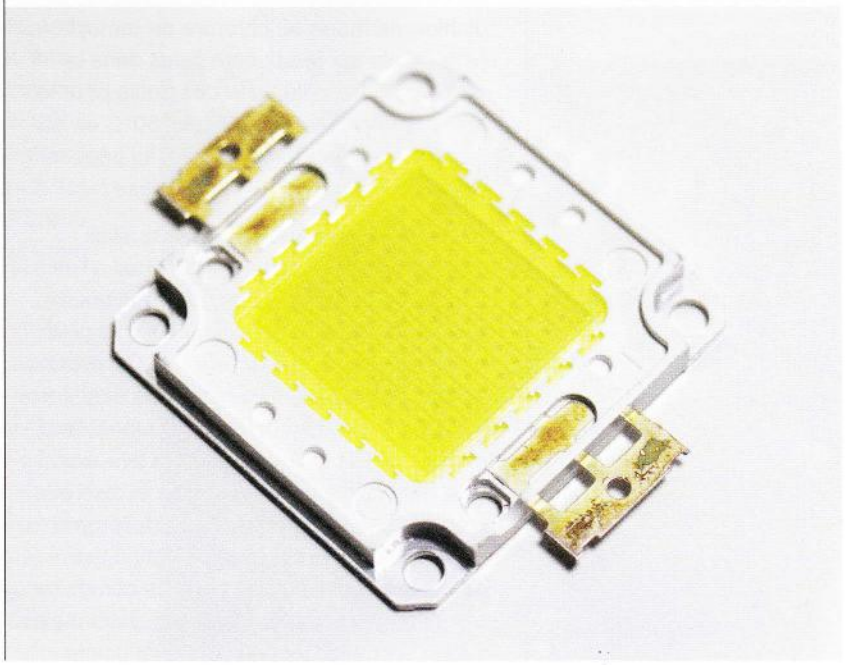
Une dernière solution est encore plus économique, mais se prêtera plutôt à la dissimulation qu'à un usage en filtre : le Plexiglas. Je ne parle pas ici du polyméthacrylate de méthyle (PMMA) générique, mais bel et bien de la marque commerciale de cette matière. J'ai été très surpris de constater que le fabricant original de ce matériau dispose d'une boutique en ligne (**www.plexiglas-shop.com**) et vend directement aux particuliers un nombre incroyable de déclinaisons du produit tant en couleurs qu'en formes. Il y a même un service de découpe intégré vous permettant de spécifier les dimensions souhaitées avec ou sans polissage des bords.

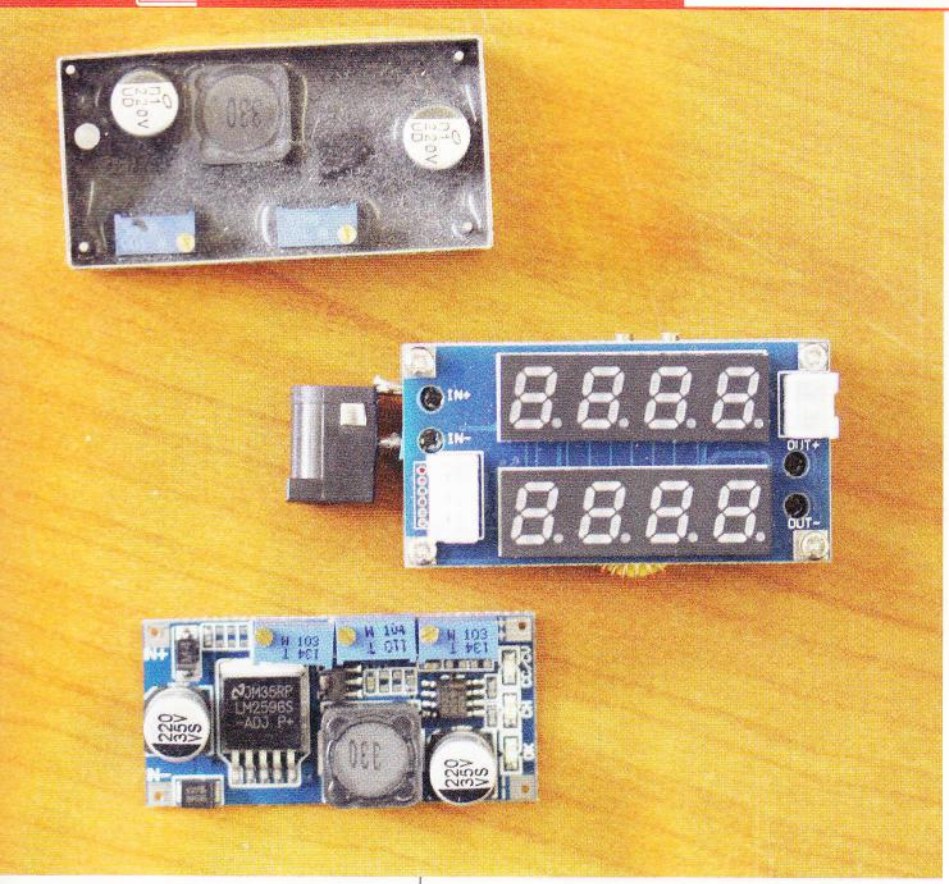
Parmi les produits proposés, il en est un qui se prête parfaitement à une utilisation avec un module caméra NoIR, le « PLEXIGLAS GS Noir 9C20 GT », disponible en plaque de 3 mm d'épaisseur au prix de



Une plaque de Plexigas GS 9C20 GT est opaque à la lumière visible, mais transparente aux infrarouges. Bien qu'il ne s'agisse pas d'une solution viable pour faire office de filtre, ce type de matière peut être utilisé pour dissimuler la caméra et la Raspberry Pi.

Ce type de led, encore relativement cher, est de plus en plus populaire. Celle-ci émet une lumière blanche très intense en consommant quelques 100 watts. Il existe de nombreuses déclinaisons en couleurs ainsi que des versions ultraviolettes et infrarouges, aussi bien en 850nm qu'en 940nm.





Pour alimenter une led de 1 à 5 watts, il est nécessaire d'utiliser un circuit spécialisé permettant de fournir un courant précis au composant. Ici, trois versions différentes de ce genre de module d'alimentation. De haut en bas : un modèle ancien moulé dans la résine, une version récente à 10€ équipée d'un afficheur pour le courant et la tension et un modèle à une paire d'euros permettant aussi bien d'alimenter une led que de servir de chargeur d'accumulateur.

219,23€ le mètre carré TTC, ce qui donne 12,78 € pour un rectangle de 20 cm par 10 cm (taille minimale).

Ce matériau peut être collé avec un solvant comme de l'acétone ou mieux, du dichlorométhane ou chlorure de méthylène (voir article sur les *bubble lights* dans le numéro 9), travaillé avec des outils pour le bois et même thermoformé à 160°C en utilisant un simple four et de quoi lui imposer une forme avant qu'il ne refroidisse.

Certes, il ne s'agit pas d'une qualité « optique » et vous aurez grand mal à l'utiliser en guise de filtre, mais ce que vous pourrez faire en revanche c'est vous en servir pour masquer votre caméra. Le 9C20 GT est totalement opaque à la lumière visible et absorbe les UV, mais transmet les infrarouges. Une Raspberry Pi placée derrière un tel « écran » sera donc parfaitement invisible et discrète, mais le module NoIR pourra parfaitement voir au travers. C'est exactement ce type de matière qui est utilisé pour les dômes de caméras de sécurité motorisées, évitant ainsi à tout un chacun de voir vers quoi pointe l'objectif. Avec une plaque de Plexiglas 9C20 GT vous pourrez faire de même, mais en grand, en beaucoup plus grand !

5. « ÉCLAIRAGE » INFRAROUGE

Nous venons de voir comment filtrer les rayonnements et dissimuler notre caméra, mais encore faut-il que nous ayons quelque chose à voir. De jour, et en utilisant un filtre passe-haut, il n'y a pas de problème. La grosse boule de feu nucléaire qui nous sert d'étoile et de chauffage central arrose littéralement la planète de rayonnement en tous genres et d'infrarouges. Nous n'avons donc pas besoin de source d'appoint.

Il n'en va cependant pas de même lorsque notre cher astre lumineux est couché. Certes, les sources lumineuses classiques comme les ampoules à incandescence et les leds émettent également des infrarouges proches, mais dans ce cas nous n'avons pas besoin d'une caméra infrarouge. Non, ce qui est véritablement intéressant c'est justement de n'avoir aucune source de lumière et un rayonnement invisible pour « éclairer » notre sujet.

Il existe des projecteurs à leds infrarouges spécialement destinés aux applications de vidéosurveillance, mais la solution la plus économique est tout simplement d'en construire un soi-même.

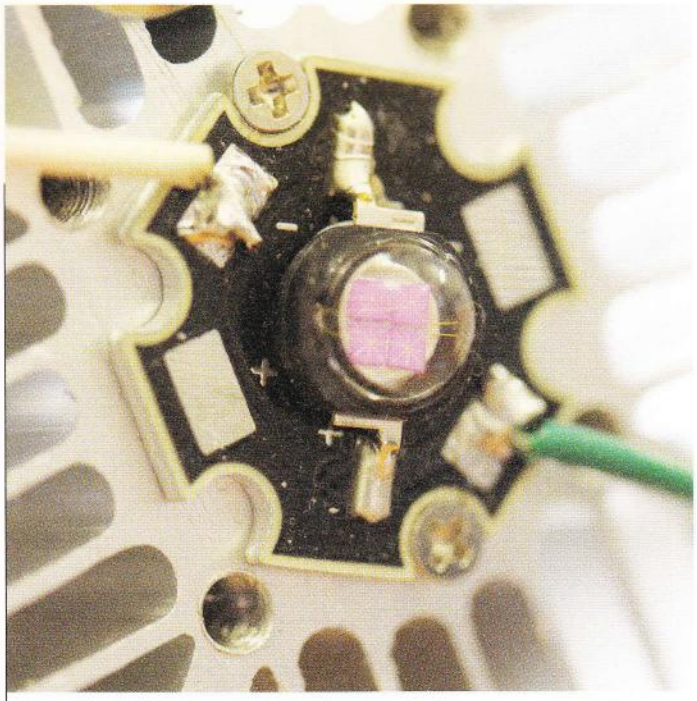
Des leds capables d'émettre un rayonnement infrarouge d'une longueur d'onde de 850nm ou

940nm existent sous plusieurs formes. Pour ce genre d'application, il conviendra d'éviter ce qu'on appelle les leds de signalisation principalement destinées à la transmission de codes infrarouges comme pour les télécommandes. Ce qui nous intéresse ici est davantage des leds de puissance très différentes de celles que vous utilisez par exemple pour vos montages.

Nous en avons déjà parlé dans le magazine (numéro 3) lorsque nous avons construit une super télécommande universelle capable d'éteindre n'importe quel téléviseur à une grande distance. Le principe est ici le même, ces leds de puissance sont capables de fournir un rayonnement très important avec une consommation entre 1 et 5 watts sous la forme d'un petit composant monté sur un circuit imprimé dédié.

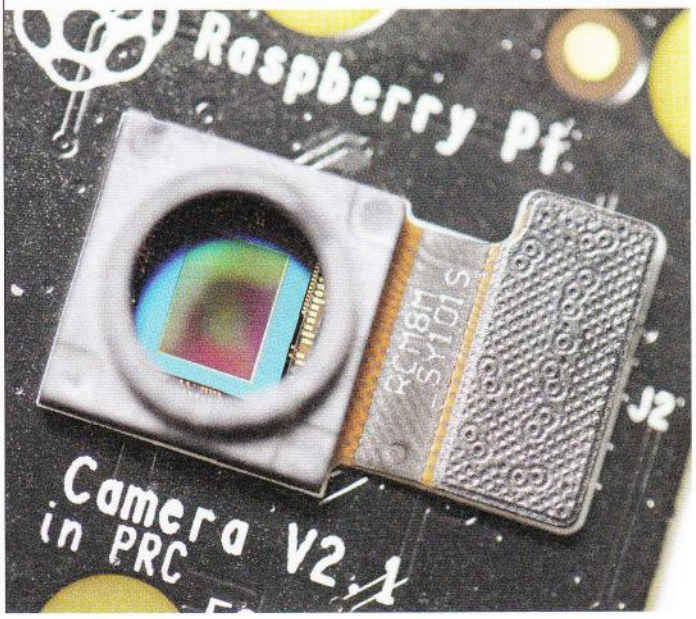
Mais depuis mon article fin 2014, un autre type de leds s'est énormément popularisé. Regroupant sur un support carré de un pouce de côté (2,54 cm) entre 25 et 100 leds, ces composants se déclinent en plusieurs couleurs et bien entendu en plusieurs longueurs d'onde dans l'infrarouge. Alimentés avec une tension de 14 à 17 volts ces « lampes » utilisent un courant de 1A à 3,5A, consommant ainsi entre 30W et 100W. Bien entendu, toute cette énergie n'est pas convertie entièrement en rayonnement infrarouge et le rendement est loin d'être optimal. Ces composants doivent généralement s'attacher à un dissipateur thermique (radiateur) exactement comme un processeur de PC afin de ne pas les détruire presque immédiatement lors de la mise sous tension.

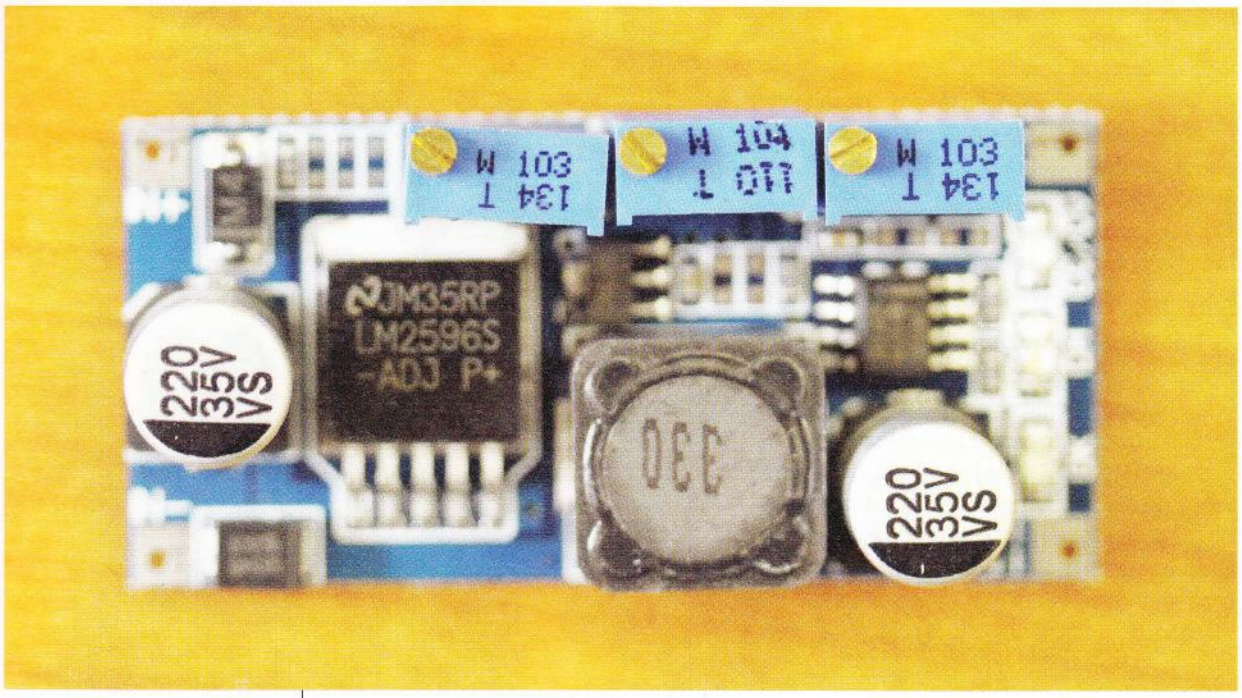
Il faut également savoir que ce type d'élément est encore relativement cher : une telle lampe de 100W en 940nm vous coûtera la bagatelle de 65€, alors que la déclinaison en 30W quelques 26€. On réservera donc ce type de composants à des applications bien précises comme l'éclairage d'un jardin, un parking ou d'une cour. Pour des usages en intérieur, les leds consommant entre 1 et 5 watts apportent généralement des résultats très satisfaisants. Il est relativement aisé d'en trouver sur eBay



Voici une led de forte puissance de 4 watts émettant dans l'infrarouge avec une longueur qu'onde de 850 nm. Le Canon EOS 700D ayant servi à prendre cette photo est légèrement sensible aux infrarouges et capture une faible lueur rose alors qu'en réalité, pour nos yeux, la lumière émise est rouge, mais également d'une très faible intensité. Cette lueur n'est pas perceptible avec les leds en 940 nm.

En dévissant (de force) l'optique d'un module caméra NoIR, on découvre le capteur CMOS Sony IMX219 ainsi que l'absence de filtre infrarouge. Je ne vous recommande pas de le faire, car même si cela peut permettre d'ajuster la mise au point, l'opération est extrêmement risquée et la lentille de l'optique se raye avec une facilité déconcertante...





Gros plan sur un module d'alimentation « DC-DC Step-down CC/CV » utilisant un LM2596. Les trois potentiomètres en haut permettent respectivement de régler la tension, le niveau de détection de charge (pour une utilisation en chargeur) et le courant. Ce type de module existe sous bien des formes avec des prix entre 2€ et 8€.

pour quelques euros (4 watts), ce à quoi il sera nécessaire d'ajouter un radiateur et de la pâte ou colle thermique. Le budget total s'élèvera donc à une bonne dizaine d'euros.

Vous vous en doutez, avec ce genre de puissance il n'est pas question de simplement limiter le courant avec une résistance. Il est donc nécessaire, que ce soit pour les leds de quelques watts ou celles de forte puissance, d'utiliser un circuit de régulation. Souvent désignés par le nom de *led drivers* ces circuits permettent de régler l'alimentation de la led avec un courant fixe.

Là encore, depuis mon article quelques petites choses ont changé. En un peu plus d'une année, la popularisation des leds de puissance s'est généralisée aussi bien dans les produits finis que sous forme de composants. De ce fait, les circuits électroniques destinés à réguler leur alimentation ont vu leur prix baisser et leurs fonctionnalités s'étoffer.

Les modules d'alimentation simples permettant un réglage de la tension ou du courant se trouvent pour moins de 2€ sur le Web. Il vous suffit de chercher « DC-DC Step-down CC/CV » sur un site d'enchères en ligne (eBay) ou un site de vente directe (DX, BangGood, etc.). Cette désignation cryptique décrit un convertisseur (*Set down*) en courant continu (*DC-DC*) contrôlable en courant constant (CC) ou en tension constante (CV). Les intitulés de produits inclus également souvent le nom du circuit intégré principal en œuvre : LM2596.

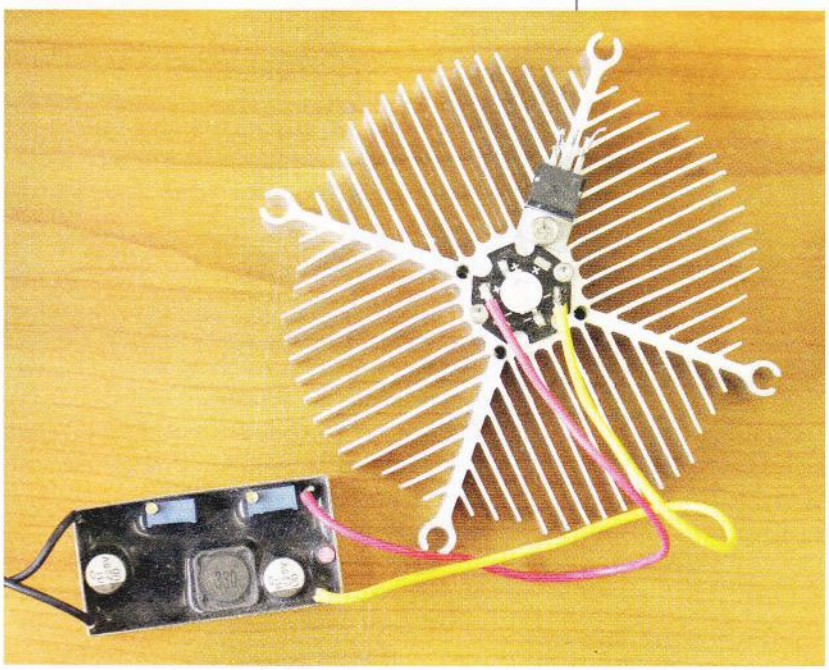
L'utilisation de ces modules d'alimentation nécessite l'usage d'un multimètre pour procéder aux réglages. Quel que soit le module utilisé, sa forme et sa couleur, il dispose en général de deux potentiomètres : un pour régler le courant et un autre pour la tension. Une led intégrée indique également si le module est en mode limitation de courant ou de tension (généralement allumé pour indiquer la limitation de courant).

Il n'est pas nécessaire de disposer de deux multimètres pour procéder au réglage (un pour le courant et l'autre pour la tension) bien que cela soit plus facile. Avec un seul multimètre, la procédure est la suivante :

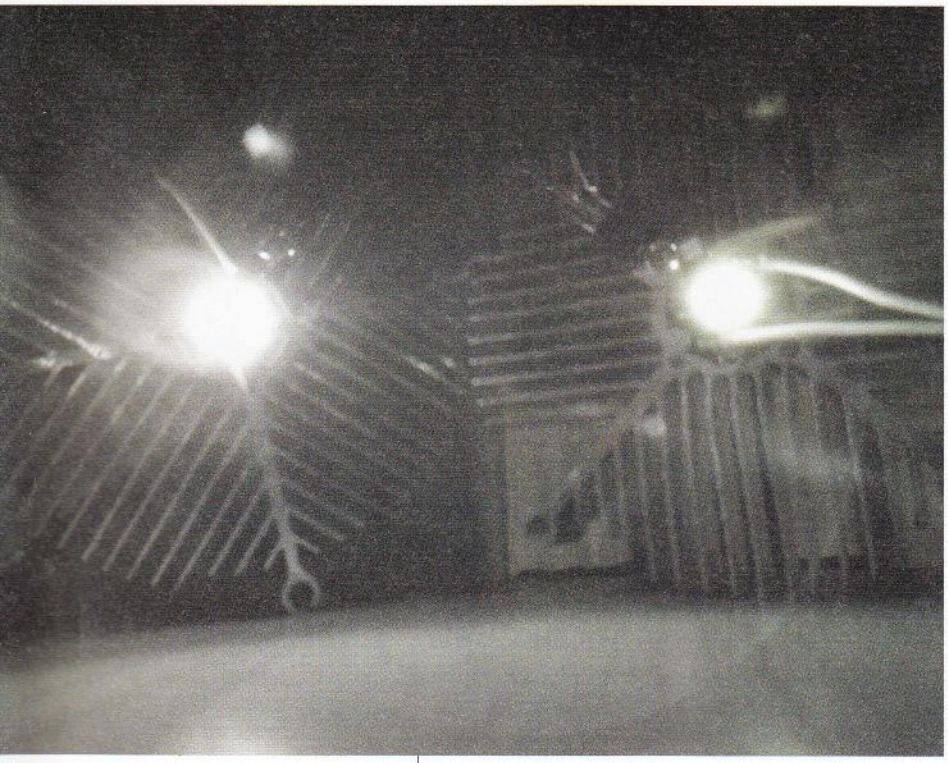
- prenez connaissance du courant (If) et de la tension maximum (Vf) utilisés par votre led de puissance,
- connectez le module d'alimentation à une source électrique,
- mesurez la tension en sortie et ajustez le réglage de la tension jusqu'à arriver à la tension de la led,
- passez votre multimètre en mode de mesure de courant et connectez la sonde rouge à la broche « 10A » du multimètre,
- mesurez le courant en court-circuit en sortie du module d'alimentation et ajustez jusqu'à arriver à la valeur désirée,
- la led du module signalant la limitation de courant doit s'allumer,
- déconnectez la source d'alimentation et connectez votre led, sur son radiateur, au module,
- mettez à nouveau sous tension et mesurez la tension aux bornes de la led, celle-ci doit être inférieure ou égale à celle de sa documentation,
- éventuellement, débranchez l'anode de la led et mesurez le courant circulant entre le module et la led pour vous assurer de fournir le courant optimal.

Certains modules à base de LM2596 disposent d'un troisième potentiomètre permettant de régler la tension de coupure pour une utilisation en charge d'accumulateur. Bien que disposant de ce type de modules, je dois avouer qu'étant donné le danger en cas de dysfonctionnement (et le coût d'un accu), je n'utilise jamais cette fonctionnalité, préférant faire confiance à un chargeur de qualité plutôt qu'à un module chinois à une paire d'euros.

Il existe également des versions plus avancées autour de 10€ intégrant directement un voltmètre et un ampèremètre affichant les valeurs réglées, ce qui vous évitera de jongler avec un multimètre dans plusieurs modes de mesure. Notez également un point très important : ces modules à afficheur sont généralement constitués d'un module d'alimentation et d'un circuit supplémentaire de mesure et d'affichage. L'interconnexion des deux circuits se fait par l'intermédiaire des entretoises qui les maintiennent ensemble ! En clair, les vis et entretoises sont connectées à la masse et à la sortie du module de régulation. Ne les utilisez donc pas pour fixer le module sur un support conducteur sous peine de provoquer un court-circuit.



Cette led de 1W est montée sur un radiateur et alimentée par un module avec un courant constant. C'est exactement le même montage que nous avons utilisé pour simuler une télécommande dans un précédent numéro, si ce n'est que la led infrarouge est alimentée de façon permanente.



Cette image a été capturée avec le module NoIR équipé d'un filtre passe-haut et présente à gauche une led de 4W émettant sur une longueur d'onde de 850nm et à droite une led de 1W sur 940nm. On constate clairement que la caméra est sensible aux deux longueurs d'ondes, contrairement à certaines netcams destinées à la vidéosurveillance.

Pour un budget entre 5€ et 20€, vous pourrez vous construire un système d'illumination infrarouge pour une pièce. Le choix des leds de puissance est une question de préférence et de coût. D'après les tests réalisés, le module caméra NoIR est capable de capter des rayonnements d'une longueur d'onde entre 750nm et 940nm.

Notez cependant que la valeur généralement utilisée, 850nm, provoque une faible lueur rouge sur les leds. La proximité avec le spectre de lumière visible et le fait que le rayonnement n'est pas borné à une longueur d'onde unique impliquent qu'une led dans l'infrarouge très proche aura toujours une faible lueur rouge. Si vous optez, en revanche, pour du 940 nm, votre « projecteur » sera parfaitement et totalement invisible (sauf éventuellement avec un appareil photo). Ceci peut être très important pour un système de surveillance de nuit ou éventuellement pour l'observation de la faune animale sauvage.

CONCLUSION

Pour terminer, je préciserai tout d'abord que je n'ai pas d'actions chez Edmund Optics ou Plexiglas/Evonik, il s'avère tout simplement que mes recherches m'ont conduit à l'utilisation de ces produits qui s'avèrent de bonne qualité. Ensuite, et c'est plus important encore, je me dois de préciser que les explications données dans cet article ne sont en rien une excuse vous permettant de justifier d'un comportement illégal ou même simplement douteux.

Une législation encadre l'usage des caméras de vidéoprotection et bien qu'il soit possible d'installer ce type de matériel à son domicile pour protéger ses biens, sa santé et sa vie et même, dans une certaine mesure, couvrir un espace public, il est important de vous renseigner sur vos droits et leurs limites.

Au-delà même du champ purement légal, cet article n'a pas pour objectif de vous permettre d'espionner quelqu'un à son insu qu'il s'agisse d'un(e) inconnu(e) ou qu'une personne que vous connaissez. Montrez-vous respectueux de la vie privée de vos congénères et restez raisonnable dans l'utilisation de ces technologies. Il y a suffisamment de choses intéressantes à faire dans le domaine de la prise de vue infrarouge, tant techniquement qu'artistiquement, sans pour autant sombrer dans la perversion et le voyeurisme. **DB**

PROFESSIONNELS DES TICE, COLLECTIVITÉS, ÉCOLES D'INGÉNIEURS, UNIVERSITÉS, R & D, ENSEIGNANTS, ...

HACKABLE MAGAZINE

PROFESSIONNELS !

DÉCOUVREZ NOS OFFRES D'ABONNEMENTS ...

...EN VOUS CONNECTANT À L'ESPACE DÉDIÉ AUX PROFESSIONNELS SUR :

www.ed-diamond.com

PDF COLLECTIFS PRO

OFFRE	ABONNEMENT	1 - 5 lecteurs Réf	1 - 5 lecteurs Tarif TTC	6 - 10 lecteurs Réf	6 - 10 lecteurs Tarif TTC	11 - 25 lecteurs Réf	11 - 25 lecteurs Tarif TTC
PROHK2	6n° HK	☐ PRO HK2/5	156,-	☐ PRO HK2/10	312,-	☐ PRO HK2/25	624,-

PROFESSIONNELS : N'HÉSITEZ PAS À NOUS CONTACTER POUR UN DEVIS PERSONNALISÉ PAR E-MAIL : abopro@ed-diamond.com OU PAR TÉLÉPHONE : 03 67 10 00 20

ACCÈS COLLECTIFS BASE DOCUMENTAIRE PRO OPEN SILICIUM

OFFRE	ABONNEMENT	1 - 5 connexion(s) Réf	1 - 5 connexion(s) Tarif TTC	6 - 10 connexions Réf	6 - 10 connexions Tarif TTC	11 - 25 connexions Réf	11 - 25 connexions Tarif TTC
PROOS+3	OS	☐ PRO OS+3/5	90,-	☐ PRO OS+3/10	180,-	☐ PRO OS+3/25	360,-
PROH+3	GLMF + HS + LP + HS + MISC + HS + OS	☐ PRO H+3/5	447,-	☐ PRO H+3/10	894,-	☐ PRO H+3/25	1788,-

Les abréviations des offres sont les suivantes : LM = GNU/Linux Magazine France HS = Hors-Série LP = Linux Pratique OS = Open Silicium

SÉLECTIONNEZ VOTRE OFFRE DANS LA GRILLE CI-DESSUS ET RENVOYEZ CE DOCUMENT COMPLET À L'ADRESSE CI-DESSOUS !

Voici mes coordonnées postales :	
Société :	
Nom :	
Prénom :	
Adresse :	
Code Postal :	
Ville :	
Pays :	
Téléphone :	
E-mail :	

☐ Je souhaite recevoir les offres promotionnelles et newsletters des Éditions Diamond.

☐ Je souhaite recevoir les offres promotionnelles des partenaires des Éditions Diamond.

En envoyant ce bon de commande, je reconnais avoir pris connaissance des conditions générales de vente des Éditions Diamond à l'adresse internet suivante : boutique.ed-diamond.com/content/3-conditions-generales-de-ventes et reconnais que ces conditions de vente me sont opposables.

HACKABLE MAGAZINE

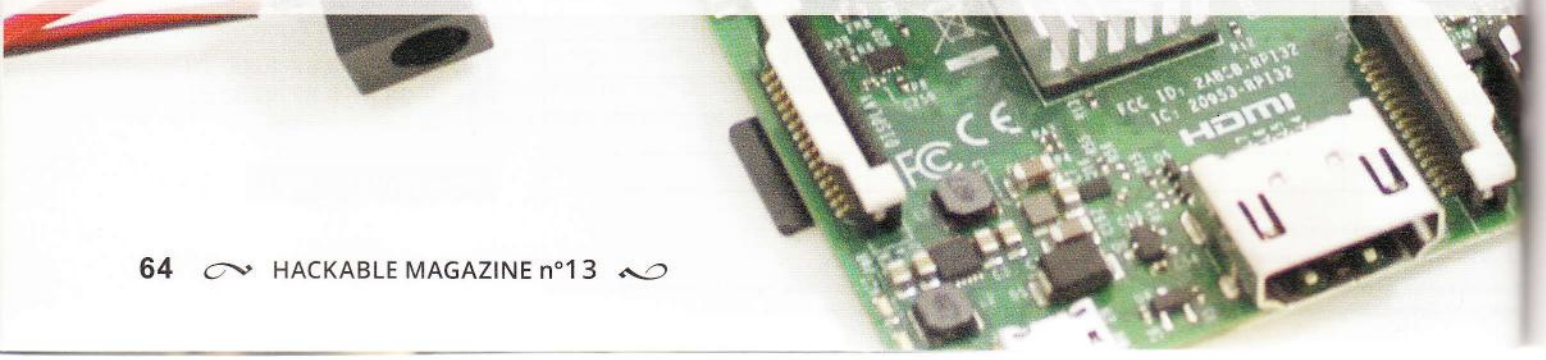
Les Éditions Diamond
Service des Abonnements
10, Place de la Cathédrale
68000 Colmar – France
Tél. : + 33 (0) 3 67 10 00 20
Fax : + 33 (0) 3 67 10 00 21

Vos remarques :

UTILISEZ UNE IMPRIMANTE THERMIQUE AVEC VOTRE RASPBERRY PI

Denis Bodor

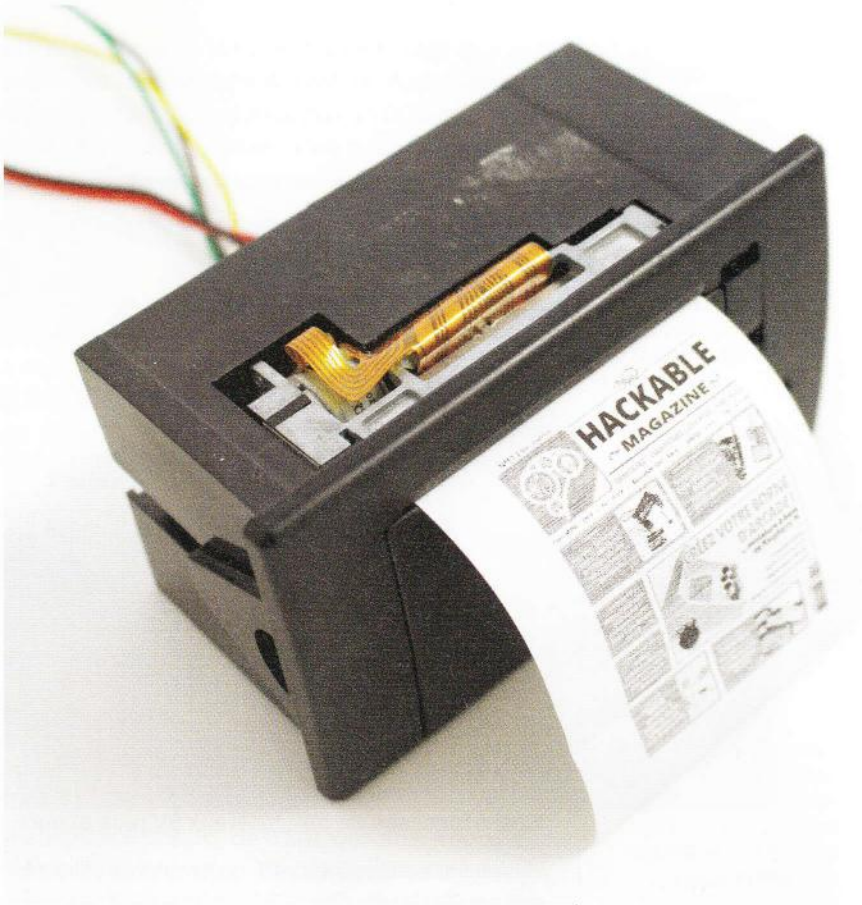
Dans un précédent numéro du magazine, nous avons découvert le monde des imprimantes thermiques sur plateforme Arduino et nous avons pu constater les limitations qui s'imposaient à nous. Mais un tel périphérique pourra également être utilisé avec une Raspberry Pi et ainsi nous permettre de bien plus pleinement et facilement exploiter les possibilités offertes...



Une imprimante thermique, dans les grandes lignes puisque je l'ai précédemment détaillé, est un périphérique d'impression utilisant un papier particulier traité à l'aide de produits chimiques. L'impression elle-même n'utilise pas d'encre ou de toner, mais se fait par application de chaleur sur le support qui passe alors du blanc au noir. L'imprimante qui nous occupe ici est de taille réduite et utilise pour seul consommable des rouleaux de papier thermique tels que ceux utilisés pour les caisses enregistreuses de supermarché ou encore les terminaux de paiement (TPE).

La communication entre un hôte (PC, système embarqué, Raspberry Pi, Arduino, Ti Launchpad, etc.) se fait par l'intermédiaire d'une liaison série TTL (0-5V) et l'imprimante est généralement alimentée par une source distincte, souvent en 5V. L'alimentation demande un courant relativement important en raison de l'utilisation d'éléments chauffants qui constituent la tête d'impression occupant toute la largeur du papier (384 points). La seule partie mécanique de l'ensemble est le système de défilement du papier. Ainsi, l'absence d'encre et cette mécanique réduite au minimum expliquent facilement l'utilisation de ce système d'impression nécessitant une maintenance limitée, pour bon nombre d'usages présents et futurs.

Nous l'avons vu dans un précédent article sur le sujet, la plateforme Arduino montre rapidement ses limites lorsqu'il s'agit de produire des graphismes ou des éléments complexes (autres que du texte) à imprimer. Ce problème ne se pose pas avec une Raspberry Pi grâce à sa mémoire, son processeur puissant et surtout la foison d'espace de stockage et d'outils à notre disposition. L'imprimante thermique pourra ici exprimer tout son potentiel sans aucune limitation.



Ce modèle d'imprimante thermique petit format est relativement courant et facile à trouver en ligne. Pour quelques 40€, vous disposerez ainsi d'une solution d'impression compacte et originale pour votre projet.

1. CONNEXION À LA RASPBERRY PI

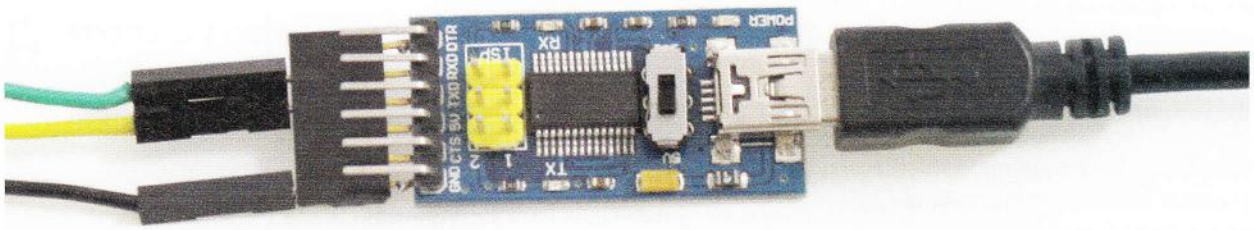
L'imprimante thermique dont il est question dans cet article se pilote via une liaison série avec des niveaux de tension entre 0 et 5 volts. Certains modèles peuvent être contrôlés avec 0-3,3V à condition d'être alimentés non pas en

L'imprimante testée s'interface via une connexion série utilisant des signaux 0-5V. Ceci n'est pas compatible avec les niveaux de tension d'une carte Raspberry Pi. Mieux vaut alors utiliser un convertisseur USB/série non seulement fonctionnel, mais également plus simple à mettre en œuvre.

5V, mais en 3,7V (via un accumulateur LiPo par exemple). Ce n'est pas une caractéristique commune à toutes ces imprimantes et le plus souvent il est nécessaire de s'en tenir aux 5V purs et durs.

Les cartes Raspberry Pi, en revanche, utilisent des entrées/sorties en 0-3,3V qui ne sont pas tolérantes au 5V. Il est alors hors de question de connecter directement l'imprimante sur les broches de la carte. De plus, selon les modèles de RPi, il n'est pas possible d'utiliser correctement le port série sans quelques problèmes (au moment du démarrage en particulier). La solution qui, de plus, offre un certain nombre d'avantages, consiste à tout simplement utiliser un adaptateur USB/série. Exactement le même que celui utilisable pour obtenir une console pour communiquer avec la RPi depuis un PC ou un Mac.

L'adaptateur USB/série est un outil indispensable que ce soit pour votre Pi ou une carte type Arduino. Un choix judicieux se portera sur un ou plusieurs exemplaires d'adaptateurs capables de travailler en 5V ou en 3,3V par simple positionnement d'un interrupteur (ou d'un cavalier).



La plupart de ces adaptateurs permettent de choisir la tension utilisée à l'aide d'un petit interrupteur ou d'un cavalier. Il est généralement recommandé de préférer ces modèles, car ils peuvent ainsi être utilisés aussi bien avec des cartes comme Arduino (UNO, Mega, Leonardo, 101, etc.) typiquement en 5V, qu'avec les Ti Launchpad, l'Arduino Due ou Zero, les cartes STM32 Discovery, etc., en 3,3V.

Notre imprimante série TTL se transformera alors en imprimante thermique USB, bien plus facile à utiliser avec n'importe quel système embarqué ou SBC (*Single-Board Computer*, ordinateur mono-carte), ou même un PC.

La connexion en elle-même est relativement simple puisqu'il suffit d'alimenter l'imprimante avec un bloc fournissant quelques 2,5A en 5 volts, et de relier la ligne TX à RX de l'adaptateur ainsi que RX au TX, puis de brancher le tout à un port USB de la carte Raspberry Pi. Automatiquement, un nouveau port série **/dev/ttyUSBn** ou **/dev/ttyACMn** devrait apparaître (où **n** est un numéro). La commande **dmesg** vous montrera également que la détection de l'adaptateur s'est bien déroulée :

```
usb 1-1.2: new full-speed USB device number 5 using dwc_otg
usb 1-1.2: New USB device found, idVendor=0403, idProduct=6001
usb 1-1.2: New USB device strings: Mfr=1, Product=2, SerialNumber=3
usb 1-1.2: Product: FT232R USB UART
usb 1-1.2: Manufacturer: FTDI
usb 1-1.2: SerialNumber: A9UH977J
ftdi_sio 1-1.2:1.0: FTDI USB Serial Device converter detected
usb 1-1.2: Detected FT232RL
usb 1-1.2: FTDI USB Serial Device converter now attached to ttyUSB0
```

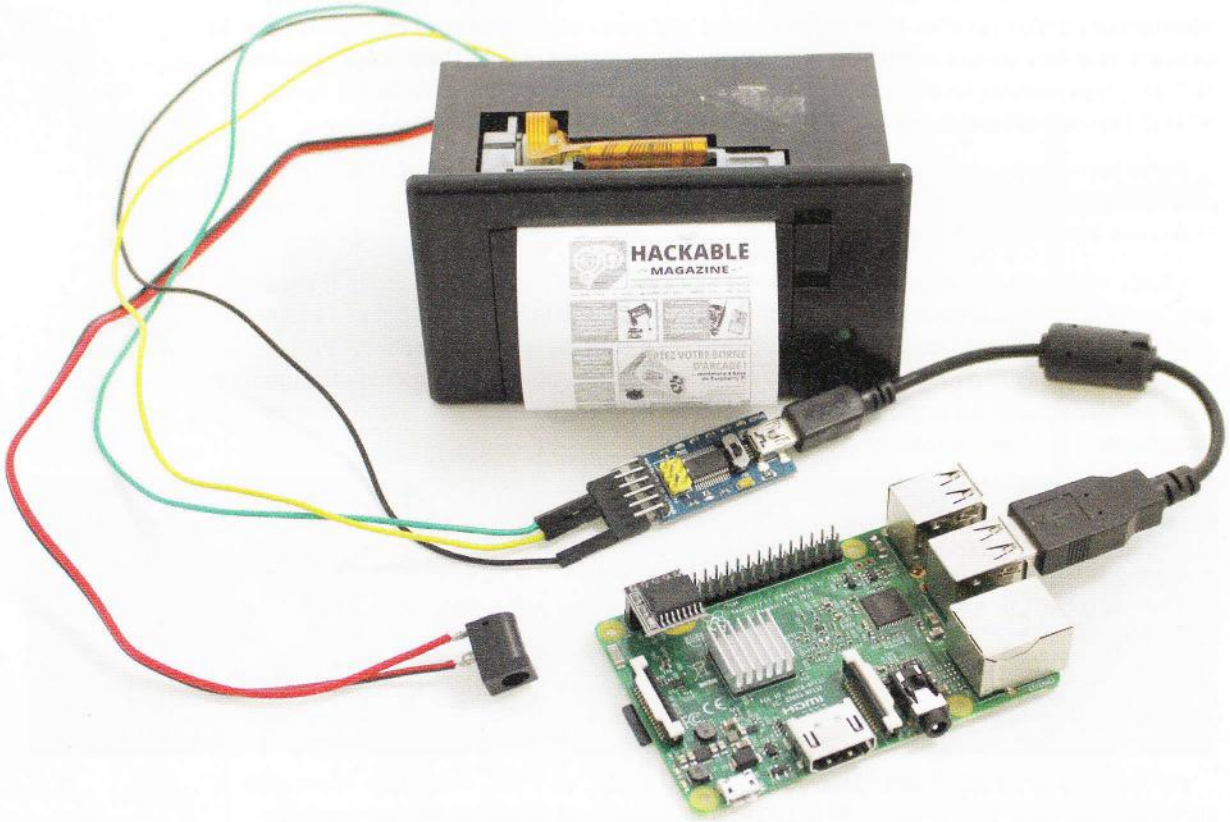
Nous avons ici un adaptateur basé sur la puce FT232RL de FTDI, désormais accessible via le nom **ttyUSB0** (**/dev/ttyUSB0** pour le chemin complet). Cette information sera utile dans les scripts Python que nous allons utiliser. Notez qu'en cas d'utilisation de plusieurs adaptateurs, le nom des pseudo-fichiers dans **/dev** se termine par un numéro correspondant à l'ordre de détection. Pour éviter toutes confusions, vous pouvez également utiliser le contenu de **/dev/serial** pour accéder au périphérique :

```
$ ls /dev/serial/by-id/
usb-FTDI_FT232R_USB_UART_A9UH977J-if00-port0
```

Le nom est ici composé du bus, du constructeur, de la fonction, du numéro de série, de l'interface et du port, rendant alors le nom unique, indépendamment de l'ordre de branchement (remarquez le « A9UH977J » dans le nom du fichier, correspondant au numéro de série apparaissant dans la sortie de **dmesg**). Les fichiers dans **/dev/serial** sont des liens symboliques, sorte de raccourcis pointant sur les **ttyUSBn** ou **ttyACMn** correspondants dans **/dev**.

2. PILOTAGE AVEC PYTHON

Tout comme il existe une bibliothèque Arduino développée par Adafruit pour son imprimante thermique (qu'on peut trouver de bien d'autres manières à un prix plus raisonnable, mais avec moins d'informations techniques vérifiées), il existe un équivalent sous la forme d'un module Python, utilisable sur Raspberry Pi. Également développé par Adafruit, celui-ci est téléchargeable sous la forme d'un fichier Zip, par exemple, depuis **https://github.com/adafruit/Python-Thermal-Printer**.



La connexion par l'intermédiaire de l'adaptateur USB permettra une utilisation sur Raspberry Pi, mais également n'importe quelle autre carte du même type (comme la BeagleBone Black ou le C.H.I.P.), mais aussi sur PC ou Mac avec le même module Python.

Afin de découvrir un autre biais pour récupérer ce code, nous utiliserons la commande `git` nous permettant de directement obtenir tout cela depuis GitHub, directement en ligne de commandes sur la RPi. Pour ce faire, installez tout d'abord l'outil avec un simple `sudo apt-get install git`.

Cet utilitaire est bien plus qu'un outil de téléchargement, c'est un logiciel de gestion de versions décentralisé. Il permet de maintenir des fichiers (le plus souvent du code) à jour sur plusieurs machines en même temps, offrant ainsi la possibilité de travailler à plusieurs sur un seul et même projet, en gérant les différentes versions et modifications tout en limitant le risque de conflits.

En principe et techniquement, il s'agit d'un outil décentralisé et donc sans serveur spécifique à installer, mais dans le cas de GitHub, une plateforme très importante d'hébergement gratuit de projets, un dépôt GitHub peut être vu comme un serveur central.

Pour récupérer le code Python qui nous intéresse, il nous suffit d'invoquer la commande :

```
$ git clone https://github.com/adafruit/Python-Thermal-Printer.git
Clonage dans 'Python-Thermal-Printer'...
remote: Counting objects: 165, done.
remote: Total 165 (delta 0), reused 0 (delta 0), pack-reused 165
Réception d'objets: 100% (165/165), 73.39 KiB | 0 bytes/s, fait.
Résolution des deltas: 100% (74/74), fait.
Vérification de la connectivité... fait.
```

Ceci aura pour effet de créer un répertoire **Python-Thermal-Printer** contenant une copie (un clone) des fichiers se trouvant sur GitHub. Le lien utilisé est affiché directement sur la page web (à gauche du bouton ***Download ZIP***). Ici, c'est un moyen rapide de récupérer les fichiers, mais Git est bien plus puissant que cela en vous permettant de suivre les modifications que vous apporterez aux fichiers et éventuellement de participer au développement du projet cloné.

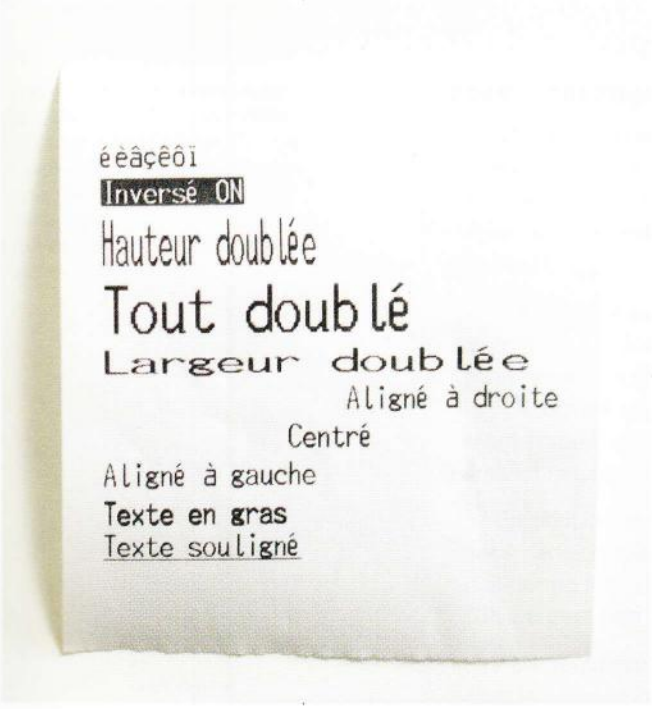
Le contenu du répertoire obtenu est composé du module Python **Adafruit_Thermal.py** et de différents exemples. Un sous-répertoire **gfx/** contient des éléments graphiques dans différents formats pour ces mêmes exemples.

Pour développer vos scripts Python, il vous suffira de placer le fichier **Adafruit_Thermal.py** dans le même répertoire que votre ou vos scripts. Vous pourrez ensuite écrire un premier programme :

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-

# utilisation du module
from Adafruit_Thermal import *

# déclaration objet imprimante
printer = Adafruit_Thermal("/dev/ttyUSB0", 9600)

# configuration
printer.begin(140)
# restauration valeurs pas défaut
printer.setDefault()

# normal
printer.println("Normal")

# Texte inversé
printer.inverseOn()
printer.println("Inversion")
printer.inverseOff()

# tailles doubles
printer.doubleHeightOn()
printer.println("Double Hauteur")
printer.doubleWidthOn()
printer.println("Double Tout")
```

```
printer.doubleHeightOff()
printer.println("Double Largeur")
printer.doubleWidthOff()

# alignement
printer.justify('R')  # Right
printer.println("Droite")
printer.justify('C')  # Center
printer.println("Centre")
printer.justify('L')  # Left
printer.println("Gauche")

# gras
printer.boldOn()
printer.println("En gras")
printer.boldOff()

# souligner
printer.underlineOn()
printer.println("Souligne")
printer.underlineOff()

# défilement du papier
printer.feed(2)
```

La puissance de la Raspberry Pi couplée à l'utilisation de Python nous permet de facilement imprimer des caractères accentués en convertissant les chaînes de caractères UTF-8 en CP437. Chose également possible sur Arduino, mais uniquement au prix d'efforts considérables.



Une fois le script saisi avec votre éditeur de texte préféré (Vim est un excellent choix), il ne vous restera plus qu'à l'enregistrer sous un nom quelconque (comme **premier.py**) et de changer ses permissions pour le rendre exécutable avec **chmod +x premier.py**. Vous pourrez alors le lancer avec **./premier.py** ou encore avec **python premier.py**.

L'élément important du script est principalement l'initialisation (instanciation) de l'objet **printer** qui prend en argument le port série précédemment détecté, ainsi que la vitesse de communication de l'imprimante (qui peut être connue en maintenant appuyé le bouton du périphérique tout en branchant l'alimentation). Notez l'utilisation d'un argument pour **printer.begin()**, celui-ci permet de déterminer le « temps de chauffe » de la tête d'impression afin d'obtenir un résultat optimal en fonction de la qualité du papier thermique. Le script exemple **calibrate.py**, édité pour ajuster les paramètres de l'imprimante, vous permettra de choisir rapidement une valeur adaptée.

3. PROBLÈMES D'ACCENTS ET SOLUTION

L'impression de texte est la fonction de base de ce matériel. Malheureusement, la conception du périphérique ne date pas d'hier et est dépendante d'autres systèmes qui n'utilisent pas une gestion de caractères moderne. Aujourd'hui, le standard d'encodage de caractères est UTF-8, mais l'imprimante utilise par défaut un jeu de caractères CP437, initialement défini par IBM pour son premier PC et son système DOS en 1981. Ceci ne pose pas de problème pour les 128 caractères ASCII standards, ce qui explique que le texte de notre premier exemple s'imprime correctement.

Mais dès lors que des caractères contenus dans le jeu étendu sont utilisés, les problèmes arrivent. Ainsi, les caractères accentués que vous voulez envoyer à l'imprimante ne sont pas ceux qu'elle est en mesure d'imprimer et ils sont remplacés par d'autres caractères. Votre script utilise UTF-8 et l'imprimante veut du CP437.

Pour corriger le problème, il suffit de traduire nos textes d'un standard vers l'autre. Contrairement à la mise en œuvre sur Arduino, ceci est bien plus facile sur Raspberry Pi et avec Python. Les objets de type chaînes de caractères disposent d'une méthode permettant le décodage et l'encodage de caractères. Pour décoder un texte en UTF-8, il suffit d'utiliser `decode('utf-8')`. Exemple :

```
"éèâçêôï".decode('utf-8')
```

Ceci fait, nous avons alors un objet pouvant être réencodé dans un jeu de caractères différent, comme CP437, avec `encode('cp437')` :

```
"éèâçêôï".decode('utf-8').encode('cp437','replace')
```

Et voici notre chaîne de caractères pleine d'accents transcodée en CP437 et intelligible par l'imprimante. Nous pouvons alors utiliser `println()` pour l'imprimer :

```
printer.println("éèâçêôï".decode('utf-8').encode('cp437','replace'))
```

Ça marche parfaitement bien, mais ce n'est vraiment pas pratique. Plusieurs options s'offrent à nous pour simplifier cela : écrire une fonction, développer une nouvelle classe qui hérite de `Adafruit_Thermal`, ou tout simplement modifier le module d'Adafruit en ajoutant deux méthodes pour l'impression de texte UTF-8. C'est cette voie que nous choisirons ici.

Il vous faudra alors éditer le fichier `Adafruit_Thermal.py` puis, en fin de fichier ajouter :

```
def printutf8(self, *args, **kwargs):
    for arg in args:
        self.write(str(arg).decode('utf-8').encode('cp437','replace'))

def printlnutf8(self, *args, **kwargs):
    for arg in args:
        self.write(str(arg).decode('utf-8').encode('cp437','replace'))
        self.write('\n')
```

Il s'agit ni plus ni moins des déclarations des méthodes `print()` et `println()` se trouvant déjà dans le fichier, étoffées de notre petit bout de code permettant la traduction UTF-8 vers CP437. Pour les utiliser, il nous suffit donc d'oublier les deux précédentes méthodes et d'utiliser `printutf8()` et `printlnutf8()` à la place. Exemple :

```
# normal
printer.printlnutf8("éèâçêôï")

# taille
printer.doubleHeightOn()
printer.printlnutf8("Hauteur doublée")
printer.doubleWidthOn()
printer.printlnutf8("Tout doublé")
printer.doubleHeightOff()
printer.printlnutf8("Largeur doublée")
printer.doubleWidthOff()
```

Et voilà ! Nous pouvons maintenant imprimer du texte bien français dans toute sa splendeur typographique.

4. IMPRESSION D'IMAGES

Si vous avez lu l'article sur l'impression avec Arduino dans le précédent numéro, vous savez que c'est là quelque chose de relativement pénible à mettre en œuvre et que cela passe par la conversion des images sous la forme de données brutes (*raw*) intégrées au croquis. Il est possible de faire de même avec le module Python, mais la puissance de la Raspberry Pi ainsi que la diversité des modules Python disponibles nous permet une autre approche, bien plus simple : imprimer directement des images au format PNG ou JPEG par exemple.

Pour cela, il nous faut le module *Python Imaging Library* qui pourra être rapidement installé via `sudo apt-get install python-imaging`.

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-

# module Adafruit
from Adafruit_Thermal import *

# module Python Imaging
import Image

# configuration
printer = Adafruit_Thermal("/dev/ttyUSB0", 9600, timeout=5)
printer.begin(140)

# un texte
```

```
printer.println("Voici une image :")
printer.feed(1)

# une image
printer.printImage(Image.open('image.png'), True)
printer.feed(2)
```

Il n'est pas même nécessaire de transformer l'image en noir et blanc pour que l'impression donne un résultat acceptable, mais elle devra cependant avoir la bonne taille et ne pas dépasser le maximum de 384 pixels de la tête d'impression.

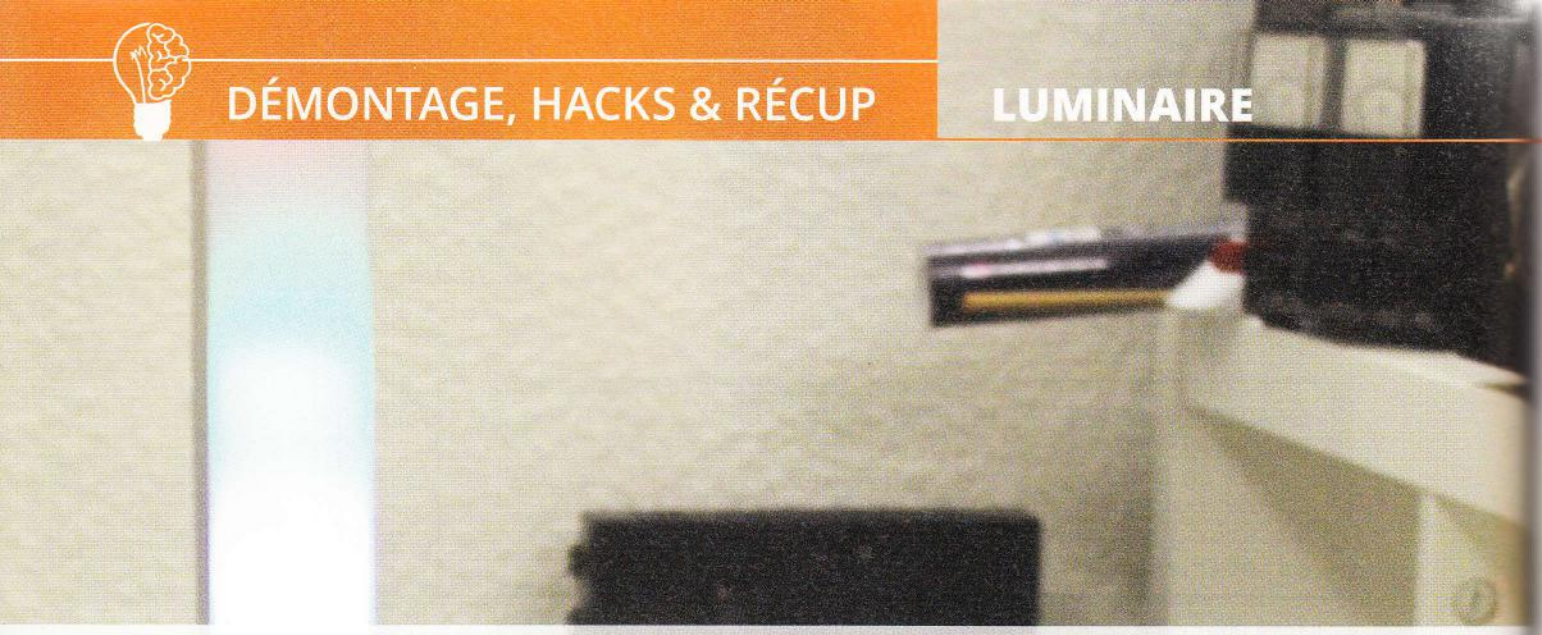
Il vous sera même possible à l'aide de *Python Imaging Library* de composer une image à partir d'éléments graphiques contenus dans des fichiers. C'est ce que fait le script d'exemple **sudoku-gfx.py** à l'aide du fichier **sudoku.png** (une grille vide) et en modifiant l'image pour intégrer du texte avant l'impression. C'est une excellente base pour apprendre à manipuler ainsi des données graphiques (eh oui, c'est bien un générateur de Sudoku).

L'un des exemples livrés avec le module d'Adafruit est un générateur de grilles de Sudoku. Ce script Python est un excellent point de départ pour apprendre à composer des images à imprimer à partir d'un fichier de base et de données dynamiques.

CONCLUSION

L'adjonction d'une imprimante thermique miniature à la Raspberry Pi offre des possibilités importantes. La programmation en Python ainsi que les modules disponibles pour bien des tâches permettent d'envisager une foule de réalisations allant de l'impression de la météo, la production de QRcodes à la demande, la composition d'images, l'impression de tweets, la création de listes et de mémos, etc.

Contrairement à la plateforme Arduino où nous étions bien à l'étroit tant en termes de mémoire et de puissance que de connectivité, nous pouvons avec une Pi laisser libre cours à notre imagination et exploiter au mieux les capacités de l'imprimante. Les fonctionnalités de cette dernière, déjà importantes et surtout notre imagination sont la seule limite ! DB



CONVERTISSEZ UNE RÉGLETTE LUMINEUSE EN AFFICHEUR MULTICOLORE

Denis Bodor

Ce n'est pas un secret, créer un montage fonctionnel, même si cela demande parfois beaucoup de travail n'est pas grand-chose par rapport au fait de transformer ce même montage en quelque chose de propre, présentable et soigné. En effet, ceci demande généralement des aptitudes qui vont au-delà de l'électronique et de la programmation de croquis, il faut des mains expertes d'artisan. Il existe toutefois une solution : réutiliser et détourner de son usage premier un produit existant !

Si, comme moi, vous êtes coutumier des visites dans les magasins de bricolage, entraînez-vous à ne pas voir les produits. En effet, parcourez les allées, regardez les produits et voyez-y plutôt des potentialités. C'est ainsi que lors d'une petite visite au Mr Bricolage non loin de chez moi, j'ai remarqué une réglette lumineuse à leds. Vendue comme une alternative faible consommation aux tubes fluorescents, celle-ci se présente, pour ce modèle, comme une barre lumineuse blanche affichant une consommation de 10 watts, fournissant 850 lumens et donnée pour 25000 heures de fonctionnement.

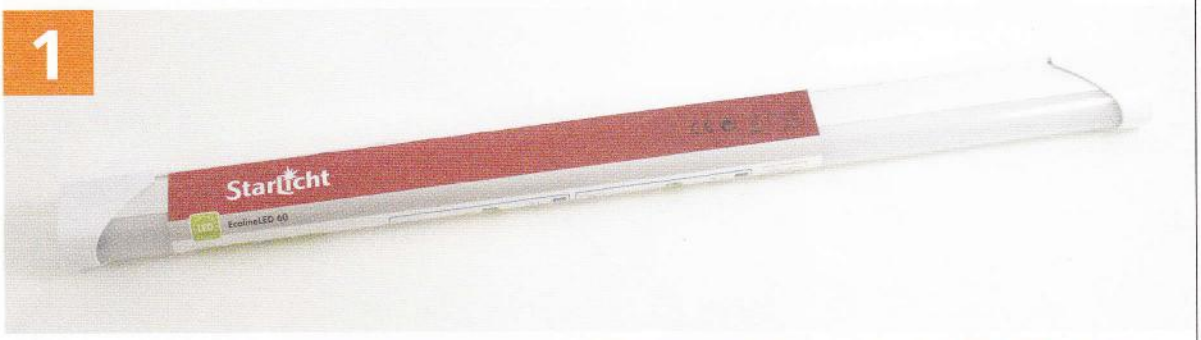
Le produit est déjà intéressant en lui-même par opposition aux classiques supports pour tubes fluorescents (maladroitement souvent appelés « néons »), même ceux équipés de ballasts électroniques. Certes plus cher, quelques 15€ pour ce modèle de 60 cm, ce produit dispose d'une fonction très intéressante : il ressemble à s'y méprendre à un tube fluorescent classique, mais est en réalité juste une « coque » translucide pour une barre de leds.

En voyant le produit, je n'ai pas pensé à ce qu'il était, mais à ce qu'il pouvait devenir en en remplaçant les tripes par un ruban de leds intelligentes WS2812b. Adieu la monotonie de la lumière blanche et de l'uniformité de l'éclairage, et bonjour les effets lumineux à profusion, le tout dans un « emballage » tout beau et tout propre.

Les leds WS2812b sont des composants regroupant un circuit intégré et trois leds (rouge, vert, bleu). Disposant de quatre broches, elles s'alimentent en 5 volts et sont contrôlées par une unique broche destinée à accepter des données sous la forme d'impulsions électriques d'une durée précise. Ainsi, en utilisant une seule broche d'une carte Arduino, il est possible d'ordonner au composant d'allumer chaque led avec une intensité comprise entre 0 (éteint) et 255 (maximum). Mieux encore, la dernière broche relaye les données pour le composant suivant : il est possible d'enchaîner les WS2812b et ainsi contrôler une série de leds avec, toujours, une seule broche de la carte Arduino (ou d'une autre plateforme).

Les WS2812b ainsi que d'autres composants du même type sont vendus sous bien des formes : individuellement, en panneau, en réglette rigide ou en ruban. Les rubans se déclinent en différentes tailles et densités, en nombre de leds par mètre (30, 60 ou 144), et se prêtent tout à fait à la modification dont nous allons parler de suite...

1. DÉMONTAGE, HACK ET TRANSFORMATION

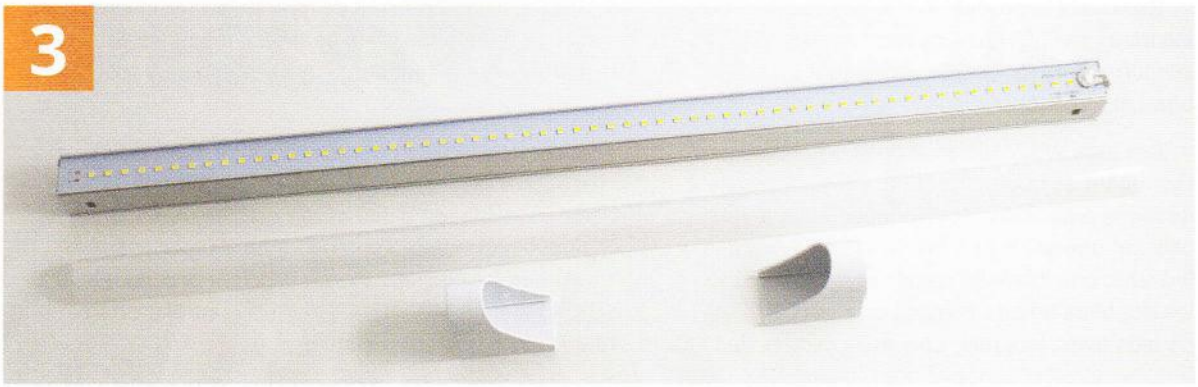


Voici notre victime, l'EcoLineLED 60 de StarLicht, vendue par Mr Bricolage : http://www.mr-bricolage.fr/catalog/product/view/id/81957. Notez au passage que les caractéristiques techniques du produit sur la page web sont inexactes, comme la tension de 12V au lieu de ~230V ou encore la mention « led integres et ballast électronique » (ce n'est pas un ballast, c'est une alimentation à courant continu à base de MT7813). Le même modèle semble également en vente à Castorama.

2

Gros plan sur les caractéristiques précisées sur l'emballage : lumière relativement froide, puissance 10W, 850 lumens de puissance lumineuse, 15000 mises sous tension, non réglable en intensité... Déjà un produit relativement intéressant pour moins de 15€. Mais comme le dirait le très australien Dave Jones d'EEVblog, « Don't Turn it On, Take it Part ! » (« Ne le mettez pas en route, démontez-le ! »).

3



La réglette se démonte assez facilement. Il suffit de retirer les caches à chaque extrémité pour ensuite faire glisser la partie supérieure dans les rails et découvrir les leds montées sur un circuit imprimé.

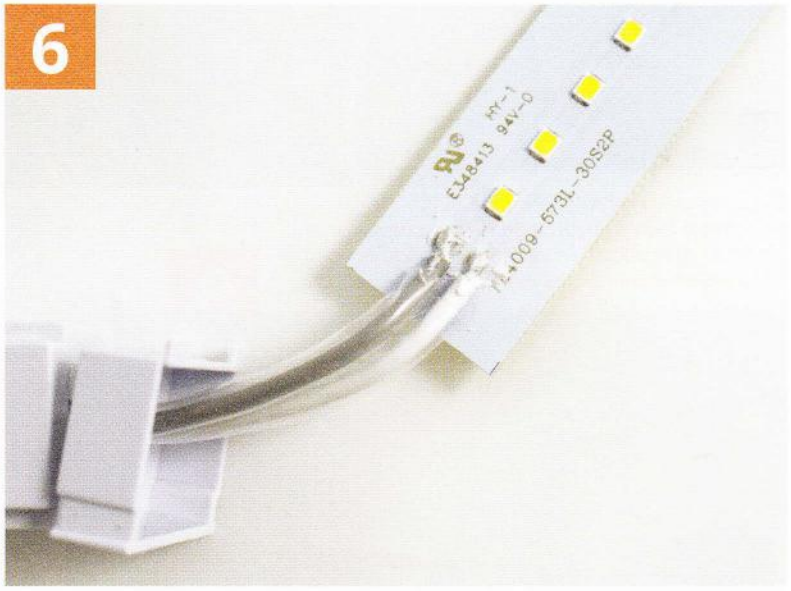
4

Le circuit supportant les leds est lui aussi glissé dans des rails du support en aluminium. Après démontage du bloc d'alimentation et des différents connecteurs, l'ensemble peut être désossé assez aisément. Le circuit de leds n'est pas collé contrairement à l'alimentation qui demande un peu de « persuasion mécanique manuelle ».

5

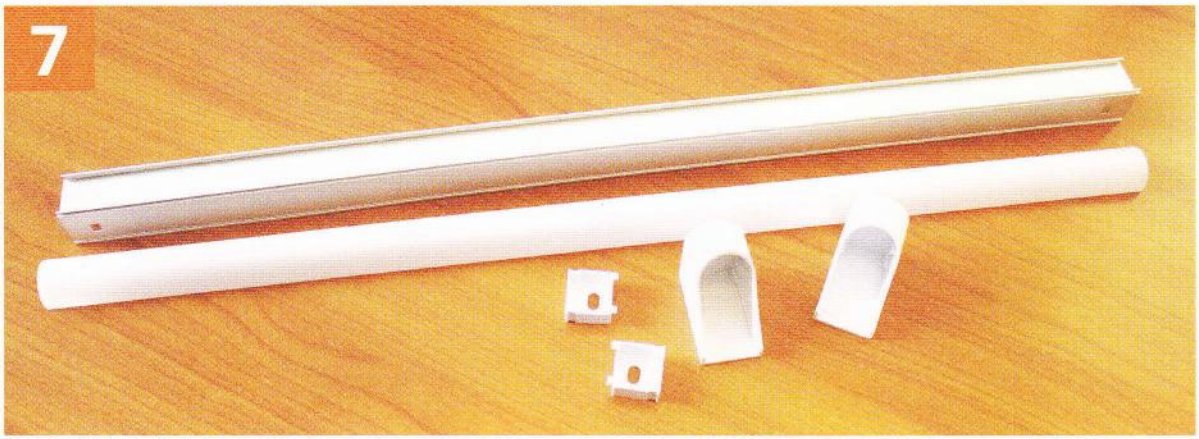
L'alimentation extraite ainsi n'est pas perdue. Celle-ci, de relativement bonne facture, pourra toujours servir pour un projet ultérieur même si 90 volts en courant continu et 100 mA sont des valeurs relativement atypiques.

6



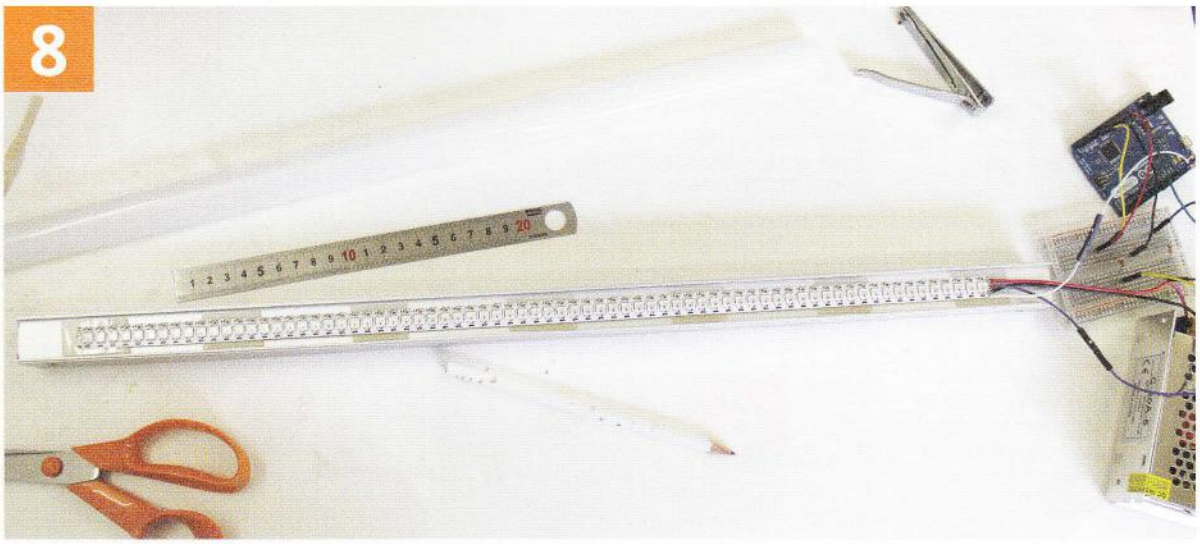
Aux deux extrémités de la réglette en aluminium sont placés des supports en plastique permettant à l'ensemble de garder une certaine rigidité. L'un est relativement facile à extraire, mais le second est traversé par les câbles d'alimentation des leds blanches. Il est donc nécessaire de retirer le silicone protégeant la connexion et de dessouder les câbles. Sauf, bien entendu, si vous n'accordez que peu d'intérêt à ces éléments, auquel cas un coup de pince coupante fera l'affaire.

7



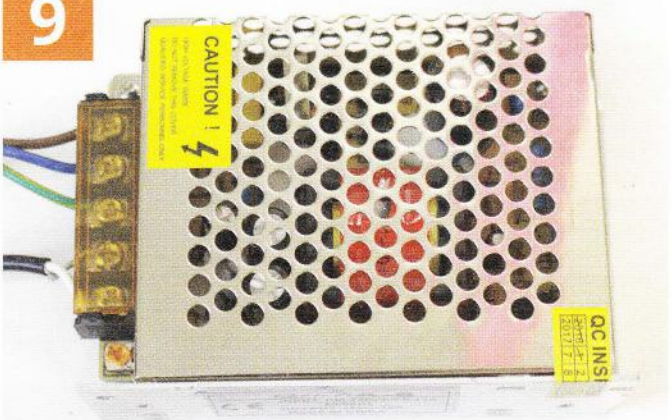
Voici ce que nous récupérons après démontage : la réglette en aluminium, les deux supports structurels internes, le cache translucide et les deux embouts.

8



L'intégration du ruban de leds WS2812b est une simple affaire de mesure et de précision. Le ruban d'un mètre ne peut être coupé qu'à des emplacements précis, ici en son milieu. Ceci nous donne 72 leds sur 50 cm avec une petite marge à chaque extrémité. Cela n'est pas optimal étant donné la longueur du support sans pour autant être vraiment critique. Il y aura simplement une petite zone moins lumineuse de chaque côté du « tube ».

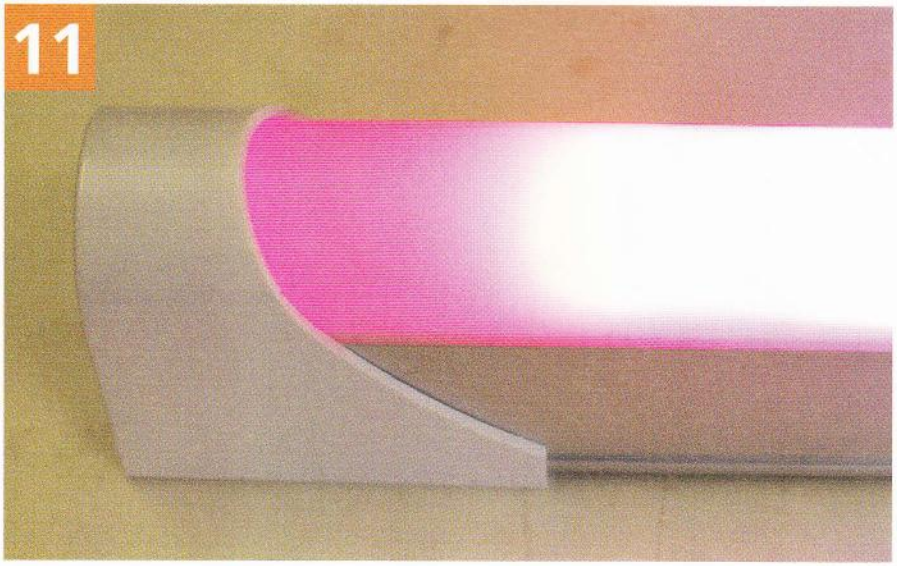
9



L'alimentation du ruban est un problème. Chaque WS2812b comporte une led haute luminosité de chaque couleur (rouge, vert, bleu). En partant du principe qu'avec une intensité maximum chacune d'elle a besoin de 20 mA, nous arrivons à un total maximum de 0,02 x 3 x 72 soit 4,32 A. Ceci nécessite une alimentation conséquente et volumineuse. Celle-ci a été achetée sur eBay pour moins de 12€.

10

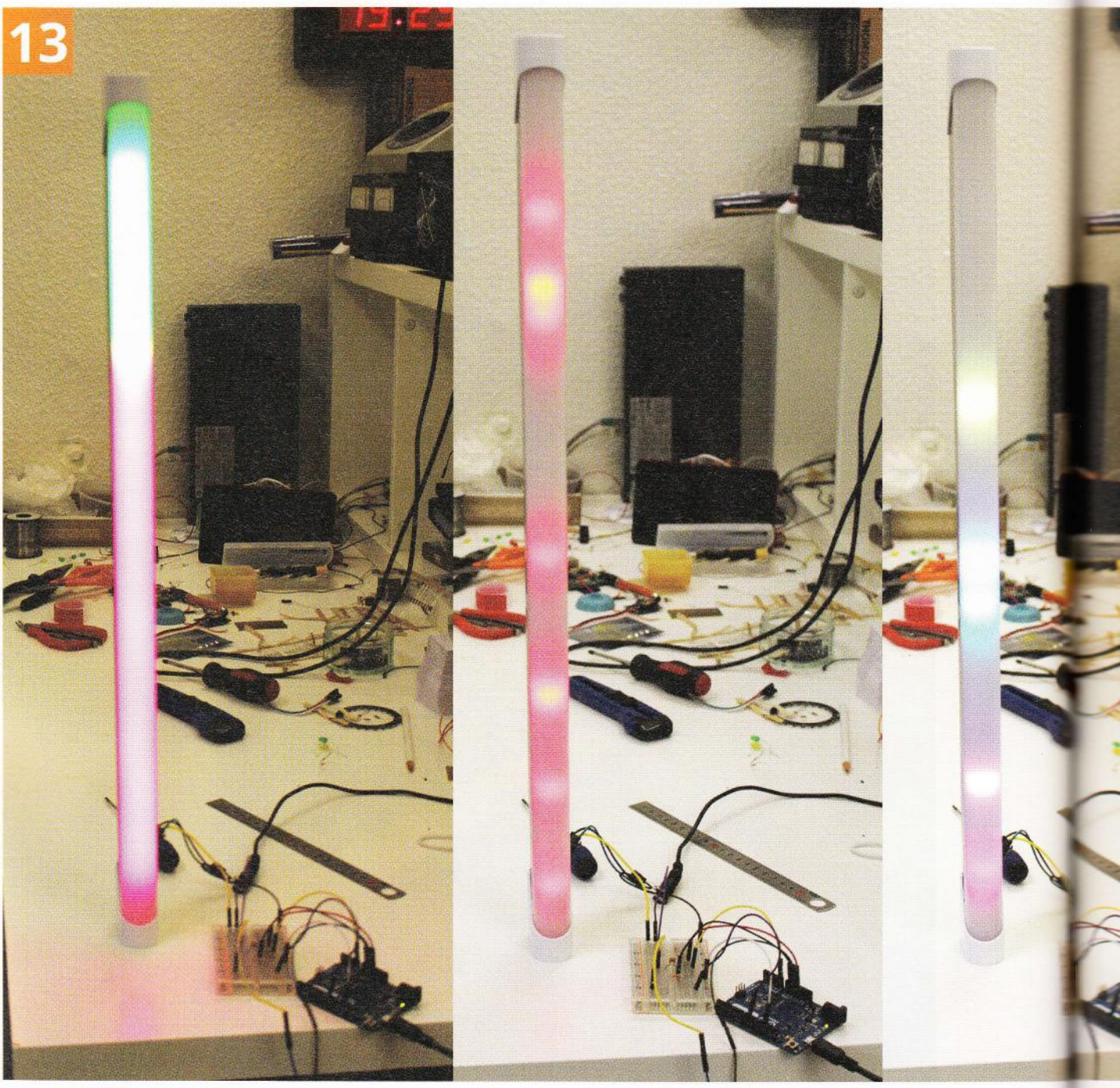
10 ampères en 5 volts, voilà ce qu'est en mesure de fournir l'alimentation utilisée. C'est plus qu'il n'en faut, mais le modèle immédiatement inférieur ne propose qu'un courant de 2A. De plus, une telle alimentation permettra également de fournir le courant nécessaire pour un montage équivalent, mais en version 120 cm (réglette à 19,95€).



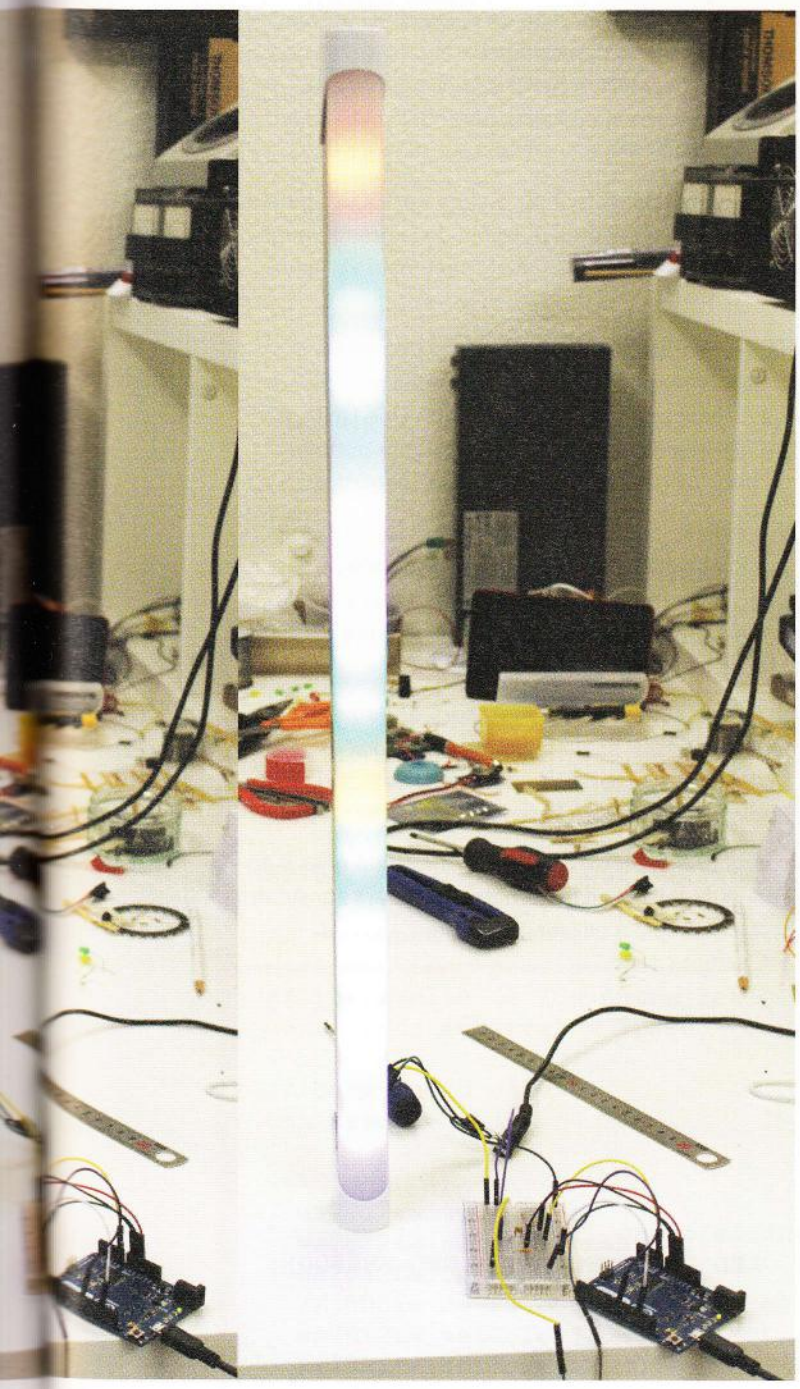
Une fois le ruban collé et la connectique judicieusement passée par l'extrémité du support, on peut remettre en place les éléments en plastique et les caches. La finition est, bien entendu, aussi propre que le modèle original.

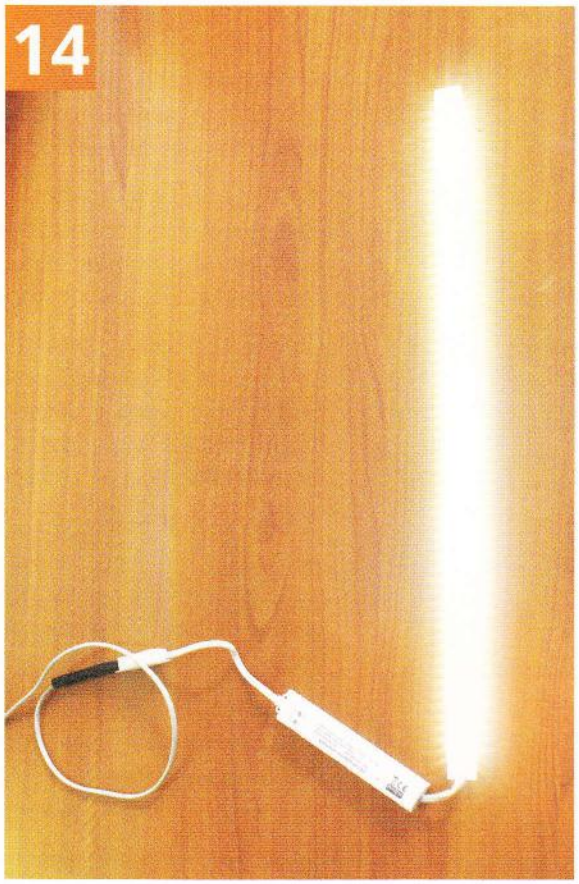
Un premier essai avec un croquis maison affichant un dégradé coloré arc-en-ciel permet de constater le rendu. Ce n'est pas très visible en photo, mais la luminosité globale est aussi importante que les leds blanches originales.

13



En travaillant un peu davantage sur le contrôle des leds par la carte Arduino, on arrive rapidement à obtenir des effets très intéressants : défilement, dégradés, animation, allumage aléatoire. Ici l'ensemble est totalement réassemblé, mais le travail n'est pas fini. Il faudra encore trouver un moyen de dissimuler la carte Arduino et l'alimentation pour que le projet soit totalement abouti.





La barre de leds originale ainsi que son alimentation nous reste sur les bras. Après avoir ressoudé les deux éléments, il ne restera plus qu'à leur trouver un usage. Le principal problème ici est la tension utilisée : 90V. Ceci ne peut rester à l'air libre et nécessite donc un minimum de protection. Il est également très probable que le choix du matériel utilisé pour le support, l'aluminium, excellent dissipateur thermique, n'était pas le fruit du hasard. Ceci sera également un point à garder en tête en réutilisant ce matériel dans le futur.

2. CROQUIS DE DÉMONSTRATION

La connexion d'une ou plusieurs leds WD2812b est relativement simple puisqu'il suffit en principe de connecter l'entrée de donnée de la première led à une broche de l'Arduino. On prendra soin cependant d'ajouter une résistance en série,

entre 200 et 470 ohms, le plus proche possible de la broche de la led. Ceci permet de protéger les leds, car celles-ci sont relativement sensibles aux écarts de tension par rapport à celle d'alimentation. Il est, par exemple, à éviter d'appliquer un état logique haut (+5V) sur l'entrée alors qu'une led n'est pas alimentée. Branchez toujours l'alimentation des leds AVANT l'Arduino.

On prendra soin également d'ajouter un condensateur de lissage de 100µF entre la tension d'alimentation et la masse, là encore le plus près possible de la première led. Ceci permettra également de protéger les composants en absorbant toute variation brutale de tension sur l'alimentation.

Pour les tests, j'ai utilisé la broche 9 en sortie d'une carte Arduino Leonardo. Il n'y a pas de raison particulière au choix de ce modèle, c'est simplement la seule carte qui me restait et n'était pas déjà utilisée pour un projet en cours. Une UNO, une Mega 2560 ou n'importe quel modèle utilisant des tensions de 0/+5V fera parfaitement l'affaire.

Côté logiciel, la bibliothèque la plus utilisée pour prendre en charge les WS2812b est Adafruit NeoPixel, développée par la société éponyme. Vous devrez donc l'installer via le gestionnaire de bibliothèques de votre environnement Arduino. Nous n'allons cependant pas l'utiliser directement, mais reposer sur une autre bibliothèque qui en fait usage : WS2812FX de Harm Aldick (alias kitesurfer1404).

Celle-ci est disponible sur GitHub à l'adresse **https://github.com/kitesurfer1404/WS2812FX**. Il vous faudra récupérer le fichier ZIP sur la page (bouton ***Clone or download***) puis placer le répertoire `WS2812FX-master` qu'il contient dans le sous-répertoire `libraries` de votre répertoire de croquis en le renommant `WS2812FX`.

Cette bibliothèque, relativement jeune (22/05/2016) permet de choisir un effet à appliquer en utilisant vos WS2812b et ce, quel qu'en soit le nombre. Reposant sur Adafruit NeoPixel, cette bibliothèque est tout aussi simple à utiliser. Exemple :

```
#include <WS2812FX.h>

// nombre de WS2812
#define LED_COUNT 72
// broche pour les données
#define LED_PIN 9

// instanciation de l'objet représentant le ruban
WS2812FX ws2812fx = WS2812FX(LED_COUNT, LED_PIN, NEO_GRB + NEO_KHZ800);

// configuration
void setup() {
  // initialisation
  ws2812fx.init();
  // luminosité sur 1/4 du maximum (255)
  ws2812fx.setBrightness(64);
```

```
  // réglage de la vitesse des effets
  ws2812fx.setSpeed(500);
  // un arc-en-ciel glissant
  ws2812fx.setMode(FX_MODE_RAINBOW_CYCLE);
  // démarrage
  ws2812fx.start();
}

void loop() {
  // rafraîchissement de l'animation
  ws2812fx.service();
}
```

La bibliothèque, à cette date (13/06/2016), comprend quelques 44 effets différents désignés par un numéro et par des macros comme `FX_MODE_COLOR_WIPE`, `FX_MODE_TWINKLE`, `FX_MODE_COMET`, `FX_MODE_FLASH_SPARKLE`... (jetez un œil au fichier `WS2812FX.h`). Il est relativement difficile de décrire ces effets et le plus simple est de tout simplement modifier notre croquis de façon à les utiliser tous, les uns à la suite des autres, avec un intervalle de 10 secondes :

```
#include <WS2812FX.h>

#define LED_COUNT 72
#define LED_PIN 9

WS2812FX ws2812fx = WS2812FX(LED_COUNT, LED_PIN, NEO_GRB + NEO_KHZ800);

// décompte du temps
unsigned long previousMillis = 0;
// intervalle de 10s entre les effets
const long interval = 10000;

// compteur d'effet
int effet = -1;

void setup() {
  ws2812fx.init();
  ws2812fx.setBrightness(25);
  ws2812fx.setSpeed(500);
  // choix de la couleur pour les effets monochromes
  ws2812fx.setColor(255, 255, 0);
  ws2812fx.setMode(FX_MODE_STATIC);
  ws2812fx.start();
 // initialisation série à 115200 bps
  Serial.begin(115200);
}
```

```
void loop() {
  ws2812fx.service();

  unsigned long currentMillis = millis();
  // est-il temps de changer d'effet ?
  if (currentMillis - previousMillis >= interval) {
    ws2812fx.stop();
    effet++;
    // si on est au maximum on reprend à 0
    if(effet >= MODE_COUNT) effet = 0;
    // affichage du nom de l'effet
    Serial.println(ws2812fx.getModeName(effet));
    ws2812fx.setMode(effet);
    ws2812fx.start();
    // mise à jour du compteur de temps
    previousMillis = currentMillis;
  }
}
```

Ce croquis va boucler sur les effets disponibles de façon continue tout en affichant leur nom sur le moniteur série. Le développeur a, en effet, prévu une méthode `getModeName()` permettant de récupérer le nom correspondant à l'effet en fonction de son numéro. Quelques autres méthodes sont également intéressantes et étendent les fonctionnalités offertes par Adafruit NeoPixel :

- `increaseSpeed()` et `decreaseSpeed()` permettent d'augmenter et réduire la vitesse des animations avec une valeur d'incrément passée en paramètre ;
- `increaseBrightness()` et `decreaseBrightness()` font de même avec l'intensité lumineuse des leds ;
- `getBrightness()` permet de récupérer la valeur actuelle de luminosité ;
- `getMode()` fait de même pour l'effet.

Notez que cette bibliothèque étend Adafruit NeoPixel et s'utilise donc en lieu et place de cette dernière. Sont donc à votre disposition les mêmes méthodes et fonctions que la bibliothèque originale, mais surchargées des éléments spécifiques à WS2812FX. L'auteur ajoute régulièrement de nouveaux effets et, puisqu'il s'agit de la plateforme GitHub, vous pouvez également contribuer à son développement. Il vous suffit, pour cela, de *forker* son dépôt depuis votre compte GitHub et apporter les modifications avant de les soumettre au projet original (via une *pull request*).

3. BUDGET

Des dispositifs à leds permettant de varier les couleurs existent dans le commerce, mais rien n'existe avec la souplesse que nous avons ici et certainement pas sous la forme d'un faux tube fluorescent. Car c'est bien là tout l'intérêt de cette réalisation. En effet, en choisissant une couleur fixe judicieusement, il ne sera pas vraiment possible

pour un œil non averti de faire la différence entre un luminaire classique et votre version personnalisée. Imaginez la surprise lorsque ce qui paraît être un simple éclairage se mettra à s'agiter !

Au final, le budget total pour une version 60 cm aura été :

- alimentation « *DC5V 2A-60A LED driver Switch Power Supply Transformer For Led Strip module* » (vendeur chinly2012) : 11,12€ ;
- ruban WS2812 « *WS2812B WS2811 5050 SMD RGB Bande Strip 30 60 144 led/m Pixel Ruban Light Décor* » (vendeur bobowaytoway) : 31,99€ à diviser par deux ;
- réglette STARLITE/StarLicht 60cm : 14,90€.

Voilà qui nous donne environ 42€, ce à quoi il faut ajouter une carte Arduino, quelques composants passifs (résistance et condensateur) et câbles. C'est certes trois fois plus cher qu'une réglette classique du commerce, mais avec un plus incontestable en termes de personnalisation.

Les idées de réalisations ne manquent pas, à commencer par la simple lumière d'ambiance ou la décoration interactive (en ajoutant quelques boutons de sélection d'effet). Il sera également possible de faire évoluer le montage en lui ajoutant un mode de communication permettant de configurer les effets : liaison radio, Bluetooth, liaison série, etc. L'option du décompte de temps est également envisageable, tout comme celle de l'horloge binaire, de la barre de progression, de l'indicateur de charge système ou mémoire pour un PC ou une Raspberry Pi et pourquoi pas un afficheur morse...

En ce qui me concerne, je vais maintenant être obligé de me trouver une autre idée car, à présent que vous êtes au courant, je n'aurai plus le seul faux tube fluorescent multicolore animé existant. Zut alors, il faut que je retourne au magasin de bricolage ! DB

ACTUELLEMENT DISPONIBLE

GNU/LINUX MAGAZINE n°195

VISION ASSISTÉE PAR ORDINATEUR

ANALYSEZ VOS IMAGES AVEC OPENCV

NE LE MANQUEZ PAS CHEZ VOTRE MARCHAND DE JOURNAUX ET SUR :

www.ed-diamond.com

APPRENEZ À DÉCHIFFRER UNE DATASHEET

Yann Guidon

Lorsqu'on veut utiliser un composant, surtout inconnu, le premier réflexe est de se procurer sa documentation et la consulter. La datasheet (ou « fiche technique » en bon français) décrit les propriétés distinctives et essentielles du composant, mais d'une manière parfois tellement concentrée que l'avalanche de nombres et de termes peut décourager les débutants. Sans connaître le jargon ou les conventions, il est d'ailleurs facile de passer à côté de détails critiques ! Je vous propose de décortiquer la structure d'une datasheet pour mieux comprendre et trier les informations pertinentes.

La datasheet * est incontournable pour déterminer si un composant convient pour un montage donné. Idéalement, vous avez déjà choisi un circuit et ses caractéristiques, comme les tensions et les courants mis en jeu. Ou au contraire, vous avez besoin de savoir jusqu'où vous pouvez pousser le montage et vous devez estimer les limites de fonctionnement du composant. Que vous disposiez déjà du composant dans vos stocks, ou que vous cherchiez celui qui correspond le mieux, dans tous les cas vous devez savoir ce qu'une référence cache sous le capot.

Cependant, le rôle de la datasheet s'arrête souvent là. Elle vous indique les performances, les caractéristiques et les options, mais elle ne s'attarde généralement pas sur comment le composant fonctionne, ni comment l'utiliser (du moins pour les composants simples). La datasheet n'est pas un cours et vous devrez vous procurer des notes d'application (« *appnotes* » en anglais), des guides d'utilisation (« *user guide* ») ou des manuels pour exploiter au mieux le composant.

1. SÉLECTIONNER UN COMPOSANT

Pour cet exemple, j'ai choisi le **BS170**, car il est très répandu, abordable et il trouve sa place dans d'innombrables applications courantes. Nous allons l'utiliser par la suite, car :

- Il n'est pas dispendieux : on trouve facilement 100 pièces pour 10$ sur des sites de vente en ligne, soit 10 centimes l'unité.
- Le BS170 est l'un des seuls MOSFET canal N disponibles en boitier TO-92, il est facile à planter dans une platine de prototypage sans soudure. La majorité des MOSFET sont disponibles en boitier TO-220 (pour les applications de puissance, ils sont plus lents et prennent de la place) ou en montage en surface (moins évident pour les débutants).
- Les montages envisagés sont en basse tension (autour de 3V) donc il n'est pas nécessaire de chercher un composant supportant les hautes tensions. On trouve quelques références en TO-92 supportant jusque 450V, mais le surcoût n'est pas justifié.
- Je m'attends à commuter un courant faible, mais il est intéressant de savoir jusqu'où on peut aller.
- Il est assez rapide. Mais quelle vitesse de commutation est possible ?
- Le BS170 peut être remplacé par un 2N7000 si cela devient nécessaire, à condition d'adapter le sens des broches. D'ailleurs, la page du BS170 sur Wikipédia est une redirection vers **https://en.wikipedia.org/wiki/2N7000**

2. TROUVER LA (BONNE) DATASHEET

Puisque la référence est choisie, la prochaine étape consiste à se procurer la datasheet. Votre moteur de recherche favori est votre ami et la référence devrait suffire pour trouver les liens attendus. En bonus, des schémas et des articles utiles peuvent se glisser entre les publicités des distributeurs ou les sites de collecte de datasheets, souvent payants ou couverts de publicités (encore).

Il faut admettre que nous autres, les électroniciens, sommes une cible appréciée des annonceurs donc une simple recherche sur Google revient à annoncer être un client demandeur. C'est aussi alléchant pour

* Oui je mets ce mot au féminin, car le mot anglais « sheet » signifie « une » feuille ou « une » fiche.

eux que si vous tapiez la recherche « *hôtel (nom-deville)* », laissant supposer que vous préparez un voyage, donc comptez dépenser de l'argent... Il vaut mieux éviter de laisser des traces sur le web, et obtenir un lien direct vers le site du fabricant (par copier-coller tant qu'à faire). *

Rapidement, une autre évidence apparaît : « BS170 » est une référence générique. Ce n'est pas un produit d'un seul constructeur, mais plutôt un terme, une étiquette sous laquelle les fabricants vendent « *un MOSFET canal N en boitier TO-92 qui supporte 60V ou quelques centaines de milliampères* ». Une rapide recherche sur des sites de distribution de composants montre qu'il existe plusieurs versions du BS170 supportant 150, 270 et 500mA, ce qui recouvre aussi le domaine du 2N7000.

Le problème se pose maintenant : est-ce que le fabricant du composant que vous avez en stock est aussi l'auteur de la datasheet ? Le fabricant n'est pas toujours indiqué sur le composant ou le sachet, si vous en avez acheté un lot. « *Ce n'est pas grave* », pensez-vous, « *c'est la même référence donc les caractéristiques sont identiques, alors allons-y gaiement* ».

Ce serait ignorer des faits pratiques, tels que les différences entre les usines des fabricants, les différentes procédures de contrôle de qualité ou les recettes de chacun pour réduire les coûts, qui sont déjà incroyablement bas et il faut encore dégager quelques bénéfices...

La datasheet est donc *indicative* et ne doit pas être prise comme une référence absolue (du moins dans ce cas). Des composants spécialisés et propriétaires ont une documentation très précise sur la variance, car elle peut être contrôlée précisément en usine par des procédés éprouvés. Ceci n'étant pas possible pour un composant générique, les informations doivent être prises avec des pincettes.

Cela conduit à une situation endémique dans le monde de l'électronique : certains constructeurs ajoutent des références génériques à leurs catalogues et « empruntent » parfois les informations des autres constructeurs. Au final, on ne sait plus d'où vient l'information originale, qui a copié qui, et il n'est pas rare de retrouver des courbes de transfert identiques d'une datasheet à l'autre. C'est encore plus frappant avec les composants fabriqués sous un nom chinois et je me demande d'ailleurs souvent s'ils comprennent ou vérifient ce qu'ils écrivent.

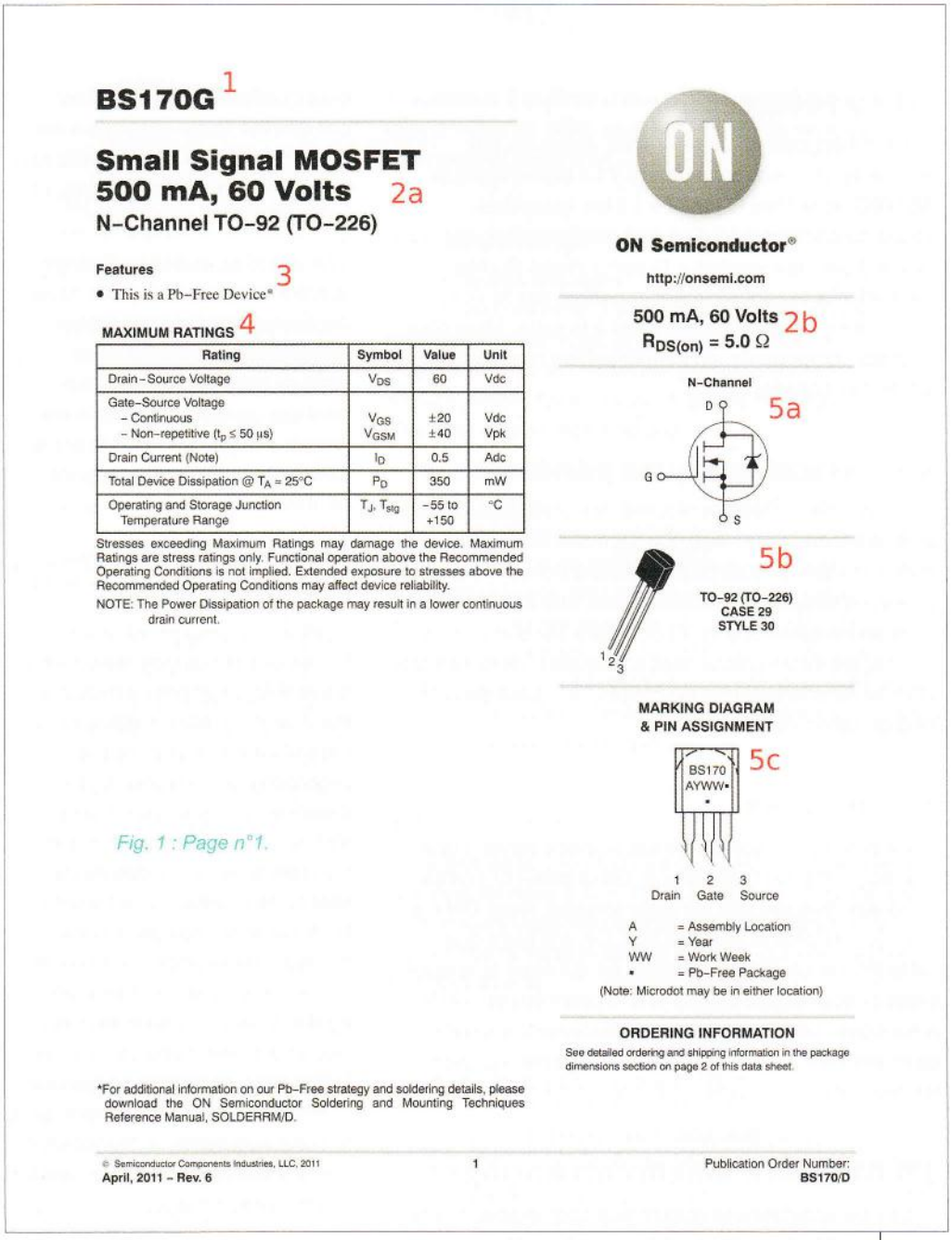
Si une caractéristique est très importante, n'hésitez pas à comparer les datasheets de différents constructeurs et gardez la moyenne ou le minimum de la caractéristique en question.

Pour notre utilisation du BS170, il y a une bonne marge sur tous les paramètres. J'ai donc choisi d'illustrer cet article avec la datasheet de ON Semiconductors **[1]**, car elle est l'une des plus simples et courtes, ce qui devrait être digeste pour les débutants. Nous constaterons progressivement que cet avantage a aussi des inconvénients.

3. LA PREMIÈRE PAGE

Vous avez téléchargé le fichier PDF et l'avalanche commence. Vous voici submergé par des informations et vous n'avez pas encore l'habitude de les trier ou les interpréter. On dirait presque un débat politique télévisé en période électorale, qu'est-ce que tout cela veut dire et comment savoir à quelle information se fier ?

* DB me suggère qu'on peut basculer en navigation anonyme avec les navigateurs comme Firefox ou Chrome. Cela réduit en effet les cookies, mais je ne fais pas confiance à cette méthode, pour la simple et bonne raison que l'adresse IP ne change pas.



BS170G 1

Small Signal MOSFET 500 mA, 60 Volts 2a

N-Channel TO-92 (TO-226)

Features 3

- This is a Pb-Free Device*

MAXIMUM RATINGS 4

Rating	Symbol	Value	Unit
Drain-Source Voltage	V_{DS}	60	Vdc
Gate-Source Voltage – Continuous – Non-repetitive ($t_p \leq 50$ µs)	 V_{GS} V_{GSM}	 ±20 ±40	 Vdc Vpk
Drain Current (Note)	I_D	0.5	Adc
Total Device Dissipation @ T_A = 25°C	P_D	350	mW
Operating and Storage Junction Temperature Range	T_J, T_{stg}	−55 to +150	°C

Stresses exceeding Maximum Ratings may damage the device. Maximum Ratings are stress ratings only. Functional operation above the Recommended Operating Conditions is not implied. Extended exposure to stresses above the Recommended Operating Conditions may affect device reliability.

NOTE: The Power Dissipation of the package may result in a lower continuous drain current.

*For additional information on our Pb-Free strategy and soldering details, please download the ON Semiconductor Soldering and Mounting Techniques Reference Manual, SOLDERRM/D.

ON Semiconductor®

http://onsemi.com

500 mA, 60 Volts 2b
$R_{DS(on)}$ = 5.0 Ω

N-Channel

D 5a
G
S

5b

TO-92 (TO-226)
CASE 29
STYLE 30

1 2 3

MARKING DIAGRAM & PIN ASSIGNMENT

5c

BS170
AYWW▪
▪

1 Drain 2 Gate 3 Source

A = Assembly Location
Y = Year
WW = Work Week
▪ = Pb-Free Package

(Note: Microdot may be in either location)

ORDERING INFORMATION

See detailed ordering and shipping information in the package dimensions section on page 2 of this data sheet.

© Semiconductor Components Industries, LLC, 2011

Fig. 1 : Page n°1.

Pourtant les informations ont une certaine logique et même une organisation typique. En particulier, la première page regroupe les informations les plus importantes afin de les trouver du premier coup d'œil.

Un regard exercé devrait pouvoir déterminer si le composant ne convient pas, à condition évidemment de savoir ce qu'on cherche et comment le trouver. La première page est donc une synthèse qui permet de passer au plus vite à une autre référence. Le premier réflexe est de trouver le détail qui cloche.

3.1 La référence

Commençons par la référence, marquée par l'étiquette (1) : est-ce la bonne ? Le fichier indique **BS170G**, le suffixe G modifie-t-il les caractéristiques du composant ? Ce n'est pas le cas ici, car cela indique une version « Green » (sans plomb). Cela affecte essentiellement le boitier, pas le semiconducteur, donc on peut passer à la suite. Mais pour d'autres composants, un simple suffixe peut avoir de profondes conséquences.

3.2 Caractéristiques générales

L'étiquette (2) est un résumé des principales caractéristiques du composant. Ce sont des valeurs maximales et elles ne sont pas forcément possibles toutes en même temps, elles doivent donc être interprétées dans un contexte précis. Et pour être sûr d'attirer l'attention, les informations sont dupliquées, dans l'en-tête (2a) et dans la colonne de droite (2b), juste sous le logo du fabricant.

3.3 Features

La description sommaire est souvent suivie d'une liste de points forts du produit. L'étiquette (3) correspond aux *features* (les caractéristiques), mais ON n'a pas été très prolixe dans ce cas. On voit juste que cette référence est sans plomb, ce qui étaie la supposition concernant le suffixe G à la première étiquette. À l'opposé, certains produits extrêmement sophistiqués peuvent remplir la première page rien qu'avec les *features*.

3.4 Absolute Maximum Ratings

Les caractéristiques maximales sont indiquées près de l'étiquette (4). Cette partie s'appelle « *Maximum ratings* » ou « *Absolute maximum ratings* » selon le constructeur. Beaucoup de débutants pensent qu'il s'agit des valeurs pour une utilisation normale ou nominale alors qu'en fait, ce sont les conditions à partir desquelles le composant est endommagé ! Il ne faut donc pas s'en approcher, mais prévoir une marge raisonnable. Le bloc de texte sous la table est une mise en garde (à la fois vague, mais explicite) bien qu'elle soir rarement lue, car elle est écrite en anglais en petits caractères. En résumé, ne dépassez jamais ces valeurs, et comme le dit Toshiba :

The absolute maximum ratings are the values which must never be exceeded during operation even for a moment ***[2]****. (Les valeurs maximales absolues sont des valeurs qui ne doivent jamais être dépassées durant le fonctionnement, même durant un moment.)*

3.4.1 Les valeurs maximales s'influent entre elles

Pour compliquer les choses, toutes ces conditions maximales doivent être remplies simultanément alors qu'elles s'influencent mutuellement. Par exemple, regardons la puissance totale dissipée («*Total Device Dissipation* $@T_A=25°C$ »). Étant donné que la puissance est le produit du courant et de la tension, ce paramètre P_D limite aussi ces deux paramètres. Par exemple, le courant continu maximal I_D est indiqué égal à 0,5A, mais si on veut respecter la limite de puissance, la tension V_{DS} ne doit pas dépasser 0,35W/0,5A=0,7V seulement. Si la tension augmente, le composant devra dissiper plus de puissance et sera endommagé.

Corollairement, connaissant la puissance et la tension V_{DS} maximale, on trouve que ce MOSFET ne peut laisser passer que 0,35W/60V=5,8mA si la tension de drain reste à 60V lorsque le courant passe. Rien ne vous empêche d'utiliser le MOSFET comme une résistance de 10KΩ,

mais on voit que ce n'est pas forcément une bonne idée à haute tension.

Notez aussi que la puissance est donnée à une température ambiante T_A=25°C. Il n'est pas indiqué combien de temps ni avec quel type de dissipation. La résistance thermique et la température maximale de jonction ne sont pas non plus indiquées, ce qui empêche de calculer un ajustement en fonction de la température ambiante.

Une puissance d'un tiers de Watt peut faire augmenter la température considérablement si le boitier n'est pas adapté. Le boîtier TO-92 n'est pas prévu pour dissiper de fortes puissances, il faudrait alors recourir à un modèle de transistor plus adapté, en boîtier TO-220 par exemple, avec éventuellement un radiateur.

D'autre part, si le composant est utilisé dans d'autres conditions (comme à l'intérieur d'une boîte métallique à l'extérieur qui peut dépasser 60°C lorsqu'elle est exposée au soleil), la puissance doit être diminuée. Habituellement, des courbes d'ajustement (« *derating* ») sont fournies dans les datasheets, mais il n'y en a pas ici.

Ainsi, cela change la lecture de la dernière ligne de la table : T_j et T_{stg} sont dans une fourchette de -55°C à +150°C, mais la puissance maximale P_D ne s'applique plus et doit être réduite considérablement. D'ailleurs, à une température de fonctionnement de 150°C, si le composant est toujours fonctionnel, il suffirait d'une petite augmentation de la température de 30°C pour faire fondre la soudure (si elle contient du plomb)...

3.5 Brochage

Les figures étiquetées **(5a)**, **(5b)** et **(5c)** sont indispensables et on les cherche souvent, en cours de montage par exemple, pour vérifier où brancher quoi. Par chance, ici, le dessin est presque limpide, même s'il faut parfois faire des vérifications croisées pour s'assurer d'avoir bien compris.

D'autres datasheets sont moins claires et il faut parfois mener une enquête pour retrouver la correspondance entre les broches et les fonctions. Il n'est pas rare de se tromper à cause d'un sens ambigu, si la datasheet ne précise pas si le boitier est vu de haut ou du bas par exemple. Pour notre MOSFET, le doute est facilement dissipé grâce à la technique du multimètre présentée dans l'article précédent **[3]**, mais pour d'autres composants, la chance entre en jeu (avec son inséparable ami Murphy).

3.6 Révision

Le bas de la page indique un numéro de version et une date de mise à jour. C'est pratique si vous disposez de plusieurs versions du fichier, il est préférable de ne garder que la version la plus récente, car elle devrait être plus exacte.

4. LA DEUXIÈME PAGE

La datasheet se poursuit habituellement par une liste détaillée de paramètres physiques, qui mettent en perspective les données de la première page. Il ne faut pas perdre de vue que ces valeurs font partie d'une de ces trois classes :

- soit la valeur est testée en usine, à la fabrication, et la valeur peut être prise au pied de la lettre ;
- soit la valeur est testée sur un petit échantillon de chaque lot produit, donc on suppose que le reste des composants du lot est conforme, mais les statistiques ajoutent « une certaine incertitude » ;

BS170G

ELECTRICAL CHARACTERISTICS (T_A = 25°C unless otherwise noted)

	Characteristic	Symbol	Min	Typ	Max	Unit
	OFF CHARACTERISTICS					
6	Gate Reverse Current (V_{GS} = 15 Vdc, V_{DS} = 0)	I_{GSS}	–	0.01	10	nAdc
	Drain–Source Breakdown Voltage (V_{GS} = 0, I_D = 100 μAdc)	$V_{(BR)DSS}$	60	90	–	Vdc
	ON CHARACTERISTICS (Note 1)					
7	Gate Threshold Voltage (V_{DS} = V_{GS}, I_D = 1.0 mAdc)	$V_{GS(Th)}$	0.8	2.0	3.0	Vdc
	Static Drain–Source On Resistance (V_{GS} = 10 Vdc, I_D = 200 mAdc)	$r_{DS(on)}$	–	1.8	5.0	Ω
	Drain Cutoff Current (V_{DS} = 25 Vdc, V_{GS} = 0 Vdc)	$I_{D(off)}$	–	–	0.5	μA
	Forward Transconductance (V_{DS} = 10 Vdc, I_D = 250 mAdc)	g_{fs}	–	200	–	mmhos
	SMALL–SIGNAL CHARACTERISTICS					
8	Input Capacitance (V_{DS} = 10 Vdc, V_{GS} = 0, f = 1.0 MHz)	C_{iss}	–	–	60	pF
	SWITCHING CHARACTERISTICS					
9	Turn–On Time (I_D = 0.2 Adc) See Figure 1	t_{on}	–	4.0	10	ns
	Turn–Off Time (I_D = 0.2 Adc) See Figure 1	t_{off}	–	4.0	10	ns

1. Pulse Test: Pulse Width ≤ 300 μs, Duty Cycle ≤ 2.0%.

ORDERING INFORMATION

Device	Package	Shipping†
BS170G	TO–92 (TO–226) (Pb–Free)	1000 Unit/Tube
BS170RLRAG	TO–92 (TO–226) (Pb–Free)	2000 Tape & Reel

†For information on tape and reel specifications, including part orientation and tape sizes, please refer to our Tape and Reel Packaging Specifications Brochure, BRD8011/D.

Fig. 2 : Page n°2.

• soit la valeur n'est pas testée, mais est dite « garantie par conception ».

Chaque test prend du temps, occupe des outils très spécialisés et consomme de l'énergie donc ajoute un coût à chaque composant. Les tests sont réalisés à température ambiante et hygrométrie courante, car les chambres climatiques sont chères, lentes, et ne facilitent pas les manipulations, alors qu'une usine doit fabriquer des centaines de milliers de composants chaque jour tout en étant plus économique que la voisine.

Malheureusement, la validité des mesures n'est pas souvent indiquée (il n'en est pas fait mention dans cette datasheet) et on se demande parfois si les composants sont vraiment testés. Je ne parle même pas des composants trop bon marché pour être vrais ou conformes (les clones, les rebuts reconditionnés et les tromperies font beaucoup de tort à l'industrie).

Afin de mettre un peu d'ordre dans tous ces nombres, ils sont regroupés par régimes de fonctionnement (voir l'article de présentation des MOSFET).

4.1 Régime bloqué

L'étiquette **(6)** désigne la partie « *Off Characteristics* » qui contient des mesures des imperfections du MOSFET lorsqu'il ne conduit pas. Par exemple, même si la grille est isolée, sa finesse et la mécanique quantique laissent quand même passer quelques électrons par *effet tunnel.*

4.1.1 Courant de fuite

La mesure I_{GSS} (le terme « *Gate Reverse Current* » peut être traduit par « courant inverse dans la grille ») donne un ordre de grandeur de ce courant de fuite (entre 10 picoampères et 10 nanoampères). C'est trop faible pour être mesuré précisément avec un multimètre de base, mais il faut en tenir compte pour les circuits extrêmement sensibles. Cela inclut par exemple les circuits d'alimentation à base de très petites piles ou batteries, ainsi que les circuits avec une très longue constante de temps.

Le courant de fuite dépend des différentes tensions. Ici le courant vers la source est mesuré avec une tension de grille (V_{GS}) de 15V, ce qui est très élevé : c'est les trois quarts de la tension maximale de 20V. Avec une tension de grille de 3V et en supposant que l'effet tunnel est exponentiel, le courant de fuite devrait être tout à fait négligeable.

4.1.2 Tension maximale de blocage

Ensuite, la mesure $V_{(BR)DSS}$ (« *Drain-Source Breakdown Voltage* ») est la tension entre le drain et la source à partir de laquelle un courant non négligeable commence à circuler, alors que la grille est court-circuitée à la source. L'effet tunnel laisse aussi passer des électrons alors que le transistor est coupé ! Cet autre courant de fuite dépend évidemment de la température et la mesure a été effectuée à 25°C, donc il faudra réduire la tension $V_{(BR)}$ si vous vous attendez à travailler à plus haute température.

4.2 Régime passant

L'étiquette **(7)** désigne la partie « *On Characteristics* ». Elle regroupe les caractéristiques lorsque la grille est à une tension positive, ce qui nous intéresse habituellement le plus. Là encore, il est important de lire les petits caractères, car la **Note 1** de la datasheet indique que les mesures sont effectuées durant de très courtes impulsions. Une période de répétition de 300µs correspond à 3KHz environ, et la période active de 2% implique que l'impulsion ne dure que 6µs. Le composant testé ne reçoit plus qu'un cinquantième de l'énergie maximale et chauffe beaucoup moins qu'en régime continu. Donc les valeurs de cette partie ne doivent pas être prises au pied de la lettre !

4.2.1 Tension de déclenchement

La mesure $V_{GS(Th)}$ (« *Gate Threshold Voltage* ») est une tension de grille à partir de laquelle le courant commence à circuler (significativement) entre le drain et la source. Ce courant significatif I_D est fixé arbitrairement à 1mA. $V_{GS(Th)}$ dépend non seulement de la température, mais aussi des variations inévitables à la fabrication.

Curieusement, pour cette mesure, la grille est connectée au drain alors que cette tension est habituellement fixe. Cela peut s'expliquer par l'effet Miller et les capacités parasites, qui entrent en jeu en régime impulsionnel (lorsqu'il y a des variations très rapides). La remarque sur les impulsions (au paragraphe précédent) commence à prendre du

sens et le court-circuit grille-drain réduit facilement l'influence des parasites. D'un autre côté, cela rend la mesure plus difficile à interpréter et à appliquer dans un montage pratique, puisqu'il est rare que la grille et le drain soient connectés...

4.2.2 Résistance minimale

La résistance drain-source $r_{DS(on)}$ est une des caractéristiques les plus emblématiques des MOSFET. Le document indique qu'on doit s'attendre à 2Ω en moyenne, mais n'exclut pas que cela atteigne 5Ω (comme indiqué en gros à l'étiquette (2b)). Ce composant n'est donc pas adapté à commuter des alimentations, mais plutôt des signaux ou informations.

Encore une fois, il ne faut pas oublier de prendre en compte d'autres paramètres. La mesure est effectuée avec la tension de grille V_{GS}=10V, ce qui n'est pas la tension de fonctionnement escomptée, il faudra donc estimer à la hausse (V_{GS} plus basse, donc résistance plus forte) et mesurer en vrai. Le régime impulsionnel de la mesure évite de faire chauffer le composant, mais dans une application réelle (comme par exemple commuter la cathode de LEDs), l'échauffement va augmenter la résistance.

Enfin, la résistance $r_{DS(on)}$ n'est pas mesurée avec un multimètre en mode ohmmètre, comme vous le feriez à la maison. La tension drain-source (V_{DS}) est mesurée (probablement à l'oscilloscope) lorsqu'une impulsion de 200mA passe. Mais dans un vrai circuit, feriez-vous passer 200mA dans un BS170 de cette manière ? Avec une tension de grille plus faible, on peut s'attendre à 5 ou 10Ω de résistance continue, ce qui se traduit (en appliquant la loi d'Ohm) par une chute de tension de 1 à 2V (est-ce tolérable ?) et un échauffement non négligeable avec un courant continu.

4.2.3 Courant de fuite à l'état bloqué

La valeur suivante $I_{D(off)}$ devrait logiquement se trouver listée dans la partie précédente, car c'est le courant de fuite (maximal) entre le drain et la source, alors que la grille est à 0V (donc le transistor est en régime bloqué). Ce courant est 50 fois plus important que le courant de grille, donc on en tiendra compte en priorité. Vous connaissez la chanson maintenant : la température et les autres tensions influent sur cette valeur.

4.2.4 Conductance

Enfin, la conductance g_{fs} est une autre étrangeté de ce datasheet. Bien qu'elle soit rarement utile, elle est pourtant indiquée par la plupart des autres datasheets. Pourtant la conductance est l'inverse de la résistance ($g_{fs} = 1/r_{DS(on)}$) et on peut la déduire facilement (même si le courant de mesure n'est pas tout à fait le même).

En plus de cette redondance, l'auteur utilise une unité de mesure qui n'est pas standard, car la conductance est exprimée en Siemens (notée S) dans le Système International, alors que l'unité indiquée en **mmhos** (Ohm écrit à l'envers, avec un préfixe **m** pour diviser par 1000 et un suffixe **s** qui n'a pas lieu d'être). Cette ligne de la table n'est donc pas vraiment pertinente.

4.3 Régime ohmique

L'étude du composant dans le domaine « petits signaux » (*Small Signal Characteristics*, désigné par l'étiquette **(8)**) suppose que le transistor est dans une zone de fonctionnement linéaire, en plein milieu de son régime ohmique. Un signal sinusoïdal de faible amplitude peut donc être amplifié avec une distorsion minimale.

Ici, seule la capacité d'entrée C_{iss} est mesurée, dans des conditions qui ne sont pas vraiment représentatives d'une utilisation pratique. La valeur maximale donnée (60pF) dépend de la tension de fonctionnement, comme on le verra dans la partie 5.3.

4.4 Performance en commutation

L'étiquette (9) désigne les valeurs t_{on} et t_{off}, qui sont les temps de montée et de descente. La figure 2 de la datasheet (sur la page 3) montre que c'est aussi le temps de propagation entre la grille et le drain, car la tension de la grille est aussi prise en compte.

Les valeurs moyennes de 4ns sont identiques pour la montée et la descente donc on peut supposer que le composant a un comportement symétrique, contrairement à d'autres composants. Cependant la datasheet ne spécifie pas la symétrie et il y a deux lignes séparées, donc même si les temps sont identiques, il ne faut pas se reposer sur cette supposition.

Les conditions de test sont précisées à la page suivante, car chaque élément du circuit influe potentiellement sur les mesures. La qualité de l'alimentation ou son découplage ne sont pas indiqués par exemple, alors que les mesures sont réalisées en mode impulsionnel (comme dans la partie 4.3). En utilisant la loi d'Ohm, le courant de drain Id=0,2A est obtenu en appliquant 25V aux bornes d'une résistance de 25V/0.2A=125Ω. On suppose que la tension V_{DS} est nulle lorsque le MOSFET est totalement passant. En pratique, on sait que la résistance minimale de notre composant est environ 2Ω (pour V_{GS}=10V) donc cela ajoute une incertitude de quelques pourcents. Cette incertitude est inférieure à celle de la mesure à l'oscilloscope donc elle est négligée.

La mesure est effectuée avec un courant relativement fort (la moitié du courant maximal). Le temps de commutation sera plus court avec des courants plus faibles et surtout des tensions plus faibles. On peut approximer la vitesse de transition à 25V/4ns=6V/ns, donc il suffirait théoriquement d'une demi-nanoseconde pour monter de 0V à 3V. Peut-on déduire que le BS170 peut osciller à 1GHz ? Malheureusement, les capacités et inductances parasites ne le permettent probablement pas, mais on peut espérer travailler facilement à plusieurs dizaines de mégahertz, en analogique comme en numérique.

4.5 Références pour commander

La dernière table ne vous intéressera que si vous comptez acheter des BS170 par bobines entières (on ne sait jamais). On devine le fonctionnement du système de suffixes puisque le G se retrouve tout à la fin de la deuxième référence BS170RLRA**G**. C'est donc bien l'indication d'un composant dit « *green* » (sans plomb ni autres substances nocives).

5. LA TROISIÈME PAGE

La page numéro 3 de la datasheet contient des schémas, des diagrammes et des courbes..

Les figures 1 et 2 expliquent comment les valeurs des pages précédentes ont été mesurées. Cela a été examiné dans les paragraphes précédents et nous avons aussi vu que ces valeurs dépendent de nombreuses conditions. Les courbes des figures 3, 4, 5 et 6 expliquent les relations entre ces valeurs. On peut voir si les relations sont linéaires, logarithmiques, exponentielles...

5.1 Sensibilité à la température

La courbe de la figure 3 vient compléter la partie 4.2.1 qui traite de la tension de déclenchement. On voit que cette mesure est inversement proportionnelle à la température. La courbe est très linéaire et normalisée pour lire 1,0 vers 25°C. En réalité, ce n'est pas vraiment une mesure, car le PDF contient une simple ligne droite qui relie les points 1,2 à -50°C et 0,6 à 150°C...

BS170G

RESISTIVE SWITCHING

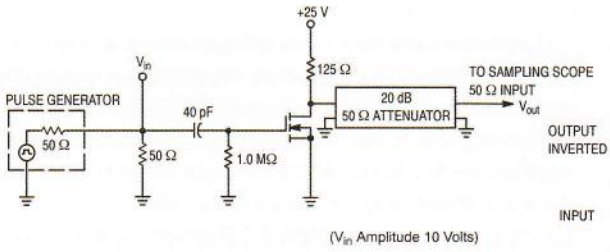


Figure 1. Switching Test Circuit

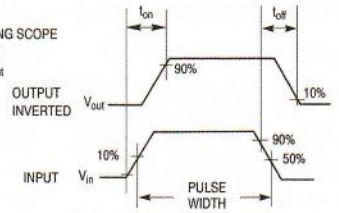


Figure 2. Switching Waveforms

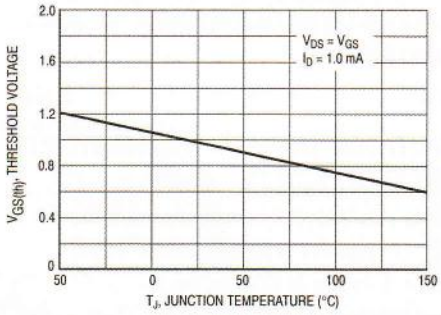


Figure 3. $V_{GS(th)}$ Normalized versus Temperature

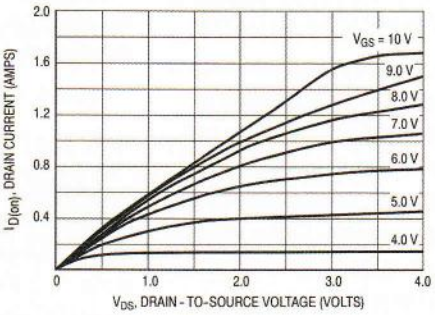


Figure 4. On-Region Characteristics

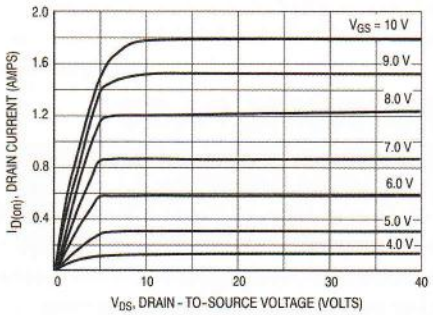


Figure 5. Output Characteristics

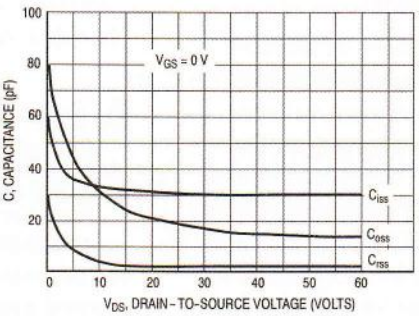


Figure 6. Capacitance versus Drain-To-Source Voltage

Fig. 3 : Page n°3.

5.2 Courbes de transfert

S'il ne fallait retenir que quelques informations, ce serait les figures 4 et 5 du PDF. Elles sont plus complexes, avec une superposition de 7 courbes comprenant des zones linéaires et arrondies. Elles sont aussi les plus intéressantes, car elles offrent un aperçu global du comportement électrique du composant. La figure 4, allant de 0 à 4V, est une vue agrandie de la figure 5, qui va jusqu'à 40V. On voit facilement que le transistor commence à saturer lorsque la tension de drain atteint 5V, donc il est plus efficace dans les basses tensions.

Ces courbes sont très utiles : on voit aussi un lien direct entre trois grandeurs physiques différentes. Le quadrillage permet d'en déduire une en fonction des autres. Deux de ces grandeurs étant un courant et une tension, on peut aussi appliquer la loi d'Ohm et évaluer la résistance en fonction de la tension de grille.

5.3 Courbes de capacitance

De toutes les courbes et données, c'est peut-être la plus étrange (du moins pour moi). La figure 6 contient trois courbes montrant la baisse de capacitance lorsque la tension drain-source augmente. D'ailleurs, je ne m'attendais pas à une baisse de ce type et je m'attendais plutôt à une augmentation, étant habitué à ce comportement dans les diodes de type varicap.

D'autre part, la désignation de chaque courbe n'est pas évidente. La partie 4.3 a défini la mesure C_{iss} comme étant la capacitance d'entrée, on peut alors supposer que C_{oss} serait la capacitance de sortie, mais alors que signifie C_{rss} ? Et par quel circuit ces valeurs ont-elles été mesurées ? Et est-ce que la tension moyenne de grille (V_{GS}) influe aussi la capacitance ? Tant de questions et bien d'autres qui ne sont pas pertinentes pour nos petits montages à basse fréquence et basse tension, mais qui méritent une réponse.

6. LA QUATRIÈME PAGE

Pour finir, un peu de mécanique. Il n'y a plus de paramètres électriques (à part l'assignation des broches), mais du dessin technique et des spécifications de dimensions. Ici ce n'est pas très original (puisque le format TO-92 est très connu), ni critique (il est possible d'ajuster manuellement la forme des pattes quand on soude le composant). Cette partie finale est toutefois vitale quand le boitier est très particulier ou nécessite de créer une empreinte de circuit imprimé. Certaines datasheets contiennent même des conseils pour les dessiner. Le brochage devient plus compliqué à mesure que le nombre de broches augmente. D'autres informations pratiques peuvent se trouver, concernant la fixation d'un radiateur ou la forme recommandée d'une piste de circuit imprimé (Figure 4, page suivante).

CONCLUSION

Cette datasheet est concise, mais malheureusement incomplète. Il manque par exemple les caractéristiques thermiques détaillées. Le composant est donc limité à des applications de faible puissance à température ambiante (entre 0 et 50°C environ). Nous n'avons même pas d'information sur la diode intrinsèque ! Pour en savoir plus, il faudra tester le composant en conditions réelles, mais les informations mesurées ne s'appliqueront qu'au lot du composant.

Je trouve paradoxal d'écrire autant à partir de si peu d'informations. Pour un document censé être une référence, claire et précise, il pose plus de questions qu'il ne répond. Il n'y a même pas de glossaire expliquant les suffixes, que l'on doit déduire de la partie « *Ordering Informations* ». Dans le cas présent, choisir la datasheet la plus concise s'est retourné contre nous.

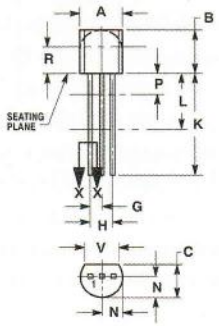
Les performances du BS170 sont « juste adéquates » pour l'utilisation que j'envisage, mais cela ne va pas plus loin. Les avantages de cette référence ont été listés au début, mais la performance n'en fait pas partie. On trouve des MOSFET plus rapides et plus puissants, mais ils sont aussi plus chers.

Malgré cela, le BS170 est un MOSFET canal N très utile, répandu et suffisant pour de nombreux montages de débutants. Grâce à la lecture de cette datasheet, vous saurez quels paramètres ne pas dépasser et vous pourrez déterminer quel composant conviendra mieux si le BS170 ne vous suffit plus. YG

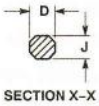
BS170G

PACKAGE DIMENSIONS

TO-92 (TO-226)
CASE 29-11
ISSUE AM



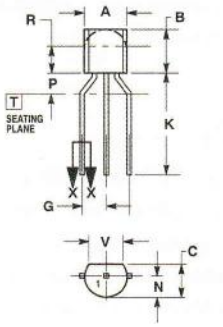
STRAIGHT LEAD
BULK PACK



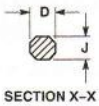
SECTION X-X

NOTES:
1. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
2. CONTROLLING DIMENSION: INCH.
3. CONTOUR OF PACKAGE BEYOND DIMENSION R IS UNCONTROLLED.
4. LEAD DIMENSION IS UNCONTROLLED IN P AND BEYOND DIMENSION K MINIMUM.

	INCHES		MILLIMETERS	
DIM	MIN	MAX	MIN	MAX
A	0.175	0.205	4.45	5.20
B	0.170	0.210	4.32	5.33
C	0.125	0.165	3.18	4.19
D	0.016	0.021	0.407	0.533
G	0.045	0.055	1.15	1.39
H	0.095	0.105	2.42	2.66
J	0.015	0.020	0.39	0.50
K	0.500	---	12.70	---
L	0.250	---	6.35	---
N	0.080	0.105	2.04	2.66
P	---	0.100	---	2.54
R	0.115	---	2.93	---
V	0.135	---	3.43	---



BENT LEAD
TAPE & REEL
AMMO PACK



SECTION X-X

NOTES:
1. DIMENSIONING AND TOLERANCING PER ASME Y14.5M, 1994.
2. CONTROLLING DIMENSION: MILLIMETERS.
3. CONTOUR OF PACKAGE BEYOND DIMENSION R IS UNCONTROLLED.
4. LEAD DIMENSION IS UNCONTROLLED IN P AND BEYOND DIMENSION K MINIMUM.

	MILLIMETERS	
DIM	MIN	MAX
A	4.45	5.20
B	4.32	5.33
C	3.18	4.19
D	0.40	0.54
G	2.40	2.80
J	0.39	0.50
K	12.70	---
N	2.04	2.66
P	1.50	4.00
R	2.93	---
V	3.43	---

STYLE 30:
PIN 1. DRAIN
2. GATE
3. SOURCE

Fig. 4 : Page n°4.

ON Semiconductor and ◡ are registered trademarks of Semiconductor Components Industries, LLC (SCILLC). SCILLC reserves the right to make changes without further notice to any products herein. SCILLC makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does SCILLC assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation special, consequential or incidental damages. "Typical" parameters which may be provided in SCILLC data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including "Typicals" must be validated for each customer application by customer's technical experts. SCILLC does not convey any license under its patent rights nor the rights of others. SCILLC products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the SCILLC product could create a situation where personal injury or death may occur. Should Buyer purchase or use SCILLC products for any such unintended or unauthorized application, Buyer shall indemnify and hold SCILLC and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that SCILLC was negligent regarding the design or manufacture of the part. SCILLC is an Equal Opportunity/Affirmative Action Employer. This literature is subject to all applicable copyright laws and is not for resale in any manner.

PUBLICATION ORDERING INFORMATION

LITERATURE FULFILLMENT:
Literature Distribution Center for ON Semiconductor
P.O. Box 5163, Denver, Colorado 80217 USA
Phone: 303-675-2175 or 800-344-3860 Toll Free USA/Canada
Fax: 303-675-2176 or 800-344-3867 Toll Free USA/Canada
Email: orderlit@onsemi.com

N. American Technical Support: 800-282-9855 Toll Free USA/Canada
Europe, Middle East and Africa Technical Support:
Phone: 421 33 790 2910
Japan Customer Focus Center
Phone: 81-3-5773-3850

ON Semiconductor Website: www.onsemi.com

Order Literature: http://www.onsemi.com/orderlit

For additional information, please contact your local Sales Representative

BS170/D

RÉFÉRENCES

[1] Datasheet du BS170 de ON Semiconductor,
http://www.onsemi.com/pub_link/Collateral/BS170-D.PDF

[2] Toshiba, « *Power MOSFET in Detail* », chapitre « *3. Maximum Ratings* »
https://toshiba.semicon-storage.com/info/docget.jsp?did=13414 (fichier nommé *application_note_en_20041001.pdf*)

[3] Article « Nos amis les MOSFET », publié dans le numéro 12 de *Hackable*, p. 70

ACTUELLEMENT DISPONIBLE

LE 1ER HORS-SÉRIE DE HACKABLE !

VOUS UTILISEZ DÉJÀ LA RASPBERRY PI ? METTEZ-VOUS À L'ARDUINO !

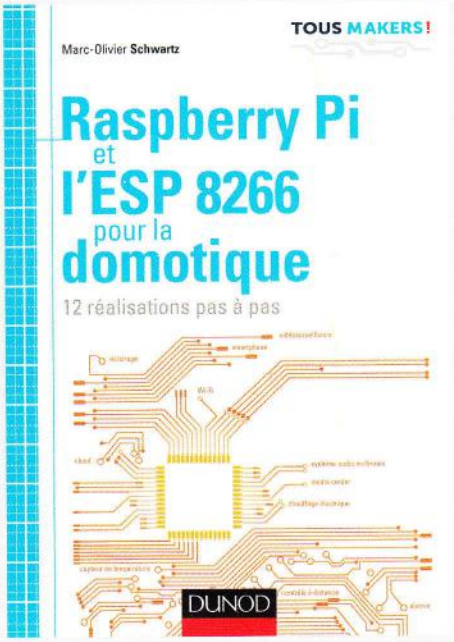
NE LE MANQUEZ PAS

CHEZ VOTRE MARCHAND DE JOURNAUX ET SUR :

www.ed-diamond.com

TOUS MAKERS !

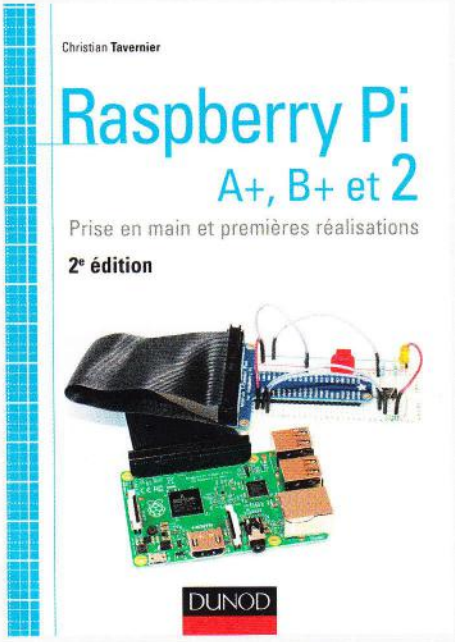
RÉVÉLEZ VOTRE POTENTIEL PAR LA CRÉATION



9782100746835, 200 p., 24,90 €

MARC-OLIVIER SCHWARTZ

Initiez-vous à la domotique avec le Raspberry Pi associé à la puce Wi-Fi ESP 8266 : 12 projets concrets pour rendre votre maison plus « intelligente ».



9782100742639, 352 p., 24,90 €

CHRISTIAN TAVERNIER

Cet ouvrage décrit les différentes versions de Raspberry Pi et explique comment le configurer et le paramétrer correctement.

9782100738106, 192 p., 22 €

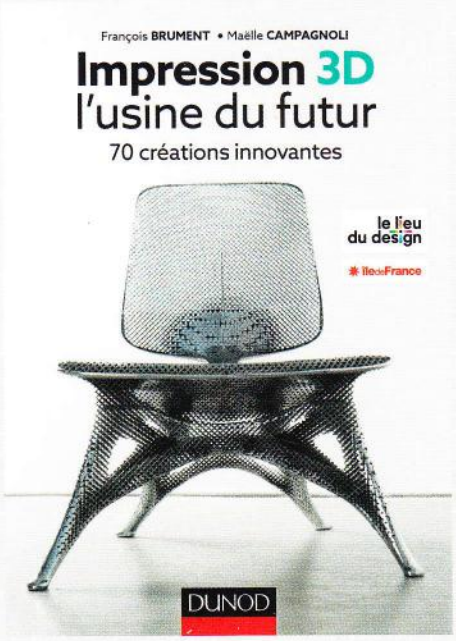
PATRICE OGUIC

Un guide pas à pas pour construire votre machine CNC idéale pour la gravure et le perçage des circuits imprimés, les maquettistes et les modélistes.

9782100742561, 160 p., 19,90 €

PASCAL LIÉGEOIS

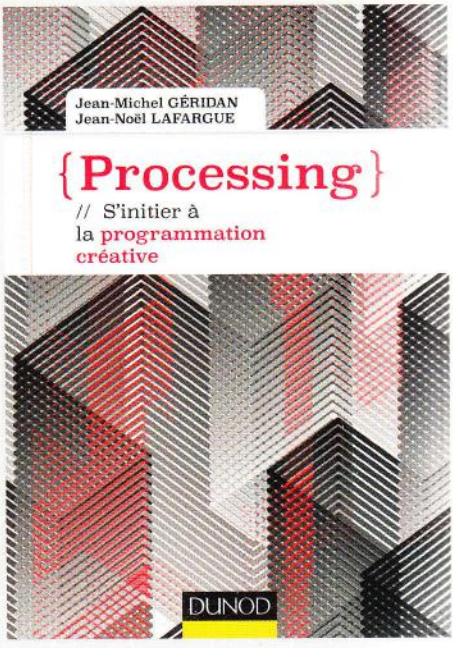
Réalisez un drone terrestre radiocommandé, contrôlable à vue ou à distance, et pouvant embarquer un appareil photo ou une caméra.



9782100746842, 160 p., 19,50 €

FRANÇOIS BRUMENT, MAËLLE CAMPAGNOLI

À travers 70 réalisations, cet ouvrage présente les nombreuses possibilités de l'impression 3D, afin de comprendre en quoi elle va tout changer.



9782100737840, 280 p., 29 €

JEAN-MICHEL GÉRIDAN, JEAN-NOËL LAFARGUE

Tout savoir sur Processing, le logiciel open source pour les créateurs qui veulent produire des installations interactives.

TOUT LE CATALOGUE SUR **WWW.DUNOD.COM/TOUS-MAKERS**

DUNOD
ÉDITEUR DE SAVOIRS