NOUVEAU – NOUVEAU – NOUVEAU – NOUVEAU – NOUVEAU – NOUVEAU – NOUVEAU

N°2 | SEPTEMBRE-OCTOBRE 2014

HACKABLE MAGAZINE

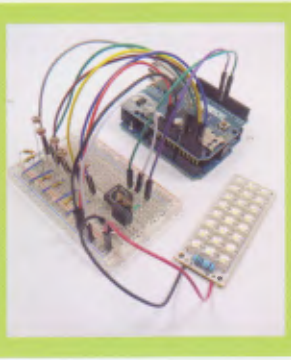
DÉMONTEZ | COMPRENEZ | ADAPTEZ | PARTAGEZ

France METRO : 7,90 € – CH : 13 CHF – BEL/PORT.CONT : 8,90 € – DOM TOM : 8,50 € – CAN : 14 $ cad – TUNISIE : 18 TND – MAR : 100 MAD

ARDUINO

Faites tweeter votre boîte à lettres lorsque vous avez du courrier

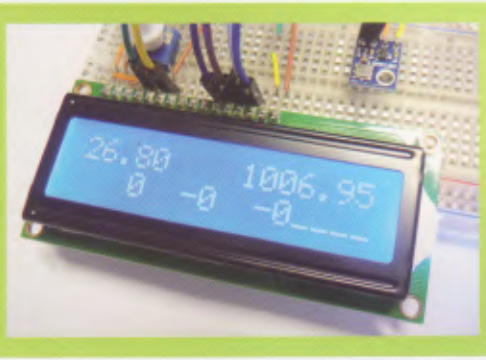
p. 22



ARDUINO

Transformez votre Arduino en baromètre pour prévoir la météo

p. 94



ACTUALITÉ

Retour sur la 1ère édition de la Maker Faire Paris

p. 06

ARDUINO

Quel système de notification pour mon projet connecté ?

p. 88

Que faire avec un récepteur TNT USB ?

HACK : EXPLOREZ LES ONDES !

p. 36

- Configurez la réception
- Lisez le trafic aérien
- Captez la radiomessagerie



RASPBERRY PI

Emportez votre serveur Minecraft partout avec vous !

p. 78



OUTILS

Bien choisir son multimètre : une question de sécurité

p. 10



L 19338 - 2 - F: 7,90 € - RD

À DÉCOUVRIR DÈS LE 05 SEPTEMBRE !

PRENEZ EN MAIN LES TECHNOLOGIES RFID / NFC



NE LE MANQUEZ PAS : OPEN SILICIUM N°12 !

DISPONIBLE DÈS LE 05 SEPTEMBRE 2014

CHEZ VOTRE MARCHAND DE JOURNAUX

ET SUR : boutique.ed-diamond.com

ÉDITO



Il faut que je vous raconte un truc...

Sans pour autant dire que ce numéro du magazine n'a pu être terminé dans les temps que grâce à cela, je dois avouer qu'un distant artiste a apporté une contribution bien involontaire et sans le savoir à ce numéro. Si vous me lisiez dans d'autres publications ou m'avez d'ores et déjà percé à jour suite à mon précédent édito, vous devez savoir que la technologie n'est pas, pour moi, un passe-temps ou une activité, mais bel et bien quelque chose qui imprègne et tisse la trame de tout ce que je fais.

Je ne vais pas dire que je ne pourrai pas vivre sans. Non parce que c'est faux, mais simplement parce que cela n'aurait rien d'exceptionnel. Au sens strict du terme, plus personne ne peut plus vivre sans et ce depuis très longtemps (presque depuis le « début » en réalité). De plus, si on nous privait de technologie, la première chose que nous ferions serait de la recréer, de préférence dans un temps suffisant pour assurer au mieux notre survie. Rappelez-vous, la technologie est la mise en pratique du savoir, or le savoir et sa transmission sont le propre de l'humain, du moins tout ce qui dépasse le niveau de la maman chat qui montre aux chatons ce qu'est une litière et ce à quoi elle sert (nous, on est une espèce plus évoluée, il nous faut un système hydraulique complexe et de la cellulose manufacturée pour ce type de choses).

Mais je diverge (cf P. Desproges) alors que je manque de place. Ce à quoi je fais référence est tout simplement le fond musical omniprésent lors de la préparation, la rédaction et la composition de ce numéro. Pour apporter de l'énergie dans la réalisation d'un magazine traitant de bidouilles, de hacks, de prise de contrôle et de détournement des technologies, quoi de mieux qu'un fond musical puissant qui lui-même repose sur de tels concepts ? Là, vous devez avoir un air interloqué ou du moins dubitatif. Un fond musical reposant sur le hack ? Qu'est-ce que c'est ?

En résumé, on pourrait répartir les créateurs de musique en deux catégories : ceux qui utilisent les instruments de la façon prévue par leur constructeur et les autres. Ceci est applicable aux instruments classiques, mais aussi aux instruments électroniques. Ainsi, utiliser un mélange technologique regroupant des appareils récents avec des Gameboy (fin des années 90) et des ordinateurs personnels (années 80) pousse le concept du *chiptune* bien au-delà de la simple nostalgie des « bonnes vieilles sonorités 8 bits ». C'est ainsi que j'ai découvert un artiste indépendant polonais du nom de LukHash ayant quelques albums à son actif (cf. Bandcamp, YouTube, Spotify, Jamendo, etc.) qui ont littéralement rythmés une bonne partie de ce numéro.

C'est un réel plaisir, inspirant une grande motivation, que d'entendre le fruit de ce travail (ceci dit, on aime ou on n'aime pas) et surtout de constater que des technologies de 20 ans d'âge ou plus sont encore aujourd'hui poussées à leur extrême de façon créative et avec un résultat impressionnant. Entendre de tels morceaux, c'est littéralement, écouter le son du hack et de la créativité, aussi bien l'ancienne (circuit, design, architecture) que la nouvelle (composition, interprétation). Comme pour les jeux et les démos sur ces plateformes, ceci nous rappelle que même sur un matériel qui ne change pas, l'évolution est là.

Je vous souhaite une excellente lecture de ce numéro qui a comme sujet central, lui aussi, une technologie qu'on peut qualifier de bien plus évoluée (SDR) que le concept de départ (TV).

Denis Bodor

Hackable Magazine
est édité par Les Éditions Diamond

LES ÉDITIONS DIAMOND
STIMULANT DE MATIÈRE GRISE

B.P. 20142 – 67603 Sélestat Cedex
Tél. : 03 67 10 00 20 – Fax : 03 67 10 00 21
E-mail : lecteurs@hackable.fr
Service commercial : cial@ed-diamond.com
Sites : boutique.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
IMPRIMÉ en Allemagne - PRINTED in Germany
Dépôt légal : À parution,
N° ISSN : en cours
Commission paritaire : en cours
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

SOMMAIRE

COURRIER DES LECTEURS
04

ACTUALITÉ
06
Manifestation :
Retour sur la Makerfaire Paris 2014

ÉQUIPEMENT
10
Les indispensables :
le multimètre numérique

ARDUINO'N'CO
22
Créez une boite à lettres qui tweete

EN COUVERTURE
36
Émission radio : vous n'imaginez pas comme ça cause autour de nous
44
Installation et utilisation de RTLSDR sous Linux
52
Utilisation d'une radio logicielle sous Windows
59
Bonus : Améliorer la réception c'est réduire le bruit
60
ADS-B ou comment écouter les avions tout autour de vous
68
La radiomessagerie : captez les messages d'un autre temps

EMBARQUÉ & INFORMATIQUE
78
Minecraft et la Raspberry Pi

REPÈRE & SCIENCE
88
Choisir un système de notification adapté à son projet

DOMOTIQUE & ROBOTS
94
Prédire la météo avec son Arduino : le baromètre BMP180

ABONNEMENT
71/72
Bon d'abonnement

COURRIERS DES LECTEURS

Bonjour ou bonsoir,

Je voulais tout d'abord vous remercier pour ce magnifique magazine qu'est Hackable. La façon dont vous abordez le « hack » est superbement pédagogique et intuitifs. On a du déjà vous le dire, mais votre magazine m'a redonné goûts à ce passe-temps. J'ai enfin pu utiliser ce pack Arduino offerte à Noël !

Au niveau de l'article sur la loi d'Ohm, tout est très claire et compréhensible. Le seul oublie est le fait que vous n'évoquiez pas la manière pour identifier la valeur d'une résistance. Ceci fait peut-être partie des bases, mais personnellement je n'avais aucune idée à quelle valeur correspondaient les bandes de couleur.

J'ai aussi une proposition au niveau du contenu, en lisant cet article je me suis dit : une série de poster détaillant les caractéristiques de composants électronique à chaque magazine pourrais vraiment être ludique et resterais dans le thème. Par exemple, un poster sur la résistance avec l'explication des différentes couleurs, comment le représenter sur un schéma électrique, de quoi est-il composé, à quoi sert-il, etc. Je vois bien ce concept sur différent composant : condensateur, RTC, LED, régulateur de tension, puce FTDI, ... Je pense à des posters qui auraient tous le même style et dont le but est de l'accrocher dans son « labo ».

Merci pour ce super magazine et pour vos efforts.

Guillaume

Tout d'abord merci pour ces commentaires sympathiques. En effet, glisser un tableau avec les codes de couleurs aurait été une bonne idée mais nous étions tellement concentrés sur le fait de rendre l'aspect théorique compréhensible que ça nous a un peu échappé. Votre idée de poster est très intéressante, nous allons réfléchir à la question en prenant en compte les contraintes techniques de fabrication et évaluer ce qui peut être fait.

D.B.

Bonjour. Tout d'abord je voulais vous féliciter pour votre magazine, qui change de l'ordinaire et qui est très intéressant et enrichissant. J'aimerais voir un petit tutoriel pour fabriquer un détecteur de courrier dans la boîte au lettre (avec une led qui s'allume si courrier). Une petite chose qui à mon avis, peut intéresser un grand nombre de lecteurs. Sinon continuez comme le premier, c'est parfait.

Cordialement. Kévin.

Vous devriez être ravi par ce numéro. Votre message est précisément l'élément déclencheur du projet de la boite à lettres qui tweete que vous trouverez dans les pages qui suivent. C'est certes un peu plus qu'une led qui s'allume mais il est parfaitement possible de « réduire » les fonctionnalités.

D.B.

Cher service lecteurs,

J'ai acheté votre magazine il y a quelques jours et l'ai activement lu ce week-end. Le contenu est intéressant et m'incitera à tester l'électronique, notamment avec Arduino.

Cependant, une chose m'a frappé : les fautes d'orthographe et d'accord ! Il y a au moins une faute par article (j'ai lu les 2/3 du magazine), et pour certains, bien une par page. Je trouve cela inadmissible qu'un magazine que dans la presse, il y ait autant de fautes. Je conçois qu'il puisse y avoir des coquilles, fautes de frappe, erreurs de mise en page, et j'en passe, mais pas des fautes si simples et pas autant, pas dans chaque article.

Le pire, c'est que la plupart de ces erreurs sont des fautes qu'un correcteur orthographique sait corriger. Malheureusement pour vous, lorsque je vois quelques fautes, telles des « er » à la place des « é », je finis par ne plus voir que ces fautes et vérifier tous les accords et conjugaisons ; j'en trouve donc plus qu'avec une lecture normale... Et cela me la gâche un peu... [...]

Je ne suis ni professeur de français, ni littéraire : j'ai une formation bac S et suis développeur ; cependant, je fais attention lorsque j'écris des courriers/courriels à mes clients, et ferais encore plus attention si j'écrivais un article public pour un magazine ou livre...

Pour le bien de tous, merci de faire attention lors de la rédaction de vos prochains articles.

Mathieu

Vous avez absolument raison et la faute est nôtre. Vous n'êtes pas le seul à avoir relevé ce problème et nous vous devons à tous une explication (qui n'est en rien une excuse). La masse importante de fautes est en lien direct avec la pression subie par le service de relecture. L'énergie investie pour créer le premier numéro a majoritairement trouvé son utilité dans le fond rédactionnel et le soucis de présenter des informations, qui restent très techniques, de la manière la plus claire et simple possible. Le fait de mettre la pression et de réduire le temps impartie à la relecture (vous savez, le « il me le faut pour dans 5 minutes ») a eu des conséquences bien malheureuses. C'est une regrettable maladresse. Suite aux messages reçus, nous avons donc immédiatement pris les mesures qui s'imposent pour organiser la relecture avec une cadence bien plus viable et les fruits de ces changements doivent normalement déjà apparaître dans le présent numéro.

D.B.

Bonjour Denis (et toute l'équipe ?),

Juste un petit mot pour te/vous dire que je trouve hackable vraiment inspirant. J'ai déjà ma petite idée de projet, reste le matériel a recevoir (dont un starter kit). J'espère que vous ferez un petit article sur comment documenter son projet. Et pourquoi pas ou le partager, les bonnes adresses et les bonnes communautées. Merci encore et bonne continuation.

Chaleureusement, Florentin

Et effet, documenter son projet et trouver la bonne manière de diffuser ses expériences et réalisations sont deux facettes importantes du partage d'informations. Les plateformes sont nombreuses même si certaines d'entre elles sont plus populaires que d'autres (comme GitHub). Nous gardons donc cette idée sous le coude et le sujet fera l'objet d'un article dans un prochain numéro.

D.B.

Bonsoir,

Tout d'abord, merci pour ce numéro 1, je l'ai trouvé très pédagogique et accessible tout en abordant certaines notions un peu plus avancées. J'aurais juste une question par rapport à l'article sur la station de soudage : vous y parlez de pannes plates de 2mm de largeur. Pourquoi sont-elles à privilégier par rapport aux pannes coniques à pointes fines (dans quels cas ?) ?

Merci. Bien cordialement,

Bruno

L'article recommande, en effet, les pannes plates « pour un usage général ». Il faut ici comprendre « pour des soudures courantes » (connecteurs, composants traversants, etc) avec des cartes et des composants généralement utilisés lorsqu'on débute (les mêmes que ceux utilisés avec les breadboards). Nous ne souhaitions pas sous-entendre que les pannes coniques sont inutiles. Elles sont indispensables dans certains cas et en particulier pour la soudure des composants CMS/SMD.

Le problème qui se pose lorsqu'on utilise une panne conique à pointe fine avec une résistance ou un condensateur, tel qu'on en utilise sur une platine à essais, est la transmission de l'énergie thermique. Comme la pointe est fine, la surface de contact avec le circuit et la patte du composant est très réduite. De plus, la patte elle-même fait office de radiateur.

Mais à chaque situation il existe un outil adapté. Il est tout aussi cauchemardesque de souder des composants traversants avec une panne à pointe conique que de souder du SMD avec une panne plate, surtout de 2 mm.

D.B.

Bonjour,

Je viens de recevoir le premier numéro de « Hackable » et je l'ai trouvé tellement intéressant que je n'ai pas pu le ranger sans en avoir survolé tous les articles. Enfin, tous les articles sauf celui de la cigarette, sujet dont le choix est discutable. A quand un article sur la chaise électrique ?

Donc première impression très favorable, mais déjà une lacune pour un lecteur qui, comme moi, part de zéro : quelle plate-forme pour mon projet ? Je veux détecter le courrier dans ma boite aux lettres, surveiller le refroidissement de mes PCs ou construire une centrale d'alarme. Arduino convient-il mieux que Raspberry ?

[...]

Cordialement, André

L'un des objectifs du magazine est de détailler techniquement le fonctionnement des objets technologiques qui nous entourent. La e-cigarette est entrée dans les mœurs de beaucoup de personnes et notre but était clairement de démystifier son fonctionnement, non d'en faire la promotion (ni pour un non-fumeur, ni pour un fumeur). Nous ne pensons pas qu'éviter certains sujets favorise leur disparition, bien au contraire. En guise de réponse, je me permettrai simplement de citer Carl Sagan : « Nous vivons dans une société extraordinairement dépendante de la science et de la technologie, dans laquelle très peu de personnes connaissent quoi que ce soit en science et en technologie. Ceci préfigure clairement un désastre. ». Essayons d'éviter les désastres et parlons ouvertement des technologies, de toutes les technologies.

L'autre point que vous soulevez concernant le choix d'une plateforme ne se pose généralement pas ainsi car il n'existe pas de matériel universel. En ces termes, Arduino et Raspberry Pi (ou toutes autres plateformes) ne peuvent se placer en concurrence. L'un est une carte à microcontrôleur et l'autre un nano-ordinateur. Il faut avant toute chose définir son projet et établir un cahier des charges en répondant à quelques questions : « que doit faire mon montage ? », « comment doit-il le faire ? », « quelles fonctionnalités sont idéales, souhaitables et indispensables ? ». Si nous prenons l'exemple de la sonnette du mois dernier, l'Arduino était suffisant mais s'il avait été question de capturer une image du « sonneur », il aurait été bien plus raisonnable de basculer sur une Raspberry Pi. Inversement, certains projets, utilisant un grand nombre d'entrées/sorties par exemple, ne permettent pas d'utiliser une Pi sans ajout de matériel. Le coût global au regard des fonctionnalités inutilisés (HDMI, audio, puissance) écarterait ainsi la plateforme.

Tout dépend finalement de la nature de votre projet et surtout de la manière dont vous souhaitez qu'il réalise cette tâche.

D.B.

MANIFESTATION : RETOUR SUR LA MAKERFAIRE PARIS 2014



Certes, c'était en juin et nous avons tout fait pour que le premier numéro de ce magazine soit prêt pour ce moment mémorable. Donc fatalement, le retour sur cette première édition parisienne arrive deux mois après les faits. Mais comment se retenir d'exprimer notre ravissement et les émotions ressenties, d'autant que, qui dit « première édition à succès » dit forcément « ça va être long d'attendre la prochaine ».

Cet article s'adresse principalement aux lecteurs n'ayant pas eu l'occasion de se rendre à Paris les 21 et 22 juin derniers et plus précisément au 104 pour la première édition de la Maker Faire à la française. Loin de moi l'idée de sous-entendre que la Bretagne n'est pas en France ou que le dicton/devise « Ni Français, ni Breton, Malouin suis » s'applique à la Maker Faire. Il y eu, en effet, un précédent dans notre pays mais sous la forme d'une mini Maker Faire, mi-octobre 2013 dans la belle ville de Saint Malo. Appelons cela un tir de somation puisque les organisateurs étaient les mêmes pour les deux manifestations.

1. QU'EST QU'UNE MAKER FAIRE ?

Depuis quelque temps déjà, un mouvement est en marche. De plus en plus de personnes se rendent compte qu'il est possible de « faire des choses », de créer et d'inventer. A la différence d'autres régions du monde où la créativité est omniprésente, ceci est un véritable phénomène outre-atlantique. Il est vrai qu'en Europe, le loisir créatif, la décoration, la couture ou encore le simple fait de réparer soi-même ses objets cassés ont toujours été présents et sont restés bien ancré dans la culture. Les habitants d'autres pays, moins favorisés, quant à eux n'ont guère le choix, récupérer, modifier et adapter les objets courant ou endommagés pour un usage alternatif n'est pas culturel, c'est une solution pour survivre. Aux état-unis, patrie du capitalisme, du tout consommable et du jetable, le fait de « faire soi-même » est relativement atypique. Ceci au point qu'un terme a été inventé et est consacré à cette activité : le DIY pour *Do It Yourself* (« faites-le vous-même »), sous-entendu « ne l'achetez pas tout fait ».

Le DIY américain s'applique à un très vaste domaine d'activités ou plus précisément, des domaines où justement le fait de faire soi-même les choses a eu tendance à disparaître ses dernières années. La-bas, des loisirs simples tel que faire la cuisine, décorer son intérieur avec des réalisations « maison » ou entretenir son potager de trois carottes et deux poireaux en utilisant quelques astuces de base, deviennent du DIY. C'est caricatural mais j'exagère à peine, car il y a une réelle différence culturelle. Chez nous construire une mangeoire pour les oiseaux c'est faire un peu de bricolage du dimanche avec trois bouts de bois, là-bas cela devient un projet DIY pour le weekend.

Vous l'avez compris, le phénomène a prit une telle ampleur et touche tellement de domaines qu'il apparaît naturellement des sous-domaines. Ainsi, la culture *maker* est une sous-culture du DIY principalement axée autour de l'électronique, la robotique, l'impression 3D, les machines, outils ainsi que les activités en relation ou dépendantes. Ceci détache sensiblement la culture maker des activités qui entrent dans d'autres catégories comme le *crafting* que l'on appel chez nous « loisir créatif » ou encore l'origami, la cuisine, la broderie, etc.

Voyant les choses ainsi bouger, l'éditeur O'Reilly créa *Make:* en 2005, une publication très typé US présentant les réalisations des *makers* et qui depuis a évolué et s'est plus ou moins détaché de l'éditeur (la publication est aujourd'hui édité par *Maker Media*, société dirigée par l'un des co-fondateur d'O'Reilly) pour finalement trouver un modèle économique qui semble plus ou moins lié à la vente de matériel pour makers.

Dans la foulée, toujours aux états-unis, ont fait leur apparition des manifestations dès 2006, sponsorisées

par la publication et d'autres partenaires, permettant aux makers de se rencontrer, présenter leurs projets et partager leurs expériences. Ces dernières années, ces manifestations, appelées *Maker Faire*, ont attiré de plus en plus de curieux et d'acteurs, au point qu'il existe maintenant un kit de démarrage, permettant de créer sa mini Maker Faire locale indépendante. Les manifestations (mini ou non) sont de plus en plus fréquentes à travers le monde et en juin dernier, l'une d'elle à même trouver sa place à la maison blanche (les américains ont le sens du sensationnel et de la démesure, aucun doute).

Ainsi, c'est avec cette structure et cette rationalisation qu'une Maker Faire permet de réunir à la fois ceux qui réalisent et créent, avec les amateurs curieux de savoir « comment s'y mettre », les associations, les autres lieux de rencontres (fablab, hackerspace, etc), ainsi que les revendeurs d'équipements, de petits matériels et de kits de découverte.

2. UNE PREMIÈRE POUR PARIS

Ce sont donc les 21 et 22 juin derniers qu'à eu lieu la Maker Faire Paris, organisée d'une main experte par le FabShop, importateur et distributeur d'imprimantes 3D et le cent quatre paris, un espace de résidences, de production et de diffusion artistique dans le 19ème arrondissement à Paris.

Nous avions à cette occasion un stand destiné à présenter notre nouvelle publication et étions donc au cœur de l'action de cette première édition. Une première que l'on pourrait sans peine qualifier de fantastique succès car l'affluence était agréablement surprenante.

Les visiteurs ont ainsi pu découvrir bien des aspects de la culture maker et du DIY avec un vaste nombre de domaines allant de l'impression 3D bien sûr en passant par les cartes Arduino et consort mais également la sérigraphie,



la couture, la menuiserie, l'électronique, la biologie, la domotique, la robotique ou encore le tissage.

L'ambiance était, pendant deux jours, au beau fixe et il régnait dans les différents halls une atmosphère presque électrique. Les échanges et les discussions allaient bon train à la fois entre les stands mais aussi et surtout avec la foule importante de visiteurs. Nombreux étaient ceux venu par simple curiosité et qui après leur visite sont repartis la tête pleine de projets et de motivation pour débuter.

La pratique n'était pas en reste et les partenaires de la manifestation dont Leroy Merlin, MathWorks, Intel, Trimble SketchUp, Autodesk et Roland qui ont proposés leur stand avec démonstrations et prise en main. Ainsi l'atelier d'initiation à l'électronique offrait l'occasion d'assembler et souder votre badge Maker Faire en forme de petit robot aux yeux lumineux. C'est avec un réel plaisir qu'on pouvait ainsi voir à toute heure de la journée des dizaines de visiteurs, fer à souder en main, faire fondre l'étain, la plupart pour la première fois, pour assembler les leds et le support de pile.

La robotique, l'impression 3D et les drones n'étaient pas en reste avec des démonstrations impressionnantes qui invariablement





attiraient les visiteurs qui s'empressaient de poser des questions par dizaines, « comment ça marche ? », « où ça se trouve ? », « combien ça coûte ? », « c'est fabriqué comment ? », « c'est quoi les différences entre les modèles ? », « ça vole longtemps ? », « je peux faire tous les objets que je veux ? », etc. De mémoire, et pour avoir assisté à bon nombre de salons et d'expositions, c'est la première fois que nous constatons un tel déluge de questions, mais aussi et surtout de réponses. Les exposants, les makers passionnés mettaient un point d'honneur à apporter des réponses utiles et, par la même occasion, à eux-mêmes glaner quelques idées aussi farfelues qu'intéressantes.

Parce qu'une Maker Faire c'est justement cela. Il ne s'agit pas d'une simple exposition où les présentateurs font les démonstrations et où les visiteurs reçoivent passivement l'information et se laissent convaincre. C'est un réel échange et bien des questions de personnes découvrant les différents stands, de par leurs interrogations, déclenchaient eux aussi un élan créatif. C'est à maintes occasions que j'ai vu, de ci de là, des makers griffonner une idée dans un carnet ou sur un coin de papier (nous avons fait de même à plusieurs reprises).

En tout, ce sont quelques 120 makers qui ont rencontrés et partager des idées, des concepts et des réalisations avec près de 7500 visiteurs, le tout dans une ambiance festive et pleine d'énergie. Qui sait combien de ces visiteurs ont décidé d'explorer eux aussi les différents domaines qu'ils ont découvert pendant ces deux jours et sont à présent eux-même des makers !

2.1 Encore, encore, encore !

En conclusion, et sans surprise, le sentiment général après cette première édition est d'en vouloir plus. Il est indéniable qu'il se passe quelque chose et que cela se propage. C'est une chose d'avoir une conviction et se dire que tout un chacun a effectivement la capacité de créer mais s'en est une autre de voir, de ses propres yeux, que cet engouement est réellement entrain de renaître auprès du public. Or c'est précisément ce que nous avons constaté et bon nombre de makers et de visiteurs, au terme de ces deux journées, ont non seulement été ravi mais surtout rassuré. Car même si dans cette bulle temporelle il était normal de penser en premier lieu « comment je peux moi aussi faire ça » et ensuite seulement « où puis-je trouver ça », cela reste encore trop atypique. Créer ses outils, réaliser ses montages ou démonter les objets du quotidien est toujours un peu hors norme et l'on peut se sentir tantôt seul. Mais preuve est faite qu'exprimer sa créativité n'est pas un comportement étrange, bien au contraire.

Les organisateurs de cette Maker Faire Paris ont accompli un travail remarquable et tout ou presque s'est déroulé de la meilleure manière qui soit, chose qui est relativement exceptionnelle pour une première édition. La seule minuscule ombre au tableau, s'il fallait vraiment en trouver une, fut la désertion complète du lieu le samedi 21 au soir alors qu'un événement musicale était prévu. L'ensemble des participants ont presque intégralement préférés se rendre dans le cœur de la capitale à l'occasion de la fête de la musique, laissant le 104 dans un calme presque surréaliste.



Mais ce furent deux jours de pur bonheur et nous ne pouvons que remercier mille fois l'organisation pour son investissement et son sérieux. Vivement la prochaine édition qui, je pense, ne manquera pas d'attirer encore plus de personnes ! DB

LES INDISPENSABLES : LE MULTIMÈTRE NUMÉRIQUE

D. B.

Le multimètre est sans conteste l'appareil de mesure le plus utile de l'électronicien, amateur comme professionnel. Il vous permet de mesurer, de vérifier et de diagnostiquer aussi bien les composants eux-mêmes que les circuits qui vous passent entre les mains. Le multimètre est au bidouilleur ce que le pinceau est au peintre. Choisir son multimètre ne doit donc pas être quelque chose à faire à la légère.

Un modèle classique de multimètre ultra bas de gamme à 10 ou 15 euros. L'aspect général de l'ensemble va de paire avec le prix et n'inspire absolument aucune confiance malgré les différents marquages. Je ne me risquerai jamais à utiliser ce type de produit pour mesurer une tension ou un courant potentiellement mortel (typiquement un courant domestique à 230V)

Dans le précédent numéro, nous avons vu que l'achat d'équipement découle d'un subtil équilibre entre votre budget et le niveau de qualité que vous devez attendre du matériel. Le multimètre n'échappe pas à cette règle mais les pièges ici sont bien plus nombreux que dans d'autres domaines. Cependant, avec de l'expérience et les bonnes informations, choisir son multimètre n'est pas si difficile, même s'il s'agit d'une décision bien plus importante encore que celle concernant, par exemple, la station de soudage.

1. LE MULTIMÈTRE NUMÉRIQUE, UN OUTIL GÉNÉRIQUE

Contrairement à d'autres équipements, le multimètre est un appareil polyvalent. Il vous sera utile non seulement dans vos activités d'électronicien que pour régler des problématiques de la vie de tous les jours. Mais avant de rentrer dans les détails des points, fonctionnalités et critères importants pour guider votre choix, rappelons ce qu'est précisément un multimètre.

Un multimètre numérique (ou digital) est un appareil de mesure et de test. Il regroupe des fonctions de mesure pour principalement 4 choses :

- la tension (voltmètre),
- le courant (ampèremètre),
- la résistance (ohmmètre),
- la continuité (testeur).

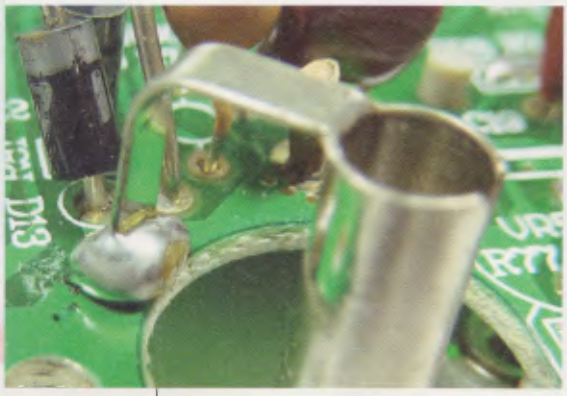
A cela peut s'ajouter la mesure de la fréquence d'un signal, de l'inductance d'une bobine, de la capacité électrique d'un condensateur ou encore de la température. Tout ceci réuni en un seul appareil, le plus souvent portatif et autonome (alimenté par pile).

En pratique, le multimètre est à la fois un outil de mesure et de diagnostic. Il permet de vérifier *in vivo* si les valeurs effectives correspondent aux valeurs théoriques attendues. Ceci peut aller de la simple mesure d'un composant comme une résistance afin de s'assurer qu'elle possède bien la valeur pour laquelle elle est conçue, jusqu'à la vérification d'une tension en sortie d'un circuit. En comparant les valeurs mesurées aux valeurs théoriques, il est ainsi possible de remonter la piste jusqu'au composant défectueux en cas de problème (ou l'erreur commise).

Le multimètre est également l'outil de prédilection de toutes



Un bon multimètre doit avoir ces quatre connecteurs séparant clairement la mesure de tension et de courant afin d'éviter de placer l'appareil dans une situation où vous faites passer le courant au travers de l'instrument alors qu'il s'attend à mesurer une tension. Notez le « UNFUSED » sur le connecteur 20A. Ceci indique clairement qu'en dépassant cette limite aucun dispositif n'est en place pour vous protéger.



Les connecteurs sur le multimètre sont importants. Ici, non seulement ceux-ci se limitent à une soudure directe ce qui implique des contraintes mécaniques fatales au bout que quelques connexion/déconnexion mais, en plus, la soudure elle-même fait littéralement peur !

personnes souhaitant faire de la récupération de composants, afin de s'assurer que ceux-ci sont encore utilisables et fiables, qu'il s'agisse de piles ou de tous autres composants sauvagement dessoudés d'une carte ou d'un appareil mis au rebut (notez que pour la plupart des composants passifs c'est une mauvaise habitude, pas vraiment rentable, sauf pour se sortir d'une situation bloquante).

Il existe un certain nombre de vidéos Youtube, de pages de blogs et de fils de discussions sur les forums concernant l'éternelle question « quel multimètre dois-je acheter ». Les critères peuvent êtres regroupés en plusieurs catégories que nous allons traiter une à une à l'exception de la dernière :

- Les critères généraux permettant d'éliminer sommairement mais surement les modèles à éviter,
- les éléments de décision portant sur des règles simples de sécurité,
- les fonctionnalités nécessaires pour un usage à court, moyen et long termes du matériel,
- et enfin la catégorie regroupant les points que nous qualifierons de « religieux ». Entre dans celle-ci les affirmations du type « Fluke c'est les meilleurs », « ne te pose pas de question, prend Agilent », ou encore « de toute façon il n'y a que Gossen ». Chaque utilisateur a ses préférences, qui sont le plus souvent justifiées mais découlent de sa propre expérience. On évitera cependant d'en faire des vérités universelles pouvant s'appliquer invariablement à tous.

2. REMARQUES GÉNÉRALES

Une fois n'est pas coutume et la remarque la plus générale possible, s'appliquant à bien des choses, est celle qui guide bien des achats : vous payez pour ce que vous achetez ! Comprenez par là qu'en optant pour un multimètre à 20€ vous ne pouvez espérer disposer d'un appareil sûr, précis, solide et ergonomique. Pour 20€, vous aurez un multimètre qui vaut 20€ et rien de plus. Grossièrement, on peut même étendre le résonnement en définissant des limites arbitraires relativement fiables : avec un budget de plus de 100€ on obtient souvent un bon produit et, en dessous de 50€, on est quasi-assuré d'avoir quelque chose de parfaitement inutile, voir dangereux. Mitigeons ces propos puisque, comme pour le vin, il reste possible théoriquement de trouver de très bon produits très peu chers. C'est une question de probabilité mais lorsqu'on cherche un produit qu'on souhaitera utiliser des dizaines d'années, on tente généralement de minimiser les risques.

En dehors du prix de l'appareil, il existe un élément qui permet de reconnaître à coup sûr un multimètre d'un modèle à éviter comme la peste : la présence d'un testeur de transistor. Prenant généralement la forme d'un emplacement à 6 connecteurs, celui-ci permet de tester les transistors PNP et NPN et d'en mesurer le gain. Disons-le clairement, ceci ne sert strictement à rien. Personne, ou une fraction minuscule des utilisateurs, n'a besoin de ce type de fonctionnalités sous cette forme (il est relativement simple de tester un transistor à l'aide de multimètres). On notera d'ailleurs que les appareils haut de gamme ou très haut de gamme, intégrant des fonctionnalités pourtant exotiques, ne possèdent pas ce type de choses. Et ce, tout simplement, parce que ce n'est pas une vraie fonctionnalité mais un simple argument de vente douteux. Mais nous pouvons retourner cela à notre avantage en partant simplement du principe que si le multimètre dispose de cette emplacement, il est probable qu'il s'agisse d'un modèle à éviter.

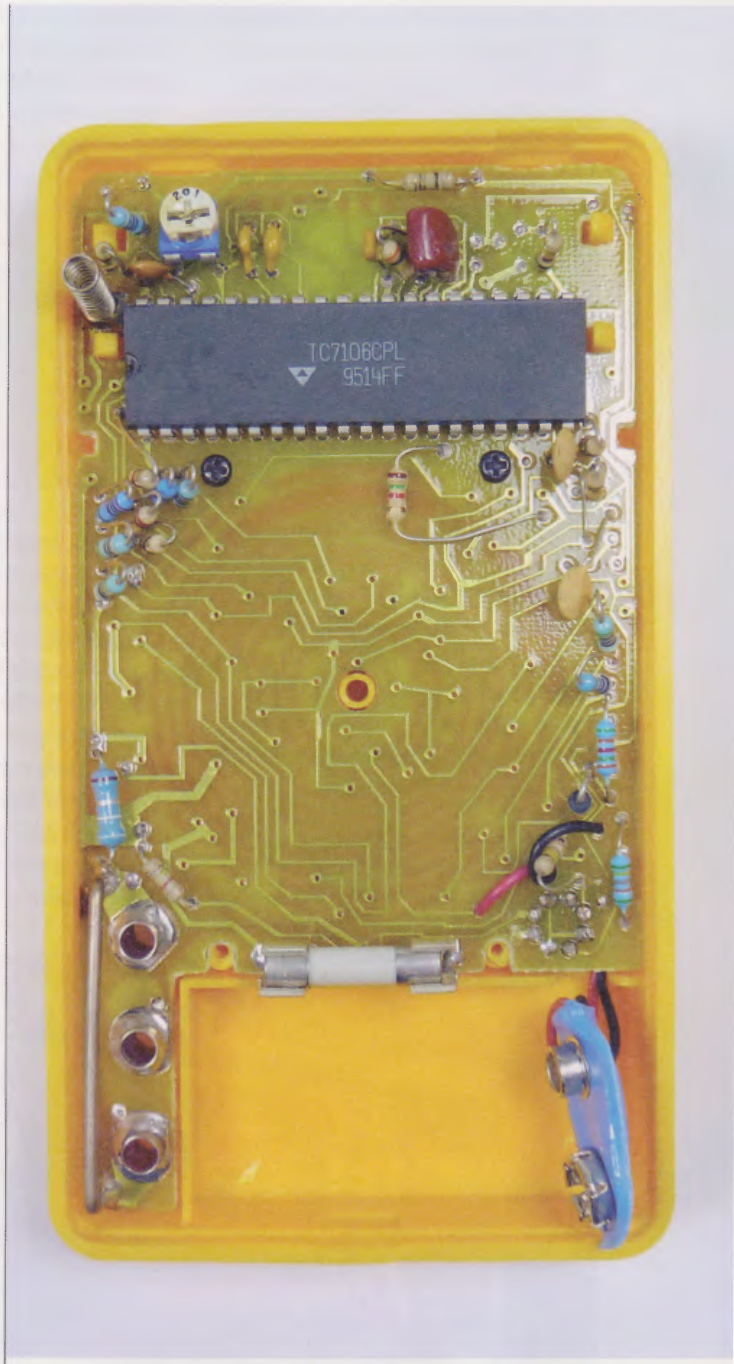
Actuellement, il est courant d'acheter son matériel en ligne

et dans le cas de l'achat qui nous intéresse ici, c'est un problème plus qu'un avantage. Le multimètre est un appareil « de terrain ». Il doit donc être conçu de manière à servir le plus longtemps possible et ce quelque soit votre « brutalité » et votre environnement de travail. Ce genre de qualité, n'importe qui peut l'évaluer en ayant simplement l'appareil en main.

Un bon multimètre doit vous durer 20 ans ou plus (certains sont même garantis à vie). Ceci implique qu'il doit être constitué d'un boitier solide, renforcé de préférence avec une coque souple permettant d'amortir les chocs. Toujours sur l'aspect générale, le boitier de préférence disposera de trappe permettant l'accès aux fusibles et aux piles, sans avoir à démonter l'ensemble. Très souvent dans les appareils bas de gamme c'est tout l'arrière de l'appareil qui s'ouvre et expose son contenu. Avec ce type de produits, le boitier est fermé par de simples vis s'enfonçant dans le plastique. Un bon produit utilise généralement des emplacements métallisés pour les vis, assurant ainsi la fermeture correcte du boitier même après des dizaines de vissage/dévissage.

Davantage sur le plan ergonomique, un bon multimètre dispose généralement d'un grand écran de manière à permettre une lecture claire des valeurs. Dans le même ordre d'idées, l'angle de lecture doit être optimal. C'est un des principaux défauts des écrans à cristaux liquides monochromes, ils n'offrent pas le même contraste en fonction de l'angle de vue. Avec un appareil bas de gamme l'écran sera, bien entendu, le moins cher possible et disposera de caractéristiques assez pauvres. Il n'y a rien de pire que de mesurer une tension sur un circuit, se tourner vers le multimètre et de constater que l'écran est illisible car mal orienté. La plupart des multimètres ont des écrans éclairés. Là encore la qualité impacte directement l'utilisation du produit car dans les modèles bas de gamme, ceci se limite à une ou deux leds sur un côté de l'écran, ce qui n'améliore en rien la lisibilité, au contraire. Profitons-en pour parler de l'alimentation. Un multimètre annonçant une durée de vie de la pile de moins de 300 heures est un mauvais multimètre, mal conçu. Vous n'êtes pas censés changer la pile de votre appareil tous les 2 mois.

En plus de tout cela, il y a des considérations ergonomiques à prendre en compte. Deux sont particulièrement importantes : l'auto-range et l'auto-hold.



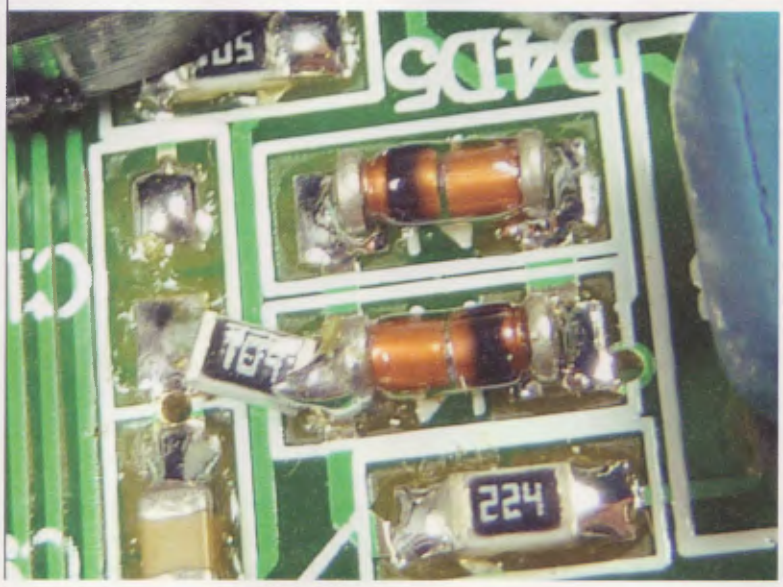
Le circuit exposé d'un multimètre à 15€. Tout est construit autour d'un circuit logique unique et constitué de composants « traversants » (through-hole). Notez l'absence totale de protection en dehors du fusible.

L'auto-range est une fonctionnalité qui doit être présente sur le multimètre. Il s'agit de la possibilité qu'a l'appareil à déterminer seul et de manière automatique dans quelle gamme de valeur (calibre) la mesure doit s'effectuer. Sur des vieux multimètres, il était d'usage de choisir la plage de mesure manuellement. Si, par exemple, vous souhaitiez mesurer une tension d'environ 15 volts, il fallait tourner le sélecteur dans la position adéquate de manière à choisir une mesure de tension de 2 à 20 volts. En choisissant la gamme 0 à 2 volts, l'appareil étaient incapable de mesurer la tension car tout simplement au delà des valeurs tolérées. Pour mesurer une tension inconnue il était donc d'usage de commencer par la gamme la plus importante pour ensuite descendre jusqu'à pouvoir procéder à la mesure correctement.

Cette technologie est complètement dépassée. Un multimètre digne de ce nom aujourd'hui intègre l'auto-range et doit être en mesure de définir tout seul la gamme de valeurs adaptée pour ainsi permettre à l'utilisateur de se concentrer sur le problème et non sur l'utilisation de l'appareil. Bien entendu, si nécessaire, cette sélection de gamme automatique doit pouvoir être désactivée mais l'élément clé ici se résume ainsi : un multimètre sans auto-range est un appareil d'un autre âge.

L'auto-hold est une autre fonctionnalité ergonomique permettant également à l'utilisateur de se concentrer uniquement sur ce qu'il fait. L'idée est de procéder à une mesure et de conserver cette mesure à l'écran. En effet, si vous devez mesurer une tension, un courant ou une résistance *in-situ*, vous manipulez généralement les pointes des cordons de test et concentrez votre attention sur leur position pour procéder à la mesure et non sur l'écran du multimètre. Certains modèles d'appareil disposent d'un bouton « hold » permettant de conserver à l'écran la valeur mesurée. Ce bouton est absolument inutile dans une telle situation puisque vos deux mains sont occupées, vous ne pouvez simplement pas appuyer sur le bouton. C'est là tout l'intérêt de la fonctionnalité d'auto-hold : on prépare le multimètre pour la mesure, on place les pointes, le multimètre procède à la mesure qu'il signal généralement d'un bip et la valeur reste à l'écran pour que vous puissiez la lire.

Enfin, puisque nous parlons de sondes (cordons de mesure à pointe ou autre) il est important de noter que ces éléments sont tout aussi importants que le multimètre lui-même. Elles doivent être de bonne facture, avec une pointe fine pour des positionnements précis, confortables à la manipulation et disposant d'un câble souple de préférence en silicone. Évitez absolument les cordons à bas prix, de provenance douteuse et composés de plastique rigide prompt à s'user au point de se dénuder et découvrir l'âme métallique. Envisagez cela exactement de la même manière qu'un instrument d'écriture. C'est la partie du multimètre que vous tiendrez en mains et vous n'avez vraiment pas envie de ressentir une gène à chaque utilisation.



Voici typiquement ce que vous risquez de trouver en ouvrant un multimètre « économique » chinois. Cette résistance est-elle seulement censée se trouver là ? Est-ce une erreur de fabrication ? Une correction de dernière minute ? Quoi qu'il en soit ceci n'est pas digne d'un outil de mesure dans lequel on est censé placer sa confiance et sa sécurité.

3. SÉCURITÉ

Une chose capitale à retenir : un multimètre est un appareil qui se place entre le courant et vous. Il est donc un outil mais également la seule protection contre des courants et des tensions potentiellement mortels. Soyons tout à fait clair sur ce point : une différence de prix de 50 ou 100 euros n'est absolument rien comparé au fait de risquer votre vie en mesurant, par exemple, une tension sur le réseau électrique domestique (prise de courant).

Il existe un ensemble de normes destiné à limiter les risques pour l'utilisateur d'appareils de mesures électriques. Dès lors qu'on utilise du courant, il s'agit potentiellement d'une situation dangereuse. Les risques sont principalement l'électrisation, c'est à dire le contacte du corps avec l'énergie électrique (le mot « électrocution » est généralement mal utilisé, puisqu'il décrit littéralement la mort par électrisation) et la projection de matière en fusion (provoquée par un arc électrique). La norme décrie des catégories permettant de choisir son équipement de mesure en fonction de l'utilisation qui en sera faite :

- CAT IV : distribution directe EDF
- CAT III : distribution industrielle
- CAT II : distribution domestique
- CAT I : électronique protégée

Le choix d'un produit respectant les normes de sécurités définies par catégorie dépend de votre utilisation. Le plus souvent CAT II sera suffisant pour un usage tel que celui fait des différents domaines couverts par ce magazine. Généralement le choix d'un multimètre de qualité s'accompagne cependant, dans le cas d'un produit de marque, de normes de sécurité de CAT III. Et c'est là précisément que le bas blesse avec les appareils chinois bas de gamme, vendus à petit prix un peu partout : il y a bien un marquage CAT II ou CAT III sur le produit mais il ne correspond parfois à rien du tout ! Exactement comme d'autres produits affichent un logo « CE » mais ne respectent pas les normes l'autorisant à être commercialisé en Europe. La même remarque s'applique aux produits présentant le logo d'un constructeur mais n'étant absolument pas de sa fabrication. On parle ici clairement de contrefaçon mais avec des conséquences bien plus importantes que celle d'un faux sac Chanel sortant d'une usine malaisienne...



Cette grosse barre est une résistance de shunt... placée à proximité des connecteurs du multimètre qui sont soudés directement sur le circuit. Remarquez les soudures ignobles et l'aspect sensiblement différent du connecteur central. Le constructeur aurait-il utilisé ce qui lui passait sous la mains ? Ne riez pas, c'est un processus de fabrication courant pour ce type de produit.

C'est un problème qui va bien au delà des appareils de mesure comme les multimètres. Les normes de sécurités imposant un certain nombre de procédés techniques garantissant la sécurité de l'utilisateur ne sont tout bonnement pas respectées par un certain nombre de fabricants utilisant des canaux de distributions pouvant se soustraire aux contrôles. Ceci va de l'absence de mise à la terre dans un appareil électroménager au défaut d'isolation d'un bloc d'alimentation en passant, bien sûr, par l'absence de protection sur un multimètre.

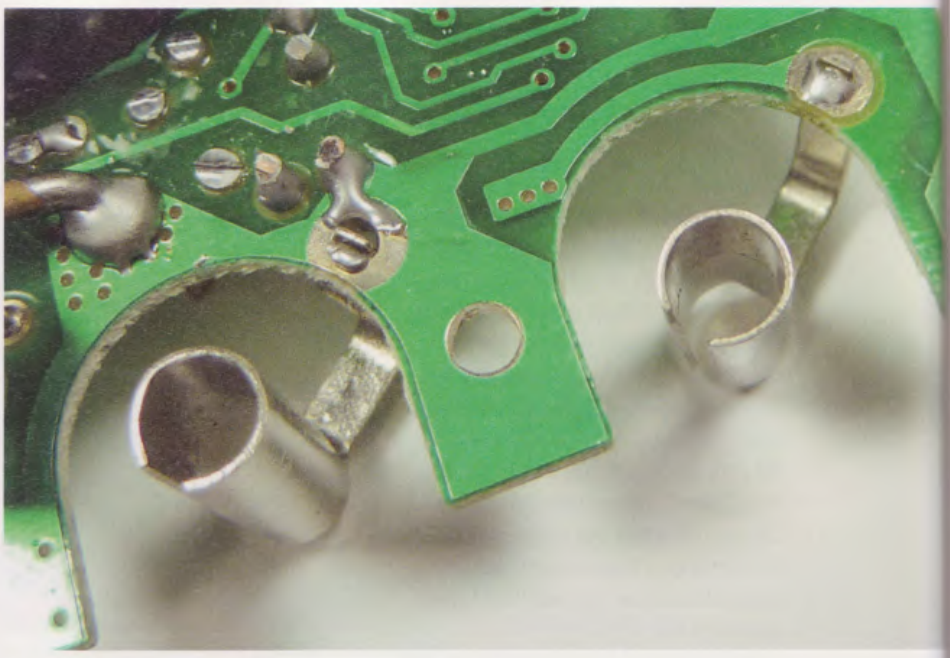
C'est là une raison importante de choisir un équipement de marque, vendu par un distributeur ayant pignon sur rue. Mais en dehors de ce simple critère, il existe quelques points qui permettent de mettre en évidence un manque

évident de protection. L'une d'elle concerne les connecteurs des cordons de test. Il doivent normalement être au nombre de quatre :

- « A » pour la mesure de courant important (généralement avec un maximum autour de 20A),
- « mA/µA » pour la mesure de courant « standard »,
- « V » pour la mesure de tension, de résistance et de fréquence,
- et « COM » pour « commun » associé au cordon noir faisant office de référence et connecté à la masse du circuit mesuré.

Le point important concerne l'existence d'un connecteur différent pour la mesure de courant et de tension. Aucun multimètre digne de ce nom (en dehors des modèles de poche) ne propose un seul et même connecteur pour ces deux mesures. Pourquoi ? Parce que si vous mesurez un courant avec le multimètre réglé sur une mesure de tension, ce courant traversera le multimètre et détruira ses composants. En l'absence de protection et du passage d'un courant important, il peut en découler une destruction du matériel et peut-être de ce qui se trouve à proximité, comme votre main par exemple.

A propos des protections justement, le multimètre DOIT intégrer un fusible à la fois pour la mesure de faible courant (mA/µA) et de courant important. Bon nombre de multimètres bas de gamme offre ce type de protection uniquement dans en premier cas. Ainsi ces constructeurs partent du principe qu'il est peu probable que vous utilisiez l'appareil pour mesurer un courant plus important que celui pour lequel il est conçu. Pourtant, dans la vraie vie, cette probabilité existe et vous êtes en droit d'attendre du produit qu'il vous protège, même contre vos propres erreurs.



Avec ce modèle, le connecteur COM à gauche ne tient presque à rien tant la soudure est bâclée et celui de droite est celui permettant la mesure tension/résistance. Remarquez la faible distance qui sépare les pistes à certains endroits permettant la formation d'arcs électriques avec une tension élevée.

La protection elle-même doit également être sûre. Le type de protection contre le dépassement d'une certaine intensité de courant (découlant souvent d'un court-circuit) prend généralement la forme d'un fusible. Celui-ci est conçu pour un certain courant et si l'intensité est dépassée, le fusible est détruit ouvrant le circuit et arrêtant donc la circulation du courant. Mais tous les fusibles ne se valent pas. Ainsi le modèle courant constitué d'un tube en verre contenant un filament n'est pas suffisamment sûr pour un appareil de mesure comme un multimètre. La raison est simple : le passage d'un courant important peut vaporiser le filament et l'augmentation de pression ainsi provoquer peut faire littéralement exploser le fusible et le boitier dans lequel il se trouve. Pour éviter ce type de comportement, les multimètres utilisent normalement un fusible HRC (*High Rupture Capacity*) ressemblant à ceux des tableaux électriques.

Enfin précisons que les mécanismes de protection comme les fusibles ne sont pas censés être soudés dans le multimètre mais facilement accessibles et isolés du reste de l'appareil. Un fusible doit être présent pour chaque connecteur de mesure de courant (mA/µA et A) mais ce n'est pas tout. D'un point de vue électronique, le temps de rupture d'un fusible n'est pas suffisamment court pour une complète protection. Beaucoup d'énergie peut

entrer dans l'appareil avant la rupture et d'autres protections sont alors normalement mises en place sous la forme de ponts de diodes afin de laisser au fusible le temps de faire son travail. Le connecteur de mesure de tension/résistance est normalement également protégé via un savant mélange de résistances dites *wirewound* de thermistances (PTC) et de varistances (MOV).

Gardez à l'esprit qu'un bon multimètre est avant tout une protection entre vous et le courant électrique. Le fait que le matériel devienne inutilisable après un problème est hors de propos dès lors qu'il vous sauve la vie. Les 100€, 300€ ou 800€ que vous aurez investi dans votre multimètre valent de toute façon moins que la main dans laquelle il se trouve.

Personnellement, il ne me viendrait pas à l'idée d'utiliser un multimètre à 15€ ou 20€ tel qu'on en trouve dans les grandes surfaces et les magasins de bricolage, pour mesurer la tension sur une prise murale. Même si celui-ci affiche des caractéristiques rassurantes, sa conception même le rentre, à mon sens, dangereux (cf photos).

Tout n'est pas à jeter avec les multimètres bas de gamme. Ici, un effort a été fait sur les connecteurs et les câbles livrés et le conducteur n'est pas exposé quelque soit l'état de la connexion. Bon point ! Malheureusement, il ne s'agit pas simplement de mesurer les bons et les mauvais points d'un appareil et de faire des concessions sur la sécurité. Ici, il s'agit du même produit que celui intégrant le fusible en verre et présentant des soudures douteuses.

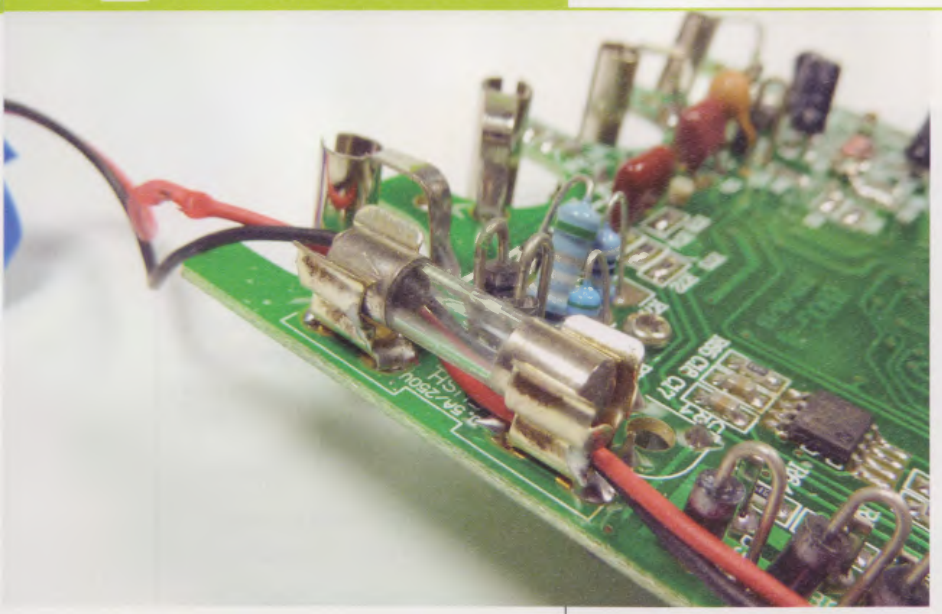
N'oubliez pas que le courant domestique, même si on est accoutumé à sa présence, est quelque chose de très dangereux. Pensez à ce radiateur électrique de 1000 watts que vous branchez en hiver, votre aspirateur, votre lave-ligne et imaginez cette énergie instantanément conduite dans un multimètre d'un volume de la taille d'un paquet de café. Mieux vaut que le dispositif soit correctement conçu et ce avec la protection de l'utilisateur à l'esprit en premier lieu (et non la marge bénéficiaire potentielle).

4. FONCTIONNALITÉS ET CARACTÉRISTIQUES

A présent que nous avons fait un tour d'horizon des éléments qualitatifs généraux et de la notion très importante de sécurité, il est temps de passer aux points précis qui font d'un multimètre un bon multimètre. Un tel appareil doit pouvoir mesurer la tension, le courant et la résistance. C'est le strict minimum. Voyons chacun de ces éléments individuellement.

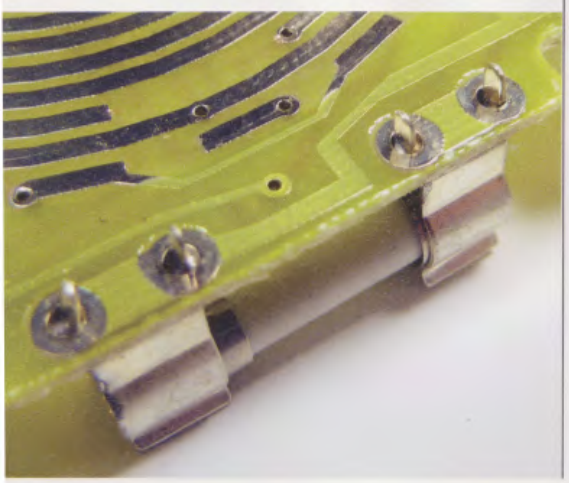
Le multimètre doit pouvoir mesurer la tension d'un courant continu (DC) et alternatif (AC) dans une plage allant de 200 mV à 1000 V. Pour des travaux d'électronique, un multimètre n'étant pas capable de mesurer une tension en dessous d'un volts est parfaitement inutile. Ce type de dispositif est destiné aux électriciens pas aux électroniciens. Si vous comptez mesurer des tensions avec des courants alternatifs, il sera important de choisir un appareil RMS ou TRMS capable de mesurer une tension « efficace vraie ». Sans entrer dans le détail, un multimètre non RMS mesurera une tension d'un signal sinusoïdal qu'il estimera être parfait et vous affichera donc une valeur qu'on peut qualifier d'approximative. Dans la réalité, un signal électrique n'est presque jamais parfait,





On ne peut pas toujours démonter un appareil avant achat (sic), les mauvaises surprises en misant sur du « sans marque » arrivent donc après avoir déboursé ses deniers. Ici le câble d'alimentation pour la pile 9V passe littéralement dans le support du fusible 250V/800mA ! Remarquez aussi que le câble rouge est dans un sale état puisque celui-ci a été écrasé dans le boitier.

Gros plan sur la soudure d'un support de fusible d'un multimètre à quelques euros. Non, vous ne rêvez pas, il n'y a presque pas d'étain de ce côté du circuit et la piste provenant du connecteur « 1000 VDC max » passe à une distance défiant toutes les normes de sécurité (clearance). Ce type de produit ne devrait même pas être commercialisé en France...



à commencer par le courant alternatif domestique 230V/50Hz qui est loin d'être un parfait sinus.

La mesure du courant en électronique nécessite de pouvoir travailler en dessous du milliampère. Vous n'aurez peut-être pas besoin immédiatement de mesurer des courants de quelques microampères (µA) mais arrivera forcément le moment où vous n'aurez pas le choix. Les fonctionnalités d'économie d'énergie, de plus en plus utilisées, impactent directement la mesure du courant et il n'est pas rare aujourd'hui de voir un microcontrôleur consommer seulement quelques microampères en veille. Ce comportement va se généraliser et devenir la norme et vous n'avez pas envie de vous retrouver démuni lorsque cela arrivera. Investissez sur le long terme, optez pour un multimètre capable de mesurer ce type de faibles courants. De l'autre côté de la gamme, mesurer des courants de 10A ou 20A est largement suffisant pour la quasi-totalité des usages. Si vous savez que vous devez mesurer des courants supérieurs à ceux-ci, il est peu probable que cet article vous apprenne quoi que ce soit.

La résistance, enfin, doit pouvoir être mesurée entre environ 100 milliohms et 20 mégaohms avec une précision dépendant, bien entendu, de la gamme choisie (exemple, précision de 0,1 ohm dans la gamme des 500 ohms, 1 ohm dans la gamme des 6 Kohms, etc). Ceci devrait être suffisant pour la plupart des usages. Précisons ici que le multimètre lui-même présente également une résistance interne (c'est le principe même de son fonctionnement). Un multimètre « normal » doit avoir une résistance interne de 10 Mohm ou plus. La plupart des modèles bas de gamme possèdent une résistance interne de 5 Mohm ou pire, de 1 ou 2 Mohms. Ceci peut poser de sérieux problèmes de mesure avec des faibles courants puisque la résistance, trop faible, du multimètre qui fait office de diviseur de tension va alors complètement fausser les résultats. Notez cependant que la mesure de tensions avec de très faibles courants nécessite des sondes particulières. Le cas typique étant celui des compteurs Geiger-Müller.

Accompagnant ces trois types de mesures, on trouve également le test de continuité permettant de vérifier l'existence d'une conduction entre deux points d'un circuit. En cas de contact, le multimètre émet alors un signal sonore. La qualité de ce test dépend à la fois des sondes elles-mêmes et du multimètre, ou plus exactement sa rapidité à signaler la conduction. Là malheureusement, il s'avère que certains modèles haut de gamme peuvent être relativement lents (1/2 seconde) dans le test de continuité. Il n'y a pas d'autres

solutions que de lire méticuleusement les spécifications techniques ou de partir à la recherche de critiques concernant un modèle spécifique sur le Web.

D'autres fonctionnalités de mesure peuvent également être présentes sur un multimètre :

- le test et la mesure de diodes : permet non seulement de tester le blocage du courant en fonction de la polarité mais également la mesure de la chute de tension à ses bornes,
- la mesure de la capacité, autrement dit des condensateurs, peut être très utile en particulier si le multimètre est capable de mesurer de petites capacités de l'ordre du nano farad. Ceci cependant ne remplacera pas RLC mètre (ou *LCR meter*) dédié spécifiquement à la mesure et au test de condensateurs et de bobines, très utile pour les opérations de réparation,
- la mesure de fréquences : ceci peut être tantôt utile mais si vos besoins s'orientent réellement vers ce type de mesures, il vous faudra de toutes façons un oscilloscope et la fonctionnalité sera alors redondante entre les deux appareils,
- la mesure de température. Là, contrairement à ce qu'on peut penser, ceci peut réellement devenir intéressant. Dès lors, par exemple, qu'on touche à la régulation de courant et aux transistors de puissance, la température devient un point important sinon critique. Certains modèles de multimètres sont livrés avec une sonde de température -40°C à +260°C,

Enfin, en termes de fonctionnalités logicielles, on apportera une attention particulière à :

- la vitesse de rafraîchissement de l'affichage, autour de 3 ou 4 par seconde,
- la présence d'un bouton permettant une mesure relative servant à la fois à compenser les écarts dus à la présence des sondes mais également à mesurer des variations de résistance, tension ou courant,
- la présence de fonction de collecte de données et à l'affichage de valeurs moyennes, minimums et maximums qui peuvent s'avérer utiles pour certains travaux,
- la vitesse de choix automatique de la gamme de mesure. Un test simple consiste à placer le multimètre en position de mesure de résistance et de mettre les sondes à pointe en contact. Si le multimètre met plus d'une seconde à choisir la gamme de mesure, il aura toutes les chances de rapidement vous pousser à bout si vous voulez rapidement régler un problème.

4.1 Digits et précision

Finissons en parlant d'un point important souvent mis en avant de manière trompeuse dans les spécifications de matériel bas de gamme : la précision. Mais avant cela décryptons un point figurant souvent dans ces spécifications. Celui-ci concerne l'affichage sur l'écran d'un multimètre numérique et évoque « 3 1/2 *digit* » ou encore « 4 1/2 *digit* ». Comment est-il possible d'avoir un demi *digit* ou



Un multimètre bas de gamme à environ 40 euros. Sélection de gamme manuelle, absence de fusible sur le 20A et une profusion de fonctionnalités (transistor, température, tension, courant, fréquence, capacité, etc). Il y a quelque chose de pas très normal ici. Impression clairement vérifiée après ouverture de l'appareil.

chiffre ? Ça ne l'est pas. Ceci fait en réalité référence au nombre de segments permettant de former un chiffre sur un écran LCD ou led.

Un chiffre sur ce type d'affichage se compose de 7 petites barres permettant, si elles sont toutes actives de former un « 8 ». On parle généralement d'afficheurs 7 segments. En ajoutant le point comme un élément d'affichage, on en arrive à obtenir 8 éléments d'affichage par position décimale. Or un afficheur LCD ou led est généralement composé de 4 ou 5 de ces « blocs » mais celui le plus à gauche n'est pas toujours complet. Il ne permet d'afficher que « 1 » et »-1 », avec ou sans le point. Le « 1 » étant composé de deux petits segments, ceci nous donne, pour cette position, uniquement 4 éléments. Le chiffre le plus à gauche n'utilise donc que la moitié des segments ou éléments utiles pour un chiffre complet. Un afficheur 3 1/2 *digit* peut ainsi afficher une valeur de -1999 à 1999 et un afficheur 4 1/2 *digit* de -19999 à 19999.

On parle également de compte ou de *counts* pour définir cet affichage. Ceci fait référence directement aux valeurs qu'il est possible d'afficher et plus exactement à la valeur maximum + 1. Dans le cas d'un afficheur 3 1/2 digits ceci nous donne 1999 + 1 soit 2000 *counts*, pour un 4 1/2 digit ceci sera donc 20000 *counts*.

La précision la plus importante est celle de la mesure d'une tension en courant continu (DC). Ainsi, il est entendu que, pour une utilisation courante professionnelle comme amateur, une précision impactant le plus petit chiffre d'une valeur (celui le plus à droite) est acceptable.

Cependant, la précision est généralement indiquée sous la forme d'un pourcentage. Il faut donc faire un peu de calcul car il existe un lien entre le nombre de *counts*, le chiffre le plus à droite et le pourcentage minimum attendu. La formule est la suivante : 10 / nombre de *counts* * 100. Ainsi pour un multimètre 3 1/2 *digit* de 2000 *counts* nous arrivons à 10/2000*100 = 0,5 %. Mais pour un multimètre 4 1/2 *digit*, cela nous fait 20000 *counts* et donc 10/20000*100 = 0,05 %.

Une fois la formule et le principe assimilés, il est relativement facile de détecter les multimètres bas de gamme à éviter. Il suffit de pointer son navigateur sur un moteur de recherche pour trouver rapidement des multimètres à moins de 20 euros sur des sites de ventes discount chinois. Ceux-ci affichent des spécifications comme :

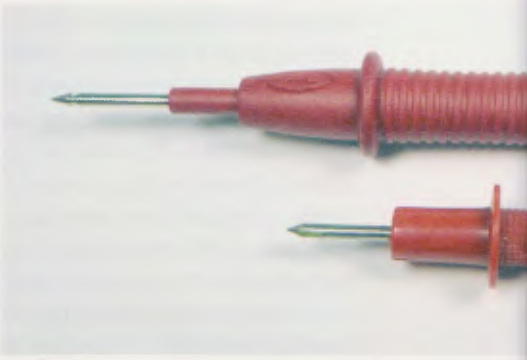
- « Max.affichage de chiffres: 3999 »
- « Tension continue:Précision: +/- (0.8%) »

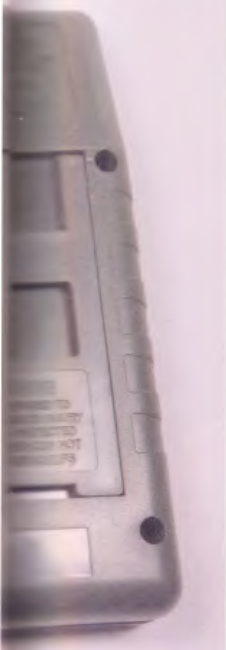
Si nous faisons notre petit calcul 10/3999*100 = 0,25 %. Pour pouvoir faire des mesures acceptables selon nos critères, nous devrions avoir AU MINIMUM une précision de 0,25 % or le produit affiche 0,8 % ce qui signifie clairement que bien plus que le dernier chiffre est dans la plage d'imprécision. Là une seule réaction s'impose : cliquer sur la petite croix sur l'onglet du navigateur et chercher un vendeur/produit plus sérieux.



CONCLUSION

En résumé, en partant du principe que vous pouvez avoir confiance dans les spécifications du constructeur (ce qui élimine d'office bon nombre de modèles bon marché au design cloné sur des modèles de marque), il est relativement facile de trouver de bons multimètres pour vos activités d'électronique. Nous avons décrit ici les points clés importants et vous constaterez par vous-même qu'en faisant le tri dans les constructeurs, on en revient souvent autour des mêmes





Le multimètre EC-890G ouvert. Derrière la façade qui peut paraître attractive : connecteurs bas de gamme fragile, un seul fusible, absence presque totale de protection (GDT, MOVs, etc), aucune isolation correcte, pas de trappe pour le fusible ou la pile, soudures douteuses...

marques : Fluke, Gossen, Extech, Amprobe, Protek, BK Precision, Hioki, Agilent... et j'en oublie certainement.

Dans vos recherches, ne perdez pas de vu l'usage que vous comptez avoir de votre multimètre. Les fabricants, même réputés, proposent des gammes très diversifiées avec tantôt des produits avoisinant les 30 euros. Ceux-ci sont donc clairement sous la barre des 50 euros fatidiques. N'en concluez pas pour autant que le constructeur vend des produits bas de gammes.

Il serait déraisonnable d'évaluer un multimètre décrit clairement pour un « usage résidentiel » sur des critères exigés pour un matériel semi-professionnel. Cette valeur de 50 euros s'applique uniquement pour un usage d'électronicien passionné ou de professionnel et si vous « remontez » la gamme du constructeur en question, il est fort probable que vous arriviez sur un matériel qui vous convient mieux, mais dans une autre tranche de prix...

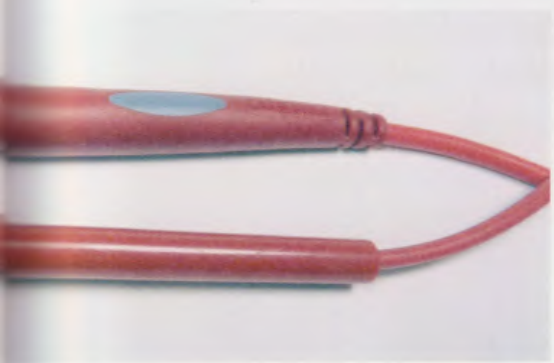
Une dernière petite chose qui impactera forcément votre budget. Il est communément admit qu'il est de bon ton d'avoir à disposition non pas un multimètre mais deux. La raison est simple et pragmatique : c'est le moyen permettant de mesurer simultanément le courant et la tension pour un circuit donné. Si vous vous lancez à corps perdu dans le domaine fantastique qu'est l'électronique, il ne fait aucun doute qu'il vous faudra rapidement un second multimètre. Ceci vous permettra également de vérifier/confirmer vos mesures avec des montages ou circuits sensibles ainsi que vous éviter de vous trouver dans la situation frustrante nommée « j'ai encore oublié d'acheter des piles et j'ai 3 projets en cours »... **DB**



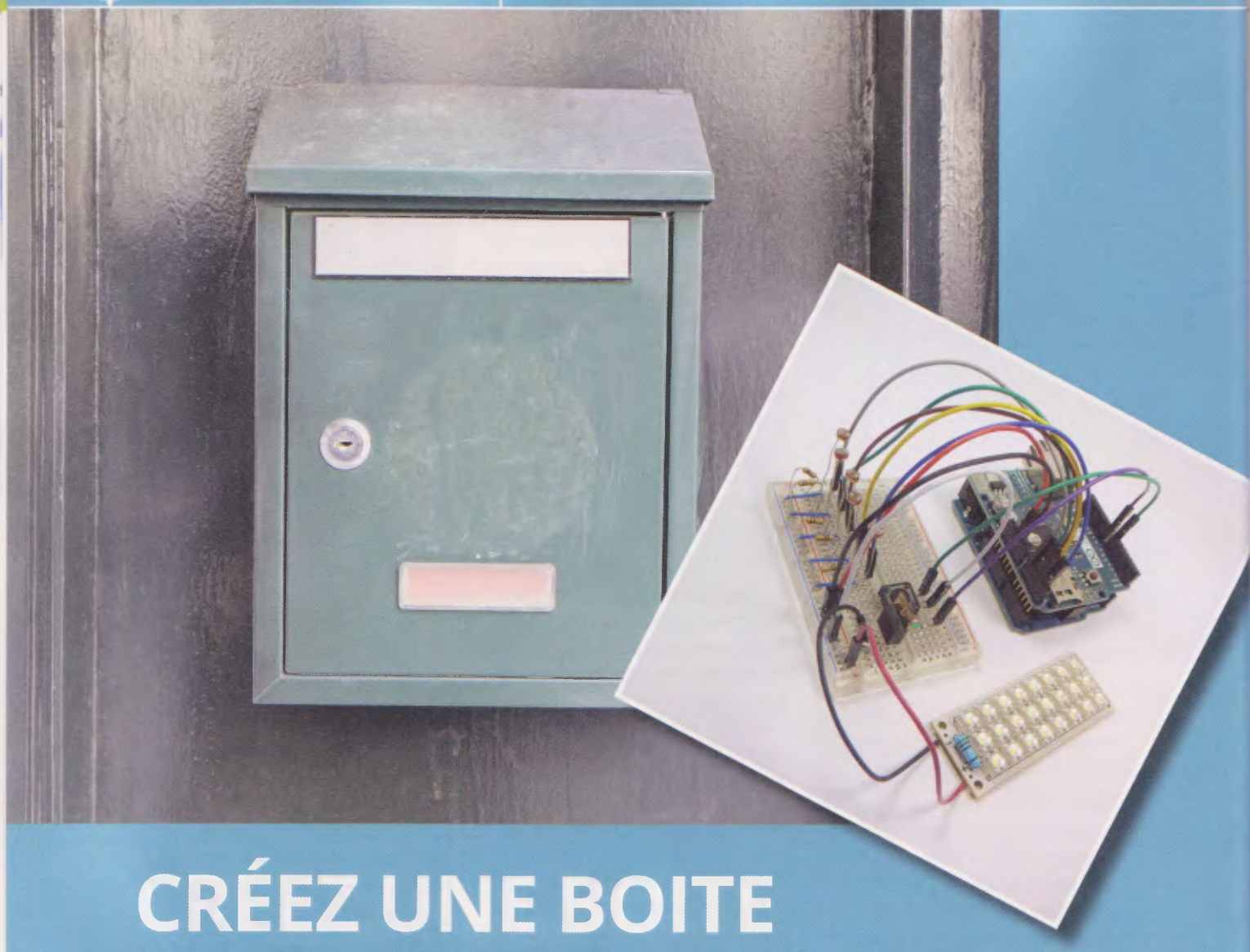
Le fameux emplacement permettant le test de transistors. Souvent présent sur les appareils bas de gamme mais systématiquement absent sur les produits haut de gamme. Quelqu'un se sert-il réellement de ce genre de « fonctionnalité supplémentaire » dans la vraie vie ?

Emplacement de vis d'un multimètre bas de gamme. Ceci couplé à la nécessité d'ouvrir tout l'appareil pour changer la pile ou le fusible conduit généralement toujours au même résultat : l'impossibilité de refermer correctement le boitier. Vous ne trouverez normalement jamais ce type de solution sur un produit destiné à être utilisable 20 ans.





Les sondes ou « probes » en anglais sont également très importantes. Celle du bas accompagne le multimètre à quelque euros : un simple bout de plastique avec une pointe métallique. Vous imaginez tenir ce genre de choses pour mesurer des tensions de 750V ou 1000V pour lesquelles, soi-disant, l'appareil est conçu ? La sonde du haut est de meilleure facture et marquée CAT III 1000V. Cela est acceptable mais loin de pouvoir être qualifiée de haut de gamme.

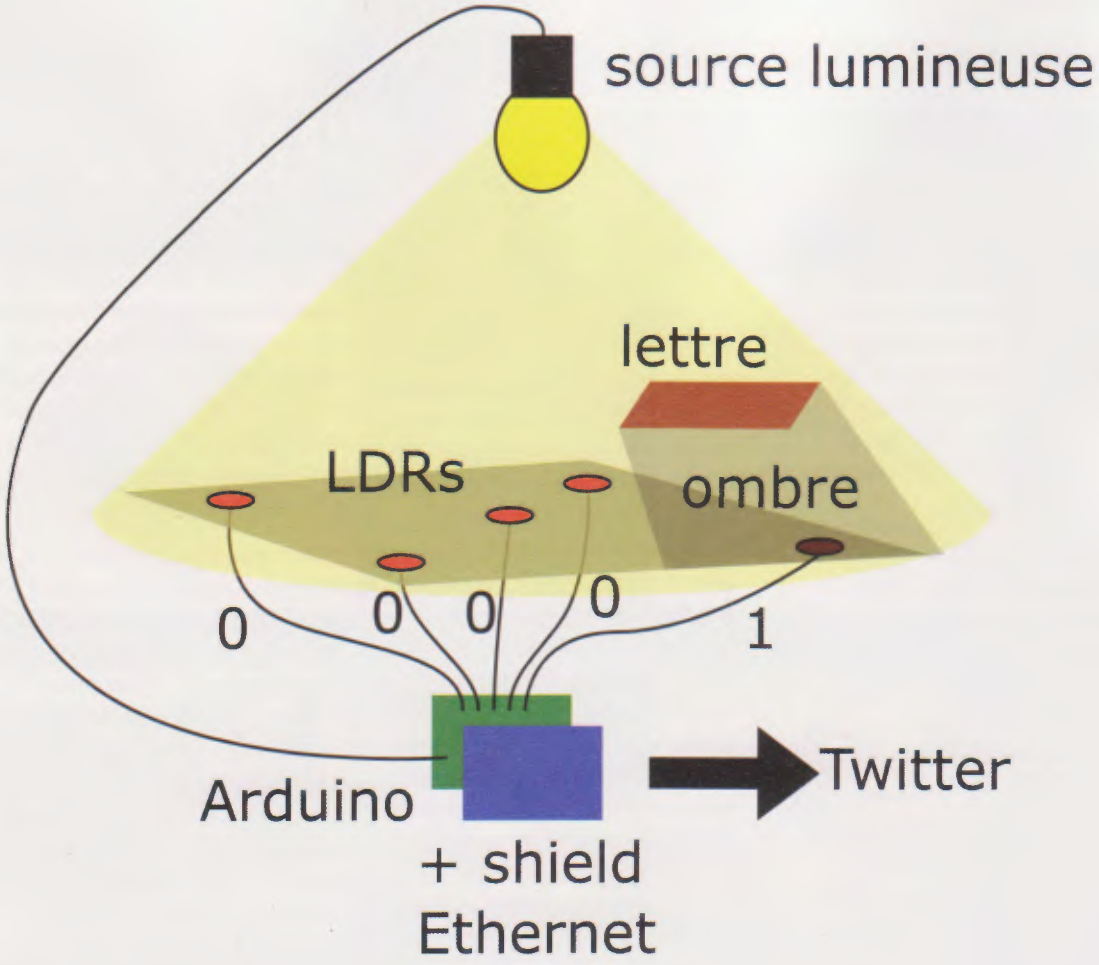


CRÉEZ UNE BOITE À LETTRES QUI TWEETE

D. B.

En électronique il y a plusieurs projets absolument incontournables tels la montre binaire, la machine qui s'éteint elle-même, l'éclairage à leds RVB contrôlable par tous les moyens possibles, la chatière qui reconnaît le chat mais ne laisse pas entrer celui du voisin ou encore la fameuse boite à lettres qui vous dit si vous avez du courrier. Voilà un bel exercice mêlant plusieurs disciplines, faisant appel à notre imagination et à notre créativité. En route pour cette réalisation remise au goût du jour à la sauce « réseaux sociaux ».

NIVEAU	TEMPS	BUDGET
	60 minutes	environ 60 € (Arduino inclus)



Non, nous ne sommes pas des gros feignants en pantoufles ne voulant pas même prendre la peine de décoller leur nez de l'écran pour vérifier le contenu de la boite à lettres ! Rien à voir. C'est beaucoup plus subtil et en rapport, attention, avec le besoin pathologique de ressentir mensuellement (voir hebdomadairement) un plaisir et une excitation quasi-juvénile normalement attendue vers la fin décembre. En d'autres termes, on veut tous que ce soit noël tous les jours. Que celui d'entre vous, amis lecteurs, qui ne ressent pas cela les jours suivants une commande et particulièrement lors du coup de sonnette du facteur me jette la première pierre (je lui jetterai les suivantes).

Si vous vous reconnaissez dans ces mots, ne vous sentez pas pour autant atteint d'un quelconque trouble. D'autres attendent ainsi des courriers du cœur, des cartes postales, des chèques ou leur avis d'imposition (?!).

Or voilà ! Dans la vie, il n'est pas toujours possible de faire le guet à la fenêtre, même si je dois avouer que me voir ainsi sur le pied de guerre, l'oreille dressée, les sens en éveil et entraîné à reconnaître le bruit du coulissement d'une portière de camionnette Colissimo, UPS ou DHL est, paraît-il, très amusant. Quel cauchemar alors de se dire qu'en notre absence, le colis ou le pli tant attendu est peut-être arrivé... ou pas. C'est insoutenable !

Nous en appelons donc à la déesse technologie pour exhausser nos vœux et faire cesser ces tourments. Comme à son habitude, elle répond à l'appel pour peu que l'on fasse preuve d'une pointe d'imagination et d'astuce. Elle est très gentille la déesse technologie...

LE PRINCIPE

Dans les très grandes lignes, une carte Arduino sera chargée de capter la présence d'objets dans la boîte à lettres et notifiera de cet état de fait via une connexion Internet. Deux principaux points constituent le défi : détecter les lettres/colis et notifier de la situation.

La notification via un shield Ethernet, reposera sur Twitter pour plusieurs raisons :

- Des clients Twitter existent pour presque toutes les plateformes, ordinateur+navigateur ou application mobile,
- il est possible, via une plateforme particulière, d'envoyer des tweets automatiquement,
- plusieurs bibliothèques Arduino permettent d'intégrer cette automatisation au sein d'un croquis.

La détection est un peu plus problématique et on peut aisément imaginer énormément de solutions. Par exemple :

- On peut surveiller la petite trappe permettant de glisser le courrier. Mais que se passe-t-il si le facteur ouvre la boîte à lettres pour y placer un petit colis (sans signature) ? Pire encore, si quelqu'un ou quelque chose manipule la trappe sans glisser de courrier ? Imaginez votre déception...
- On peut surveiller le volume de la boite à lettres avec un module radar à ultrason comme ceux permettant aux robots de détecter les obstacles. C'est un peu plus efficace que la solution précédente mais le placement judicieux du capteur est capital car une lettre, ce n'est pas bien épais.
- On peut placer des contacteurs mécaniques sur le fond de la boite à lettres. Il faut cependant ne pas oublier que celle-ci se trouve à l'extérieur, dans un environnement qui n'est pas contrôlé (c'est le principal problème du « dehors », le ménage n'est pas souvent bien fait). Poussière, vibrations, humidité, froid... sont autant de sources de problèmes qu'il faut prendre en compte en plus de celui de la détection effective de lettres pouvant peser quelques grammes seulement. Et je ne parle même pas de l'éventuelle brutalité postale...
- Laser, capteurs volumétriques, caméras infrarouges... deviennent des options mais se pose alors le problème de la gestion d'énergie et de l'investissement. Le but n'est ni d'avoir une boite à lettres surpuissante hors de prix, ni de créer un gouffre à énergie. Si la boite à lettres ou la consommation électrique générée sont supérieurs à la valeur des marchandises que vous attendez, cela n'a pas vraiment de sens.

Notre solution sera d'utiliser la lumière. Comme le montre le schémas ci-contre, nous allons parsemer le fond de la boîte à lettres de photorésistances qui permettront donc à l'Arduino de juger de la lumière présente dans la boite. Nous accompagnons ce dispositif d'un système d'éclairage interne. Le principe sera alors le suivant : on éclaire depuis le haut le contenu de la boite, si l'une des photorésistances ne captent pas correctement cette lumière, c'est qu'un objet fait obstacle.

Ce choix s'avère facile à protéger puisque ne disposant pas de partie mécanique. Seule la saleté peut alors impacter le fonctionnement mais nous pouvons corriger cela de manière logicielle en calibrant l'ensemble et en procédant à une mesure de référence « à vide ». Le dispositif peut également être parfaitement intégrer à la boite, soit en ajoutant un faux fond et un plafond, soit en insérant l'éclairage et les photorésistances dans le corps de la boite à lettres (le pistolet à colle est votre ami).

Enfin, il est possible d'étendre facilement l'ensemble pour plus de sensibilité. En principe, une seule photorésistance pourrait suffire mais si la lettre se place juste à côté et ne masque pas le capteur, elle ne sera pas détectée. Pour régler ce petit soucis, il suffit d'utiliser plusieurs photorésistances de manière à ce qu'elles soient forcément masquées par une enveloppe de taille minimum standard ou un petit colis. Ensuite, rien ne vous empêche d'en faire davantage et d'ajouter des photorésistances partout sur les côtés et de travailler avec des zones...

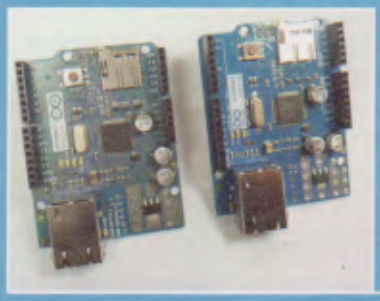
CE QU'IL VOUS FAUT



- Une carte Arduino : nous avons ici utilisé une Uno mais les autres cartes acceptant un *shield* Ethernet feront parfaitement l'affaire,



- 1 à 5 résistances 10 Kohms : elles seront du même nombre que les LDR,



- un shield Ethernet : ce circuit supplémentaire permet d'offrir à une Arduino une connectivité réseau filaire Ethernet,



- un MOSFET IRF520 : ce composant est un type de transistor à effet de champ généralement utilisé pour les circuits de puissance afin de contrôler avec un petit signal de grosses charges,



- une platine à essais et ses cordons de connexion pour le prototypage (mais il faudra sérieusement envisager un montage « en dur » pour l'installation définitive, sur plaque pastillée par exemple),



- quelques leds blanches ou autre système d'éclairage : le but est d'éclairer l'intérieur de la boite à lettres. Peu importe le système utilisé, son alimentation sera contrôlé par le MOSFET,

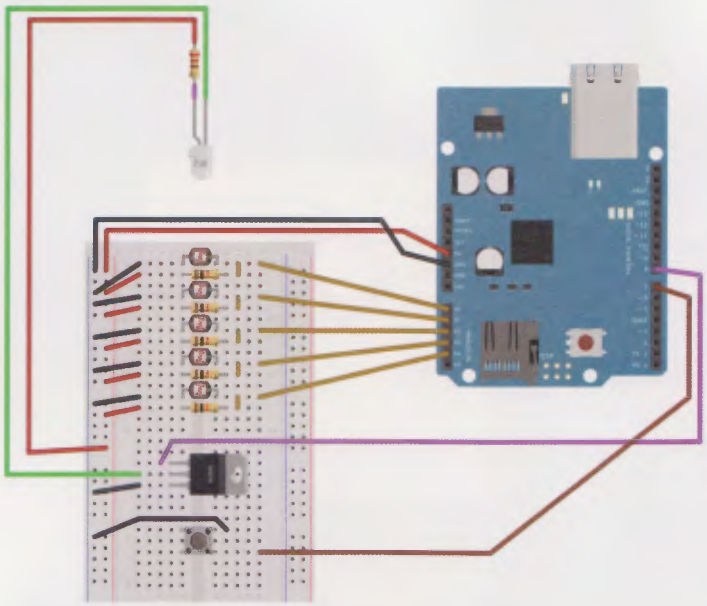


- 1 à 5 LDR (*Light-Dependent Resistor*) ou photorésistance: ces composants changent leurs résistances en fonction de la présence de lumière,



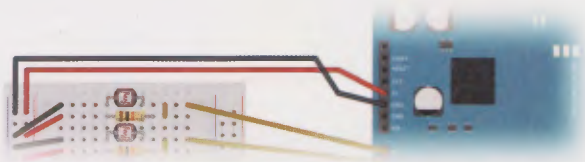
- un bouton poussoir : n'importe quel bouton ou contact fera l'affaire.

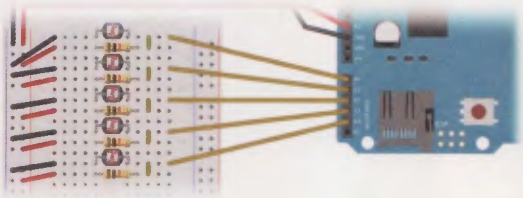
LE MONTAGE



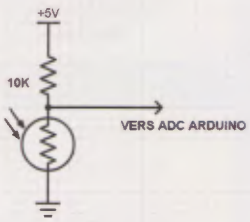
0 Nous utilisons ici un shield Ethernet afin de nous connecter à Internet et Twitter (via une application Appspot). Ce type de shield se place tout simplement sur votre carte Arduino, vous laissant accéder presque intégralement aux broches de cette dernière.

1 Avant toute chose, on connecte l'alimentation, fournie par l'Arduino à la platine à essais. Inutile ici d'ajouter un quelconque témoin lumineux pour la mise sous tension ou une éventuelle notification. Le montage à terme est destiné à être placé dans une boite étanche, très certainement sûre, sous ou contre la boite à lettres. Moins il y aura de trous dans cette boite mieux ce sera. Bien sûr, libre à vous de faire évoluer l'ensemble et d'ajouter plein de petites leds partout (on aime tous les leds).

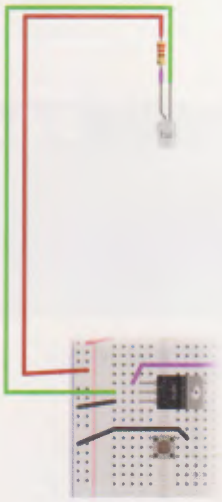




2 Les photorésistances sont connectées aux entrées analogiques de l'Arduino en utilisant un montage en pont diviseur de tension. Ainsi, l'Arduino va mesurer non pas une résistance mais une tension dépendante du rapport entre la résistance de 10 Kohms connectée à +5V et la photorésistance connectée à la masse selon le schémas ci-contre :



Dans ce montage, plus il y aura de lumière arrivant sur la photorésistance plus sa résistance sera faible et donc moins la tension mesurée par l'ADC de Arduino sera importante. En revanche, si la photorésistance est plongée dans l'obscurité, sa résistance sera maximale et la valeur mesurée sera importante.



3

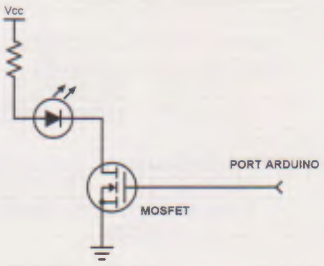
L'intérieur de la boite à lettres est censé être relativement obscure et nous n'avons pas l'intention de reposer sur la lumière fournie par une ouverture parce qu'une telle source risque de varier (météo, jour/nuit, passage devant l'ouverture, etc). Nous allons donc contrôler une source lumineuse artificielle mais n'avons aucunement l'intention de la laisser active en permanence. Nous l'activerons uniquement lorsque nous souhaiterons procéder à une mesure des photorésistances.

S'il s'agissait simplement d'une led blanche, la simple connexion à une sortie de l'Arduino, assortie d'une résistance, serait suffisante. Cependant, ceci ne serait sans doute pas suffisant pour éclairer l'ensemble de l'intérieur de la boite à lettres de manière convenable. Pour ce projet, en fouillant un peu dans le fourbi de récupérations glanés au fil des années, j'ai retrouvé deux ensembles de 3*3 leds et 3*8 leds blanches soudées sur une platine et accompagnées d'une grosse résistance. Ces modules récupérés je-ne-sais-où disposent d'un marquage 5V qui correspond à leur tension d'alimentation. Un essai préalable a permit de vérifier leurs fonctionnements et leurs efficacités.

Bien entendu 9 ou 24 leds blanches alimentées avec une tel tension consomment bien trop de courant. Plus qu'une sortie de l'Arduino ne peut en fournir. Il nous faut une sorte d'interrupteur permettant de contrôler l'alimentation. Dans ce genre de situation, un composant s'avère très pratique : le MOSFET ou transistor à effet de champ à grille isolée.

A la différence d'un transistor « simple » qui est contrôlé par un courant et nécessite des calculs complexes, un MOSFET est contrôlé par une tension. Sans entrer trop dans le détail, un MOSFET est composé de trois broches : le drain, la source et la grille (*gate*) généralement notés respectivement D, S et G. Le courant passe du drain à la source en fonction d'une différence de potentiel entre la source et la grille. Si cette tension dépasse un certain seuil, un canal de conduction va se créer dans le composant, laissant alors circuler les électrons de la source vers le drain. Comme le sens de circulation réel des électrons (charges négatives) est l'inverse du sens conventionnel, ceci signifie que le courant va alors circuler du drain vers la source. Cette tension de seuil est notée Vth dans les documentations techniques (*datasheet*) et celle entre la grille et la source Vgs.

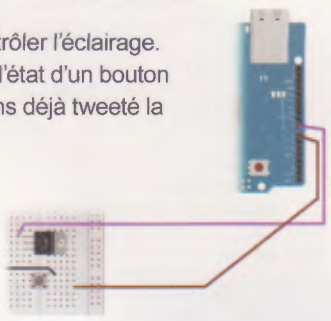
L'IRF520 utilisé ici est un modèle très courant. La valeur de Vgs pour que le MOSFET deviennent conducteur est de 4V. Attention, la mention « max » dans les documentations est trompeuse. Ce n'est pas la tension maximum utilisable (qui est de 20V avec ce modèle) mais la tension à partir de laquelle le MOSFET devient conducteur (fonctionnement en régime saturé). Les 5V d'une sortie Arduino sont au delà de ces 4V et ainsi, en mettant simplement la sortie à l'état logique 1 notre MOSFET devient conducteur. On peut résumer ce montage au schéma suivant :



Le modèle IRF520 permet une tension drain-source allant jusqu'à 100V avec un courant allant jusqu'à 10A (mais il chauffe et doit alors être équipé d'un dissipateur). Vous pouvez donc alimenter ainsi presque n'importe quelle source lumineuse, dont nos ensembles de leds de récupération mais également une ampoule à filament de 12V ou encore une led blanche de forte puissance. Inutile cependant de tomber dans des excès, nous voulons simplement un peu de lumière dans la boite à lettres, pas brunir le courrier...

4

La grille du MOSFET sera connectée à la sortie 8, nous permettant ainsi de contrôler l'éclairage. Mais nous utiliserons également la broche 7, en entrée cette fois, afin de vérifier l'état d'un bouton poussoir. Celui-ci sera utilisé pour deux choses : réinitialiser le fait que nous ayons déjà tweeté la notification de courrier et pour déclencher la calibration des photorésistances.



Notez que, contrairement au montage concernant la sonnette du précédent numéro, nous connectons directement le bouton à la fois sur la broche 7 et, de l'autre côté, à la masse. Pas de résistance de rappel externe ici, cette fois nous utiliserons celle activable dans l'Arduino même si cela consomme un peu de courant (et puis il faut bien changer pour montrer que c'est possible).

LE CROQUIS

Fichier Édition Croquis Outils Aide

```
#include <SPI.h>
#include <Ethernet.h>
#include <Wire.h>
#include <Twitter.h>

#define DELTA 20

// éléments réseau
byte mac[] = {
  0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED
};
IPAddress ip(192,168,111,11);
EthernetClient client;

// stockage mesures
int ldrnow[5] = {0,0,0,0,0};
int ldrbase[5] = {1023,1023,1023,1023,1023};

// éléments twitter
Twitter twitter("2560861158-bffdac91b4eb7c95ac5e929343480fdf");
char msg[141];
int tweeted = 0;

// configuration
void setup() {
  randomSeed(analogRead(5));
  Serial.begin(9600);
  pinMode(8, OUTPUT);
  pinMode(7, INPUT_PULLUP);

  if (!Ethernet.begin(mac)) {
    Serial.println("DHCP erreur !");
    Ethernet.begin(mac, ip);
    delay(2000);
  }
  Serial.println(Ethernet.localIP());
}

// fonction utilitaire
void blinky() {
  digitalWrite(8, HIGH);
  delay(100);
  digitalWrite(8, LOW);
  delay(150);
  digitalWrite(8, HIGH);
  delay(100);
  digitalWrite(8, LOW);
  delay(1000);
}

// calibration
void calib() {
  Serial.println("Debut calibration/reset tweeter");
  blinky();
  digitalWrite(8, HIGH);
  delay(50);
  ldrbase[0] = analogRead(0);
  ldrbase[1] = analogRead(1);
```

```
  ldrbase[2] = analogRead(2);
  ldrbase[3] = analogRead(3);
  ldrbase[4] = analogRead(4);
  delay(50);
  digitalWrite(8, LOW);
  delay(1000);
  blinky();
  tweeted=0;
  Serial.println("Fin calibration/reset tweeter");
}

// programme principal
void loop() {
  int status;

  if(digitalRead(7) == LOW)
    calib();

  digitalWrite(8, HIGH);
  delay(50);
  ldrnow[0] = analogRead(0);
  ldrnow[1] = analogRead(1);
  ldrnow[2] = analogRead(2);
  ldrnow[3] = analogRead(3);
  ldrnow[4] = analogRead(4);
  delay(50);
  digitalWrite(8, LOW);

  for(int i=0; i<5; i++) {
    if(ldrnow[i] > ldrbase[i]+DELTA) {
      ldrnow[i] = 1;
    } else {
      ldrnow[i] = 0;
    }
  }

  Serial.print(ldrnow[0]);
  Serial.print(' ');
  Serial.print(ldrnow[1]);
  Serial.print(' ');
  Serial.print(ldrnow[2]);
  Serial.print(' ');
  Serial.print(ldrnow[3]);
  Serial.print(' ');
  Serial.print(ldrnow[4]);
  Serial.print(' ');

  if(ldrnow[0] | ldrnow[1] | ldrnow[2] | ldrnow[3] | ldrnow[4]) {
    Serial.println("y'a du courrier !");
    if(tweeted == 0) {
      sprintf(msg, "y'a du courrier dans la boite (%lu)", random(2000000000));
      if (twitter.post(msg)) {
        status = twitter.wait();
        if (status == 200) {
          Serial.println("tweet OK");
        } else {
          Serial.println("erreur tweeter");
        }
      }
      tweeted=1;
    } else {
      Serial.println("c'est deja tweete");
    }
  } else {
    Serial.println("...");
  }

  delay(2000);
}
```

Arduino

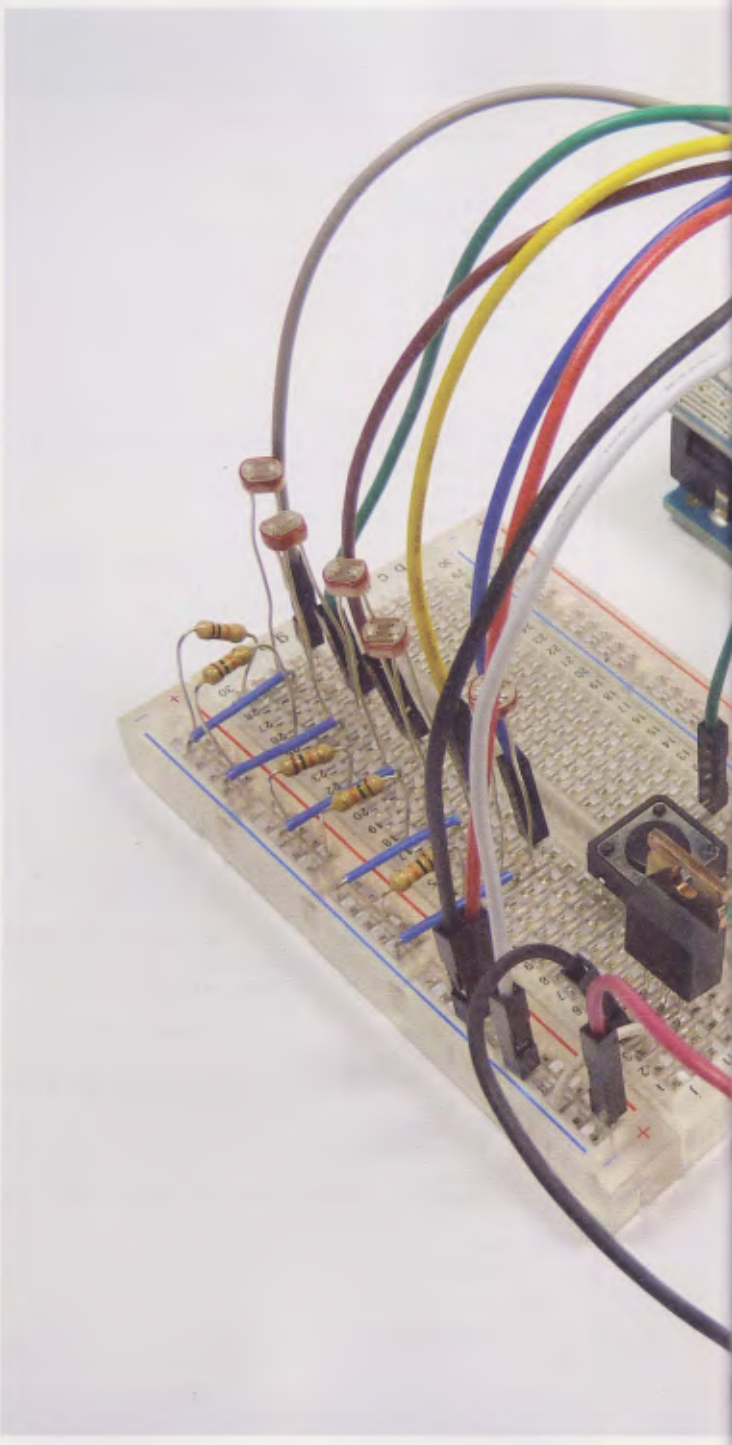
À PROPOS DU CROQUIS

En dehors du fait que le croquis est relativement conséquent, il reste relativement simple pour peu qu'on le découpe judicieusement en fonction de ce qu'il fait. La boucle principale s'exécute toutes les deux secondes environ. C'est le délai minimum entre deux vérifications du courrier. La logique est la suivante :

- On commence par tester si le bouton est enfoncé : si c'est le cas on appelle la fonction de calibration ;
- On relève les valeurs des cinq entrées analogiques sur lesquelles sont connectées les photorésistances. Le tableau `ldrnow[]` contient alors la valeur mesurée pour chacune d'entre elle ;
- On parcours les mesures pour les comparer avec le niveau maximum. Si l'une d'elle est supérieure à la valeur de calibration stockée dans la cellule correspondante de `ldrbase[0]` plus une petite marge (`DELTA`), elle est remplacée par `1`, sinon elle passe à `0`. Ceci est une simple technique pour mettre au point le programme et afficher les informations sur la sortie série (le moniteur série de l'environnement de développement Arduino), tout en nous facilitant ensuite l'évaluation d'une présence de courrier ;

- Vient ensuite le test via une simple condition `if` testant l'ensemble des cases du tableau via un *OU* logique. Si une seule d'entre elles contient une valeur non nulle (donc un `1`), une mention est tout d'abord envoyée sur le moniteur série, puis un message est composé pour Twitter incluant une valeur aléatoire entre 0 et 2000000000 (car Twitter n'aime pas les tweets identiques et refuse de traiter des tweets similaires à répétition). On teste ensuite la bonne marche de l'envoi et informons la liaison série du résultat. Enfin, comme un tweet a déjà été émis, on place la variable `tweeted` à `1` signifiant que la notification a été faite et qu'il n'est pas nécessaire de répéter l'opération lors du prochain passage dans la boucle.

Dans l'architecture générale du croquis, nous n'avons donc aucune particularité remarquable si ce n'est l'utilisation d'un tableau et non d'une série de variables. C'est un peu plus élégant que quelque chose comme :





```
ldr1 = analogRead(0);
ldr2 = analogRead(1);
[...]
```

Mais c'est largement perfectible. Pour économiser de la mémoire, nous pourrions également n'utiliser qu'un seul octet (8 bits) et faire usage des bits comme « tableau ». Les répétitions de `analogRead()` peuvent également être éliminés et remplacés par une boucle puisque nous avons déjà une variable `i` dans une autre boucle, ceci ne coûterait donc rien. Tout ceci est laissé, bien entendu, en guise d'exercice au lecteur qui est invité à étendre, améliorer et optimiser ce croquis...

ADC, CONVERSION ET CALIBRATION

Alors que les sorties dites analogiques ne le sont pas vraiment mais utilisent la PWM comme détaillé dans le précédent numéro, les entrées analogiques, elles, le sont. Le microcontrôleur intégré dans l'Arduino, l'Atmel ATmega328 par exemple, possède un ADC (*Analog-to-Digital Converter*) ou CAN en français (Convertisseur Analogique-Numérique) 6 canaux capables de mesurer une tension entre la masse et une tension de référence interne ou externe.

Par défaut, un croquis Arduino utilisera comme référence la tension d'alimentation de 5V ou 3,3V selon le modèle de carte. Il est cependant possible de configurer la référence via la fonction `analogReference()` en lui passant en argument, au choix :

- `DEFAULT` : par défaut la tension d'alimentation,
- `INTERNAL` : une référence interne dépendante du modèle d'Arduino (2,56V pour une Uno par exemple),
- `INTERNAL1V1` : 1,1V (disponible uniquement sur l'Arduino Mega),
- `INTERNAL2V56` : 2,56V (disponible uniquement sur l'Arduino Mega),
- `EXTERNAL` : une tension entre 0 et 5V appliquée sur la broche marquée AREF de la carte.

Pour notre projet, nous utilisons la référence par défaut et n'avons donc pas besoin de toucher à la configuration

avec cette fonction. `analogRead()` prend en argument la broche de l'Arduino à lire et retourne une valeur entre 0 et 1023 en fonction de la tension présente par rapport à celle de référence. La valeur retournée par la fonction est linéairement proportionnelle à la tension. Ceci signifie que si la tension est 0 la valeur sera 0, si la tension est celle de référence la valeur sera 1023 et si la tension est celle de référence divisée par 2, la valeur sera donc la moitié du maximum. Les LDRs, elles, ne fonctionnent pas comme cela et affichent une résistance en fonction de la lumière suivant une courbe définie par le constructeur. A la moitié de l'éclairage maximum vous n'aurez pas la moitié de la valeur de la résistance.

`blinky()` est une simple fonction permettant de faire clignoter le système d'éclairage. Le but est de signaler le début et la fin de la calibration. On part ici du principe que la boite à lettres laisse, d'une manière ou d'une autre, entrevoir la lumière qui s'y trouve. `calib()` est la fonction de calibration. Son travail est de procéder à une mesure unique et de stocker les valeurs dans le tableau `ldrbase[]` pour comparaison ultérieur.

LE SHIELD ETHERNET

Le shield Ethernet se plaçant sur la carte Arduino utilise une puce Wiznet W5100 interfacée via une liaison SPI. Le fait d'utiliser ce shield vous prive d'un certain nombre de broches sur la carte : 10, 11, 12, et 13 sur une Uno par exemple, correspondant au bus SPI via le connecteur ICSP. Vous ne pouvez donc plus utiliser ces broches pour vos projets. Ceci concerne en particulier la broche 13 puisqu'il s'agit de celle où est connectée la led L, du moins sur une Uno/Leonardo/Duemilanove (le problème ne se pose pas sur une Mega). Notez que les premières versions du shield Ethernet disposaient d'une led L mais que la révision 3 a réglé le problème.

Côté logiciel, un tel shield est directement supporté sans ajout d'éléments dans l'environnement de développement. Il vous suffit, en effet, d'ajouter ces deux petites lignes pour intégrer la bibliothèque dans vote croquis/projet :

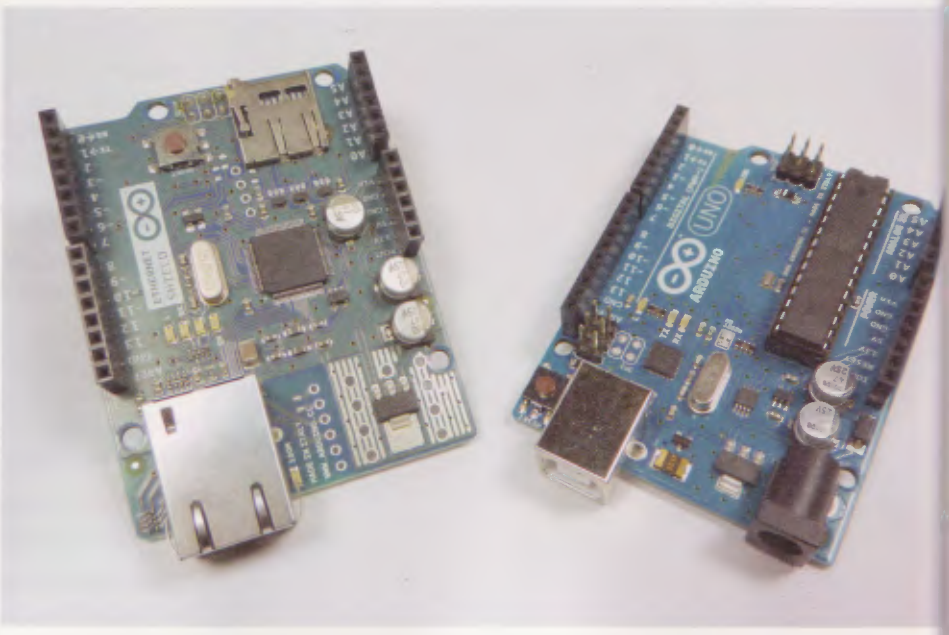
```
#include <SPI.h>
#include <Ethernet.h>
```

La première permet d'utiliser le bus SPI et la seconde le shield Ethernet. Dès lors vous pourrez définir quelques variables utilisées pour configurer le shield :

```
// définir l'adresse matériel (MAC)
byte mac[] = {
  0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED
};

// choisir une adresse IP au cas où
// DHCP ne fonctionne pas
IPAddress ip(192,168,111,11);

// définir un objet représentant la
// connexion Ethernet
EthernetClient client;
```



Mais ce n'est pas fini, dans la fonction `setup()`, vous devez activer la connexion IP, exactement comme vous le faites sous Windows ou Linux (ou comme cela est fait automatiquement à la connexion du câble) :

```
if (!Ethernet.begin(mac)) {
  Serial.println("DHCP erreur !");
  Ethernet.begin(mac, ip);
  delay(2000);
}
Serial.println(Ethernet.localIP());
```

Ici, la fonction `Ethernet.begin(mac)` prend en argument l'adresse MAC définie précédemment et l'utilise pour faire une requête DHCP. Normalement, votre box devrait y répondre et retourner une adresse IP utilisable ainsi que d'autres paramètres de connexion au réseau. Si l'appel à la fonction ne retourne pas une valeur positive c'est qu'un problème est survenu. Nous signalons l'erreur et initions alors une connexion avec une IP fixe, également choisie précédemment. La carte aura une adresse mais ne sera pas en mesure de se connecter à Internet. Enfin, la fonction `Ethernet.localIP()` nous permet d'obtenir l'adresse IP, soit obtenue, soit choisie, afin de l'afficher sur la liaison série (moniteur). Nous sommes dès lors connectés au réseau et, potentiellement, à Internet.

Petite remarque : l'adresse MAC pour le shield Arduino est un paramètre que le développeur (vous) spécifie. Cette adresse est normalement un élément de configuration unique propre à chaque interface Ethernet. La valeur utilisée ici, `0xDEADBEEFFEED` est totalement arbitraire et repose sur le fait qu'une autre interface sur le réseau ne dispose pas de la même adresse. Si vous utilisez plusieurs Arduino avec shield Ethernet, assurez-vous de définir des adresses différentes.

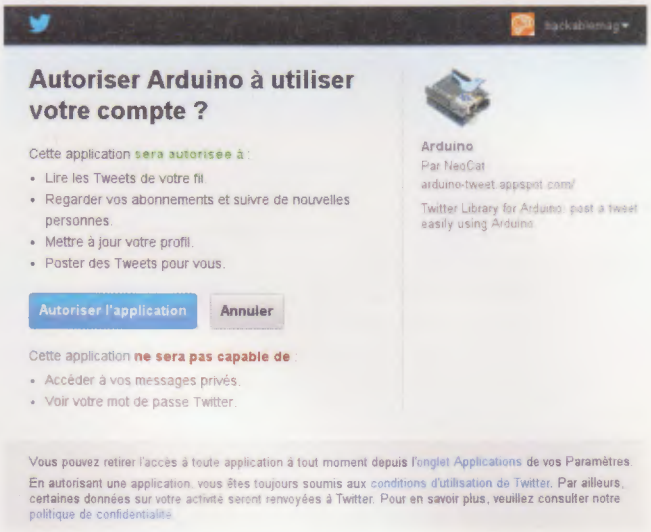
TWITTER

L'accès à Twitter pose un problème. En effet, bien que cette plateforme permette de développer des programmes tweetant des messages, les connexions depuis quelques mois doivent être sécurisées via HTTPS. Seulement voilà, une carte Arduino équipée d'un shield de type Ethernet ou Wifi n'est pas en mesure de supporter ce type de connexion. La puissance de calcul et surtout la mémoire nécessaire à des connexions reposant sur SSL/TLS (le « S » de HTTPS) est tout simplement au delà des capacités d'une carte Arduino.

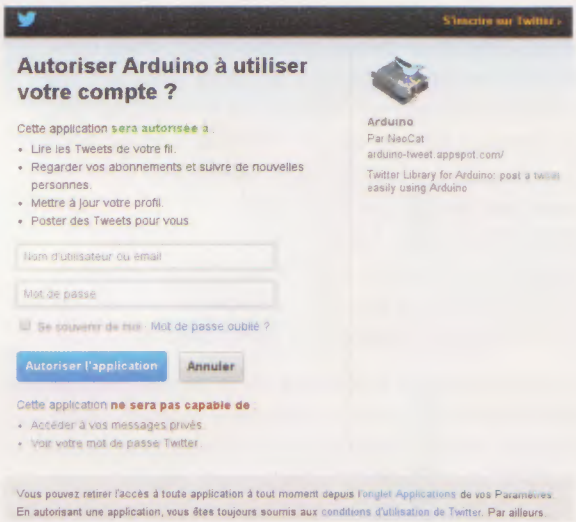
La solution consiste à passer par une plateforme intermédiaire. Pointez alors votre navigateur sur **https://arduino-tweet.appspot.com**. Vous trouverez à cet endroit tout ce qu'il vous faut pour que votre Arduino puisse Tweeter. Le principe de fonctionnement est fort simple. Comme l'Arduino ne peut pas directement utiliser l'API Twitter, une application reposant sur Google AppEngine se pose en intermédiaire. Ainsi votre Arduino va envoyer le tweet à cette application Appspot, dans SSL/TLS, qui va alors utiliser le service mis à disposition par Twitter via son interface pour développeur (API), avec SSL/TLS.

Pour que cela fonctionne en toute sécurité, vous devez autoriser cette passerelle à contrôler votre compte Twitter. Nous vous recommandons cependant de ne pas utiliser votre compte régulier pour cela (celui que vous utilisez en tant qu'humain) mais se créer un compte spécifique pour ces notifications. Ceci évitera de signaler aux personnes qui vous suivent que vous venez de recevoir du courrier.

Donner cette autorisation est relativement simple puisqu'il vous suffit de vous rendre sur le site et de cliquer sur « *Step 1 : Get a token to post a message using OAuth* ». Si vous êtes déjà connecté sur Twitter, vous arriverez directement sur l'écran d'autorisation :



Il vous suffira alors de cliquer sur « Autoriser l'application ». Dans le cas contraire, vous devrez également entrer votre identifiant et votre mot de passe Twitter pour ce compte :



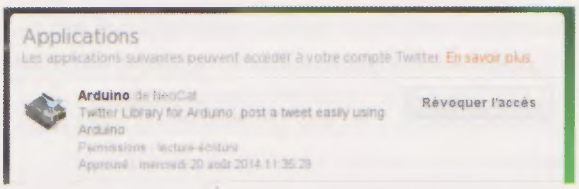
Notez que l'entrée de ces informations se fait bel et bien sur Twitter et non ailleurs, votre mot de passe reste caché. C'est tout le principe des autorisations par *token*. Celui-ci se matérialise par une longue suite de chiffres et de lettres qui, dès l'autorisation donnée, sera affichée sur l'écran du navigateur. Conservez cette chaîne de caractères précieusement, c'est elle qui permet à votre Arduino de Tweeter. Vous devrez alors la copier/coller en début de code :

```
Twitter twitter(">>> ICI <<<");
char msg[141];
int tweeted = 0;
```

Pour que notre croquis fonctionne il faudra, bien entendu, qu'une bibliothèque supporte une telle fonctionnalité. C'est l'étape numéro 2 de l'installation. Toujours sur le site officiel, cliquez sur « *Step 2 : Add some Libraries to your Arduino IDE* » et récupérez le fichier **Library-Twitter-1.3.zip**. Décompressez ce fichier. Il contient un répertoire **Twitter** qu'il vous suffira de placer dans un sous-répertoire **libraries** du répertoire où vos croquis sont enregistrés. Vous devez donc avoir une « Bibliothèques \ Documents \ Arduino \ libraries \ Twitter » sous Windows et un **~/sketchbook/libraries/Twitter** sous GNU/Linux. N'oubliez pas d'intégrer la bibliothèque dans votre croquis :

```
#include <Wire.h>
#include <Twitter.h>
```

Enfin, et non des moindres, il est important de préciser que vous pouvez retirer l'autorisation en place à tout moment. Il vous suffit pour cela de vous connecter à Twitter avec le compte utilisé pour votre croquis et de vous placer dans « Paramètres » et « Applications ». Parmi les autorisations en place (TweetDeck, Google, Twitter pour Anrdoid, etc), vous devez retrouver celle nommée « Arduino de NeoCat » :



Il vous suffit de cliquer sur « Révoquer l'accès » et le tour est joué.

Avant de vous lancer dans la réalisation du projet tel qu'il est présenté ici, nous vous recommandons de créer une première version sans la fonctionnalité de tweet et de tester cela en parallèle avec le croquis fourni en exemple sur le site. Enfin, une fois que vous êtes satisfait du fonctionnement, fusionnez les deux.

NÉCESSITÉ DE L'ALÉATOIRE

Le problème évoqué précédemment concernant les tweets identiques est réglé en ajoutant une valeur pseudo-aléatoire aux messages sous la forme d'un nombre entre parenthèses. Cette valeur est issue de la fonction **random()** qui doit être appelée après initialisation depuis la fonction **setup()** avec **randomSeed(analogRead(5))**. Le générateur de nombre pseudo-aléatoire utilise une « graine » pour créer une série aléatoire de valeurs dans laquelle piocher. Cette graine détermine la série de valeurs et doit donc être choisie de manière vraiment aléatoire. Une technique courante est, comme ici, d'utiliser une lecture d'une entrée analogique non connectée. Ceci retourne une valeur difficile à prédire et de ce fait, à chaque mise sous tension, nous avons de bonnes chances (1/1024) de démarrer le générateur avec une valeur au hasard.

Notez également la manière de produire la chaîne **msg** en utilisant la fonction C classique **sprintf()**.

Le `%lu` placé dans la chaîne de caractères est un spécificateur de format. Ici, nous incluons une variable numérique `unsigned long`, ce qui a pour effet de transposer une valeur en une succession de chiffres et donc de produire la chaîne de caractères complète qui sera tweetée.

LOGIQUE D'UTILISATION

Un dernier mot sur le fonctionnement « côté utilisateur » du projet. L'idée générale est la suivante : après mise sous tension du montage, le niveau de déclenchement de la détection est au maximum. Il n'est donc pas possible qu'une détection ait lieu car la valeur mesurée ne peut pas être supérieure à 1023+20 (base+DELTA). Une pression longue sur le bouton poussoir va déclencher la calibration. Celle-ci sera signalée par deux brefs clignotements du système d'éclairage, puis deux autres clignotements après la mesure.

Cette opération va mesurer le niveau de lumière dans la boite en l'absence d'obstacle et donc de courrier. La valeur ainsi obtenue sera augmentée de 20 (`DELTA`) par sécurité et sera la base de comparaison. Dès lors, les mesures faites toutes les deux secondes déclencheront une notification si le niveau lumineux est suffisamment bas. Si cela arrive, la notification Twitter est faite et le système continue ses mesures mais en se passant de retweeter inutilement.

A la relève du courrier, il suffira d'une pression longue sur le bouton pour procéder à une nouvelle calibration de la boite vide et à l'effacement du blocage interdisant de nouveaux tweets. Alors que la seconde raison est évidente, la première mérite une explication. Invariablement, l'environnement va dégrader les conditions de mesure (crasse, poussière, etc). En calibrant à chaque relève de courrier, nous gardons le système synchronisé avec les références de mesure. DB

CE QUE NOUS AVONS APPRIS



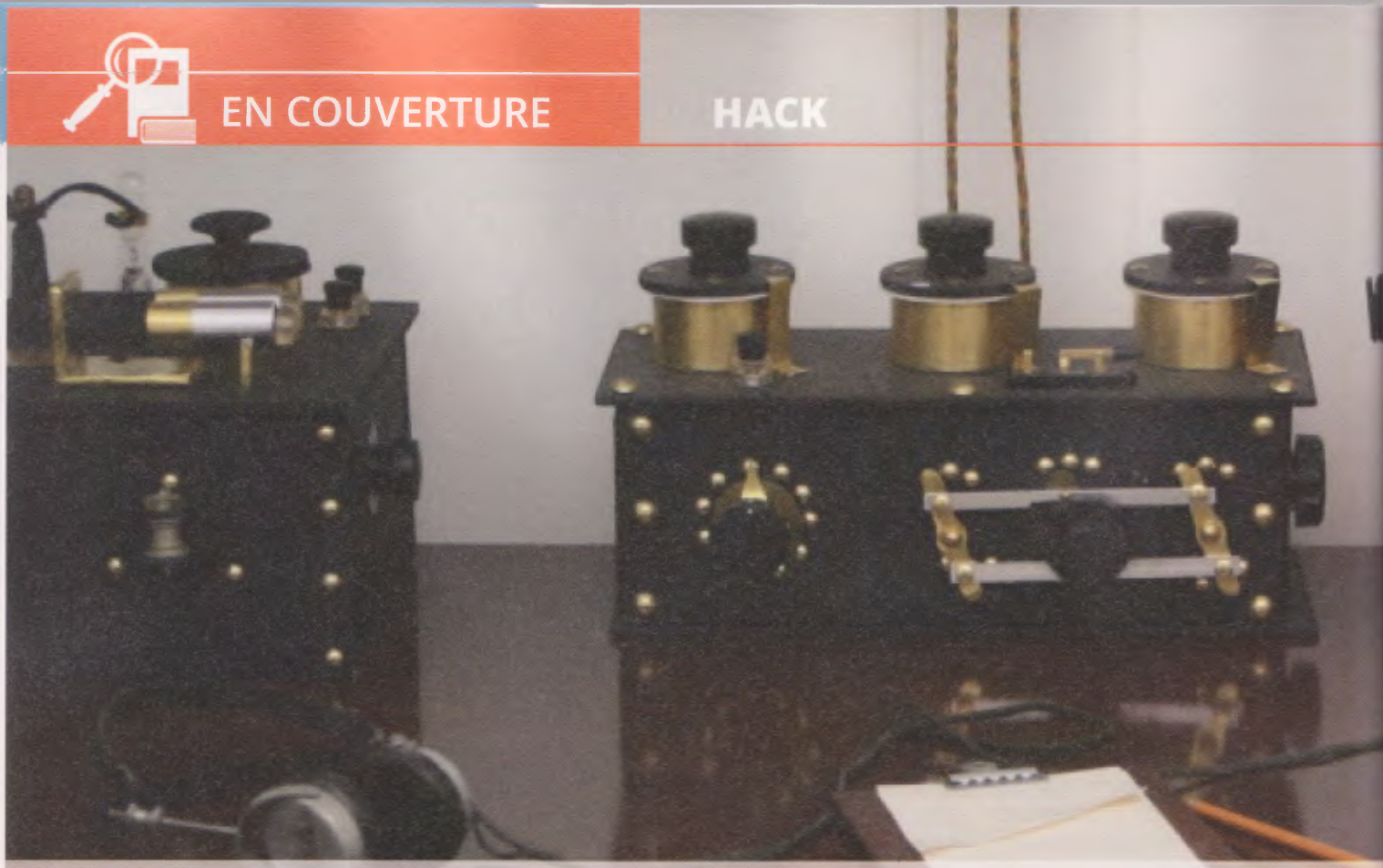
Ce long projet nous a permis d'appréhender bon nombre de domaines et de fonctionnalités que le duo Arduino + Ethernet est en mesure de nous fournir. L'utilisation du convertisseur analogique/numérique et des entrées dédiées avec les photorésistances montées en diviseur de tension est très représentatif de la manière d'utiliser n'importe quel élément résistif variable. Ici, il s'agissait de la lumière mais une résistance peut varier en fonction de bien d'autres paramètres comme l'hygrométrie, la température ou encore en raison d'un effet mécanique.

Mais l'aspect le plus intéressant est sans le moindre doute l'utilisation même de la connexion Ethernet et donc d'Internet pour nos projets. Certes les Arduino, du fait qu'ils ne soient pas des monstres de puissance, possèdent des limitations très pénalisantes mais ils n'en deviennent pas pour autant inutils. Des solutions existent, comme ici avec Twitter, pour contourner la nécessité d'utiliser des connexions dites *over SSL*.

Ce projet possède, à mon avis, un potentiel d'extension très important. L'Ethernet peut être remplacé par le Wifi (bien que j'ai personnellement une fâcheuse tendance à préférer les câbles bien solides et matériels) ou d'autres types de liaisons : série, CAN, Bluetooth, ZigBee... L'autonomie est également une voie pour améliorer le projet. En effet, dans l'état, il suppose un fonctionnement avec une alimentation continue fournie de manière externe. Il est cependant parfaitement envisageable de réfléchir à un fonctionnement sur batterie ou via un système d'alimentation solaire. Certes, ceci demande un shield supplémentaire (plusieurs sont disponibles) et possède donc un coût mais vous évitera le casse-tête de l'alimentation en extérieur.

Au niveau logiciel, ce croquis délibérément simpliste, laisse une grande place à la personnalisation. Parmi les idées qui viennent immédiatement nous avons, par exemple, une meilleure gestion du bouton poussoir, un délai de mesure réglable, un affichage LCD ou encore une notification lumineuse ou sonore supplémentaire.

Enfin, nous avons la réalisation elle-même. Ici il s'agit d'un simple prototype. La mise en oeuvre effective et l'installation du système fera sans le moindre doute appel à votre imagination et à vos talents de bricoleur. Si vous sautez le pas, n'hésitez pas à nous envoyer les photos de vos réalisations, nous nous ferons un plaisir de les partager.



ÉMISSION RADIO : VOUS N'IMAGINEZ PAS COMME « ÇA CAUSE » AUTOUR DE NOUS

Denis Bodor

Nous le savons tous, autour de nous existe tout un environnement électromagnétique et se promènent des émissions radio en tous genres. Mais connaissez-vous vraiment la nature à la fois de ces émissions et des informations qui transitent de la sorte ? Vous allez être surpris car les signaux et les sources que vous devinez sans peine (radio FM, téléphonie, Wifi, Bluetooth, etc) ne représentent qu'une petite partie de ce qui circule autour de nous...

Précisons de suite que cet article ne traitera pas de l'électro-hypersensibilité et nous n'aborderons même pas le sujet. En dehors des considérations sur les aspects imaginaires, psychiatriques ou bien réels de ce syndrome, ce choix est fait pour deux raisons : ceci serait totalement hors du cadre du magazine et, plus important encore, les personnes qui seraient atteintes d'un syndrome d'intolérance aux champs électromagnétiques font mathématiquement de très mauvais bidouilleurs car notre domaine de prédilection et nos tendances compulsives à nous entourer d'appareils en tous genres sont évidemment une source de rayonnement électromagnétique non négligeable. Voilà, c'est dit !

1. LA RADIO, UNE INVENTION QUI NE DATE PAS D'HIER

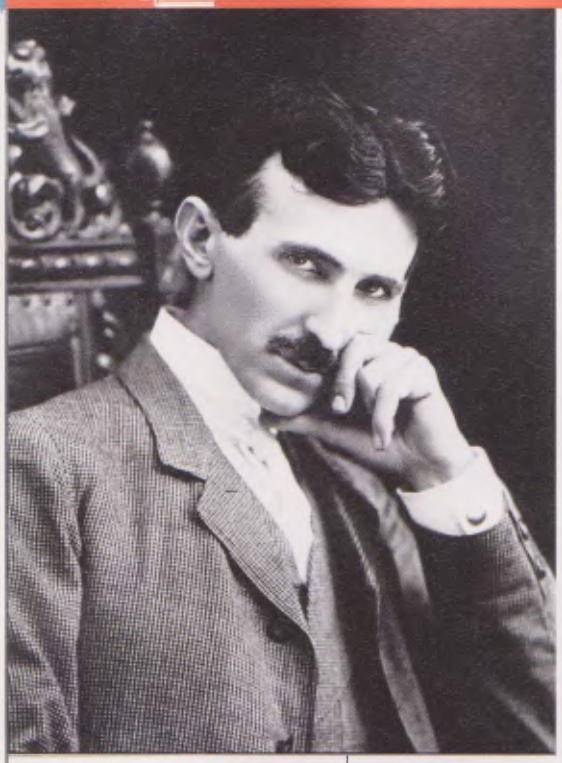
J'adore casser du mythe et celui-ci me procure un plaisir tout particulier car impliquant un personnage de l'histoire des sciences que j'affectionne tout particulièrement. C'est en 1886 que débute, si l'on peut dire, l'histoire de la radio avec la découverte des ondes radio par le physicien allemand *Heinrich Rudolf Hertz*. Derrière ce personnage peu connu du grand public se cache la démonstration expérimentale du fait que la lumière est une onde électromagnétique (théorie de Maxwell) ainsi que la mise en évidence de l'existence d'autres ondes électromagnétiques, invisibles, qui peuvent également se réfracter et se diffracter. Hertz en arrive même, dans ses travaux, à établir une liaison entre un émetteur et un récepteur. On parlera ensuite d'ondes hertziennes et le nom de ce chercheur deviendra plus tard une unité de mesure : le hertz. La rumeur (ou la légende) prétend pourtant qu'il n'y voyait pas d'application pratique...

C'est peu de temps après sa mort qu'un certain *Guglielmo Marconi* qui, grâce à ce qu'on appellerait maintenant des talents marketing, a vu son nom popularisé en reprenant les travaux de Hertz (1895) et en créant un télégraphe sans fil, déposant par la même occasion un brevet (1896). Quelques temps après, il réalise une liaison transmanche puis transatlantique entre Poldhu et Terre-Neuve, l'inscrivant définitivement dans l'histoire comme l'inventeur de la radio.

Ce que l'histoire ne dit pas, ou du moins pas suffisamment, c'est qu'un génie des sciences, dès 1889 expérimente la première communication radio grâce à un générateur de sa conception. Cet inventeur et ingénieur est le mythique *Nikola Tesla*, qui lui aussi possède aujourd'hui une unité à son nom (mesurant l'induction magnétique). Malheureusement, Tesla est un inventeur hors paire (au sens strict du terme) mais un bien mauvais commercial ou financier. Ainsi, ses créations n'obtiennent pas l'écho qu'elles méritent alors qu'elles ont changées le monde. La radio entre tout à fait dans cette catégorie, tout comme ses nombreuses inventions exploitant le courant alternatif dont il a été le principal promoteur. Il jettera les bases de la robotique, du radar, du microscope électronique, des lampes basse-consommation ou encore des accélérateurs de particules. Mais cette prolifération d'inventions étonnantes pour l'époque et la tendance qu'a Tesla à extrapoler les applications pratiques de ses découvertes ont largement contribué à laisser de lui l'image d'un savant fou, reprise et exagérée par la littérature et le cinéma.



Guglielmo Marconi, l'homme à qui est généralement attribué l'invention de la radio



Nikola Tesla en 1896, un véritable génie, excentrique, passionné et visionnaire qui, au fil du temps, est presque devenu un personnage mythique à qui certains attribuent des inventions loufoques (rayon de la mort, modification du climat, téléportation) alors que ses travaux bien réels sur le courant alternatif, la transmission d'énergie, la radio, la résonance électromagnétique ou les hautes fréquences ont drastiquement changés et façonnés le monde moderne. Le web grouille d'arguments du type « Ça existe, Nicolas Tesla l'a fait ! », et de la réponse adaptée par ceux qui se sont effectivement documentés : « C'est Nikola, et non, il ne l'a pas fait »...

Il disparaîtra en janvier 1943 à l'âge de 86 ans, seul, couvert de dettes et reclus dans une chambre d'hôtel, alors même que le monde lui est redevable, suite à la guerre des courants qu'il livre contre Thomas Edison, pour avoir trouvé la solution au problème de distribution électrique à grande échelle. C'est après une procédure judiciaire que sa famille et en particulier son neveu obtient la restitution intégrale de ses documents de travail et ses effets personnels ainsi que l'urne funéraire contenant les cendres du scientifique... plus de 20 ans après sa mort. Là encore, la confiscation par l'administration américaine de ces documents à son décès ne fait qu'amplifier le mythe.

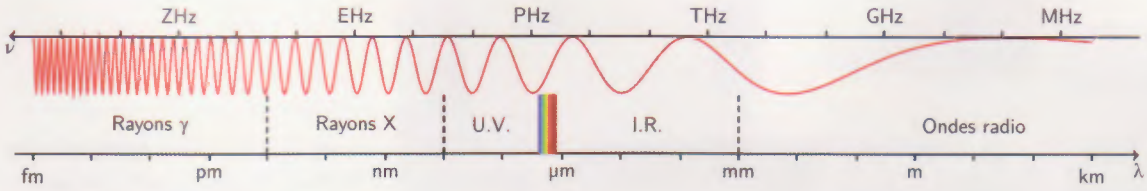
Aujourd'hui peu nombreux voient simplement en Tesla l'inventeur de la radio et d'autres technologies très importantes utilisées abondamment de nos jours, et donc le génie qu'il était réellement, mais versent soit dans la plus parfaite ignorance, soit dans le mythe et le fantasme de l'énergie libre, gratuite et disponible a profusion, ignorant les lois de l'univers que Tesla lui, savait parfaitement interpréter. Son nom est aujourd'hui davantage associé au rayon de la mort, à la téléportation ou au contrôle du climat qu'à ses travaux pourtant remarquables sur tous les points. C'est au nom de Marconi que la radio est associée dans la mémoire populaire et à Edison les nombreuses innovations qu'il n'a fait qu'acheter à Tesla. Un peu comme aujourd'hui où le nom de Steve Jobs est indéniablement associé à Apple et à la naissance de l'informatique personnelle, repoussant dans l'ombre des noms comme Stephen Wozniak, Clive Sinclair, Chuck Peddle. Ainsi va le monde...

2. QU'EST CE QU'UNE ONDE RADIO ?

Une onde électromagnétique est une forme de transfert d'énergie qui se manifeste sous la forme d'un champ électrique couplé à un champ magnétique (whaou !). Nous sommes tous équipés de récepteurs d'ondes électromagnétiques capables de percevoir une petite partie de ce rayonnement électromagnétique : nos yeux. En effet, ces globes gélatineux, parfois très jolis, sont en mesure de détecter ce rayonnement mais pas dans son ensemble. On nomme cette minuscule partie, la lumière.

Une onde électromagnétique est exactement comme une onde ou une vague à la surface de l'eau. Elle se définie par plusieurs caractéristiques dont sa fréquence et son amplitude. La fréquence est le nombre de fois qu'un phénomène périodique se reproduit par unité de temps. En voyant ainsi une onde électromagnétique comme une perturbation cyclique, sa fréquence est le nombre de fois par seconde où cette perturbation se reproduit. On parle alors de hertz (hz). Mais on peut également quantifier l'importance de cette perturbation, son amplitude. Enfin, pour faire le tour des termes d'usages nécessaires, une onde électromagnétique se déplace dans l'espace à la vitesse de la lumière (299792458 m/s). Comme le phénomène est cyclique, il existe une distance entre deux points de l'onde strictement identiques à un instant précis : c'est la longueur d'onde (oui, comme l'expression « être sur la même longueur d'onde » et donc communiquer sur la même fréquence et donc se comprendre).

Nos yeux sont en mesure de percevoir des ondes électromagnétiques d'une fréquence comprise entre 790000000000000 hz à 385000000000000 hz ou, 790 à 386 terahertz (milliers de milliard de hertz). Comme ces gros chiffres ne sont pas très pratiques, on préfère généralement utiliser la longueur d'onde, soit entre 390 nanomètres



Le spectre électromagnétique. Le ridicule petit morceau multicolore au centre est la partie que nous pouvons percevoir avec nos yeux. Pour le reste, nous avons besoin de construire des outils comme des détecteurs et des antennes (source : Benjamin ABE - Wikipédia - CC BY-SA 3.0)

pour une lumière violette et 780 nanomètres pour le rouge. Entre se trouve le bleu, le vert, le jaune et l'orange. A une fréquence plus élevée on trouve l'ultraviolet, les rayons X et les rayons Gamma. A une fréquence plus basse nous avons les infrarouges puis les ondes radio (300 Ghz et en dessous).

Frustrés de n'avoir accès qu'à une petite part du gâteau, les humains ont, comme à leur habitude, inventé des équipements permettant de palier à leurs limitations. Nous avons donc créé entre autres choses des émetteurs et des récepteurs d'ondes radio et découpé joyeusement cette région des ondes nous étant enfin accessible en de plus petits morceaux en leur attribuant des noms (ça aussi, chez les humains, on aime bien). On trouve ainsi, par exemple :

- UHF (ultra haute fréquence) entre 300 MHz et 3 GHz,
- VHF (très haute fréquence) entre 30 MHz et 300 MHz,
- HF (haute fréquence) entre 3 MHz et 30 MHz, fut un temps les « grandes ondes »,
- MF (moyenne fréquence) entre 300 kHz et 3 MHz,
- et LF (basse fréquence) entre 300 kHz et 3 MHz (les « petites ondes »).

Bien entendu, ces morceaux se divisent encore en ce qu'on appel des bandes de fréquences. En fonction de leurs caractéristiques, les ondes électromagnétiques peuvent avoir diverses propriétés comme le fait de traverser plus facilement certaines matières, de se propager sur de plus grandes distances ou encore de permettre des communications plus rapides.

3. QUEL MATÉRIEL POUR LA RADIO

La réception d'émissions électromagnétiques repose sur le phénomène d'induction. La définition de l'induction est la suivante (merci Wikipédia) : « phénomène physique produisant une différence de potentiel électrique dans un conducteur électrique soumis à un champ magnétique variable ». En clair, en présence d'une onde électromagnétique, un courant va être induit dans un conducteur électrique. Ceci dépend de la fréquence de l'onde, de son amplitude mais également de la longueur du conducteur et de son orientation par rapport au champ électromagnétique.

Inversement, lorsqu'un courant passe dans un conducteur, il se forme un champs électromagnétique. Ainsi, si le courant n'est pas continu mais varie à une certaine fréquence, le champs magnétique ainsi créé fait de même et une onde est produite et émise.

C'est sur ces deux principes que reposent la genèse des ondes radio et des communications utilisant des rayonnements électromagnétiques. Deux noms sont associés à ces découvertes : *Hans Christian Oersted* qui constate dès 1820 qu'un fil conducteur parcouru par un courant fait dévier l'aiguille d'une boussole et *Michael Faraday* en 1831 qui fait le constat inverse, un champ magnétique peut induire un courant.

Dans les grandes lignes, la radio c'est simplement ça. En émission, on « excite » un conducteur avec un courant variable de manière à lui faire émettre une onde et en réception, on mesure le courant et les tensions induites dans un conducteur exposé à une onde.



Une station de radiotélégraphie telle que celle présente dans le Titanic et dite de type Marconi destinée à envoyer et recevoir des messages codés en morse (source Wikipedia - Cliff1066 - CC BY 2.0)

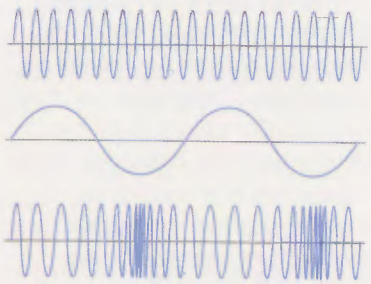
L'électromagnétisme est la branche de la physique qui étudie ces phénomènes en grande partie régis par les résultats du travail de *James Clerk Maxwell* et plus précisément les équations dites de Maxwell-Lorentz (en référence au travaux de *Hendrik Antoon Lorentz*).

Sur le papier ceci à l'air fort simple mais dans la pratique les choses se compliquent rapidement. En effet, un récepteur radio se résume à une antenne et un système permettant de mesurer les tensions et les courants induits dans l'antenne. Le problème cependant est qu'une antenne, quelle qu'elle soit, ne reçoit jamais qu'une seule fréquence mais tout un tas de signaux d'où il faut alors extraire celui que nous voulons capter.

De plus, pour transmettre des informations, il est nécessaire de manipuler le signal ou plus exactement de le moduler. La forme la plus simple de modulation est le « tout ou rien » (en fait ce n'est pas vraiment une modulation car rien n'est modulé). En gros, lorsqu'il y a un signal émis sur la fréquence choisie, c'est une formation et lorsqu'il n'y en a pas, s'en est une autre. Du coup, il est possible de jouer sur la durée du signal avec, par exemple, des émissions courtes et des émissions longues et ainsi définir une correspondance entre des suites de « longs » et de « courts » avec des lettres de l'alphabet. Et voici comment on en arrive au code morse du nom de Samuel Morse qui l'aurait inventé (certains pensent que ce serait plutôt son assistant, Alfred Vail). Le code morse est initialement créé pour le télégraphe (filaire donc) mais sera rapidement utilisé en radiotélégraphie.

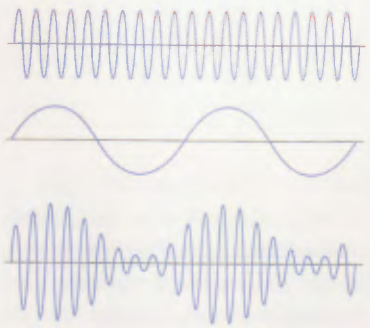
Tout le monde connait d'ailleurs le signal « SOS », trois courts, trois longs, trois courts ou » ... --- ... mais il est amusant également de remarquer que le code morse était aussi utilisé avec les téléphones Nokia (vous savez, lorsqu'ils étaient leaders du marché) pour signaler les messages avec ... -- ... signifiant « SMS ».

Mais depuis l'invention de ce code, les choses ont bien changé et il existe aujourd'hui plusieurs techniques de modulation. Il est ainsi possible de moduler un signal en amplitude, en fréquence, en phase, etc. Tous ces systèmes reposent sur le même principe : changer les caractéristiques du signal émis en fonction de l'information qu'on souhaite transmettre. Le récepteur a alors à sa charge de démoduler le signal pour récupérer l'information.



Exemple de signal modulé en fréquence (FM). En haut le signal porteur, au centre le signal contenant l'information et en bas, la porteuse modulée d'après le signal. Remarquez les correspondances entre les variations du sinus central et sa répercussion sur le signal modulé.

Si nous prenons l'exemple d'un récepteur « radio FM » capable de recevoir des émissions de 87,6 MHz à 107,9 MHz, l'utilisateur règle tout d'abord l'appareil sur une fréquence porteuse de base, disons 90 Mhz. L'appareil va alors devoir filtrer et ne récupérer que les fréquences comprises entre +/- 40 Khz autour de 90 Mhz.



Modulation d'amplitude. Ici la porteuse (en haut) est combinée au signal à transmettre (au centre) pour obtenir le signal modulé (en bas). Les variations d'amplitude sont littéralement copiées sur la porteuse.

Enfin, il va démoduler, c'est à dire extraire l'information utile des variations du décalage de la fréquence au cours du temps. Il obtient alors un autre signal qui correspond à l'émission audible qu'il envoi aux haut-parleurs qui produisent le son. Il est important de faire la différence entre la « bande FM » qui est la gamme de fréquences utilisées par la radio FM (~ 20 Mhz) et la bande utilisée par une émission de ce type (quelques dizaines de Khz).

Un autre exemple est la modulation d'amplitude. Là ce n'est plus la fréquence du signal sur laquelle on joue pour transmettre une information mais sa « force » ou plus exactement la différence entre le « haut » et le « bas » de l'onde, c'est à dire l'amplitude crête à crête. C'est l'un des systèmes de modulation qui fut utilisé en premier car pour moduler le signal il suffit de prendre l'onde porteuse et de l'additionner au signal qu'on souhaite émettre comme de la voix ou de la musique. Le récepteur lui, se calera sur la fréquence et soustraira l'onde porteuse pour obtenir le signal original et le restituer, par exemple, sur des haut-parleurs.

Ce ne sont là, bien entendu, que deux systèmes de modulations. On distingue de nos jours :

- CW (*Continuous Wave*) : il n'y a pas de modulation à proprement parlé, c'est l'émission et la non émission du signal qui transporte l'information,
- AM (*Amplitude modulation*) : la modulation d'amplitude,
- FM (*Frequency Modulation*) : modulation de fréquence. On distingue NFM (*Narrowband Frequency Modulation*) et WFM (*Wideband Frequency Modulation*) qui se différencient par la largeur de bande utilisée pour la modulation. Il existe aussi l'UWB (*Ultra-Wide Band*).
- USB (*Upper SideBand*) et LSB (*Lower SideBand*) qui sont des types de modulations d'amplitude regroupées sous la désignation BLU pour Bande Latérale Unique ou SSB en anglais (*Single SideBand*). Avec ce type de modulation, la porteuse est fixée sur une certaine fréquence et on observe l'amplitude de la fréquence juste en dessous (LSB) ou juste au dessus (USB).

A cela, il faut ajouter les techniques permettant de transmettre des données numériques. C'est le cas du FSK (*Frequency-Shift Keying*) ou modulation par déplacement de fréquence (MDF). Là, le signal varie entre plusieurs fréquences prédéfinies arbitrairement ce qui permet d'encoder des 0 et des 1 qui sont ensuite utilisés pour transporter l'information. On trouve également d'AFSK qui travail dans les fréquences audibles (c'est le signal démodulé qui doit être re-démodulé) et le MFSK utilisant plus de deux fréquences.

Pourquoi parler de toutes ces choses et de tous ces acronymes barbares ? Tout simplement parce que, si vous souhaitez vous pencher sur le domaine, vous allez être obligatoirement confrontés à ce jargon qu'on retrouve naturellement dans les documentations mais également dans les logiciels dont nous allons parler.

4. L'AVÈNEMENT DE LA SDR ET DE RTLSDR

Pendant une période très longue qui a débutée avec les premières heures de la radio, l'émission, la réception, la modulation et la démodulation des signaux n'étaient l'affaire que du matériel. Au fil du temps, des circuits de plus en plus complexe ont été utilisés pour ces tâches,

nécessitant des connaissances poussées en électronique analogique et un apprentissage long et intense. Alors que les industriels fabriquaient du matériel dédié aussi bien pour émettre et recevoir, les amateurs ou plus exactement les radio-amateurs eux construisaient leurs équipements et exploraient le domaine.

La communauté radio-amateur est toujours présente et très active (bien que vieillissante). Mais peu nombreux sont ceux qui aujourd'hui prennent ou disposent du temps nécessaire pour apprendre et commencer à maîtriser le domaine (et j'en fais parti). A la clé, pourtant, se trouve une masse de connaissances en électronique et une fraternité sans pareil. Pour devenir radio-amateur et avoir le droit d'émettre, la législation impose un examen permettant de décrocher une licence. Avec ce diplôme il devient possible d'utiliser les ondes et de réaliser son propre équipement sans homologation technique. En France, une licence de classe unique (classe 2 CEPT) remplace depuis 2012 les trois classes précédentes et place ainsi la barre relativement haut. Selon Wikipédia, la moyenne d'âge des radio-amateurs en France est proche de 60 ans et moins de 1% d'entre eux ont moins de 25 ans.

La disparition de la licence de classe « novice » (3) permettant de débuter rapidement ne facilite pas les choses mais la technologie est peut-être entrain de venir au secours de cette communauté. En effet, en parallèle à l'évolution du domaine, l'informatique et les circuits numériques sont apparus et ont envahi le quotidien. La puissance de calcul disponible a rendue possible le traitement efficace des signaux par de simples ordinateurs personnels.

La SDR ou *Software Defined Radio* met en oeuvre une architecture particulière. Ce n'est plus le matériel qui procède à la démodulation mais un logiciel installé sur un PC ou un Mac. Utilisé en réception, la radio logicielle, permet à tout un chacun de devenir radio-écouteur, scrutant les ondes et découvrant ainsi un domaine captivant. Du point de vu de la législation, le radio-écouteur n'a pas l'obligation de posséder une licence mais doit simplement respecter des obligations :

- Utiliser du matériel autorisé par la loi et en vente libre,
- et respecter la confidentialité des communications et donc ne pas divulguer le contenu des émissions qu'il pourrait entendre.



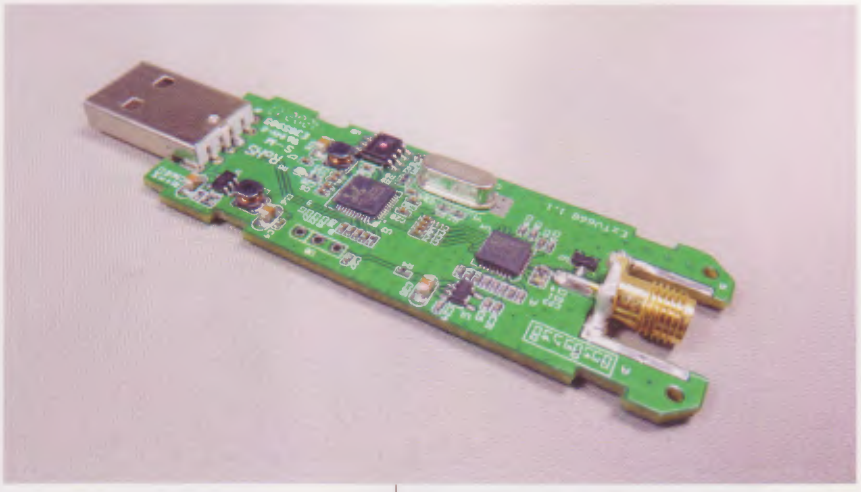
Exemple de « dongle » permettant la réception DVB-T. Le matériel est généralement livré avec une antenne à peine utilisable, une télécommande et un CD contenant les logiciels pour Windows. Avec la popularité grandissante entourant l'utilisation ce type de matériel dans le domaine de la SDR, les revendeurs en font maintenant directement mention dans leur description produit.

Du point de vu pratique, un matériel spécifique, en plus d'un ordinateur, est toujours nécessaire mais celui-ci se borne à la réception des émissions et leur numérisation. Il transmet alors ces données brutes au logiciel qui démodule et traite le signal. Pendant longtemps ceci prenait la forme de récepteur radio connectés à un PC, puis sont apparus des systèmes numériques connecté en USB, toujours assez chers et donc peu accessibles.

Et puis un jour, c'est un développeur Linux (*Antti Palosaari*) qui fait une découverte intéressante et en fait mention sur une liste de diffusion. Alors qu'il développe un pilote pour un récepteur DVB-T (télé TNT), il se rend compte que cette clé USB n'est rien d'autre qu'un récepteur qui numérise tout ce qu'il reçoit et laisse l'ordinateur et le logiciel Windows faire tout le reste du travail, chose qu'à également constater *Eric Fry* qui étudiait le protocole USB utilisé par le *dongle* dans le but de le faire fonctionner sous GNU/Linux.

Au final, il s'avéra que la puce à la base du périphérique, la Realtek RTL2382U était précisément destinée à la SDR par le fabricant et les choses alors s'accélèrent. Les développeurs du projet Osmocom, qui avaient déjà commencés à concevoir sur leur propre matériel, se rendent compte que leur travail est adaptable à toute une gamme de récepteur DVB en USB disponible dans le commerce. Naît alors RTLSDR.

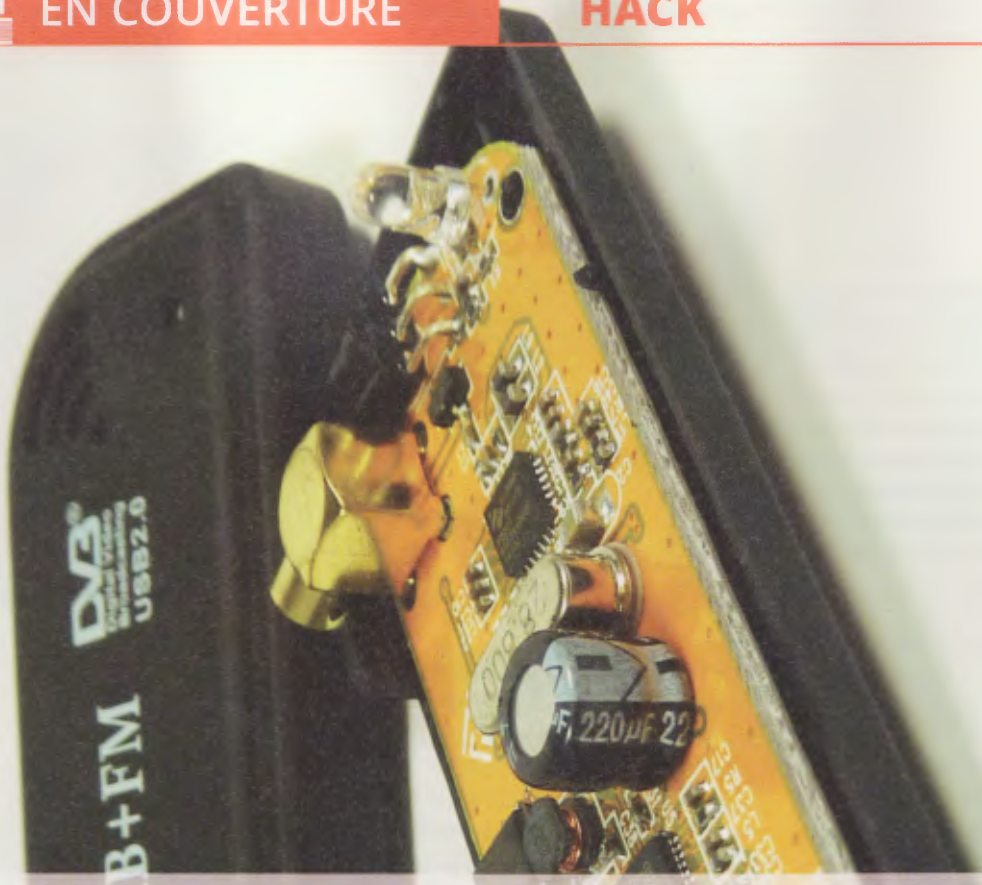
Au final, il s'avère qu'un récepteur DVB-T vendu dans le but de recevoir les émissions TNT est en mesure de faire bien plus que cela. Le matériel initial se composait d'un circuit RTL2382U numérisant les données avant de les envoyer via USB et d'un tuner (syntoniseur) Elonics E4000 chargé de recevoir le signal émis sur une fréquence particulière. Depuis, les clés USB intégrant un E4000 sont devenues presque introuvables mais un autre composant a pris sa place, le RV820T.



Récepteur DVB démonté et modifié. Ce modèle s'est vu retiré de récepteur infrarouge destiné à la télécommande et le connecteur d'antenne 9,52 (Belling Lee ou « nouvelle norme ») a été remplacé par du SMA bien plus pratique. La puce de droite est le tuner E4000 et celle de gauche le convertisseur/ADC/USB Realtek RTL2382U.

Aujourd'hui, ce matériel est disponible partout pour 15€ à 20€ et est en mesure de recevoir les émissions dans des fréquences comprises entre 24 Mhz et 1850 Mhz. Cela couvre énormément de bandes de fréquences largement utilisées pour une quantité impressionnante d'applications. Ceci inclus les systèmes de signalisation, les télécommandes, les émissions d'aéronefs en vol, la radio FM bien sûr, les capteurs domotiques, les télécommandes de jouets et de modélisme, etc.

Dans la suite de ce dossier, nous allons découvrir comment utiliser un tel récepteur DVB aussi bien sous Windows qu'avec GNU/Linux pour explorer les ondes et constater qui il y a là, autour de nous, bien des choses qui circulent dans les airs... DB



INSTALLATION ET UTILISATION DE RTLSDR SOUS LINUX

Lorsqu'il s'agit de bidouille, de hack et de détournement de matériel, il est évident qu'un système sort généralement du lot. Celui-ci nécessite tantôt un peu plus d'expérience au niveau technique mais offre une ouverture et une souplesse inégalées. Ce système est GNU/Linux et c'est donc tout naturellement qu'on cherchera à utiliser RTLSDR avec un tel environnement. Les explications qui suivent s'appliqueront aussi bien à GNU/Linux sur PC mais aussi à la Raspberry Pi, par exemple, avec quelques nuances et limitations.

Fort heureusement et contrairement aux affirmations de quelques éditeurs de solutions propriétaires qui n'existent pas sous GNU/Linux, il existe avec ce système une certaine universalité découlant de l'utilisation de standards (dont POSIX). De ce fait, ce qui fonctionne avec une distribution, fonctionne souvent avec une autre et ce, quelque soit la plateforme. Les explications qui vont suivre concernent principalement les distributions Ubuntu et Debian sur PC et par extension Raspbian sur Raspberry Pi. L'utilisateur curieux et un peu souple pourra très rapidement transposer ces indications sur une autre distribution ou une autre carte (comme la BeagleBone Black, la CubieBoard, la nouvelle Atmel SamA5D3 Xplained ou encore l'Arietta d'ACME).

1. CE QUI VA SE PASSER

Si vous n'avez pas l'habitude de manipuler GNU/Linux en ligne de commandes, vous allez devoir rester concentré et faire très attention. Non que tout ceci soit dangereux ou risqué pour votre matériel mais il est aisé de rater une étape et se retrouver le bec dans l'eau. Dans les grandes lignes, ce que nous allons faire consiste en les étapes suivantes :

- Nous assurer que le système voit le matériel et vous donne les permissions adéquates pour jouer avec le matériel librement,
- installer les éléments nécessaires pour créer et installer tout ce dont nous aurons besoin,
- procéder à l'installation du support pour le récepteur RTL-2832U USB et tester son fonctionnement,
- vérifier la bonne marche de l'ensemble en tentant d'écouter simplement une station radio FM locale.

2. CONNEXION DE LA BESTIOLE

Immédiatement après la connexion du *dongle* USB, vous pouvez très facilement voir ce qui a été détecté, pris en charge et fait le concernant grâce à la commande `dmesg` :

```
$ dmesg
[...]
usb 2-3: new high-speed USB device number
116 using ehci-pci
usb 2-3: New USB device found,
idVendor=0bda, idProduct=2838
usb 2-3: New USB device strings: Mfr=1,
Product=2, SerialNumber=0
usb 2-3: Product: RTL2838UHIDIR
usb 2-3: Manufacturer: Realtek
```

On reconnait ici les deux identifiants du périphérique, le *VendorID* et le *ProductID* qui permet d'universellement reconnaître un produit USB. Ces valeurs sont cependant dépendantes du fabricant et les vôtres seront peut-être différentes .

A titre indicatif, voici une liste des identifiants, marques et modèles connus de récepteur DVB-T :

- `0x0bda:0x2832` = RTL2832U générique
- `0x0bda:0x2838` = ezcap USB 2.0 DVB-T/DAB/FM dongle
- `0x0ccd:0x00a9` = Terratec Cinergy T Stick Black (rev 1)
- `0x0ccd:0x00b3` = Terratec NOXON DAB/DAB+ USB dongle (rev 1)
- `0x0ccd:0x00d3` = Terratec Cinergy T Stick RC (Rev.3)
- `0x0ccd:0x00e0` = Terratec NOXON DAB/DAB+ USB dongle (rev 2)
- `0x185b:0x0620` = Compro Videomate U620F
- `0x185b:0x0650` = Compro Videomate U650F
- `0x1f4d:0xb803` = GTek T803
- `0x1f4d:0xc803` = Lifeview LV5TDeluxe
- `0x1b80:0xd3a4` = Twintech UT-40
- `0x1d19:0x1101` = Dexatek DK DVB-T Dongle (Logilink VG0002A)
- `0x1d19:0x1102` = Dexatek DK DVB-T Dongle (MSI DigiVox? mini II V3.0)
- `0x1d19:0x1103` = Dexatek Technology Ltd. DK 5217 DVB-T Dongle
- `0x0458:0x707f` = Genius TVGo DVB-T03 USB dongle (Ver. B)

- `0x1b80:0xd393` = GIGABYTE GT-U7300
- `0x1b80:0xd394` = DIKOM USB-DVBT HD
- `0x1b80:0xd395` = Peak 102569AGPK
- `0x1b80:0xd39d` = SVEON STV20 DVB-T USB & FM

Mais Linux affiche également les informations qu'il a détecté concernant le matériel comme son nom et le constructeur avec ici un `Realtek RTL2838UHIDIR`. S'il n'y a que cela qui s'affiche, tant mieux. Ceci signifie que le système n'est pas allé plus loin en essayant de prendre en charge lui-même le matériel en chargeant un module. En effet, nous ne voulons pas qu'un pilote se charge pour que nous puissions regarder la TV, ça n'a pas vraiment d'intérêt à part se détruire les neurones par centaines, nous avons mieux à faire.

Si vous voyez quelque chose comme ceci, c'est mauvais signe :

```
usbcore: registered new interface driver dvb_usb_rtl28xxu
usb 1-1.2: dvb_usb_v2: 'Realtek RTL2832U reference design'
   error while loading driver (-71)
usb 1-1.2: dvb_usb_v2: 'Realtek RTL2832U reference design'
  successfully deinitialized and disconnected
```

Ceci signifie que le noyau Linux a tenté d'utiliser le pilote `dvb_usb_rtl28xxu` et, qu'en plus, quelque chose n'a pas fonctionné. Débranchez alors le périphérique et jetez un coup d'œil aux pilotes/modules chargés avec un petit `lsmod` :

```
Module                  Size  Used by
dvb_usb_rtl28xxu       14333  0
dvb_usb_v2             14860  1 dvb_usb_rtl28xxu
rc_core                16313  2 dvb_usb_rtl28xxu,dvb_usb_v2
rtl2830                 7378  1 dvb_usb_rtl28xxu
dvb_core               92797  2 rtl2830,dvb_usb_v2
[...]
```

Et voilà le problème ! Il y a effectivement un module et ses petits copains qui ont été chargés parce qu'un matériel, semble-t-il compatible, a été détecté. Cet exemple d'erreur provient d'une Raspberry Pi sous Raspbian et ceci se corrige assez facilement, en particulier lorsqu'on utilise la méthode barbare : modifiez la configuration et redémarrez.

Dans ce cas, il faut modifier le fichier `/etc/modprobe.d/raspi-blacklist.conf` (avec, par exemple `sudo nano` suivi du nom du fichier) et ajouter les lignes suivantes :

```
blacklist dvb_usb_rtl28xxu
blacklist rtl2832
blacklist rtl2830
```

Ceci aura pour effet d'interdire au système de charger automatiquement ces modules et nous laissera gérer tout cela par nous-même.

Notez que seule la ligne avec `dvb_usb_rtl28xxu` est effectivement utile ici, puisque c'est le module qui a été automatiquement chargé. Cependant, comme il existe plusieurs modèles différents de *dongle* DVB-T, autant interdire tous les modules correspondant dans la foulée.

Sur une distribution GNU/Linux sur PC, la même chose peut se produire. Comme il n'y a pas de `raspi-blacklist.conf`, il vous suffit de créer un fichier d'un nom intelligent, terminant par `.conf` dans le répertoire `/etc/modprobe.d/`, contenant les mêmes lignes. L'effet sera le même.

Pour désinstaller de la mémoire ce qui y a été chargé automatiquement, nous allons redémarrer le système (on a dit « à la barbare »). Mais avant cela, nous allons également spécifier au système qu'il doit nous laisser utiliser le périphérique sans autorisation particulière. Linux est très pointilleux sur la sécurité et par défaut, un simple utilisateur n'a pas le droit d'accéder au matériel directement. Seul le super-utilisateur `root` peut faire cela. Pour configurer ce point particulier, il faut créer un fichier `/etc/udev/rules.d/99_rtlsdr.rules`.

Sans entrer dans le détail, udev est un mécanisme qui permet de définir ce qui se passe lorsque certains événements surviennent. En créant ce fichier qui devra contenir la ligne suivante (sur une seule ligne) nous spécifions que n'importe qui peut lire et écrire sur ce périphérique et qu'il appartient au groupe `adm` :

```
SUBSYSTEM=="usb",
ATTRS{idVendor}=="0bda",
ATTRS{idProduct}=="2838",
MODE="0666",GROUP="adm"
```

Cette modification n'est pas strictement nécessaire. Elle vous évite simplement de devoir utiliser la commande `sudo` à chaque fois que vous lancerez un des outils que nous allons installer. Après cette modification, vous pouvez redémarrer. Si vraiment vous ne souhaitez pas le faire, vous pouvez également décharger les modules avec la commande `rmmod` suivi de leur nom et recharger la configuration udev avec `sudo /etc/init.d/udev restart`.

Après redémarrage, branchez le récepteur DVB-T et scrutez à nouveau la sortie de la commande `dmesg`, il ne devrait plus y avoir d'erreur et avec `lsmod` vous ne devez plus voir de mention du module chargé précédemment.

Une astuce pour vous assurer que tout fonctionne bien est d'ajouter `,SYMLINK+="rtl_sdr"` à la fin de la ligne du fichier udev. Ceci aura pour effet de rajouter un lien nommé `rtl_sdr` dans le répertoire `/dev` lorsque le périphérique est branché. De plus, ce lien pointera vers le vrai fichier qui permet l'accès au matériel. Après connexion, si udev a bien fait son travail, vous devez donc avoir un `/dev/rtl_sdr` et en utilisant la commande `ls -l` sur ce dernier, vous devez voir l'emplacement (du type `/dev/bus/usb/001/004`) qui, s'il est listé avec `ls -l`, doit faire apparaître les bonnes permissions.

3. ON INSTALLE LES TRUCS QUI VONT BIEN

A ce stade, le périphérique DVB-T est pris en charge, mais pas trop, par le système et est maintenant accessible par les programmes qui permettront d'en faire usage. Je parle en premier lieu, bien entendu, de RTLSDR et ses utilitaires mais l'aventure n'est pas finie. Faire mumuse avec les ondes, ça se mérite.

RTLSDR n'est pas disponible directement pour la distribution Ubuntu, Debian ou Raspbian. Il va falloir le compiler. Pas d'inquiétude, ce n'est pas aussi compliqué qu'il y paraît. Il suffit de suivre quelques étapes simples. Dans un premier temps, nous devons installer ce qui va permettre cette compilation avec :

```
$ sudo apt-get install git cmake \
libusb-1.0-0.dev build-essential
```

Les paquets installés sont les suivants :

- `git` est un outil permettant de récupérer la dernière version du logiciel (c'est bien plus que cela, jetez un oeil à Open Silicium 11, un article complet y est consacré),
- `cmake` est un configurateur de sources chargé de détecter automatiquement des éléments nécessaires pour compiler un programme,
- `libusb-1.0-0.dev` est un ensemble de fichiers permettant aux développeurs de créer des applications qui pilotent les périphériques USB,
- `build-essential` est un méta-paquet qui va installer tout ce qu'il faut de manière générale pour compiler des programmes.

Ceci fait, il est temps de passer aux choses sérieuses. On crée donc un répertoire dans son répertoire personnel et on s'y place :

```
$ mkdir SDR
$ cd SDR
```

Puis on récupère les sources de RTLSDR :

```
$ git clone git://git.osmocom.org/rtl-sdr.git
Cloning into 'rtl-sdr'...
remote: Counting objects: 1582, done.
remote: Compressing objects: 100% (676/676), done.
remote: Total 1582 (delta 1157), reused 1213 (delta 898)
Receiving objects: 100% (1582/1582), 340.30 KiB, done.
Resolving deltas: 100% (1157/1157), done.
```

Comme on ne veut pas tout salir en compilant, on va se placer dans les sources, créer un répertoire `build` et s'y placer :

```
$ cd rtl-sdr
$ mkdir build
$ cd build
```

Nous allons ensuite configurer les sources avec CMake. Celui-ci va tenter de détecter tout ce qui est nécessaire pour compiler RTLSDR et agencer tout cela de manière à ce que tout se passe sans encombre :

```
$ cmake -DCMAKE_INSTALL_PREFIX:PATH=/usr ..
-- The C compiler identification is GNU 4.6.3
-- Check for working C compiler: /usr/bin/gcc
[...]
```

```
-- Using install prefix: /usr
-- Configuring done
-- Generating done
-- Build files have been written to:
   /home/pi/SDR/rtl-sdr/build
```

L'option utilisée ici, souvent absente des tutoriels en ligne, **-DCMAKE_INSTALL_PREFIX**, nous permet de demander à CMake de configurer la compilation pour qu'on puisse tout installer dans le système par la suite, comme n'importe quel programme. Par défaut, l'installation se ferait dans **/usr/local** mais cet emplacement n'est pas toujours pris en charge avec toutes les distributions. En précisant **/usr**, on est sûr de notre coup.

Si tout s'est bien passé, ce qui est normalement le cas, il ne vous reste plus qu'à utiliser la commande **make** qui va provoquer la construction :

```
$ make
Scanning dependencies of target
convenience_static
[  5%] Building C object src/
CMakeFiles/[...]
Linking C static library
libconvenience_static.a
[...]
Linking C static library librtlsdr.a
[100%] Built target rtlsdr_static
```

Cette étape sera plus ou moins longue en fonction de votre système. Sur un PC de bureau, il y en a pour quelques secondes mais sur une Raspberry Pi, vous pouvez compter sur plus de 2 minutes, ce qui ne manque pas de rappeler quelques souvenirs d'un temps où on lançait les compilations avant d'aller se coucher...

Après la fin de la construction, **make** a produit énormément de choses dans le répertoire **src/** dont quelques outils dont le nom débute par **rtl_**. C'est ce que nous cherchions à obtenir. Il ne nous reste plus qu'à installer tout cela avec :

```
$ sudo make install
[  5%] Built target convenience_static
[ 35%] Built target rtlsdr_shared
[ 40%] Built target rtl_adsb
[...]
-- Installing: /usr/bin/rtl_power
-- Removed runtime path from "/usr/
bin/rtl_power"
```

Et voilà, le tour est joué ! Les outils RTLSDR sont installés et le moment de vérité est venu : testons si tout fonctionne. Pour cela, il existe un outil, **rtl_test** qui va communiquer avec le périphérique et interroger le tuner E4000 pour voir sur quelle fréquence il peut se régler :

```
$ rtl_test -t
Found 1 device(s):
  0:  Realtek, RTL2838UHIDIR, SN:

Using device 0: Generic RTL2832U OEM
Found Elonics E4000 tuner
Supported gain values (14): -1.0 1.5 4.0 6.5 9.0
11.5 14.0 16.5 19.0 21.5 24.0 29.0 34.0 42.0
Sampling at 2048000 S/s.
Benchmarking E4000 PLL...
[E4K] PLL not locked for 51000000 Hz!
[E4K] PLL not locked for 2214000000 Hz!
[E4K] PLL not locked for 1108000000 Hz!
[E4K] PLL not locked for 1245000000 Hz!
E4K range: 52 to 2213 MHz
E4K L-band gap: 1108 to 1245 MHz
```

Nous avons ici affaire à un tuner Elonics E4000 qui accepte de se caler sur des fréquences allant de 52 Mhz à 2213 MHz avec un « trou » de 1108 Mhz à 1245 MHz. Il ne sera pas possible d'écouter ces fréquences. C'est quelque chose de normal pour le E4000. Les récepteurs DVB-T USB à base de RTL2832U vendus actuellement, utilisant un R280T, n'ont pas de « trou » (mais tout le monde n'est pas encore d'accord pour décider si l'un est mieux que l'autre ou non).

Mais peut-être n'avez-vous pas eu de chance et votre écran s'est monstrueusement remplis de messages d'insulte du type :

```
rtlsdr_demod_write_reg failed with -4
rtlsdr_demod_read_reg failed with -4
rtlsdr_demod_write_reg failed with -4
rtlsdr_demod_read_reg failed with -4
```

Ceci peut potentiellement arriver avec la Raspberry Pi et à cela il peut y avoir plusieurs raisons :

- Vous devriez brancher le récepteur DVB-T via un hub USB alimenté et non directement sur le port de la carte. C'est un problème connu des Pi qui, pour certains modèles ont des petits problèmes lorsqu'il s'agit d'alimenter des périphériques USB gourmands en énergie.

- Vous utilisez la Pi au travers d'une liaison réseau de type SSH. L'interface Ethernet de la Raspberry Pi est en réalité un adaptateur USB/Ethernet et le contrôleur USB est connu pour être de triste facture. En utilisant les deux périphériques en même temps, vous avez peut-être surchargé la bande passante sur le bus et le récepteur met trop de temps à répondre (généralement les leds FDX et LNK font alors n'importe quoi). Essayez d'utiliser la Pi directement, via un écran/clavier ou une liaison série.
- Vous avez overclocké la Pi et vous l'avez poussée à ses limites, ce qui provoque des bugs. Revenez à une configuration normale en utilisant `raspi-config` et en redémarrant.
- Vous avez un demi-million de choses qui traînent en mémoire (genre un serveur Minecraft) et la pauvre carte est incapable de suivre la demande (vous êtes un monstre sans aucune pitié et votre Pi à présent vous déteste).
- Vous êtes tombés sur un récepteur défectueux et vous devriez vérifier en le testant sur un PC pour vous assurer que là, il fonctionne.

Soyons honnêtes, la Raspberry Pi et la faible qualité de son contrôleur USB n'ont pas vraiment les reins assez solides pour ce genre de choses. Nous en avons testé plusieurs et il est arrivé fréquemment que nous ayons des problèmes de stabilité et ce, alors que les récepteurs DVB-T fonctionnaient à merveille sur PC ou avec une BeagleBone Black ou une CubieBoard de première génération. De nombreux utilisateurs ont eu ce problème et certains ont réussi à le contourner en utilisant un noyau Linux plus ancien, une LibUSB plus récente, en utilisant un autre câble USB (blindé), un autre hub USB ou en bidouillant sauvagement le port USB pour l'alimenter avec une tension plus stable. Dans une grande partie des cas cependant, ils se sont finalement rabattus vers leur PC, faute de fonctionnement suffisamment stable.

4. TESTONS !

Il est maintenant temps de procéder à un vrai test en recevant une véritable émission et démodulant le signal. Rien de bien complexe ici, RTLSDR met à disposition un outil qui vous évite de devoir utiliser un logiciel qui nécessite plus de technicité comme GNU Radio. `rtl_fm` nous permet, en ligne de commandes, de spécifier une fréquence et un type de modulation. Le signal démodulé peut alors ensuite être renvoyé à un autre programme pour traitement.

Pour notre premier essai, cet autre programme sera `aplay`, un outil livré avec le système audio de Linux appelé ALSA. L'objectif est d'utiliser le signal démodulé et de simplement l'envoyer au périphérique audio pour que nous puissions l'entendre. Voici la commande magique :

```
$ rtl_fm -f 91.0e6 -M wbfm -s 200000 \
-r 48000 - | aplay -r 48k -f S16_LE

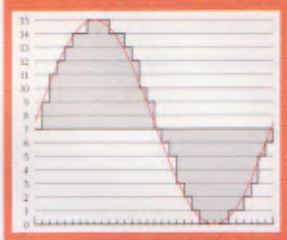
Found 1 device(s):
  0:  Realtek, RTL2838UHIDIR, SN:

Using device 0: Generic RTL2832U OEM
Found Elonics E4000 tuner
Tuner gain set to automatic.
Tuned to 91316000 Hz.
Oversampling input by: 6x.
Oversampling output by: 1x.
Buffer size: 6.83ms
Sampling at 1200000 S/s.
Output at 200000 Hz.
Lecture données brutes 'stdin' : Signed 16 bit\
Little Endian, Fréquence 48000 Hz, Mono
```

L'option `-f` nous permet de préciser la fréquence à écouter, ici 91 Mhz. La syntaxe `91.0e6` utilise la notation scientifique et signifie 91*10^6 soit 91000000 hz. Vous pouvez tout aussi bien spécifier la valeur sous cette forme longue. `-M` précise le type de modulation entre `fm`, `wbfm`, `raw`, `am`, `usb` et `lsb` qui corresponde à des types de modulation (sauf `raw` qui passe les données brutes). Nous utilisons ici une démodulation WFM comme FM *Wideband* car c'est le type de modulation utilisé pour « la radio ». `-s` et `-r` correspondent à des fréquences d'échantillonnage et de ré-échantillonnage nous permettant à la fois d'avoir assez de données pour la démodulation et pour renvoyer le résultat sous une forme intelligible par `aplay`. Enfin, via un `|` et en s'abstenant de préciser un nom de fichier (remplacé par un simple `-`), nous redirigeons le résultat vers `aplay` qui, grâce aux options spécifiées (échantillonnage et format), va prendre ces données et les envoyer au périphérique audio du PC ou de la Raspberry Pi.

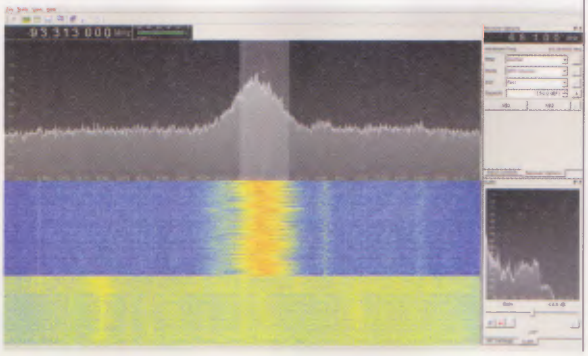
ÉCHANTILLONNAGE ?

Un ordinateur est incapable de manipuler des données analogiques, il ne sait traiter que du numérique. Une onde ou un signal sonore par exemple pour être pris en compte doit être découper en petits morceaux ou échantillons. Chaque tranche est alors mesurée. C'est cette succession de valeurs résultantes qui est utilisée par l'ordinateur.



La fréquence ou cadence d'échantillonnage correspond au nombre de tranches faites dans un laps de temps donné. On parle également de quantification avec les systèmes informatiques car la mesure est numérique et non analogique. La mesure des échantillons correspond à un choix dans une liste finie de valeurs. Ainsi un ordinateur peut, par exemple, mesurer les échantillons sur 8 bits et donc entre -128 et 127. Le fait de mesurer une valeur analogique et de la faire correspondre à l'une de ces valeurs est la quantification.

Le logiciel Gqrx sous Linux permet de recevoir une quantité impressionnante de signaux à l'aide d'un récepteur supporté par RTLSDR. Ici, on distingue clairement une émission de la bande radio FM sur le graphique FFT et il suffit de sélectionner les bons paramètres pour que le signal soit démodulé et que le son se fasse entendre

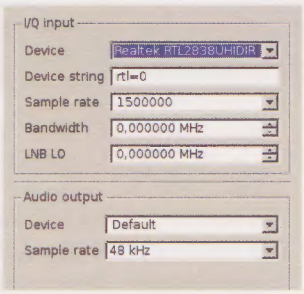


On voit alors dans la sortie à l'écran que `rtl_fm` va caler le tuner sur la fréquence et commencer à démoduler le signal. Si on est effectivement sur une fréquence d'une radio FM, le son de cette dernière va alors se faire entendre sur les haut-parleurs ou dans le casque. Nous venons de démoduler notre premier signal et ce, de manière entièrement logicielle. Bienvenu dans le monde de la SDR !

5. UN PEU PLUS LOIN SUR PC

La Raspberry Pi ne dispose pas de paquet permettant de faire fonctionner l'un des logiciels SDR les plus populaires sous Linux. Sur PC en revanche, celui-ci est disponible : Gqrx. Une simple installation via le système de gestion de paquets vous permettra de disposer d'un logiciel capable de piloter le récepteur et de vous fournir une interface plus simple à utiliser que la ligne de commandes.

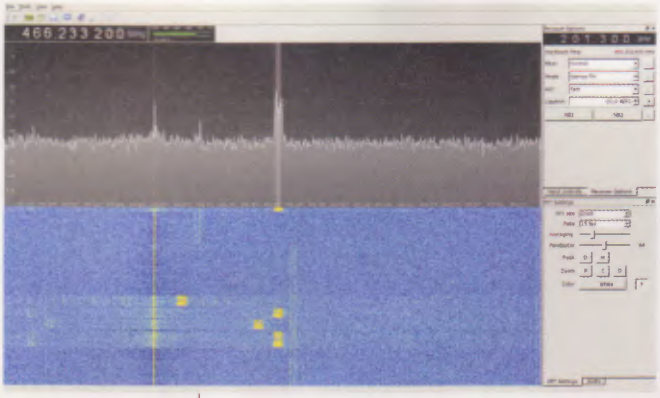
Lors de son premier lancement, Gqrx vous présentera une fenêtre vous demandant de choisir le matériel que vous souhaitez utiliser. Vous devez normalement avoir dans la liste le *dongle* Realtek et il vous suffira de le sélectionner en laissant les autres valeurs par dé*faut*.



L'interface qui se présente après validation est typique des logiciels de SDR. Vous trouvez en haut, la barre de menu et d'icônes et non loin de là, le réglage de la fréquence que vous souhaitez « écouter ». Avec Gqrx, ce réglage se fait en positionnant simplement le pointeur de la souris et en utilisant les flèches de direction pour régler, chiffre par chiffre.

La majeure partie de l'interface présente en haut une visualisation des signaux reçus dans le domaine fréquentiel. Ce graphique, issue d'une transformée de Fourier rapide (FFT comme *Fast Fourier Transform*) permet de voir les fréquences en présence autour de la fréquence sur laquelle est réglée le récepteur. Nous avons donc une plage complète nous présentant dans l'axe X les fréquences et en Y la puissance du signal. Ceci permet donc d'avoir une idée relativement précise des émissions captées et de simplement cliquer pour ajuster la réception.

La partie inférieure affiche les mêmes informations mais avec une composante temporelle. C'est ce qu'on appel un graphique *waterfall*. Nous avons là aussi les



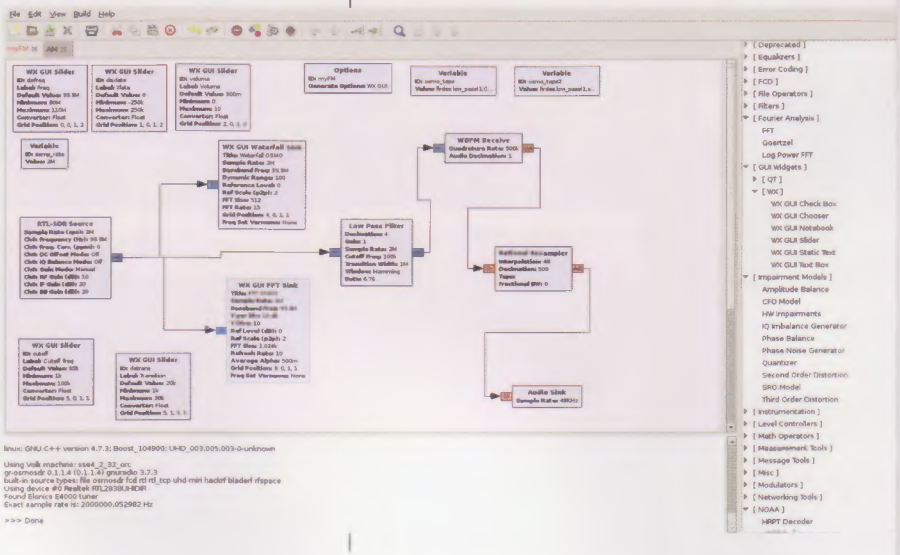
Parcourir les gammes de fréquences à la recherche de signaux intéressants est un passe-temps très amusant. Certaines fréquences sont connues pour véhiculer des informations numériques qu'il sera possible de décoder par ailleurs. Sur cette capture, le graphique supérieur montre un signal relativement fort entrain d'être reçu. En bas, on remarque que le même type d'émission apparaît sur d'autres fréquences proches au fil du temps. Il s'agit de signaux de radiomessagerie utilisant le protocole POCSAG.

fréquences sur l'axe X mais le graphique s'écoule de haut en bas, avec la puissance des émissions représentées par des couleurs. Les émissions, dans le cas de la radio FM sont presque constantes, il y a donc toujours quelque chose qui est capté et ceci peut très facilement se voir sur le graphique FFT. Mais toutes les émissions ne se présentent pas de la sorte, certaines sont intermittentes et ce type de représentation permettra de les repérer facilement. De plus certains motifs caractéristiques vous permettront également d'en déduire la modulation utilisée.

Gqrx est un logiciel très complet et surtout en pleine évolution. Il représente, sous Linux, l'une des solutions les plus simples d'utilisation, idéale pour se faire une idée du domaine et découvrir la discipline. On pourra ensuite passer à quelque chose de plus complexe comme GNU Radio, permettant littéralement de composer son récepteur sur la base des signaux reçus par le matériel USB. Ceci demande cependant un certain investissement en temps et en énergie puisqu'il faut se familiariser avec le jargon, les techniques et les principes du traitement de signal. En revanche, ceci constitue un terrain d'expérimentation idéal et à faible coût si vous comptez préparer votre examen de licence radio-amateur.

6. ET LA SUITE ?

La suite c'est l'exploration ! Ce qu'il vous faut c'est de la patience, du temps et de la curiosité. Un casque audio de bonne qualité sera également fort pratique pour ne pas déranger votre voisinage proche par de longues séances passées à guetter un signal. Il est possible de capter énormément de choses, de la radio FM bien entendu aux balises marines en passant par les communications des radioamateurs, les systèmes de radiomessagerie, l'aviation, le système de synchronisation d'horloge DCF77 et même des communications de la station spatiale internationale. Ce dernier point n'est pas chose facile mais c'est possible et de nombreux utilisateurs sont parvenus à recevoir, avec un *dongle* USB à 15 euros et une antenne maison, des voix venant de l'espace. La réception d'émissions numériques provenant de satellites météo (NOAA) représente un défi intéressant à moyen terme mais nous vous recommandons de commencer doucement et ne pas vous fixer d'objectifs trop difficiles à atteindre sans une certaine expérience. DB



Le logiciel libre par excellence pour la SDR, GNU Radio, dispose d'une interface graphique permettant de composer littéralement son système de réception et de démodulation. Ici, un ensemble pour la réception de la radio FM en mono. Chaque bloc accomplie une tâche précise qui, dans la radio matérielle est un circuit particulier. Ceci illustre parfaitement le concept de radio logicielle ou SDR en anglais.

UTILISATION D'UNE RADIO LOGICIELLE SOUS WINDOWS

Ne soyons pas sectaire. Bien que la bidouille informatique rime généralement avec Linux, la majorité des utilisateurs sont sous Windows et les deux logiciels les plus populaires permettant de contrôler un récepteur USB DVB-T et de traiter les signaux le sont aussi. L'installation, cependant n'en sera pas moins technique car exactement le même problème apparaît sur les deux systèmes : l'utilisation « normal » du produit perturbe les opérations mais nous ne sommes pas du genre à nous laisser faire...



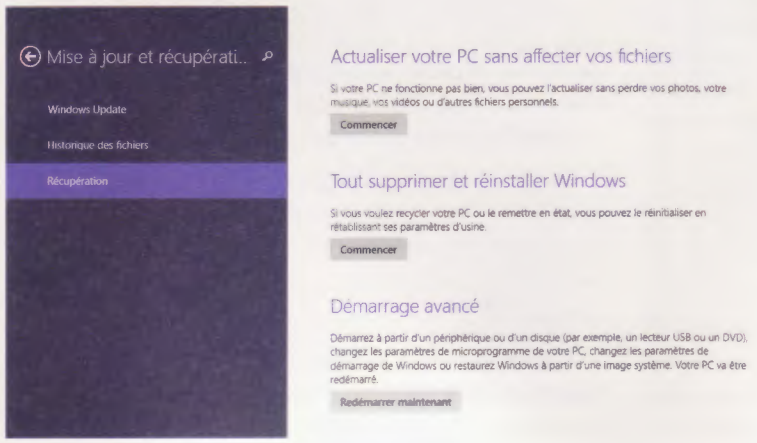
Un périphérique comme un récepteur DVB-T est censé permettre la réception des émissions de télévision numérique terrestre (TNT). Le monde étant ce qu'il est, les constructeurs de ces produits livrent le matériel avec un support logiciel sur CDROM constitué d'un pilote et d'une application de réception. Malheureusement, exactement comme c'est le cas sous Linux, la simple connexion du matériel USB provoque l'installation de pilotes sous Windows, du moins en partie. La principale difficulté consistera donc à se débarrasser des ces pilotes pour ensuite en installer d'autres et enfin procéder à la mise en place des logiciels pour la SDR que sont HDSDR et SDR# (*SDR Sharp*).

1. CONNEXION ET PROBLÈMES DE PILOTE

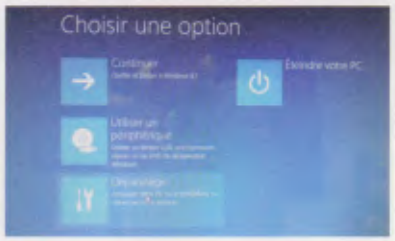
1.1 Étape spéciale Windows 8

Dans les grandes lignes, les techniques d'installation pour Windows 7 et Windows 8 sont relativement identiques. Le problème en revanche qui se pose avec la dernière génération de Windows (8 et 8.1, du moins sous architecture 64 bits) provient des dispositifs de sécurités intégrés et en particulier ceux concernant la signature des éléments logiciels et des pilotes en particulier. Ainsi, pour pouvoir installer la prise en charge SDR du périphérique, il faut avant toute chose désactiver ces sécurités qui interdisent l'utilisation de pilotes non signés.

Pour ce faire, rendez-vous dans la configuration du système depuis l'interface Metro, panneau de droite, en cliquant sur l'icône en forme de roue dentée « Paramètres », puis choisissez « Modifier les paramètres du PC », « Mise à jour et récupération » et « Récupération » :



Dans « Démarrage avancé », cliquez sur « Redémarrer maintenant » et la machine va rebooter dans un mode particulier. Après quelques secondes, un écran se présente à vous :



Sur cet écran, choisissez « Dépannage ». Vous arrivez alors sur l'écran de résolution des problèmes et vous pouvez choisir « Options avancées » :



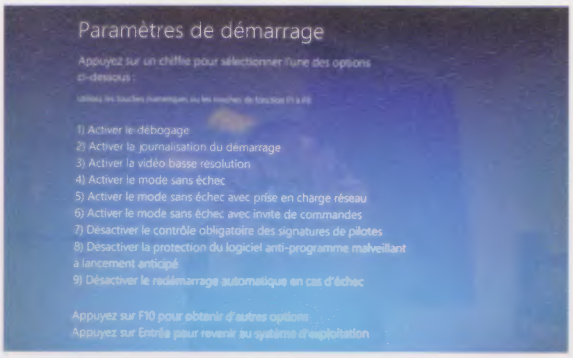
Dès validation, l'écran propose alors plusieurs actions possibles :

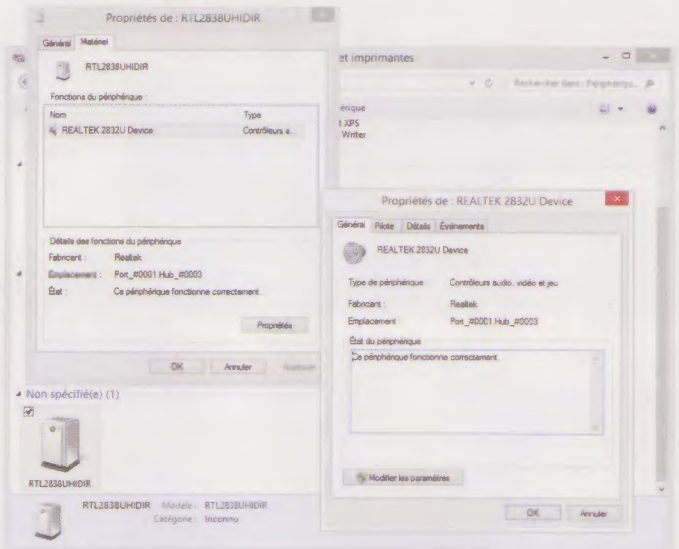


Vous devez choisir « Paramètres » afin de changer le comportement de Windows au démarrage (ce qui n'a plus grand chose à voir avec le dépannage mais c'est la logique Microsoft façon « Démarrer » pour arrêter). En choisissant cette option, vous allez arriver sur un écran qui résume ce que vous allez pouvoir faire après redémarrage (?) :



On remarque alors la mention « Désactiver le contrôle obligatoire des signatures de pilotes », c'est ce qui nous intéresse. Cliquez donc sur « Redémarrer » pour procéder à l'opération. Après quelques secondes, vous devez arriver, enfin, sur le réglage des paramètres de démarrage :





Le périphérique de réception DVB-T tel que vu par Windows 8 et avec un support matériel (pilote) installé automatiquement. Exactement ce que nous ne voulons pas !

Ce sont les touches de fonction F1 à F9 qui correspondent aux chiffres présents dans la liste. Ce qui nous intéresse est en « 7 » et nous appuyons donc sur la touche F7 (méfiez-vous des claviers multimédia disposant des touches accessibles uniquement via un modificateur/bouton type « Fnct »). Il n'y a rien à valider, dès la touche enfoncée, la machine redémarre (pour la troisième fois).

A présent, normalement, votre Windows 8 est en mesure d'accepter l'installation de pilotes non signés.

1.2 Remplacer un pilote par un support WinUSB

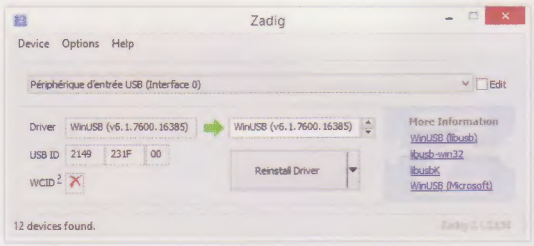
Comme avec de nombreux systèmes d'exploitation, il existe deux manières d'accéder à un périphérique. L'une de ces manières est via l'utilisation d'un pilote, un élément trouvant sa place au cœur du système et disposant de nombreux privilèges, dont celui d'accéder sans retenu au matériel. L'autre solution consiste à accéder au matériel via une application qui fonctionne au même niveau que les différents programmes que vous utilisez régulièrement. En temps normal, les simples applications ne peuvent bénéficier du droit de contrôler directement les périphériques pour des raisons de sécurité et de stabilité. Leur mode de fonctionnement normal consiste donc à parler au système qui utilisera alors les pilotes pour dialoguer avec le matériel.

Ici nous souhaitons que des applications spécifiques puissent accéder librement au récepteur DVB-T mais Windows n'est pas aussi conciliant que Linux sur certain point. En effet, avec ce système il est nécessaire d'associer tout de même un pilote au périphérique de manière à fournir une voie d'accès pour les applications. Il s'agit d'un pilote générique spécial.

Plutôt que de jongler avec les pilotes directement, nous utiliserons un logiciel dédié à cette tâche : Zadig. Celui-ci pourra être télécharger sur **http://zadig.akeo.ie/** et vous n'avez le choix qu'entre deux versions : une pour XP et une autre pour Vista/7/8/8.1.

Connectez votre récepteur USB et après avoir téléchargé le fichier depuis le site, lancez-le. Confirmez le message d'avertissement concernant

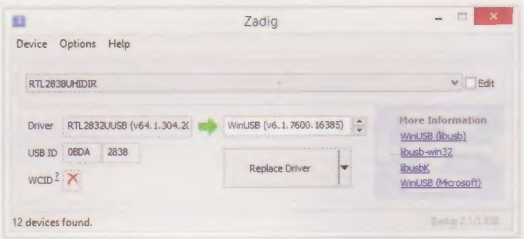
l'autorisation pour le programme d'apporter des modifications au système et vous devez alors voir apparaître l'interface suivante :



Zadig vous permet de choisir un élément dans une liste de périphériques non pris en charge et de lui associer un pilote WinUSB (ou LibUSB) afin qu'une application puisse y accéder directement. Si vous avez annulé l'installation automatique de pilote lors de la connexion ou que vous ne disposiez pas de connexion internet à ce moment, le récepteur sera listé sans rien faire sous un nom ressemblant à « Bulk-In, Interface (Interface 0) » (Win7) ou « Périphérique d'entrée USB (Interface 0) » (Win8). Il vous suffira de choisir cette entrée dans la liste et de cliquer sur « Install Driver » avant de redémarrer Windows.

Si le périphérique n'apparaît pas dans la liste, c'est qu'il est déjà pris en charge par un pilote. Utilisez alors le menu « Options » et choisissez « List All Devices ». Dans la liste vous trouverez alors l'ensemble des périphériques tels que vu par Windows. Repérez le récepteur DVB-T et utilisez le bouton qui se nomme maintenant « Replace Drivers » (remplacer le pilote).

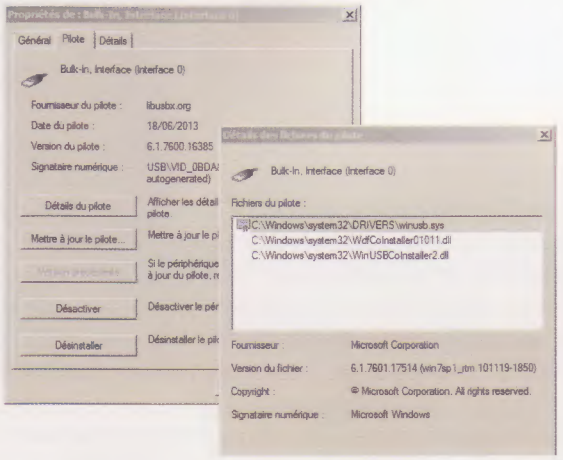
Prenez garde à bien sélectionner le bon périphérique. Notre adaptateur USB de test, utilisé pour cet article, intègre également un récepteur pour la télécommande et celui-ci est désigné par le nom RTL2838UHIDIR :



Il ne s'agit pas du périphérique de réception mais uniquement de l'interface pour la télécommande.

Ce n'est pas ce pilote qu'il faut remplacer. Cherchez « Bulk-In » ou « Périphérique d'entrée USB ». Surtout ne remplacer pas un pilote d'un autre périphérique car celui-ci cessera alors de fonctionner.

Après redémarrage, vous pouvez faire un tour dans le gestionnaire de périphériques pour vous assurer que le pilote associé est bel est bien WinUSB (c'est un Windows 7 ici, mais avec Aéro désactivé) :



Pour être tout à fait honnête, ces manipulations ont été faites sur plusieurs machines. Systématiquement, Windows 7 n'a posé aucun problème et le remplacement du pilote à fonctionné du premier coup dès le redémarrage. Sous Windows 8, en revanche, et pour des raisons que nous ne nous expliquons pas réellement, il nous a fallut procéder à l'installation et à quelques réinstallations avant que l'opération ne réussisse.

Si vous rencontrez des problèmes, en premier lieu assurez-vous de ne pas avoir connecter le récepteur sur un port USB 3. Cela peut être une source de problèmes récurrents. Dans le même ordre d'idée, essayez d'utiliser le matériel seul sur un bus et non en compagnie de tout un tas d'autres périphériques. D'autres pistes possibles sont : les câbles USB, la mise à la masse (surtout les ordinateurs portables), et les hubs qui peuvent impacter le fonctionnement du récepteur.

Et puis, nous avons les problèmes d'installation de pilotes qui relèvent souvent du vaudou ou de la sorcellerie et passent généralement par la désinstallation des pilotes depuis le gestionnaire

de périphériques, un redémarrage et une réinstallation avec Zadig. Notez que Windows associe généralement les pilotes à l'emplacement de connexion. Ainsi, en ayant utilisé Zadig pour remplacer un pilote, il n'est pas sûr que vous ne soyez pas obligé de réitérer la procédure en connectant le récepteur sur un autre port. Toute la magie de Windows en somme...

2. UN PREMIER LOGICIEL : HDSDR

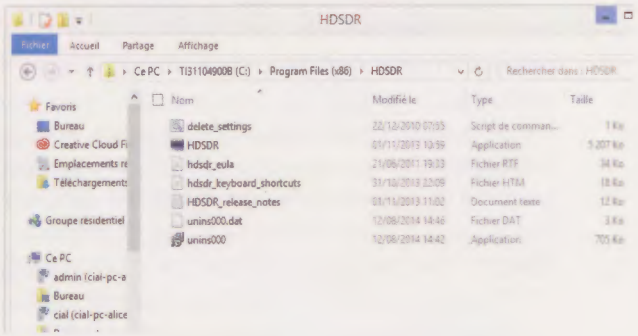
HDSDR pour *High Definition Software Defined Radio* est un logiciel freeware pour Windows 2000/XP/Vista/7/8/8.1. Il ne s'agit pas d'une application spécifiquement dédiée aux périphériques de réception DVB-T mais elle supporte une vaste gamme de matériel et, comme de nombreux outils de SDR, son utilisation principale passe par la réception avec un matériel dédié et une transmission via la carte son. Fort heureusement pour nous, les dernières versions permettent de prendre en charge directement un récepteur USB comme celui faisant l'objet de nos articles. Ceci à condition de passer par une petite étape de bidouille.

Avant toutes choses, pointez votre navigateur sur **http://www.hdsdr.de/** et au bas de la page, cliquez sur « Download » pour récupérer le fichier `HDSDR_install.exe`. Il vous suffira ensuite de double-cliquer sur ce dernier pour démarrer l'installation :

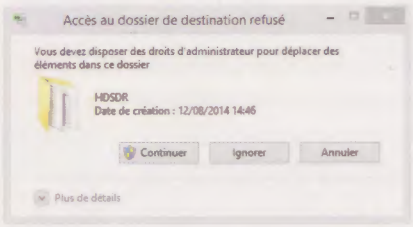


Suivez alors les étapes classiques jusqu'à terminer l'installation mais sans lancer immédiatement le logiciel comme proposé. Retournez alors sur le site, à la section « Hardware ». Dans le tableau contenant la liste du matériel supporté, repérez « RTLSDR (DVB-T/DAB with RTL2832) » et en particulier « DLL » dans la colonne juste à droite. Ce lien normalement vous conduit sur un espace de stockage en ligne (**https://app.box.com/s/7tpiy8r6qo2bbhdxtt4k**) et vous propose de télécharger un fichier `ExtIO_RTL2832.dll`. Il s'agit de la bibliothèque permettant de piloter le récepteur DVB-T. Enregistrez ce fichier n'importe où (sur le bureau par exemple).

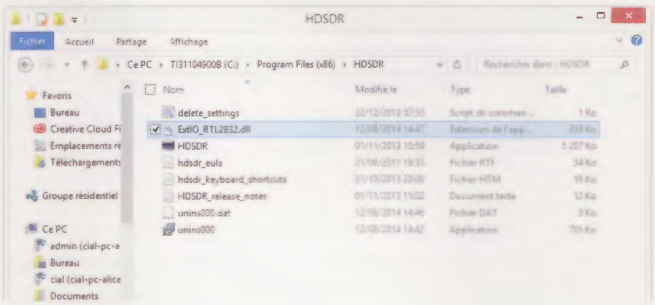
Naviguez ensuite avec l'explorateur Windows sur le disque dur de la machine dans le répertoire `Programmes` (ou `Program Files (x86)` avec un Win8 64 bits) où vous devez trouver un répertoire `HDSDR` :



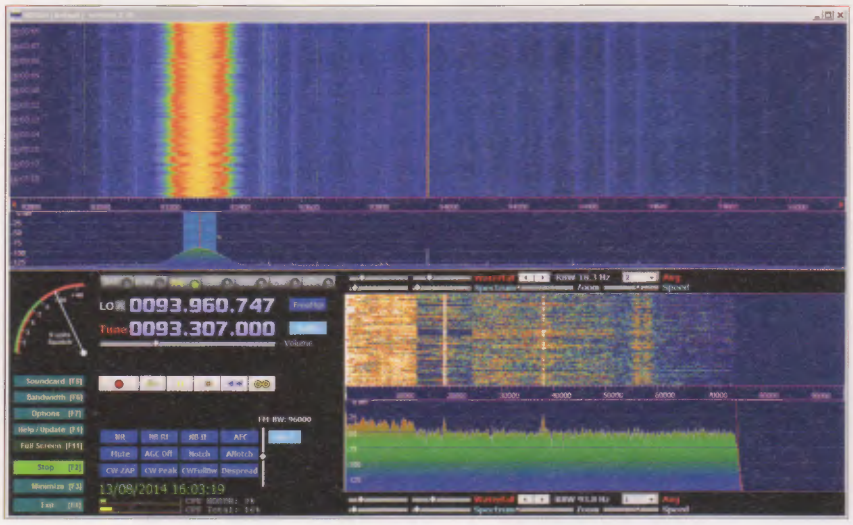
Copiez alors le fichier DLL dans ce répertoire. Un message d'avertissement vous demandant d'utiliser des droits administrateurs s'affichera sans doute, confirmez la manœuvre :



Vous devez obtenir un contenu du répertoire ressemblant à ceci :



Il ne vous reste plus qu'à croiser les doigts et à lancer HDSDR. L'interface du logiciel est assez surprenante pour un utilisateur non coutumier de ce type d'application mais parfaitement classique d'un point de vue de radioamateur :



L'interface de HDSDR est très graphique et comporte plein de petites choses à régler de ci de là. Fort heureusement, l'ensemble est relativement facile à prendre en main pour peu que l'on ait un navigateur web à porté de main pour recherche les termes inconnus.

Avant toutes choses, dans l'interface, cliquez sur « Option » pour vous assurer que dans « Select Input », c'est bel et bien « Generic RTL2832 » qui est actif. Réglez ensuite la fréquence « LO » avec la molette de la souris puis cliquez sur « Start ». Les données vont normalement immédiatement arriver et être traitées. Le graphique supérieur, dit *waterfall*, va se remplir. Il représente exactement la même chose que celui présenté immédiatement en dessous, le visualisateur FFT, qui présente la bande de fréquence que vous percevez actuellement mais avec un « étalement » dans le temps. Ceci permet de facilement repérer des signaux qui ne sont pas continus.

Réglez la fréquence sur une station radio locale et vous devrez alors littéralement voir dans la vue FFT une bosse. Faites un tour dans « Bandwith » et choisissez la valeur 192000 dans « output ». Juste au dessus du réglage de fréquence se trouve une barre de boutons correspondant aux types de démodulation choisi. Cliquez sur FM. Enfin, ajustez la fréquence en cliquant simplement sur la « bosse ». Normalement vous devez entendre la station radio !

Il ne vous reste plus qu'à explorer le reste de l'interface, les différentes fréquences, types de modulation et ajustements possible. Selon l'endroit où vous vous trouvez, vous tomberez sur des émissions radioamateurs, des balises, les communications maritimes, etc.

3. SDR# : PLUS SIMPLE ET ACTUELLEMENT LE PLUS POPULAIRE

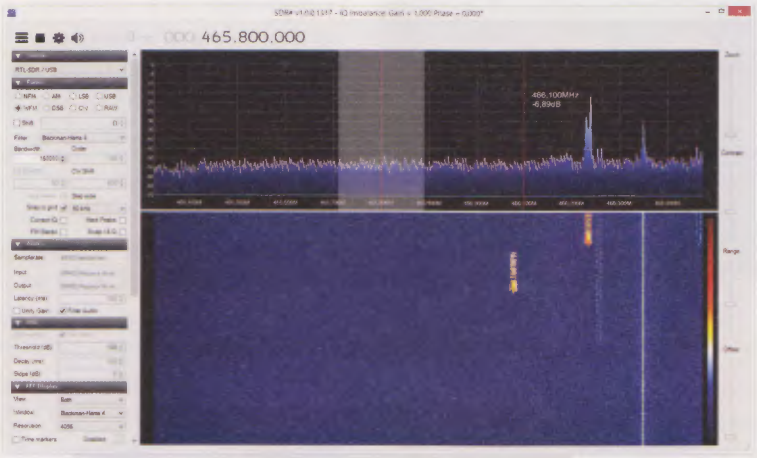
SDR# ou SDR Sharp est l'autre application très utilisée sous Windows. Vous pourrez la télécharger sur le site **http://sdrsharp.com**. Son installation est un peu surprenante mais elle présente l'avantage de ne pas nécessiter de manipulations particulières permettant la prise en charge du récepteur DVB-T.

Ainsi, vous n'avez qu'à télécharger le fichier `sdr-install.zip` en cliquant sur le « Download » de la page d'accueil. Double-cliquez ensuite sur le fichier et glissez-déposez le dossier `sdr-install` qu'il contient sur le bureau avant de fermer la fenêtre.

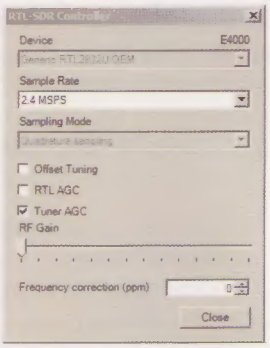
Ce dossier ne contient pas le logiciel lui même mais un script et deux petits programmes qui se chargeront de télécharger les différents éléments. Dans le dossier, double-cliquez simplement sur `install.bat` et brièvement une fenêtre DOS va apparaître avec une longue liste de fichier et terminant en vous demandant d'appuyer sur une touche pour continuer. Après l'avoir fait, la fenêtre disparaît et le dossier est maintenant peuplé de nombreux fichiers et d'un répertoire `sdrsharp`. Dans ce dernier se trouve l'application éponyme. Lancez-là :

SDR# est sans doute bien plus simple à utiliser que HDSDR. Il n'est pas nécessaire d'installer quoi que ce soit d'autre et il vous suffit dans la partie supérieure gauche (« Source ») de choisir le récepteur dont vous disposez : « RTL-SDR / USB ». Appuyez alors simplement sur la grosse icône de lecture dans la barre supérieure.

Comme précédemment, calez la fréquence proche d'une radio FM commerciale et observez le graphique FFT pour repérer la fréquence exacte de l'émission. Cliquez simplement sur la « bosse » caractéristique d'un tel signal et assurez-vous que « WFM » (*Wideband FM*) soit coché juste sous la sélection du matériel. Immédiatement vous devez entendre l'émission radio en question.



Une option intéressante facilement accessible dans SDR# est le contrôle de gain automatique. Le *dongle* de réception DVB-T est en mesure d'augmenter la puissance ou l'amplitude des signaux reçu automatiquement. Il faut savoir qu'en augmentant le gain, en particulier au delà d'une certaine valeur, les avantages sont minimes étant donné qu'on augmente également le bruit. En utilisant la fonction AGC (*Auto Gain Control*) matériel, le tuner ajustera automatiquement le gain à notre place, ce qui donne souvent de bons résultats. Faites un tour dans les paramètres, via l'icône en forme de roue dentée en haut et cochez « Tuner AGC » :



Ceci devrait améliorer sensiblement la qualité des signaux reçu. Il ne vous reste plus qu'à vous amuser à vous promener dans les fréquences pour voir ce qui peut être intéressant à écouter. Comme avec HDSDR vous risquez de tomber sur énormément de choses. L'interface de SDR# joue en sa faveur pour ce type d'exploration car elle est bien plus intuitive pour un débutant et elle rendra donc l'expérience plus amusante.

4. ET ENSUITE ?

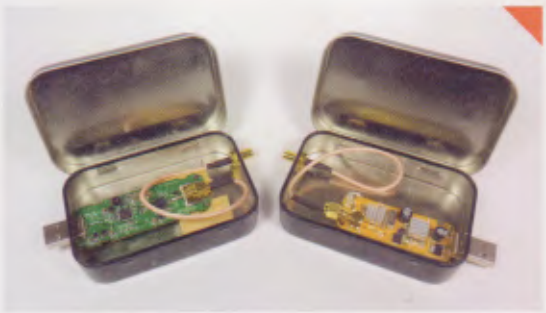
Avec un récepteur USB comme celui mis en oeuvre ici et la modeste antenne qui l'accompagne, il est déjà possible de recevoir un certain nombre d'émissions intéressantes. L'intérêt sera en premier lieu ludique mais au fil du temps, vous assimilerez la technologie et le jargon. Vous prendrez l'habitude de jongler avec les paramètres et sans doute même arriverez assez rapidement à reconnaître dans le graphique *waterfall* le type de modulation en présence.

Si le domaine vous captive, viendra alors le moment d'envisager de passer aux choses sérieuses et de vous pencher sur les antennes. Dans la majorité des cas, l'antenne est l'élément le plus important car en fonction de sa géométrie, ses caractéristiques et dimensions, celle-ci permettra de capter des signaux impossibles à recevoir et traiter autrement.

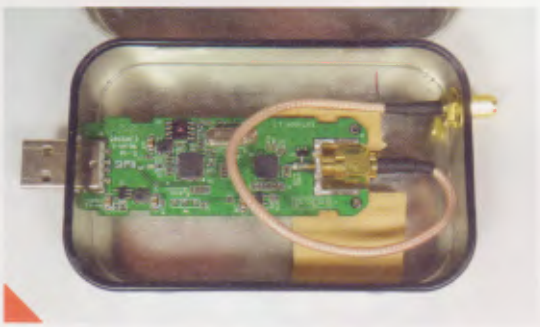
Essayer de vous rapprocher d'un club radioamateur local. Les membres de ce type d'associations sont des makers, des hackers et des bidouilleurs de la première heure avec pour la plupart un niveau technique très important et souvent un sens de l'humour très appréciable. Grâce à vos échanges, vous pourrez alors augmenter votre niveau et sérieusement envisager de passer l'examen pour obtenir la licence qui vous permettra non plus de simplement écouter mais également d'émettre. Et là c'est une toute autre aventure qui commencera pour vous... DB

BONUS : AMÉLIORER LA RÉCEPTION C'EST RÉDUIRE LE BRUIT

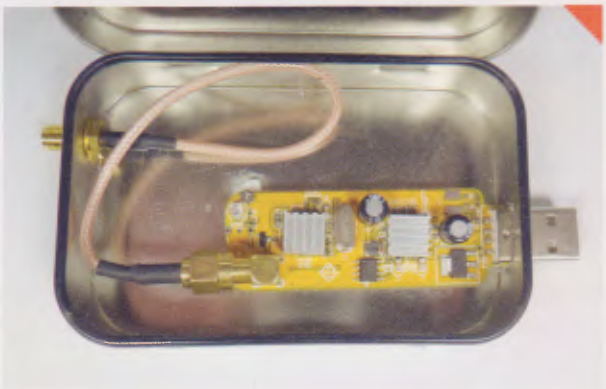
L'ennemi du signal c'est le bruit ! Les récepteurs DVB-T utilisables avec RTLSDR sont très économiques mais ce faible coût se traduit forcément par une qualité de fabrication bien différente de celle des récepteurs dédiés à la SDR comme le FUNcube Dongle, le FDM-S2 ou encore les très coûteux USRP de Ettus Research. Il est cependant possible de procéder à quelques modifications de votre récepteur chinois pour améliorer sensiblement la réception. ■



Vous l'avez peut-être remarqué en faisant vos essais, même sans antenne, un récepteur RTLSDR est en mesure de recevoir des émissions et ce, tout simplement, parce que le circuit lui-même fait office d'antenne. Ceci implique également qu'en utilisant une antenne "maison" pour 433 Mhz par exemple, énormément de bruit sur d'autres fréquences, sera également capté. La solution : la mise en boite !



Voici un récepteur à base de Elonics E4000 débarrassé de son habit inutile de plastique et inséré dans une boite en métal faisant office de blindage (RF shielding) en contact avec le connecteur USB et la masse de l'antenne. Le connecteur d'antenne 9 mm a également été remplacé par un connecteur SMA (à ne pas confondre avec le RP-SMA des antennes Wifi) et le récepteur IR de la télécommande a été dessoudé.



Cette autre récepteur DVT-T, plus récent, avec un tuner Rafael Micro R820T, a curieusement tendance à énormément chauffer et la mise en boite n'arrange rien à cela. Le tuner et le convertisseur RTL2832U ont été équipés de radiateurs (vendu initialement pour une Raspberry Pi). Là encore, le connecteur d'antenne a été remplacé et le récepteur IR a disparu.



En plus de fournir un blindage autour du récepteur ces petits bricolages créatifs permettent de donner une petite touche "geek" au matériel selon la boite utilisée. Bien sûr, la seule chose importante est le fait que l'emballage soit conducteur (idéalement ce serait du cuivre) mais les bonbons caféinés ça va forcément de paire avec des longues nuits d'écoute des ondes...



En dehors du blindage du récepteur, des améliorations importantes peuvent être constatées en s'occupant également du câble USB. Une recommandation dans ce sens consiste à débarrasser le connecteur USB de son blindage. Ceci permet d'éviter que la masse de l'ordinateur et le blindage du câble induise du bruit dans le signal. Une autre astuce consiste à équiper le câble de tores en ferrite comme on en trouve sur certains câbles d'alimentation ou sur de vieux câbles parallèles d'imprimante. Ceci devrait également réduire le bruit et améliorer la réception mais d'après nos essais dans une mesure bien moindre que l'ajout de blindage ou la déconnexion de la masse.

ADS-B OU COMMENT ÉCOUTER LES AVIONS TOUT AUTOUR DE VOUS

D. B.

C'est un bel après midi d'été, le ciel est bleu et il n'y a pas un nuage à l'horizon. Dans un coin du ciel, une traînée blanche et au bout, un tout petit point brillant, c'est un avion de ligne. Savez-vous que ces appareils sont très bavards et ne cessent de hurler leurs positions, leurs vitesses ou encore leurs altitudes ? Tout ce qu'il vous faut pour les entendre c'est un récepteur TNT à 15 euros et les bons logiciels...

L'histoire du contrôle aérien débute avec les premiers vols commerciaux. L'objectif est de gérer le trafic de manière à ce que tout ce petit monde qui évolue dans le ciel ne se rentre pas dedans. Ceci peut paraître peu probable au premier abord, les avions sont petits et le ciel est grand, très grand. Malheureusement, il est bien nécessaire pour ces aéronefs de décoller et d'atterrir en des endroits bien précis et il y a donc, par endroit, des concentrations inquiétantes d'appareils.

Initialement, ce sont les communications radio entre les pilotes et la tour de contrôle qui permettaient d'apporter une réponse à ce problème. L'ensemble était cependant basé entièrement sur une évaluation visuelle de la situation. Ainsi, dès que les conditions météo se dégradaient, il devenait excessivement risqué, sinon suicidaire, de voler en aveugle à proximité d'un aérodrome (mais comme dirait Régis Laspalès « Y'en a qu'ont essayé, ils ont eu des problèmes ! C'est vous qui voyez ! »).

Le radar améliora ensuite grandement la situation. Ce système, fort simple dans son principe de fonctionnement, consiste à envoyer une onde et à écouter un éventuel retour après réflexion sur un objet (comme les chauve-souris, mais en mieux). D'abord militaire et utilisé pour repérer les avions ennemis, l'utilisation du radar à rapidement trouvé sa place dans l'aviation civile permettant ainsi aux tours de contrôles équipées d'avoir une vision d'ensemble de tout ce qui se trouvaient autour d'elles. Il suffisait alors d'informer les pilotes de la situation et de leur donner des instructions précises pour éviter tout problème en jouant littéralement le rôle de chef d'orchestre.

Le radar était une importante avancée pour la sécurité mais il ne réglait pas tous les problèmes. En effet, bien qu'il était possible de voir les appareils et d'en déduire leur position, leur altitude, leur vitesse et leur cap, il fallait tout de même contacter les pilotes pour savoir qui était qui et ainsi mettre des noms sur les petits points à l'écran.

Les transpondeurs ont ensuite fait leur apparition. Là ce n'était plus les pilotes qui s'identifiaient mais les appareils eux-mêmes. Utilisé de concert avec le reste des technologies le niveau de sécurité n'a cessé de croître. L'ajout du GPS permettant aux avions de connaître avec précision leur position à ensuite donné naissance à une autre innovation sur la base d'une constatation simple : l'avion via le GPS connaissait finalement sa position mieux que le système radar au sol, dont la précision est limitée. L'avion pouvait alors lui-même envoyer cette position, en même temps que d'autres informations telles que sa vitesse ou des données permettant de l'identifier clairement. Le système ADS-B était né.

ADS-B pour *Automatic Dependent Surveillance-Broadcast* est un système de surveillance coopératif car ce sont les acteurs eux-mêmes qui coopérativement informent leur environnement de leur situation. Ainsi les stations de contrôle au sol n'ont qu'à recevoir ces informations et les mettre en relation avec les autres données en leur possession pour obtenir une image fidèle du ciel environnant. L'ADS-B repose sur plusieurs technologies mais les choix sont généralement dépendants des contraintes techniques et financières.

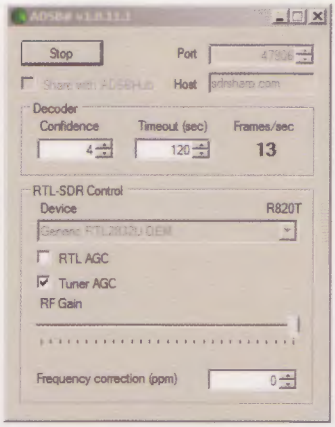
Un autre système de sécurité existe et est en place depuis longtemps, c'est TCAS pour *Traffic Collision Avoidance System*. Comme son nom l'indique TCAS a pour objectif d'éviter les collisions en vol. Il permet d'avertir un pilote qu'il se trouve sur un trajet qui risque de découler sur une collision avec un autre appareil. En cas de problème et d'incapacité du pilote à contrôler son appareil, le système TCAS peut éventuellement prendre les mesures qui s'imposent pour éviter l'accident en changeant de cap ou d'altitude.

Les systèmes TCAS embarqués peuvent être étendus et évoluer pour prendre en charge ADS-B, c'est l'extension ES1090 qui utilise alors la fréquence 1090 Mhz. TCAS équipant tous les avions de transport commercial à turbine de plus de 19 sièges passagers depuis 2005 (30 sièges depuis 2000), l'ajout d'ES1090 est la solution la moins coûteuse. Pour les appareils plus petits cependant, l'intégration d'ES1090 à partir de zéro s'avère trop chère. Une autre liaison est alors utilisée, l'UAT (*Universal Access Transponder*) ou la VDL mode 4 (*VHF Data Link Mode 4*). Là ce seront respectivement les fréquences 978 MHz et la bande 108-118 MHz qui seront utilisés.

Ce qui nous intéressera ici, c'est l'ES1090 et donc l'aviation de transport commercial. En effet, ADS-B fonctionne en mode diffusion (*broadcast*) et les données sont transmises en continu, pour qui veut bien les entendre et les décoder. J'ai bien dis « décoder » et non « déchiffrer », car tout ceci est diffusé « en clair », sans aucune protection, mais nous en reparlerons en fin d'article. Pour l'heure, place à la démonstration !

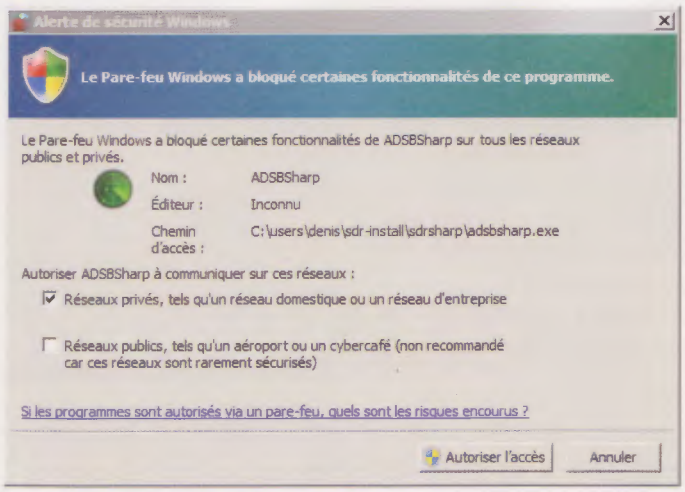
1. SOUS WINDOWS

Sous Windows, en installant SDR# (cf article précédent dans ce numéro), on obtient également l'application ADSB# (`ADSBSharp.exe`) se trouvant dans le même dossier. Il vous suffira de lancer cette application pour obtenir la fenêtre suivante :



Comme avec SDR#, on retrouve la sélection du matériel (*Generic RTL2832U*) et les options permettant entre autres choses d'activer le réglage automatique du gain.

Cette application ne vous présentera pas directement les données mais fonctionne en tant que serveur. Ceci signifie qu'en cliquant sur le bouton « Start », un serveur sera accessible sur votre machine, sur le port spécifié (47806 par défaut). Notez d'ailleurs que lorsque vous cliquez pour la première fois, il est fort probable que le pare-feu Windows s'en mêle :



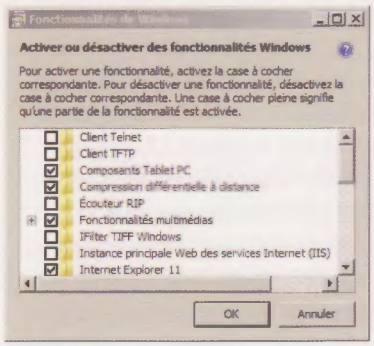
Vous devez alors confirmer que l'application ADSBSharp peut effectivement attendre des connexions en cliquant sur « Autoriser l'accès ». Ensuite, il ne se passe plus rien, semble-t-il. Le seul indicateur d'une quelconque activité est le chiffre « Frames/sec » présent dans la fenêtre qui indique le nombre de messages ADS-B décodés par seconde. En fonction de votre proximité géographique avec un couloir aérien ou un aéroport, cette valeur peut ou non grimper rapidement.

Pour vérifier le bon fonctionnement de l'application, le plus simple est de tout bonnement pointer votre navigateur web sur l'adresse 127.0.0.1:47806. Vous devriez alors voir apparaître une longue liste de lignes ressemblant à :

```
*04000000000000;
*08000000000000;
*67C3DACC601DA7;
*0F800000000000;
*1E83F3EE27020F;
*40000000000000;
*0FE7CC3FF11870;
*09F77001863979;
*38000000000000;
*18000000000000;
*27308FC1F87D3C;
*1B01F8E0C00C20;
```

Il n'est pas certain que votre navigateur puisse effectivement afficher les données. Il est préférable d'utiliser un outil adapté, qui en plus pourra

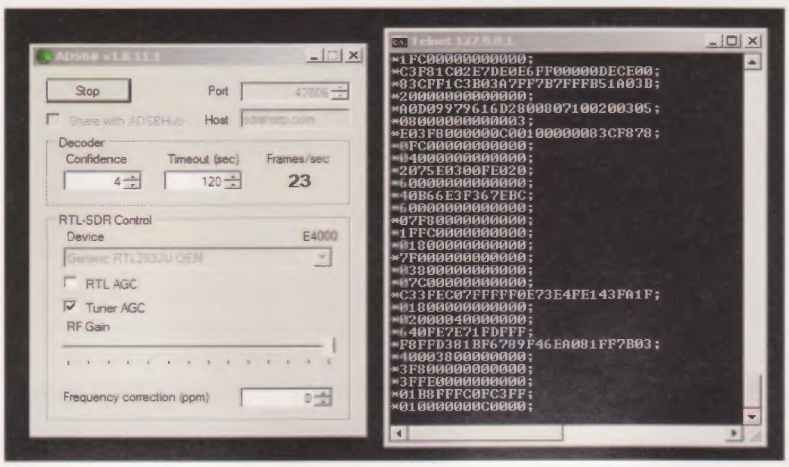
servir pour d'autres manipulations par ailleurs. Rendez-vous donc dans le panneau de configuration, à la section « Programmes » et « Désinstaller un programme ». Vous arrivez sur la longue liste des applications installées dans votre Windows. Sur la droite, cliquez sur « Activer ou désactiver des fonctionnalités Windows ». La fenêtre qui s'ouvre vous présente une liste de « fonctionnalités » :



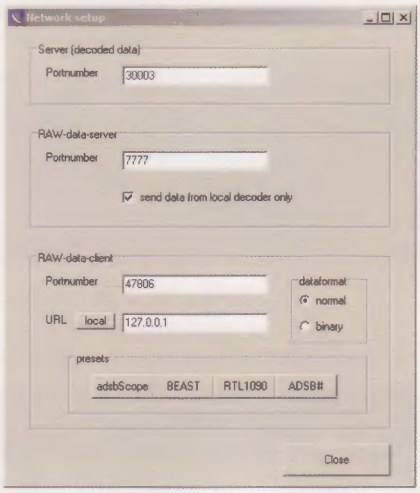
Ce qui nous intéresse est « Client Telnet » tout en haut de la liste. Cochez la case correspondante et cliquez sur « Ok ». Une fois la modification terminée, il ne vous reste plus qu'à lancer l'émulateur de terminal `cmd.exe` et d'y taper `telnet 127.0.0.1 47806` pour obtenir quelque chose comme :

Ce que vous voyez défiler, ce sont les messages issus de la réception de signaux ADS-B démodulés par ADSB#. Derrière ce charabia, se cachent des informations très intéressantes mais encore faut-il pouvoir les interpréter. Il est possible de décoder cela à la main mais je pense que vous serez d'accord pour confier cette tâche fastidieuse à votre ordinateur.

Pour cela, nous devons installer une application appelée adsbSCOPE, créée par un développeur et électronicien allemand (*sprut*) pour son propre projet de décodage ADS-B (sur Microchip PIC18F2550). Rendez-vous à l'adresse **http://www.sprut.de/electronic/pic/projekte/adsb/adsb_en.html**. Là, derrière cette page à l'aspect peu engageant mais d'un style tout à fait courant pour un site d'électronicien radioamateur, vous trouverez le logiciel qu'il vous faut sous la forme d'un lien vers le fichier `adsb_all.zip` (tout en bas de la page). Téléchargez donc ce fichier et ouvrez-le.



Vous y trouverez un dossier `pc_software/adbscope/27` dont le contenu pourra être copié, par exemple, dans un répertoire `adsbscope` sur le bureau ou dans `C:\Programmes`. Ceci fait, lancez le programme `adsbscope27_256.exe` et allez directement dans le menu « other », « Network » et « Network setup » :



Dans cette fenêtre de configuration, ce qui nous intéresse est la partie inférieure référencée par « RAW-data-client ». L'objectif est de configurer adsbSCOPE de manière à ce qu'il se connecte à notre serveur ADSB# pour recevoir les données et les utiliser. Spécifiez alors 47806 pour « Portnumber », 127.0.0.1 pour « URL », et « normal » pour « dataformat ». Cliquez sur « Close » pour terminer la configuration.

Tout est prêt, nous pouvons y allez. Utilisez le menu « other », « Network » et « RAW-dara Client active » ce qui connectera le logiciel à la source de données. Vous n'avez plus qu'à attendre. Parmi la masse de messages ADS-B qui

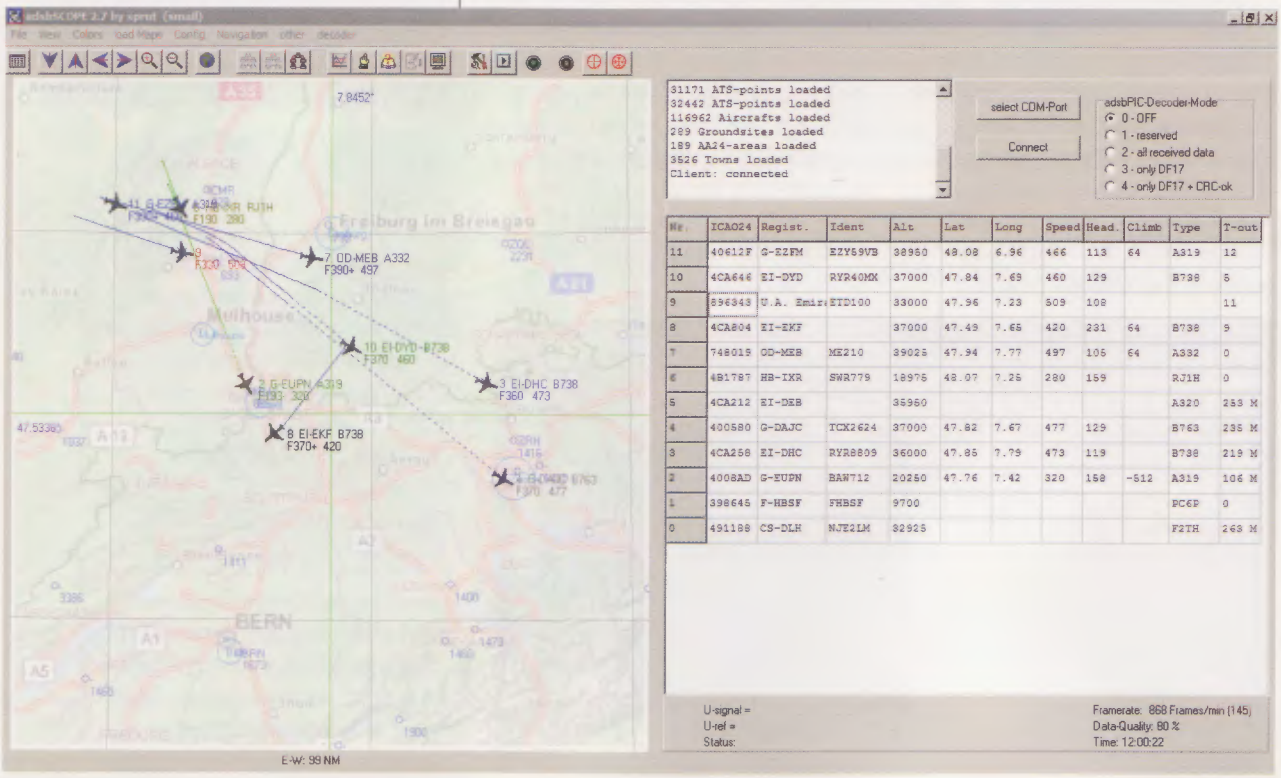
arrivent, tous ne sont pas utilisables et/ou ne contiennent pas toutes les données utiles. Mais selon la fréquentation du ciel à proximité, assez rapidement, un tableau sur la droite devrait se remplir.

Utilisez alors les fonctions de zoom/dézoom pour repérer les avions détectés qui signalent leurs positions et centrez la carte. Utilisez éventuellement le bouton représentant une planète pour ajouter un fond cartographique. Vous devez obtenir quelque chose ressemblant à la capture présentée ici.

En dehors de la carte qui est très impressionnante et qui voit les avions placés en fonction des coordonnées longitude/latitude qu'ils annoncent, nous avons également droit à quelques données captivantes à commencer par le modèle d'avion avec ici un Airbus A319, A320 ou encore A332 et quelques Boeing (B738). Mais plus intéressant encore, nous avons le numéro de vol nous permettant de nous renseigner sur sa destination (tapez simplement l'identifiant dans Google et vous saurez tout de lui) ou encore sa vitesse et son altitude.

Notez au passage que ces deux valeurs sont respectivement données en nœud et en pieds. Dans la capture prise pour cet article nous avons par exemple le vol ME210 (Middle East Airlines vol n°210 (et non Messerschmitt Me 210)) parti de Paris et se rendant à Beyrouth au Liban. L'appareil est un Airbus A332 volant à 497 nœuds soit quelques 920 km/h et se trouvant à une altitude de 39025 pieds, presque 12000 mètres !

Avec un récepteur TNT, ADSB# et adsbSCOPE, vous pouvez visualiser directement les avions autour de vous dans un rayon assez vaste. Ici le récepteur est à Sélestat (67) et nous sommes arrivés, avec une antenne de bien triste facture et depuis l'intérieur des bureaux de la rédaction, à recevoir des émissions d'avions atterrissant à Bâle (Suisse), quelques 100 km plus loin...



Je ne sais pas ce que vous en pensez mais personnellement je trouve cela tout simplement merveilleux. Les avions que nous voyons dans le ciel ne sont plus simplement de stupides petits points traçant de grand trait blanc mais prennent maintenant une toute autre forme. Peut-être comme moi, passerez-vous un certain temps les yeux sur l'écran et le nez pointé vers le ciel (ce qui n'est physiquement pas facile) et vous étonnerez-vous des folles valeurs que vous lirez telle la vitesse, la distance ou encore la destination de chacun de ces appareils...

2. SOUS GNU/LINUX

Une fois n'est pas coutume, si vous êtes utilisateur à la fois de Windows et d'un système GNU/Linux, vous allez vous rendre compte que les choses sont bien plus simples avec le système que tout le monde pense plus difficile à utiliser. En effet, sous Linux, tout ce qu'il vous faut (en dehors de l'installation de RTLSDR) c'est installer un seul et unique logiciel : Dump1090.

Celui-ci cependant n'est pas disponible sous une forme immédiatement installable (paquet). Vous devrez donc le construire à partir de ses sources. Pointez alors votre navigateur sur **https://github.com/antirez/dump1090** et téléchargez simplement le fichier Zip puis décompressez-le. Vous obtiendrez alors un répertoire `dump1090-master` et le reste se passera dans un terminal.

Nous partirons ici du principe que vous avez décompresser le fichier dans votre répertoire personnel. Nous entrons donc, tout d'abord, dans le répertoire puis lançons directement la compilation avec :

```
$ cd dump1090-master
$ make
```

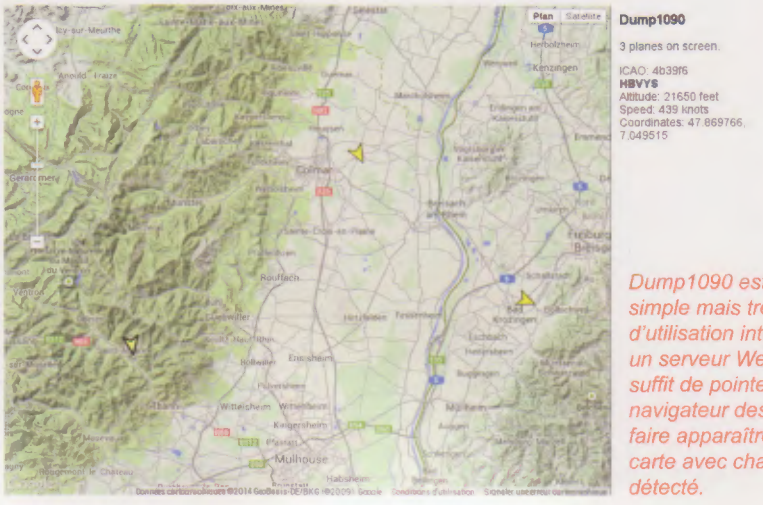
Si vous avez précédemment installé RTDSDR sur votre système, il ne devrait pas y avoir le moindre problème pour cette étape et vous retrouverez alors le programme `dump1090` dans le répertoire courant. Exécutez alors ce programme en utilisant :

```
$ ./dump1090 --metric --interactive --net
```

Les options utilisées ici permettent respectivement d'utiliser le système métrique pour l'affichage des données, d'afficher les données décodées en temps réel et d'activer les fonctions réseau. Sur l'écran s'affichera alors quelque chose comme :

```
Hex    Flight   Altitude Speed Lat      Lon    Track  Messages Seen .
--------------------------------------------------------------------
4b17a3 SWR76R   6709     687   47.815   7.602  338    71       0 sec
4b7f4c          7173     0     0.000    0.000  0      80       1 sec
4692ca          10661    885   48.071   7.065  114    32       11 sec
38113a AIB02DK 11887     827   47.770   7.833  197    475      28 sec
```

L'option `--metric` permet d'avoir l'altitude en mètres et la vitesse en kilomètres/heure, bien plus intelligible pour nous français que les pieds et les nœuds. Mais ce n'est pas tout. Le fait d'avoir activé les fonctions réseau permet à `dump1090` de faire de même que l'application Windows ADSB#. Il est donc possible d'utiliser adbsSCOPE pour afficher les données. Mais ceci ne sera pas nécessaire car nous disposons également d'un serveur Web intégré. Avec `dump1090` en marche, il vous suffit d'indiquer l'adresse **http://127.0.0.1:8080** dans un navigateur lancé sur la même machine pour voir apparaître une carte Google Map avec la position des avions dont nous recevons les messages :



Dump1090 est un outil simple mais très souple d'utilisation intégrant un serveur Web. Il suffit de pointer son navigateur dessus pour faire apparaître une carte avec chaque avion détecté.

Et ma Raspberry Pi alors ? L'installation se fera exactement de la même manière sur un PC sous GNU/Linux et un nano-ordinateur Raspberry Pi. Contrairement à la réception de radios commerciales FM, nous n'avons pas rencontré de problèmes particuliers en termes de stabilité en utilisant `dump1090`, si ce n'est en l'absence de hub USB actif qui systématiquement perturbe le fonctionnement de l'ensemble. Ceci aussi bien avec un récepteur à base de E4000 que les derniers modèles utilisant un tuner R280T.

L'élément intéressant dans l'utilisation d'une Raspberry Pi est la mobilité. En effet, utiliser une antenne telle que celle livrée avec le récepteur DVB-T est déjà passablement problématique puisqu'elle n'est pas conçue pour la réception à 1090 Mhz. Si vous en faites usage à l'intérieur d'un bâtiment c'est encore pire.

La solution consisterait donc à déplacer l'antenne en terrain découvert mais un autre problème se pose : plus le câble est long, plus on perd de signal et moins la réception est bonne. Avec la Raspberry Pi, il devient possible de placer le couple récepteur/Pi en terrain découvert et de faire transiter les données via un câble réseau (qui peut aller jusqu'à 135 mètres en Ethernet). Ainsi, si vous avez une maison individuelle vous pouvez passer un câble de votre appartement jusqu'au toit de votre immeuble, il suffit d'installer l'antenne sur le toit, la Raspberry Pi dans les combles et votre PC dans votre salon et le tour est joué. Petite astuce : si votre immeuble est ancien et votre appartement au rez-de-chaussé, les anciens conduits de cheminée sont vos amies ;)

Débarrassée de son écran et de son clavier, une Raspberry Pi ou tout autre carte du même type comme la BeagleBone Black peut ainsi faire office de système satellite dédié à la réception. Presque toutes les applications décrites dans ce dossier, aussi bien pour Windows que pour GNU/Linux, sont en mesure de travailler en réseau. Les outils installés avec RTLSDR inclus par exemple `rtl_tcp` capable de piloter le récepteur et d'envoyer les données brutes sur le réseau. Une autre solution est de démoduler le signal sur la Pi et de faire transiter le signal démodulé sur le réseau, pour le traiter et le décoder sur PC. Ceci demandera plus de travail à la carte et pourra poser quelques problèmes de stabilité. Dans le cas de `dump1090`, ce n'est pas un soucis. Tout ce qui transitera sur le réseau ce seront des données ADS-B ou HTTP.

3. SÉCURITÉ ET INTERROGATIONS LÉGITIMES

Nous venons de le démontrer, la réception des signaux *broadcast* ADS-B ne nécessite aucun mécanisme de sécurité. Les données sont envoyées des aéronefs à qui veux bien les recevoir. Ceci n'est pas un problème en soit si ce n'est pour certaines personnes de l'autre côté de l'atlantique qui se disent « Oh mon dieu ! Les méchants terroristes peuvent choisir les avions qu'ils souhaiteraient abattre avec un missile ! Vite surveillons les mails de toute la planète ! ».

L'écoute des signaux ADS-B est très courante et certains sites comme **www.flightradar24.com** centralisent les informations pour présenter une carte mondiale des avions dans le ciel. Jetez-y un œil, vous serez surpris de la quantité de choses qui volent au dessus de nous...

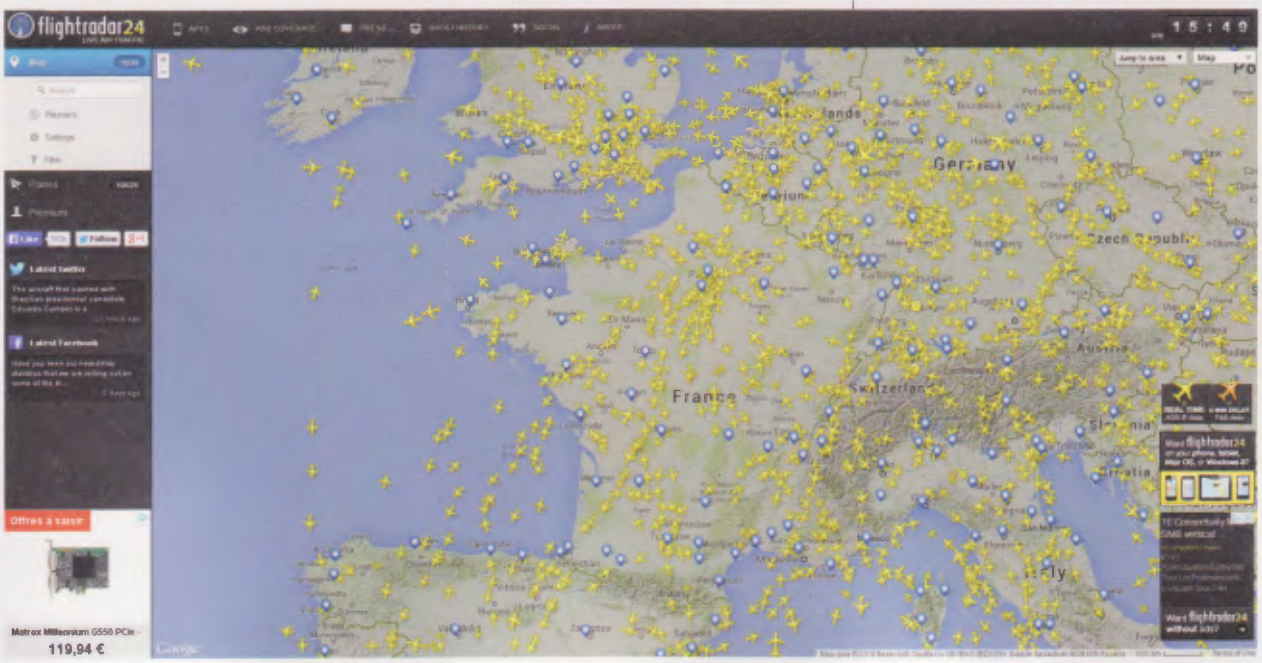
Mais le problème n'est pas réellement l'écoute des signaux mais d'éventuels émissions. Bien entendu il est parfaitement interdit d'émettre sur cette fréquence même en disposant d'une licence radioamateur, donc ce n'est pas censé arriver. Le problème est qu'on parle ici d'un comportement délibérément malveillant et l'aspect légal ou non n'entre pas en ligne de compte. Le problème a été signalé plus d'une fois lors de conférences de sécurités mais pourtant la question reste ouverte : que ce passerait-il si une personne mal intentionnée décidait d'émettre de

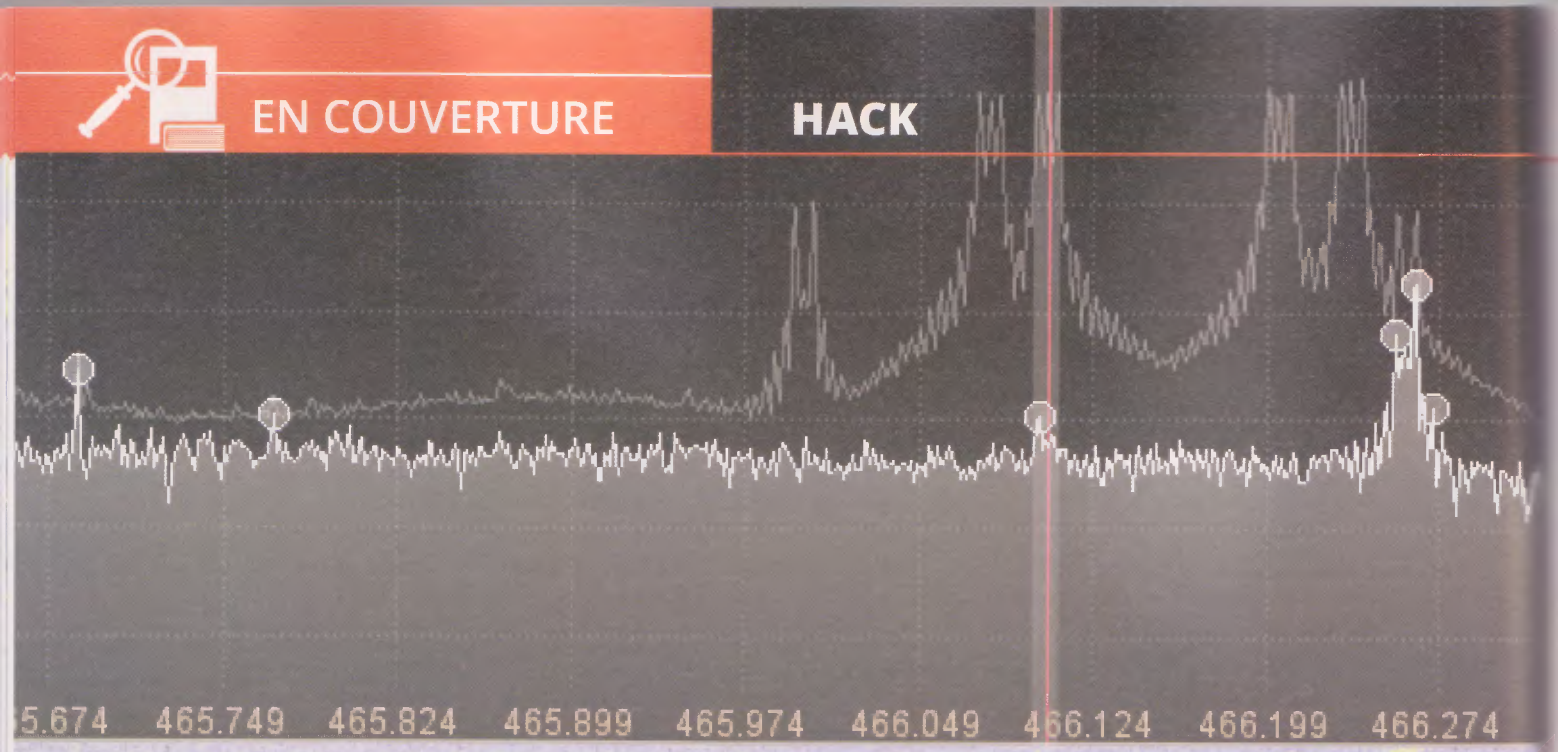
fausses informations ADS-B ? Celle-ci serait potentiellement en mesure de faire apparaître des avions fantômes ou pire encore, de cloner les informations d'avions en vol, semant une pagaille sans nom dans le ciel. Bien sûr, les aéroports disposent toujours des autres systèmes de contrôle comme les radars même si certains, outre atlantique, commencent à émettre l'idée qu'ADS-B, TCAS et d'autres systèmes de ce type sont moins coûteux, plus précis et pourraient donc faire disparaître les radars.

Notez que TCAS non plus n'est pas sécurisé et dans ce cas d'autres questions, plus grave encore, peuvent être posées : que se passe-t-il si une fausse alerte de collision est émise ? Les pilotes ont pour obligation de réagir à ce type d'événements. Existe-t-il un protocole de vérification ?
Les spécialistes en sécurité qui exposent le fonctionnement de ces systèmes lors de conférences et alertent sur les risques induis, se trompent-ils parce qu'ils ne disposent pas de toutes les informations ? Nous sommes en droit, je pense de nous poser la question.

Pour sa part, suite aux expérimentations du spécialiste en sécurité Nick Foster, la FAA (*Federal Aviation Administration* américaine) a répondu que ADS-B est sécurisé et que les faux aéronefs seraient détectés et filtrés par les systèmes, la preuve ADS-B a fait l'objet de certifications. Nous voilà rassuré, le porte-parole d'une administration qui a déjà investis énormément dans le système nous assure qu'on ne risque rien et que tout est parfaitement sous contrôle. Tout va bien dans le meilleur des mondes. J'évitai déjà de prendre l'avion avant d'écouter l'ADS-B, voilà une bonne raison de ne rien changer à mon comportement... DB

FlightRadar est un site collectant les informations reçue via ADS-B partout dans le monde et proposant ainsi une carte mondiale des avions dans le ciel. Oui, il y a du monde là haut, beaucoup de monde !





LA RADIOMESSAGERIE : CAPTEZ LES MESSAGES D'UN AUTRE TEMPS

D. B.

Non, cet article ne vous propose pas de capter les messages du passé ou d'une dimension parallèle. On parle ici de technologie et non de science-fiction. Ce titre fait en réalité référence à un système que beaucoup pensent inutilisé aujourd'hui car d'un autre âge, celui de la radiomessagerie permettant à un utilisateur de recevoir une notification ou un message textuel sur un petit boitier de réception.

La radiomessagerie est vue de nos jours, en Europe, comme quelque chose d'obsolète. Elle a connue ses heures de gloire dans les années 80 et 90, avant l'arrivée massive de la téléphonie mobile. Son principe de fonctionnement est simple : des petits récepteurs sont transportés par les utilisateurs qui reçoivent par le biais des ondes des notifications et des alertes. Le système est unidirectionnel et le porteur d'un tel récepteur, appelé *pager* ou *beeper*, doit alors utiliser un autre moyen de communication pour y répondre.

Il serait aisé d'en conclure que cette technologie n'est absolument plus du goût du jour car totalement remplacée par la téléphonie mobile permettant non seulement la réception de messages (SMS) mais également une communication bidirectionnelle aussi bien textuelle, vocale et multimédia (MMS). Il faut cependant prendre en compte quelques points qui rendent la radiomessagerie toujours très efficace :

- L'autonomie des récepteurs est très importante et se compte en mois,
- l'infrastructure de radiomessagerie a été utilisée très longtemps et est mature,
- les fréquences utilisées sont bien plus basses que celles de la téléphonie mobile et la portée des signaux s'en trouve augmentée,
- la couverture du réseau est très importante.

L'ensemble de ces caractéristiques fait que la probabilité de réception d'un message est très importante. Il sera pratiquement systématiquement reçu par le destinataire et cette « garantie » fait du système de radiomessagerie un support de choix pour la transmission d'alertes.

Ainsi, l'époque où les « gens importants » se promenaient avec leur pager à la ceinture a laissé sa place à celle des téléphones portatifs (type Radiocom 2000) puis à celle des premiers « mobiles » pour arriver maintenant à l'ère des smartphones. Mais ces pagers sont toujours très utilisés à la fois par des personnes mais également par des systèmes de surveillance et de contrôle (sondes, alarmes, capteurs, etc).

Plusieurs standards et normes existent permettant de transporter les notifications et les messages de radiomessagerie. En France, c'est le protocole POCSAG qui est mis en oeuvre. POCSAG signifie *Post Office Code Standardisation Advisory Group* et provient d'un comité de standardisation britannique, il est utilisé par la société E-Message France plus connue de par le nom commercial de son service de radiomessagerie, Alphapage, qui appartenait jusqu'en 1999 à France Telecom.

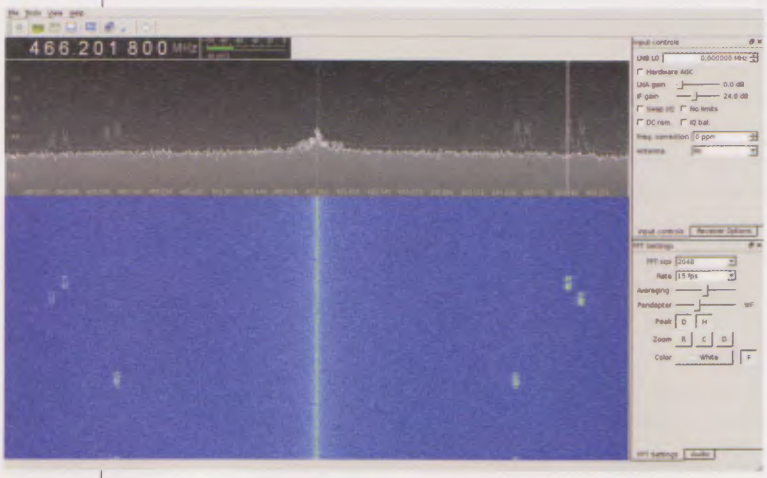
Les messages du système Alphapage sont émis, relayés et diffusés par des postes émetteur de 60 watts installés partout en France et fonctionnant en UHF avec des fréquences entre 466,025 MHz et 466,231 MHz. Le système de propagation de messages entre les relais est relativement complexe mais ceci n'a pas d'incidence sur les explications qui vont suivre. En effet, à l'aide d'un récepteur TNT (DVB-T) et d'un logiciel SDR, il est parfaitement possible de simplement capter les émissions POCSAG sur ces fréquences et les démoduler pour obtenir les messages sous une forme lisible. La modulation utilisée par POCSAG est appelée DFSK (*Direct Frequency-Shift Keying*) impliquant une porteuse au centre de la fréquence à écouter et des signaux émis en décalage de cette porteuse à +/- 4,5 kHz. Une émission sur le décalage haut représente un 1 et le décalage bas un 0. Avec ces signaux binaires, le protocole encode des mots de 32 bits incluant un type et un système de correction d'erreur. Chose remarquable et très pratique lorsqu'on explore cette gamme de fréquence, un message est toujours précédé d'une succession de 1 et de 0 sur 576 bits, formant un préambule permettant de synchroniser le signal. Ceci, en écoute, produit un son caractéristique facilement reconnaissable.

1. C'EST PARTI ! VERSION LINUX

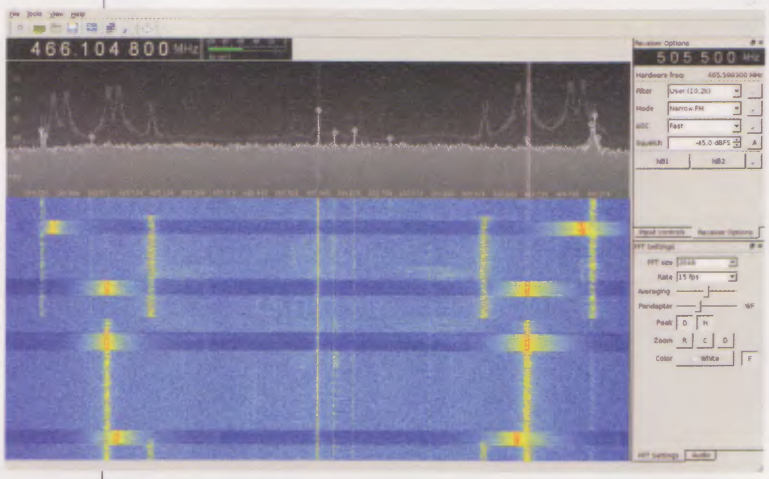
Avant toute chose, il faut que votre installation SDR fonctionne correctement. Nous partons donc du principe que vous avez correctement et avec succès suivi les explications détaillées dans l'article sur l'installation du dongle DVB-T et de RTLSDR sur votre système. Les émissions POCSAG sont très puissantes et il ne sera

pas nécessaire de disposer d'une antenne particulière. Celle livrée avec le récepteur TNT devrait donc parfaitement faire l'affaire même si elle n'est absolument pas conçue pour travailler dans des fréquences autour de 466 Mhz.

Pour nous en assurer, il suffit d'utiliser Gqrx et de régler la fréquence précisément sur 466 Mhz. Au bout de quelques instants, vous devez entendre et voir dans la visualisation *waterfall* quelque chose de ressemblant à ceci :



En jouant sur le gain, les réglages du récepteur et l'AGC, vous devriez être en mesure d'améliorer la réception au point que les émissions de signaux soient beaucoup plus clairs :



Ce que vous entendez en plaçant le curseur, ce sont les messages modulés du système Alphapage utilisant le protocole POCSAG. Les 576 bits de synchronisation, envoyés avant chaque message, produisent un son distinct suivi d'un chuchotement assez similaire à ce que produisaient nos bons vieux modems d'antan ou ce que vous entendez partiellement lorsque vous faites l'erreur d'appeler un numéro correspondant à un Fax. C'est normal, le Minitel par exemple utilisait la norme V.23 qui précisément repose sur une modulation FSK tout comme le V.27ter du Fax en DFSK (ah, souvenirs).

Pour obtenir les messages véhiculés par ces modulations, il nous faut convertir l'alternance de fréquences hautes et basses en bit puis traiter ces données. Rassurez-vous, nous n'allons pas faire cela à la main car Elias Önal a créé `multimon-ng` (un fork amélioré de `multimon`). Cet outil est en mesure de décoder une grande quantité de signaux numériques dont les protocoles POCSAG512, POCSAG1200 et POCSAG2400.

Comme pour les autres outils utilisés dans ce dossier consacré à la SDR et au détournement des dongles DVB-T, il va falloir compiler l'application à la main. Rendez-vous sur **https://github.com/EliasOenal/multimon-ng** et téléchargez l'archive Zip dans votre répertoire personnel. Il vous faudra également installer quelques paquets pour votre distribution afin que les sources puissent être compilée. Avec Ubuntu, Mint et/ou Debian, ce sera `libpulse-dev`, `qt4-dev-tools` et `qt4-qmake`.

Le reste se passe en ligne de commandes. Nous commençons par désarchiver les sources avant de nous y placer :

NOUVEAU ! NOUVEAU ! NOUVEAU ! NOUVEAU !

ABONNEZ-VOUS!

DÉCOUVREZ NOS NOUVELLES OFFRES !

LE CLASSIQUE !

VERSION PAPIER



Retrouvez tous les 2 mois votre bimestriel favori en papier dans votre boîte à lettres !



Envie de lire tous les 2 mois votre magazine sur votre tablette ou votre ordinateur ? Voici l'abonnement PDF !

Attention, les offres mentionnées dans la page suivante sont réservées aux particuliers, professionnels n'hésitez pas à nous contacter à : abopro@ed-diamond.com



L'ÉLECTRONIQUE !

VERSION PDF

boutique.ed-diamond.com

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

Édité par Les Éditions Diamond
Service des Abonnements
B.P. 20142 - 67603 Sélestat Cedex
Tél. : + 33 (0) 3 67 10 00 20
Fax : + 33 (0) 3 67 10 00 21

Vos remarques :

Découvrez une sélection d'offres page suivante

FAITES DES ÉCONOMIES !

Venez découvrir toutes nos offres de couplage avec :

LINUX MAGAZINE / FRANCE

GNU LINUX PRATIQUE — DÉCOUVRIR, COMPRENDRE ET UTILISER LINUX

MISC — Multi-System & Internet Security Cookbook

Open Silicium

sur :

boutique.ed-diamond.com

Veuillez cocher ci-dessous l'offre souhaitée, et veuillez renvoyer l'ensemble du coupon à l'adresse indiquée page précédente avec votre règlement. Vous pouvez également directement vous abonner sur notre site internet !

HACKABLE MAGAZINE

Merci de cocher ci-dessous l'offre souhaitée...

		TARIF NORMAL	ÉCONOMIE	TARIF SPÉCIAL*
☐	PAPIER — HK 6N°	47,40	8,40	39,-€
☐	PDF** — HK 6N°	47,40	8,40	39,-€
☐	BASE DOCUMENTAIRE** — ACCÈS 1 AN			Disponible septembre 2015
☐	PAPIER + PDF** — HK 6N° + HK 6N°	94,80	35,80	59,-€

*Prix TTC France Métro. Pour les tarifs hors France Métro et étranger, veuillez consulter notre boutique en ligne !

**Offre réservée aux particuliers, professionnels contactez-nous par mail : abopro@ed-diamond.com

HACKABLE MAGAZINE + Open Silicium

Merci de cocher ci-dessous l'offre souhaitée...

		TARIF NORMAL	ÉCONOMIE	TARIF SPÉCIAL*
☐	PAPIER — HK 6N° + OSI 4N°	83,40	18,40	65,-€
☐	PDF** — HK 6N° + OSI 4N°	83,40	18,40	65,-€
☐	PAPIER HK 6N° + OSI 4N° + BASE DOCU OSI** ACCÈS 1 AN	113,40	23,40	90,-€
☐	PAPIER HK 6N° + OSI 4N° + PDF** HK 6N° + OSI 4N°	166,80	66,80	100,-€
☐	PAPIER + PDF** + BASE DOCUMENTAIRE OSI** OSI 4N° + HK 6N° + OSI 4N° + HK 6N° + ACCÈS 1 AN	196,80	66,80	130,-€

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

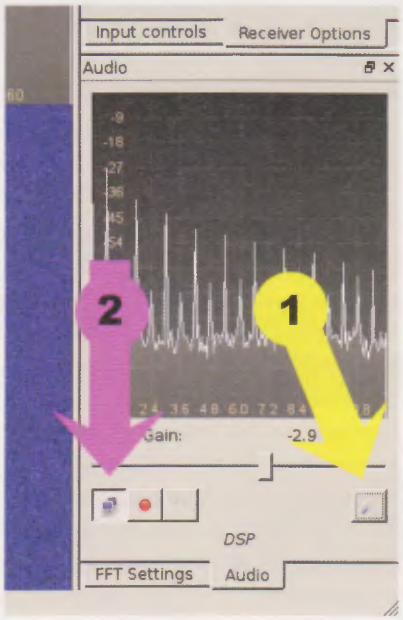
```
$ unzip multimon-ng-master.zip
$ cd multimon-ng-master
```

Comme pour RTLSDR, nous allons créer un répertoire pour construire l'utilitaire après avoir configuré les sources. Nous devons donc faire :

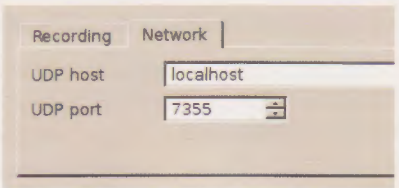
```
$ mkdir build
$ cd build
$ qmake ../multimon-ng.pro
$ make
```

Si la compilation s'est bien déroulée et que rien ne manquait, vous devez trouver dans le répertoire le programme `multimon-ng` et nous pouvons passer à la suite.

Commençons par la méthode pédagogique et implicitement la plus délicate (mais aussi celle qui vous apprend le plus de choses). Le principe sera le suivant : Gqrx va recevoir le signal et le démoduler puis il enverra les données audio obtenues via le réseau ou elles seront récupérées et envoyées à `multimon-ng` après une petite conversion. Cet enchaînement de commandes peut paraître démesuré, en particulier une fois que vous aurez lu la partie sur la méthode simple mais elle présente l'avantage d'avoir le contrôle sur la réception. Depuis Gqrx, c'est vous qui choisirez la fréquence et les paramètres tout en entendant les signaux et en visualisant les différentes émissions reçues.



Lancez donc Gqrx puis rendez-vous dans la configuration de la gestion réseau du flux audio, en bas à droite, onglet « Audio » sur la petite icône de configuration (flèche jaune) :



La fenêtre qui s'ouvre vous propose de configurer à quel serveur envoyer les données audio, par défaut *localhost* (la machine elle-même) sur le port UDP 7355 :

Vous pouvez laisser ces valeurs en place. Nous devons maintenant installer un serveur sur la machine attendant les connexions sur ce même port. Rien de bien complexe à installer nous allons utiliser le couteau suisse du réseau : Netcat (paquet `netcat-traditional`). Ce fantastique outil nous permet d'établir des connexions mais aussi d'en attendre, sur le port de notre choix et aussi bien en TCP qu'en UDP. L'enchaînement de commandes qui va suivre étant relativement complexe, nous allons tout d'abord en étudier chaque morceau. Tout d'abord, voyons Netcat lui-même. Nous pouvons rapidement lancer un serveur qui écoutera les connexions en UDP sur le port 7355 :

```
$ nc -l -u -p 7355
```

Nous avons `-l` pour *listen* (écouter), `-u` pour UDP et `-p` pour spécifier le port. Comme son nom l'indique, Netcat est une commande `cat` pour le réseau. Tout ce qui arrivera sur le port UDP 7355 sera donc affiché à l'écran semant une monstrueuse pagaille. Nous pouvons alors rediriger la sortie avec le caractère `|` (le *pipe*) vers une autre commande, comme `aplay` permettant de jouer des fichiers sonores. Bien entendu, il ne s'agit pas là d'un fichier mais de données brutes, nous devons donc indiquer les caractéristiques des données avec `aplay -r 48k -f S16_LE -t raw -c 1`. Nous précisons la fréquence d'échantillonnage de 48000 Hz (`-r`), le format de données en 16 bits signé *little endian* (`-f`), des données brutes (`-t`) et un seul canal (`-c`), pas de stéréo. En enchaînant les commandes ainsi : `nc -l -u -p 7355 | aplay -r 48k -f S16_LE -t raw -c 1`, vous allez

entendre la même chose qu'avec Gqrx mais au travers du réseau. Ceci peut être très intéressant en cas de réception sur un PC et traitement sur un autre. Je vous recommande de toujours installer et garder Netcat sous le coude, il peut vous rendre d'important services.

Dans le cas qui nous intéresse, il n'y a pas grand intérêt à écouter deux fois la même chose. Nous voulons que `multimon-ng` reçoive ces données et les utilise. Malheureusement, le format que nous envoi Gqrx utilise une fréquence d'échantillonnage trop importante, il nous faut le réduire. Pour cela nous utiliserons un autre outil magique : `sox`. Tout comme Netcat est le couteau suisse du réseau, `sox` est celui de la conversion audio. Installez donc `sox`, via le paquet du même nom. Observons ensuite la ligne de commandes qui fera la conversion. C'est un peu plus dense que la commande précédente : `sox -t raw -esigned-integer -b16 -r 48000 - -esigned-integer -b16 -r 22050 -t raw -`.

Pas de panique ! C'est plus simple qu'il n'y paraît, il suffit de découper la commande. Nous avons d'une part les données en entrée dont nous spécifions les caractéristiques, comme avec `aplay` : `-t raw -esigned-integer -b16 -r 48000 -`. Nous avons, comme précédemment, des données brutes, signées, sur 16 bits et avec un taux de 48000 Hz. Le `-` placé à la fin n'est pas une option ou une erreur, il désigne l'entrée standard en guise de fichier. Ceci permet d'enchaîner les commandes avec des `|`. Nous avons ensuite le format des données en sortie : `-esigned-integer -b16 -r 22050 -t raw -`, données brutes, signées, 16 bits mais en 22050 Hz. Là encore le `-` désigne un pseudo fichier mais cette fois c'est la sortie standard et ce toujours pour les même raisons. C'est tout.

Il ne nous reste plus qu'à passer tout cela à `multimon-ng` qui se trouve, je le rappelle, dans le répertoire où vous vous trouvez. Ses options sont bien plus simples : `multimon-ng -t raw -a POCSAG512 -a POCSAG1200 -a POCSAG2400 -`. `-t` précise des données brutes et chaque occurrence de `-a` ajoute un protocole à décoder. Ici dans le doute, nous utilisons toutes les déclinaisons de POCSAG. Enfin, le `-` en fin de ligne précise que les données arrivent par l'entrée standard. En assemblant tout cela on obtient :

```
$ nc -l -u -p 7355 | \
sox -t raw -esigned-integer \
-b16 -r 48000 - \
-t raw -esigned-integer -b16 \
-r 22050 - | \
./multimon-ng -t raw -a POCSAG512 \
-a POCSAG1200 -a POCSAG2400 -
```

Les `\` supplémentaires permettent simplement d'écrire cette longue suite de commandes sur plusieurs lignes, vous pouvez parfaitement tout saisir sur une seule ligne en retirant ces caractères ajoutés.

Votre serveur est maintenant lancé et, dans Gqrx, vous pouvez cliquer sur l'icône avec deux ordinateurs dans la partie audio en bas à droite de l'interface (flèche magenta sur la capture). Assurez-vous d'être en « Narrow FM » dans la partie « Receiver Options » (haut/droite) et mettez en route la réception. Dès lors, le signal capté sera démodulé et envoyé par le réseau à Netcat, `sox` puis `multimon-ng`. Ajustez le curseur pour le placer sur une fréquence où un signal est reçu et attendez simplement la prochaine émission. Comme vous pouvez le constater, il y a plusieurs « canaux » ou fréquences POCSAG. 466,205 Mhz, 466.080 Mhz, 466,203 Mhz, 466,236 Mhz, 466,105 Mhz... Ces valeurs pourront être très sensiblement différentes pour vous. Tous les récepteurs peuvent avoir une petite dérive et la plupart des applications SDR permettent de spécifier, à cet effet, une correction. Notez ces fréquences dans un coin, elles vous serviront dans la version simplifiée des manipulations.

Normalement, à chaque réception de signal que vous devez entendre clairement, dans la fenêtre de terminal où vous avez lancé la longue ligne des commandes doit s'afficher des messages. Toutes les transmissions ne sont pas nécessairement textuelles. La plupart des pagers ne font qu'émettre une alerte ou déclencher un signal lumineux. Ces notifications correspondent aux lignes `Function:` suivie d'un chiffre. Les messages de texte eux débutent par `Alpha` et peuvent être soit du texte lisible,

soit du texte lisible, soit une suite de caractères sans aucun sens (pour vous). Vous allez voir apparaître des messages du type `"MERCI DE RAPPELER"`, `"info : Test"`, `"Defaut CTA"` ou encore `"PING CRITICAL - Paquets perdus = 100%"`. Ce dernier genre de messages correspond à ceux des systèmes de surveillance de serveurs qui sont très courants. En passant un peu de temps en écoute, il est possible de tomber sur des informations amusantes, intéressantes, inquiétantes et même ce qui semble être du SPAM (?!). Nous ne partagerons pas dans ces pages les messages captés. Souvenez-vous, le respect de la confidentialité des communications est une obligation pour un radioécouteur et/ou un radioamateur.

A présent que vous connaissez les fréquences précises à écouter, nous pouvons passer à la version simplifiée. Nous n'aurons plus besoin de Gqrx ou d'un autre logiciel nécessitant notre contrôle direct. Avec RTLSDR est livré `rtl_fm` capable de piloter le récepteur, le caler sur une fréquence et de démoduler le signal. Il nous suffit alors d'utiliser les bons paramètres et d'envoyer tout cela directement à `multimon-ng`. Ainsi, en utilisant la commande suivante, nous obtenons un résultat identique au précédent :

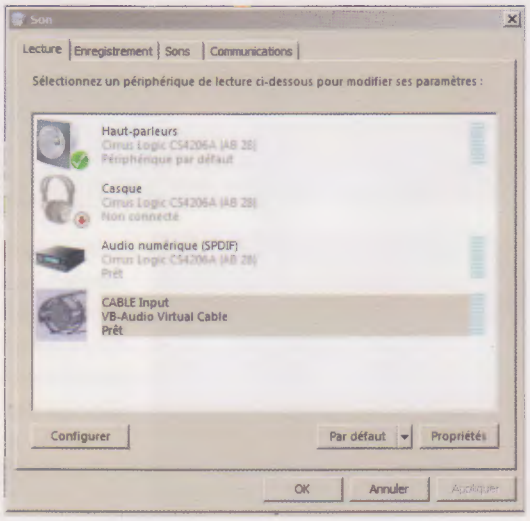
```
$ rtl_fm -s 22050 -f 466.103M - | \
./multimon-ng -t raw -a POCSAG512 \
-a POCSAG1200 -a POCSAG2400 -
```

La commande `multimon-ng` ne change pas. `rtl_fm` prend en option la fréquence d'échantillonnage (`-s`) et la fréquence à écouter (`-f`). On pourra sensiblement améliorer la réception en ajoutant les options `-A lut -o 4`. Là encore, il suffit d'attendre pour voir s'afficher des messages et notification divers. En l'absence de logiciel SDR complet, ceci ouvre des possibilités intéressantes avec, par exemple, une Raspberry Pi qui pourra facilement trouver sa place dans un endroit dégagé propice à la réception.

2. VERSION WINDOWS

Sous Windows, l'installation sera plus simple que sous Linux, en dehors de l'aspect audio. En effet, ce système étant principalement orienté vers l'interface graphique, il n'existe pas d'équivalent au caractère *pipe*. L'architecture générale consistera à recevoir et démoduler le signal avec HDSDR et à utiliser un logiciel particulier pour renvoyer le son à un outil de décodage POCSAG nommé PDW.

En effet, HDSDR est conçu pour démoduler un signal et l'envoyer sur la sortie audio standard de Windows. PDW, quant à lui, fait l'inverse, il prend un signal audio via une entrée standard de Windows et tente de décoder le protocole FLEX et/ou POCSAG. Il nous faut donc un moyen de connecter la sortie audio à l'entrée par un câble virtuel. Il existe plusieurs solutions pour ce faire, certaines payantes, d'autres gratuites. *VB-Audio CABLE* (**http://vb-audio.pagesperso-orange.fr/Cable/index.htm**) est un pilote Windows qu'il vous suffira de télécharger sur le site officiel. Après décompression du fichier Zip (`VBCABLEDriver_Pack42b.zip`), lancez simplement « Setup » en tant qu'administrateur pour l'installer. Dès lors, vous trouverez un nouveau périphérique audio dans votre configuration Windows :

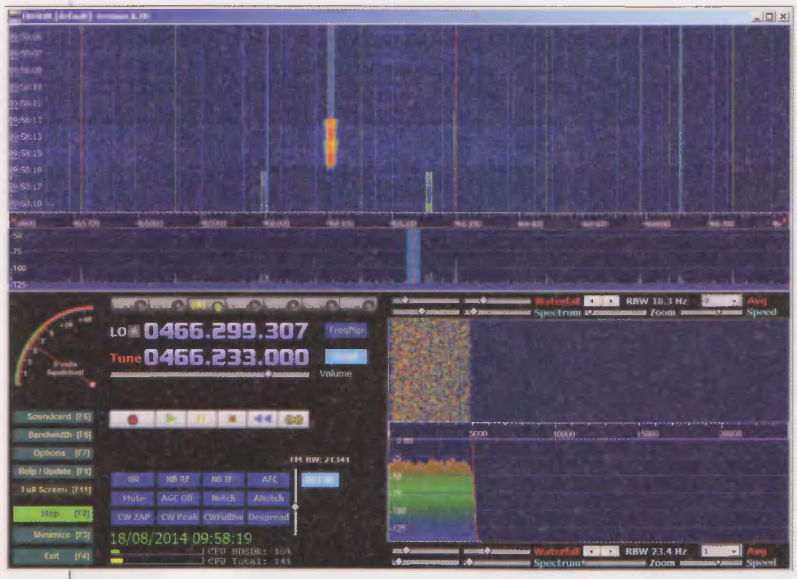


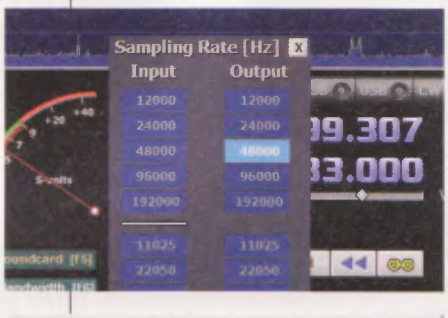
VB-Audio CABLE trouve ainsi sa place à la fois dans les périphériques de lecture et d'enregistrement audio, et pourra être utilisé par toutes les applications. Le fonctionnement est simple : ce qui rentre dans « CABLE Input » sort par « CABLE Output ».

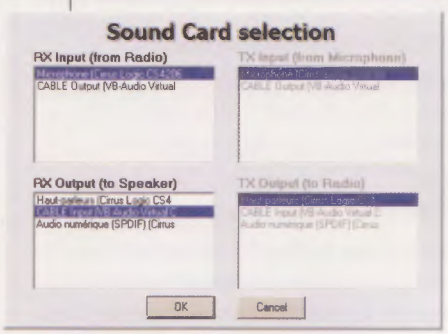
Il vous faudra ensuite installer PDW que vous trouverez à l'adresse **http://www.discriminator.nl/pdw/index-en.html**. La dernière version est la 3.12

et c'est celle que nous avons testé. L'installation se résume au téléchargement du fichier `pdw3.12-full.zip` et à sa décompression. Vous trouverez alors le programme `PDW.exe`.

La suite est une affaire de configuration. Dans HDSDR, commencez par caler votre récepteur vers 466 Mhz. Attendez quelques instants et vous devriez assez rapidement voir apparaître des émissions puissantes autour de cette fréquence. Placez le curseur sur l'une de ces fréquences clairement visible sur le graphique *waterfall* et vous pourrez entendre les messages POCSAG.



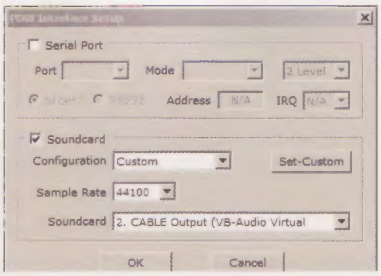




Assurez-vous d'utiliser la démodulation FM et d'ajustez la bande passante (clic droit sur le bouton « FM »). Faites également un tour sur le réglage de la bande passante en sortie via le bouton « Bandwidth [F6] » et réglez « Output » sur 48000 Hz.

Enfin, allez dans « Soundcard [F5] » pour régler la sortie audio non plus sur votre carte son mais sur « CABLE Input » pour « RX Output ».

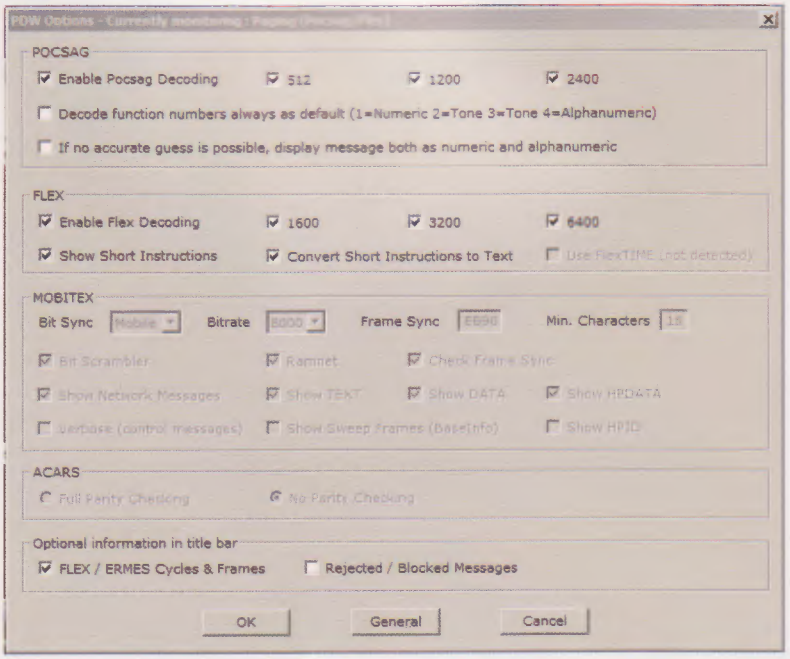
Voilà qui est réglé pour HDSDR, nous pouvons maintenant nous pencher sur PDW. Lancez le logiciel et commencez par faire un tour dans le menu « Interface » puis « Setup ». Là, assurez-vous que « Serial port » ne soit pas coché et que « Soundcard » le soit. Dans la configuration, choisissez « CABLE Output » :



La source du signal est à présent configurée mais il faut également choisir le type de protocole que nous souhaitons décoder. Cela se passe dans le menu « Options » puis « Options... ». Dans la fenêtre qui se présente, cochez « Enable Pocsag decoding » pour toutes les valeurs (vitesse) et confirmez en cliquant sur « OK » : voir la figure ci-contre (en haut).

Dans l'interface de PDW, utilisez la dernière icône de la barre, celle en forme de cible, jusqu'à ce que le titre de la fenêtre devienne « PDW 3.12 - POCSAG/FLEX » ou passer par le menu « Monitor » et choisissez POCSAG/FLEX. Il ne vous reste plus qu'à attendre : voir la figure ci-contre (en bas).

HDSDR va envoyer à PDW les signaux démodulés et ce dernier va tenter de les décoder et les afficher. Un VU-mètre en haut à droite de la fenêtre vous indiquera le niveau de réception. Si des émissions sont bien reçues par HDSDR mais que PDW ne semble pas réagir, avant toutes choses, essayez simplement de pousser le volume dans HDSDR (juste sous « Tune »).



3. MAIS QUEL INTÉRÊT ?

Au delà de la démonstration technologique découlant du fait d'utiliser un simple récepteur DVB-T à 15 euros pour recevoir, démoduler et décoder des messages, cet exercice en lui-même n'a que peu d'intérêt pratique. A moins de vouloir se prendre pour la NSA et collecter des informations à 99,999999% inutiles, disons-le franchement, c'est amusant un temps mais on s'en lasse vite. Moralement et politiquement en revanche c'est une toute autre histoire. Alors que les communications GSM/UMTS/4G/HSPA sont chiffrées et protégées, ainsi que celles de la police, de l'armée, des pompiers et du Samu (système ANTARES), on constate un certain laxisme au niveau des systèmes de radiomessagerie. La plupart des utilisateurs ne sont pas au courant que le premier venu, un rien curieux, peut lire tout cela sans le moindre problème.

La technologie n'est pas à mettre en cause, après tout elle à presque 25 ans ! Mais le manque d'informations lui est un grave problème. Deux facteurs sont à prendre en compte à ce niveau. Celui du fournisseur qui n'a pas de raison de préciser que les transmissions se font à tout va et celui du client qui ne pense certainement pas à poser la simple question : « mes communications seront-elles en sûretés avec ce système ? ».

POCSAG et les autres protocoles de radiomessagerie ne sont pas un problème en eux-mêmes mais cela dépend de la nature des données qui y transitent. De la même manière que vous n'allez pas confier un secret à un ami en le hurlant d'un bout à l'autre d'une rue, vous ne devriez pas envoyer d'informations sensibles via un système de radiomessagerie reposant sur des communications non sécurisées et non chiffrées. A cela s'ajoute un problème similaire à celui de l'ADS-B vu précédemment. Si les communications ne sont pas chiffrées et authentifiées, quel risque y a-t-il pour qu'une personne malintentionnée utilise ce système pour tromper un utilisateur ou un ensemble d'utilisateurs ? Cela relève-t-il de la paranoïa de penser que quelqu'un, quelque part, serait en mesure de commettre un acte illégal et de détourner l'attention en tentant de créer des fausses alertes ?

Nous ne pouvons rien faire de plus que de signifier le problème pour améliorer la situation. Si vous-même utilisez ces technologies ou que vous connaissez des personnes faisant reposer leur travail ou leur sécurité sur ce type de choses, posez-vous les bonnes questions et posez-leurs ces mêmes questions. La prise de conscience sur l'ensemble des technologies qui nous entourent et leur utilisation est un pas capital vers leur utilisation raisonnée, raisonnable et sûre. **DB**





MINECRAFT ET LA RASPBERRY PI

D. B.

Ah Minecraft ! Toute la beauté d'un jeu 3D actuel avec le design pixeleux d'un hit des années 80, où l'on passe son temps à creuser frénétiquement et se découvrir des talents de programmeur dès qu'on tombe sur les premiers fragments de minerai de Redstone... Mais pour l'utilisateur curieux (et un peu tordu) Minecraft peut être bien plus qu'un jeu de Lego virtuel, il peut être une interface pour piloter les entrées/sorties d'une Raspberry Pi !



Connaissez-vous Minecraft ? Non ? Alors après cette brève introduction suivi d'un essai, sachez que vous allez forcément vous classer dans l'une de ces deux catégories : celle de ceux qui disent « ce jeu n'a aucun sens, c'est n'importe quoi » ou celle des utilisateurs qui tombent instantanément amoureux du concept ne voyant littéralement pas les heures passées à casser des blocs, collecter des ressources, creuser toujours plus profondément et construire des cathédrales bourrées d'automatismes en tout genre. Car c'est là effectivement le but de Minecraft qui n'est rien d'autre qu'un environnement virtuel 3D où l'on débarque avec absolument rien et où l'on collecte des ressources pour créer ce dont on a besoin.

En début de jeu on ramasse du bois, pour en faire des planches, qui servent à créer un établi, qui permet de créer des outils, pour collecter de la pierre, dont on peut faire des outils en pierre, pour collecter du charbon et du minerai de fer, pour créer des outils en fer pour... Bref ! Vous avez compris.

En transformant les ressources et en les combinant entre elles, on arrive rapidement (ou pas) dans la boucle infinie consistant à collecter pour créer, pour collecter encore plus et donc créer davantage et ainsi de suite. Tout cela avec quelques contraintes imposées en fonction du mode de jeu choisi, comme par exemple en mode survie où il est de bon ton de rapidement se créer un abri pour passer la nuit. Car c'est à ce moment que des monstres sortent de leurs cachettes (squelettes, *creepers*, araignées, zombies, *slimes*, *endermans*, etc). Le reste du temps, le monde est rempli de créatures passives qui se promènent comme les poules, les vaches, les cochons, les lapins et même les très rares vachampignons (mi-vache, mi-champignon).



Au premier regard, Minecraft peut paraître surprenant : non, ce n'est pas une capture de CPC464 ou de MO5 mais bien un jeu vendu à plus de 16 millions d'exemplaires ces trois dernières années...



Le monde de la surface de Minecraft, l'Overworld... des blocs, des blocs, des blocs...

Vous l'avez compris, le principe du jeu peut paraître anodin au départ mais les possibilités sont énormes puisque vous devez gérer une grande quantité de choses tout en fabriquant littéralement votre environnement. Et ceci va de l'exploitation minière en passant par la culture, l'élevage, la confection de vêtements, les fortifications, la magie, etc. En début de jeu, on passe généralement la moitié de son temps sur le Wiki Minecraft pour apprendre comment *crafter* (confectionner) les objets dont on a besoin. Et comme si cela ne suffisait pas, il existe deux dimensions parallèles, l'*Ender* dans lequel se trouve l'un des boss du jeu, l'*Ender Dragon*, et le *Nether*, un monde souterrain obscur et peuplé de monstres agressifs. Bien entendu, certains éléments permettant de confectionner des objets se trouvent dans ces dimensions qui ne sont accessibles qu'en fabriquant des portails... avec d'autres éléments qui ne se trouvent pas facilement comme l'obsidienne, obtenue à partir de la lave et qui ne peut être collecter qu'avec une pioche en diamant.

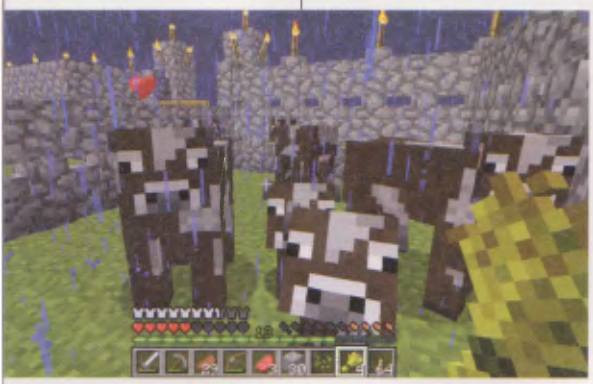


Objectif premier du jeu : Construire !

Terminons cette courte description avec, sans le moindre doute, ce qui est l'aspect le plus séduisant de Minecraft pour un programmeur : la *redstone*. Ce minerai particulier permet de produire la poudre de redstone avec laquelle on peut créer des circuits logiques et, en quelque sorte, programmer le monde de Minecraft via des automatismes. Il existe des quantités impressionnantes de circuit redstone faisant l'objet de tutoriels sur Youtube, allant de la simple porte automatique aux usines de production de ressources en passant par des ordinateurs 8 bits complets fonctionnant dans le monde de Minecraft.

Minecraft est une création du développeur suédois Markus Alexej Persson alias Notch. Le jeu est écrit en Java et fonctionnera, en principe, sous n'importe quel système d'exploitation disposant des ressources nécessaires (3D) dont Windows, Mac OS X et GNU/Linux. Notch lui-même s'est un peu éloigné du développement de Minecraft depuis 2011 pour le laisser entre les mains de ses collaborateurs, dont Jens Bergensten (Jeb), au sein de la société qu'il a créé à cet effet, Mojang. Notch s'était lancé sur un autre projet appelé 0x10c (*Ten to the C*), également un jeu de type *sandbox* qui intégrerait un processeur 16 bits (DCPU-16) capable d'exécuter les codes des joueurs. Depuis août 2013 le projet est cependant mis à l'écart mais Notch a précisé qu'il pouvait encore y avoir une possibilité pour qu'il voit effectivement le jour dans l'avenir.

Non, il n'y a pas un problème d'affichage. Ces vaches sont toutes pixeleuses et c'est fait exprès



En attendant, Minecraft fait littéralement un carton mondial et est disponible sur un nombre impressionnant de plateformes dont Android, Xbox 360/One, PS3/4/Vita, iOS et... Raspberry Pi ! Une importante communauté de joueurs créatifs s'est formée générant des quantités incroyables de tutos, de gameplay en vidéo et de démonstrations techniques ainsi qu'une masse importante de serveurs de jeux. Car Minecraft peut être joué de manière individuelle mais également à plusieurs en réseaux (LAN ou Internet) soit en mode découverte/aventure/création soit en joueur contre joueur. Un merchandising important gravite autour de Minecraft avec tout un tas d'accessoires, de gadgets, de vêtements et même une série de boîtes Lego sur le thème du jeu.

Enfin, si vous aviez des doutes sur la question, non Minecraft n'est pas open source. Il existe des clones jouables et de bonne qualité mais sans commune mesure avec le jeu original qui peut être utilisé gratuitement en version démo mais vous coûtera 20€ pour une version complète. On trouve cependant une mention sur le site officiel qui précise l'avenir du jeu : « *Once sales start dying and a minimum time has passed, I will release the game source code as some kind of open source* » (« une fois que les ventes commenceront à se tarir et qu'un temps minimum sera écoulé, je diffuserai les sources du jeu sous une forme d'open source »). C'est un type de politique déjà appliquée par certains éditeurs dont *id Software* créateur de Doom, Quake ou Wolfenstein mais qui n'est applicable que si le jeu est entièrement conçu par un seul ayant droit.

En effet, la plupart des jeux très populaires actuels sont le fruit d'un travail entre plusieurs sociétés, exactement comme la réalisation d'un film. La libération d'un tel projet n'est souvent jamais envisagé tout simplement pour des raisons de complexité juridique. Aussi, une telle annonce, si elle sera effectivement suivie des faits, est déjà un pas intéressant qui mérite d'être relevé.

1. MINECRAFT ET LA RASPBERRY PI

Le jeu Minecraft, bien qu'il soit disponible pour de nombreuses plateformes, n'est pas identique sur chacune d'elle. On doit ainsi distinguer la version classique de l'édition Pocket pour les smartphones et tablettes, de celle pour console et de celle pour Raspberry Pi, disponible depuis février 2013. Les éditions diffèrent sur les fonctionnalités proposées aux joueurs et en particulier la taille des cartes, la diversité des textures et les types de blocs disponibles.

L'édition pour Raspberry Pi, appelée « *Minecraft Pi Edition* » est assez proche de la Pocket Edition destinée aux appareils mobiles. Il n'est pas possible, par exemple de changer la texture des joueurs et seul le mode de jeu créatif est proposé. Ceci limite grandement l'intérêt de l'ensemble mais les barrières ne sont pas arbitrairement placées et dépendent des ressources limités des nano-ordinateurs. En effet, même si le processeur de la Pi peut être poussé à 900 Mhz ceci n'est en rien comparable à une architecture de type PC. Votre Pi, en termes de puissance de calcul est équivalente à un PC équipé d'un Pentium 3 d'il y a une dizaine d'années.



Minecraft n'est pas qu'un jeu de construction mais également de gestion où vous pouvez par exemple cultiver du blé

Malgré ces limitations et en raison de l'accélération 3D OpenGL/ES de la Pi, Minecraft est parfaitement jouable dans cette version. Il est donc possible de transformer son nano-ordinateur en mini console de jeux et se mettre à creuser frénétiquement. Cependant, il faut bien avouer que si l'on dispose d'un clavier/souris et d'un moniteur HDMI/DVI ce n'est pas vraiment pour se limiter à une version Pocket du jeu. On préférerait grandement profiter de la version PC, bien plus intéressante et complète. Malheureusement ceci n'est pas possible. La version Raspberry Pi est dédiée à la plateforme ARM et une version PC dispose de dépendances qui la lient avec l'architecture PC/x86 en termes de performances. On ne peut simplement copier le programme Java du PC sur la Pi.

Ainsi, l'aspect ludique est bien là mais l'expérience de jeu n'est largement pas optimale. Il y a mieux à faire avec une Raspberry Pi concernant Minecraft : créer un serveur...

2. LE SERVEUR : UNE VRAIE SOLUTION QUI MARCHE

Minecraft peut être joué en solo comme à plusieurs en partageant le même monde. Mais l'option de jeu en ligne n'est pas uniquement liée au mode multijoueur. Imaginez, en effet que vous disposiez de deux machines, disons une chez vous et une au bureau (oui, certains peuvent jouer au travail, j'ai un boulot sympa même si je ne dors pas beaucoup).

En mode solo, les données du monde sont conservées localement sur la machine, dans le répertoire `%appdata%\.minecraft\` sous Windows, `~/.minecraft` sous GNU/Linux et `~/Library/Application Support/minecraft` avec Mac OS X. C'est également à cet endroit que se trouve les parties

sauvegardées. Si vous souhaitez jouer sur une machine puis sur la seconde, vous devrez copier tout le contenu de ce répertoire d'une machine sur l'autre avant chaque session de jeu. Il existe des astuces permettant de déplacer tout cela sur une clé USB, voir pour automatiser l'utilisation du support au lancement du jeu (via la variable `APPDATA`). Mais tout ceci reste relativement pénible et surtout problématique si une machine est un Mac et une autre un PC Windows ou GNU/Linux.

Il existe une autre solution : un serveur. Dans ce cas, les données du monde ne sont plus sur votre machine mais le jeu se connecte, en mode multijoueur, sur un serveur qui fournira les données du monde tout en conservant vos modifications (exploitation, bâtiment, stock, etc). Mojang propose un service d'hébergement (mondialement depuis une paire de mois) mais il vous en coûtera quelques 10€/mois (avec un tarif dégressif sur la durée). À ce prix, toute la gestion, la maintenance et la sécurisation du serveur repose sur Mojang et non sur vous. On appel cela des *Realms* (royaumes) et il n'est pas nécessaire d'installer quoi que ce soit sur vos ordinateurs si ce n'est le jeu lui-même, vous vous connectez simplement à votre royaume et invitez éventuellement vos amis pour jouer.

En dehors de cette solution de facilité qu'on pourrait qualifier de réservée aux non-techniciens, vous avez la possibilité d'installer votre propre serveur, soit sur votre réseau local, soit via un hébergement internet. Certains hébergeurs proposent d'ailleurs des offres spéciales Minecraft. Mojang met à disposition à cet effet un programme, lui aussi écrit en Java, qu'il suffit d'installer sous Windows (`minecraft_server.1.7.10.exe`) ou via un fichier `minecraft_server.1.7.10.jar` sur n'importe quel système disposant d'une machine virtuelle Java fonctionnelle.

Tout le monde le sais, Java ça marche peut-être partout mais c'est lourd, très lourd. Il est impensable d'utiliser Java sur une machine ne disposant pas des ressources suffisantes et pour ceux d'entre vous qui en douteraient, pensant à Android, sachez que le système Google pour smartphone utilise une version tellement modifiée de la machine virtuelle Java qu'ils lui ont donné un autre nom, Dalvik.

Des développeurs ont donc eu la très bonne idée de créer un serveur Minecraft dans un langage plus adapté à ce type d'environnement, du C++, et de le mettre à disposition de tous en logiciel libre : c'est MCserver (**http://mc-server.org/**). Des versions pour GNU/Linux, Windows et Mac OS X existent et s'installent très facilement.

Pour utiliser MCServer sur votre Raspberry Pi, sachez que les recommandations précédentes à propos de l'overclocking et d'économie de mémoire sont tout aussi valables. Plus il y aura de ressources, plus le serveur sera à l'aise et fonctionnera efficacement.

Avant de vous précipiter sur le site officiel pour télécharger cette merveille, vous devez connaître le type de système que vous utiliser et découvrir, par la même occasion, quelques caractéristiques techniques de la Raspberry Pi.

Un ordinateur est une grosse machine à calculer. Il procède à toutes sortes d'opérations arithmétiques qu'on peut classer en deux catégories : celle sur des entiers et celle sur des nombres à virgule flottante. Pour augmenter les performances, une aide matérielle est parfois disponible sous la forme de circuits spécialisés, un FPU (*Floating Point Unit* ou « unité de calcul en virgule flottante »). Les utilisateurs les moins jeunes se souviendront peut-être des Intel 80387, des 80386DX et autres 80486SX/DX. Dans le monde ARM c'est la même chose. Certains processeurs disposent d'un FPU et d'autres non. Mais ce n'est pas tout, il faut également que le système d'exploitation le prenne en charge pour en tirer parti.

Le script de configuration permet de très simplement overclocker la Raspberry Pi pour augmenter ses performances. Attention cependant aux problèmes de température du processeur et aux risques d'instabilité

```
Chose overclock preset

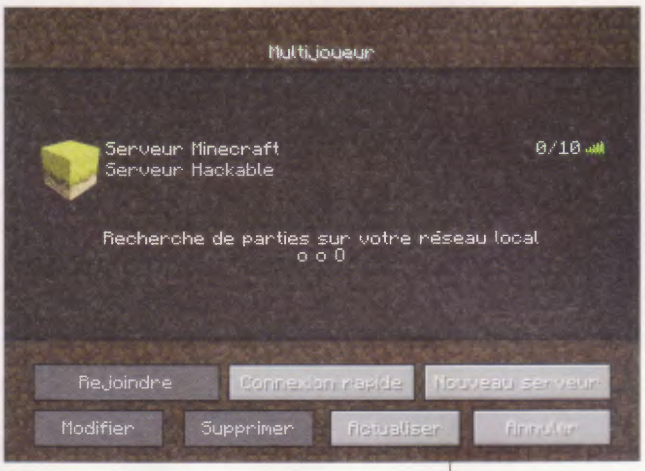
None   700MHz ARM, 250MHz core, 400MHz SDRAM, 0 overvolt
Modest 800MHz ARM, 250MHz core, 400MHz SDRAM, 0 overvolt
Medium 900MHz ARM, 250MHz core, 450MHz SDRAM, 2 overvolt
High   950MHz ARM, 250MHz core, 450MHz SDRAM, 6 overvolt
Turbo  1000MHz ARM, 500MHz core, 600MHz SDRAM, 6 overvolt
```

Avec une Raspberry Pi et Debian/Raspbian les choses sont un peu particulières. En effet, le projet Debian dispose effectivement d'un support pour un FPU dans un ARM mais uniquement à partir de l'architecture ARMv7 et au delà (`armhf`). Or, La Pi et son SoC/CPU Broadcom en particulier utilise une architecture ARMv6 avec FPU. Pour Debian c'est une machine sans FPU (`armel`) et avec la fonctionnalité qui est émulée (*Softfloat*). Voilà pourquoi Raspbian n'est pas vraiment une distribution Debian mais une sorte de monstre de Frankenstein.

La Raspberry Pi a donc bien une unité de calcul matériel mais vous devez savoir si le système que vous utilisez (Raspbian ou autre) supporte cette fonctionnalité. Pour cela, le plus simple est de taper la commande suivante :

```
$ readelf -A /proc/self/exe
Attribute Section: aeabi
File Attributes
  Tag_CPU_name: "6"
  Tag_CPU_arch: v6
  Tag_ARM_ISA_use: Yes
  Tag_THUMB_ISA_use: Thumb-1
  Tag_FP_arch: VFPv2          <---------
  Tag_ABI_PCS_wchar_t: 4
  Tag_ABI_FP_denormal: Needed
  Tag_ABI_FP_exceptions: Needed
  Tag_ABI_FP_number_model: IEEE 754
  Tag_ABI_align_needed: 8-byte
  Tag_ABI_align_preserved: 8-byte,
except leaf SP
  Tag_ABI_enum_size: int
  Tag_ABI_HardFP_use: SP and DP
  Tag_ABI_VFP_args: VFP registers
  Tag_DIV_use: Not allowed
```

Si vous voyez la ligne `Tag_FP_arch: VFPv2`, vous utilisez un système `armhf` ou *Hardfloat*. Si elle n'est pas présente, vous avez un système `armel` ou *Softfloat*. C'est riche de cette connaissance que vous pourrez alors pointer votre navigateur sur le site officiel de MCServer, cliquer sur le petit Tux et choisir soit *RasPi Hardfloat* soit *RasPi Softfloat*. Si vous vous trompez de version, soit le serveur fonctionnera en deçà de ses performances optimales, soit il va planter au démarrage puisqu'il va tenter d'exécuter des instructions qui ne sont pas connues du processeur.



Dès son lancement, le serveur est visible par n'importe quel client Minecraft présent sur le réseau

Le fichier que vous récupérerez sera une archive contenant la dernière version qui a été construite automatiquement : `MCServer.tar`. Placez-là directement dans votre répertoire personnel sur la Pi. Puis, en ligne de commandes, utilisez `tar xf MCServer.tar`. Vous allez obtenir un répertoire `MCserver` contenant le serveur. Placez-vous dans ce répertoire avec `cd MCserver` et profitez-en pour créer un répertoire `players` qui semble manquer (avec `mkdir players`), puis lancez le programme avec `./MCserver`.

Ce premier lancement prendra plus de temps que les suivants car le serveur va créer le monde, morceau (*chunk*) par morceau. Patientez jusqu'à voir la ligne `Startup complete` :

```
[03:45:03] 46 chunks remaining to light
[03:45:05] 22 chunks remaining to light
[03:45:08] Startup complete, took 130028 ms!
```

130038 millisecondes, soit 130 secondes ou un peu plus de 2 minutes. Les prochains démarrages se feront normalement en moins de 10 secondes. Dès que ce message est affiché, le serveur est accessible et vous pouvez connecter votre client Minecraft en utilisant l'adresse de la Pi. La connexion sera signalée à l'écran, en jaune :

```
User XXXXXX authenticated with UUID
e536754c5530433583328333c07a7734
Player XXXXXX has joined the game
```

Cela n'y ressemble pas mais l'endroit où s'affiche tous ces messages (là où vous avez tapé la commande) est également une interface pour contrôler le jeu. Vous pouvez donc y taper des commandes permettant d'influer sur ce dernier, le monde, l'environnement et les joueurs. Tapez par exemple `list` et vous verrez apparaître la liste des joueurs connectés. Tapez `stop` et vous arrêtez le serveur. Utilisez `help` pour avoir une liste de commandes possibles et leurs options. Il est possible de faire plein de choses amusantes comme téléporter un joueur, lui donner des objets ou encore changer la météo et même vider un inventaire. Bienvenue dans un monde Minecraft où vous régnez comme un dieu !

3. CONFIGURER SON SERVEUR

Pour l'heure, la configuration en place est celle par défaut et n'importe qui peut se connecter, il faut personnaliser tout cela. MCServer se configure soit via des fichiers `ini` soit via une interface Web. Les développeurs ont très bien fait les choses, c'est adorable. L'interface cependant n'est pas accessible. Arrêtez donc votre serveur puis ouvrez le fichier `webadmin.ini`. Le fichier contient déjà des informations de configuration. Ce sont celles qui ne débutent pas par un point-virgule (celles-ci sont des commentaires) :

```
[WebAdmin]
Enabled=1
Port=8080
PortsIPv6=
```

Avec `Enabled`, l'interface de gestion Web est active et `Port` défini le port TCP utilisé. `8080` n'est pas un port standard pour le web ce qui vous oblige à utiliser une URL du type **http://adresse:8080**. Si vous n'avez pas de serveur Web qui fonctionne sur la Raspberry Pi, vous pouvez changer `8080` en `80`, ce qui simplifiera les choses (mais vous devrez lancer le serveur en utilisant `sudo ./MCServer`, chose de toute façon nécessaire pour la suite de l'article). Il suffira alors de taper l'adresse de la Pi dans un navigateur pour tomber sur la page de gestion. Mais ce n'est pas tout, par défaut, l'accès n'est autorisé qu'aux utilisateurs qui se connectent. Or, il n'y a encore rien dans le fichier qui dit quel identifiant et mot de passe doit être utilisé. Ajoutez donc les lignes suivantes au fichier :

```
[User:admin]
Password=mot2passe
```

L'utilisateur sera `admin` et le mot de passe `mot2passe` (bien entendu, utilisez quelque chose de plus malin que cet exemple). Enregistrez le fichier, démarrez le serveur comme précédemment et pointez un navigateur sur l'adresse de la Pi (`ifconfig`). Là cliquez sur « *Log in* » et entrez les informations demandées. Vous voici dans l'interface de gestion, vous n'avez plus besoin de toucher aux fichiers `ini` pour administrer votre serveur (sauf cas particulier).

Précipitez-vous sur « *Server Settings* » et réduisez le nombre de joueurs à ce que vous estimez correspondre à vos besoins. Pour une carte Raspberry Pi, on essayera de ne pas dépasser 10 joueurs simultanés pour éviter les problèmes de consommation excessive de la précieuse mémoire.

À ce stade, votre serveur est publique. Non qu'il soit accessible depuis Internet mais n'importe qui sur votre réseau peut s'y connecter et jouer s'il possède un nom d'utilisateur (compte) Minecraft. Rendez-vous donc sur « *Whitelist* » et cliquez sur « *here* » (« *Click here to enable* »). Dès lors, le système de liste blanche est activé ou, en d'autres termes, personne ne peut se connecter sauf les joueurs dont le nom est sur la liste. Spécifiez donc votre nom d'utilisateur dans le champ et cliquez sur « *Add player* ».

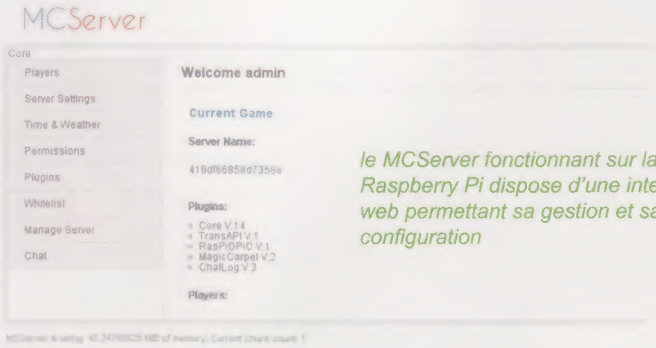
Si vous ne connaissez pas votre nom d'utilisateur, vous pouvez le retrouver dans le jeu ou tout simplement essayer de vous connecter au serveur. Le nom apparaîtra dans les messages.

Dernière étape préliminaire : vous donnez les pouvoirs que vous méritez en tant que créateur d'un monde (si déjà). En effet vous pouvez, sur le serveur lui-même, taper des commandes pour contrôler le monde mais il est bien plus agréable de le faire dans le jeu lui-même (*in game*). Cliquez donc sur « *Permissions* », écrivez votre nom d'utilisateur dans le champ « *Player* » et choisissez, juste en dessous, « *Owner* » (propriétaire) avant de cliquer sur « *Add player* ». Vous voici le chef incontesté et incontestable du monde (enfin, ce monde).

Owner est un des groupes d'utilisateurs existant par défaut. Les utilisateurs peuplent les groupes et ses groupes se voient attribués des permissions. Ainsi on trouve les groupes suivants par défaut :

- *Owner* : peut tout faire,
- *Moderator* : peut gérer le temps, les objets, téléporter les joueurs, bannir, etc. Tout ce qui est nécessaire à un modérateur devant appliquer la politique décidé en haut lieu,
- *Player* : peut jouer et se déplacer dans d'autres mondes via des portails,
- *Default* : un groupe spécial pour les utilisateurs qui ne sont pas dans la liste des utilisateurs.

Généralement, si vous n'êtes pas seul à utiliser le serveur, vous serez *Owner* et vous aurez un « capitaine » qui fera le *Moderator* (et passera pour le Père Fouettard à votre place) et les autres seront simplement des *Player*.



le MCServer fonctionnant sur la Raspberry Pi dispose d'une interface web permettant sa gestion et sa configuration

Le reste des paramètres pourra être laissé comme tel par défaut en attendant que vous ayez du temps pour vous plonger dans la documentation (**http://wiki.mc-server.org/doku.php?id=configure**) pour créer d'autres mondes en utilisant d'autres paramètres de génération avancés.

A ce stade, vous disposez d'un monde Minecraft portable que vous pouvez emporter avec vous un peu partout. Tout ce que vous avez à faire, c'est de connecter la Pi au réseau, la démarrer, relever son adresse, lancer le serveur et vous amuser. Soyez indulgent en cas de problème, MCServer est un projet en cours de développement, mais aussi en cas de ralentissement, votre Raspberry Pi fait ce qu'elle peut avec ce qu'elle a.

4. ET SI ON SORTAIT DU VIRTUEL ?

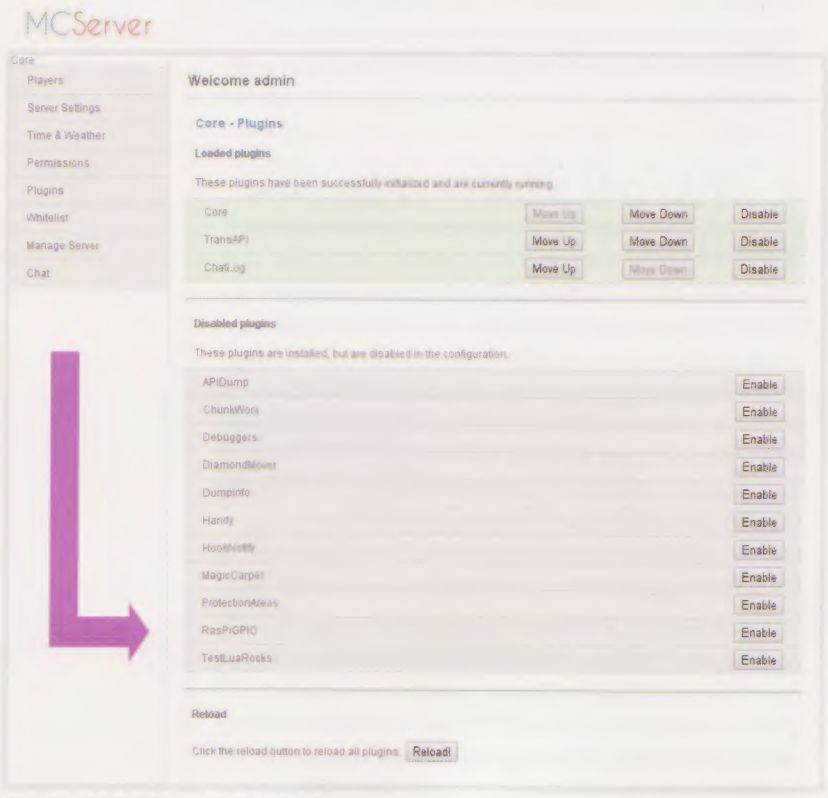
Évoluer dans un monde virtuel et créer plein de choses qui n'existent pas vraiment, c'est très rigolo mais ça ne change pas vraiment le monde. C'est bien là tout le problème du virtuel, tout ceci n'existe que dans les méandres de la mémoire d'un ordinateur. Que diriez-vous de faire interagir le monde virtuel avec le monde bien réel ? Et si je vous disais qu'en basculant un interrupteur dans votre monde Minecraft, vous pouvez influer sur l'état des broches de la Raspberry Pi ? Tordu ? Loufoque ? Impensable ? Oui, oui et non.

Un développeur nommé *matemat13* a conçu un greffon pour MCServer permettant ce genre de choses, car il faut savoir que MCServer permet la création d'extensions écrites en langage Lua. Ce dernier ne cesse de voir sa popularité grandir et est spécialement conçu pour ce type d'usage. On le retrouve déjà

MCServer intègre un système d'extensions écrites en Lua permettant d'ajouter des fonctionnalités et un développeur allemand a créé un module Lua pour accéder aux ports de la Raspberry Pi. Combinez les deux et le serveur Minecraft pourra agir sur les ports en installant RasPiGPIO !

embarqué dans de nombreux jeux dont World Of Warcraft ou encore Star Wars Battlefront, les Sims, Crysis, Warhammer... Or un développeur allemand (**http://www.andre-simon.de/**) a eu la bonne idée de développer une extension Lua pour accéder aux ports de la Raspberry Pi. Il n'en fallait pas plus à *matemat13* pour créer *MCserver-RaspberryGPIO*.

Pour réaliser ce petit miracle avec votre Pi, vous devrez tout d'abord installer le paquet `liblua5.1-0` avec la commande `sudo apt-get install`. Pointez ensuite votre navigateur sur **https://github.com/matemat13/MCserver-RaspberryGPIO**. Dans le répertoire `Lua only` et téléchargez le fichier `GPIO.so` qui devra être placé dans `/usr/local/lib/lua/5.1/`. Vous pouvez également télécharger tout le dépôt GitHub (bouton « *Download ZIP* » sur la droite) pour ensuite désarchiver `MCserver-RaspberryGPIO-master.zip` avec la commande `unzip` et copier plus facilement les fichiers.



Cette copie aura pour effet d'ajouter les fonctions Lua permettant d'agir sur les ports de la Raspberry Pi. Vous devez ensuite copier le sous-répertoire `Lua only/Plugins/RasPiGPIO` dans le répertoire `Plugins` du répertoire `MCServer`. Enfin, démarrez le serveur via `sudo ./MCServer` (l'utilisation de `sudo` est obligatoire pour accéder aux ports). Dans l'interface Web, à la section « *Plugins* », vous devez trouver une ligne *RasPiGPIO*. Cliquez sur le bouton « *Enable* » puis sur « *Reload now!* » pour recharger les extensions. Le greffon doit alors apparaître dans le haut en compagnie des autres extensions actives.

Sur l'écran de la Raspberry Pi, une ligne vous signale également le chargement :

```
Initialised RasPiGPIO v.1
```

Il ne vous reste plus qu'à vous connecter au serveur avec votre client Minecraft. L'extension de *matemat13* se souvient du dernier bloc posé et s'il s'agit d'un levier, un bouton de pierre ou de bois, une torche ou une lampe Redstone, celui-ci peut faire office d'actionneur. Posez par exemple un levier, puis tapez la commande `/assignlast` suivie du numéro de broche de la Raspberry Pi sur laquelle vous avez placé une résistance de 330 Ohm et une led reliée à la masse. Activez le levier et... la led s'allume. Actionner-le encore et elle s'éteint !

Et ça marche dans le sens inverse aussi. Placez un bouton, utilisez `/assignlast 13 IN` et lorsque la broche 13 de la

Raspberry Pi sera à la masse, le bouton sera désactivé, lorsqu'il sera à 3,3 volts, le bouton sera activé. Bien entendu les leviers comme les boutons dans Minecraft peuvent être reliés avec de la Redstone à d'autres éléments comme des pistons et il devient possible de créer toutes sortes d'assemblage faisant interagir le jeu avec le monde extérieur et inversement.

Le reste, ce n'est que de l'électronique et il devient possible d'envisager quelques utilisations pratiques. Si la Raspberry Pi est équipée d'un module permettant de piloter des relais, vous voici en mesure de contrôler un appareil depuis Minecraft. Pourquoi ne pas créer une interface domotique de la sorte en contrôlant l'éclairage de votre maison depuis votre monde virtuel ? Certes, ce n'est pas ce qu'il y a de plus pratique mais c'est terriblement amusant. *matemat13* a également intégré dans son extension une autre fonctionnalité : si vous placez un panneau et y écrivez « \RAM » ou « \CPU », ce dernier affichera alors des statistiques d'utilisation de la mémoire et du processeur de la Raspberry Pi.

5. POUR FINIR

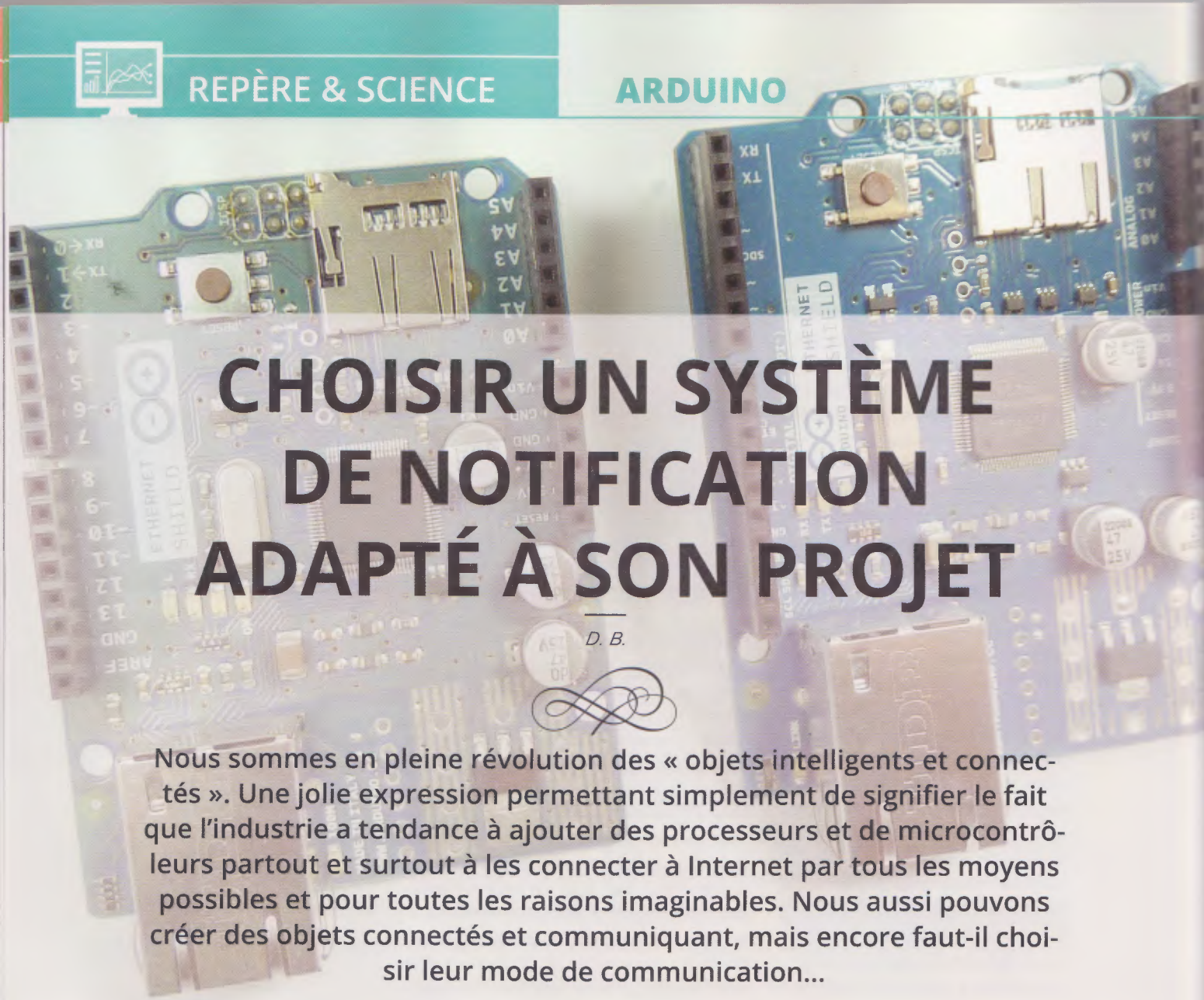
Nous avons appris ici plusieurs choses intéressantes et surtout nous avons constaté qu'un nano-ordinateur est loin de fournir assez de puissance pour remplacer une machine de bureau dans le domaine du jeu. On comprend alors mieux pourquoi les constructeurs de smartphone se sont lancés dans une course à la débauche de puissance (multi-coeurs, mémoire, accelération graphique) afin de s'ouvrir au monde des joueurs. Mais le plus intéressant reste ce qu'il est possible de faire lorsque différents projets open source sont combinés, même avec une base propriétaire comme Minecraft. Si vous avez un peu de temps, je vous recommande d'explorer MCServer et Lua et d'en apprendre un peu plus sur les possibilités de création de mondes et d'extensions. Ce ne sera pas du temps de perdu car, soyons honnêtes, casser des blocs pour construire des bâtisses géantes c'est bien amusant mais apprendre un nouveau langage et créer ses propres extensions c'est bien plus constructif. **DB**



Ce levier, planté là dans notre monde virtuel, active la broche 12 de notre Raspberry Pi sur laquelle est connecté une led



Le petit bonus livré avec l'extension RasPiGPIO de MCServer, vous pouvez planter des panneaux qui vous indiquent l'état des ressources de la Raspberry Pi



CHOISIR UN SYSTÈME DE NOTIFICATION ADAPTÉ À SON PROJET

D. B.

Nous sommes en pleine révolution des « objets intelligents et connectés ». Une jolie expression permettant simplement de signifier le fait que l'industrie a tendance à ajouter des processeurs et de microcontrôleurs partout et surtout à les connecter à Internet par tous les moyens possibles et pour toutes les raisons imaginables. Nous aussi pouvons créer des objets connectés et communiquant, mais encore faut-il choisir leur mode de communication...

Les objets intelligents c'est bien, mais ils ne disposeront jamais davantage d'intelligence que celle déjà présente dans leurs créateurs. En d'autres termes, les choix techniques faits pour rendre un objet communiquant, et en particulier communiquant via Internet, doivent faire l'objet d'une vraie réflexion préalable.

Nous allons évaluer ici les différentes options qui se présentent (ou pas) à nous. La problématique est la suivante :

- Un événement est détecté par notre carte Arduino. Il peut s'agir d'une pression sur un bouton mais également de n'importe quelle action détectée par nos montages comme un changement de température, une détection de mouvement, le déclenchement d'une alarme, l'ouverture d'une porte, etc.
- Cet événement est traité par le montage, évalué et analysé. En conséquence de quoi il décide ou non de signaler à quelqu'un cet état de fait. L'idée est de faire savoir à un utilisateur que quelque chose vient de se passer, où qu'il soit et de préférence sans intervention de sa part. On parle clairement de notification. Ce type de différentiation se trouve également au plus bas niveau, c'est la différence entre une interruption et la méthode dite de « *pollin* » consistant, pour un morceau de code, à vérifier régulièrement l'état d'un élément surveillé. Ici, il s'agit d'interrompre l'utilisateur, non que celui-ci « *poll* » notre montage.

1. NOTIFIER UN UTILISATEUR

Les solutions permettant de notifier une personne sont nombreuses mais pas forcément toutes utilisables avec une carte Arduino. Précisons ici que nous ne sommes pas dans un contexte critique. Entendez par là que nous ne cherchons pas à mettre en place un système sur lequel peut reposer des vies humaines. Ce type de systèmes impose des contraintes strictes comme l'assurance de la transmission et la réception de la notification, ainsi que le délai contrôlé de cette opération. Des systèmes comme Isys de *e*message* permettent ainsi de traiter le cas des sites industriels à risque susceptibles d'émettre des alertes via plusieurs canaux de communication de manière sûre avec un maximum d'efficacité. Nous, ici, ne sommes pas dans ce cas de figure. Nous voulons simplement signaler d'un événement non critique. Ceci signifie également que vous ne devez, en aucun cas, considérer ce qui sera exposé ici comme une solution adaptée pour, par exemple, gérer des alertes liées à l'état de santé d'une personne.

Voici une liste non exhaustive des solutions qu'il est possible d'envisager dans notre situation, leurs avantages et leurs problèmes. Toutes se basent sur le périphérique que la grande majorité d'entre nous possède (ou presque) : un smartphone.

1.1 XMPP et messagerie instantanée

C'est initialement la solution que nous avons cherché à utiliser en premier lieu. Le fait qu'un objet puisse se connecter à un service de messagerie instantanée paraît très séduisant puisque ceci revient à créer ce qu'on appel un *bot*. Bien avant le « tchat » existait l'IRC, des salons de discussions en ligne permettant à plusieurs utilisateurs de venir dialoguer ensemble par l'intermédiaire de messages textuels. Organisés en canaux IRC, il est très courant que chacun d'entre eux dispose d'un *bot* gérant le respect des règles établies sur le canal (comme la non utilisation de certains termes grossiers). Ces *bots* généralement « fait maison » sont alors chargés de rejeter un utilisateur, voir de le bannir, suite à des violations répétées des règles.

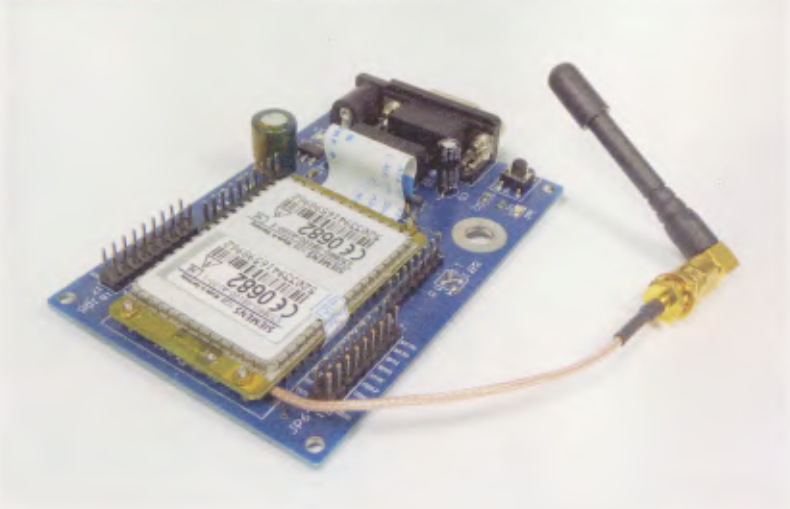
Le même principe peut s'appliquer pour la messagerie instantanée. Il suffit d'écrire un programme qui se connecte, lit les messages et en écrit, comme s'il s'agissait d'un utilisateur humain. L'une des messageries instantanées les plus présentes et la mieux intégrée dans nos smartphones est sans conteste Google Hangout, alias Google Talk. Ce niveau d'intégration permet au mobile de produire des notifications si l'application est connectée à Internet et, point positif, le système repose sur un protocole connu, documenté et ouvert : XMPP.

Il est donc possible, en principe, via un shield Ethernet ou Wifi de faire en sorte qu'une carte Arduino se connecte à Internet et au serveur de messagerie pour faire office de *bot*. Malheureusement, si cela était utilisable fût un temps, ce n'est plus le cas. Non que Google interdise ce genre de choses mais simplement en raison de la sécurisation de l'ensemble. En effet, le protocole XMPP consiste en l'échange de messages entre votre application ou le *bot* et le serveur. Ces messages sont des commandes échangées sous forme de texte qu'il est parfaitement possible d'espionner. Pour régler ce problème de sécurité, on utilise donc le chiffrement et en particulier SSL/TLS. Depuis quelques temps, la majorité des serveurs de messagerie instantanée et en particulier ceux de Google, ne tolèrent plus **que** des connexions avec SSL/TLS pour des raisons évidentes de sécurité des utilisateurs.

1.2 Les plateformes de notification

Avec la popularisation et l'intérêt grandissant pour l'Internet des objets (ou IoT en anglais pour *Internet of Things*), certains ont bien compris l'intérêt de développer et faire fructifier une activité reposant sur la notification. Dans cet ordre d'idées nous trouvons, par exemple, NMA ou *Notify My Android*. Il s'agit d'une plateforme permettant de transmettre ce qu'on appel des notifications *push* à un périphérique Android depuis littéralement n'importe quel objet connecté.

NMA se compose d'une application à installer sur son smartphone Android et d'une interface de programmation ou API (*Application Programming Interface*). Ce système n'est pas spécifique aux matériels comme l'Arduino mais permet à n'importe quelle application de « pousser » des notifications sur un périphérique Android. Pour cela NMA propose des bibliothèques pour plusieurs langages de programmation dont Java, C/C++, .Net,



PHP, Ruby, Python... Un tutoriel est disponible sur le site *Instructables* à ce propos (**http://www.instructables.com/id/Notify-My-Android/**). Tout comme avec la solution précédente, pour permettre l'utilisation de NMA tout ce qu'il faut c'est équiper sa carte Arduino d'un shield permettant sa connexion à Internet.

La société à l'origine de NMA recommande fortement d'utiliser le protocole HTTPS (avec SSL/TLS donc), chose qu'il est encore impossible d'utiliser simplement avec un Arduino (autre que Yún, etc). Notez qu'il existe de toutes petites implémentations de SSL/TLS, comme CyaSSL/yaSSL ou PolarSSL mais personne n'a encore achevé le très difficile travail de portage vers un Atmel AVR 8 bits. Ce serait une véritable prouesse technologique mais d'un intérêt pratique limité. Ce type d'implémentation trouve généralement son intérêt avec des microcontrôleurs plus puissants comme de l'ARM Cortex-M ou de l'AVR32. NMA précise cependant qu'il est possible d'utiliser la solution sans SSL/TLS et donc en communiquant « en clair » via HTTP.

Nous devons nous résoudre à cet état de fait pour l'ensemble des solutions utilisant Internet. Il nous sera très difficile d'obtenir la moindre sécurité étant donné les limites des cartes Arduino les plus courantes et populaires. Le problème qui se pose avec NMA n'est pas là, mais plutôt dans le fait que le service soit gratuit.



En effet, qu'on le veuille ou non, si le service qui vous est proposé est gracieux c'est invariablement que la société qui le propose se finance via un autre moyen, et ce moyen c'est vous ! En faisant un petit tour dans la foire aux questions proposée sur le site de NMA on se rend compte que respect de la vie privée ou de l'utilisation de vos données n'est absolument pas abordée. Il faut pour cela faire un tour à la page « Privacy Policy » où il est précisé que vos informations peuvent être transmises si la société considère qu'il est légitime de le faire pour des raisons légales ou pour protéger ses droits, ceux d'autres entités, la propriété ou la sécurité.

1.3 Envoi de SMS

Voici sans doute la solution la plus sûre mais également la plus coûteuse. Plutôt que d'opter pour un shield permettant la connexion au net, nous pouvons faire le choix d'un autre réseau tout aussi important : GSM/GPRS. Il existe, en effet, des shields permettant une connectivité mobile, exactement comme votre téléphone. Ceci demande non seulement l'utilisation d'un matériel dédié, un module GSM, mais également un abonnement dédié chez un opérateur. Heureusement, il existe actuellement des offres permettant, pour une paire d'euros, de disposer d'un forfait SMS illimité.

Car c'est bien de cela qu'il s'agit : utiliser comme système de notification l'envoi d'un SMS. En équipant votre carte Arduino d'un tel module ou shield et d'une carte SIM adaptée, il est relativement simple d'envoyer un SMS à un mobile. Il est même possible d'envoyer plusieurs SMS à des mobiles différents.
La sécurité n'entre ici pas en jeu puisque l'ensemble repose sur celle du réseau lui-même. Si vous ne vous inquiétez pas de la possibilité d'interception des SMS avec votre téléphone mobile, vous n'avez pas à le faire concernent un module GSM.

Il est sensiblement plus difficile de piloter un module GSM qu'un shield Arduino Ethernet ou Wifi depuis un croquis. Mais le problème est ailleurs puisque cela n'est largement pas insurmontable, même pour un débutant. En utilisant un tel module, vous vous retrouvez restreint à l'utilisation du support SMS et donc à la notification sur mobile.

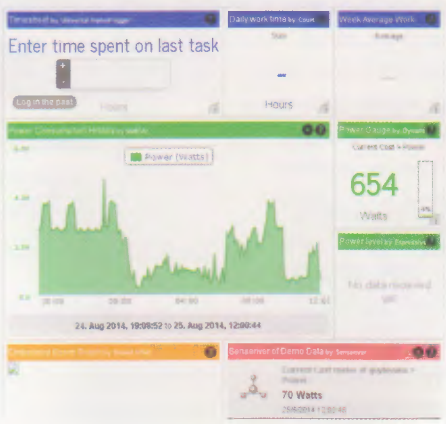
Certes, la connectivité mobile est plus facile à obtenir que celle d'Internet pour un smartphone et certes, ceci vous autorise à ne disposer que d'un appareil classique (non-smartphone, vous savez les mobiles qu'on ne recharge qu'une fois par mois et pas tous les soirs). Vous vous privez cependant de notification sur PC, dans un navigateur Web ou sur une tablette non 3G/LTE/GPRS/GSM et, ne l'oubliez pas, il vous faut souscrire un nouvel abonnement téléphonique.

1.4 Les plateformes de collecte de données IoT

Certains sites Web proposent un service de centralisation de données relevées par des capteurs de toutes sortes : température, consommation électrique, humidité, radiation, taux d'illumination, etc. Tout le travail de collecte est fait par vos périphériques et votre tâche s'arrête à l'envoi des données à la plateforme. Il est ensuite possible de définir des alertes en fonction des données et de procéder à des notifications du ou des utilisateurs.

Historiquement la plateforme la plus connue est Pachube qui est ensuite devenue Cosm et puis Xively qui est maintenant une partie de la société *LogMeIn Inc* côté au NASDAQ (LOGM). Ce qui était au départ un petit projet anglais a maintenant tous les aspects d'une *mégacorp* américaine. Ce qui reste de Pachube est intégré dans une offre *Xively Cloud Services* regroupant bien plus de choses et surtout s'adressant à des partenaires commerciaux plutôt qu'aux particuliers souhaitant centraliser ces informations. L'aspect communautaire a semble-t-il disparu de la plateforme pour qu'elle puisse se transformer en offre cloud dans l'air du temps, prête pour l'internet des objets dans les nuages...

Une autre solution reprenant, dans les grandes lignes, l'esprit laissé de côté par Pachube/Cosm, est la plateforme *Sen.se*. Celle-ci est l'oeuvre de Rafi Haladjian, créateur de la société Violet, depuis cédée à Mindscape (puis reprise par Aldebaran Robotics), à qui on doit le Nabaztag, un objet communicant en forme de lapin. La philosophie de *Sen.se* consiste à ne pas vouloir parler d'Internet des objets mais de considérer toutes les sources de données et d'informations au même niveau. Dans les faits, la plateforme *Open.Sen.se* est utilisée par les béta-testeurs pour principalement « remonter » et présenter des données provenant de capteurs ou de systèmes de mesure divers. L'ensemble est alors présenté sous une forme graphique conviviale et facilement intelligible, dans un navigateur Web.



Tout cela se présente donc très bien. Seules ombres au tableau, le fait qu'il ne s'agisse pas d'une solution de notification directe et, surtout, le caractère béta de la plateforme. En effet, à l'heure actuelle *Open.Sen.se* n'est pas ouvert au public et son utilisation ne peut se faire que via la création d'un compte découlant d'une invitation. Le site Web permet de formuler une demande d'invitation et donc l'ouverture d'un compte de test. Ces deux éléments particuliers concernant la plateforme nous font écarter cette solution, du moins pour l'objet qui nous intéresse ici.

1.5 Envoi de courrier électronique

Qu'est ce qui vous signal généralement qu'on cherche à vous contacter ? La messagerie bien sûr ! Éludons rapidement l'option en précisant que la plupart des prestataires permettant d'envoyer des mails ont sécurisé l'acheminement des données de la même manière que cela a été fait avec la messagerie instantanée. Or nous avons vu que l'utilisation de SSL/TLS n'est pas à notre portée avec un Arduino et son shield Ethernet ou Wifi. Les protocoles SMTPS, POP3S et IMAPS ont globalement remplacé SMTP, POP3 et IMAP. Celui qui nous intéressait était SMTP, le protocole utilisé par votre client de messagerie pour passer ses messages au serveur de votre prestataire qui l'utilise, lui-aussi, pour les transmettre aux serveurs

des destinataires. En vérité, si votre fournisseur ne propose pas et ne vous oblige pas à utiliser des protocoles en version sécurisée et chiffrée, mieux vaudrait vous tourner vers un autre fournisseur, car tous vos messages circulent alors en clair entre votre matériel et ses serveurs.

La solution pour contourner le problème serait de disposer d'un serveur relais acceptant les messages et les passant au serveur de manière chiffrée. Ceci impose soit la mise en place d'un PC soit celle d'un système moins gourmand en énergie comme une Raspberry Pi. Retour à la case départ, on se retrouve exactement dans la même situation que pour la messagerie instantanée...

2. LE CHOIX DE TWITTER ET COMPTE PRIVÉ

Twitter, comme vous devez le savoir est une infrastructure qu'on pourrait presque qualifier de plateforme de notification. Si vous étiez dans une caverne les dix dernières années, sachez que Twitter est un outil de microblogage. Entendez par là qu'il offre la possibilité aux utilisateurs inscrit de poster des messages de maximum 140 caractères ainsi que des liens et des images. Les personnes suivant le compte qui « tweet », les « followers » ou « abonnés », voient alors apparaître les mini-messages sur leur page d'accueil.

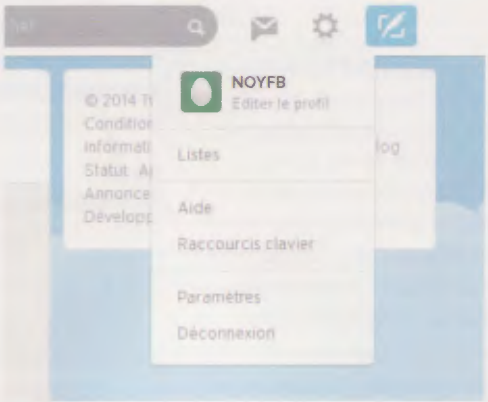
Des messages de 140 caractères peuvent être considérés comme des notifications sachant que l'objectif premier de Twitter est de partager « ce qui se passe » et de répondre à la question « Quoi de neuf ? ». La plateforme Twitter ne se limite pas au site Web mais comprend des applications dédiées pour la plupart à des smartphones et tablettes. Parfaitement intégrée aux différents systèmes mobiles, l'application notifie l'utilisateur de l'existence de nouveaux tweets de la part des personnes suivies.

Vous l'avez compris, il vous sera nécessaire de disposer d'un compte Twitter permettant de tweeter. Mais vous devrez également créer un autre compte, à titre personnel pour recevoir les tweets de votre montage. Si vous possédez déjà un compte régulièrement utilisé, conservez-le et créez-en un nouveau pour votre Arduino. Utiliser un compte « normal » est peu recommandé, vous n'avez certainement pas envie que vos followers reçoivent les notifications de votre ou vos montages comme votre sonnette, la boite à lettre ou la chatière. Notez également que si vous multipliez les montages il ne sera pas nécessaire de créer une multitude de compte. Les tweets eux-mêmes peuvent très simplement préciser textuellement le montage à l'origine de la notification.

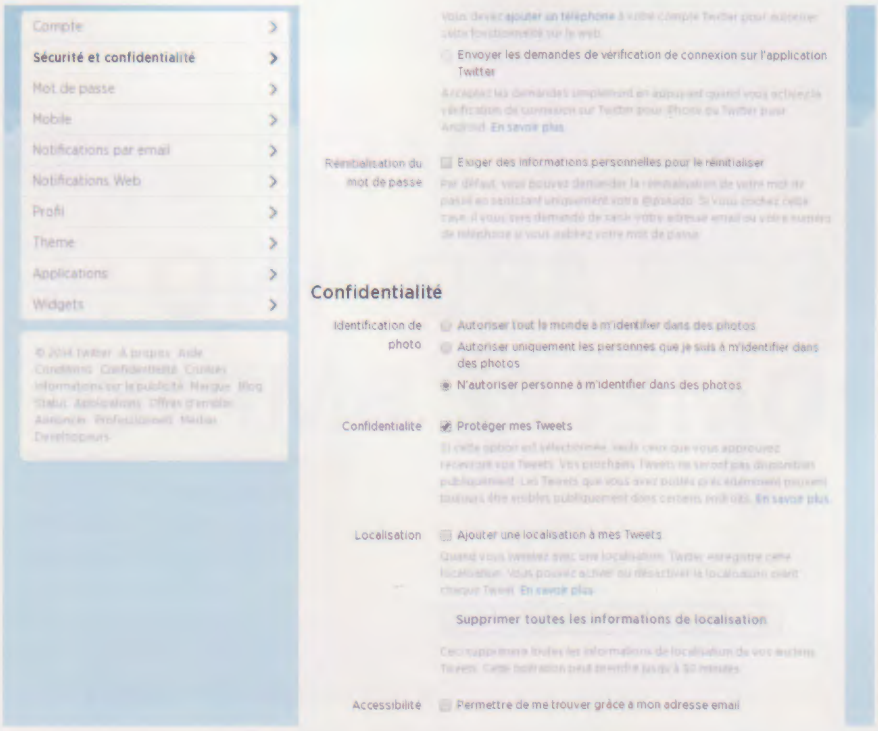
Le principe de fonctionnement de Twitter est majoritairement basé sur une divulgation public des tweets. C'est la raison d'être du système. Cependant, il est permis de configurer un compte de sorte qu'il devienne « privé ». Ceci implique les conséquences suivantes :

- Les utilisateurs doivent faire une demande pour vous suivre,
- vos tweets ne seront visibles que par vos followers,
- les autres utilisateurs ne peuvent retweeter vos tweets,
- vos tweets n'apparaissent pas dans les résultats de recherches des moteurs comme Google, ni dans la recherche sur Twitter,
- l'utilisation d'un nom d'utilisateur dans un tweet ne rendra ce tweet visible qu'aux followers qui vous suivent.

Pour votre ou vos montages, créez tout d'abord un compte Twitter tout à fait classique, confirmez l'adresse mail et activez normalement le compte. Une fois connecté, rendez-vous dans les paramètres du compte en cliquant sur la roue dentée en haut à droite :

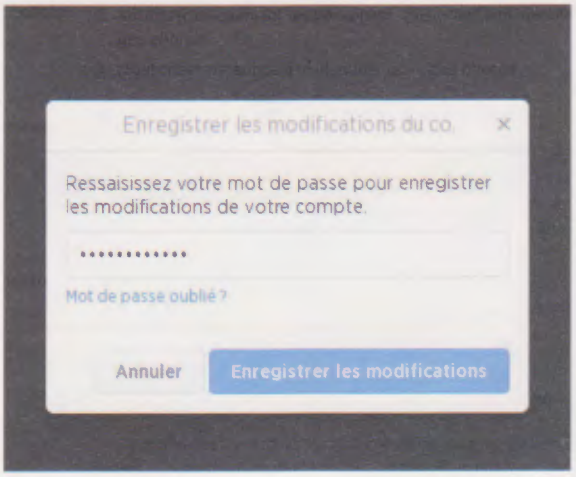


et choisissez « Paramètres » puis « Sécurité et confidentialité » :

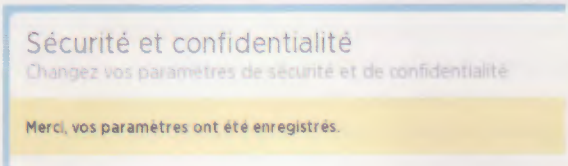


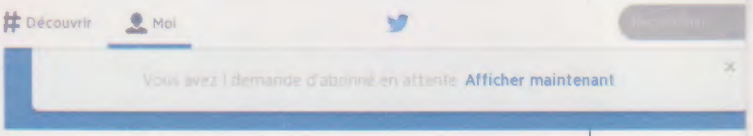
Votre compte est maintenant privé. Utilisez alors un autre périphérique pour vous connecter avec votre compte Twitter personnel ou déconnectez-vous et reconnectez-vous. Vous devrez rechercher le compte privé par son identifiant, l'afficher et vous y abonner pour le suivre. Mais cette demande devra être confirmée. Revenez sur votre compte privé, vous devez voir apparaître une notification concernant cette demande :

Là choisissez « N'autoriser personne à m'identifier dans des photos », « Protéger mes Tweets » et décochez « Permettre de me trouver grâce à mon adresse email ». Puis validez.

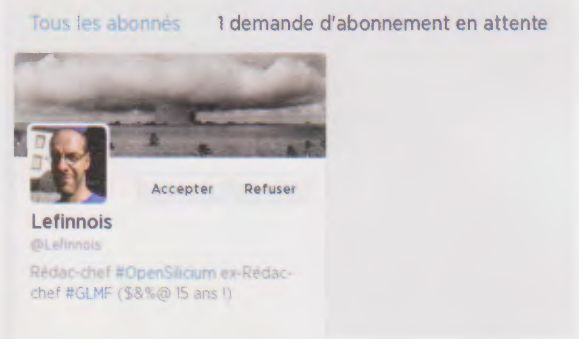


Le mot de passe du compte Twitter vous sera alors demandé pour confirmer l'enregistrement de la configuration et une confirmation sera affichée :





Il vous suffira alors de l'afficher pour découvrir le profil du demandeur (votre compte régulier) :



Vous pourrez alors accepter la demande et ce faisant, le compte en question pourra alors voir les tweets émis depuis le compte privé utilisé par votre montage. Afin de vous en assurer, vous pourrez toujours jongler avec les comptes et faire quelques tests en postant quelques tweets. Ceux du compte privé ne doivent pouvoir être vu que par votre compte personnel préalablement autorisé parmi les followers.

Vous voici prêt pour que vos montages puissent tweeter en toute sécurité... DB

PRÉDIRE LA MÉTÉO AVEC SON ARDUINO : LE BAROMÈTRE BMP180

D. B.

Prédire le temps qu'il va faire est un bien grand mot. Même si la télévision vous promet des prévisions à grands coups d'indices de confiance, le domaine revient dans son ensemble à estimer l'état futur d'un système chaotique afin de permettre aux téléspectateurs crédules de juger de la pertinence d'un achat de charbon de bois, de merguez et autres chipolatas. Au diable les prévisions officielles, mon Arduino pourra sans doute faire aussi bien !



Pour être clair, la prévision du temps qu'il fera ne se résume ni à l'utilisation d'un seul et même instrument, ni à sa simple lecture ponctuelle. Lorsqu'on s'intéresse à la météo, on fait donc généralement usage d'un thermomètre, d'un hygromètre, d'un anémomètre, d'un indicateur de direction des vents, de données visuelles et... d'un baromètre. Ce dernier pèse un poids important dans les prévisions, mais il ne fait pas tout. Cet instrument mesure une caractéristique de l'atmosphère qu'est sa pression, mais c'est la variation de la pression

dans un temps donné qui permet de prévoir, dans une certaine mesure, le temps qu'il fera.

Un baromètre se présente généralement, accompagné d'un thermomètre et d'un hygromètre, sous la forme d'un cadran avec une aiguille indiquant une pression et souvent accompagné des mentions « tempête », « pluie », « variable », « beau temps » et « très sec ». Pour autant, ceci ne fait que présenter des informations complètement inutiles, voire trompeuses. Ce qui compte, je le répète, c'est la variation sur une, deux, ou « n » heures.

L'unité de mesure d'une pression peut être le bar (de manière étonnante, aucune relation ne semble exister entre le bar et la bière pression), mais celle-ci est surtout utilisée pour la pression des pneus, la plongée ou dans l'industrie des gaz. En météorologie, on préfère généralement parler d'hectopascal (hPa), soit 100 pascals. Un pascal étant égal à 1/100000 bar, 100 pascals ou un hectopascal valent donc 1 millibar. La pression moyenne au niveau de la mer est d'environ 1013 hPa. Cette valeur change avec l'altitude (plus on monte moins il y a d'atmosphère au-dessus de votre tête et plus la pression est basse) et fluctue en fonction des changements climatiques.

Voici quelques principes simples de prévision du temps qui donnent généralement de bons résultats :

- si la pression monte rapidement (2 hPa/heure), mais reste en dessous de 1015 hPa, c'est signe qu'une dépression s'éloigne et qu'une embellie se profile ;
- si cela dépasse 1015 hPa (condition dite « anticyclonique »), on peut considérer que le beau temps risque de s'installer pour un jour ou plus ;
- si la pression chute rapidement dans les mêmes échelles, mais reste autour de 1015 hPa, on peut s'attendre à un orage ;
- si la pression baisse lentement alors qu'elle se trouve déjà sous 1015 hPa, c'est signe de temps gris et de pluie ;
- enfin, si la pression s'effondre rapidement et passe d'une valeur au-dessus de 1015 hPa, puis 1000 hPa ou 980 hPa, tous aux abris, car c'est une tempête qui se prépare !
- Accessoirement, si votre baromètre affiche une pression en dessous de 900 hPa, il serait judicieux de laisser votre Arduino tranquille et vous occuper de votre survie parce que vous vous trouvez au centre d'un ouragan de classe 5 avec des vents dépassant 250 km/h.

1. BMP180

Ce montage utilise un capteur de pression Bosh BMP180 monté sur un petit circuit d'adaptation. Sa connexion est fort simple puisqu'il n'y a que 4 connecteurs : la masse, l'alimentation, SDA et SCL. Vous l'avez sans doute compris à la désignation des deux dernières broches, il s'agit d'une interface i2c. Les cartes Arduino récentes proposent directement ce bus et il vous suffira donc de connecter cela broche à broche. Attention toutefois, le BMP180 seul est un capteur 3,3 volts ne tolérant pas 5 volts. Il faudra vous assurer, quel que soit l'endroit où vous vous procurez un tel module, que les niveaux de tension sont bien adaptés. C'est le cas, par exemple du module vendu par Sparkfun (entrée/sortie tolérant 5V, mais alimentation en 3,3V), ainsi que du produit correspondant chez Adafruit (E/S et alimentation en 5V), mais également du module que nous avons trouvé sur eBay (vendeur *alice1101983*), 5 fois moins cher que chez Sparkfun et Adafruit, livraison gratuite directe depuis ShenZhen !

Comme le module est d'une simplicité extrême et le capteur très standard, la bibliothèque fournie par Sparkfun permettra de parfaitement faire fonctionner ce module chinois. Il suffit pour cela de vous rendre sur **https://github.com/sparkfun/BMP180_Breakout** puis de télécharger et d'installer la bibliothèque dans votre répertoire de croquis.

Pour ce projet, nous utiliserons également un afficheur LCD compatible HD44780 très standard en 16*2 caractères qui lui est parfaitement déjà pris en charge par l'environnement de développement Arduino. Bien entendu, ce n'est là qu'un exemple, vous pouvez tout aussi bien utiliser le moniteur série ou tout autre système d'affichage. Nous ne reviendrons pas ici sur la connectique de ce type d'afficheur LCD, déjà détaillée dans le précédent numéro (article sur la sonnette intelligente).

2. LE CROQUIS

Fichier Édition Croquis Outils Aide

```
#include <LiquidCrystal.h>
#include <SFE_BMP180.h>
#include <Wire.h>

#define LCD_RS 2
#define LCD_E 3
#define LCD_D4 4
#define LCD_D5 5
#define LCD_D6 6
#define LCD_D7 7

#define ALTITUDE 165.0

LiquidCrystal lcd(LCD_RS,LCD_E,LCD_D4,LCD_D5,LCD_D6,LCD_D7);

SFE_BMP180 pressure;

long previousMillis = 0;
int firstrun = 1;
double logpression[96];

void setup() {
  pinMode(8, OUTPUT);
  lcd.begin(16, 2);
  if (!pressure.begin()) {
    lcd.print("ERREUR BMP180");
    while(1);
  }
}

void logpush(double P) {
  for(int i=95; i>0; i--) {
    logpression[i] = logpression[i-1];
  }
  logpression[0] = P;
}

void printdelta(int cellule) {
  char texte[6];
  if(logpression[cellule]==0) {
    lcd.print("____");
    return;
  }
  dtostrf(logpression[0]-logpression[cellule], 4, 0, texte);
  lcd.print(texte);
}

void loop() {
  char status;
  double T,P,p0;
  char ligne[16];
```

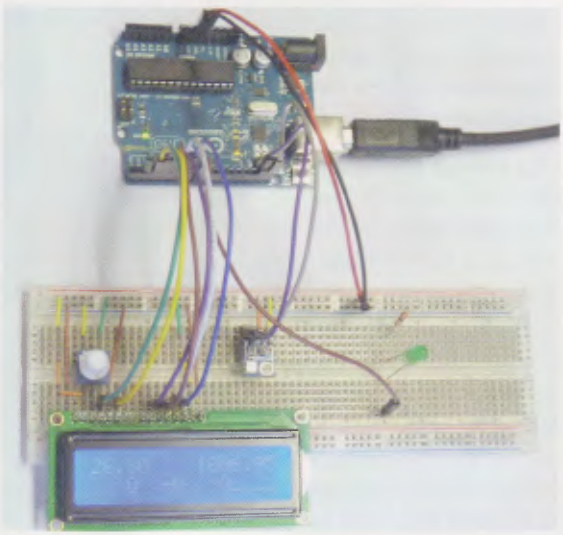
```
  unsigned long currentMillis = millis();

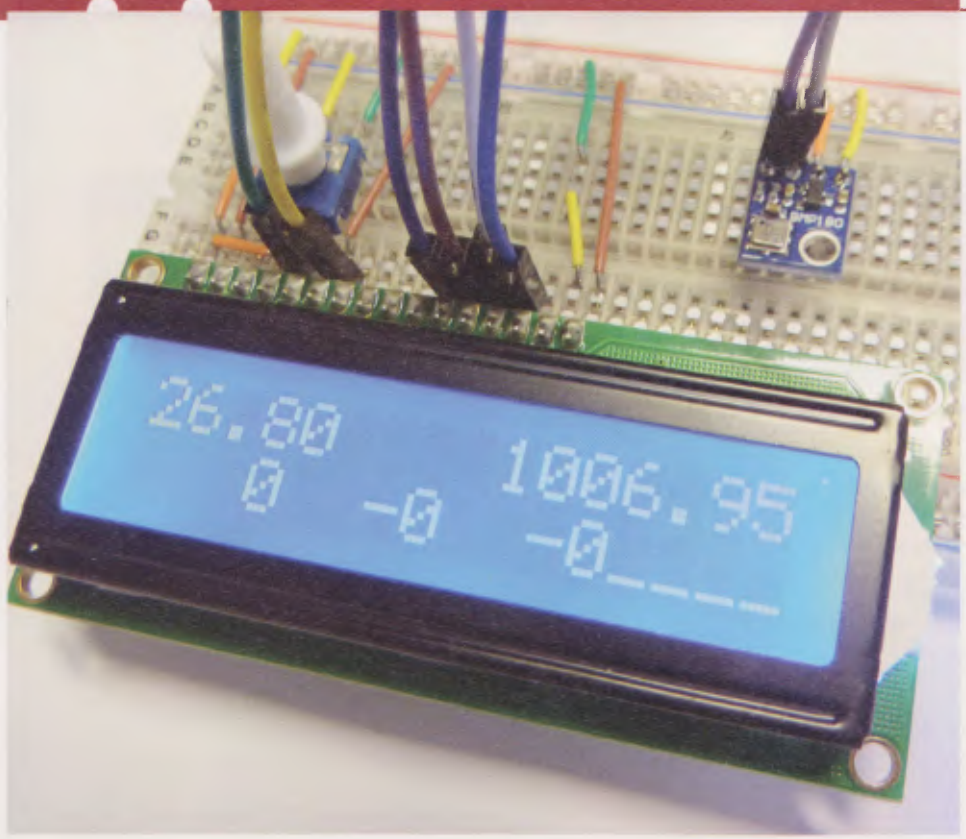
  if(currentMillis - previousMillis > (300000UL) || firstrun) {
    digitalWrite(8, HIGH);
    lcd.clear();
    status = pressure.startTemperature();
    if(status != 0) {
      delay(status);
      status = pressure.getTemperature(T);
      if (status != 0) {
        delay(status);
        status = pressure.startPressure(3);
        if(status != 0) {
          delay(status);
          status = pressure.getPressure(P,T);
          if(status != 0) {
            p0 = pressure.sealevel(P,ALTITUDE);
            dtostrf(T, 5, 2, ligne);
            lcd.print(ligne);
            dtostrf(p0, 11, 2, ligne);
            lcd.print(ligne);
            lcd.setCursor(0, 1);
            logpush(p0);
            printdelta(30/5);  // 30mn
            printdelta(60/5);  // 1h
            printdelta(90/5); //  1h30mn
            printdelta(120/5); // 2h
          }
        }
      }
    }
    if(firstrun) firstrun=0;
    previousMillis = currentMillis;
    digitalWrite(8, LOW);
  }
}
```

Arduino

Plusieurs points intéressants sont à relever dans ce croquis. Nous avons tout d'abord le fonctionnement même du BMP180 avec la démarche suivante :

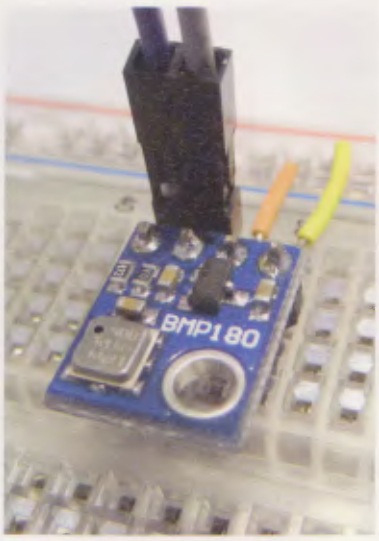
- Un premier appel à `pressure.startTemperature()` qui déclenche la mesure de la température et retourne une durée d'attente directement utilisée avec `delay()` ou `0` en cas d'erreur.
- Un second appel permettant de récupérer la température, car la mesure de pression doit être compensée en fonction de cette valeur.
- On déclenche ensuite la mesure de pression en spécifiant la valeur d'*oversampling* (précision entre 0 et 3) et, là encore, on utilise la valeur retournée pour attendre la fin de l'opération.





- On peut ensuite récupérer la mesure avec un nouvel appel utilisant en argument la température obtenue précédemment.
- Enfin, comme la mesure est faite de manière absolue, il faut ajuster la valeur en fonction de l'altitude en utilisant la fonction utilitaire `pressure.sealevel()` pour obtenir la mesure finale dans `p0`. Vous pouvez récupérer facilement votre altitude actuelle en utilisant le GPS de votre smartphone.

La logique de fonctionnement du croquis consiste à remplir un tableau de 96 valeurs de pression à la fréquence d'une mesure toutes les 5 minutes (`300000UL` millisecondes). La fonction `logpush()` nous permet de décaler les valeurs dans le tableau et d'insérer la dernière dans `logpression[0]`. Pour afficher les écarts de pression sur l'écran LCD, nous avons une fonction `printdelta()` qui affiche simplement la différence entre la valeur dans `logpression[0]` (la plus récente) et celle dans la cellule située à cellule*5mn. On peut donc afficher la différence entre la pression actuelle et celle mesurée jusqu'à il y a 95*5 = 480 minutes, soit 6 heures. Comme l'espace sur un LCD 16*2 est réduit, il n'y a de la place en seconde ligne que pour 4 écarts affichés en valeur entière.



Notez l'utilisation de la fonction `dtostrf()` provenant de l'avr-libc incluse dans l'environnement Arduino. Il n'est en effet pas possible d'utiliser quelque chose comme `sprintf(chaine, "%l", valeur_double)`, le spécificateur de format `%l` n'étant pas disponible pour convertir un `double` en chaîne de caractères (*string*).

Bien entendu, il vous est possible d'ajuster ce croquis en réduisant la fréquence des mesures ou en augmentant la taille du tableau pour travailler sur une plus grande période de temps. Et parmi les autres évolutions possibles, nous avons, par exemple :

- l'envoi des valeurs via une liaison série au PC ;
- le stockage des mesures dans l'EEPROM (1 Ko) ;
- l'utilisation d'une mémoire externe (flash, SD, EEPROM SPI) ;
- l'ajout d'un shield Ethernet/Wifi/GSM pour l'envoi via le réseau ;
- le remplacement du LCD HD44780 par un modèle graphique ;
- l'ajout d'un buzzer pour déclencher une alerte sonore en cas de chute rapide de la pression ;
- utiliser la pression atmosphérique pour déterminer l'altitude (principe de fonctionnement des altimètres),
- etc.

N'hésitez pas à nous faire part de vos expérimentations et vos créations autour du BMP180. DB

LES ANTHOLOGIES DE... NOUVEAU ! NOUVEAU ! NOUVEAU ! NOUVEAU !

COLLECTION

DÉCOUVREZ
LA COLLECTION D'ANTHOLOGIE*
DES ÉDITIONS DIAMOND...



39€
uniquement disponible
EN VERSION PDF

+ DE 300 PAGES !

* Adaptation et révision d'articles parus dans GNU/Linux Magazine, entièrement mis à jour pour Android 4 !

VENEZ DÉCOUVRIR LE SOMMAIRE ET LES EXTRAITS
À PARTIR DU 1ER SEPTEMBRE 2014 SUR
boutique.ed-diamond.com

C'EST AUJOURD'HUI DEMAIN ?

AU PETIT ECHO DE LA MODE



FÊTE DE LA SCIENCE

LE 4 OCTOBRE

DECOUVERTE DES NOUVELLES PRATIQUES NUMERIQUES :

- Imprimante 3D
- Réalité virtuelle/réalité augmentée
- Robotique
- Makey makey
- Projets innovants

Entrée gratuite

10h-18h

Châtelaudren

Découvrez aussi au Médiacentre :
27/09 : l'Ile des sciences, visite guidée en ligne
9/10 : Arduino, initiation à la domotique et à la robotique

Tél : 02 96 79 74 64 - médiacentre@leffcommunaute.fr

Liberté • Égalité • Fraternité
RÉPUBLIQUE FRANÇAISE
MINISTÈRE DE L'ÉDUCATION NATIONALE, DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE

HACKABLE MAGAZINE
Partenaire Presse

Le Leff communauté