



HACKABLE

MAGAZINE

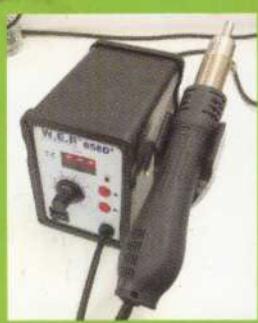
DÉMONTEZ | COMPRENEZ | ADAPTEZ | PARTAGEZ

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

~~ SOUDAGE / AIR ~~

Réparez et modifiez des circuits modernes avec une station de soudage à air chaud à moins de 30€

p. 04



~~ ESP8266 / M.A.J. ~~

Oubliez la programmation USB et ordonnez la mise à jour automatique de vos ESP8266

p. 12



~~ HISTOIRE / TECH ~~

La fabuleuse histoire des calculateurs numériques à l'ère électromécanique

p. 72

Réceptionnez des images satellite !

p. 24

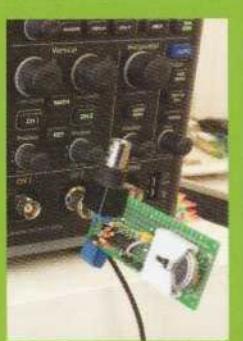


- Comprenez le jargon et les principes
- Construisez votre antenne
- Captez les signaux et obtenez les images

~~ RADIO / CÂBLES ~~

Mesurez la vitesse de la lumière dans les câbles ou comment identifier un coax avec votre oscilloscope

p. 64



NOUVELLE RUBRIQUE

~~ RETRO TECH ~~

Remplacez le vieillissant disque SCSI de votre machine vintage par une carte microSD grâce au SCSI2SD

p. 88



L 19338 - 25 - F: 7,90 € - RD



~ ÉDITO ~



C'est l'été !

En été, on a normalement du temps et, pour certains, une subite envie de sortir, de faire du sport et de grimper sur les toits (si, si). Ça tombe bien puisque (oups) car, si déjà vous êtes là-haut, vous pouvez en profiter pour installer une antenne, voire plusieurs.

Soyons honnêtes entre nous et moi, à quoi servent encore les mâts d'antenne de nos jours ? Pour ceux qui possèdent encore une TV pour une utilisation autre que faire office de moniteur, la réception se fait pour 55% des foyers via le une connexion filaire type ADSL, fibre ou câble (source CSA, « L'équipement audiovisuel des foyers aux 3ème et 4ème trimestres 2017 »). Nous devons accepter les faits, il existe toute une population de mâts d'antenne en pleine crise existentielle, contraints de supporter de vétustes « râteaux », mal orientés, en décrépitude, voire déconnectés et tout simplement oubliés.

Être un mât d'antenne est une vie difficile, éprouvante et ingrate. Mais vous pouvez donner un sens à leur existence en les gratifiant d'une nouvelle utilisation, plus prestigieuse, plus atypique, consistant à supporter des antennes conçues pour capter des émissions que tout le monde ne reçoit pas ou ne se doute même pas de leur relative accessibilité.

C'est donc dans cet état d'esprit bienveillant que je vous invite à découvrir le sujet principal de ce numéro traitant de la réception d'images provenant de satellites d'observation météorologique... Tout en présentant mes plus plates excuses à mon propre mât pour avoir temporairement placé mon antenne « maison » non pas sur le toit, mais dessous. Ne t'inquiètes pas mon grand, je pense à toi <3

Avant de vous laisser dévorer ce numéro, je profiterai de cet édito pour préciser aux lecteurs ayant suivi la série sur le Z80 que cette interruption fortuite ne durera pas, le sujet reviendra dès le prochain numéro.

Bonne lecture et bel été plein d'images !

Denis Bodor

Hackable Magazine

est édité par Les Éditions Diamond



10, Place de la Cathédrale - 68000 Colmar
Tél. : 03 67 10 00 20 - Fax : 03 67 10 00 21

E-mail : lecteurs@hackable.fr
Service commercial : cial@ed-diamond.com

Sites : <https://www.hackable.fr/>
<https://www.ed-diamond.com>

Directeur de publication : Arnaud Metzler

Rédacteur en chef : Denis Bodor

Réalisation graphique : Kathrin Scali

Responsable publicité : Valérie Fréchard,

Tél. : 03 67 10 00 27 v.frechard@ed-diamond.com

Service abonnement : Tél. : 03 67 10 00 20

Impression : pva, Landau, Allemagne

Distribution France : (uniquement pour les dépositaires de presse)

MLP Réassort : Plate-forme de Saint-Barthélemy-d'Anjou. Tél. : 02 41 27 53 12

Plate-forme de Saint-Quentin-Fallavier.
Tél. : 04 74 82 63 04

Service des ventes : Abomarque : 09 53 15 21 77

IMPRIMÉ en Allemagne - PRINTED in Germany

Dépôt légal : À parution,

N° ISSN : 2427-4631

Commission paritaire : K92470

Périodicité : bimestriel

Prix de vente : 7,90 €

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.



MIXTE
Papier issu de sources responsables

FSC® C015136

Suivez-nous sur Twitter
 @hackablemag



HACKS, HACKERS & HACKABLE

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

~ SOMMAIRE ~

ÉQUIPEMENT

04

Une station de soudage à air chaud à moins de 30€ ?

ARDU'N'CO

12

Vos ESP8266 se mettent à jour tout seuls !

EN COUVERTURE

24

Introduction à la réception d'images satellite

42

Se créer une antenne pour recevoir les images satellites

52

Réception de vos premières images satellite

RADIO & FRÉQUENCES

64

Mesurez la vitesse de la lumière dans les câbles !

REPÈRE & SCIENCE

72

La fabuleuse histoire des calculateurs numériques à l'ère électromécanique

TENSIONS & COURANTS

82

Les capteurs photovoltaïques

NOUVELLE RUBRIQUE

RETRO TECH

88

SCSI2SD ou comment remplacer un disque SCSI par une carte SD

ABONNEMENT

59/60

Abonnements multi-supports



ÉQUIPEMENT

SOUDURE

UNE STATION DE SOUDAGE À AIR CHAUD À MOINS DE 30€ ?

Denis Bodor



Dans les numéros précédents, nous avons parlé de l'indispensable station de soudage (ou de soudure), de la station de dessoudage et de différentes techniques permettant de faciliter les manipulations généralement utilisées pour réparer un circuit ou récupérer des composants. Ces matériels trouvent cependant leur limite lorsqu'il s'agit de travailler avec des composants montés en surface (SMD). Pour ce cas particulier, de plus en plus fréquent, il nous faut un tout autre équipement : une station de soudage à air chaud.

Une petite parenthèse pénible avant de commencer, afin de clarifier un point de terminologie. Faut-il parler de « station de soudure » ou de « soudage » ? De but en blanc, j'aurai tendance à répondre qu'il ne faut pas en parler, mais plutôt l'utiliser, c'est tout de même plus pratique pour avoir un résultat... Mais c'est là un commentaire qui revient souvent. Le fait est que la soudure est le résultat du soudage. En principe donc, on fait une soudure et non un soudage. Quid de l'appareil alors ? Permet-il l'action de souder et donc le soudage, ou permet-il d'obtenir le résultat et donc une soudure ?

Personnellement j'ai décidé d'appliquer la logique suivante : comme personne ne parle de « station de dessoudure », il ne peut y avoir que des stations de dessoudage. Or, dans ce cas, il paraît logique que l'opération inverse soit donc possible avec une station de soudage et non de soudure. Le seul point problématique cependant est le fait que « dessoudure » est effectivement dans le dictionnaire, mais pas « dessoudage ». Cependant je pense que, peut-être, les gens qui perdraient du temps à tergiverser sur le sujet devraient plutôt le passer à souder et dessouder qu'à en parler...

Les anglophones ont réglé ce problème autrement, le matériel dont il sera question ici est tout simplement qualifié de *Hot-Air Rework Station* ou « station de retravail à air chaud » (eh non, « retravail » n'est pas français non plus). La notion de

retravailler une soudure découle de l'utilisation primaire du matériel puisque nous parlons ici de composants de surface et forcément de reprise en main d'un circuit fabriqué d'une autre façon.

La technique utilisée en production pour ce genre d'assemblage ne consiste pas à utiliser un outil de soudage avec chaque composant. La production consiste en une série d'étapes qui peuvent être sommairement résumées ainsi :

- On dispose, sur le circuit imprimé terminé (gravé, découpé, percé, sérigraphié, verni, etc.), de la pâte de soudure grâce à un pochoir aux emplacements où les soudures devront être faites. Cette pâte ou crème à braser est constituée de flux et de micro-billes d'alliage de soudure (généralement qualifiés d'étain).
- Les composants de surface sont ensuite placés précisément là où ils doivent trouver place, à l'aide d'un robot « *pick and place* » (« prendre et placer »). C'est l'opération que l'on voit souvent présentée dans les vidéos impressionnantes sur l'industrie électronique, visant à montrer comme tout ceci est moderne et automatisé.
- Le circuit entier est ensuite placé dans un four à refusion afin de faire fondre la soudure qui se glisse par capillarité entre le circuit et les composants, et les soudes. Cette opération est réalisée en suivant une procédure précise et en contrôlant finement la température tout au long du processus. L'objectif est d'appliquer un profil de température en plusieurs étapes (préchauffage, séchage du flux, refusion et refroidissement) de manière à réaliser une soudure parfaite, sans stresser les composants ou le circuit imprimé.



La station de soudage à air chaud modèle 858D+ est compacte et surtout très économique. Sans aller jusqu'à dire qu'il s'agit d'une qualité professionnelle, elle est plutôt de bonne facture et permettra des travaux tout à fait convenables avec un peu de pratique.



Le panneau frontal affiche la température réglable via les deux boutons rouges à droite. Le débit d'air sera ajusté avec le potentiomètre sur la gauche. Notez la présence de ce qui semble être un bouton libellé « CAL », mais qui est en réalité un cache en plastique souple dissimulant l'accès à un potentiomètre miniature permettant de calibrer la sonde de température.

D'une certaine manière, on peut considérer que cette technique est à sens unique, car non réversible par des moyens identiques. Placer un circuit terminé dans un four à refusion pour remplacer un unique composant n'est ni qualitativement fiable, ni pratique. D'autre part, un outil « classique » comme une station de soudage ou de dessoudage (voir Hackable n°18) ne permettra pas de faire le travail. Tenter d'aspirer l'étain en fusion des pattes d'un composant de surface ne sert pas à grand-chose et pour les plus petits composants (résistances, diodes, condensateurs), ils peuvent tout simplement être aspirés. Il en va de même pour la panne du fer à souder. Quiconque a déjà tenté l'opération connaît la triste sensation de voir le composant rester sur la panne et brûler tristement.

Bien entendu, comme je l'avais évoqué dans un précédent article, tout dépend de ce que vous cherchez à faire : récupérer un composant ou vous débarrasser de celui-ci pour le remplacer par un nouveau.

1. SOUDAGE ET DESSOUDAGE À L'AIR CHAUD

Puisqu'il est très difficile, sinon impossible, d'utiliser une panne pour transférer la chaleur sur le circuit et le composant, simultanément à tous les endroits clés, une autre technique a été développée : transférer cette énergie thermique via un autre média, l'air. Sommairement, une station de « retravail » à air chaud n'est rien d'autre qu'un sèche-cheveux ou un décapeur thermique dont le comportement est contrôlé avec précision. Deux éléments clés sont à ajuster : la température et le débit d'air.

La température, comme avec un fer à souder, impacte la fusion de l'étain, bien sûr, mais aussi les dommages que peuvent subir les composants. Les documentations techniques (*datasheets*) indiquent une température maximum de refusion ou PRT pour Peak

Reflow Temperature ainsi qu'une référence à un profil standardisé. Ces deux éléments peuvent donner une indication sur la température à utiliser pour dessouder le composant.

Il est important de comprendre que le contexte est très différent de, par exemple, la soudure au fer à souder ou le dessoudage de composants traversants. Ici, c'est une large zone qui sera chauffée et non simplement une patte à la fois. De plus, nous parlons de composants de surface et travaillons donc du côté où se trouve le composant et non de l'autre côté du circuit qui sert alors de protection thermique. Certains composants ne supporteront tout simplement pas une température de quelques 350°C, et encore moins durant plusieurs dizaines de secondes. D'autres fonctionneront toujours, mais leur durée de vie aura été grandement réduite.

Le flux d'air, quant à lui, est d'une importance toute autre. C'est son débit qui va distribuer la chaleur. Comme la température, celui-ci est réglable sur l'appareil, mais ce n'est pas le seul élément. Le plus souvent, une telle station est livrée avec plusieurs buses (ou nozzle en anglais) dont les diamètres, et parfois les formes, sont différents. Tout naturellement à débit égal, le souffle sera plus important avec une buse fine qu'avec une buse de plus gros diamètre.

Cette dispersion de chaleur sera choisie de manière à avoir un subtil équilibre entre la vitesse avec laquelle on dessoudera le composant et les effets sur les composants à proximité. C'est une

erreur de débutant classique que de littéralement souffler au loin les petits composants comme les condensateurs et les résistances en tentant de dessouder un circuit intégré, tout simplement parce que le débit d'air est trop important.

L'utilisation d'une station à air chaud n'est donc en soi pas très compliquée. C'est avant tout une affaire de pratique et de réglage de ces deux simples paramètres. Contrairement à un fer à souder, où la chaleur est distinctement localisée, ici nous travaillons avec quelque chose de plus diffus qui nécessite une plus grande prudence de notre part. Ceci est valable non seulement pour nous-mêmes, mais également pour tout ce qui se trouve aux alentours de l'endroit de la manipulation. Il est généralement d'usage de prévoir un support de travail capable d'accepter une éventuelle montée accidentelle en température.

Dans le même ordre d'idées en rapport avec l'environnement de travail, la mise à la terre est également un point important. Ceci est valable pour vous, comme pour votre plan de travail et, bien entendu, le matériel lui-même (ce qui est normalement le cas par défaut, mais cela ne coûte rien de vérifier). Au-delà de la sécurité, il faut savoir que la circulation d'air, entraînant le déplacement des particules qui s'y trouvent en suspension, génère des charges électrostatiques. C'est pour cette raison que certains aspirateurs avec un corps métallique (type modèle professionnel gros travaux) intègrent une protection.

Une mise à la terre est donc importante et, dans tous les cas, il n'est pas question de toucher le circuit ou le composant avec la buse.

Ce qui nous amène enfin, au compagnon indispensable d'une station à air chaud : une paire de brucelles de bonne qualité. Certes, vous pouvez trouver des pinces pour presque rien, ou même utiliser ce qui vous tombe sous la main, mais je vous garantis que des brucelles de précision, de marque et de bonne facture vous changent la vie. Ce n'est que l'affaire d'une dizaine d'euros, mais dès la première prise en main, vous comprenez que la dépense est justifiée.

2. ATTEN 858D+

Comme je l'ai dit précédemment, une station à air chaud n'est rien d'autre qu'un espèce de sèche-cheveux contrôlé en température et en débit d'air. Ce qu'on attend de ce genre de matériel se résume à peu de choses :

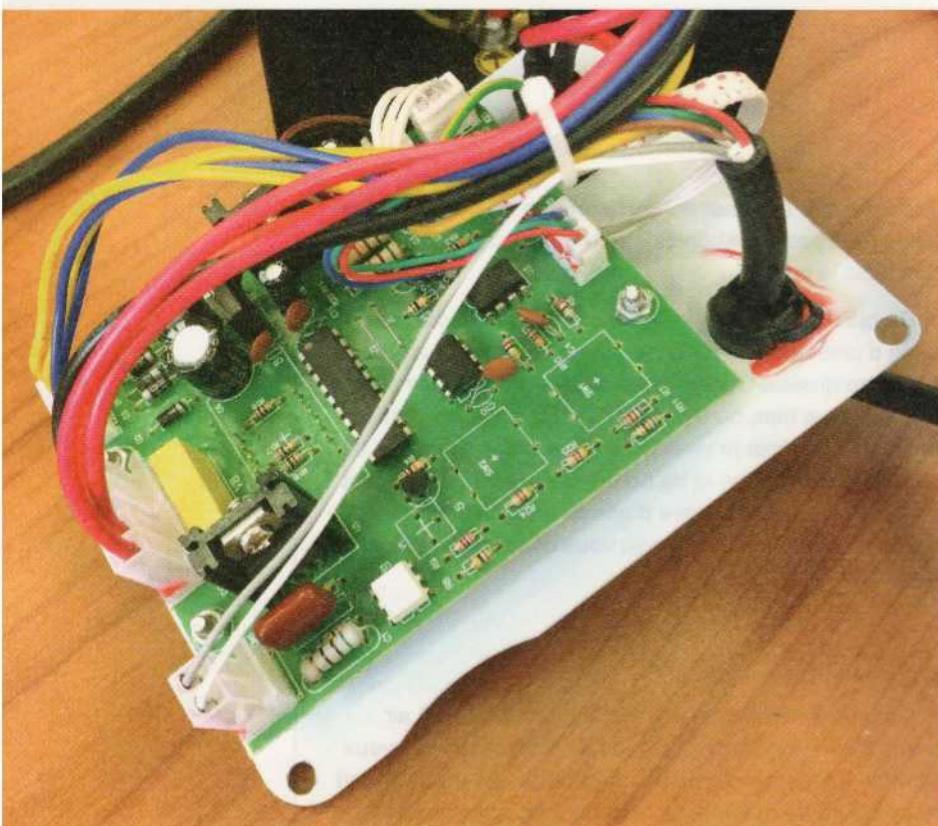
- robustesse,
- un minimum de précision en température,
- fiabilité,
- rapidité de mise en route (montée en température).

Certains ajouteraient sans doute des choses comme la plus faible nuisance sonore possible, divers éléments fonctionnels et ergonomiques ou encore un encombrement réduit, mais nous nous en tiendrons ici au minimum exigé. En particulier en gardant à l'esprit que nous parlons ici d'un équipement qu'il est possible de trouver pour environ 30€ port inclus.

Le matériel en question s'appelle tantôt Atten 858D+, mais de nombreuses variations existent aussi bien sur le nom du fabricant, le modèle, l'aspect, les accessoires fournis et, bien entendu, les spécifications annoncées. L'exemplaire donc je dispose provient d'un vendeur eBay allemand ayant pour nom « buy-freely365 », et a été acquis pour 33€. Avec le recul, les quelques 10€ de port pour une livraison par DHL depuis Germania furent une pure perte puisque le

Fixée sur la base de la station se trouve la source d'alimentation de l'ensemble : ~230V domestique pour la résistance chauffante, ~29V pour le moteur de la turbine (réglé en 24V) et ~10V pour le reste du circuit (réglé en 5V).





Derrière le panneau frontal se trouve le circuit imprimé regroupant les composants chargés du contrôle de température et de la régulation du débit d'air, ainsi que de l'affichage. L'ensemble est de relativement bonne qualité générale et propre.

colis, à peine arrivé en France une première fois, est reparti vers l'expéditeur pour une raison mystérieuse. Une livraison depuis un vendeur chinois eut été finalement plus rapide et plus économique (vous avez déjà remarqué de souvent, une livraison nationale prend parfois plus de temps qu'une expédition depuis Shenzhen ?).

L'équipement est livré dans une belle boîte contenant l'appareil lui-même, déjà entièrement assemblé et un lot de trois buses (deux rondes de 8 mm et 10 mm, et une carrée de 12 mm). La base de la station est un pavé de 15 cm de profondeur, par 13 de haut et 10 de large, disposant d'un support pour le pistolet, vissé à sa droite. On remarquera au passage que ce support est maintenu par une équerre et 6 vis qui peut être démontée pour être fixée à gauche de la base (pour les ressortissants du monde miroir également appelés gauchers).

L'architecture est relativement simpliste et contrairement à des appareils plus coûteux, le flux d'air est généré directement dans le manche du « pistolet » à l'aide d'une turbine contrôlée par la station. L'ensemble s'en trouve plus difficile à manipuler, car plus encombrant et il faut prendre soin de ne pas couvrir la prise d'air en tenant le manche. L'avantage de cet agencement, en dehors des coûts de production

réduits, est de n'avoir qu'un seul lien, fin, avec la base, contrairement à d'autres modèles qui intègrent la turbine dans la base et transportent le flux d'air via un tube souple d'un diamètre plus important (~20 mm).

Le panneau en façade de la station dispose :

- d'un interrupteur général directement relié à l'alimentation domestique (~230V),
- d'un réglage rotatif libellé de 1 à 8 permettant d'ajuster le flux d'air,
- de deux boutons pour augmenter ou diminuer la température de consigne,
- et d'un afficheur à leds 7 segments de trois chiffres affichant alternativement la température de consigne au moment du réglage et la température courante.

Lors de la mise sous tension, l'appareil ne produit ni flux d'air, ni de chaleur. En effet, le manche contient un interrupteur reed (ou interrupteur à lames souples, ILS) et le support un aimant. Ce n'est qu'en manipulant l'appareil que celui-ci se met effectivement en route et monte en température. Fort heureusement, ceci s'avère très rapide et en quelques secondes l'appareil peut être utilisé. Notez au passage que l'arrêt n'est pas immédiat. En effet, lorsque l'on pose le pistolet sur le support, l'alimentation de la résistance chauffante est coupée, mais le flux d'air perdure de manière à refroidir l'ensemble avant un arrêt complet.

L'arrière de l'appareil présente un porte-fusible à visser équipé d'un fusible de 6A et le cordon d'alimentation est serti dans le boîtier. On aurait apprécié un connecteur CEI type C14 comme on en trouve généralement sur les ordinateurs, mais à ce prix, il serait un peu déplacé de faire la fine bouche, d'autant que l'appareil est spécifique à une alimentation 220V-240V.

À propos de prix, justement, on notera avec surprise qu'un système de code d'authentification du matériel est proposé via une étiquette disposant d'une zone à gratter couvrant partiellement un QR code. Il vous suffira alors de gratter et de scanner ce code avec votre smartphone pour être envoyé directement sur le site de *Yihua Electronic Equipment* où le code sera validé et un message vous précisera normalement l'authenticité effective du matériel. Une seconde tentative vous signalerà que ce code a déjà fait l'objet d'une vérification et vous alertera sur le risque de contrefaçon. Je trouve ce déploiement d'efforts, pour un matériel si peu coûteux, très agréablement surprenant.

Afin de vérifier si ce petit détail rassurant est justifié, rien de tel qu'un petit coup d'œil à l'intérieur du matériel. L'ensemble se démonte sans grande difficulté en retirant les panneaux avant et arrière à l'aide de quelques vis, puis en faisant de même avec la partie supérieure également maintenue par deux vis (une de chaque côté à l'arrière) et deux écrous/boulons (vers l'avant). Ce désassemblage

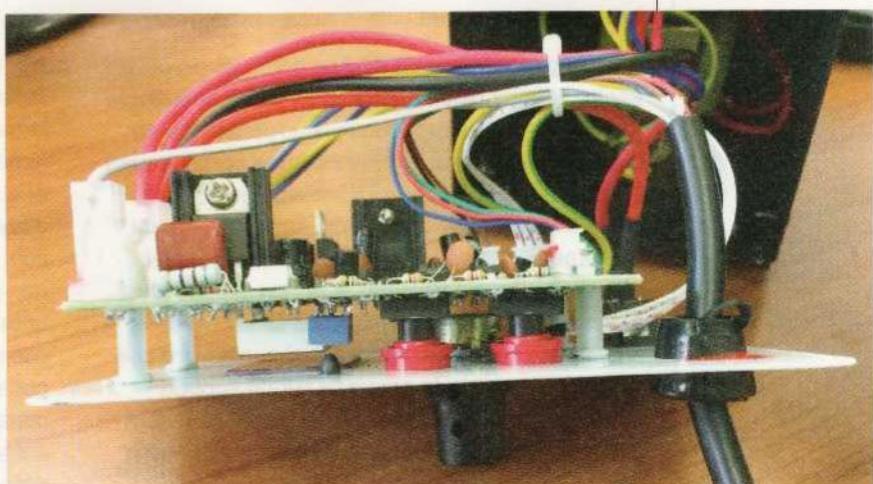
révèle un transformateur (230-240V vers 10V et 29V) fixé sur le fond de la station, connecté à un circuit se plaçant derrière le panneau frontal.

Ce circuit est construit autour d'un microcontrôleur S3F94C4 à cœur SAM88RCRI produit par Zilog. Oui, oui, Zilog ! La famille de microcontrôleurs S3 se veut compatible avec l'architecture Z8 du même constructeur qui, bien que très différente de celle du mythique Z80, car basée sur une architecture Harvard (mémoire programme et de données séparées), reprend une bonne partie du jeu d'instructions utilisé de longue date.

C'est très surprenant de trouver là un tel microcontrôleur, car les constructeurs chinois utilisent généralement des microcontrôleurs reposant sur un cœur Intel 8051 dont il existe bon nombre d'implémentations libres de droits, ou éventuellement des microcontrôleurs Atmel/Microchip dont on peut tantôt douter de la provenance. À noter au passage que certaines prises en main, pages web et vidéos en ligne (dont celle de Dave Jones datant de 2011, EEVblog #167), décrivent un matériel visuellement, nominativement et fonctionnellement identique, mais basé sur un microcontrôleur Atmel AVR. Force est de constater donc que ce matériel existe depuis bien longtemps et semble avoir eu de nombreuses révisions, comme en témoigne l'inscription sur le circuit imprimé : « YH858D_V8 » (sans doute la huitième version de ce circuit).

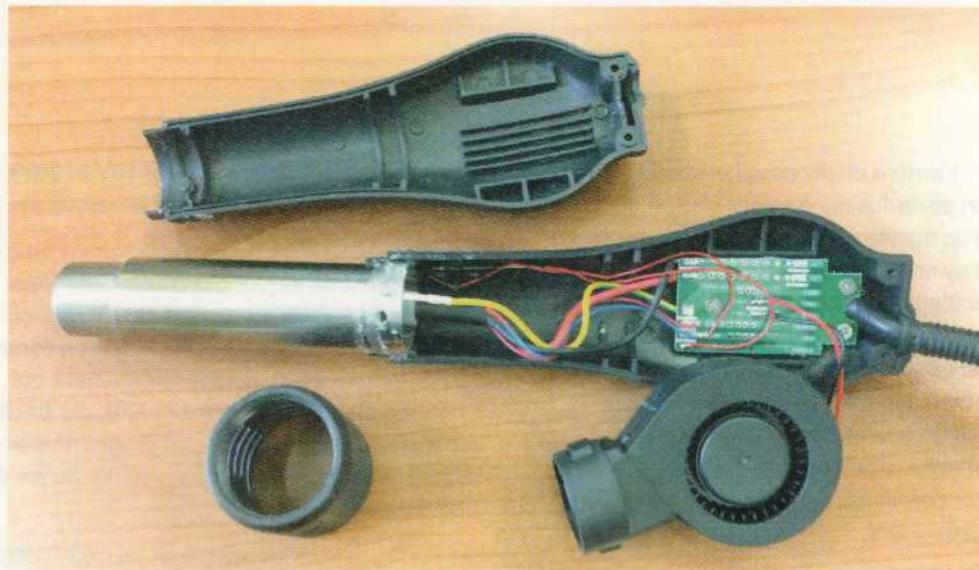
Quoi qu'il en soit, il n'y a pas grand intérêt à chercher à développer un firmware alternatif pour un aussi simple matériel et cette exploration est surtout destinée à s'assurer de

La plupart des composants se trouvent d'un seul côté du circuit, seuls les boutons, afficheurs et potentiomètres sont soudés côté cuivre. Notez la présence, en plein milieu de l'image, du potentiomètre (bleu) de calibration permettant de faire correspondre plus finement la température affichée avec celle en sortie de la buse.





Contrairement à d'autres équipements similaires, la turbine est ici placée dans le manche et non dans la base de l'appareil. Ceci rend l'ensemble plus encombrant, mais évite d'avoir à utiliser un flexible souple pour faire circuler l'air entre la base de l'appareil et la buse.



la qualité générale. Et celle-ci est au rendez-vous : connecteur pour chaque partie des liaisons avec le circuit imprimé, mise à la masse vissée au châssis, boulon sécurisé avec de la résine, colliers de serrage sur les câbles, soudures propres...

Le circuit imprimé est simple face et la majorité des composants se trouvent vers l'intérieur du boîtier. Cependant, l'afficheur, les boutons et le potentiomètre sont soudés côté cuivre pour fournir un accès par le panneau frontal. Rien de bien extraordinaire ou de problématique de ce point de vue puisque, en cas de réparation, le circuit se démonte à l'aide de 4 écrous (là encore, pas de colle, pas de sertissage). Chose amusante, l'ensemble est constitué uniquement de composants traversants. Nous n'avons donc pas besoin de station à air chaud pour éventuellement réparer ou modifier... la station à air chaud.

Le « pistolet » quant à lui se démonte en dévisant un cache placé autour de la partie chauffante en métal et en retirant deux vis à l'arrière. Le démontage révèle une turbine 24V très certainement contrôlée en PWM et un circuit imprimé faisant office de plaque de connexion, sans composant. Je ne peux que vous déconseiller de démonter plus avant cette partie, car le réassemblage n'est pas aisés. L'élément chauffant qui se trouve dans le tube métallique est simplement constitué d'une résistance en spirale enroulée sur un support céramique. Un isolant est ensuite enroulé autour de l'élément pour éviter tout contact avec le tube qui est judicieusement relié à la terre.

3. LE MIEN N'EST PAS FORCÉMENT LE VÔtre

En cherchant davantage d'informations sur le matériel que je venais de recevoir, je suis tombé sur le blog de William Osman (williamosman.com) décrivant, dans un de ses billets (01/2016), sa mésaventure avec, apparemment, le même équipement pourtant largement plébiscité sur de nombreux forums. La première mise sous tension de l'appareil a tout simplement fait sauter son disjoncteur avec, en prime, une gerbe d'étincelles sortant du panneau avant.

Il décrit l'analyse du problème sur son site qui révèle un défaut d'assemblage de son exemplaire : un court-circuit entre la terre et la résistance chauffante dans le pistolet (alors que ce câble est censé être serti sur le tube métallique), doublé d'une inversion des câbles côté alimentation, connectant le fusible sur le neutre et non sur la phase. Il semblerait donc que la personne ayant assemblé son exemplaire du produit se soit trompé dans les connexions, mais aussi et surtout que le contrôle qualité ait laissé passer le matériel alors qu'il présentait clairement un défaut.

Bien sûr, il ne s'agit pas de la même production que celle en ma possession. Ceci remonte à plus de deux ans, avec un produit destiné au marché US (~110V), via un autre canal de distribution (Amazon) et vraisemblablement avec d'autres critères de qualité (les soudures sur les photos du blog sont absolument horribles).

Cet état de fait pourrait potentiellement expliquer la présence du QRcode à gratter pour vérifier l'authenticité du produit si l'un des produits est l'original et les autres des contrefaçons douteuses et dangereuses. Quoi qu'il en soit, le problème reste entier lors de l'achat et je ne peux que vous recommander d'être prudent lors de la mise en œuvre. Vérifiez la présence du QR code et, éventuellement, désassemblez le matériel avant mise sous tension pour, au moins, vous assurer d'une mise à la terre digne de ce nom.

POUR FINIR

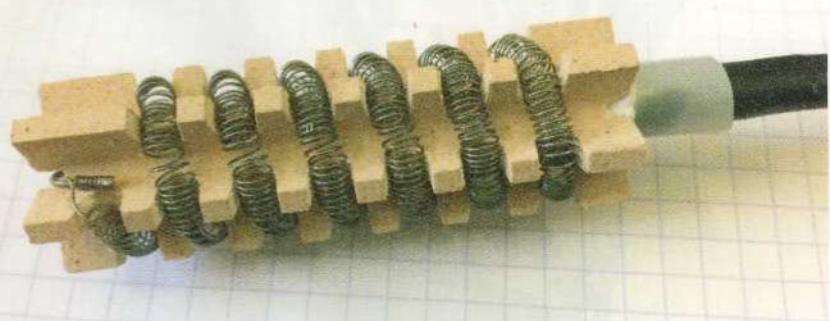
À l'usage, cette station est très efficace et, avec un peu d'entraînement et de finesse, on arrivera très rapidement à des résultats satisfaisants. Les différents essais effectués sur une carte mère de Core 2 Duo défectueuse de longue date, m'ont permis de desolder rapidement aussi bien des composants CMS de très petite taille (condensateurs, résistances, régulateurs de tension), mais également des circuits intégrés avec la même facilité. Même les connecteurs comme ceux d'alimentation ATX ou le bloc PS/2, relativement massifs, n'ont posé aucun problème.

Bien entendu, il existe des limites, principalement dues à une dissipation thermique trop importante par le circuit imprimé lui-même. Je n'ai pas été en mesure, par exemple, de dessouder le socket PGA du processeur, les emplacements PCI ou encore les supports pour la mémoire. En dehors de l'éventuel manque de patience, la cause de ces limitations est bien entendu le caractère multicouche du circuit imprimé. Chose relativement rare sinon inexistante dans le domaine qui nous occupe généralement dans ce magazine.

L'investissement d'une trentaine d'euros est donc parfaitement justifié si vous avez dans l'idée de, par exemple, remplacer la mémoire flash d'un ESP8266 (voir *Hackable n°22*), de changer la couleur d'une led CMS sur un module ou un périphérique, de modifier la tension de sortie d'un adaptateur USB (*Hackable n°19*), de changer un connecteur USB ou encore de remplacer un ULN2806A en paquet SOIC sur la carte d'une chaudière et vous éviter ainsi de dépenser 350€ de pseudo-réparation par un chauffagiste (non, remplacer une carte ce n'est pas une « réparation »).

Oh ! J'oubliai ! Une telle station à air chaud est également très pratique pour la gaine thermorétractable ainsi que pour démonter les périphériques ayant des pièces collées à l'adhésif double face comme les écrans ou les coques de smartphones et d'ordinateurs portables... **DB**

La chaleur est produite par une simple résistance chauffante enroulée autour d'un support en céramique et l'ensemble est glissé, via un isolant, dans le tube métallique où se fixe la buse. C'est simple, mais fonctionnel, et surtout économique.





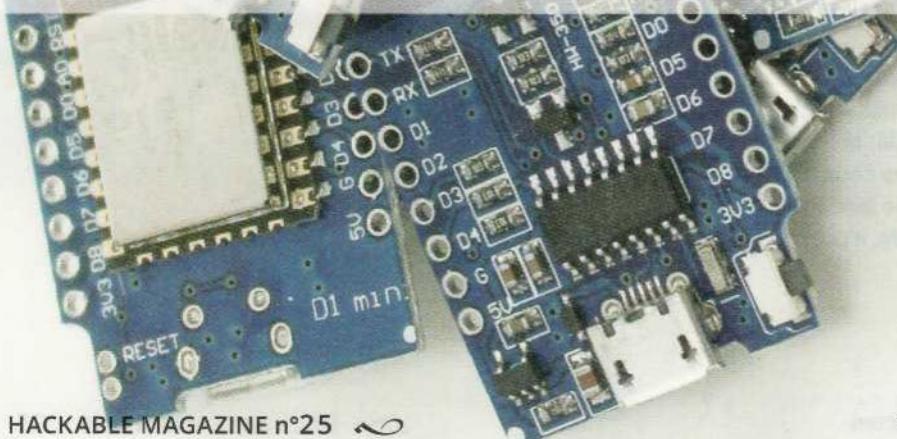
VOS ESP8266 SE METTENT À JOUR TOUT SEULS !

Denis Bodor



Lorsqu'on programme son ESP8266, quel que soit le projet en cours, la première méthode utilisée est celle consistant à le faire en USB.

Lorsque le projet devient plus mature, on basculera alors vers la programmation OTA, directement en Wifi, évitant ainsi de jongler avec des câbles. Dans certains cas cependant, le gain en souplesse ne sera pas suffisant et c'est précisément là qu'on décidera, qu'après tout, l'ESP8266 devrait se mettre à jour tout seul, comme un grand...

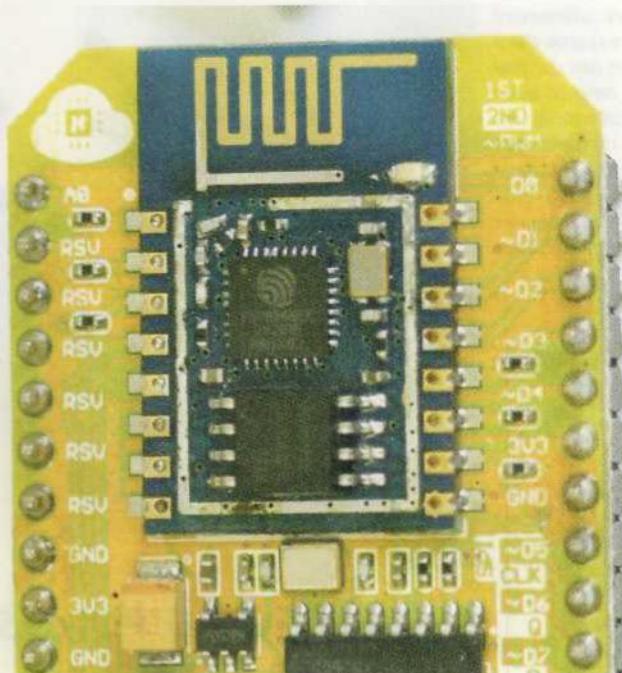


La façon de programmer un module comme un ESP8266 est directement liée, à mon avis, à l'état d'avancement d'un projet. Au début, lorsqu'on met au point le montage et le croquis, une certaine proximité est nécessaire pour rapidement corriger des erreurs ou apporter des améliorations impactant la structure même du projet, aussi bien physiquement (port utilisé, périphérique, bus) et logiquement (code).

Une fois de solides fondations posées, l'objectif de la programmation de la carte ou du module change sensiblement. Il ne s'agit plus de cerner au mieux la façon d'amener le projet dans la bonne direction, mais plutôt d'en affiner les contours et d'apporter des changements plus petits, mais souvent plus nombreux ou fréquents. Là, l'OTA, autrement dit le chargement du croquis directement via la liaison Wifi que l'ESP8266 établi, semble plus intéressante. C'est un sujet que j'avais traité en détail dans le numéro 21 et une technique qui permet, tout simplement, de mettre à jour et ainsi de finaliser le croquis dans un montage déjà en place qui matériellement ne changera plus.

Mais nous pouvons aller plus loin avec une solution qui pourra être bien utile dans certaines situations. Imaginez un instant que vous ayez déployé une armada de quelques 10 ou 12 ESP8266 équipés de capteurs de température, chez vous ou ailleurs. Ne serait-il pas absolument fantastique de simplement mettre à disposition un fichier et de voir tout ce petit monde se mettre à jour automatiquement ? C'est exactement ce que permet de faire le mécanisme de mise à jour via HTTP intégré dans les bibliothèques de base sur support ESP8266 pour Arduino.

Ma logique ici est la suivante : on commence à développer avec un seul ESP8266, on met en place et on bascule sur les mises à jour OTA pour plus de souplesse et enfin, tout en gardant un exemplaire du montage sous la main, on continue de faire évoluer le projet.



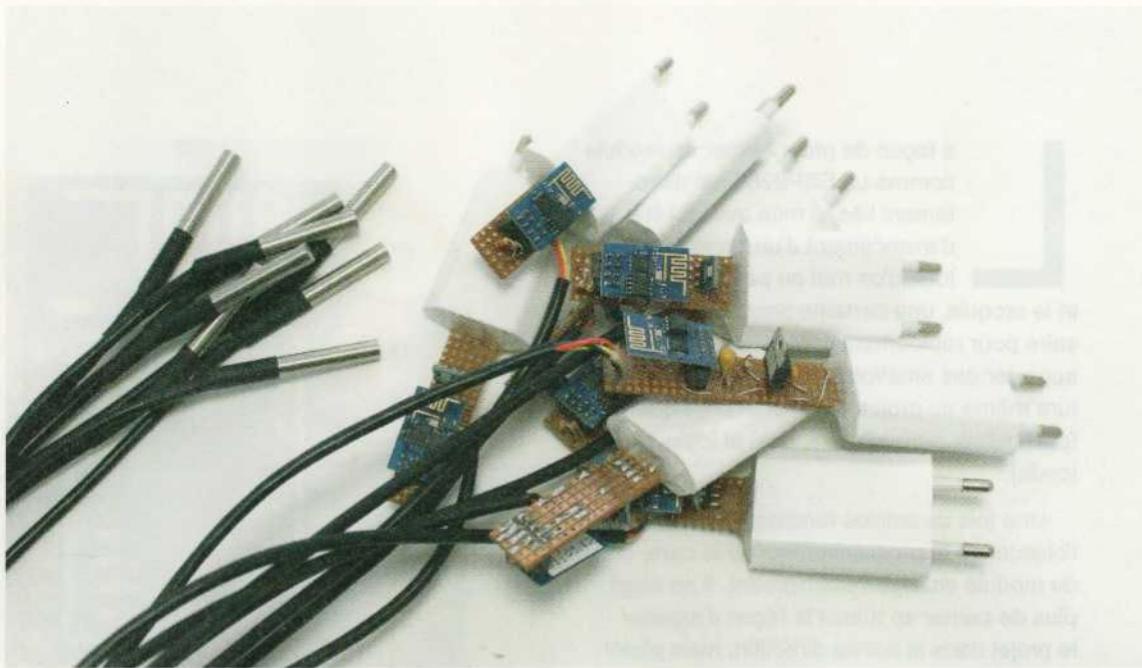
Une fois une nouvelle version du croquis au point et jugée stable et fiable, on place la version compilée du croquis (qu'on appellera désormais un firmware) sur un serveur et l'ensemble des montages passent à cette nouvelle version.

C'est exactement la même technique que celle utilisée par différents objets connectés ou votre smartphone lorsqu'une nouvelle version de leur firmware est disponible. Ici cependant, nous resterons « en local » avec un serveur de mise à jour sur votre réseau domestique mais, si vous disposez d'un serveur web sur le net, ou même d'un hébergement mutualisé, ceci est transposable presque sans le moindre changement dans le code.

Sous le blindage métallique de cette carte NodeMCU se cachent les deux composants les plus importants. En haut, l'ESP8266 lui-même et juste dessous une mémoire flash SPI de 4 Mo. Cette mémoire contient votre croquis, une zone pour les mises à jour OTA, une zone de stockage SPIFFS et une petite partie pour l'émission EEPROM.



La mise à jour via HTTP est particulièrement intéressante dans un cas comme celui-ci, avec une multitude de montages identiques à base d'ESP8266, tous faisant fonctionner le même croquis.



1. UN ENDROIT POUR STOCKER LES MISES À JOUR

Vous l'aurez compris, il nous faut avant tout un endroit où placer le firmware dont le ou les modules ESP8266 auront besoin. Pour ce faire, nous utiliserons naturellement une Raspberry Pi, mais nous ferons l'impasse sur le classique serveur Apache afin d'opter pour quelque chose de plus léger : Lighttpd (prononcer « *lighty* »). Celui-ci possède une empreinte mémoire bien plus réduite que d'autres serveurs HTTP, est facile à configurer, modulaire et parfaitement adapté à une configuration matérielle modeste (une Pi 1B est déjà bien plus qu'il n'en faut).

Nous n'en ferons pas usage ici, mais Lighttpd est parfaitement capable de gérer des pages web dynamiques incluant du code PHP, via son interface FastCGI. D'autres langages peuvent également être utilisés comme Python, Ruby, Perl ou même simplement des scripts shell. Ici cependant, tout ce dont nous aurons besoin se limitera à la fourniture de fichiers statiques.

Installer Lighttpd sur votre Pi se fera très simplement à l'aide de la commande **sudo apt-get install lighttpd**. Une fois l'opération effectuée, le serveur sera immédiatement lancé avec une configuration par défaut et vous pourrez accéder à une page de démonstration depuis un navigateur en utilisant l'adresse IP de la Pi ou son nom d'hôte suivi de « .local ». Ceci est rendu possible grâce à mDNS dont

le support est assuré par Avahi et, comme je l'avais expliqué dans le précédent numéro, vous devrez télécharger et installer les « Services d'impression Bonjour pour Windows » d'Apple pour que votre machine Windows puisse s'en servir. Sur macOS ou GNU/Linux, ceci est pris en charge sans autre forme de procès.

La configuration par défaut de Lighttpd doit être changée pour nous simplifier la vie. En effet, l'emplacement où sont normalement stockés les fichiers du serveur web est **/var/www/html**, mais ce répertoire n'est pas accessible en écriture par défaut à l'utilisateur **pi**. Mieux vaut alors spécifier un autre emplacement, nous évitant de devoir jouer du **sudo** à la moindre manipulation.

Nous éditons donc le fichier **/etc/lighttpd/lighttpd.conf** pour modifier la ligne :

```
server.document-root = "/var/www/html"
```

en

```
server.document-root = "/home/pi/www"
```

Et, par la même occasion, ajouter :

```
server.dir-listing = "enable"
```

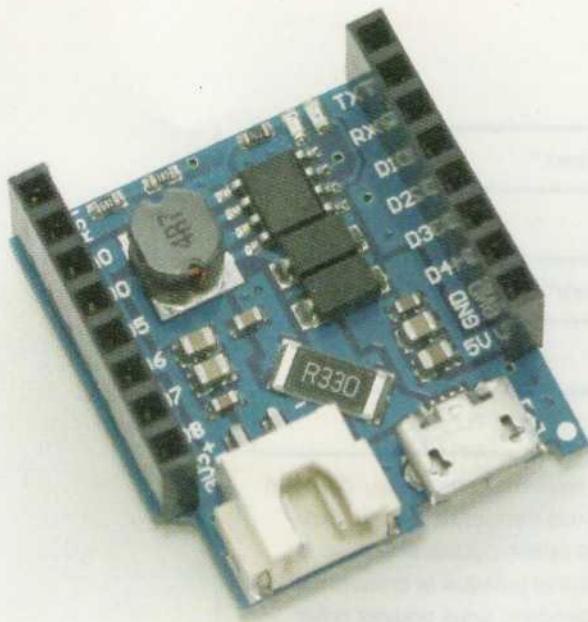
Ceci nous permettra donc de non seulement utiliser un sous-répertoire **www/** de notre répertoire personnel en guise de racine du serveur, mais nous permettra également d'avoir un aperçu de son contenu avec un navigateur. Cette option pourra être mise en commentaire (avec un **#**) une fois le tout pleinement fonctionnel puisque le croquis de l'ESP8266 ne fera pas usage de cette fonctionnalité. En attendant, vous pouvez créer le répertoire **~/www/** et y placer quelques fichiers pour vous assurer que tout fonctionne bien, après avoir redémarré Lighttpd avec **sudo service lighttpd restart**.

2. NOUS AVONS BESOIN D'UNE STRATÉGIE

La procédure de mise à jour telle que mise à notre disposition par les bibliothèques de base ESP8266 pour Arduino repose sur le simple téléchargement d'un fichier binaire. Dans sa version la plus simple, on utilise la méthode **ESPhttpUpdate.update(const String &url)** en précisant, en argument, une chaîne de caractères (objet **String**) étant l'URL où récupérer le fichier.

Grâce au polymorphisme du C++, il est également possible d'appeler cette méthode avec d'autres arguments et son comportement va alors changer. Ainsi, en appelant par exemple **ESPhttpUpdate.update("http://marasperrypi.local", "V1.0.1")**, cette seconde chaîne de caractères sera envoyée dans l'entête de la requête HTTP en compagnie d'autres informations comme l'espace disponible en flash sur l'ESP8266 ou la version du SDK (support Arduino) utilisée, sous la forme **x-ESP8266-version: V1.0.1**. Le serveur pourra alors déterminer si la version actuellement exécutée du croquis est inférieure ou non à celle qu'il met à disposition et répondre en envoyant le nouveau croquis binaire (réponse HTTP 200), ou en signifiant que rien n'a changé et que la mise à jour n'est pas nécessaire (réponse HTTP 304).

Nous avons donc deux approches possibles et l'une n'est pas forcément plus simple que l'autre. Soit c'est l'ESP8266 qui détermine seul s'il a besoin d'une mise à jour sur la base d'informations fournies par le serveur web, soit c'est le serveur web qui décide s'il faut ou non envoyer telle ou telle réponse à un ESP8266 qui fait la demande. Dans le premier cas, nous devons prévoir cela dans notre croquis et dans le second, ceci nous oblige à utiliser un langage (PHP en toute logique) pour écrire une page web dynamique. En d'autres termes, nous aurions à gérer deux codes, un en C++/Arduino et un en PHP, et ceci ne me plaît absolument pas d'un point de vue de maintenance du code et puis... c'est du PHP (beurk!).



De plus en plus facile à trouver et de moins en moins cher, ce genre de « shield » d'alimentation permet l'utilisation d'un accumulateur Li-ion ou LiPo pouvant rendre vos projets ESP8266 autonomes.

Nous partirons donc ici sur la première solution et le serveur se contentera de fournir des fichiers. Nous devons alors prévoir un mécanisme qui permettra à l'ESP8266 de comparer sa propre version du croquis à celle disponible. Pour ce faire, nous nous baserons sur la présence d'un fichier texte, judicieusement nommé **version**, sur le serveur et contenant une seule ligne de caractères numériques. Ce contenu décrira ainsi la version la plus récente du firmware proposé et le croquis, après une conversion de cette chaîne en entier, pourra comparer cela à la valeur d'une macro **VERSION** intégrée dans son code. Si le contenu du fichier **version** est numériquement supérieur à **VERSION**, la mise à jour sera déclenchée en utilisant un fichier **firmware** ayant pour extension le numéro de version en question.

Il existe plusieurs approches parallèles à partir à celle-ci et il faudra prendre en compte un autre élément important selon la manière dont vous aurez structuré votre croquis initial. En effet, il est généralement d'usage d'intégrer directement dans le code des éléments de configuration qui devraient, plus judicieusement, trouver leur place ailleurs.

Le SSID et la phrase de passe du point d'accès auquel se connecter, par exemple, sont généralement spécifiés directement dans le croquis. On peut cependant supposer que tous les ESP8266 d'une installation se connecteront au même point d'accès et donc tolérer cette approche dans ce cas. Il n'en va cependant pas de même en cas d'utilisation de mDNS, en particulier si l'on décide de nommer arbitrairement chaque ESP8266. Ceci est également valable lorsque l'on décide de conserver le mécanisme de mise à jour OTA (cf. *Hackable* n°21), en plus de la mise à jour HTTP, en cas de problème. La chaîne de caractères utilisée avec `ArduinoOTA.setHostname()`, comme avec `MDNS.begin()`, est propre à chaque module/carte qui devraient donc tous avoir un firmware différent.

L'une des solutions possibles pour contourner ce problème consiste à enregistrer ce genre d'informations de configuration, non pas dans le code du croquis lui-même, mais dans l'EEPROM intégrée à la plateforme. Ceci est directement hérité du fonctionnement des cartes Arduino basées sur un microcontrôleur Atmel AVR intégrant une zone mémoire particulière qui n'est pas effacée lorsque le croquis est reprogrammé. Dans le cas de l'ESP8266, cette EEPROM interne n'existe pas réellement et n'est qu'un segment de la mémoire flash. Elle s'utilise cependant avec la même logique consistant à déclarer une variable, l'initialiser, puis l'enregistrer une première fois avec `EEPROM.put()` pour ensuite, en récupérer le contenu à chaque démarrage avec `EEPROM.get()`.

Il y a toutefois quelques petites différences avec l'utilisation sur carte Arduino :

- Il faut utiliser `EEPROM.begin()` en spécifiant une taille entre 4 et 4096 octets, avant de pouvoir procéder aux opérations d'écriture ou de lecture dans l'EEPROM émulée. Ceci permet de mettre en place le mécanisme d'émulation tout en pouvant ajuster la taille à ses besoins, sans

consommer inutilement des ressources (une copie en RAM de l'EEPROM émulée est maintenue en permanence).

- Une écriture, avec **EEPROM.put()** ou **EEPROM.write()** doit impérativement être suivie de **EEPROM.commit()** pour réaliser effectivement l'enregistrement. **EEPROM.end()**, qui stoppe l'émulation et libère les ressources, procède automatiquement à un **EEPROM.commit()**.

Une autre approche, assez similaire dans l'esprit à l'utilisation de l'EEPROM, est la mise en œuvre du SPIFFS, le *SPI Flash File System* ou, en bon français, du système de fichiers en mémoire flash SPI. La plupart des modules ESP8266 actuels disposent de 4 Mo de mémoire flash, mais un croquis ne peut pas occuper plus de 1 Mo. Les 3 Mo restants sont utilisés pour les mises à jour OTA, l'EEPROM émulée et la configuration Wifi (SDK). Une grosse part de la mémoire flash libre peut être utilisée comme support de stockage, d'une façon assez similaire à une carte SD, avec arborescence, fichiers, etc. Nous en parlerons une autre fois plus en détail, car pour stocker une simple chaîne de 32 caractères (la taille maximum du nom d'hôte avec un ESP8266), ceci serait un peu démesuré.

Une solution totalement différente pour gérer le caractère unique de chaque ESP8266 installé, pourrait consister à proposer un firmware par carte/module sur le serveur web. On pourrait inclure

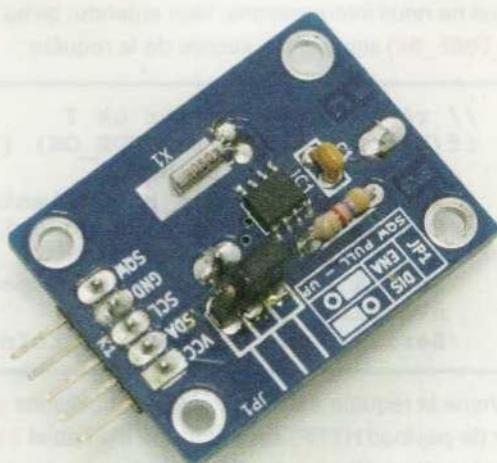
l'adresse MAC ou le nom d'hôte dans le nom du fichier afin que chaque ESP8266 sache quel fichier tenter de récupérer pour la mise à jour. Personnellement, cette approche ne me convient pas, car le code du croquis ne serait alors plus totalement générique et commun à tous les ESP8266 d'un projet donné, et surtout, en ayant 12 capteurs en service par exemple, ceci suppose 12 fichiers et donc 12 modifications, 12 compilations, 12 exportations, 12 copies sur le serveur... et 12 occasions de faire une erreur idiote de manipulation.

3. LE CROQUIS

À présent que la stratégie est établie et que le serveur est prêt, il ne nous reste plus qu'à nous tourner vers les éléments à intégrer au croquis. Bien entendu, il ne s'agira ici que d'un exemple, puisque ceci est avant tout destiné à être intégré dans le code d'un croquis existant. Comme pour de nombreuses autres fonctionnalités de ce type, nous choisirons naturellement de tout centraliser dans une unique fonction.

Les fonctionnalités qui nous intéressent sont accessibles en intégrant le fichier suivant :

```
#include <ESP8266httpUpdate.h>
```



Pour disposer d'une base de temps fiable, la solution matérielle la plus courante consiste à utiliser un module RTC comme celui-ci. Cependant, comme l'ESP8266 possède une connectivité Wifi et que nous installons un serveur web (sur Pi), nous pouvons nous en passer et tout simplement utiliser un serveur NTP local.



Pour simplifier les choses, nous définissons la macro **VERSION** destinée à identifier la version courante du croquis et déclarons/initialisons la variable **urlbase** formant la base de toutes les requêtes que nous ferons vers le serveur web :

```
#define VERSION 0002
const char* urlbase = "http://raspberrypiled.local/hygrotemp/";
```

Ceci nous permettra de facilement adapter le croquis pour un autre projet en fournissant une URL différente sans avoir à toucher au code de la fonction. Notez que j'inclus dans l'URL de base un répertoire, **hygrotemp/**, en prévision de l'utilisation de ce même code pour d'autres projets. Tout ce que j'aurais à faire sur la Pi sera d'ajouter un répertoire ayant son propre fichier **version** et les firmwares adaptés. Je peux donc me servir du serveur web pour plein de projets différents sans qu'il n'y ait conflit.

Sans tarder, entrons dans le vif du sujet avec le début de la fonction elle-même :

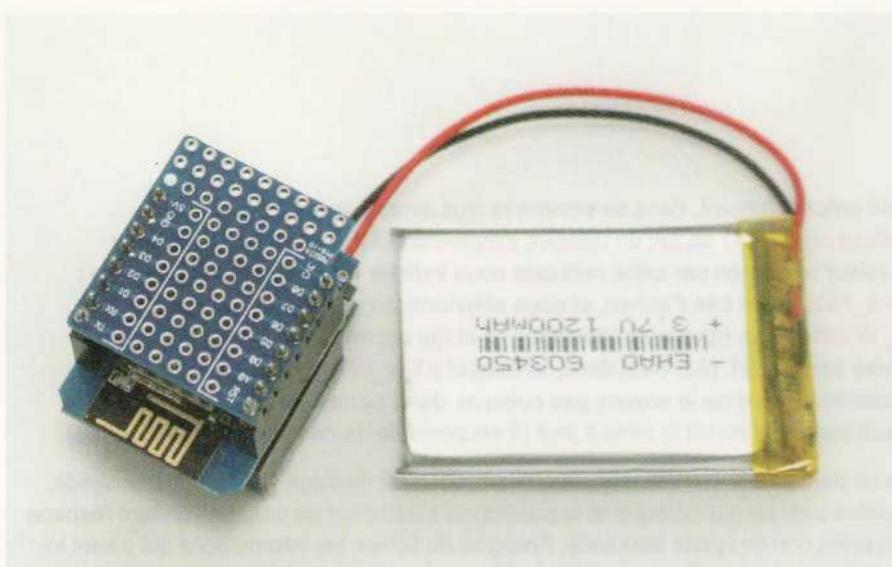
```
void maj() {
    // Accès au fichier "version"
    String urlversion = String(urlbase);
    urlversion.concat("version");
    Serial.println("Vérification version... ");
    HTTPClient httpClient;
    httpClient.begin(urlversion);
    int httpCode = httpClient.GET();
```

Nous construisons un objet de type **String** à partir de la chaîne de caractères formant l'URL de base puis ajoutons, avec la méthode **concat()**, le nom du fichier qui nous intéresse. Dès lors, **urlversion** contiendra l'URL complète vers le fichier et nous l'utilisons pour déclencher la requête après avoir créé l'objet **httpClient** pour accéder au serveur et configurer la future connexion avec sa méthode **begin()**. La requête elle-même est déclenchée par la méthode **GET()** qui nous retourne alors, dans **httpCode**, le code de réponse standardisé (RFC 2616) qui nous permet de connaître le résultat de l'opération.

Nous ne nous intéresserons, bien entendu, qu'au code 200 (correspondant à la macro **HTTP_CODE_OK**) signifiant le succès de la requête :

```
// réponse du serveur ok ?
if(httpCode == HTTP_CODE_OK) {
    // lecture du fichier
    String nVersion = httpClient.getString();
    Serial.print("Version actuelle: ");
    Serial.println(VERSION);
    Serial.print("Version disponible: " );
    nVersion.trim();
    Serial.println(nVersion.toInt());
```

Comme la requête est un succès, nous utilisons **getString()** pour obtenir le contenu du fichier (le *payload* HTTP) sous la forme d'un objet **String**. Il faut normalement faire très attention avec cette méthode, car l'intégralité du contenu sera chargée en RAM et s'il n'y en a pas assez, une erreur (exception) sera provoquée et l'ESP8266 redémarrera. Dans notre cas, nous



La carte WeMos D1 mini R2 (ou ses clones) est sans le moindre doute ma plateforme ESP8266 préférée. De taille réduite et excessivement économique, elle dispose, de plus, d'une multitude de shields tout aussi financièrement abordables.

contrôlons parfaitement la situation puisque nous avons la main sur le serveur et son contenu, mais gardez cet avertissement en tête.

Afin de comparer la version récupérée dans le fichier et celle décrite dans le présent croquis sous la forme de la macro **VERSION**, nous supprimons les espaces vides (blancs, espaces, retours à la ligne) avec **trim()** avant de convertir la chaîne en entier avec **toInt()** pour l'affichage. Ceci peut sembler inutile puisqu'une chaîne s'affichera exactement comme une valeur entière, mais ceci nous permettra de rapidement détecter un éventuel problème. Si le fichier **version** contient autre chose qu'uniquement des chiffres, cette conversion retournera **0**, et nous le verrons immédiatement.

Nous pouvons ensuite tester cette version :

```
// La version dispo est supérieure à la mienne ?
if(nVersion.toInt() > VERSION) {
    Serial.print("Mise à jour disponible: ");
    // composition de l'URL
    String urlfirmware = String(urlbase);
    urlfirmware.concat("firmware.");
    urlfirmware.concat(nVersion);
    Serial.println(urlfirmware);
    Serial.println("Mise à jour...");
```

Un simple **if** nous permet de prendre la décision de déclencher une mise à jour et si tel est le cas, nous composons une nouvelle URL en concaténant l'URL de base, la chaîne « **firmware.** » et la chaîne issue du fichier **version**. Puis nous déclenchons la mise à jour :

```
// mise à jour
t httpUpdate_return ret = ESPhttpUpdate.update(urlfirmware);
if(ret == HTTP_UPDATE_FAILED) {
    // on n'arrive pas ici si ça marche car l'ESP reboot
    Serial.print("Erreur mise à jour ");
    Serial.print(ESPhttpUpdate.getLastErrorCode());
    Serial.print(": ");
    Serial.println(ESPhttpUpdate.getLastErrorMessage().c_str());}
```



Comme dit précédemment, dans sa version la plus simple, la mise à jour avec `ESPhttpUpdate.update()` se fait en utilisant simplement l'URL du fichier binaire du croquis. La valeur renournée par cette méthode nous indique le résultat de la demande : `HTTP_UPDATE_FAILED` en cas d'échec, et nous affichons la raison de l'erreur, `HTTP_UPDATE_NO_UPDATES` que nous ne traitons pas ici et qui est en rapport avec une gestion de version côté serveur (cf. plus haut dans l'article) et `HTTP_UPDATE_OK` qui, si tout se passe correctement, nous ne le voyons pas puisque, dans ce cas, un redémarrage de l'ESP8266 suit immédiatement la mise à jour (il est possible de désactiver cette fonction).

Si vous vous demandez, comme moi, ce qu'il se passe s'il ne s'agit pas d'un fichier valide, ne vous inquiétez pas. La bibliothèque et la plateforme s'occupent de tout, en vérifiant l'espace disponible, la présence du fichier demandé, l'intégrité du fichier, les informations qui y sont intégrées, etc. Vous ne risquez donc pas d'effacer le contenu de la flash de vos ESP8266 avec un JPEG de votre fantastique (ou terrifiant) essai culinaire de macarons praliné-anchois...

Le reste de la fonction, se limitera à traiter les problèmes et erreurs :

```
    } else {
        // VERSION >= au contenu de "version"
        Serial.println("Déjà à jour");
    }
} else {
    // Mais où est "version" ??
    Serial.println("Erreur récupération de version !!!");
}
}
```

Comme on souhaitera conserver une possibilité de mise à jour OTA en plus de ce nouveau système, et puisque l'ESP8266 de développement permettant de tester une nouvelle version du croquis ne fait pas partie de la « flotte » d'ESP8266 installés « en production », nous intégrons une fonction à cet effet :

```
char myhostname[32];
const char* otapass = "123456";

void confOTA() {
    // Port 8266 (défaut)
    ArduinoOTA.setPort(8266);

    // Hostname défaut : esp8266-[ChipID]
    EEPROM.get(0, myhostname);
    ArduinoOTA.setHostname(myhostname);

    // mot de passe pour OTA
    ArduinoOTA.setPassword(otapass);

    ArduinoOTA.onStart([]() {
        Serial.println("/!\\ Maj OTA");
    });
    ArduinoOTA.onEnd([]() {
```

```
    Serial.println("\n/!\\ MaJ terminee");
});
ArduinoOTA.onProgress([](unsigned int progress, unsigned int total) {
    Serial.printf("Progression: %u%\r", (progress / (total / 100)));
});
ArduinoOTA.onError([](ota_error_t error) {
    Serial.printf("Error[%u]: ", error);
    if (error == OTA_AUTH_ERROR) Serial.println("Auth Failed");
    else if (error == OTA_BEGIN_ERROR) Serial.println("Begin Failed");
    else if (error == OTA_CONNECT_ERROR) Serial.println("Connect Failed");
    else if (error == OTA_RECEIVE_ERROR) Serial.println("Receive Failed");
    else if (error == OTA_END_ERROR) Serial.println("End Failed");
});
ArduinoOTA.begin();
}
```

Rien de bien particulier ici, si ce n'est l'utilisation de **E EEPROM.get()** afin d'initialiser **myhostname** pour ensuite donner un nom d'hôte à la plateforme, plutôt que d'utiliser simplement quelque chose comme **ArduinoOTA.setHostname("nomdemachine")**. La fonction **setup()** devra impérativement utiliser **E EEPROM.begin()** au plus tôt et avant d'invoquer **confOTA()** :

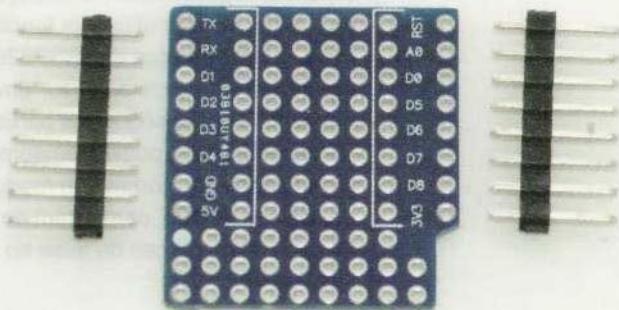
```
// activation émulation EEPROM 48 octets
EEPROM.begin(48);
{
    // Juste la première fois
    strcpy(myhostname, "espOTAdatafs");
    EEPROM.put(0, myhostname);
    EEPROM.commit();

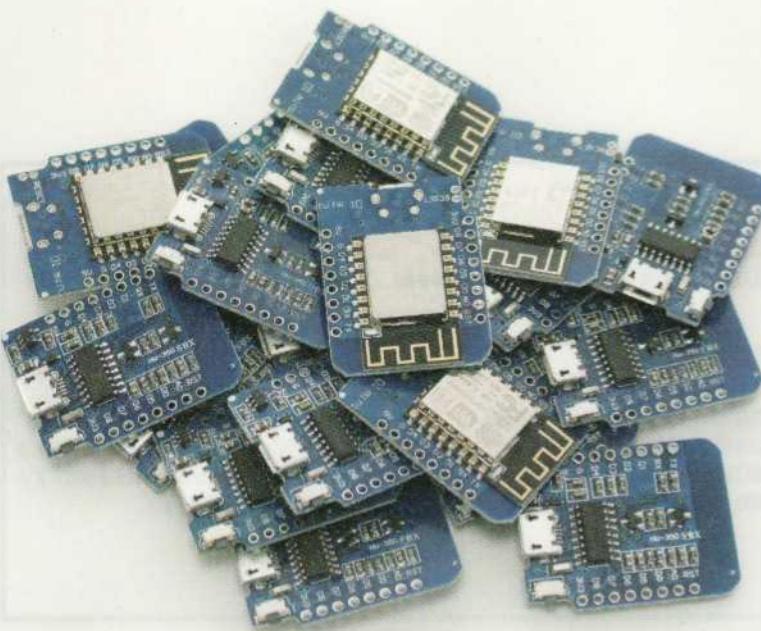
    confOTA();
}
```

Sans oublier bien sûr, l'appel récurrent à **ArduinoOTA.handle()** dans **loop()** par la suite. Notez que les trois lignes centrales ne sont là que pour une première écriture en EEPROM et devront être commentées par la suite, une fois le nom d'hôte de chaque ESP8266 inscrit en EEPROM. Pour déclencher une vérification et éventuellement une mise à jour du firmware en HTTP, il suffira d'appeler **maj()**.

Pour mettre un nouveau firmware à disposition, tout ce que vous avez à faire est de procéder à vos modifications, incrémenter **VERSION**, compiler et utiliser le menu *Croquis* et *Exporter les binaires compilées* (pourquoi le féminin ? C'est UN binaire non ?). Un fichier **.bin** dont le nom est composé du nom du croquis et de la plateforme/carte sera alors créé dans le répertoire du croquis, au côté du **.ino**. Exemple : **espFSwebNTP.ino.d1 mini.bin**. Transférez ce

Voici une « protoboard », ou shield de prototypage, spécialement destinée au WeMos D2 mini. À 1€/pièce, c'est une solution idéale pour rapidement mettre en œuvre un circuit simple reposant sur un unique capteur (température, lumière, hygrométrie, etc.).





On trouve facilement des clones de WeMos D1 mini sur les sites de vente en ligne, généralement pour un unique euro pièce. À ce prix-là, il n'y a pas de raison de lésiner sur la quantité commandée, et si d'aventure l'un des modules du lot ne fonctionne pas, on pourra toujours s'en consoler en vampirisant tous les autres composants pour d'autres projets.

fichier sur le serveur web en le renommant **firmware**, suivi du numéro de version et placez cette même valeur dans le fichier **version**. Le serveur est prêt pour la mise à jour des ESP8266.

La question est donc maintenant : quand déclencher cette vérification ? Ceci est avant tout une affaire de gestion. Vous pouvez, par exemple, choisir de faire cela lors du démarrage et donc tout simplement en fin de **setup()**. Mettre à jour vos ESP8266 dédiés à un projet reviendra simplement à les redémarrer un par un. Vous pouvez également choisir d'ajouter un bouton dont le changement d'état provoquera la mise à jour, ou encore mettre en place un serveur simpliste sur chaque ESP8266 pour déclencher la procédure à distance via le réseau.

Enfin, une autre solution consiste, tout simplement, à vérifier régulièrement la disponibilité d'une mise à jour. Vous pouvez utiliser la fonction **millis()** pour ce faire, qui dans le cas de l'ESP8266 retourne un entier sur 32 bits et boucle donc sur environ 50 jours (même si cela est sans conséquence particulière pour ce genre de choses). Vous pouvez également utiliser une horloge temps-réel, comme un module DS3231, DS1338 ou même DS1307, permettant de conserver la date et l'heure même en cas de mise en sommeil ou d'arrêt de l'ESP8266.

Et enfin, vous avez la possibilité de demander l'heure au serveur sur la Pi...

4. BONUS NTP

Comme vous le savez sans doute, une carte Raspberry Pi ne dispose pas d'une horloge interne pouvant être alimentée sur batterie comme c'est le cas pour une carte mère de PC par exemple. De ce fait, par défaut, lorsque la carte est mise hors tension ou redémarrée, elle perd la date et l'heure courante. Pour pallier à ce problème, un mécanisme intégré par défaut permet au système Raspbian de récupérer l'heure exacte au démarrage et ainsi garder une acuité temporelle nécessaire au système GNU/Linux.

Une alternative possible à ce fonctionnement consiste à ajouter une horloge temps réelle sous la forme d'un module DS3231 équipé d'une pile bouton. Ainsi lorsque la carte n'est pas alimentée, l'horloge fonctionne toujours et dès le prochain démarrage le système de la Pi la récupérera pour configurer la base de temps locale (il faut également ajouter une ligne **dtoverlays=i2c-rtc,ds3231** dans votre **/boot/config.txt**).

Mais saviez-vous que le mécanisme utilisé pour récupérer l'heure depuis une connexion Internet, via un protocole nommé NTP (*Network Time Protocol*), peut également distribuer cette information sur le réseau local ? Ce n'est pas tout, la configuration par défaut de Raspbian dispose déjà de tout ce qu'il faut pour faire cela, puisque c'est exactement le même paquet et le même service qui sert à la fois pour récupérer l'heure depuis le net et la fournir sur le LAN.

Tout ce que vous avez à faire est d'éditer le fichier `/etc/ntp.conf` et ajouter une simple ligne avant de redémarrer le service avec `sudo service ntp restart`:

broadcast 192.168.10.255

L'adresse spécifiée après `broadcast` correspond à l'adresse de diffusion de votre réseau local. Ici le réseau est 192.168.10.0/24 et l'adresse correspond donc à l'ensemble du réseau formé par les adresses IP de 192.168.10.1 à 192.168.10.254. Dès cette configuration modifiée, toutes les machines du réseau pourront faire une requête NTP vers cette Raspberry Pi et lui demander l'heure. Et donc nos ESP8266 également !

Pour que nos ESP8266 obtiennent l'heure de cette manière, il suffit d'installer la bibliothèque `NTPClient` de Fabrice Weinberg, directement depuis le gestionnaire de bibliothèques, puis d'ajouter tout d'abord ces lignes dans votre croquis :

```
#include <WiFiUdp.h>
#include <NTPClient.h>

WiFiUDP ntpUDP;
// serveur, offset (s), update (ms)
NTPClient timeClient(ntpUDP,
"raspberrypiled.local", 0, 60000);
```

On pourra alors insérer un `timeClient.begin()` dans `setup()` pour activer la fonctionnalité (démarrer le client UDP correspondant pour recevoir la réponse du serveur) puis `timeClient.update()` dans `loop()`. Une requête NTP sera faite vers le serveur uniquement suivant l'intervalle spécifié dans le constructeur (la ligne déclarant l'objet de type `NTPClient`). Le principe est le même que pour `ArduinoOTA.handle()`, on appelle la fonction et elle ne provoque quelque chose que s'il y a vraiment quelque chose à faire.

Différentes méthodes peuvent alors être utilisées sur l'objet `timeClient`:

- `getFormattedTime()` retourne une chaîne de caractères formatée (pour utiliser avec `Serial.println()`),
- `getDay()` retourne le jour de la semaine avec dimanche = 0,
- `getHours()` retourne l'heure,
- `getMinutes()` les minutes,
- `getSeconds()` les secondes,

- `getEpochTime()` retourne la date au format epoch UNIX, soit le nombre de secondes écoulées depuis le 1er janvier 1970 à minuit.

Ces valeurs peuvent donc être comparées à une valeur arbitraire permettant ainsi de vérifier la présence d'une mise à jour avec l'intervalle de votre choix : toutes les minutes, toutes les heures, etc., ou encore tous les jours à une heure précise (en pleine nuit par exemple).

CONCLUSION

Certes, ce type de mécanisme de mise à jour est particulier et principalement prévu pour un ensemble d'ESP8266 déjà installé. Pour un module seul, la mise à jour OTA sera sans doute une solution plus simple. Il est cependant possible de mettre à l'échelle très facilement cette solution puisque le serveur HTTP installé sur la Pi acceptera sans broncher la charge d'une vaste flottille de capteurs souhaitant une mise à jour.

Il est également possible d'ajouter un certain nombre de fonctionnalités relevant d'aspects plus sécuritaires comme l'utilisation de HTTPS et la vérification d'une empreinte HTTPS. Ou encore d'ajouter quelques vérifications sur le contenu du fichier `version` (somme de contrôle, signature, etc.). Sans oublier que vous avez tout loisir d'explorer la solution impliquant un code PHP côté serveur pour mettre en œuvre l'autre méthode de gestion de version.

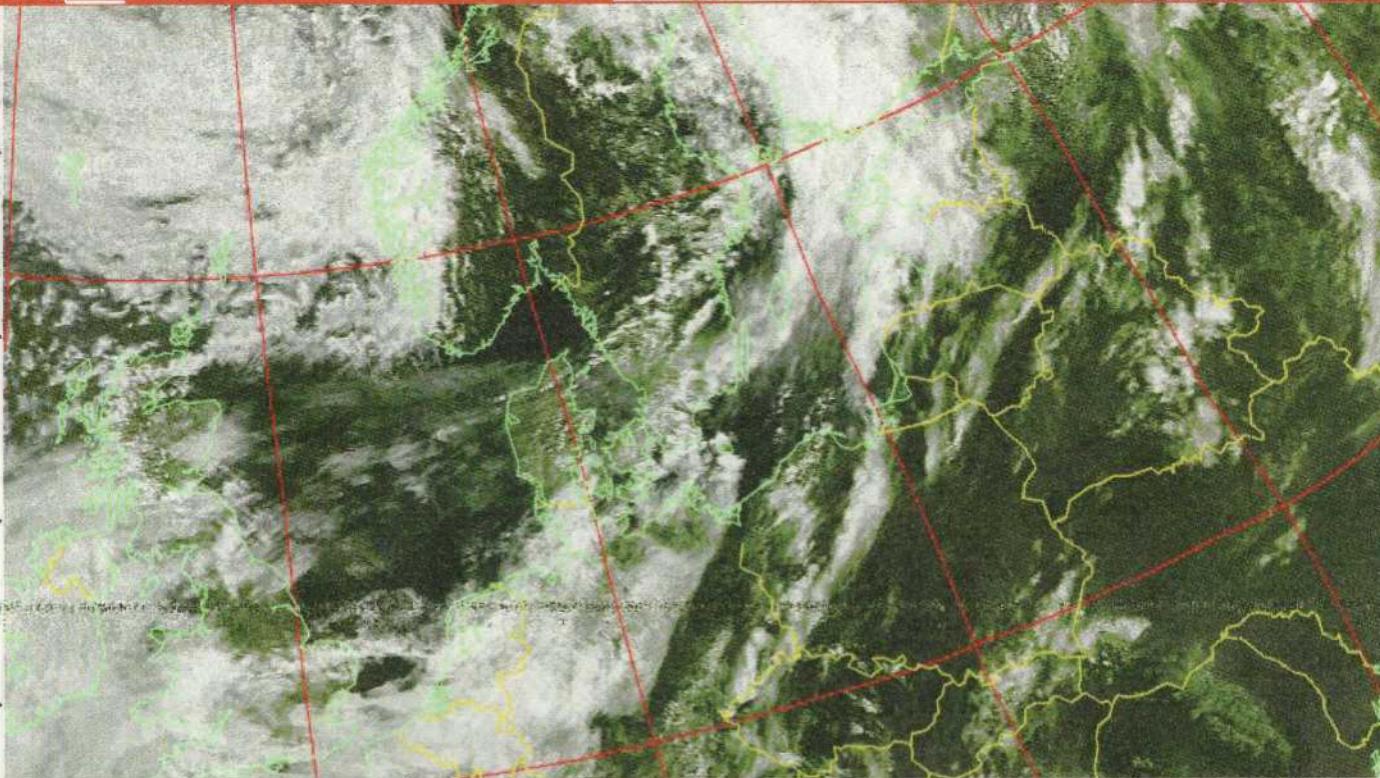
Comme toujours, le croquis complet traité ici est disponible dans le dépôt GitHub du magazine et contient tous les aspects que j'ai délibérément écartés ici pour ne pas surcharger le code (connexion au Wifi, messages sur le moniteur série, etc.).

DB



EN COUVERTURE

IMAGES SATELLITE



INTRODUCTION À LA RÉCEPTION D'IMAGES SATELLITE

Denis Bodor

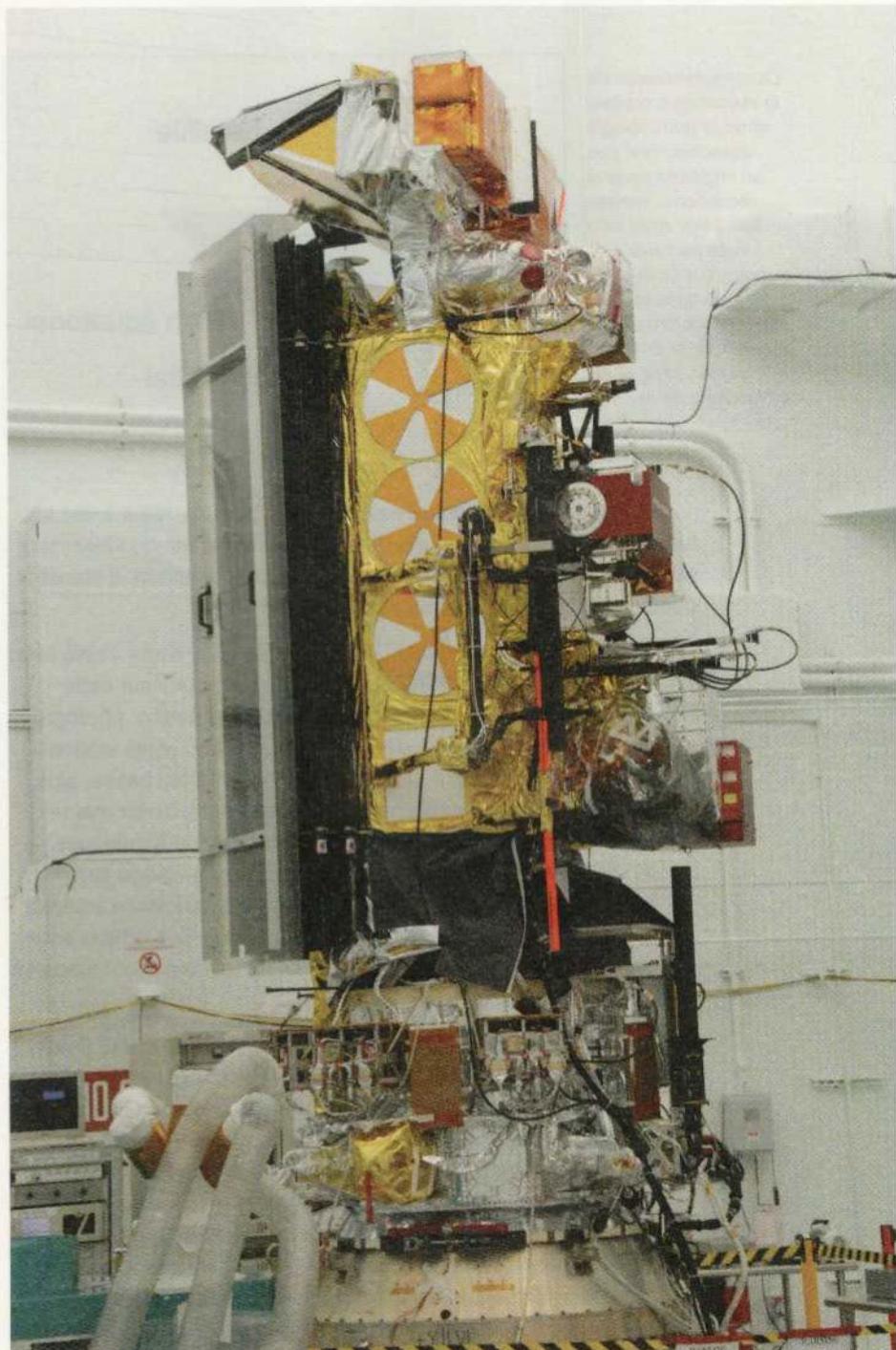


Nous ne le percevons pas nécessairement par nous-mêmes, en tant qu'humains, mais nous baignons littéralement dans un brouillard d'ondes, d'émissions électromagnétiques et de signaux radio au quotidien. TNT, radio FM, Wifi, Bluetooth, LoRa, 3G, télécommandes diverses... sont autant de sources et de types d'émissions parcourant le globe et régissant nos vies. Mais il en est qui sont bien plus captivantes que les autres : celles provenant de l'espace !

Lorsqu'on y pense sans trop entrer dans le détail, recevoir en direct des émissions provenant de l'espace peut paraître étonnant, mais en réalité, si on s'en tient à une définition aussi sommaire et large, il suffit d'allumer son smartphone et lancer une application de géolocalisation quelconque pour faire précisément cela. Le récepteur GPS embarqué dans votre téléphone mobile capte les émissions sur 1575 MHz de quelques-uns des 31 satellites en orbite à quelques 20000 km au-dessus de nos têtes.

On peut, bien entendu, se rapprocher de l'aspect plus technique en faisant fi du positionnement et en lisant les messages décodés par le récepteur au format NMEA. Ceci peut être fait avec votre smartphone, bien entendu, mais également avec des cartes comme Arduino et Raspberry Pi connectées à un module de réception GPS, généralement interfacé via une liaison série. On peut alors presque lire le « langage » des satellites et se rapprocher de l'émission originale. Enfin, grâce à la popularisation des récepteurs RTL-SDR, il devient également possible de capter directement les signaux du ou des satellites, procéder à la démodulation et au décodage soi-même pour obtenir un résultat similaire.

L'intérêt cependant, n'est pas de se compliquer la tâche et de refaire, de manière plus éprouvante, quelque chose qu'il est déjà possible de faire très facilement avec un module à quelques euros, une carte Arduino et un croquis d'une

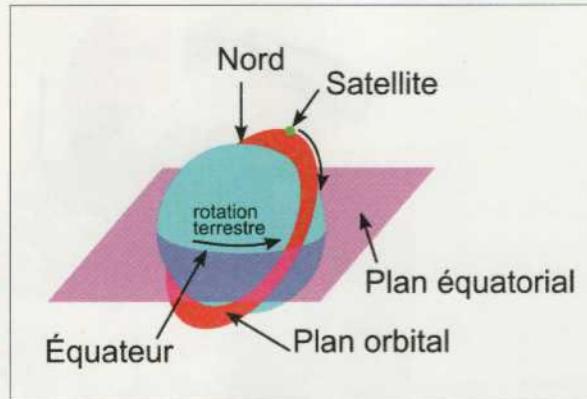


Voici le satellite NOAA-19 avant son lancement en 2009. Celui-ci fait partie des 3 satellites NOAA POES encore en service, mais qui devraient arriver en fin de vie en courant d'année.

quinzaine de lignes. Il y a bien d'autres satellites, là haut, gravitant autour de la Terre, nous envoyant des messages de toutes sortes. En réalité, il y en a même énormément puisque les dernières estimations montrent qu'il y aurait quelques 1400 satellites opérationnels en orbite, et ce n'est là qu'un chiffre officiel ne comptabilisant bien entendu pas les satellites « secrets » destinés aux applications militaires.



La compréhension de la mécanique orbitale et de la terminologie associée n'est pas un impératif dans la réception d'images satellites, mais ceci vous permettra de bien plus facilement comprendre les informations généralement données par les sites et les applications de suivi de satellites.



Tout ce petit monde tourne autour de la planète, mais à des altitudes et avec des trajectoires différentes, impactant physiquement notre capacité de réception des signaux qu'ils émettent. Il convient donc de différencier plusieurs catégories d'orbites :

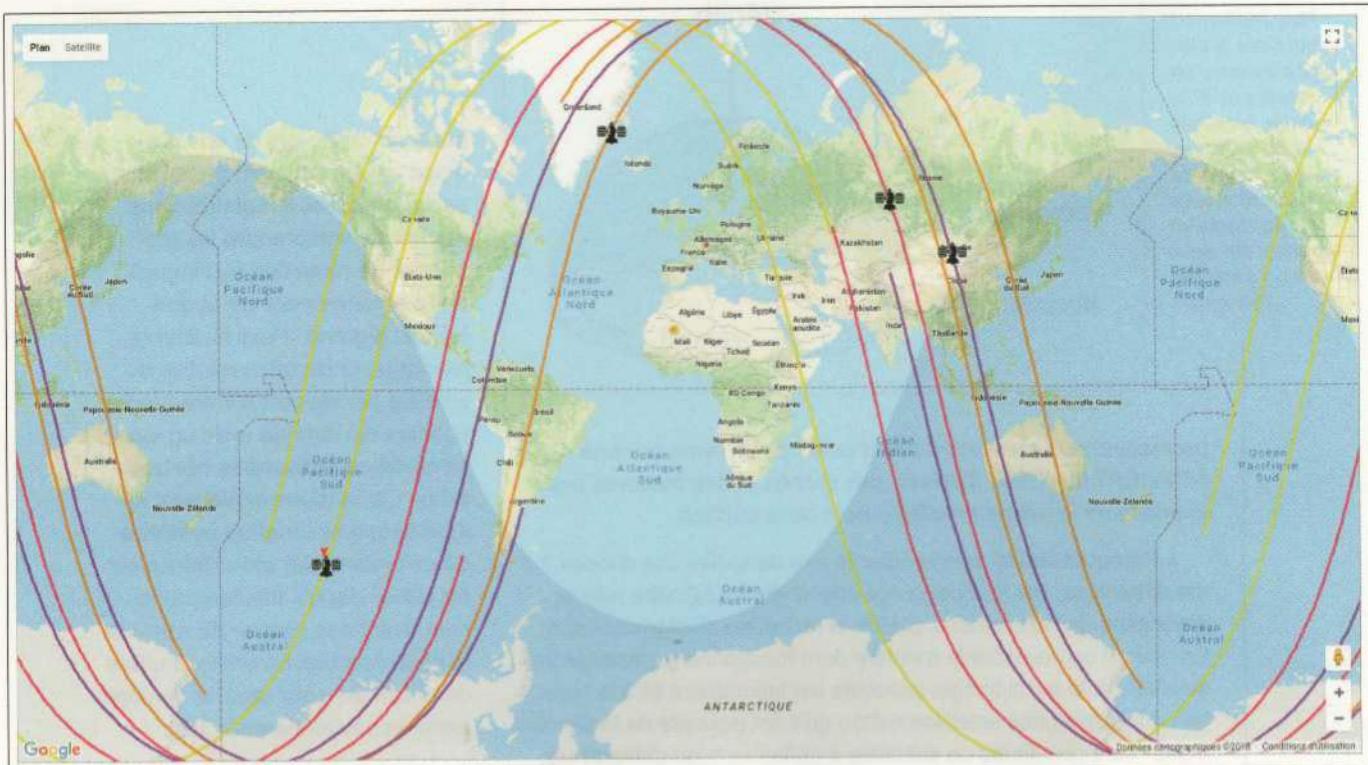
- Les **orbites basses**, ou LEO en anglais (*Low Earth Orbit*) avec une altitude jusqu'à 1500 ou 2000 km. On trouve sur cette orbite des satellites de télédétection (climat, météo, photographie, espionnage, etc.), de télécommunication et les stations spatiales (dont l'ISS à 408 km). Plus l'altitude est basse, plus un satellite doit se déplacer rapidement pour pouvoir maintenir son orbite. L'ISS, par exemple, se déplace à quelques 27500 km/h, ce qui signifie que le temps de passage sera relativement court, mais que le signal sera plus facile à capter en raison de la relative proximité avec le sol. Ces orbites sont généralement également celles qui sont le moins coûteuses en termes de déploiement.
- Les **orbites moyennes** ou MEO (*Medium Earth Orbit*) entre 1500 et 35000 km d'altitude, sont généralement le terrain de jeu des satellites de navigation comme le GPS ou Glonass, respectivement à 19100 km et 20200 km d'altitude. Ces satellites se trouvant plus haut, leur vitesse est plus réduite et leur période orbitale (le temps nécessaire à une révolution) se situe entre 2 et 12 heures. C'est ce qui explique qu'en utilisant une application de réception GPS sur smartphone, vous constaterez que les satellites en question restent relativement longtemps au-dessus de votre tête avant d'être remplacés par d'autres.
- L'**orbite géosynchrone** ou GEO (*Geostationary Earth Orbit*) est particulière, car elle se situe à une altitude de quelques 35786 km et de ce fait la vitesse de déplacement d'un satellite sur cette orbite est synchrone avec la rotation de la planète. On dit alors que la période orbitale est égale à la période de rotation sidérale de la planète (soit 23 heures 56 minutes et 4,1 secondes). Si l'orbite est dite non inclinée, circulaire (non elliptique) et située dans le plan de l'équateur, on parle d'orbite

géostationnaire. Dans ce cas précis, relativement à la position d'un observateur à la surface de la Terre, le satellite paraît immobile. Le satellite METEOSAT 10 (MSG), par exemple, est en orbite géostationnaire non loin de Libreville au Gabon, à une altitude de 35798,7 km. Il peut ainsi capturer des images de tout l'hémisphère, qu'il est théoriquement possible de recevoir, mais son éloignement devient vite problématique pour l'amateur, puisque le signal est relativement faible.

- Et les **orbites hautes** ou HEO (*High Earth Orbit*) au-delà de 35000 km d'altitude où se trouvent des satellites avec une période orbitale plus importante que celle de la Terre et donnant donc l'impression de se déplacer lentement dans le sens inverse de celui de rotation de la planète. Il existe peu de satellites civils sur ces orbites et ils ne représentent donc pas réellement d'intérêt en termes de réception amateur de signaux.

À cela s'ajoutent également d'autres caractéristiques orbitales qu'il convient de connaître si l'on veut « chasser » les satellites :

- Une **orbite circulaire**, comme son nom l'indique, forme un cercle dont le centre est la planète. Cette notion s'oppose à celle de l'**orbite elliptique** qui, par définition, possède un point le plus éloigné de la Terre, l'*apogée*, et un point le plus proche de la Terre, le *péri-*



gée. On parle d'**excentricité** d'une orbite pour caractériser cette notion.

- Une orbite peut être inclinée ou non. L'**inclinaison orbitale** est un angle relatif à l'équateur qui, s'il est nul définit une **orbite équatoriale**, le satellite tourne sur le même plan que l'équateur et, depuis le sol, ne se déplace pas sur l'axe nord-sud. On parle d'**orbite quasi-équatoriale** lorsque cet angle est inférieur à 10°.
- Si l'inclinaison orbitale forme un angle proche de 90°, et donc quasi-perpendiculaire à l'équateur, on parle d'**orbite polaire** ou **quasi-polaire**, car un satellite sur une telle orbite donne l'impression de passer d'un pôle à l'autre (dans un sens sur un hémisphère et dans l'autre de l'autre côté de la planète).

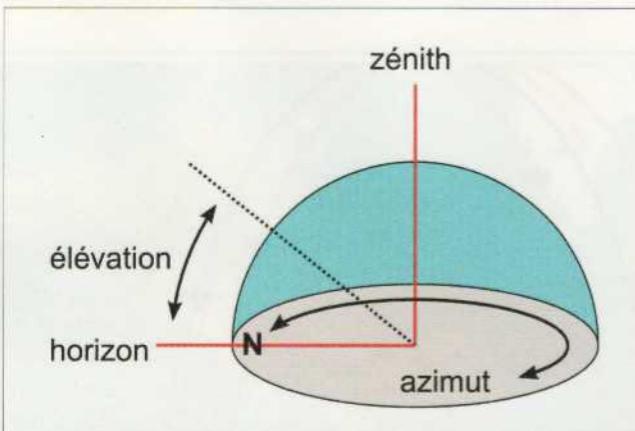
• Enfin, une inclinaison entre 10° et 80° (ou plus généralement simplement inférieure à 90°) forme une **orbite directe** ou **prograde**, car les objets qui s'y trouvent tournent dans le sens des aiguilles d'une montre lorsqu'on regarde depuis le pôle Nord (comme la planète elle-même). Inversement, si l'angle est supérieur à 90°, l'inclinaison se trouve dans l'autre sens et on parle d'**orbite rétrograde**, car les satellites se déplacent dans le sens opposé de celui de la rotation terrestre.

Enfin, pour lier tout cela et éventuellement s'assurer d'un bon mal de tête, nous avons les **orbites hélio-synchrones** qui, comme le dit Wikipedia, désignent des orbites « géocentriques légèrement rétrogrades dont on choisit l'altitude et l'inclinaison de sorte que l'angle entre le plan d'orbite et la direction du soleil demeure quasiment constant ». De quoi ? Traduction : il s'agit d'orbites relatives à la Terre (géocentriques), avec une inclinaison supérieure à 90° faisant en sorte que le satellite se déplace dans le sens inverse de la rotation terrestre, mais dont on a soigneusement défini l'angle et l'altitude pour que le soleil se trouve toujours au même endroit relativement à l'orbite du satellite. Le terme anglais de *Sun-Synchronous Orbit (SSO)* est plus parlant puisque, effectivement, ces orbites

Le site n2yo.com permet de suivre le déplacement et l'orbite d'une quantité phénoménale de satellites, mais aussi, et surtout de l'ISS. Parmi les quelques 20000 objets référencés, on trouve ceux par lesquels il convient de débuter sa carrière de chasseur de satellites : NOAA-15, NOAA-18 et NOAA-19.



Pour repérer un objet dans le ciel, qu'il s'agisse d'un satellite ou d'un astre, on utilise généralement un système de coordonnées horizontales reposant sur deux valeurs : l'azimut et l'élévation.



permettent aux satellites d'observation de la Terre, comme l'ENVISAT de l'ESA, d'obtenir des clichés et des mesures avec exactement le même ensoleillement de la surface.

« Pourquoi diable s'encombrer la tête de toutes ces choses ? » me direz-vous, car tout cela peut effectivement paraître relever davantage de l'astronomie que de la réception de signaux radio. En réalité, comprendre la manière dont fonctionne le passage des satellites et la terminologie associée est intimement lié à la réception. La plupart des satellites météo qu'il est possible de facilement capter sont constitués de satellites à défilement (ou défilant) car, s'ils bougent rapidement, c'est parce qu'ils sont plus proches de la Terre. Et s'ils bougent, mieux vaut savoir dans quelle direction, à quelle vitesse et quelle sera leur trajectoire dans le ciel.

Il existe des applications, comme Gpredict (GNU/Linux) ou WXtrack (Windows), et des sites web comme n2yo.com, permettant de prédire le passage de centaines de satellites et vous fournir un calendrier complet mais, sans connaître le sens des éléments qui caractérisent une orbite, une bonne partie des informations fournies sont inutilisables. Bien sûr, vous pouvez vous satisfaire d'un simple trait s'affichant sur une carte et réagir au cas par cas, mais une complète compréhension de l'ensemble du système orbital vous apporte un véritable plus et une logique globale facilitant implicitement l'assimilation de points plus précis. C'est un peu comme arriver à cuisiner plus facilement, car on sait quelles manipulations impactent les ingrédients, plutôt que de suivre bêtement une recette.

À propos de cartes et de trajectoires justement, la plupart des sites et applications fournissent une représentation cartographique de la trajectoire de passage pour un satellite donné. Cette représentation, bien que très intéressante visuellement pour avoir une vague idée de la position de l'objet, n'est pas très pratique lorsqu'on tente de rapprocher cela de sa propre situation géographique. En d'autres termes, c'est bien joli, mais ça n'aide pas vraiment pour trouver le satellite dans le ciel lorsqu'il passe.

Comme nous allons le voir dans la suite, il n'est pas forcément nécessaire de « viser » le satellite avec une antenne lors de son

passage mais, lors de vos premiers essais (souvent décevants), savoir où passe le satellite vous permet de comprendre les problèmes de réception ou l'impact de l'environnement sur la qualité des signaux. Pour faciliter le repérage, la situation relative du satellite par rapport à votre propre position est donnée avec un **système de coordonnées horizontales**. Ce système également appelé **système local** ou **système de coordonnées alt-azimutales** est utilisé depuis très longtemps et bien avant que l'humanité n'envoie le moindre objet en orbite. Il utilise deux valeurs pour repérer un objet dans un environnement 3D :

- L'**azimut** est défini par un angle depuis le nord cardinal entre 0 et 360°. Pour comprendre cette valeur, c'est tout simple, regardez vers le nord, vous aurez un azimut de 0°. Tournez d'un quart de tour vers votre droite, et vous avez un azimut de 90° et regardez donc vers l'est. Tournez encore d'un quart de tour et vous voici à l'azimut 180°, regardant au sud. Tournez encore d'un quart... vous avez compris. Cette valeur en degrés suffit, en compagnie de son point cardinal de référence (souvent le nord), à déterminer l'orientation du plan vertical sur lequel se trouve l'objet recherché.
- L'**élévation**, parfois nommée hauteur est également un angle, mais cette fois entre l'horizon (plan horizontal) et l'objet visé. Il varie normalement entre 0°, à l'horizon, et

90°, au zénith, juste au-dessus de vous. Il est également possible, dans certaines conditions, d'avoir une élévation négative, dans le cas où un objet se trouve sous la ligne d'horizon et n'est donc pas visible, ou éventuellement si vous vous trouvez sur une montagne et que le satellite apparaît plus bas que vous.

Où que vous soyez donc (vous risquez d'irrésistiblement devoir vous balader pour avoir une meilleure réception), la seule chose qui vous faut se limite à savoir où est le nord et d'avoir une bonne capacité d'approximation des angles pour trouver, grossièrement, dans quelle direction se trouve le satellite de vos rêves. Ce n'est pas tout, vous le comprendrez rapidement après quelques essais, tous les passages ne se valent pas. Tantôt, le satellite passera avec une élévation maximum de quelques degrés et il ne sera pas intéressant de tenter une réception car, qui dit faible élévation dit forcément distance plus importante et donc davantage d'atténuation du signal. Personnellement, en dessous d'une élévation de 30° et étant donné ma situation géographique, je n'essaie pas d'obtenir un signal de qualité.

Enfin, deux termes qu'on retrouve souvent en ligne ou dans les applications sont liés à ce système de coordonnées horizontales : AOS pour *Acquisition Of Signal* (obtention du signal) lorsque le satellite apparaît à l'horizon et LOS pour *Loss Of Signal* (perte du signal) lorsqu'il disparaît sous l'horizon. AOS et LOS sont associés à une date/heure et un azimut vous permettant donc de savoir quand et où le satellite apparaît puis disparaît dans le ciel.

1. QUELS SATELLITES SUIVRE ET ESSAYER D'ÉCOUTER ?

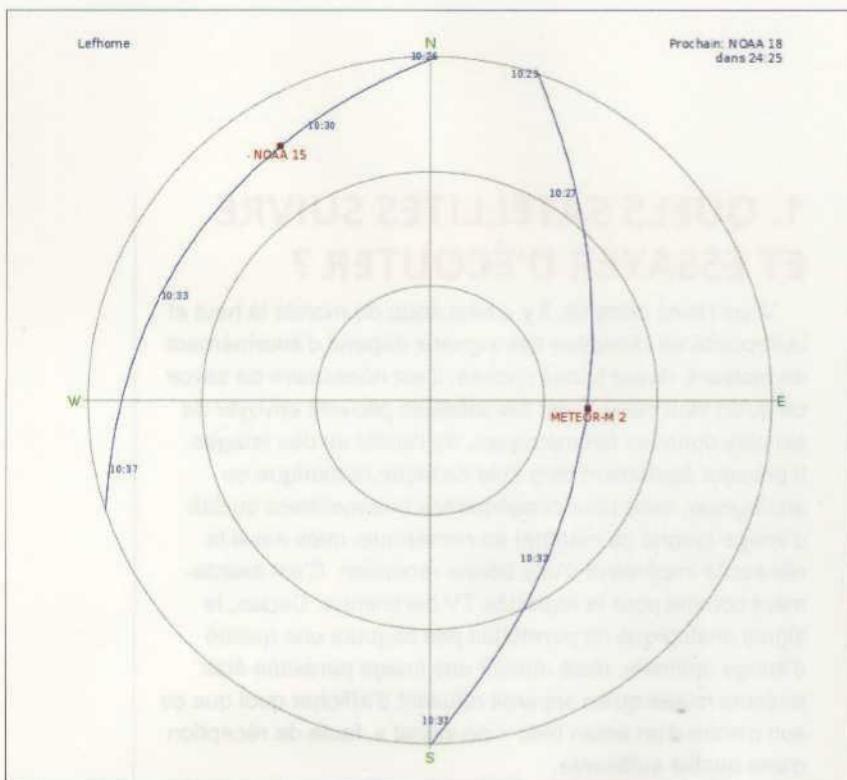
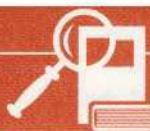
Vous l'avez compris, il y a beaucoup de monde là haut et la capacité de réception des signaux dépend énormément de facteurs. Avant toutes choses, il est nécessaire de savoir ce qu'on veut obtenir, car les satellites peuvent envoyer de simples données télémétriques, de l'audio ou des images. Ils peuvent également faire cela de façon numérique ou analogique, avec pour conséquence une meilleure qualité d'image (quand ça marche) en numérique, mais aussi la nécessité impérative d'une bonne réception. C'est exactement comme pour la regrettée TV hertzienne. Certes, le signal analogique ne permettait pas toujours une qualité d'image optimale, mais obtenir une image parasitée était toujours mieux qu'un appareil refusant d'afficher quoi que ce soit d'autre d'un écran bleu « no signal », faute de réception d'une qualité suffisante.

Dans le cadre des expérimentations qui vont suivre, l'objectif sera de recevoir des images, tout simplement parce que c'est là quelque chose qu'on peut « collectionner » et sur quoi capitaliser. Les données télémétriques sont intéressantes, mais n'ont pas cet aspect « démonstratif », et il en va de même pour d'éventuels enregistrements audio (de l'ISS par exemple). C'est certes amusant, mais cela reste très ponctuel alors qu'avec la réception d'images envoyées par les satellites météorologiques, en revanche, chaque tentative est l'occasion d'obtenir quelque chose d'unique, de compléter un « bestiaire » et donc un objectif et, surtout, c'est quelque chose de susceptible de présenter un certain intérêt, même pour un regard profane (vous savez, lorsque vous restez des heures à écouter des « schhhhhh » devant votre écran).

Nous souhaitons donc obtenir des images, et ce relativement facilement, afin d'éviter d'être découragés trop rapidement par une succession d'échecs n'apportant strictement aucun résultat. Ce qui signifie donc qu'il faut s'intéresser en premier lieu aux émissions analogiques.

Dans le domaine de l'imagerie satellite, trois principaux types de systèmes de transmission existent :

- APT pour *Automatic Picture Transmission* est le système le plus ancien (années 60) et repose sur une transmission analogique en VHF sur ~137 Mhz avec une modulation FM. C'est le système par lequel commencer puisque c'est celui le plus tolérant aux problèmes de



Le système de coordonnées horizontales permet de facilement visualiser la trajectoire sous la forme d'un diagramme comme celui-ci. Ici, on peut voir que le satellite METEOR-M 2 passe du nord au sud et a actuellement une élévation d'environ 45°. NOAA-15, sur la gauche, est apparu au nord et disparaîtra près de l'ouest tout en restant à peine au-dessus de l'horizon.

réception, mais également celui dont les jours sont malheureusement comptés. Les principaux satellites utilisant APT sont ceux du programme POES (*Polar Orbiting Environmental Satellites*) de l'agence Américaine NOAA (*National Oceanic and Atmospheric Administration*). Sur les cinq satellites « NOAA » mis en orbite, seuls trois sont encore en fonction : NOAA-15 (1998), NOAA-18 (2005) et NOAA-19 (2009).

- LRPT pour *Low-Rate Picture Transmission*) est une transmission numérique utilisée depuis 2000, également en VHF sur ~137 Mhz. Elle est utilisée par les satellites russes Meteor-M. Étant donné que la fréquence (porteuse) de 137 Mhz est la même que pour l'APT, ce sera l'étape suivante logique après un certain nombre de réceptions réussies en analogique. La modulation utilisée pour LRPT est QPSK (modulation par changement de phase avec quatre valeurs de phase possibles) qui est non seulement plus complexe à traiter, mais demande également un signal de relativement bonne qualité.
- HRPT (*High-Resolution Picture Transmission*) est le grand frère de LRPT destiné à la transmission de données numériques pour des images de plus haute résolution. Qui dit haute résolution

dit, en toute logique, débit de données plus important et donc fréquence de porteuse plus élevée. Le signal transmis est également modulé par changement de phase, mais sur une fréquence UHF entre 1,670 et 1,710 GHz avec une bande passante de 4500 KHz. HRPT est également utilisé par les satellites NOAA sous cette désignation alors qu'EUMETSAT utilise AHRPT (« A » pour *Advanced*) et les satellites chinois Feng Yun utilisent CHRPT (« C » pour *Color*). La réception HRPT nécessite un signal clair et donc une antenne directionnelle de qualité (souvent une parabole), avec un système de motorisation pour cibler le satellite. Ce n'est donc ni à la portée du tout débutant, ni ce par quoi il est conseillé de commencer.

Comprenez bien qu'un système de transmission n'est pas nécessairement lié à un satellite ou un autre. APT est utilisé par NOAA-15, NOAA-18 et NOAA-19, mais également par Meteor-M N2. De la même façon, ce satellite russe transmet également des signaux AHRPT et LRPT, tout comme les trois NOAA transmettent également en HRPT, ou encore Metop-B d'EUMETSAT/ESA transmet en AHRPT et en LRPT. Voyez cela comme des périphériques de transmission pouvant être embarqués dans un satellite, tout comme votre ordinateur portable peut avoir de l'Ethernet et/ou du Wifi et/ou Bluetooth.

Pour obtenir des informations sur un satellite donné, qu'il s'agisse de ses transmetteurs, ses appareils embarqués, son état ou encore son affiliation à une famille ou un projet, le plus simple est de consulter une base de données en ligne comme OSCAR (*Observing Systems Capability Analysis and Review Tool*) de l'Organisation Météorologique Mondiale (OMM ou WMO en anglais) : <https://www.wmo-sat.info/oscar/satellites>.

2. PETIT RAPPEL SUR RTL-SDR

Pour obtenir une image envoyée par un satellite, il vous faut un récepteur radio connecté à votre PC, votre Mac ou votre Raspberry Pi. Jusqu'à il y a quelques années, la seule manière de procéder était de disposer d'un matériel radio capable de recevoir le signal, le traiter et d'envoyer le résultat de ce traitement (démodulation) à un système informatique, généralement via la carte son, pour que celui-ci puisse décoder les informations et produire le résultat final.

Cette technique est toujours d'usage, mais l'équipement nécessaire est souvent onéreux et représente un investissement assez conséquent pour qui n'a pas forcément pour objectif de faire d'une telle activité un loisir à plein temps. Heureusement, les choses ont radicalement changé il y a quelques années avec l'apparition des périphériques dits « RTL-SDR ». Suite à une tentative de développement d'un pilote Linux

pour un récepteur DVB-T (TNT), Antti Palosaari se rend compte que le matériel sur lequel il travaille est en réalité un récepteur radio « universel ». Le périphérique se contente de recevoir les signaux et de les numériser avant de les envoyer à l'ordinateur en USB et c'est un logiciel sur ce dernier qui fait tout le travail de la démodulation à l'affichage en passant par le décodage.

Cette approche n'est pas nouvelle et s'appelle communément la **radio**

logicielle ou, en anglais, la **SDR** pour *Software Defined Radio*. Ce qui est très intéressant avec ces récepteurs TNT c'est qu'ils ne se limitent pas à la fréquence utilisée pour une telle application, mais permettent de recevoir ainsi des signaux sur une très vaste gamme de fréquences (de 65 MHz à 2,300 Ghz) couvrant la TNT bien sûr, mais également la radio FM, les transmissions d'informations de l'aviation civile, les messages des pagers, les fréquences GSM, etc. Tout n'est plus qu'une question de développement d'applications, d'outils et de bibliothèques et il devient possible de recevoir et traiter n'importe quel signal, le tout pour une poignée d'euros de matériel.

Depuis cette découverte et au fil des développements qui ont suivi, tout un écosystème a vu le jour permettant à tout un chacun de découvrir le domaine, d'expérimenter et d'apporter sa contribution à l'ensemble sous forme de code open source. Et s'il y a quelque chose que les hackers du monde entier savent faire lorsqu'ils peuvent avoir entre les mains un nouveau jouet, c'est précisément expérimenter à tout va et développer des outils fantastiques.

Mais l'aventure ne s'arrête pas là, car l'attrait pour ce matériel initialement destiné à un autre usage a donné l'occasion à certains fabricants d'adresser ce besoin particulier en produisant de nouveaux équipements, améliorés et plus adaptés

| Date | Az | El | Portée | Footp |
|---------------------|---------|--------|--------|-------|
| 2018/06/13 16:27:11 | 153.90° | 0.00° | 3390 | 6238 |
| 2018/06/13 16:27:58 | 153.32° | 2.95° | 3080 | 6240 |
| 2018/06/13 16:28:45 | 152.58° | 6.24° | 2771 | 6242 |
| 2018/06/13 16:29:32 | 151.62° | 10.00° | 2466 | 6245 |
| 2018/06/13 16:30:19 | 150.33° | 14.41° | 2165 | 6248 |
| 2018/06/13 16:31:06 | 148.51° | 19.75° | 1872 | 6251 |
| 2018/06/13 16:31:53 | 145.78° | 26.46° | 1592 | 6254 |
| 2018/06/13 16:32:39 | 141.28° | 35.19° | 1336 | 6257 |
| 2018/06/13 16:33:26 | 132.82° | 46.60° | 1118 | 6260 |
| 2018/06/13 16:34:13 | 113.23° | 60.00° | 968 | 6264 |
| 2018/06/13 16:35:00 | 66.58° | 67.21° | 919 | 6267 |
| 2018/06/13 16:35:47 | 23.80° | 58.28° | 985 | 6271 |
| 2018/06/13 16:36:34 | 6.58° | 45.03° | 1148 | 6274 |
| 2018/06/13 16:37:21 | 358.92° | 34.07° | 1373 | 6278 |
| 2018/06/13 16:38:08 | 354.78° | 25.71° | 1633 | 6281 |
| 2018/06/13 16:38:55 | 352.24° | 19.25° | 1915 | 6284 |
| 2018/06/13 16:39:42 | 350.55° | 14.08° | 2209 | 6288 |
| 2018/06/13 16:40:29 | 349.37° | 9.79° | 2511 | 6291 |
| 2018/06/13 16:41:15 | 348.51° | 6.12° | 2816 | 6294 |
| 2018/06/13 16:42:02 | 347.87° | 2.90° | 3125 | 6297 |
| 2018/06/13 16:42:49 | 347.38° | -0.00° | 3434 | 6299 |

Print

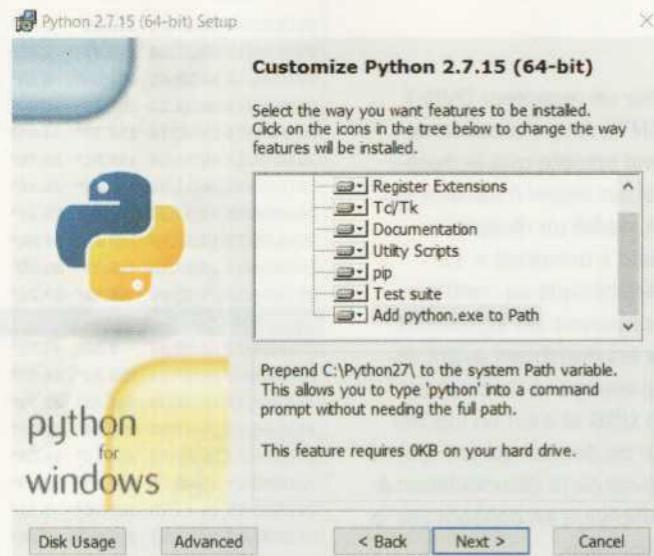
Save

Close

Les applications comme Gpredict fournissent toutes les informations nécessaires pour un observateur concernant le passage d'un satellite. Sur cette capture, nous avons une acquisition de signal (AOS) à 16h27 et une perte (LOS) à 16h42 avec une élévation maximale à 67,21° où l'objet sera presque au-dessus de nous.



Installer Python sous Windows n'est pas très difficile, mais il est important d'activer l'option « Addpython.exe to Path » afin que l'interpréteur soit accessible de n'importe où dans le système. Notez qu'avec les dernières versions disponibles, Pip est également automatiquement installé, ce qui facilite grandement l'ajout de modules.



à la SDR : changement du connecteur d'antenne de TV à SMA par exemple, amélioration de la qualité générale du circuit, meilleure dissipation thermique, utilisation d'un oscillateur plus stable et plus précis, atténuation du bruit de l'alimentation, etc.

On trouve donc maintenant des produits spécialement destinés à la SDR sous la désignation « RTL-SDR », sensiblement plus chers que le simple récepteur TNT détourné, mais également plus performants, tout en restant dans une gamme de prix raisonnable, en dessous des 30€ ou 40€. Comme le récepteur TNT initial, ils se composent de deux éléments, un tuner (Elonics E4000 ou Rafael Micro R820T2) et un convertisseur analogique-numérique Realtek RTL2832U. Le tuner, piloté par l'ordinateur, secale sur une fréquence et le convertisseur numérise le signal avant de transmettre les données en USB.

Pour contrôler le matériel et traiter le signal, il vous faut un logiciel adapté et, là aussi, le temps et les efforts des développeurs ont porté leurs fruits. Côté GNU/Linux, nous avons *SDRangel*, *GQRX* et *Linrad* et sous Windows, les plus utilisés sont *SDR#* et *HDSDR*. Plus surprenant, il existe également des applications Android relativement complètes comme *SDR Touch* et *RFAnalyzer*, parfaitement utilisables si votre tablette ou votre smartphone supporte l'USB hôte. Ceci s'ajoute, bien entendu, au classique GNU Radio, dont la création date de bien avant l'arrivée des périphériques RTL-SDR.

Quel que soit le matériel que vous utiliserez ou que vous utilisez, son type, son origine et son prix ne joueront qu'une

petite part dans les expérimentations qui vont suivre. Bien entendu, la sensibilité du récepteur et sa qualité générale restent importants puisqu'il est fort peu probable que vous vous limitez à ce type d'application (personnellement, j'aime beaucoup le matériel de chez NooElec comme le NESDR SMArt ou le NESDR SMArt XTR disponibles via Amazon), mais lorsqu'on parle de réception satellite, les clés du succès résident en premier lieu dans deux autres points : l'antenne et l'environnement, deux éléments que nous allons voir ensemble sous peu.

3. RTL-SDR EN BREF

Afin de rafraîchir vos connaissances si vous avez déjà expérimenté avec un périphérique RTL-SDR, ou de prendre tout cela en main si c'est votre première fois, nous allons faire d'une pierre deux coups avec une petite mise en jambe simple. Une antenne spécifique ne sera pas forcément nécessaire puisque l'objectif ne sera pas de réellement capter et enregistrer le signal d'un satellite, mais de simplement visualiser sa présence en tirant quelques leçons.

Nous utiliserons ici l'un des outils en ligne de commandes fournis avec le support RTL-SDR. Pour installer ce support sur un PC GNU/Linux ou une Raspberry Pi, ceci se limitera à l'installation du paquet **rtl-sdr** (et éventuellement **librtlsdr-dev** si vous comptez compiler des outils par

la suite) avec `sudo apt-get install rtl-sdr librtlsdr-dev`. Inutile de préciser que cette solution est celle que je préconise puisqu'un grand nombre d'outils SDR sont beaucoup plus faciles à installer et utiliser sur ce système (comme `rtl-433`, `multimon-ng`, `dump1090`, etc.). Il sera sans doute nécessaire de débrancher et rebrancher le récepteur en USB, car le paquet `librtlsdr0` (dépendance de `rtl-sdr`) installera un fichier de configuration pour udev (`60-librtlsdr.rules`) faisant en sorte qu'il ne soit pas nécessaire d'être super-utilisateur pour accéder au matériel. Il est possible que vous ayez éventuellement à redémarrer le service udev avec `sudo service udev restart` pour que les changements soient appliqués avant la première déconnexion/reconnexion.

Sous Windows, les choses sont un peu plus compliquées (voire pénibles) puisque vous devrez installer le pilote USB universel et gestionnaire de pilotes Zadig (<http://zadig.akeo.ie/>) avant toutes choses. Une fois votre récepteur RTL-SDR branché au PC, démarrez Zadig **en tant qu'administrateur**. Dans la fenêtre qui apparaît se trouve un menu déroulant affichant présentement « *Bulk-In, Interface (Interface 0)* ». Dans cette liste doit normalement figurer votre récepteur RTL-SDR, sans doute avec un nom contenant « *RTL2832U* ». Si ce n'est pas le cas, passez par le menu *Option* et décochez *Ignore Hubs or Composite Parents*, votre périphérique doit alors être visible, sélectionnez-le.



Le reste de la fenêtre se divise en deux parties, à gauche se trouvent les informations sur le périphérique, dont la mention d'un éventuel pilote déjà associé, et à droite vous pouvez choisir le pilote qui doit être utilisé désormais. Dans les deux cas, vous devez choisir le pilote WinUSB. Si votre périphérique n'a pas encore de pilote le bouton sera marqué *Install Driver* et *Replace Driver* (remplacer pilote) dans le cas contraire. Vérifiez tous les paramètres et cliquez sur le bouton, puis confirmez en choisissant *Oui* dans la mini-fenêtre qui vient de s'afficher. Enfin, après quelques secondes un message « *The driver was installed successfully* » s'affiche témoignant du bon déroulement des opérations.

Voici un petit exemple de la diversité qui existe en termes de récepteurs RTL-SDR. Les matériels les plus récents comme le NESDR SMArt en bas ou le minuscule NESDR Nano 2+ sur la gauche sont spécifiquement conçus pour la SDR, mais possèdent une architecture similaire aux premiers récepteurs DVB-T utilisés pour ce type de réception (comme celui en haut et massivement bricolé).



Ce que Zadig permet de faire consiste à court-circuiter la prise en charge d'un périphérique par Windows, le rendant accessible directement par les applications capables de le gérer. C'est le cas, par exemple, des applications *SDR#* et *HDSDR*, mais également d'autres outils, en ligne de commandes, qui sont des transpositions (portages) de ce qui est généralement utilisé sous GNU/Linux, comme les outils standards RTL-SDR. Malheureusement, Windows n'est pas le terrain de jeu de prédilection des bidouilleurs et hackers les plus créatifs et les plus capables, et c'est là qu'on se heurte à une certaine difficulté.

Compiler ces outils, comme on le fait généralement sous GNU/Linux est, sous Windows, quelque chose de relativement difficile. On doit donc, très souvent, se rabattre sur le résultat du travail de quelqu'un d'autre et le plus souvent il ne s'agit là que d'une manipulation ponctuelle, pour le plaisir de relever le défi. C'est ainsi que les outils RTL-SDR toujours à jour sous GNU/Linux, ne sont disponibles en version compilée pour Windows que dans une version datant de janvier 2014. Ceux-ci sont disponibles ici : <https://osmocom.org/attachments/2242/ReIWithDebInfo.zip>. Le fichier Zip pourra être désarchivé n'importe où, puis on accèdera au dossier **rtl-sdr-release**, sous-dossier **x32** ou **x64** selon le caractère 32 ou 64 bits de votre Windows, via l'invite de commandes Windows (**cmd.exe**) pour pouvoir enfin exécuter les différents utilitaires placés à cet endroit (je vous avais dit que c'était pénible). Vous pouvez également utiliser le PowerShell qui est sans doute plus agréable à utiliser (Win+X et « Windows PowerShell »).

Et si vous vous posez la question, non, le *Windows Subsystem for Linux* (WSL) installable dans Windows 10 et permettant d'utiliser des applications et outils GNU/Linux dans Windows (cf. *Hackable* n°23), ne permet pas d'accéder aux périphériques USB et n'est donc pas utilisable pour prendre en charge un récepteur RTL-SDR. Du moins, pour l'instant...

Que ce soit sous GNU/Linux ou Windows, à ce stade vous pouvez vérifier très simplement que le périphérique soit correctement accessible en utilisant la commande **rtl_eeprom** qui devrait vous afficher quelque chose comme :

```
Found 1 device(s):
  0: Generic RTL2832U OEM

Using device 0: Generic RTL2832U OEM
Found Elonics E4000 tuner

Current configuration:

Vendor ID:          0x0bda
Product ID:         0x2838
Manufacturer:       Realtek
Product:            RTL2838UHIDIR
Serial number:      00000001
Serial number enabled: no
IR endpoint enabled: yes
Remote wakeup enabled: no
```

Ceci nous donne un aperçu du matériel détecté en remontant des informations comme son nom de produit et son numéro de série, démontrant ainsi que tout est fonctionnel. Vous pouvez également utiliser **rtl_test** pour tester la réception et la communication en USB. Si tout fonctionne correctement, vous ne devriez pas voir de message apparaître après la ligne « *Reading samples*

in async mode... ». Vous pourrez stopper l'exécution du test avec CTRL+C et un dernier message devrait vous afficher « Samples per million lost (minimum): 0 » signifiant que rien n'a été perdu et que le test est réussi.

4. PETITE SURVEILLANCE, JUSTE POUR VOIR

Parmi les outils RTL-SDR se trouve **rtl_power** dont je vous avais parlé en détail dans le numéro 9 (pfiou ! Déjà ?!). Cet utilitaire permet de surveiller une plage de fréquences et d'enregistrer le niveau de signal reçu pour chaque fréquence dans cette plage sous la forme de valeurs dans un tableau au format CSV. Ce ne sont pas les signaux qui sont enregistrés, mais simplement leur intensité au fil du temps. Ainsi, en laissant fonctionner **rtl_power** entre 137 Mhz et 138 Mhz, sur une plage de temps relativement importante, il est possible de stocker suffisamment d'informations pour ensuite transformer cela en image et voir le passage d'un éventuel satellite NOAA.

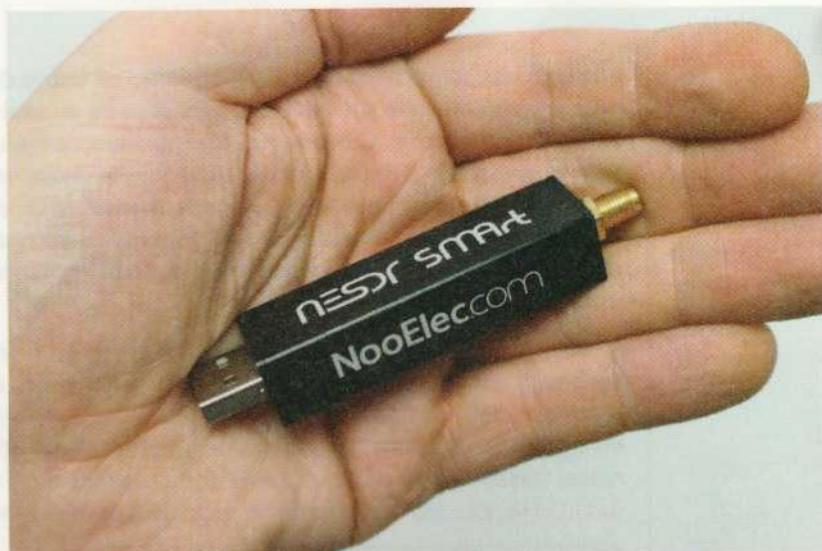
Pour tricher un peu, vous pouvez vous aider d'un site comme n2yo.com ou d'une application comme Gpredict pour estimer le moment de passage du satellite. Ce qui est impératif, en revanche, c'est de faire cette réception dans de bonnes conditions : essayez d'avoir un ciel dégagé ou, au pire, de ne pas avoir deux ou trois étages d'un immeuble au-dessus de votre tête ou ne pas vous trouver enclavé entre des bâtiments. Moins il y aura d'obstacles entre l'antenne et le ciel, meilleure sera la réception, et ceci est également valable pour tenter de recevoir réellement un signal comme nous le ferons par la suite.

Il ne vous restera plus, ensuite, qu'à utiliser **rtl_power** avec, par exemple :

```
$ rtl_power -f 136900k:138000k:250 \
-i 5s -w hamming -c 20% satellite.csv
```

L'option **-f** permet de préciser la plage de fréquence avec, ici, de 136,9 Mhz (136900 KHz) à 138 Mhz et un pas de 250 Hz. Plus la plage est grande, plus **rtl_power** sera obligé de la découper en tranches. Plus le pas est petit, plus le fichier résultant sera gros. L'option **-i** précise un temps d'intégration. Ici, les données collectées sur 5 secondes formeront une ligne du fichier. Plus cette valeur sera petite, plus il y aura de lignes dans le fichier, pour un intervalle de temps donné.

Enfin, **-w** et **-c** permettent de procéder à ce qu'on appelle du fenêtrage, qui est une technique permettant d'écartier une partie du signal qui se trouve aux limites hautes et basses du spectre de fréquence. Imaginez simplement que vous souhaitez prendre



Les récepteurs SDR comme ce NESDR SMART de NooElec ne coûtent qu'une trentaine d'euros, mais intègrent un certain nombre d'améliorations par rapport à un simple récepteur TNT DVB-T : oscillateur fiable bruit compensé en température (TCXO), alimentation adaptée et silencieuse, boîtier alu pour une dissipation thermique optimale, blindage électromagnétique des parties sensibles, connecteur SMA, etc.



en photo un spaghetti géant, mais que celui-ci se trouve dans une pièce de taille réduite. Vous ne pouvez pas reculer suffisamment pour prendre un seul cliché de tout le spaghetti (il n'est pas cuit et donc tout droit). Ce que vous allez faire alors est tout simplement de prendre une photo, vous déplacer sur la droite, prendre une nouvelle photo, vous déplacer, etc. Comme les bords de chaque cliché sont un peu moins nets que le reste de l'image, vous allez supprimer une partie de cette bordure et vous déplacer un peu moins avant chaque cliché. Le fenêtrage c'est à peu près cela, garder le meilleur du signal et jeter le reste. « hamming » est un type de fenêtre et les 20% c'est ce qu'on ne garde pas.

Vous pouvez éventuellement ajouter une option `-e` suivie d'une durée afin de stopper automatiquement une fois le temps écoulé. Les lettres « s », « m » et « h » désignent respectivement des secondes, des minutes et des heures. Ainsi, `-e 60m` ou `-e 1h` aura pour effet de laisser fonctionner `rtl_power` une heure avant de s'arrêter. Sans cette option, vous devrez arrêter vous-même l'utilitaire avec CTRL+C. Dans un cas comme dans l'autre vous retrouverez, dans le fichier `satellite.csv`, les données des signaux captés, mais sous une forme totalement absconse :

```
2018-06-08, 08:38:48, 136900012, 137999988, 137.02,  
1172, -32.97, -33.29, -33.00, -32.93, -33.18, -33.26,  
-33.15, -32.23, -32.50, -33.03, -33.06, -33.04,  
-32.73, -32.65, -33.32, -32.44, -31.26, -32.65,  
-32.64, -32.59, -32.95, -31.41, -31.34, -32.81,  
-30.38, -24.88, -27.05, -32.70, -32.73, -32.83,  
-33.11, -33.10, -33.21, -33.37,  
[...]
```

Pour traiter ces données et obtenir une image, vous pouvez utiliser le script que j'avais évoqué dans le numéro 9 et sensiblement amélioré par mes soins, sur la base d'un code de Kyle Keen. Celui-ci est écrit en Python et utilise un module appelé PIL (*Python Imaging Library*) pour la gestion des images et fichiers graphiques.

Python est normalement installé par défaut avec GNU/Linux sur PC et avec Raspbian sur Pi. Vous devrez cependant installer le module PIL (ou plutôt son fork appelé Pillow) avec `sudo apt-get install python-pil`.

Sous Windows c'est, sans surprise, moins facile. Vous devrez tout d'abord installer Python 2.7 en pointant votre navigateur sur <https://www.python.org/downloads/windows> et en téléchargeant « Windows x86 MSI installer » ou « Windows x86-64 MSI installer » en fonction de votre Windows (32 ou 64 bits). **ATTENTION**, lors de l'installation, une fenêtre va lister les composants/fonctionnalités à installer qui sont toutes cochées par défaut... sauf celle dont nous avons besoin : « *Add python.exe to Path* » (tout en bas). Vous **devez** activer cette option pour rendre l'interpréteur accessible facilement et automatiquement depuis le PowerShell, par exemple.

Une fois l'installation terminée, ouvrez un PowerShell puis vérifiez que Python fonctionne en utilisant la commande `python`. Vous devez vous retrouver sur l'invite du langage représentée par `%>>>%`. Tout fonctionne, vous pouvez quitter avec CTRL+C. Utilisez ensuite `pip`, qui heureusement est maintenant inclus avec Python pour Windows, pour installer PIL avec la commande `pip install pillow`.

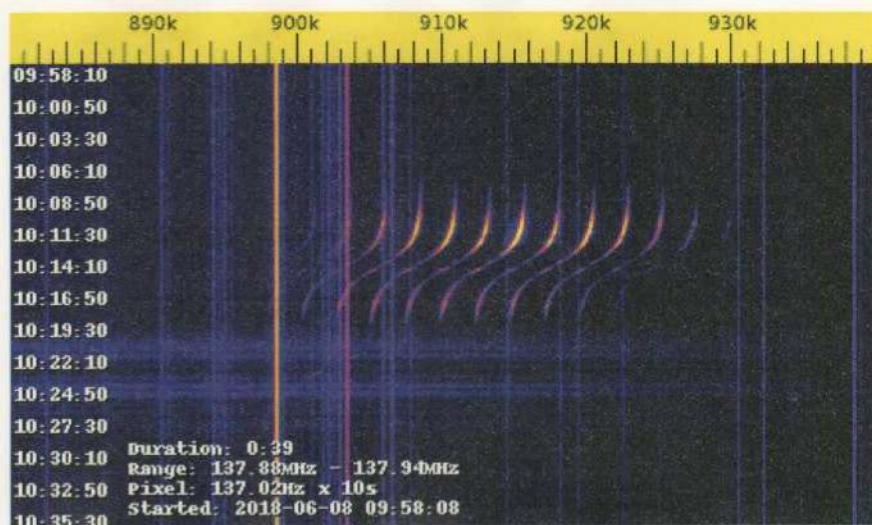
Il ne manque plus qu'une pièce au puzzle, `heatmapcg.py`. Cet outil est disponible via un de mes dépôts GitHub : <https://github.com/Lefinnois/SDR-heatmap>. Utilisez le bouton *Clone or download* et *Download ZIP* pour récupérer une archive ZIP que vous pourrez désarchiver où cela

vous chante. Sous GNU/Linux, vous pourrez invoquer la commande `./heatmapcg.py` après vous être placé dans le répertoire désarchivé du ZIP. Sous Windows, vous devrez lancer ce script Python avec `python heatmapcg.py`.

Un certain nombre d'options sont disponibles pour impacter le rendu final, mais dans sa version la plus simple, il s'agit tout bonnement de lancer le script suivi du nom du fichier CSV et d'un nom de fichier graphique (JPEG, PNG, etc.). Vous pouvez également utiliser l'option `--gradient` en précisant l'un des fichiers `gradient*.png` pour obtenir une palette de couleurs un peu plus sympathique que celle par défaut. L'option `--yঠick` suivie d'une durée permet d'ajouter une graduation verticale pour le temps écoulé. Quelque chose comme `heatmapcg.py --gradient gradient1.png --yঠick 10s satellite.csv satellite.png` devrait donner des résultats satisfaisants et plaisants.

La durée du calcul dépendra de la taille du fichier CVS et, bien entendu, de votre configuration matérielle. Un CVS de quelques 40 Mo sera traité en une trentaine de secondes sur un PC moderne et une image de 8000 par 700 pixels sera générée pour une capture de deux heures environ.

Ce que vous devriez voir dans l'image est une forme caractéristique d'une transmission APT : une série de lignes verticales parallèles s'étalant sur environ 36 KHz, comme une trace laissée par une fourchette sur la purée. Il est égale-



ment possible que vous distinguiez des bandes un peu différentes ayant également cette inclinaison étrange, mais aussi des bandes parfaitement droites.

Ces images représentent le spectre de fréquences sur l'axe x et le temps en y. J'ai dit précédemment que les satellites émettaient sur une fréquence précise :

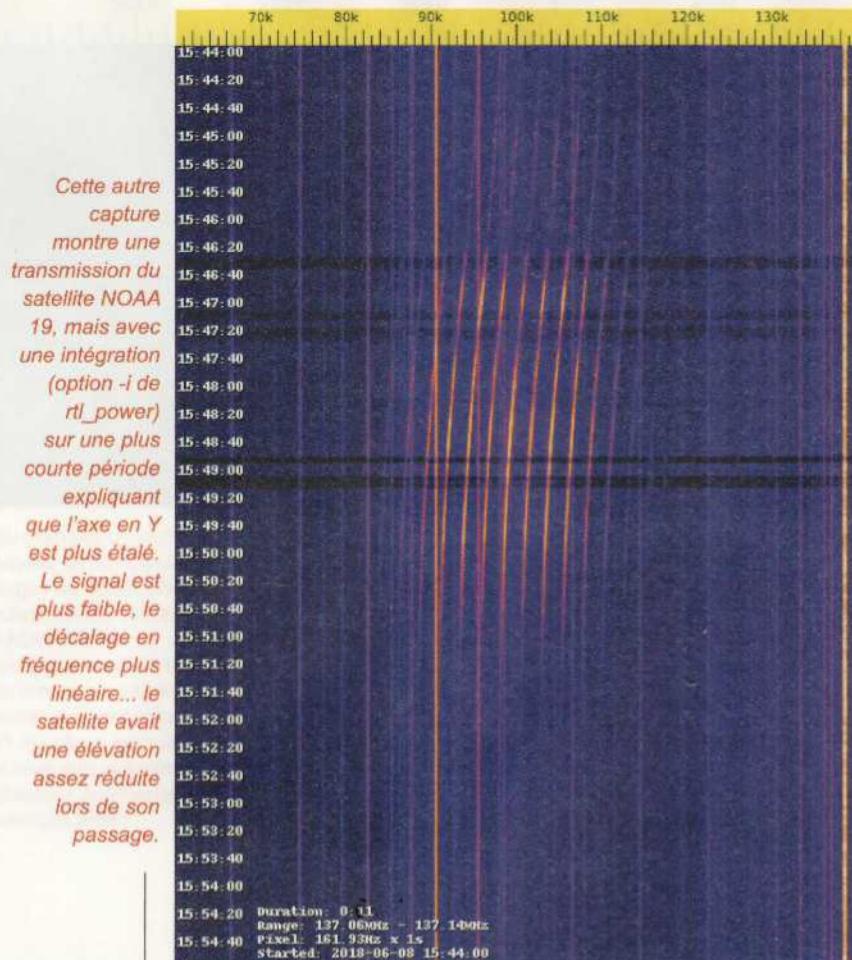
- NOAA-15 APT : 137,620 MHz
- NOAA 18 APT : 137,9125 MHz
- NOAA 19 APT : 137,1 MHz

Pourquoi diable alors ces émissions semblent commencer à une fréquence et dériver lentement sur une fréquence plus basse ? L'explication tient un seul mot : Doppler. L'effet Doppler ou plus exactement l'effet Doppler-Fizeau est un décalage de fréquence observé lorsque la distance entre l'émetteur et le récepteur varie au cours du temps. C'est exactement ce que vous entendez lorsqu'une ambulance vous croise avec la sirène activée ou lorsqu'un scooter passe devant vous : le son semble plus aigu à mesure que l'objet se rapproche puis plus grave à mesure qu'il s'éloigne.

Là, c'est la même chose, mais avec un satellite qui se déplace à quelques 26000 km/h au-dessus de votre tête et n'émet pas de son, mais un signal radio. À mesure qu'il s'approche, la fréquence baisse jusqu'à ce qu'il se trouve au-dessus de vous (ou presque) puis s'éloigne. Ce décalage en fréquence doit être pris en compte lors de la réception, d'une façon ou d'une autre.

Dans l'image ou les images obtenues, vous constaterez également que le signal du satellite n'est pas le seul présent. Les barres verticales témoignent d'émissions bien terrestres comme sur 136,975 Mhz

Voici un exemple de capture avec rtl_power. Cette forme en vague est une transmission du satellite NOAA 18 et la raison pour laquelle on observe un décalage en fréquence sur les quelques 10 minutes que dure le phénomène s'appelle l'effet Doppler.



avec un sémaphore pour les transmissions VDL mode 2, mais aussi des signaux plus intermittents pouvant être d'autres satellites ou des communications d'aéronefs.

Utiliser **rtp_power** pour procéder à une surveillance avant d'attaquer une tentative de réception est généralement une bonne idée, ne serait-ce que pour se faire une idée de la qualité et de la forme du signal reçu. Personnellement, je me suis amusé à cela pendant plusieurs jours pour affiner les réglages et surtout obtenir un tout premier résultat sans avoir à rester scotché au PC toute une journée.

5. PRÊT POUR LA SUITE ?

Il est temps de passer aux choses sérieuses et de tenter une réception. Deux points importants doivent être considérés : l'antenne et l'emplacement. Ce dernier point n'est malheureusement pas quelque chose qu'il est possible de régler technique-ment et dépendra entièrement de votre situation. Vous devez avoir une vue dégagée sur le ciel et une maison isolée en haut d'une colline, avec un horizon parfaitement ouvert, sans

la moindre ligne électrique et le moindre bâtiment, est l'emplace-ment parfait.

Ce n'est là, bien entendu, pas le cas de la plupart des habitations, en particulier en ville. Dans ce cas, il faudra faire preuve, soit de créa-tivité, soit de concession, soit de mobilité. Une antenne à la fenêtre d'un grenier fera peut-être l'affaire, mais il faudra accepter le fait qu'une partie du signal ce sera pas reçue correctement, car tout simplement bloquée par un mur, un morceau de toit, voire un bâtiment complet.

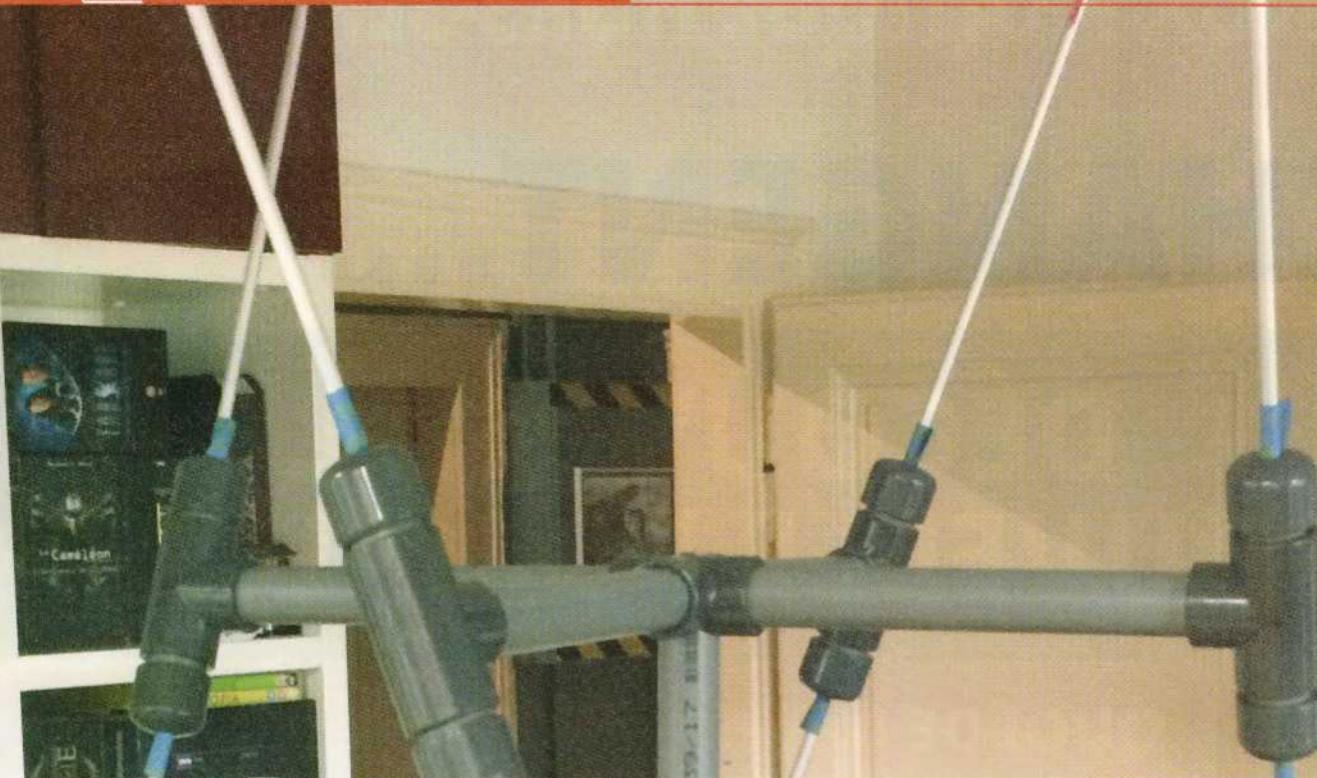
Notre objectif principal concer-nera ici les trois satellites NOAA avec une orbite quasi-polaire ré-trograde. Ce qui signifie que leurs trajectoires dans le ciel, avec une élévation suffisante ($> 45^\circ$), seront presque alignées sur un axe nord/sud. Si comme moi, vous habitez un immeuble orienté est/ouest, et que votre grenier est au sud, vous vous trouvez presque dans la pire des situations, car vous vous retrouvez avec une fenêtre de réception très réduite avec, à l'est et à l'ouest le pâté de maison, et au nord votre propre bâtiment.

Heureusement, la panoplie du chasseur de satellites météo n'est pas très lourde : un PC portable (ou une tablette, ou une Pi avec une ali-mentation d'appoint), un récepteur RTL-SDR alimenté directement en USB et une antenne. Ceci est donc parfaitement transportable et, de toute façon, les plus belles images sont celles ayant la plus légère couverture nuageuse. Donc si le temps est à la pluie en France, ce n'est pas grave, car il y a de fortes chances que vous ne verrez que des nuages de toute façon... **DB**



EN COUVERTURE

ANTENNE



SE CRÉER UNE ANTENNE POUR RECEVOIR LES IMAGES SATELLITES

Denis Bodor



La transmission de données d'un satellite ne peut être captée correctement et dans de bonnes conditions avec une antenne classique, dite « monopôle », en raison de la nature même de l'onde qui est émise. Pour recevoir un tel signal, il est donc nécessaire d'utiliser une antenne spécifique ou, plus exactement, d'en construire une.

La fabrication d'antennes pour une application donnée est un exercice mêlant plusieurs disciplines. Il est nécessaire de comprendre les principes de base du fonctionnement des ondes hertziennes, mais aussi d'avoir un certain sens du bricolage. Nous allons donc commencer par un petit rappel des bases afin que vous puissiez, modestement, comprendre le pourquoi du comment de la construction d'une telle antenne.

1. ONDES, FRÉQUENCE, LONGUEUR, CHAMP MAGNÉTIQUE ET CHAMP ÉLECTRIQUE

Pour commencer, décrivons ce qu'est une onde « hertzienne » : il s'agit tout simplement d'un rayonnement électromagnétique qui se produit lorsqu'on injecte un courant variable dans un conducteur électrique. Ce phénomène est la base de toutes transmissions radio, de la TNT au Wifi en passant par la 3G et les télécommandes de toutes sortes, mais aussi du fonctionnement d'équipements comme les pinces ampèremétriques permettant de mesurer un courant dans un câble ou encore, tout simplement, des transformateurs qu'on trouve un peu partout.

Vous connaissez sans doute le terme « électromagnétique » généralement utilisé dans un contexte comme « rayonnement électromagnétique », « perturbation électromagnétique » ou encore « champs électromagnétiques ». Mais avez-vous déjà remarqué que c'est un mot composé, « électro » et « magnétique » ? C'est tout simplement parce que ce terme couvre en réalité deux choses différentes, mais intimement liées : les champs électriques et les champs magnétiques.

Pour comprendre ce que sont ces champs et la façon dont ils sont liés, imaginez simplement deux brins conducteurs d'une longueur arbitraire et alignés sur le même axe : un dipôle. En faisant circuler un courant dans les brins, un champ magnétique se forme autour de chacun d'eux. Ce champ magnétique est ce qui fait dévier l'aiguille d'une boussole lorsqu'elle est placée à proximité d'un conducteur parcouru par un courant, comme l'avait découvert Hans Christian Oersted dès

1820. À chaque point des deux brins, des lignes de champs magnétiques apparaissent et sont **perpendiculaires** au conducteur.

La différence de potentiel, la tension, existant entre les deux brins, elle, provoque l'apparition d'un champ électrique dont les lignes se répartissent sur des plans qui se croisent au niveau de la ligne formée par les deux brins.

C'est relativement abstrait, mais nous pouvons trouver une illustration pour représenter tout cela. Prenez un oignon et plantez un crayon au travers de la base (là où sont les racines du bulbe) vers le haut (là où ça germe). Le crayon est votre paire de brins et les pelures de l'oignon représentent le champ électrique. Si vous coupez l'oignon le long du crayon, vous constaterez que la surface ainsi créée, un plan, laisse apparaître des lignes concentriques. Si vous aviez coupé l'oignon, toujours le long du crayon, mais avec un autre angle, vous auriez un autre plan et ainsi de suite. Vous avez une infinité de plans et tous se croisent (sont sécants) sur la ligne formée par le crayon. Si vous prenez un livre et le retournez en faisant se toucher les couvertures, vous avez la même chose.

Transpercez une tranche de salami danois bien ronde, en son centre, avec le crayon et vous aurez une représentation du champ magnétique à un endroit donné du crayon/brin. Multipliez mentalement les tranches le long du crayon. Maintenant, placez le crayon à l'horizontale et éloignez-vous au maximum du tout en imaginant l'oignon et les tranches de salami grossir. Avec suffisamment de recul et en ne



considérant qu'une partie de l'énorme oignon que vous avez devant vous, vous aurez l'impression que les lignes de champ électrique sont horizontales et celles du champ magnétique verticales, car la surface de l'oignon semblera être un plan et non la surface convexe d'une sphère. Le crayon lui, sera un petit point au loin, minuscule, dans l'oignon.

Le champ magnétique est créé par le passage du courant et le champ électrique par la différence de potentiel entre les brins. Bien entendu, courant et tension sont intimement liés et c'est leur variation d'amplitude qui permet la formation d'une onde radio. Sans variation il n'y a pas d'ondes électriques et donc pas de propagation de signal.

Le point important ici est l'orientation du champ **électrique** créé par les deux brins qui forment le dipôle. On parle de **polarisation** de l'onde radioélectrique. Dans notre exemple, avec le crayon (ou le dipôle) ainsi orienté, nous avons une polarisation horizontale. Inversement, un dipôle vertical permettra une réception optimale pour une onde polarisée verticalement. On représente généralement la polarisation du champ électrique par un vecteur. Si sa direction reste constante (c'est uniquement sa longueur qui change), on dit que la polarisation est linéaire. Mais si elle change au cours du temps et que son amplitude reste constante, nous avons une **polarisation circulaire**.

Les directions successives du vecteur, perpendiculairement à la direction de propagation, peuvent changer par un mouvement de rotation de deux façons : dans le sens des aiguilles d'une montre, nous avons une polarisation circulaire horaire ou RHCP en anglais pour *Right-Hand Circularly Polarized*, et inversement, une polarisation circulaire anti-horaire ou LHCP, *Left-Hand Circularly Polarized*.

Cette polarisation est importante pour la réception même si dans la pratique elle ne reste pas forcément identique à celle que lui a imprimée l'antenne d'émission. La moindre réflexion de l'onde sur un obstacle perturbera sa polarisation (entre autres). En revanche, en vue directe, un signal ayant une certaine polarisation sera plus fort s'il est capté par une antenne polarisée identiquement. Or justement, dans le cas des signaux émis par les satellites, nous sommes normalement en vue directe (sans le moindre obstacle). « Mais le satellite bouge et tourne, non ? » me direz-vous et vous avez parfaitement

raison. C'est précisément pour cela que ce type de transmission n'utilise pas une polarisation linéaire, mais circulaire. Dans le cas des signaux APT des satellites NOAA ou Meteor-M 2, il s'agit d'une polarisation RHCP. Pour une réception correcte, il vous faut donc une antenne adaptée qui ne peut donc pas être une simple antenne monopôle verticale ou un simple dipôle horizontal.

Un autre point, plus simple, est également important : la longueur d'onde. L'image classique permettant de comprendre de quoi il s'agit est celle de vagues sinusoïdales concentriques à la surface d'une marre, qui se déplacent après qu'on y ait jeté une pierre. Le point d'impact de la pierre est l'émetteur et les vagues concentriques sont l'onde émise.

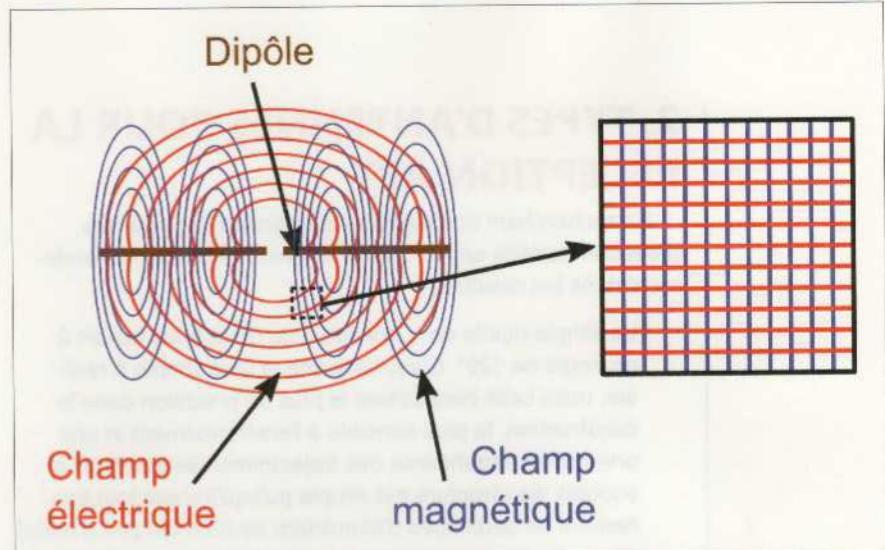
- La hauteur des vagues entre le point le plus bas et celui le plus haut est l'amplitude du signal.
- En un point donné fixe, le nombre de fois par seconde où la hauteur de la surface est maximum est la fréquence du signal, notée f ou par la lettre grec ν (nu). L'unité utilisée est le hertz : un hertz égale une répétition par seconde.
- La distance entre deux sommets de vagues est la longueur d'onde. Désignée par la lettre grecque lambda, λ . L'unité de la longueur d'onde est le mètre ou une de ses fractions décimales (centimètre, millimètre, etc.).

Ces deux dernières caractéristiques sont liées puisque la

vitesse de déplacement d'une onde, sa célérité, dans un milieu idéal (le vide) est constante : c'est la vitesse de la lumière : 299792458 m/s. Comme la vitesse est connue, ceci veut dire que la fréquence peut être utilisée pour déterminer la distance en mètres entre deux répétitions identiques de l'onde.

Prenons (pas vraiment) au hasard la fréquence de 137 Mhz, soit 137000000 fois par seconde où l'amplitude est maximum. La **période** sera de 1/137000000, soit 0,000000073 secondes. Pendant cette durée, comme la célérité de l'onde est la vitesse de la lumière (notée « c »), que nous arrondissons à 300 km/s, ce point se sera déplacé dans l'espace de $0,000000073 \times 300000000 = 2,189$ mètres. C'est la **longueur d'onde** correspondant à cette fréquence. Un calcul plus simple consiste à utiliser des unités plus pratiques et réécrire cela via la relation directe entre la fréquence et la longueur d'onde : $300/137 = 2,189$ mètres, puisque nous avons des millions de mètres par seconde et des millions de hertz par secondes.

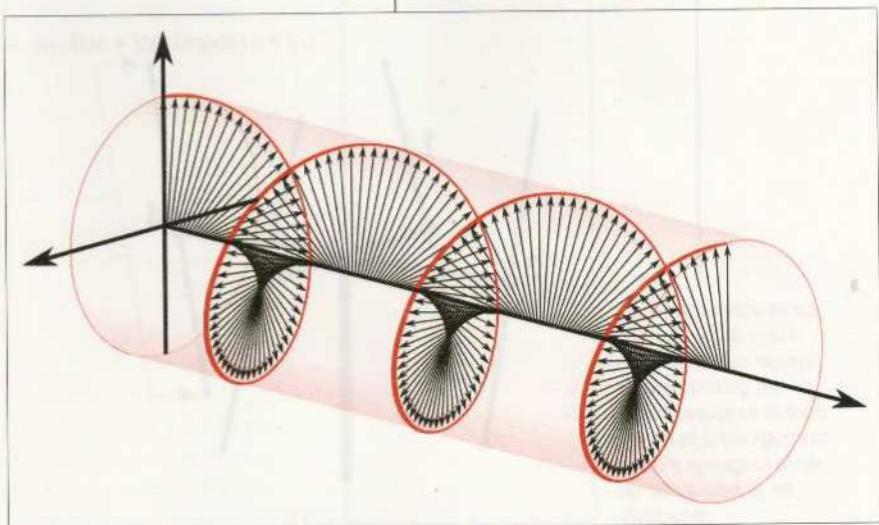
La longueur d'onde est très importante dans la conception d'antennes, car on utilise généralement une de ses fractions comme unité de mesure pour bon nombre d'éléments physiques, comme la taille d'un brin de dipôle ou la longueur d'un câble. Le quart de la longueur d'onde ou « quart d'onde » possède des propriétés particulières impactant l'impédance et donc l'efficacité de la réception. Pour être clair, et je



I'ai déjà dit dans un autre article, la réelle conception d'antennes relève presque de la magie noire. Nous nous contenterons ici de suivre des principes, de nous inspirer de concepts et de comprendre la raison de certains choix. L'important est de simplement comprendre de quoi il retourne et non de nous lancer dans la conception pure nécessitant une masse de connaissances très importante.

Pour comprendre la notion de polarisation, il faut savoir quels phénomènes physiques prennent place lorsqu'un courant circule dans un dipôle. Le rayonnement électromagnétique, comme son nom l'indique, est en réalité composé de deux éléments : un champ magnétique et un champ électrique. Leur polarisation dépend de la structure et de l'orientation de l'émetteur.

Lorsque la polarisation n'est pas linéaire, elle est elliptique et si elle est d'amplitude égale dans toutes les directions, on parle de polarisation circulaire. Ici, chaque vecteur représente l'orientation du champ électrique à un instant t et celui-ci évolue dans le temps. C'est la polarisation utilisée pour la transmission des images par les satellites NOAA.

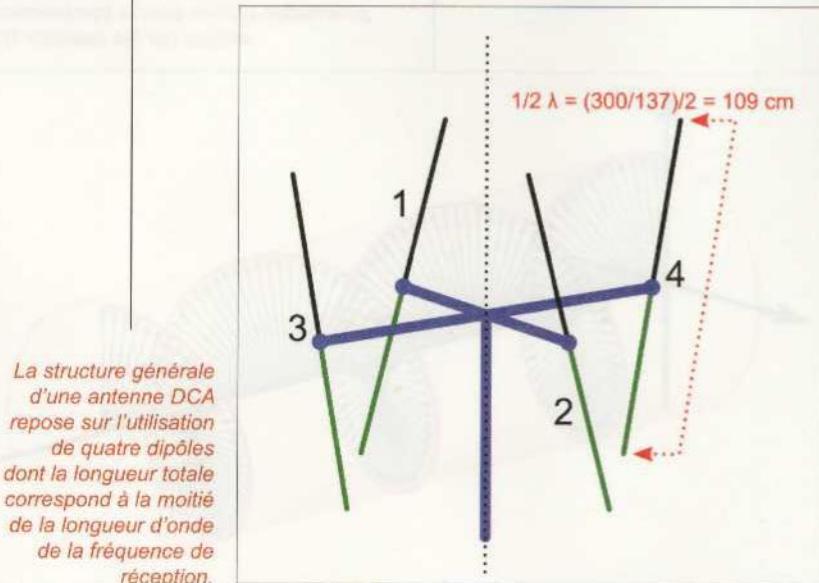




2. TYPES D'ANTENNES POUR LA RÉCEPTION APT

En recherchant des modèles d'antennes dédiées à la réception satellite en APT, quatre types remontent généralement dans les résultats :

- Le simple dipôle en V constitué de deux brins ouvert à un angle de 120°. C'est l'antenne la plus simple à réaliser, mais celle nécessitant le plus de précision dans la construction, la plus sensible à l'environnement et une orientation dépendante des trajectoires des satellites à écouter. Sa structure est simple puisqu'il s'agit tout bonnement de deux tiges d'aluminium de 53,4 cm (1/4 d'onde) disposées horizontalement, mais elle n'est pas adaptée à la réception de signaux polarisés circulairement.
- La *crossed-dipole antenna*, également appelée antenne *turnstile*, utilise comme son nom l'indique, deux dipôles 1/2 onde croisés à 90° et disposés à l'horizontale. L'ensemble forme une croix placée au bout d'un mât et une croix supplémentaire, non connectée au récepteur, est souvent ajoutée pour servir de réflecteur. L'idée est ici d'avoir deux dipôles croisés reliés entre eux de manière à obtenir un déphasage de 1/4 d'onde. Grossièrement, l'effet recherché consiste à superposer l'onde captée par un dipôle sur celle de l'autre, de façon parfaitement alignée. Plus complexe à construire, mais plus efficace, cette antenne n'est toutefois pas idéale pour la réception



dans le sens où sa sensibilité est réduite selon la position du satellite dans le ciel.

• L'antenne QFH pour *Quadrifilar Helix Antenna* est tantôt appelée simplement « antenne quadrafilaire ». Elle se compose de deux boucles ou *loops* verticales perpendiculaires l'une à l'autre et disposées en hélice sur un quart de tour. Le matériel nécessaire n'est pas très coûteux puisqu'en principe un tube de PVC et du fil de cuivre électrique de gros diamètre peuvent suffire. De plus, elle permet une réception optimale des signaux émis avec une polarisation circulaire et ce sur toute la trajectoire du satellite de l'horizon au zénith... si elle est réalisée correctement. Cette condition est sans le moindre doute le point faible de cette antenne puisque la moindre erreur de mesure ou de construction peut ruiner la réception.

• Enfin, nous avons la *Double Cross dipole Antenna* (l'antenne à double dipôles croisés) ou DCA, qui reprend le concept de l'antenne *turnstile* et qui... le double. Cette antenne est très tolérante aux erreurs de mesure et d'ajustement, possède une bonne sensibilité dans toutes les directions et, même si le matériel nécessaire à la construction est plus conséquent, la fabrication est relativement simple puisque tout est droit.

Une quatrième solution existe et ouvre la porte à toute une collection d'autres modèles : l'antenne directionnelle. Comme son

nom l'indique, celle-ci nécessite de pointer le satellite et donc de suivre en temps réel sa trajectoire dans le ciel. Ceci se fait généralement avec des *trackers* motorisés et certains logiciels comme Gpredict possèdent un support pour le pilotage de tels *rotors*. Ce n'est généralement pas l'antenne avec laquelle débuter, car plus il y a de paramètres à régler, plus il est difficile d'isoler un problème.

Vous l'avez compris, le modèle choisi pour les expérimentations sera donc la DCA et le matériel utilisé sera :

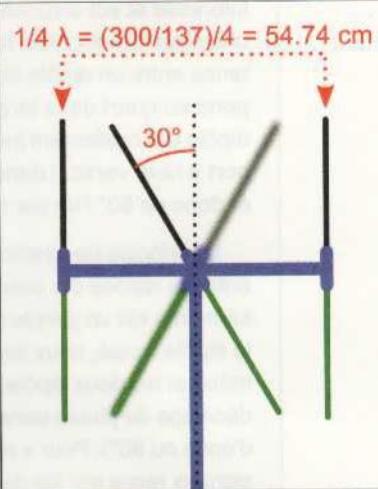
- 2 mètres de tube PVC 32 mm généralement utilisés pour les travaux de plomberie et d'évacuation d'eau ;
- 4 raccords en « T », en prenant bien soin de vérifier qu'ils forment effectivement un angle droit. Certains modèles qu'on trouve en magasin ne sont pas vraiment à 90° mais, heureusement, ceci est clairement visible en magasin ;
- un raccord en croix ;
- 8 bouchons pour canalisation PVC de 32 mm ;
- environ 3m de câble coax RG58/U 50 ohms : j'ai utilisé du vieux câble réseau Ethernet 10base2 présentant l'avantage d'avoir la bonne impédance (50 ohms) et de posséder un connecteur BNC déjà serti. Il m'a donc suffi de trouver un adaptateur BNC/SMA pour connecter l'antenne au récepteur RTL-SDR ;
- 1 mètre de baguette en bois 6 mm. Ceci est un point extrêmement variable puisque

l'objet de ce composant est de maintenir les deux pôles ensemble, de façon rectiligne, sans pour autant avoir un contact électrique ;

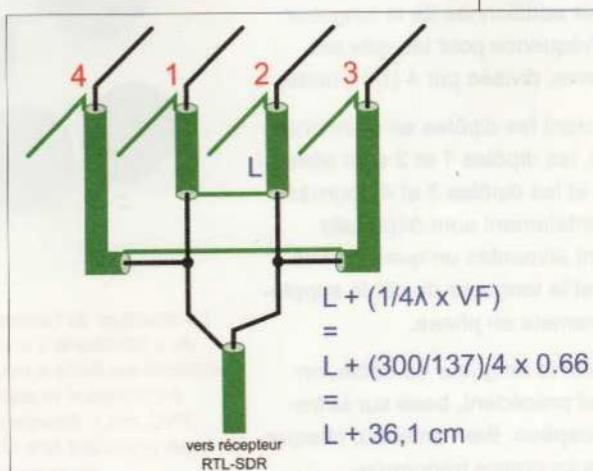
- 4 fois 1 mètre de tube aluminium 8 mm.

Comprenez bien que je vous donne ici la liste et la méthode de construction que j'ai utilisée, mais que vous êtes libre d'avoir votre propre approche, en fonction de vos préférences, avec les matériaux de votre choix et vos affinités avec ces derniers. L'architecture de l'antenne, les dimensions, les écartements, l'orientation des dipôles et la taille de deux des câbles coaxiaux sont ce qui importe réellement. Le reste n'est que la structure qui soutient l'ensemble. Certains amateurs ont construit ce type d'antenne avec de simples bouts de bois de récupération et quelques vis. J'ai préféré utiliser des tubes PVC donnant un ensemble fini plus propre, ajustable et durable.

La construction de l'antenne DCA se divise en deux parties. Nous avons la structure portante et les dipôles avec leurs connexions. Chaque dipôle a une taille totale correspondant à une demi-longueur d'onde de la fréquence qui nous



Chaque dipôle est incliné de 30° par rapport à l'axe vertical et donc de 60° par rapport au dipôle opposé. De plus, chacun d'eux est espacé de celui opposé d'une distance correspondant au quart de la longueur d'onde, de façon à ce que la paire ainsi formée soit alimentée en phase et la seconde paire, tournée à 90° sur le plan horizontal, est alimentée avec un déphasage d'1/4 d'onde.



L'interconnexion des dipôles est l'élément clé de l'antenne DCA. Les dipôles opposés sont connectés en série et ces deux paires de dipôles sont connectées en parallèle. Pour « aligner » les phases du signal, on ajoute à la longueur arbitrairement choisie du câble de la première paire de dipôles un segment supplémentaire à la seconde paire, d'une taille correspondant au 1/4 de la longueur d'onde multipliée par le velocity factor du type de câble utilisé.



intéresse et est disposé verticalement sur une structure en croix horizontale. La distance entre un dipôle et celui opposé correspond au quart de la longueur d'onde. Chaque dipôle est également incliné de 30° par rapport à l'axe vertical dans le sens anti-horaire et donc de 60° l'un par rapport à l'autre.

Le principe de fonctionnement de cette antenne repose sur celui du dipôle croisé qui lui-même est un simple dipôle doublé. Avec le dipôle croisé, nous avons une croix formée par les deux dipôles, ce qui introduit un décalage de phase dans le signal reçu ($d/4$ d'onde ou 90°). Pour « réaligner » les deux signaux reçus par les deux dipôles, on utilise un morceau de câble coaxial dont le but est de « retarder » l'un des deux signaux de façon à ce qu'il s'aligne sur l'autre. Dans le cas de l'antenne DCA le principe est le même, mais tout est en double. Nous avons donc deux dipôles en série, avec une longueur de coax supplémentaire, connectée, en parallèle aux deux autres et le tout complètement le câble coax qui sera relié au récepteur RTL-SDR.

La clé du fonctionnement de cette antenne repose donc sur la différence de longueur de câbles reliant une paire dipôle et le point de connexion globale. Deux longueurs de câble sont choisies arbitrairement, en fonction de la structure porteuse de l'antenne, et les deux autres ont une longueur identique, mais additionnée de la longueur d'onde de la fréquence pour laquelle est conçue l'antenne, divisée par 4 ($1/4$ d'onde).

En numérotant les dipôles en quinconce (cf. schémas), les dipôles 1 et 2 sont alimentés en phase et les dipôles 3 et 4, tournés de 90° horizontalement sont déphasés de 90°, ils sont alimentés un quart d'onde « plus tard » et la longueur de câble supplémentaire les remets en phase.

Pour calculer la longueur du câble, on utilise le calcul précédent, basé sur la fréquence de réception. Bien entendu, chaque satellite utilise sa propre fréquence :

- NOAA-15 : 137,620 MHz. $300/137,620 = 2,179915$ mètres, $1/2$ onde = 1,089 mètre $1/4$ d'onde = 54,49 cm
- NOAA 18 : 137,9125 MHz. $1/4$ d'onde = 54,38 cm
- NOAA 19 : 137,1 MHz. $1/4$ d'onde = 54,70 cm

Ces longueurs, cependant, sont celles concernant la propagation d'une onde dans le vide, celle de la lumière. Or nous utilisons un câble dont le matériau conducteur réduit cette vitesse. Chaque type de câbles possède ses caractéristiques, dont une est très importante dans cette réalisation : le *velocity factor* (VF), correspondant à un pourcentage de la vitesse de la lumière. Un câble avec un VF de 66% permet la transmission d'un signal à 66% de la vitesse de la lumière ©. Il a donc besoin d'être moins long pour obtenir le « retardement » qui nous est nécessaire. Avec un quart de la longueur d'onde de 54,7 cm (137,1 Mhz) dans le vide, nous arrivons donc à environ 36 cm de câble à ajouter pour deux des dipôles. Les différences de fréquence entre l'un et l'autre satellite, ramenées à l'échelle du quart d'onde, sont minimes. On peut donc, tout simplement, faire tous les calculs pour environ 137 Mhz sans réellement impacter l'efficacité de l'antenne.

3. CONSTRUCTION



1

La structure de l'antenne est entièrement composée d'éléments de « plomberie » en PVC au diamètre de 32 mm. Ce type de matériel est facile à trouver en magasin de bricolage, relativement économique et aisément à travailler (scie, lime, perceuse, colle PVC, etc.). Attention cependant, les pièces en « T » doivent impérativement être en angle droit, ce qui n'est étrangement pas toujours le cas dans certains magasins.



2

Chaque base de dipôle est un assemblage d'un « T » avec deux sections de tube PVC et des bouchons. Les dimensions de ces éléments n'ont pas réellement d'importance puisqu'il ne s'agit que de fournir un support à chaque dipôle et de permettre une orientation ajustable.



4

Chaque mètre de tube en aluminium de 8 mm est coupé en deux et glissé sur une baguette en bois de 6 mm, poncée pour s'adapter parfaitement. La soudure sur aluminium, en particulier avec une telle masse de métal, est un véritable cauchemar. C'est pourquoi la solution des vis sauvagement insérées dans les tubes a été préférée ici pour connecter le câble coax.



6

La réalisation du support en croix présente comme seule difficulté l'estimation juste de la longueur des segments de tubes PVC. Les dipôles opposés doivent être séparés d'environ 54 cm, une fois le tout assemblé. La technique la plus simple pour obtenir ce résultat est de simplement mesurer chaque pièce, couper en gardant une marge et de procéder à un assemblage de test suivi d'une mesure effective. Dans les grandes lignes, on tâtonne et on scie jusqu'à arriver aux bonnes dimensions.



3

Les bouchons sont percés avec une mèche de 8 mm afin de laisser passer les tubes d'aluminium.

Une certaine précision est nécessaire afin de garantir que le plan du dipôle restera parallèle à l'axe vertical et donc que chaque brin sera effectivement uniformément parallèle à ceux du dipôle opposé.



5

Chaque dipôle, équipé de son morceau de coax, est inséré dans l'assemblage en « T ». À ce stade, il est important de repérer quel brin du dipôle est connecté à l'âme du câble et quel brin est relié au blindage. Un coup de feutre indélébile ou un bout d'adhésif de masquage fera parfaitement l'affaire pour cela.



7

Le raccord en croix est percé au moyen d'une scie cloche de 32 mm afin de permettre à la fois l'assemblage avec la base verticale et le passage des câbles. Dès ce moment et afin de limiter les erreurs, chaque pièce est marquée afin de lui faire correspondre le numéro de chaque dipôle du schéma.



8

Un premier essai de montage permet de confirmer que l'ensemble de la structure est fiable tout en restant ajustable. Il ne reste plus qu'à construire les trois ensembles pour les dipôles restants.



9

Voici la partie supérieure totalement terminée, mais rien n'est encore sécurisé (collé). L'ensemble des câbles reliés aux dipôles, qui ressortent au centre du support en croix percé, est dénudé et prêt pour la phase la plus critique : la connexion et la soudure des câbles. Les deux segments les plus courts et les deux segments les plus longs sont soudés entre eux en série par le blindage. Les deux paires sont ensuite soudées en parallèle, puis au coax qui fera la liaison avec le récepteur RTL-SDR.



10

L'antenne entièrement assemblée est presque prête pour son installation. Pour l'heure, tout tient par simple friction et avec l'aide d'un peu de scotch de masquage. Notez le marquage des dipôles : le brin de chaque dipôle qui est connecté à l'âme de son câble se trouve en haut et celui connecté à la masse, en bas.



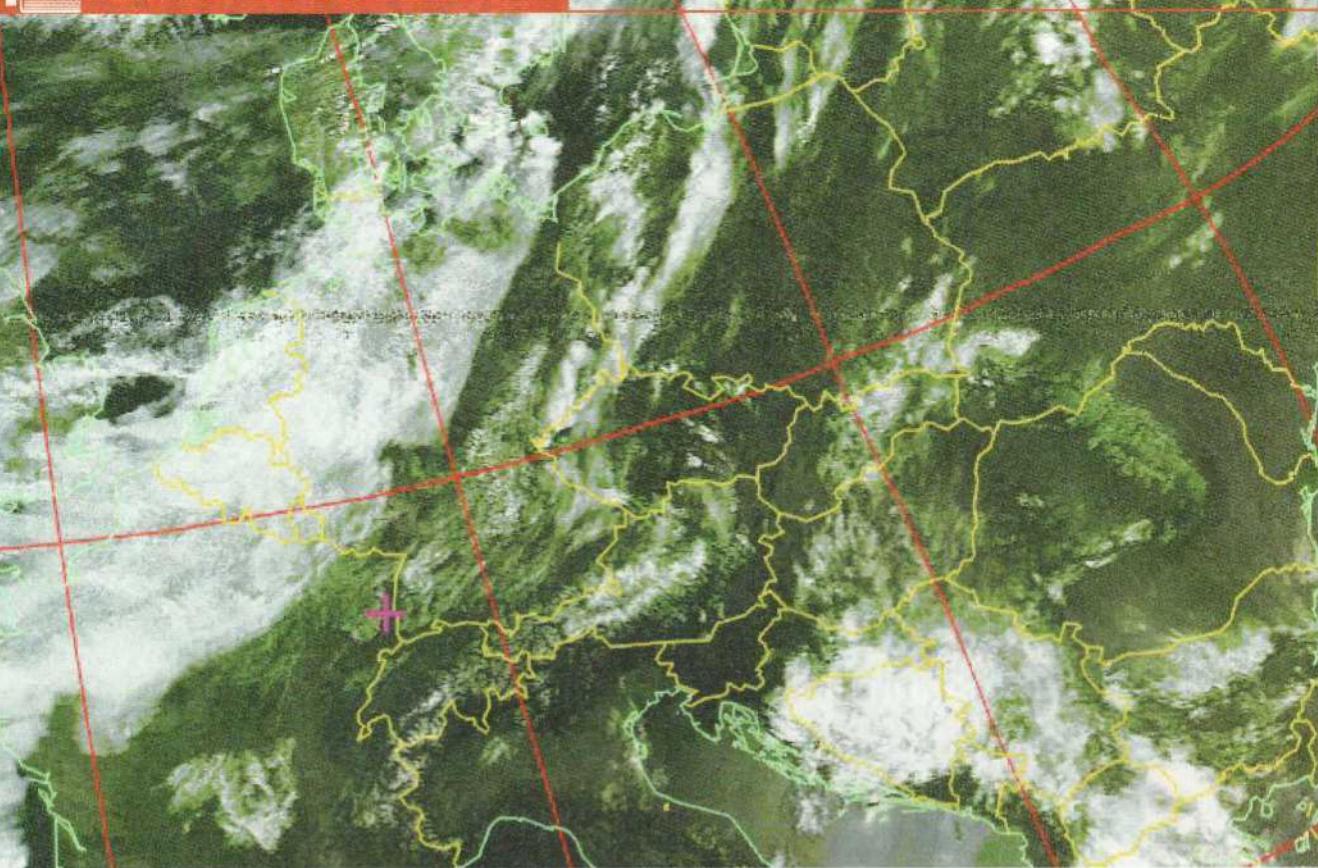
11

Afin d'incliner correctement les dipôles de 30° par rapport à la verticale, et ce de façon reproductible, un assemblage de rapporteur, de niveau et de règlette en bois forme un outil fort pratique. Ici, je suis passé dans la seconde phase de l'assemblage en passant du scotch au pistolet à colle afin d'ajouter un peu plus de rigidité à l'ensemble. Seuls les « T » sont encore mobiles afin de procéder au réglage après transport et l'antenne en situation. La colle PVC ne sera utilisée qu'une fois les premiers essais effectués et les premiers résultats acceptables validés. **DB**



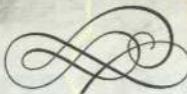
EN COUVERTURE

RÉCEPTION D'IMAGES



RÉCEPTION DE VOS PREMIÈRES IMAGES SATELLITE

Denis Bodor



...et des très nombreuses images qui suivront, bien entendu. Vous connaissez la théorie, vous avez votre antenne, il est maintenant temps de faire vos premiers essais et de vous mettre à l'écoute du signal qui, j'en suis sûr, deviendra rapidement pour vous une mélodieuse et douce musique. Car la réception est d'abord le plaisir des oreilles avant de devenir celui des yeux...

Avant toutes choses, il est excessivement important que vous gardiez l'espoir et n'abandonniez pas après des réceptions infructueuses, furent-elles par dizaines. Ma toute première réception, à peine audible, s'est traduite par une image absolument catastrophique, relevant plus de l'échographie d'un cachalot échoué une nuit de tempête que de quoi que ce soit de vaguement ressemblant à une image prise depuis l'espace. Comme vous devez le penser présentement, même inconsciemment, j'avais dans l'espoir de simplement brancher le tout, attendre que le prochain satellite passe, enregistrer et obtenir une magnifique image comme à la TV. Mais cet exercice demande de la patience, car bon nombre de facteurs entrent en jeu pour de bons résultats : emplacement de l'antenne, conditions météo, environnement, trajectoire du satellite, réglage de la réception, logiciels utilisés, qualité du récepteur, bruit environnant...

Mon conseil tient en un seul mot : persévérez.

1. LOGICIEL ET ENREGISTREMENT

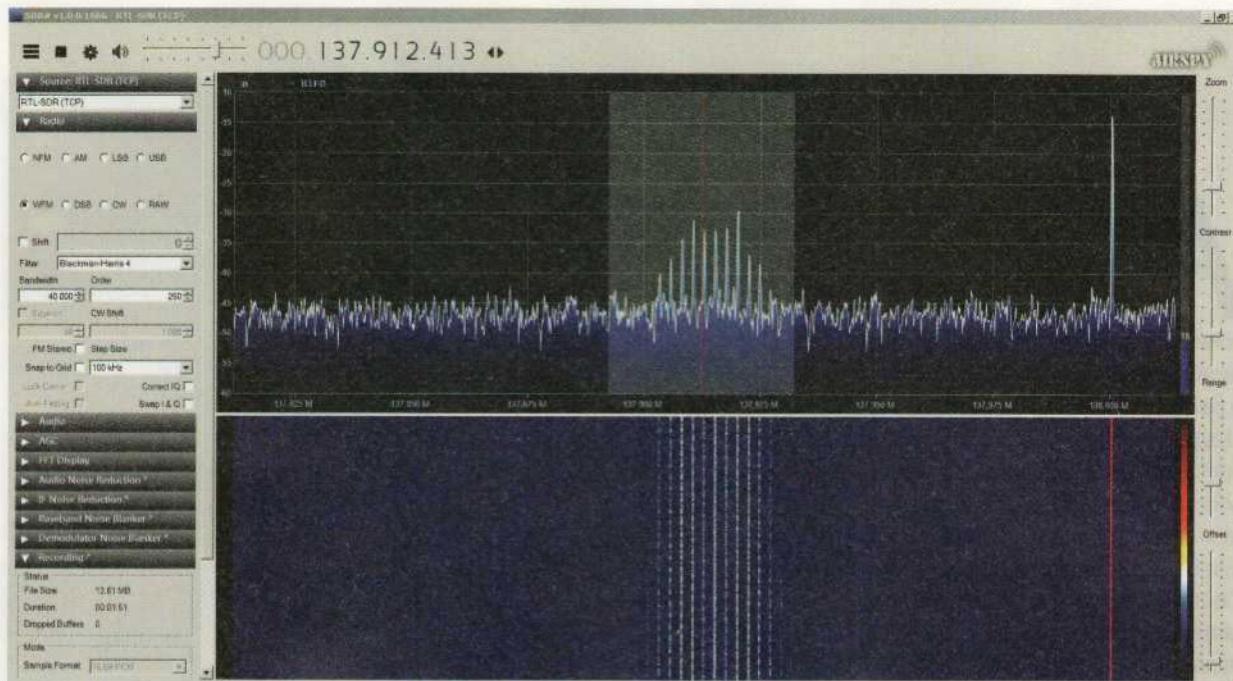
Qu'il s'agisse de la réception de transmissions provenant d'un satellite ou de n'importe quel autre type d'émission radio, il est nécessaire d'utiliser une application adaptée. Comme expliqué précédemment, le récepteur SDR se contente de se caler sur une fréquence et d'échantillonner les signaux reçus pour les numériser (quantifier) avant de les transmettre au PC ou à la Raspberry Pi via USB. C'est donc à la charge d'un logiciel de recevoir ces données et d'en faire quelque chose d'utilisable.



Les satellites NOAA transmettant en APT utilisent chacun une fréquence spécifique : NOAA-15 137,620 MHz, NOAA-18 137,9125 MHz et NOAA-19 137,100 MHz. Le signal est modulé en fréquence (FM) comme pour les radios commerciales (bande FM 87,5 à 108 MHz), mais ce n'est pas du son ou de la musique qui sont ainsi modulés, mais les données d'une image. En d'autres termes, les images sont modulées en amplitude et ce signal audible (-2400 Hz) est ensuite modulé en fréquence puis transmis.

Démoduler le signal FM revient donc peu ou prou à faire exactement la même chose qu'un poste radio (ça existe encore ?) et c'est aussi la première chose que l'on teste pour vérifier qu'un récepteur SDR et son application associée fonctionnent correctement. Vous vous calez sur une station locale et vous vérifiez que vous pouvez effectivement entendre l'émission. C'est donc quelque chose à la portée de n'importe quelle application de radio logicielle, relativement facile à utiliser et il n'est pas

Voici ma toute première réception décodable d'une image provenant de l'espace. Naïvement, j'espérai quelque chose de bien plus impressionnant, mais, dans le même temps, c'était déjà incroyable d'avoir sous les yeux une image ressemblant vaguement à des masses nuageuses photographiées depuis l'espace.



SDR# est l'application la plus populaire pour la réception SDR sous Windows. On voit ici l'outil en pleine réception d'une transmission du satellite NOAA-18, n'utilisant pas un récepteur RTL-SDR USB local, mais celui connecté sur une machine en réseau, trois étages plus haut, dans le grenier.

nécessaire de faire usage de choses plus complexes comme un diagramme GNU Radio (même si cela est parfaitement possible).

Il existe plusieurs approches possibles et certaines applications SDR possèdent même parfois des greffons permettant de décoder les images en temps réel. En ce qui me concerne, je préfère largement enregistrer le signal démodulé sous la forme d'un fichier audio WAV, pour ensuite le traiter et décoder les données qu'il contient. Ceci permet de se constituer une belle collection d'enregistrements, éventuellement d'appliquer des filtres et d'analyser les problèmes ponctuels. L'avantage est également de pouvoir utiliser n'importe quelle application SDR (ou outil en ligne de commandes) sans qu'il soit nécessaire d'être lié à l'une d'entre elles ou à un système d'exploitation spécifique.

Il existe une myriade d'applications SDR et certaines d'entre elles ont vu le jour récemment (comme *SDRRange!*). Toutes sont, sans le moindre doute, utilisables pour ce type d'application puisque la démodulation d'un signal FM et l'enregistrement de l'audio résultant est une fonction de base. Je m'en tiendrai cependant aux deux applications les plus populaires, une sous Windows et une sous GNU/Linux (PC ou Raspberry Pi). Que vous soyez utilisateur de l'un ou l'autre système, lisez les explications complètes pour les deux applications puisque j'aborde des concepts communs.

1.1 Windows

L'application Windows sera ici SDR# qui est initialement développé pour servir de support logiciel au récepteur SDR Airspy, mais est parfaitement compatible avec un récepteur RTL-SDR quel qu'il soit. C'est une application très mature et, si vous avez suivi les instructions dans l'introduction, à propos de Zadig, il vous suffira de télécharger SDR# en cliquant sur *Windows SDR Software Package* sur la page de téléchargement (<https://airspy.com/download/>).

Le fichier ZIP obtenu, **sdrsharp-x86.zip**, se désarchive tout simplement dans le dossier de votre choix et contient, entre autres choses, le programme **SDRSharp.exe**. Pensez également à lancer **install-rtlsdr.bat** dans ce dossier pour installer la DLL permettant l'utilisation d'un récepteur RTL_SDR et télécharger automatiquement Zadig (si vous ne l'avez pas déjà utilisé). L'interface de SDR#, bien que très

complète, est relativement simple pour notre usage. Vous avez, tout en haut, la fréquence centrale sur laquelle caler le récepteur et, juste dessous, la visualisation des fréquences (FFT) et l'affichage *waterfall* présentant l'intensité des signaux au fil du temps.

Sur la gauche, une succession de boîtes dépliables donnent accès aux différentes fonctionnalités de l'application. Dans *Source*, choisissez *RTL-SDR (USB)* puis cliquez sur la roue dentée en haut à gauche. Là vous devez retrouver votre récepteur RTL-SDR avec un nom comme « Generic RTL2832U OEM » et pourrez régler le gain automatique (case « *RTL AGC* »).

Mettez en route la réception en cliquant sur le bouton *play* triangulaire en haut à gauche et amusez-vous avec les fréquences. Vous devriez être en mesure de recevoir plusieurs stations radio FM sans le moindre problème. La boîte *Radio*, juste sous *Source*, vous permet de régler les options de démodulation. Ce qui nous intéresse ici c'est la FM et en particulier ce que les américains appellent la « FM bandes larges » ou WFM, qui est le même type de modulation que pour les stations FM commerciales. Une différence notable cependant est la largeur de bande ou, en d'autres termes, la plage de fréquences dans laquelle la modulation fera dévier le signal. Pour la radio FM, une largeur de bande de 100 khz permet de moduler l'audio mono, stéréo et les données RDS. Dans le cas des transmissions APT, cette largeur de bande est théoriquement de 38 Khz.

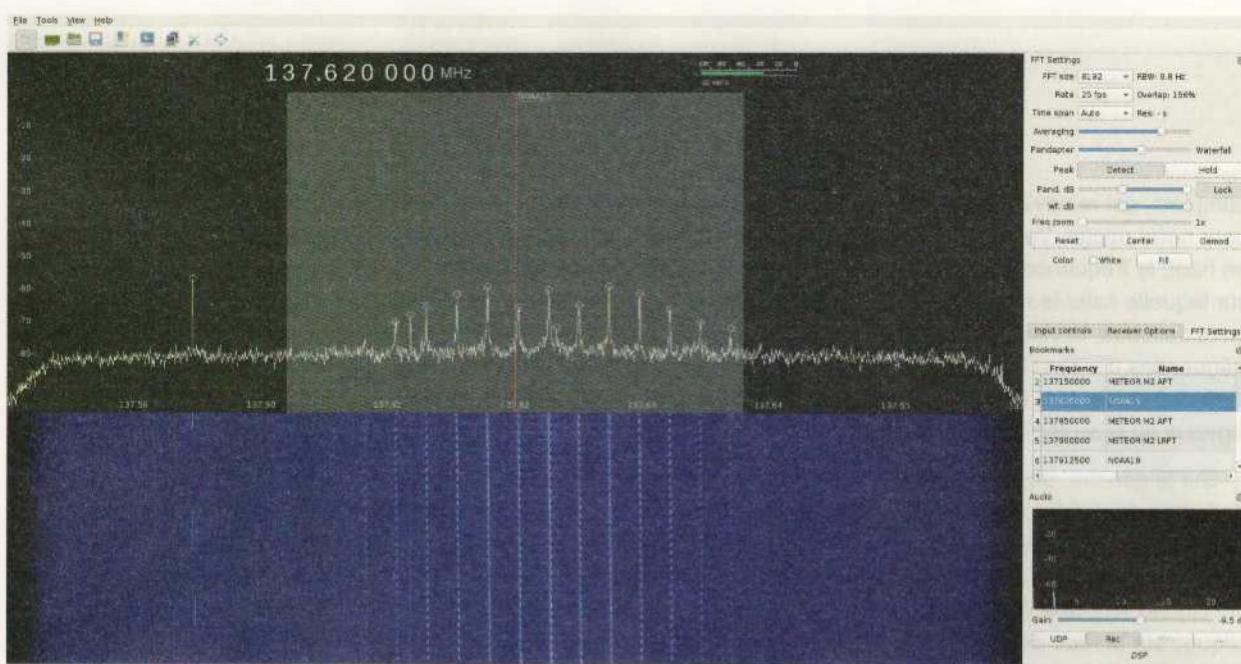
Vous devrez donc régler ce paramètre dans le champ *Bandwidth* ou cliquer et déplacer le bord de la zone grise verticale dans l'affichage FFT. Ajustez une bande un peu plus large sur 38 Khz est généralement une bonne idée, puisqu'en raison de l'effet Doppler, la fréquence de réception va diminuer au fil du passage du satellite. 40 Khz est généralement une bonne valeur.

Il ne vous reste plus qu'à attendre qu'un satellite NOAA passe à votre portée (en vous aidant de n2yo.com ou de Gpredict) et vous devrez voir apparaître et entendre un signal caractéristique : un sifflement intermittent doublé d'un son de métronome régulier. Pour conserver une trace de ce signal, dépliez la boîte *Recording* sur la gauche et assurez-vous que la case *Audio* soit cochée et *Baseband* décochée. Nous voulons uniquement enregistrer la transmission démodulée et non l'ensemble du signal. Dès que vous êtes prêt, cliquez sur *Record* pour enregistrer (puis « *stop* » pour arrêter). Les enregistrements WAV sont automatiquement placés dans le répertoire où se trouve l'application SDR# et nommés *SDRSharp_* suivis de la date, l'heure (GMT) et la fréquence utilisée.

1.2 GNU/Linux et Raspberry Pi (2 & 3)

Avec ce système, et en dehors de l'utilisation de GNU Radio et d'outils en ligne de commandes, l'application de référence pour la SDR est Gqrx (mais cela risque de changer, car SDRangel commence à devenir très populaire). Son installation se fera très simplement, que ce soit avec un PC sous Debian GNU/Linux ou avec une Raspberry Pi sous Raspbian, à l'aide d'un simple **sudo apt-get install gqrx-sdr**. Notez qu'une version plus récente et spécialement dédiée aux Raspberry Pi 2 et 3 est téléchargeable depuis la page <http://gqrx.dk/download/gqrx-sdr-for-the-raspberry-pi>.

Gqrx, se présente comme toutes les applications de ce type et ressemble donc fortement à SDR#. Au premier lancement, en revanche, Gqrx vous demandera de choisir un périphérique SDR à utiliser et vous devriez reconnaître assez aisément votre matériel s'il est connecté au système. Une fois cet élément validé, vous retrouverez l'interface standard avec la fréquence centrale en haut, la visualisation du spectre de fréquences (FFT) en dessous et le *waterfall* ensuite.



Voici Gqrx, sous GNU/Linux, en train de recevoir le signal émis par NOAA-15. Remarquez la largeur du filtre et la forme sur le spectre de fréquences, décalé vers la droite. Au fil de la transmission, le signal va glisser à gauche en raison de l'effet Doppler, mais continuera à être démodulé correctement, car toujours dans la bande de 40 KHz.

Sur la droite, cette partie est divisée en deux par défaut (il est possible de déplacer chaque élément de l'interface) avec en haut tout ce qui concerne la réception en tant que telle et en bas l'aspect audio. Trois onglets vous permettent de paramétrier la réception :

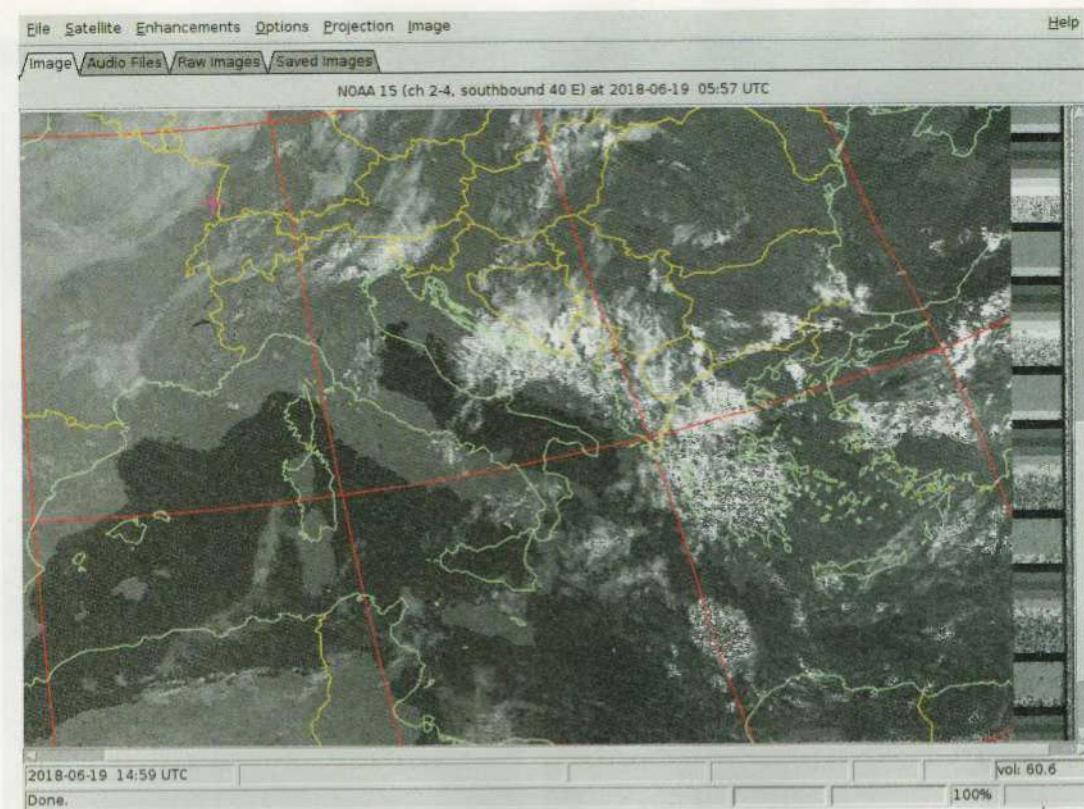
- **Input controls** : ici se trouvent les éléments de configuration du récepteur avec, dans le cas qui nous occupe, le réglage du gain manuel et/ou automatique (option **Hardware AGC**) ;
 - **Receiver Options** : permet de régler les options de démodulation avec, pour APT/NOAA, **Mode** sur **WFM (Mono)** et **Filter width** sur **User** avec un ajustement directement en cliquant et glissant le bord de la zone grise verticale sur l'affichage FFT ;
 - **FFT settings** : ces paramètres n'impactent pas directement la réception, mais sont utilisés pour la visualisation des signaux. **Panadapter** par exemple règle la séparation entre la visualisation du spectre de fréquences et le **waterfall**.

Les deux éléments clés de la réception APT sont une démodulation FM et une largeur de filtre (ou de bande) d'environ 40 Khz. Comme avec SDR#, cliquez sur le bouton *play* en haut à gauche et commencez par rechercher les stations radio FM sur la bande 87,5 à 108 MHz. Si tout fonctionne correctement, réglez ensuite la fréquence vers 137 Mhz, ajustez la largeur du filtre et attendez le passage d'un satellite.

En bas à droite se trouve la partie contrôlant le signal démodulé (l'audio) et vous verrez, tout en bas, un groupe de quatre boutons : *UDP* pour envoyer le signal à une machine en réseau, *Rec* pour enregistrer un fichier WAV, *Play* pour relire un enregistrement et ... permettre d'accéder à la configuration de ces éléments. Là, sur l'onglet *Recording*, vous pourrez spécifier un chemin vers un répertoire où seront stockés les enregistrements (dans le cas contraire, les fichiers trouveront place dans votre répertoire personnel).

Une fois ces petits réglages effectués, vous pouvez partir à la chasse aux satellites et débuter la réception, puis enregistrer les signaux démodulés. Comme avec SDR#, les fichiers seront nommés sur la base du nom de l'application, **gqrx_**, suivi de la date, de l'heure (GMT) et de la fréquence de réception. Chaque enregistrement démarré et stoppé donnera un nouveau fichier.

Que ce soit avec SDR# ou Gqrx, veillez à ajuster parfaitement les



WXtoImg est l'outil multiplateforme de prédilection pour le décodage APT. On voit ici le résultat d'une de mes meilleures réceptions et on reconnaît sans peine la « botte » de l'Italie, la Corse et la Sardaigne. Seule retouche nécessaire : réaligner la couche cartographie.

réglages avant l'enregistrement, en prévoyant du mieux possible la dérive en fréquence en raison de l'effet Doppler. Selon le logiciel utilisé pour décoder le signal et obtenir une image, cette dérive n'est pas un gros problème, du moment que l'ensemble de la transmission reste démodulée. Prévoir un filtre un peu plus large que les 38 KHz théorique vous évite donc de devoir ajuster la fréquence de réception au fil du passage du satellite. Triturer les réglages en cours d'enregistrement, en revanche, provoque généralement une confusion de l'outil de décodage et donc un résultat plus ou moins catastrophique.

Ne vous inquiétez pas, je sais qu'une réception maladroite peut être très frustrante (tout comme un oubli d'enregistrement (vécu)), mais l'un des trois satellites NOAA repassera par chez vous et ce ne sera que partie remise. Je l'ai dit et le répète : persévérez !

2. TRAITEMENT AUDIO ET DÉCODAGE DES IMAGES

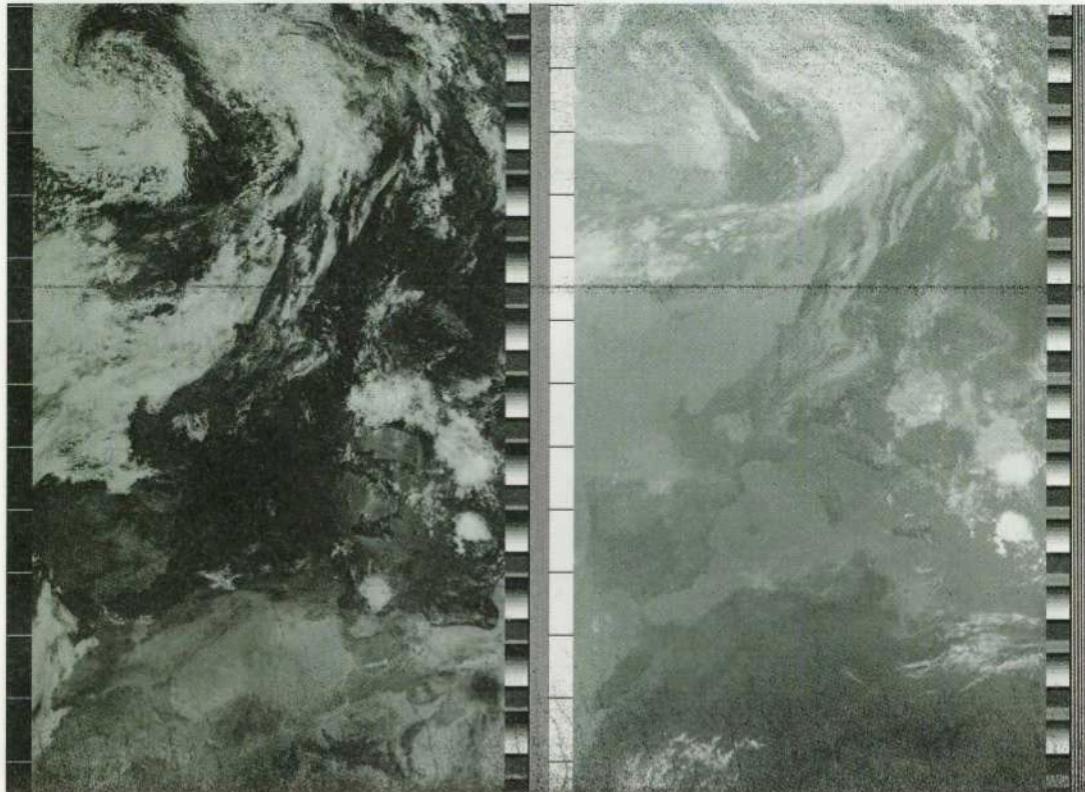
À présent que vous avez capturé votre première transmission d'un satellite NOAA sous la forme d'un fichier audio, il est temps de vérifier si les données sont bien là. Il existe plusieurs outils permettant de décoder les informations et d'obtenir une image et la très grande majorité d'entre eux sont sous Windows et sont payants. Quelques projets open source existent, mais, force est de constater qu'ils n'offrent de loin pas le même niveau de fonctionnalités.

Je vais sans doute me faire quelques ennemis en écrivant cela, mais, ne vous attendez pas à une ergonomie et un design moderne concernant ces applications. À l'instar des pages web sur le sujet, les applications de décodage APT et WeFAX sont très « teintées années 2000 » (sinon 90), tant sur l'aspect interface que sur le côté très shareware/freeware de leur distribution.

L'une de ces applications, et certainement la plus connue, est WXtoImg de Craig Anderson, diffusé via le site <http://www.wxtoimg.com/>. Ce logiciel est disponible gratuitement en version standard (2.10.11) pour Windows, Mac OS X (G4, G5, et Intel), Debian



Les satellites NOAA transmettent deux images, une dans l'infrarouge très proche et l'autre dans l'infrarouge thermique. Les bandes sur le côté des images sont les données télémétriques du satellite. Ceci est le résultat d'une excellente et longue réception.



GNU/Linux, Red Hat Linux, ainsi d'une version générique pour GNU/Linux (32 et 64-bits). Il existe également une version pour GNU/Linux sur plateforme ARM en version expérimentale (2.11.2).

Une autre solution que j'ai testée est APT3000, un applicatif entièrement en JavaScript accessible en ligne (via <http://jthatch.com/APT3000/APT3000.html>), dont les sources sont disponibles sur GitHub (<https://github.com/ThatcherC/APT3000>). Fonctionnelle, cette solution peut être utilisée en ligne ou installée sur un serveur web et fait le travail de base, mais sans « fioritures » comme l'ajout d'une couche cartographique, la fusion des deux canaux (visible et IR), le rendu en fausses couleurs, l'amélioration par analyse multispectrale (MSA), etc.

Mais avant de vous lancer dans le décodage, il est nécessaire de traiter vos fichiers audios. Wxtolmg attend des fichiers WAV PCM mono 16 bits avec une fréquence d'échantillonnage de 11025 Hz. SDR# produit des fichiers WAV stéréo 32000 Hz et Gqrs enregistre des fichiers WAV mono en 48000 Hz. Un passage par un logiciel de traitement comme Audacity est donc obligatoire pour éventuellement fusionner les pistes stéréo en mono et rééchantillonner le tout avant d'exporter dans un nouveau fichier. En bas à gauche de la fenêtre Audacity vous avez une mention *Projet à*. Réglez cela sur 11025 puis passez par *Piste* et *Rééchantillonner* pour convertir les données.

Une autre solution, sous GNU/Linux, consiste à installer SoX (`apt-get install sox`) puis de simplement se plier d'un `sox fichierIN.wav -c 1 -r 11025 fichierOUT.wav` pour obtenir une version mono à 11025 Hz.

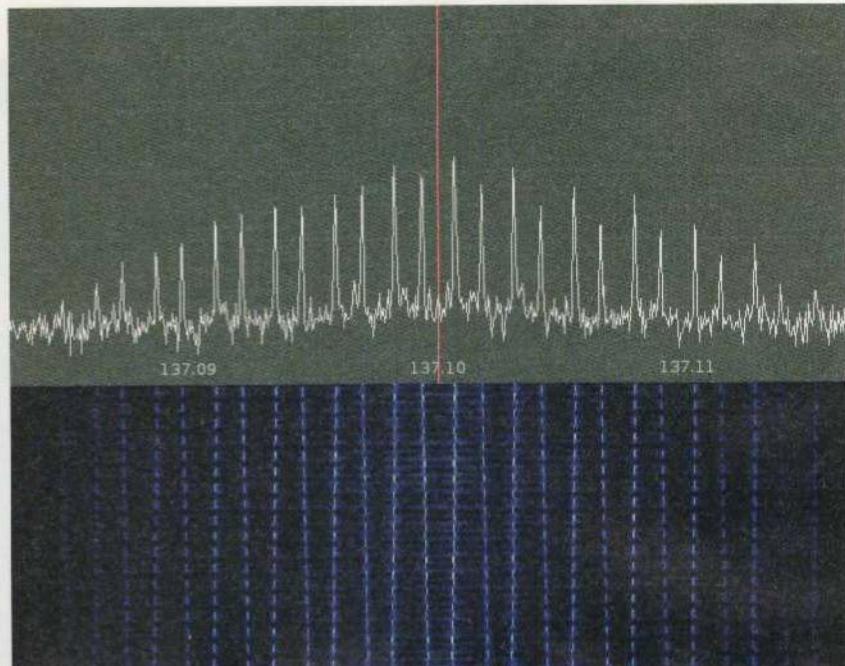
Lors du premier lancement de Wxtolmg (`wxtoimg -G` pour avoir l'interface graphique avec GNU/Linux), l'application vous demandera de spécifier les coordonnées géographiques de l'observateur (vous) en fournissant latitude et longitude. Ne passez pas cette étape, car dans le cas contraire vous ferez l'impasse sur les fonctionnalités les plus intéressantes de Wxtolmg.

Immédiatement après cette configuration, il est impératif de passer par le menu *File* puis *Update Kepler* permettant de mettre à jour les orbites des différents satellites connus de Wxtolmg (et le faire régulièrement). L'une des fonctions

les plus utiles de cette application est le fait d'ajouter des éléments dans les images décodées comme les côtes et les frontières des pays, permettant par la même occasion de colorer les terres pour les rendre plus visibles. Ceci n'est possible que si WXtoImg a une idée de votre emplacement géographique et de celui du satellite utilisé.

Ouvrez ensuite votre fichier WAV rééchantillonné en passant par *File* et *Open Audio File* et laissez WXtoImg faire son travail. Si tout se passe correctement et que la réception a été bonne, vous devez non seulement voir apparaître une image intelligible, mais également la surcouche géographique. Si celle-ci n'est pas alignée correctement, vous pouvez passer par *Image* et *Move Map Overlay*. Cliquez ensuite sur un endroit du tracé géographique, maintenez le bouton de la souris enfoncé et glissez jusqu'au même endroit reconnaissable sur la photo décodée. Après un certain temps de calcul, le tout sera réaligné. Enfin, explorez les options de visualisation via *Enhancements* pour obtenir quelques versions optimisées de l'image.

Les satellites NOAA encore en service transmettent deux images, accompagnées de données télemétriques. Les deux images sont transmises côté à côté, l'une en infrarouge proche (quasi visible), c'est le canal A, et l'autre en IR thermique, le canal B. Le satellite ne fait pas un cliché qui est transmis, mais procède de façon continue, tranche par tranche. Plus longue est la réception de qualité, et plus haute sera l'image résultante.



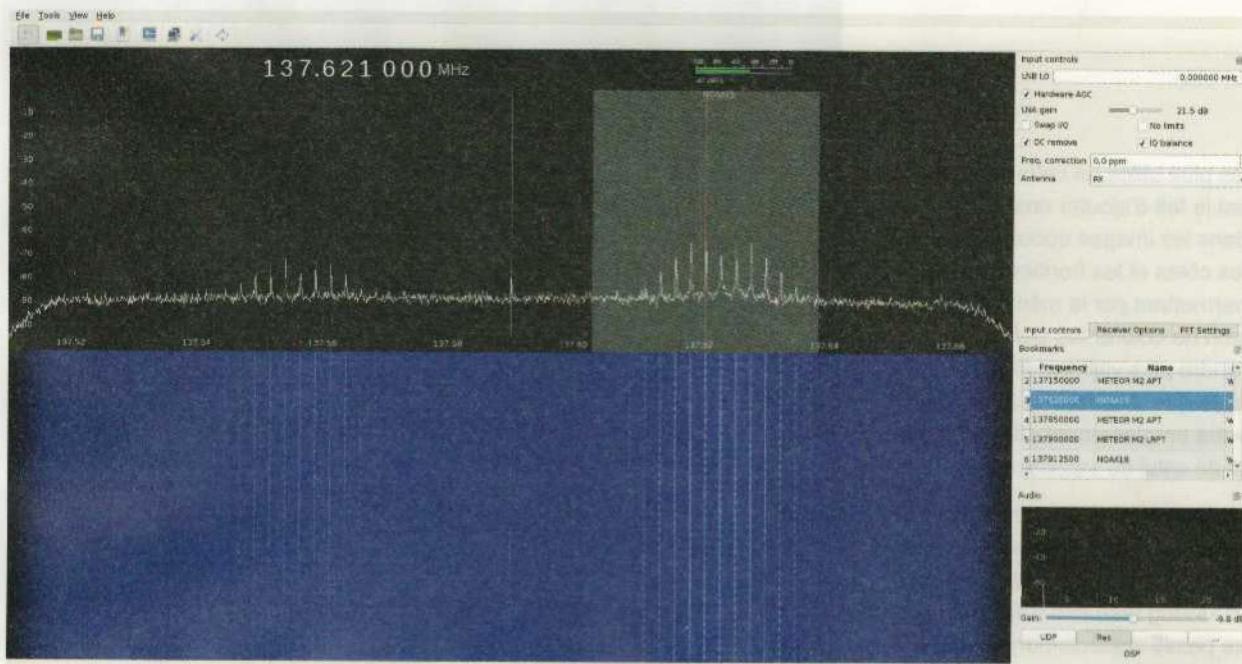
Voici un signal qui sera très problématique à démoduler et décoder en raison d'un phénomène appelé aliasing. Les pics de fréquences sont doublés, inversés et superposés. Ce problème provient généralement d'un gain trop important, faisant apparaître un signal fantôme qui n'existe pas. Je soupçonne également Gqrx d'être en cause lors d'une réception reposant sur rtl_tcp.

3. QUELQUES PROBLÈMES ET SOLUTIONS

Comme je l'ai déjà dit, ne vous attendez pas à des résultats mirifiques du premier coup (ou alors vous avez bien plus de chance que moi, ou tout simplement avec un meilleur emplacement). Personnellement, entre l'horreur reçue lors de ma première tentative et ma dernière image avant de conclure le présent article, c'est quelques 35 passages de satellites et captures de signaux qui ont été nécessaires.

Parmi les points que j'ai eu le loisir de régler, le plus important est, **sans le moindre doute**, l'emplacement de l'antenne. J'ai eu d'énormes doutes quant à sa réelle efficacité, mais elle n'était en rien en cause (sauf quand j'ai mesuré 30° par rapport à l'axe horizontal au lieu de l'axe vertical), pas plus que la longueur du coax jusqu'au récepteur ou encore les réglages de Gqrx. J'ai ainsi appris, et vous pouvez vous moquer, que les plaques de BA13 et d'isolant ne tiennent pas par miracle et reposent sur une structure en métal, catastrophique pour une telle réception. Le fait de simplement déplacer l'antenne de mon grenier privatif vers les combles a radicalement changé les choses.

L'utilisation d'un amplificateur, tel que le LNA4ALL, n'apporte pas grand-chose si l'antenne est efficace. Au contraire, utiliser un tel matériel, ou trop pousser



En cas de problème d'aliasing et si la réduction manuelle du gain ne suffit pas, la meilleure solution consiste à décaler la fréquence de réception (ici sur 137,590 Mhz au lieu de 137,625 Mhz). L'aliasing est toujours là, mais, comme les signaux fantômes ne sont plus superposés aux signaux réels, il devient facile de ne démoduler que ces derniers.

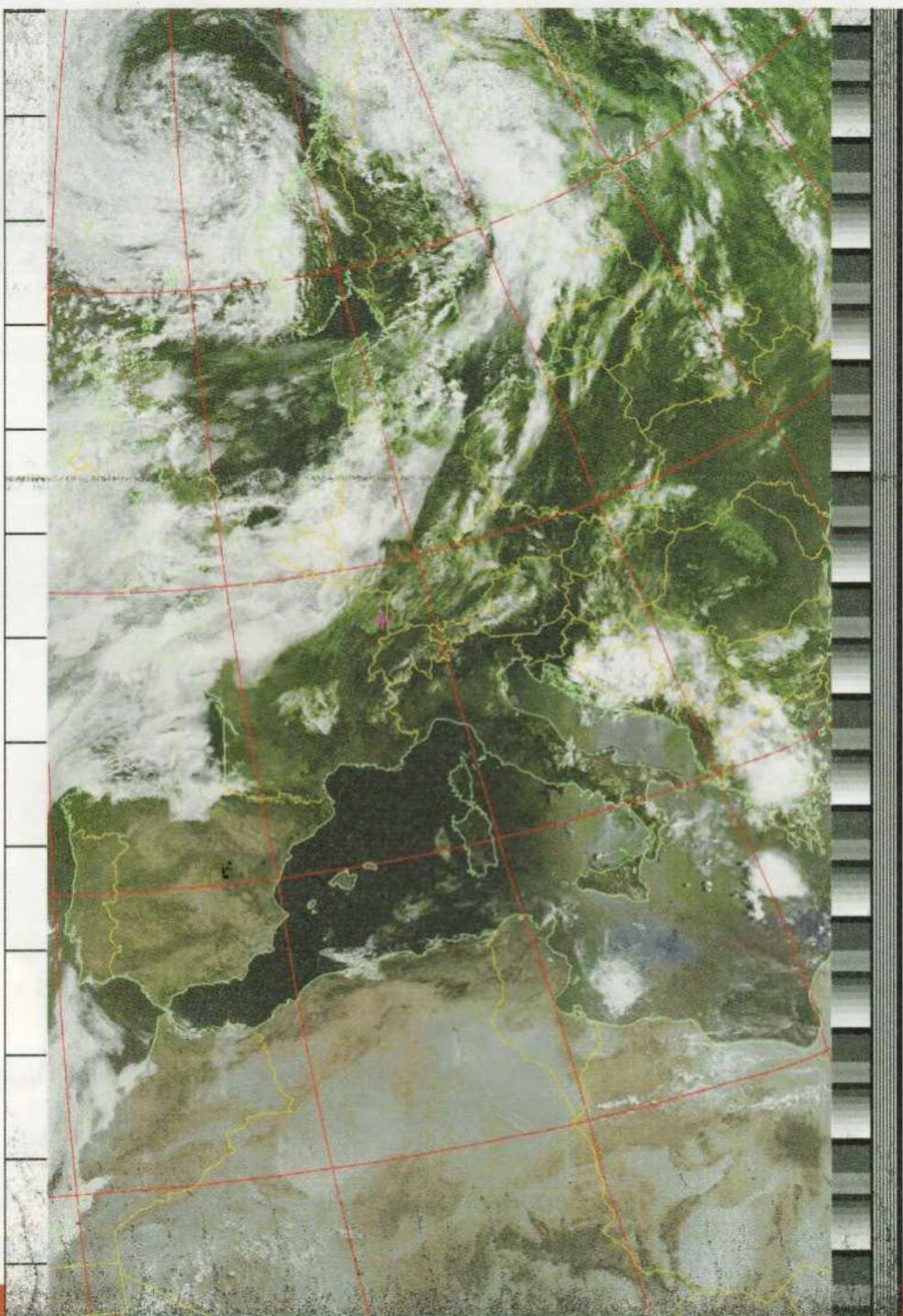
le gain, provoque plus de problèmes qu'autre chose. Lorsque vous amplifiez un signal non filtré, vous amplifiez également le bruit et un phénomène de repliement de spectre (ou *aliasing* en anglais) apparaît. Ceci se manifeste sous la forme d'une réception « miroir » apparaissant de l'autre côté de la fréquence de base (cf. capture). Mieux vaut d'ailleurs régler le récepteur sur une fréquence proche de celle souhaitée et capturer sur une bande légèrement décalée.

L'antenne sur le toit, ou comme chez moi, dans les combles, c'est très bien, mais lorsqu'on habite en immeuble et non dans une maison, on ne peut pas passer son temps au grenier. Il est cependant possible de travailler à distance en faisant, par exemple, fonctionner **rtl_tcp** sur une Pi ou un vieux PC et en demandant à GQRX ou SDR# de s'y connecter via le réseau local. Les données brutes reçues en USB sont alors transmises par ce biais et traitées, au calme, dans son salon, trois étages plus bas. Sachez cependant que dans la version actuellement packagée dans Debian, un bug dans **gr-osmosdr** (utilisé par GQRX) provoque tantôt des comportements étranges (le problème ne se pose pas avec SDR#).

Enfin, est c'est sans doute une suite possible de mes expérimentations, il est possible de procéder à la réception entièrement en

ligne de commandes. Comme expliqué, la modulation utilisée pour les transmissions APT n'a rien d'exotique et l'outil **rtl_fm** peut parfaitement s'en charger et, avec l'aide de SoX, produire des fichiers WAV qu'il suffira de rééchantillonner après copie sur un PC via le réseau. La ligne de commandes à laquelle je suis arrivé et donnant, pour moi, les meilleurs résultats est : **rtl_fm -f 137912k -M fm -s 40000 -F 5 - | sox -t raw -r 40000 -e signed -b 16 -c 1 - NOAAout.wav** (réception sur 137,912 Mhz, NOAA 18, démodulation FM, largeur de bande de 40 Khz, filtre RIF taille 5, et le tout passé à SoX pour créer un fichier WAV).

Pour finir, je dirai que l'important est de ne surtout pas jeter l'éponge et, vous aussi, vous recevrez des photos prises par un bolide se déplaçant à 16000 km/s et se trouvant à 850 km d'altitude. Si j'y suis arrivé, alors vous le pouvez aussi. Je n'en doute pas un instant ! **DB**



Voici l'une des dernières images que j'ai capturées sur une longue transmission très claire. On reconnaît immédiatement les différents pays et un bel anticyclone à l'ouest de la Norvège. Entre cette image et l'amas grisâtre de pixels de ma première capture, c'est presque 40 réceptions qui m'ont été nécessaires pour affiner les réglages, corriger les problèmes et trouver un bon emplacement pour l'antenne.

ATTENTION

Juste avant de finaliser le magazine, il apparaît que le site wxtoimg.com ait actuellement des problèmes et soit inaccessible. Si c'est toujours le cas au moment où vous lirez ceci, deux solutions s'offrent à vous : télécharger WXtoImg ailleurs (merci Google) ou utiliser un autre outil.

SatSignal est un bon choix mais il ne fonctionne que sous Windows. Les fichiers WAV qu'il utilise

sont au même format que ceux pour WXtoImg, mes explications ne changent donc pas.

Vous pouvez télécharger la version gratuite de SatSignal sur satsignal.net ainsi que les DLL nécessaires (<http://www.satsignal.eu/software/runtime.htm>) et le programme de tracking WXtrack (<http://www.satsignal.eu/software/wxtrack.htm>).

Une autre alternative possible (peut-être meilleure) est APTDecoder, toujours sous Windows, disponible sur <http://leshamilton.co.uk/aptdec.htm>

Menu
ON/OFF

Menu

Run
Control

Intensity/Adjust

Cursors

ACQUIRE

SAVE
RECALLDEFAULT
SETUP

SINGLE

MEASURE

DISPLAY

UTILITY

HELP

RUN
STOP

MESUREZ LA VITESSE DE LA LUMIÈRE DANS LES CÂBLES !

Push-Variable

MATH

Denis Bodor

CH 1



Push-Zoom

SET TO
50%HORIZ
MENU

FORCE

Position

Derrière ce titre délibérément accrocheur se cache non seulement l'exploration d'un phénomène très intéressant, mais aussi une technique permettant de mesurer une valeur très importante dès lors qu'on parle de câblage, de radiofréquence et d'antenne. Je vous invite donc à me suivre dans cette aventure, accompagné de votre oscilloscope, pour littéralement voir la vitesse de la lumière en action... ou presque.

Print

⚠ All Inputs 1MΩ//18pF ≤ 100Vpk

Oui, pour cet article, un oscilloscope sera nécessaire et c'est justement l'une des utilisations peu connues de ce matériel que nous allons découvrir. L'histoire commence lors de la création d'une antenne et plus exactement une antenne à double dipôle croisé (ou DCA pour *Double Cross Antenna*). Ce type d'antenne est très intéressant pour qui s'intéresse aux ondes polarisées circulairement, comme celles des émissions des satellites NOAA, mais implique l'utilisation de câbles coaxiaux de longueurs précises, dépendante de la longueur d'onde devant être réceptionnée et du type de câble lui-même.

1. LE POURQUOI DU COMMENT

Dans les grandes lignes, la raison pour laquelle les câbles utilisés pour assembler l'antenne et connecter les dipôles n'ont pas tous la même longueur, découle de la nécessité d'introduire un délai permettant de combiner correctement les signaux captés par les éléments de l'antenne. Dans les diagrammes de construction, deux des câbles ont une longueur dépendante du quart de la longueur d'onde et les deux autres d'un 1/2 de cette longueur d'onde. Mais la taille exacte dépend également d'un autre élément : le facteur de vitesse ou *velocity factor* en anglais, abrégé VF (tantôt désigné sous « *Velocity of Propagation* » dans les spécifications techniques des câbles).



Et c'est précisément là que les choses deviennent très intéressantes. La vitesse de propagation d'une onde électromagnétique, et donc des signaux radio, est généralement décrite comme étant la vitesse de la lumière, soit 299792458 mètres par seconde... dans le vide. Seulement voilà, un câble coaxial, ce n'est pas « le vide » et la propagation du signal s'en trouve ralentie. C'est cette vitesse de propagation qui sera précisément dans le cas d'une telle antenne, pour introduire un délai indispensable au bon fonctionnement du dispositif.

Ce ralentissement est généralement exprimé sous la forme d'un pourcentage de la vitesse idéale (celle de la lumière dans le vide) et est une valeur critique dans ce genre de réalisation. Dans le cas du très courant câble type RG 58/U avec une impédance de 50 ohms (le même que pour le bon vieux réseau coax 10base2), ce VF est de 66, soit 66% des quelques 300000 km/s. Cela fait tout de même une différence de quelques 102000 km/s !

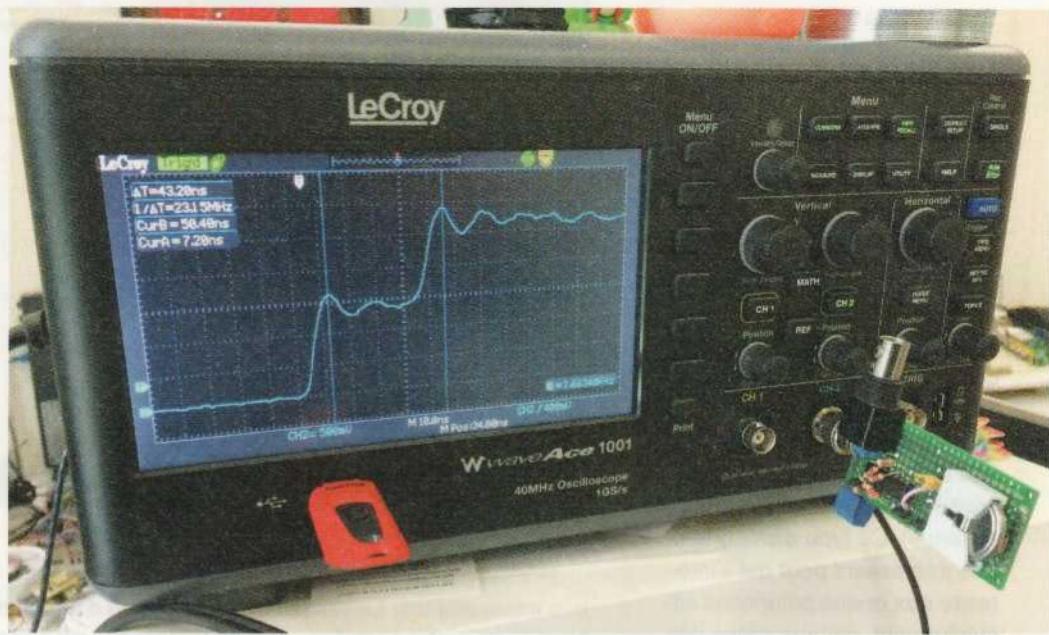
L'impédance, qui est le pendant au courant alternatif de ce qu'est la résistance au courant continu, est également un point important dès lors qu'on parle de réception radio et de câbles. Bon nombre de débutants commettent l'erreur d'utiliser le câble qui leur tombe sous la main, mais l'impédance caractéristique de ce dernier devrait impérativement être égale à l'impédance en entrée du récepteur. Certains récepteurs DVB-T (TNT) utilisés comme périphériques SDR ont tantôt une impédance en entrée d'environ 75 ohms (comme les câbles d'antenne TV), mais les récepteurs plus récents, pourtant construits sur la même base, mais spécialement dédiés à la radio logicielle (avec connecteur SMA), préfèrent quelque chose de plus classique dans le domaine : 50 ohms.

La question qui se pose alors est la suivante : en tombant sur un bout de câble coaxial quelconque, sans

Le générateur de signal généralement intégré aux oscilloscopes ne convient pas pour créer un TDR, car la transition de l'état bas à l'état haut prend plus de temps (ici 400ns) que celui que le signal mettra à être réfléchi en bout de câble.



Voici le résultat qu'on souhaite idéalement obtenir avec, sur la droite, le montage générant les impulsions et à l'écran le signal caractéristique. Le délai entre les deux points mesurés correspond au temps aller/retour de l'impulsion dans le câble, à une fraction de la vitesse de la lumière.



référence précise, comment connaître non seulement l'indispensable *velocity factor* mais, tant qu'à faire, vérifier également son impédance ? La réponse est simple, il faut construire un réflectomètre temporel ou TDR, de l'anglais *Time Domain Reflectometer*).

2. LE RÉFLECTOMÈTRE TEMPOREL

Le principe de fonctionnement d'un TDR est relativement simple et consiste à envoyer une impulsion électrique dans un câble tout en observant le signal à l'aide d'un appareil de mesure. Dans le cas d'un câble adapté en impédance, généralement équipé d'un « bouchon » n'étant rien d'autre qu'une résistance dont la valeur est égale à l'impédance du câble, il ne se passera rien. C'est d'ailleurs pour cette raison que les bons vieux réseaux 10base2, utilisant du câble coax, étaient équipés de « bouchon » ou « terminator ».

Si le câble n'est pas adapté en impédance, en revanche, l'impulsion va parcourir le câble et être réfléchie à son extrémité avant de revenir vers la source. On verra donc apparaître sur l'appareil de mesure la première impulsion, suivie d'une seconde qui renforcera la première avec un léger décalage temporel. La vitesse à laquelle cette

impulsion et sa réflexion se déplacent dans le conducteur est dépendante du VF du câble. Le temps de parcours, lui, est dépendant de ce facteur, mais aussi de la longueur du conducteur.

Un autre cas de figure peut se présenter, si le câble est en court-circuit et donc avec une impédance ramenée à 0 ohm (comme un « bouchon » de non pas 50 ohms, mais 0). Dans cette situation, l'impulsion va parcourir le conducteur jusqu'à atteindre le court-circuit, mais au lieu d'être réfléchie ou absorbée par une résistance, celle-ci va poursuivre sa ballade et ensuite, apparaître comme une chute de tension. Ce que l'on constate alors à la source est une réduction de l'impulsion avec un délai, là encore, dépendant de la longueur du câble jusqu'au court-circuit et du VF du conducteur.

Un TDR s'avère donc utile pour plusieurs applications pratiques. Il permet de détecter les courts-circuits avec une relative précision si l'on connaît le VF du câble. Il permet également de déduire le VF pour un câble donné si l'on connaît sa longueur, en mesurant simplement le temps de réflexion. Et enfin, il est possible de déterminer l'impédance en plaçant une résistance variable (potentiomètre) en bout de câble puis en l'ajustant jusqu'à n'avoir ni de réflexion ni de chute de tension et donc de façon à voir l'impulsion totalement absorbée à l'extrémité du conducteur. La valeur de la résistance correspond alors à l'impédance du câble.

3. CONSTRUIRE SON TDR

La façon la plus simple de créer un TDR consiste à reposer sur l'utilisation d'un oscilloscope. Celui-ci permettra de très facilement visualiser les variations de tension et de mesurer les délais entre ces variations avec une relative exactitude. Il suffit donc de connecter le câble à mesurer en lieu et place de la sonde de l'oscilloscope et d'y injecter une impulsion de façon régulière.

Les oscilloscopes sont généralement équipés d'un générateur de signal permettant leur calibration. La fréquence ici n'est pas le problème puisque nous nous intéressons uniquement au flanc montant du signal et ce à une échelle de temps de l'ordre de quelques dizaines de nanosecondes. N'importe quelle fréquence en dessous du mégahertz pourra donc convenir.

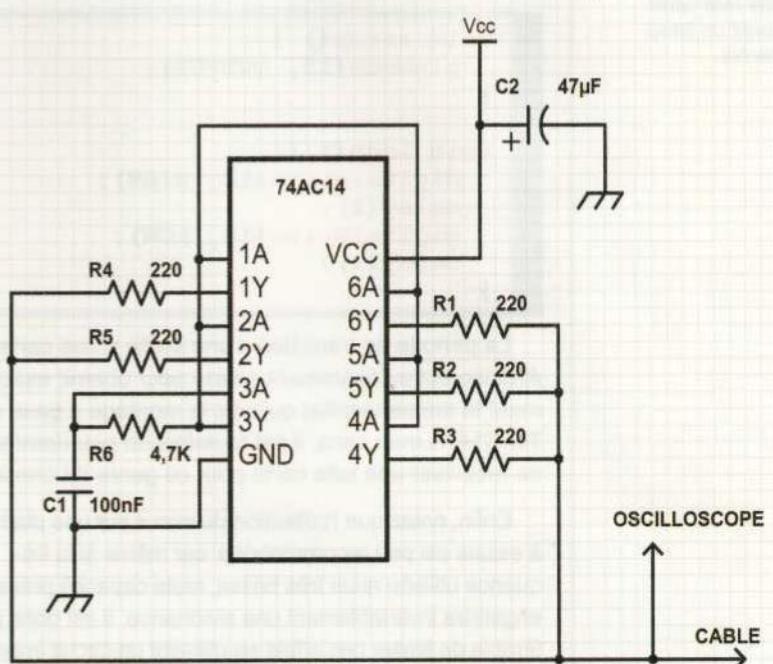
Ce qui est plus problématique, en revanche, est la vitesse à laquelle la transition de l'état bas à l'état haut se fait. Dans le cas de l'oscilloscope LeCroy WaveAce1001 que j'ai actuellement sous les yeux, par exemple, le signal de calibration à 1 Khz ne présente pas de front montant suffisamment net. En observant le signal, on se rend compte que le passage de 0V à l'état haut prend quelques 400 ns. En faisant un rapide calcul, si l'on voulait effectivement mesurer une quelconque réflexion dans un câble avec un VF de 66%, et que celle-ci soit donc visible après les 400 ns, il nous faudrait un câble d'une longueur minimum de presque 10 mètres. Et ce, sans compter le fait qu'un signal qui n'a presque rien de carré à cette échelle n'est absolument pas adapté pour ce genre de mesure.

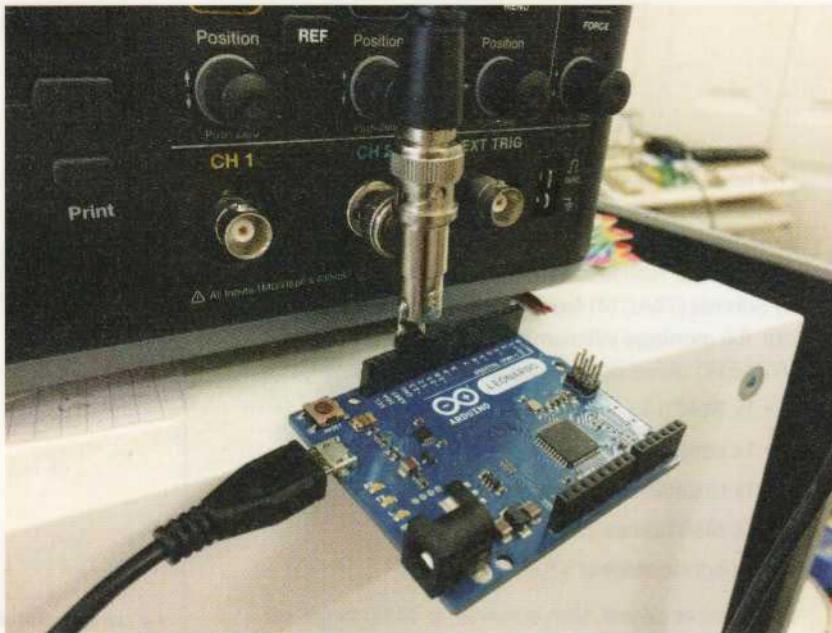
Il nous faut donc une autre source, capable de nous fournir des fronts montants bien plus nets. L'une des solutions possibles consiste à créer un petit circuit reposant sur l'utilisation d'un sextuple inverseur à trigger de Schmitt (74AC14) fonctionnant comme un oscillateur. Ce montage initialement décrit par Alan Wolke (W2AEW) utilise peu de composants :

- 1x 74AC14 ;
- 1x condensateur 100nF ;
- 1x résistance 4,7 Kohms ;
- 5x résistances 220 ohms ;
- 1x condensateur 47µF.

L'élément clé est, bien entendu, le 74AC14 qui est composé de six inverseurs logiques comme un 74HC14, mais utilise une technologie Advanced CMOS (d'où le « AC ») permettant des transitions rapides. Comme le montre le schéma, l'une des entrées est connectée à la sortie correspondante via une résistance et un condensateur de manière à former un réseau RC et donc un circuit oscillant. La sortie de l'inverseur est alors connectée aux entrées des cinq autres inverseurs du composant dont les sorties sont reliées entre elles via des résistances de 220 ohms.

La construction du TDR ne nécessite pas beaucoup de composants. L'élément clé est le circuit logique 74AC14, utilisé en partie comme oscillateur et présentant surtout une transition très rapide dans la génération des impulsions.





Pour générer le signal dont nous avons besoin, nous pouvons également utiliser une carte Arduino et un croquis simpliste. Pour une mesure ponctuelle, cela fera parfaitement l'affaire, mais monopoliser ainsi une telle carte serait un beau gâchis.

Le signal carré généré peut alors être connecté à la fois au câble à mesurer et à l'entrée de l'oscilloscope où va apparaître non seulement le signal en question, mais également les artefacts découlant de la réflexion dans le conducteur, si le câble n'est pas terminé correctement.

Une autre approche possible consiste à utiliser une carte à microcontrôleur comme une Arduino pour générer un signal identique avec un croquis simpliste comme :

```
void setup() {
    pinMode(12, OUTPUT);
}

void loop() {
    digitalWrite(12, HIGH);
    delay(1);
    digitalWrite(12, LOW);
    delay(1);
}
```

La période de transition d'une sortie d'une carte Arduino est suffisamment courte pour obtenir exactement le même résultat qu'avec le montage à base de 74AC14. À mon sens, il est toutefois un peu dommage de mobiliser une telle carte pour ce genre de choses.

Enfin, notez que l'utilisation du circuit sur une platine à essais est peu recommandée, car même si la fréquence utilisée reste très basse, toute capacité présente engendre invariablement une résonance. Il est donc préférable de limiter ces effets en utilisant un circuit imprimé ou une plaque pastillée, et en limitant au maximum la

longueur des connexions, aussi bien entre le montage et le câble à tester, qu'entre le montage et l'oscilloscope. Pas question par exemple d'utiliser les sondes de ce dernier, mieux vaut se connecter directement à la fiche BNC de l'appareil.

4. UTILISER SON TDR

Une fois le montage en place, connecté à l'oscilloscope et alimenté, on cherchera tout d'abord à procéder à tous les réglages nécessaires pour visualiser avec la plus grande finesse le front montant de l'impulsion. Vous devriez constater que ce front montant s'étale sur une durée relativement courte, de l'ordre d'une dizaine de nanosecondes, mais également que le circuit résonne. En effet, du fait des inévitables capacités parasites, la tension fluctue un certain temps après le passage de l'état bas à l'état haut. Ce phénomène s'est avéré moins important dans le cadre de l'utilisation d'une carte Arduino, tout simplement parce que le connecteur BNC en « T » était relié directement à l'oscilloscope.

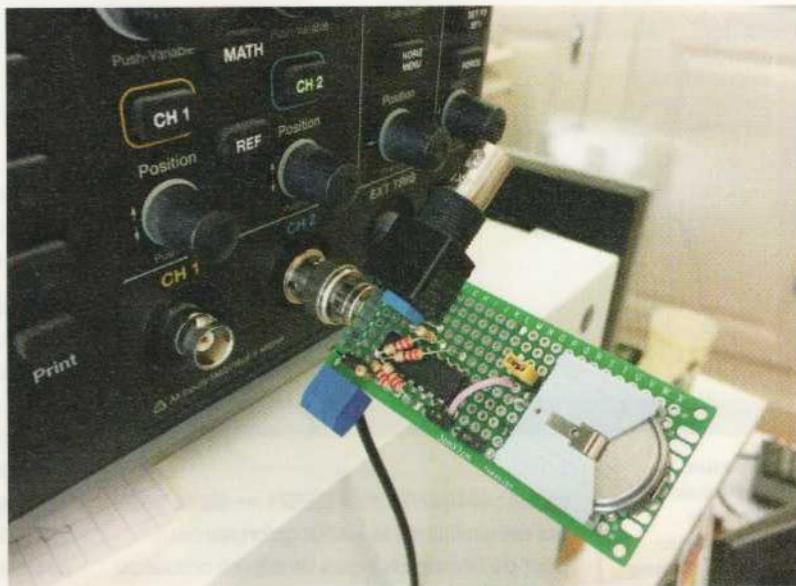
La seconde étape consiste à connecter le câble à tester après en avoir mesuré la longueur. La forme du signal va alors changer et vous devrez voir apparaître une forme en escalier. La première transition de l'état bas à l'état haut est celle de l'impulsion que nous générerons et la seconde est le signal réfléchi au bout du câble. On utilise alors les curseurs de l'oscilloscope afin de mesurer le temps écoulé entre ces deux changements de niveau. Cette durée correspond au temps de par-

cours du front montant de l'impulsion, dans un sens, puis dans l'autre.

J'ai testé deux types de câbles. Le premier, de type RG 58/U avec deux connecteurs BNC provient de ma caisse « vieux trucs de PC ». C'est un câble réseau 10base2 de 2,50 mètres. Là, aucune surprise, ce type de câble coaxial est bien connu, possède une impédance de 50 ohms et j'ai même encore des « T » et des « terminators » 50 ohms correspondants (je savais bien qu'il ne fallait pas les jeter).

Il suffit alors d'utiliser la fonction « curseur » de l'oscilloscope pour placer des repères en abscisse et celui-ci nous affiche le delta entre les deux en nanosecondes. Notre câble de 2,50 mètres (plus avec les connecteurs) est parcouru par l'impulsion et sa réflexion en 25,60 ns. Cela devrait être beaucoup moins puisque la vitesse de la lumière dans le vide est d'environ 300000 km/s, soit 30 cm/ns. Si notre mesure était « juste », le câble devrait faire presque 4 mètres : 25,60 ns divisés par deux pour avoir le temps d'un trajet, fois 30 cm/ns, soit 384 cm.

Cette différence entre la valeur mesurée et la valeur théorique « dans le vide », ou plus exactement le rapport entre les deux, est précisément le VF que nous cherchons à calculer. Il nous suffit donc de prendre la distance en centimètres, la diviser par la moitié du temps que nous avons mesuré en nanosecondes et de diviser le tout par la distance normalement parcourue en une nanoseconde dans le vide. En ajoutant 5 centimètres pour les connecteurs BNC, ceci nous donne : $255 / (25,60/2) / 30 = 0,6415 = 64\%$. Ceci est tout à fait acceptable, sachant qu'il faut également considérer les connecteurs et le montage lui-même. La valeur proche, de 65%, est celle affichée sur la page Wikipédia « Velocity factor » concernant la valeur minimum pour câble « RG-58A/U » utilisée en Ethernet 10base2. Nous sommes donc dans les clous.

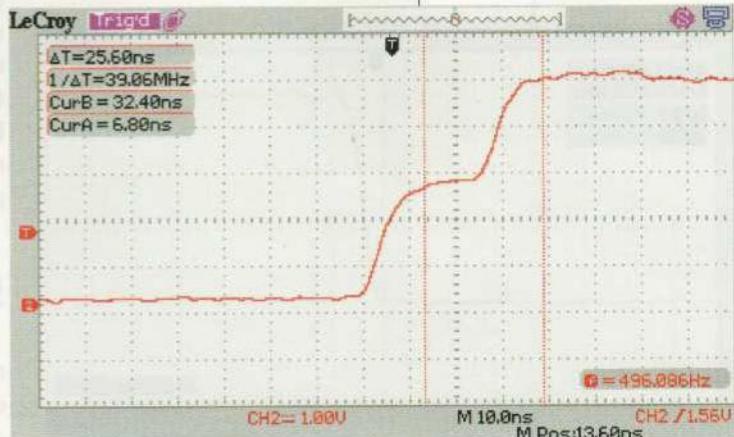


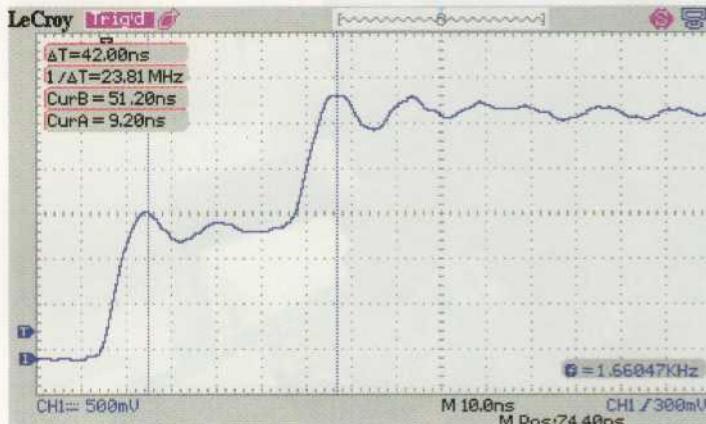
Si nous nous penchons sur le second exemple, avec un câble bien plus fin, long d'un peu plus de 4 mètres équipé à une extrémité d'un connecteur SMA (sans doute quelque chose en rapport avec une antenne pour un périphérique SDR), nous pouvons faire de même.

Cette fois le câble est branché au montage via un connecteur à visser (type domino) et on constate immédiatement plusieurs choses : il y a davantage de résonance et, le câble étant bien plus long, le délai entre l'impulsion initiale et la mesure de la réflexion est plus important : 42 ns. On applique alors le même

Le montage final n'est pas forcément le plus beau qui soit, mais le résultat est fonctionnel. L'ajout d'une source d'alimentation comme cette pile bouton CR2032 conviendra parfaitement au 74AC14 pouvant fonctionner avec une tension entre 2V et 6V.

Exemple de mesure faite sur un câble réseau 10base2 RG 58/U à l'aide d'une carte Arduino. Le câble étant relativement court (~2,5 m), il est plus difficile de déterminer l'écart entre les deux niveaux de tension et donc d'avoir des calculs précis.





Cette mesure a été effectuée en utilisant le montage à base de 74AC14 et avec un câble d'environ 4 mètres. On voit clairement que la distance temporelle entre l'impulsion originale et la réflexion est plus importante.

Le microcontrôleur d'une carte Arduino affiche une transition d'un état bas à un état haut relativement rapide (~12ns) et ce avec peu de résonance. Ici, aucun câble n'est connecté, il n'y a donc pas de signal réfléchi.

calcul : $414/(42/2)/30 = 0,6571 \approx 66\%$. Voilà qui est similaire à la valeur qu'on retrouve pour de nombreux types de câbles coaxiaux, dont le RG 58/U dans la plupart des documentations de fabricants.

Comme vous pouvez le voir, le calcul en lui-même ne présente pas de grandes difficultés à partir du moment où l'on garde à l'esprit qu'il doit être fait en utilisant les mêmes unités de mesure. Ici, nous avons utilisé des nanosecondes et des centimètres, d'où les 30 cm/ns correspondants aux 299792 km/s de la vitesse de la lumière dans le vide. Comme on l'a tous appris en classe, il ne faut simplement pas additionner des choux et des carottes, mais l'erreur peut être vite commise par inattention.

Comme ce second câble est « mystérieux » (du moins à ce stade, cf. plus loin), nous pouvons en profiter pour mesurer son impédance. Là, nous n'avons presque rien à changer à notre installation, si ce n'est

ajouter une résistance variable, sous la forme d'un potentiomètre multitour de 100 ohms, en bout de câble entre le conducteur et le blindage.

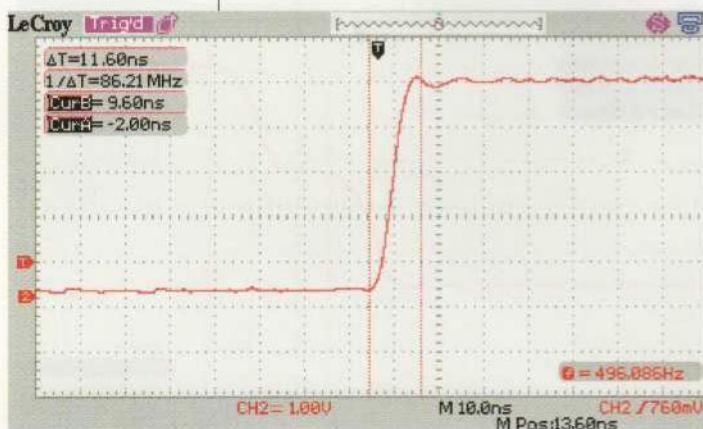
Le choix de ce type de composant n'est pas anodin, car un potentiomètre multitour permet une plus grande précision dans l'ajustement et la valeur doit rester dans une plage raisonnablement typique de l'impédance caractéristique généralement utilisée. La valeur de 50 ohms se retrouve généralement dans l'instrumentation (générateurs de fréquence, oscilloscopes, analyseurs de spectre, etc.) et la connexion d'antennes (SDR, Wifi, etc.), alors que du 75 ohms est utilisé pour le domaine audiovisuel (radio, TV, vidéo, etc.).

Pour trouver la bonne valeur, il suffit d'ajuster le potentiomètre jusqu'à supprimer la réflexion. Si la valeur est trop importante, la réflexion sera présente, mais de moindre impact, puisqu'on s'éloigne de la valeur idéale et qu'on se rapproche d'une résistance infinie (pas de connexion). Si la valeur est trop petite, une chute de tension apparaîtra, car on se rapproche du cours-circuit. Une fois la bonne position trouvée, il suffira de déconnecter le potentiomètre et de mesurer la résistance à ses bornes avec un multimètre. Celle-ci correspondra à l'impédance du câble.

Avec ce dernier câble, on retrouve tout naturellement les classiques 50 ohms confirmés par... l'inscription sur le câble lui-même qui était passée inaperçue durant tous les essais. Une inspection attentive révèle que j'avais en réalité l'information depuis le début puisqu'une inscription « RG-174/U COAXIAL CABLE 50 OHM » est présente (mais pas tout à fait lisible ! Na !). Tant mieux, ceci permet de confirmer les mesures, il s'agit effectivement d'un câble avec précisément ces spécifications.

5. MAIS ENCORE ?

La relation qui existe entre le temps de parcours de l'impulsion et sa réflexion, la longueur du câble et le VF associé nous permettent de calculer rapidement n'importe laquelle des trois





Déterminer l'impédance du câble se résume à le terminer avec une résistance de la valeur correspondante. Ici, la chute de tension provoquée lorsque l'impulsion arrive à la résistance se manifeste visuellement. La résistance n'est pas la bonne, elle est trop importante...

Dans le cas d'un câble correctement terminé, la résistance supprime toute réflexion et toute chute de tension. L'impulsion est totalement absorbée et, en dehors de la résonance même du circuit, nous n'avons aucune trace du signal à l'emplacement où se trouve le curseur. La valeur de la résistance correspond à celle de l'impédance du câble.

valeurs si elle est inconnue, à partir des deux autres. Nous pouvons donc, comme ici, déterminer VF pour un câble aux spécifications mystérieuses mais, connaissant cette valeur, nous pouvons également déduire la longueur d'un câble, par exemple. Ceci pourra être utile pour obtenir une estimation avec une relative précision sans avoir à dérouler et mesurer une bobine.

L'une des autres utilisations d'un TDR consiste à détecter un court-circuit. Si vous avez un câble coaxial présentant un tel problème, vous pourrez déterminer à quelques centimètres près, l'emplacement exact du court-circuit, et donc régler bien plus facilement le problème que par une inspection sur toute sa longueur.

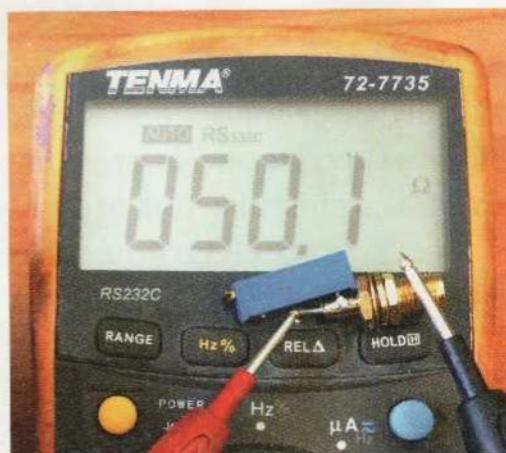
Cependant, il faut bien comprendre que les limites de cette technique sont celles dépendantes de la qualité de votre matériel. Un petit oscilloscope de poche économique, par exemple, n'aura ni la bande passante ni la fréquence d'échantillonnage nécessaire pour visualiser des signaux aussi brefs. Vous ne verrez tout simplement pas la réflexion.

Ces caractéristiques déterminent également la taille minimum des câbles que vous pouvez mesurer. Comme le montrent les exemples utilisés ici, le câble RG 174/U de 4 mètres présente une courbe bien plus lisible que le RG 58/U de 2,50 mètres. Plus le câble est long, plus le temps de parcours sera important, et moins la qualité de l'oscilloscope aura d'impact.

Quoi qu'il en soit, j'espère que cet article vous aura fait découvrir une utilisation peu



connue de votre équipement et, qu'au-delà de l'aspect purement pratique, vous avez comme moi été fasciné par l'aspect scientifique physique de ces manipulations. Après tout, nous avons utilisé une vitesse qui, dans la vie de tous les jours, paraît totalement irréelle et impossible à appréhender... DB



Une fois la bonne valeur trouvée, il ne reste plus qu'à la mesurer. Ici, elle est de 50 ohms, ce qui correspond parfaitement aux caractéristiques du câble qui s'avère être un type RG 174/U.



LA FABULEUSE HISTOIRE DES CALCULATEURS NUMÉRIQUES À L'ÈRE ÉLECTROMÉCANIQUE

Yann Guidon ["Fallait juste y penser !"]

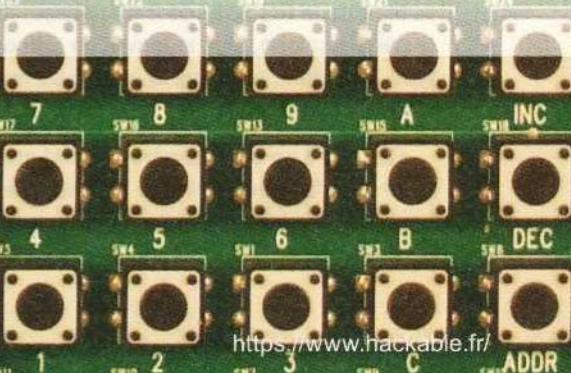


Notre monde est aujourd'hui totalement baigné de technologie digitale ! Pour en arriver à regarder des vidéos de chats sur notre téléphone dans le métro, il aura fallu bien plus que de la théorie :

Boole et de Morgan avaient déjà couvert les bases mathématiques du calcul binaire au 19^e siècle. À partir de là, des générations d'ingénieurs

ont cherché et inventé des nouveaux circuits, puis ont travaillé sans relâche pour réduire leur taille et leur consommation, tout en accélérant leur fonctionnement. Je vous propose de retracer cette longue évolution, pour mieux apprécier et comprendre les avantages et inconvénients de chaque génération technologique. Et la première étape décisive dans l'histoire des ordinateurs fut l'utilisation des relais électromécaniques.

888888888



Vous me direz, « l'histoire c'est gentil, mais ce n'est pas ce qui me permettra de faire clignoter une LED avec un Arduino. » Et pourtant, chaque invention, chaque étape, a laissé des traces et les plus importantes subsistent encore, grâce à leurs avantages propres. Vous serez probablement amené à connecter ces composants à votre microcontrôleur et j'espère que cette perspective historique vous permettra de mieux les choisir pour votre prochain montage. Car si la

course à la performance se poursuit encore, certaines technologies jugées obsolètes pour certains usages ont encore un intérêt dans d'autres, puisqu'elles sont largement disponibles, simples et très abordables.

1. LES PRÉCURSEURS ET L'ÈRE MÉCANIQUE

Nous n'allons pas faire une histoire complète de l'électronique (un domaine fascinant et absolument inépuisable), mais aborder les techniques servant à réaliser des calculs et fabriquer des ordinateurs. Cela implique l'utilisation de la logique binaire, même si cela n'était pas évident au tout début.

Fig. 1 : Reconstitution d'une machine à calculer automatique, décimale et mécanique, imaginée par Charles Babbage il y a presque deux siècles (source : The Computer History Museum).

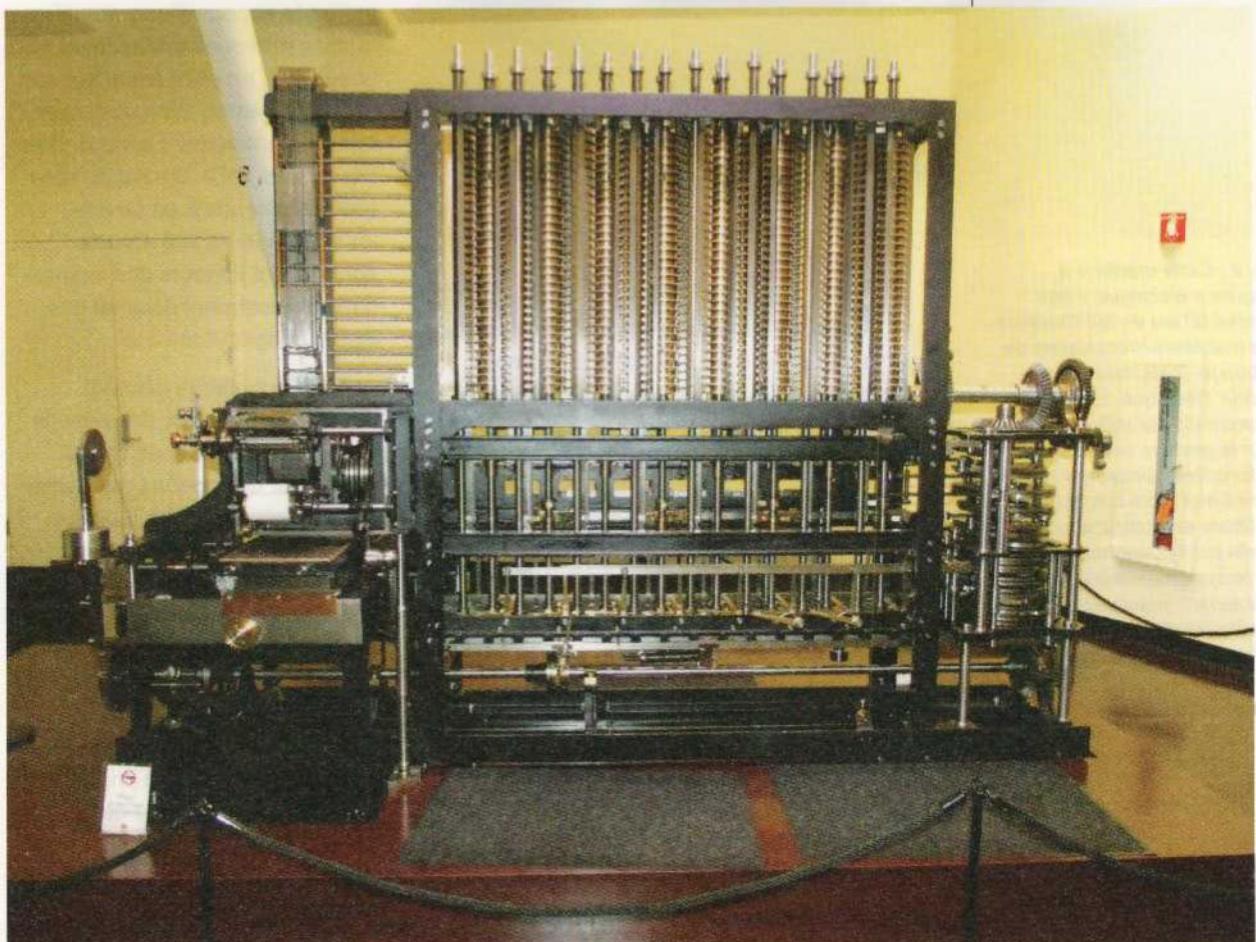




Fig. 2 : Cette machine à calculer « électrique » des années 50 est en fait l'héritière des machines mécaniques de Babbage, mais mue par un moteur électrique. La Marchant Figurematic est utilisée surtout pour la gestion, car incapable de fonctions complexes (ou alors il faut alors combiner des additions et multiplications), et elle est évidemment basée sur le système décimal. Maintenant, imaginez une salle remplie de dizaines de ces machines et leurs opérateurs...

Après tout, depuis l'Antiquité, l'hydraulique et la mécanique permettaient de compter, mesurer et même calculer : les grandeurs continues (une position sur un cadran, le niveau de remplissage d'un réservoir, ou la vitesse de rotation d'un axe) effectuent des calculs dits « analogiques ». Mais les humains ont besoin aussi de calculer des grandeurs discrètes, comme de l'argent, ou scientifiques, dont la précision dépasse les limites des bouliers, des règles à calcul ou de la Pascaline...

Ainsi, au 19ème siècle, Charles Babbage [1] avait imaginé ses machines à calculer automatiques en utilisant la technologie de l'époque, c'est-à-dire de la mécanique de précision et des cartes perforées pour machines à tisser. Ce système de cartes avait depuis peu été transposé des instruments

(comme les boîtes à musique ou les orgues de barbarie) par Basile Bouchon, puis était passé à la postérité grâce à Joseph Marie Jacquard à l'aube du 19ème siècle.

Même si les trous sont par définition binaires (il n'y a pas de demi-trou), les roues dentées ont permis de travailler naturellement en décimal, une base directement adaptée aux besoins de l'époque, comme calculer des tables trigonométriques ou gérer la comptabilité d'un ministère. Bien que ces ambitions n'aient pas pu être réalisées du vivant de Babbage, elles ont laissé une trace profonde et pour d'innombrables raisons pratiques (maintenant obsolètes), on retrouve le support natif du système décimal jusque dans les ordinateurs conçus dans les années 1970. D'ailleurs, bien qu'ils fonctionnent en binaire, les processeurs x86 actuels contiennent toujours des instructions d'ajustement décimal (les instructions **DAA** et **DAS**)...

Mais le système décimal, bien qu'évident pour des raisons culturelles, ne l'est pas forcément pour des raisons mathématiques. Selon certains, puisque la base de logarithme naturel **e=2,71828...** est plus proche de 3 que de 2, alors la base ternaire serait mathématiquement la plus adaptée. Ainsi, dans les années 1840, le britannique Thomas Fowler [2] a fabriqué une machine à calculer sur ce principe. Cette idée a été reprise dans les années 1960-1970 (surtout en Russie [3]), mais nous constatons aujourd'hui que cette approche n'a eu aucun succès [4]...

2. LE RELAIS

En plus de l'impossibilité d'avoir un demi-trou dans du carton, l'hégémonie du binaire trouve aussi ses racines dans une autre invention du 19ème siècle. Vers 1830, plusieurs inventeurs créent simultanément le relais électromagnétique [5], qui n'est finalement qu'un interrupteur électrique mu par un électro-aimant. Ainsi, du courant peut contrôler du courant, la boucle est bouclée !

Et encore mieux, un courant faible peut contrôler un courant plus fort. Ce système permet d'amplifier ou d'aiguiller du courant et il se développe avec le télégraphe (pour justement relayer les impulsions, d'où son nom), puis le téléphone (pour acheminer et aiguiller un appel, un signal analogique, sur la ligne du destinataire).

Puisqu'un relais contient un commutateur électrique qui travaille avec seulement deux états, fermé ou ouvert, l'algèbre binaire de Boole (promue au rang de la désormais incontournable *logique booléenne*) est directement appliquée et des systèmes de plus en plus sophistiqués ont été rapidement construits.

Ainsi, les automatismes de contrôle de machines (comme les ascenseurs, les machines industrielles, les centres de commutation téléphonique...) ont poussé le développement industriel de ces composants, qui se sont adaptés à d'innombrables usages. D'une conception simple et robuste, donc à un prix relativement abordable (selon le prix du cuivre) et avec une large plage

de tolérance, on les utilise encore partout pour contrôler la distribution électrique en haute tension par exemple. C'est donc un relais que vous utiliserez probablement pour contrôler l'allumage d'une lampe en 230V avec votre Arduino sous 5V.

C'est aussi un relais qui clique dans un oscilloscope lors d'un changement de calibre : la résistance quasi nulle des contacts, ainsi que la très faible atténuation en très haute fréquence, font que les relais sont souvent utilisés pour commuter des signaux de précision jusqu'à plusieurs gigahertz.

À mesure que les systèmes de communication s'électrisent, l'interfaçage avec le monde physique nécessite alors des systèmes électromécaniques adaptés et on voit apparaître à la fin du 19ème siècle les télécritures : ce terme est la contraction de « *teletypewriter* », qui signifie machine à écrire à distance. En français, ils sont aussi appelés télécritteurs et on retrouve ce terme sous linux, encore plus condensé : **tty** !

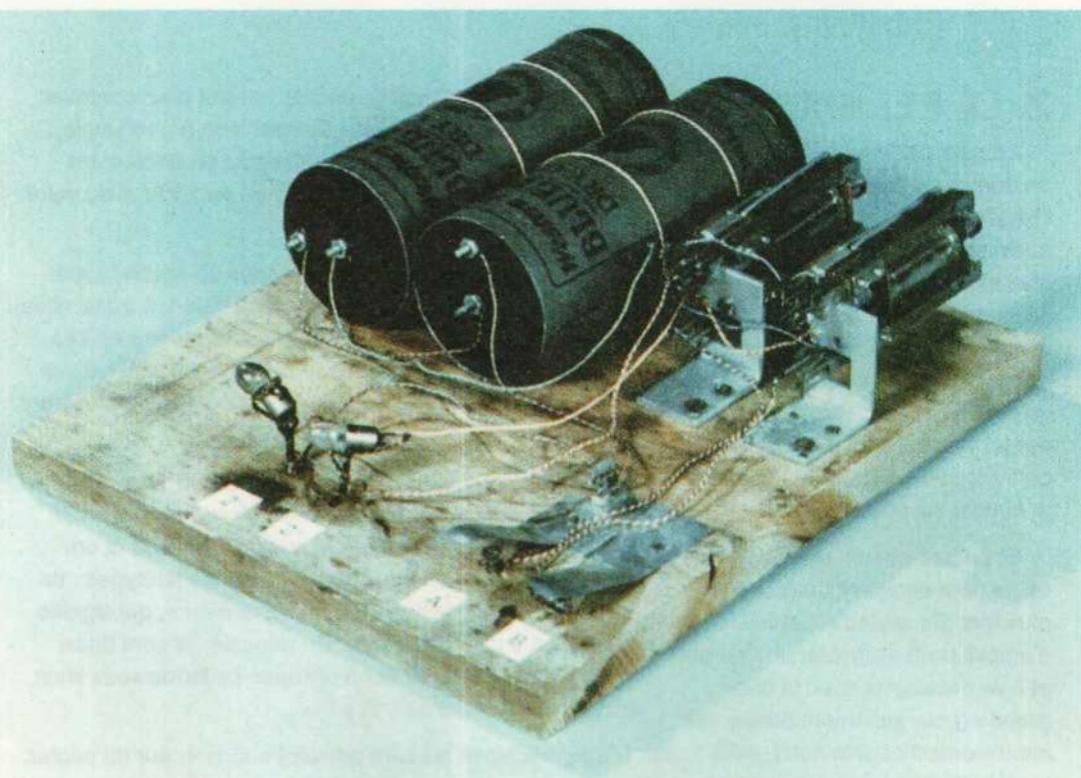
Ces machines à écrire peuvent imprimer sur du papier, mais aussi envoyer ou recevoir les caractères directement sous forme d'impulsions électriques, par ligne télégraphique, puis téléphone et même par radio. Le code morse laisse alors place au code Baudot, l'ancêtre à 5 bits du code ASCII, que l'on stocke alors sur des rubans de papier perforé. Le réseau de radiotélétype crypté allemand (appelé « *code Fish* » par les britanniques) fut un des moyens de communication ciblés par les Alliés durant la Seconde Guerre Mondiale.

Mais ce sont les relais téléphoniques qui ont le plus influencé l'informatique naissante dans les années 1930. Ils sont alors disponibles en quantités suffisantes pour équiper les nombreux commutateurs téléphoniques, qui sont alors déployés sur tous les territoires, entraînant ainsi une économie d'échelle. Les relais furent progressivement optimisés électriquement (pour être plus efficaces) et ont été dérivés dans d'innombrables configurations : on en trouve avec différentes tensions de fonctionnement, et toute une panoplie de nombre et types de contacts.

Mais surtout, on peut réaliser un nombre incroyable de fonctions logiques, car la bobine et tous les contacts sont électriquement isolés les uns des autres. C'est un véritable composant à (presque) tout faire qui peut interfaçer des circuits indépendants. Claude E. Shannon (au MIT



Fig. 3 : Un demi-additionneur binaire construit avec deux relais en 1937 par George Robert Stibitz (1904 – 1995). Les gros cylindres sont des piles sèches qui alimentent les deux bobines et les deux ampoules à incandescence. Une porte AND et une porte XOR sont très faciles à réaliser avec des relais !



puis Bell Labs) étudia, développa et présenta l'application de l'algèbre de Boole aux relais en 1936 [6], précédant d'un an Akira Nakashima au Japon.

Et par rapport à une technologie purement mécanique, la technologie électromagnétique a de nombreux avantages, comme une usure plus faible, moins d'entretien, une meilleure réactivité, une transmission instantanée des informations, donc la machine est moins lourde, moins chère et plus facile à mettre au point. Avec juste quelques types de pièces standardisés (des relais, des boutons, des lampes, du fil, une alimentation électrique et des supports), chacun peut réaliser une architecture qui lui convient, alors que les machines de Babbage nécessitaient d'innombrables pièces de précision très spécialisées...

3. ALLÔ ?

Ainsi, la première génération d'ordinateurs ressemblait un peu à un centre de commutation téléphonique, car les machines étaient souvent conçues par des ingénieurs du téléphone avec les techniques, les méthodes et les composants qu'ils connaissaient déjà. Parmi les réalisations emblématiques, on trouve par exemple :

- Les machines Z1, Z2, Z3 et Z4 qui furent créées de 1936 à 1945 par Konrad Zuse [7], le pionnier de l'informatique allemande. Comme pour la plupart des projets de ce type, la motivation était d'automatiser les lourds calculs d'ingénierie. Initialement destinée à la conception aéronautique, la Z3 utilisait les premiers nombres en virgule flottante et ses 2000 relais téléphoniques étaient contrôlés par un programme perforé sur une bobine de film 35mm. Le nombre de 2000 relais peut sembler énorme, mais c'est à peine le dixième d'un petit commutateur téléphonique automatisé. Son successeur, la Z4, fut utilisée en France jusqu'en 1959 [8]...

- Les toutes premières « Bombes » anglaises [9] étaient électromagnétiques. Ces machines de cryptanalyse des alliés, conçues uniquement pour décoder les messages allemands de l'infâme Enigma [10], étaient initialement à base de composants prévus pour les centrales téléphoniques. Plus tard, une version américaine construite par Bell utilisait aussi des relais téléphoniques [11].
- Au sortir de la guerre, quelques grands calculateurs électromécaniques sont entrés en service, comme le Harvard Mark I / Automatic Sequence Controlled Calculator (ASCC) en 1944 [12] (sur lequel Grace Murray Hopper a fait ses débuts) suivi par le Mark II en 1947 [13] avec ses 13000 relais, toujours en coopération avec IBM.
- Bell Labs n'était pas en reste avec une série de calculateurs automatiques de plus en plus complexes [5], dont le premier (le *Complex Number Computer* avec 450 relais) a effectué le premier calcul à distance en 1940 grâce à un télécopieur situé dans une autre ville [14].
- On ne compte pas non plus les innombrables *tabulatrices* IBM, des machines conçues pour avaler les cartes perforées à toute vitesse et qui en additionnent les nombres. Destinées à la comptabilité, elles ont aussi été utilisées



Fig. 4 : Un peu de minimalisme ! Le Minivac 601 n'est pas réellement un ordinateur, mais un kit éducatif fournissant 6 lampes, 6 relais et un bouton rotatif hexadécimal motorisé. Vendu en 1961 pour 85\$, la publicité disait que le kit était né d'un projet privé de Claude E. Shannon du MIT (source : Wikipédia).

pour les calculs scientifiques, en particulier pour le projet Manhattan [15] dont des multiplicatrices IBM type 601 [16].

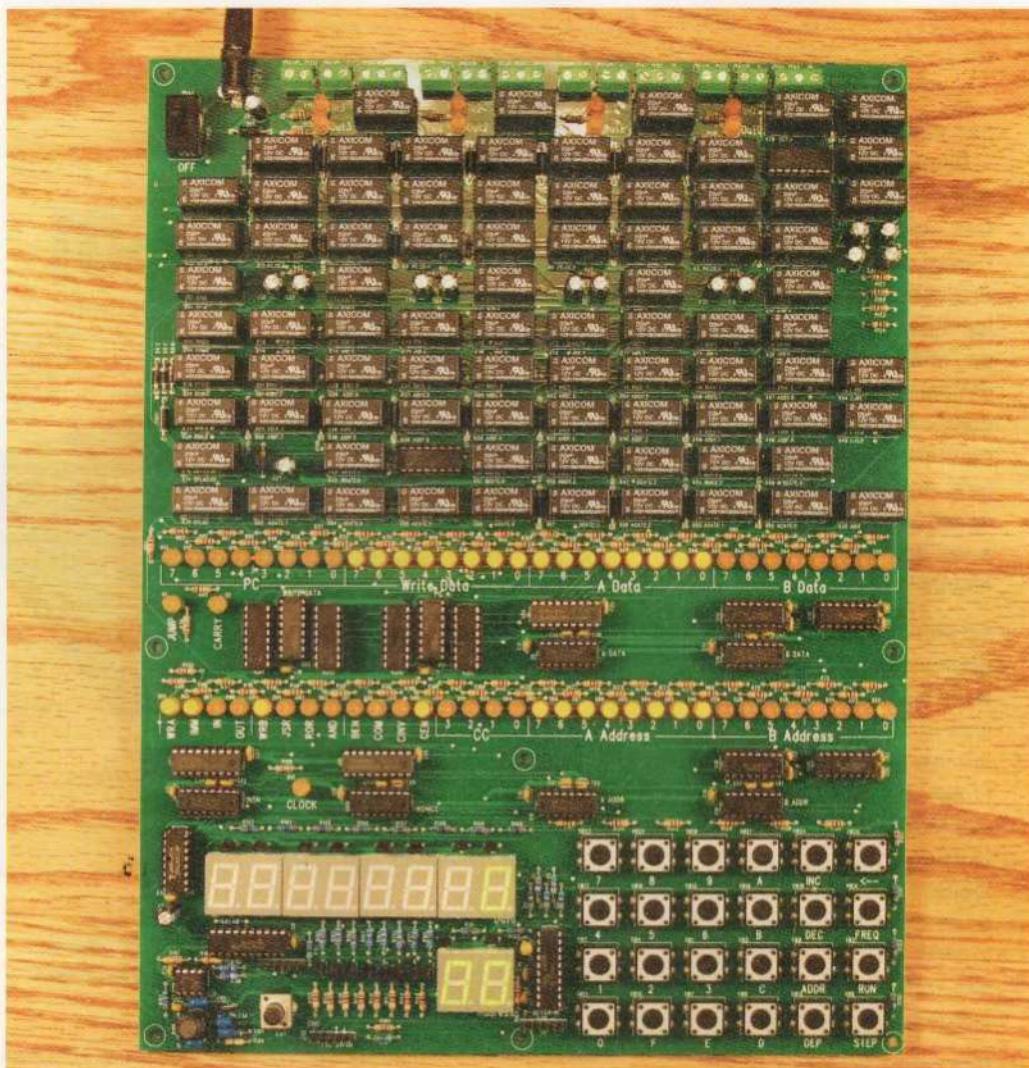
Les années 1940 sont donc l'apogée de l'ère électromécanique, dont le chant du cygne fut les énormes calculateurs à Harvard [12][13]. Même si on trouvait encore des circuits digitaux à relais dans les années 60 [17], et malgré l'invention des relais *reed* (ou ILS, Interrupteurs à Lame Souple [18]) plus rapides, on savait dès les années 30 qu'ils seraient vite détrônés par les tubes thermioniques, bien plus rapides.

Par exemple, en 1944, la machine hybride anglaise Colossus [19] [20], destinée à cryptanalyser les flux de radiotélétypes allemands, utilisait 1500 tubes pour traiter 5000 caractères par seconde. Aux États-Unis, dès 1942, Atanassoff et Berry [21] avaient déjà construit un solveur binaire d'équations linéaires en technologie hybride, avant l'ENIAC.

Cependant, la Seconde Guerre Mondiale avait redirigé toutes les ressources industrielles vers l'armée. En dehors des grands programmes militaires (comme l'ENIAC), les



Fig. 5 : Joseph Hallen a créé un circuit permettant d'explorer les systèmes logiques à base de relais. Bien qu'assisté par un microcontrôleur (qui émule la mémoire et les communications), c'est un ordinateur 8 bits minimalistre qui peut exécuter des programmes simples.



premiers calculateurs *modernes* sont souvent prototypés avec des pièces de récupération ou de surplus, « en attendant de faire mieux ». Au moins, ça a le mérite de fonctionner, même si, dans les premiers temps, les gains de vitesse ne sont pas flagrants par rapport aux *calculateurs humains* (qui étaient souvent des grandes équipes de femmes, entassées dans des bureaux, opérant nuit et jour des calculatrices mécaniques comme sur la figure 2).

Mais une fois la guerre finie, l'énorme infrastructure industrielle mise en place pour fabriquer des millions de composants militaires (pour les radars, les postes radio, et surtout des détecteurs pour les ogives d'artillerie [22]) s'est retrouvée livrée à elle-même. Le complexe militaro-industriel-académique était en place et s'est renforcé durant la guerre froide qui s'ensuivit. Les nouveaux besoins des services de renseignements, l'expertise des

ingénieurs américains, ainsi que les usines qui devaient chercher de nouveaux contrats dans le civil, permirent le développement frénétique des premiers *vrais* ordinateurs durant les années 50.

CONCLUSION

Cet article vous a proposé un petit tour d'horizon sur la riche histoire du calcul électromécanique. Je n'ai pas trouvé de référence de travaux russes, mais je ne doute pas qu'il y en ait eu, car on trouve beaucoup de relais soviétiques dans les surplus.

La fin des années 30 a vu une explosion industrielle, scientifique et créative. Les expérimentations à grande échelle ont commencé et la forme des futures machines se profilait...

J'espère que la perspective de ces idées centenaires aiguiseera votre curiosité et vous donnera envie d'explorer ce domaine ! La technologie thermionique qui lui a succédé était beaucoup plus difficile à mettre en œuvre, donc moins évidente à reconstituer sur le coin d'une table. Car même si ces technologies sont aujourd'hui considérées comme obsolètes, leur intérêt n'est pas seulement historique ! Les relais sont en effet moins intimidants pour apprendre la logique, l'électricité et les circuits de base, comme l'illustrent de nombreux projets comme celui de Joseph Allen [23]. Dernièrement, grâce surtout à la disponibilité de relais miniatures à bas prix et en grande quantité, beaucoup de personnes redécouvrent les particularités de ces composants. À une époque où l'électronique est nanométrique, ces mastodontes reprennent vie sous forme de sculptures électromécaniques... **YG**

RÉFÉRENCES

- [1] Wikipédia : « *Charles Babbage* », https://fr.wikipedia.org/wiki/Charles_Babbage
- [2] Mark Glusker : « *The ternary calculating machine of Thomas Fowler* », <http://www.mortati.com/glusker/fowler/fowlerbio.htm> (la première machine à calcul ternaire)
- [3] Wikipédia : « *Setun* », <https://en.wikipedia.org/wiki/Setun> Setun (du nom d'une rivière russe) est une famille d'ordinateurs en base ternaire, développés à partir de 1958 à l'Université d'État de Moscou.
- [4] Wikipédia : « *Ternary computer* », https://en.wikipedia.org/wiki/Ternary_computer
- [5] Christopher Felix McDonald : « *Lost Generation: The Relay Computers* », 10 mai 2017, blog : <https://technicshistory.wordpress.com/2017/05/10/lost-generation-the-relay-computers/>
« *[The relay] was conceived independently several times by natural philosophers and telegraph entrepreneurs in the 1830s.* » Page écrite en anglais, mais à lire !
- [6] Claude E. Shannon : « *A Symbolic Analysis of Relay and Switching Circuits* », Master's thesis, MIT, 1937
<http://hdl.handle.net/1721.1/11173>, https://fr.wikipedia.org/wiki/Claude_Shannon, https://en.wikipedia.org/wiki/A_Symbolic_Analysis_of_Relay_and_Switching_Circuits
- [7] Konrad Zuse : voir https://en.wikipedia.org/wiki/Konrad_Zuse et <http://zuse.zib.de/>
- [8] Wikipédia : « *Z4* », [https://en.wikipedia.org/wiki/Z4_\(computer\)](https://en.wikipedia.org/wiki/Z4_(computer))
- [9] Wikipédia : « *Alan Turing* », https://en.wikipedia.org/wiki/Alan_Turing#Bombe
« *Within weeks of arriving at Bletchley Park, Turing had specified an electromechanical machine that could help break Enigma more effectively than the Polish "bomba kryptologiczna", from which its name was derived.* »
- [10] Colin B. Burke : « *It wasn't all magic : The Early Struggle to Automate Cryptanalysis, 1930s-1960s* », United States Cryptologic History, Special Series Volume 6, DOCID : 4057009, declassified 05/29/2013
<https://www.nsa.gov/news-features/declassified-documents/cryptologic-histories/assets/files/magic.pdf>.
En anglais et très long, mais sa lecture est plus fascinante qu'un polar. Prévoyez de l'aspirine et du café.
- [11] Wikipédia : https://en.wikipedia.org/wiki/Bombe#US_Army_Bombe
« *The US Army Bombe was physically very different from the British and US Navy bombes. A contract was signed with Bell Labs on 30 September 1942. It did not use drums to represent the Enigma rotors, using instead telephone-type relays.* »



- [12] Wikipédia : « *Harvard Mark I* », https://fr.wikipedia.org/wiki/Harvard_Mark_I
« L'IBM ASCC (Automatic Sequence Controlled Calculator), appelé le *Mark I* par l'Université Harvard, a été le premier grand calculateur [...] à virgule flottante construit aux États-Unis et livré en 1944. [...] Il pouvait faire trois additions ou soustractions par seconde. »
- [13] Wikipédia : « *Harvard Mark II* », https://en.wikipedia.org/wiki/Harvard_Mark_II
Le Harvard Mark II était un ordinateur à base de relais terminé en 1947. 2 à 8 fois plus rapide que le Mark I, il tournait à 8Hz.
- [14] This Day in Tech History (Nick ?) : « *Kitchen Table Computer* », 11 septembre 2013
<https://thedayintech.wordpress.com/2013/09/11/kitchen-table-computer/>
- [15] Richard Feynman : « *Los Alamos from Below* », p.25, http://www.f.waseda.jp/sidoli/Feynman_2003.pdf (retranscription d'une conférence qu'on peut écouter à <https://youtu.be/uY-u1qyRM5w?t=2896>)
- [16] Ken Shirriff : « *Identifying the "Early IBM Computer" in a Twitter photo: a 405 Accounting Machine* »
<http://www.righto.com/2017/11/identifying-early-ibm-computer-in.html>, note 6 : « *The use of IBM accounting machine by the Manhattan Project is described by Feynman in Los Alamos from Below, p25; and in this page. The Manhattan Project used a [model] 402 accounting machine, several IBM 601 multiplying punches for multiplication, and other card equipment* »
Voir aussi : Christina Kohl : « *The Role of Computer Science during the Manhattan Project* », 10 juin 2015, <http://cl-informatik.uibk.ac.at/teaching/ss15/ewa/reports/ss15-CKo.pdf>
- [17] Le kit Minivac 601 : https://en.wikipedia.org/wiki/Minivac_601 http://www.oldcomputermuseum.com/minivac_601.html
- [18] Wikipédia : « *Interrupteur Reed* » (ou *ILS*), https://fr.wikipedia.org/wiki/Interrupteur_reed
- [19] Computerphile : « *Colossus & Bletchley Park* », 28 mai 2015, <https://www.youtube.com/watch?v=9HH-asvLAj4>
- [20] « *IEEE Computer: Alan Turing at Bletchley Park* » https://youtu.be/5nK_ft0Lf1s?t=1310
Description et visite de la machine hybride Colossus reconstituée à Bletchley Park. Elle contient 1500 tubes thermioniques et beaucoup de composants typiques de la technologie téléphonique britannique d'avant-guerre. Par chance, l'effort de reconstitution a coïncidé avec le remplacement de cette technologie dans le pays, ce qui a fourni des composants d'époque à peu de frais.
« *[Tommy Flowers was] An engineer that was used to designing systems for telephone switching. So a lot of the components here, everything from the relay banks, to the switches, to the power supplies, were common to British Post Offices in the pre-war period. So when he then wants to find the components to rebuild the replica, he's lucky in the sense that the last of those exchanges are being decommissioned from Post Offices around the country.* »
- [21] Wikipédia : « *ABC, Atanasoff-Berry Computer* » https://fr.wikipedia.org/wiki/Atanasoff-Berry_Computer
- [22] « *Fusées de proximité* » : <https://www.youtube.com/watch?v=sEN4HoXOqGw> ainsi que https://fr.wikipedia.org/wiki/Fusée_de_proximité :
« *En 1944, une grande partie de l'industrie électronique américaine fabriquait les fusées de proximité. Les contrats passèrent de 60 millions USD en 1942 à 200 millions USD en 1943, puis à 300 millions USD en 1944 et atteignirent 450 millions USD en 1945. Alors que les quantités fabriquées augmentaient, le prix de revient des fusées diminua pour des raisons d'efficacité : il passa de 732 USD en 1942 à 18 USD en 1945. Cette année-là, il se vendit plus de 22 millions de fusées pour une somme approximative de 1 010 millions USD.* »
- [23] Hallen, J. « *Single Board Relay Computer* » : <http://relaysbc.sourceforge.net/circuits.html> <https://www.tindie.com/products/jhallen/single-board-relay-computer/> <https://www.youtube.com/watch?v=k1hJoalcK68>

LE RÉSEAU MINIATURE

Électronique



LR PRESSE

REVUE DE LIVRE : LE RÉSEAU MINIATURE, PAR BURKHARD OERTHEL

Yann Guidon ["C'est tellement facile !"]

Lors d'une visite dans un magasin de modélisme ferroviaire, rue de Douai, j'ai découvert un petit livre qui m'a immédiatement séduit et que je souhaite faire découvrir hors du petit cercle des modélistes. Bien qu'il ne soit pas destiné au monde du DIY, il contient la plupart des techniques importantes que le bricoleur se doit de connaître.

C'est un livre « ancien », publié initialement en 1983 dans sa langue d'origine : l'allemand. Il traite donc surtout de la technologie des années 70, mais les 35 ans de rééditions ont amélioré, poli et lissé ce pamphlet. Il est donc à la fois actuel et éprouvé, même s'il n'aborde pas la programmation, les microcontrôleurs, les sketchs d'Arduino ou les autres Raspberry Pi (bref, ce que vous trouverez dans ce magazine).

Ce n'est pas une bible, ou une référence incontournable que je vous propose de découvrir ici. C'est un petit livre d'introduction presque anodin, discret, un petit condensé de 116 pages seulement qui ne vous promet pas de devenir un ingénieur confirmé. Il vous invite juste à découvrir l'électronique dite « de base », à faire le tour des techniques essentielles que vous pourrez approfondir à loisir en fonction des besoins.

Il présente les principes de base et propose de nombreux petits circuits utiles et réalisables rapidement. Il montre qu'on peut passer rapidement à l'action, sans éclipser les contraintes techniques.

L'intérêt de découvrir l'électronique sous l'angle du modélisme ferroviaire est de bien définir un domaine d'application et d'éviter l'écueil des livres plus généralistes, qui tentent de tout expliquer et qui peuvent devenir intimidants, voire vagues ou obscurs.

LES THÈMES ABORDÉS :

- Les principes du courant électrique
- Les composants de base : résistances, condensateurs, diodes, transistors
- La commande de puissance pour alimenter des moteurs électriques
- Les circuits logiques, les bascules et les oscillateurs
- Les circuits intégrés (TTL et CMOS, analogiques, puissance)
- Les capteurs
- Les techniques pratiques de montage

Après un petit tour des composants discrets, on retrouve les circuits intégrés incontournables : le NE555, le CD4017, le LM317, le 7812, le 741...

Le texte est clair et facile à lire, mais la traduction n'a probablement pas été écrite ou relue par un électronicien. Certains termes ne correspondent pas aux usages du domaine, par exemple « résistance modulable » au lieu de « résistance ajustable ». Malgré cela, et même s'il ne vous apprend rien, achetez-le quand même. Il ne vous en coûtera que 11,50€ et sera un bon cadeau pour les bricoleurs et bricoleuses débutant(e)s qui s'imaginent encore que de longues études sont nécessaires pour construire des circuits simples et utiles. Il aidera certain(e)s à mettre le pied à l'étrier et incitera les plus curieux (quel que soit leur âge) à chercher, expérimenter et explorer sans peur ce monde fabuleux. ■ YG

En vente en magasin de modélisme, sur Amazon, neuf et d'occasion, ou en ligne à : <https://trains.lrpresse.com/A-15453-le-reseau-miniature-electronique.aspx>

COLLECTION :

Le réseau miniature

TITRE :

Électronique

AUTEUR :

Burkhard OERTHEL

ÉDITEUR :

LR PRESSE SARL

DIMENSIONS :

20×16 cm

NOMBRE DE PAGES :

116

ISBN :

2-903651-44-2

ISBN-13 :

978-2903651442



LES CAPTEURS PHOTOVOLTAÏQUES

Paris Jannick



On a déjà tous eu en tête ces petits projets qui ne consomment pas beaucoup d'énergie et qu'il serait sympa d'alimenter via une cellule photovoltaïque. Puis en cherchant un capteur approprié, on croise des termes comme : monocristallin, mppt, etc. Nous allons voir cela en détail, ce qu'ils signifient et ce qu'ils impliquent techniquement.

Contrairement à ce qu'on pourrait croire, il ne suffit pas de brancher un capteur photovoltaïque sur une batterie pour que cela fonctionne. Un tel capteur est bien différent d'une pile, voyons ça en détail.

1. UN PEU DE PHYSIQUE

Pour bien comprendre les différents modèles de cellules photovoltaïques ainsi que leurs différentes applications, nous allons commencer par voir ce qu'il se passe à l'intérieur d'une diode. Vous avez certainement déjà vu/manipulé ces petits composants qui ne laissent passer le courant électrique que dans un seul sens.

Une diode est un petit morceau de silicium ultra pur (plus de 99,999 %), mais le silicium n'est pas un bon conducteur c'est pour cela qu'il faut le doper ! Pas avec de l'EPO, mais avec des atomes qui lui sont voisins dans le tableau périodique. Le plus souvent, il s'agit de bore et de phosphore.

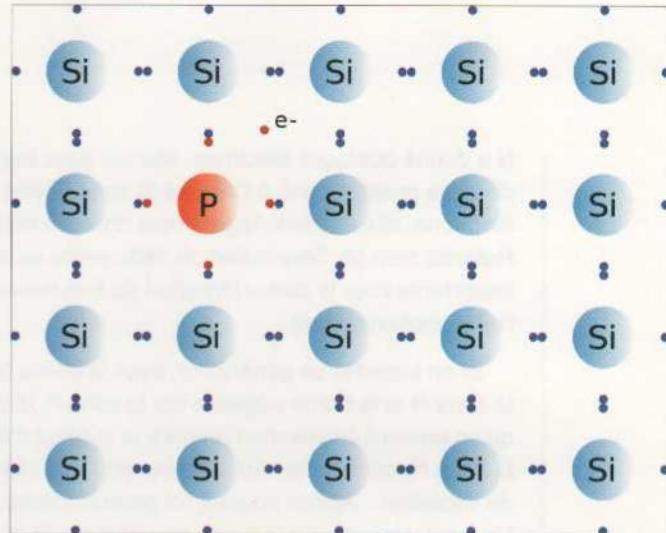
1.1 Le dopage

Si on ajoute du phosphore dans un cristal de silicium, on va perturber la structure électronique du cristal, car le phosphore possède, sur sa couche externe, un électron de plus que le silicium. En faisant ceci, on ajoute donc des porteurs de charge, or c'est justement quand des porteurs de charge sont disponibles qu'on peut faire circuler un courant électrique. On parle de dopage N.

À l'inverse, si on ajoute du bore dans un cristal de silicium, on va créer des trous dans la structure, car le bore possède, sur sa couche externe, un électron de moins que le silicium. Chose surprenante, ce trou va permettre également de conduire le courant ! Car il va laisser de la place pour un électron d'un atome voisin, au final, ce trou va se comporter comme une charge positive qui se déplace. On parle de dopage P.

1.2 Jonction PN

Imaginons qu'on dope un côté d'un cristal de silicium avec du phosphore et l'autre avec du bore. On va donc avoir un côté avec un surplus d'élec-



trons et l'autre côté au contraire aura des trous, un manque d'électrons. Les électrons du côté N vont chercher à combler les trous du côté P. Il va apparaître au milieu une zone dans laquelle les électrons auront comblé les trous, c'est-à-dire une zone dans laquelle il n'y aura plus de porteur de charge et s'il n'y a plus de porteur de charge, alors cette zone devient isolante. On l'appelle la zone de déplétion. Dans cette zone de déplétion, des charges partielles se sont formées, la zone

Fig. 1 : Représentation d'un cristal de silicium dopé N (source Wikipédia - Guillom - CC BY-SA 3.0).

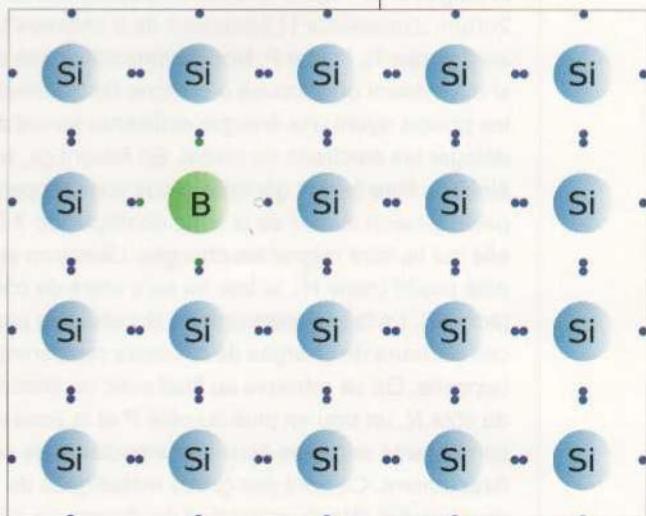


Fig. 2 : Représentation d'un cristal de silicium dopé P (source Wikipédia - Guillom - CC BY-SA 3.0).



N a donné quelques électrons, elle est donc légèrement chargée positivement, à l'inverse la zone P, elle, a reçu des électrons, elle est donc légèrement chargée négativement. Retenez bien ça, l'apparition de cette petite tension est importante pour la compréhension du fonctionnement de l'effet photovoltaïque.

Si on branche un générateur, avec la borne positive sur la zone N et la borne négative sur la zone P, la seule chose qu'on arrivera à faire c'est prendre le surplus d'électrons de la zone N, combler les trous de la zone P et élargir la zone de déplétion... Aucun courant ne pourra circuler. Si on fait l'inverse, on branche la borne négative sur la zone N, la borne positive sur la zone P. Alors on ajoute des électrons là où il y en a déjà un surplus, les forçant à franchir la zone de déplétion pour rejoindre les trous qui se trouvent de l'autre côté, qui eux sont vidés par la borne positive du générateur. On se retrouve donc dans une situation dans laquelle il y a toujours des porteurs de charges libres, le courant peut s'établir.

Petite remarque, il faut effectivement forcer les électrons à franchir cette zone de déplétion, c'est la fameuse chute de tension de 0,7V qui se produit à travers la diode.

2. L'EFFET PHOTOVOLTAÏQUE

Enfin, nous attaquons le vif du sujet, vous allez voir que finalement un capteur photovoltaïque n'est qu'une diode un peu particulière. Il faut imaginer une plaque de silicium, de la longueur et largeur que vous voulez, mais de seulement 200µm d'épaisseur ! L'épaisseur de 2 cheveux ! Une face sera dopée N, l'autre P. Nous avons donc bien une diode, si maintenant on s'amuse à envoyer de la lumière dessus, les photos ayant une énergie suffisante seront capables de déloger les électrons du cristal. En faisant ça, un trou et un électron libre seront générés. Vous vous rappelez de cette petite tension autour de la zone de déplétion ? Eh bien, c'est elle qui va faire migrer les charges. L'électron sera attiré du côté positif (zone N), le trou lui sera attiré du côté négatif (zone P). La faible épaisseur de la cellule va permettre à ces porteurs de charges de rejoindre rapidement les côtés opposés. On se retrouve au final avec un électron en plus du côté N, un trou en plus du côté P et la zone de déplétion isolante entre les deux leur interdisant de se rejoindre directement. Ce sont des grilles métalliques de surface qui vont jouer le rôle de collecteur de charge, ce sont elles les

bornes du capteur photovoltaïque. Les électrons n'auront pas d'autres choix que de passer par ces grilles et le dispositif que vous aurez branché aux bornes de la cellule pour enfin arriver à combler les trous de l'autre côté. Voilà, nous avons bien un dispositif capable de convertir de la lumière en énergie électrique.

Petit bémol cependant, l'énergie qu'il faut pour déloger un électron d'un cristal de silicium correspond à l'énergie que transporte un photon de lumière infrarouge (environ 1µm de longueur d'onde). Si on utilise des photons plus énergétiques, cela fonctionnera quand même, mais le surplus d'énergie sera converti en chaleur.

2.1 Cristallisation

Jusqu'à présent, je n'ai parlé que de cristal de silicium, c'est-à-dire des atomes de silicium rangés bien régulièrement dans l'espace. Or techniquement, obtenir un cristal de plusieurs centimètres de long est une chose assez compliquée. C'est pourquoi les fabricants de cellules solaires proposent 3 types de cellules.

Monocristallin : ce qui signifie que d'un bout à l'autre de la cellule les atomes de silicium sont rangés de la même façon et forment un tout. L'avantage est que le rendement est meilleur (12 à 20%), mais le prix est plus élevé et nécessite une intensité lumineuse plus importante que les deux autres pour « démarrer ». La réponse maximale pour ce genre de capteur est obtenue pour un rayonnement ayant une longueur d'onde aux alentours de 1µm, donc dans l'infrarouge.

Amorphe : là il n'est pas question de cristal, les atomes de silicium sont

disposés n'importe comment, ce qui est techniquement très facile à faire, donc le prix est beaucoup plus faible, cependant le rendement également (5 à 9%), mais il réagit même lorsqu'il est soumis à une lumière diffuse. Pour ce capteur, la réponse maximale est pour un rayonnement ayant une longueur d'onde d'environ $0,6\mu\text{m}$ (600nm), ce qui correspond à la couleur rouge.

Polycristallin : ce type de capteur est à mi-chemin entre les deux précédents, puisqu'il est composé d'une multitude de petits cristaux (généralement d'une taille de l'ordre du centimètre). Les cristaux lui confèrent les mêmes propriétés que le monocristallin, cependant les jonctions entre les cristaux peuvent être considérées comme du silicium amorphe. Le rendement est compris entre 11 et 15 %.

2.2 Effet de la température

Si vous deviez classer la cellule photovoltaïque, dans quelle catégorie la mettriez-vous ? Composant électronique ? Vous n'auriez pas tort, mais on pourrait également la ranger avec les machines thermiques. Oui oui, comme les moteurs à explosion ! Il va falloir étudier différents aspects du capteur avant de tous les rassembler pour en avoir une vue globale et voir que ce genre de capteur est réellement une machine thermique.

Comme on l'a vu précédemment, ce sont les photons infrarouges qui permettent de faire

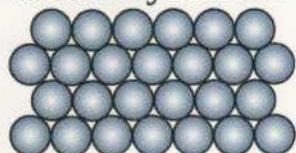
fonctionner le capteur, or ce type de rayonnement intervient quand on chauffe un corps. On peut donc dire que plus la source de lumière sera chaude, plus la réponse du capteur sera importante (en courant). Car il y aura plus de photons, donc plus d'électrons délogés.

L'autre caractéristique importante est la température du capteur. Si elle est trop importante, alors l'énergie thermique sera suffisante pour que les électrons délogés soient capables de franchir la barrière de potentiel de la zone de déplétion et ainsi générer une recombinaison électron-trou. Pour faire simple, plus la température du capteur est élevée, moins les électrons vont pousser à travers le dispositif branché aux bornes du capteur, préférant une recombinaison au sein même du silicium. Pour être efficace, il faut donc maintenir le capteur à bonne température, il faut pour cela une source froide.

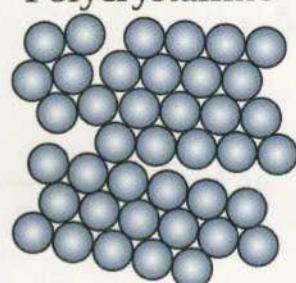
Et bien voilà, on vient de voir que pour qu'un capteur photovoltaïque fonctionne correctement, il doit être coincé entre une source chaude et une source froide. Il s'agit donc bien d'une machine thermique, au même titre qu'un moteur à explosion. On peut donc se référer aux lois de la thermodynamique pour chercher à optimiser son rendement. Loin de moi l'idée de vous faire un cours de thermodynamique, cette discipline qui donne des sueurs froides à bon nombre d'étudiants. Pour un capteur photovoltaïque, il n'y aura qu'une chose à retenir : le rendement d'une machine thermique est toujours proportionnel à la différence de température entre la source chaude et la source froide.

Dans les faits, on remarque que le courant de court-circuit d'un capteur photovoltaïque est proportionnel à l'intensité lumineuse. La tension à vide quant à elle augmente quand la température diminue (environ $133\text{mV}/^\circ\text{C}$) et entraîne donc environ une augmentation du rendement de $0,5\%/\text{ }^\circ\text{C}$.

Monocrystalline



Polycrystalline



Amorphous

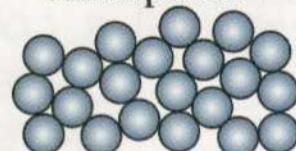


Fig. 3 : Différentes formes de silicium (source Wikipédia - Sbymes321 - CC BY-SA 3.0).

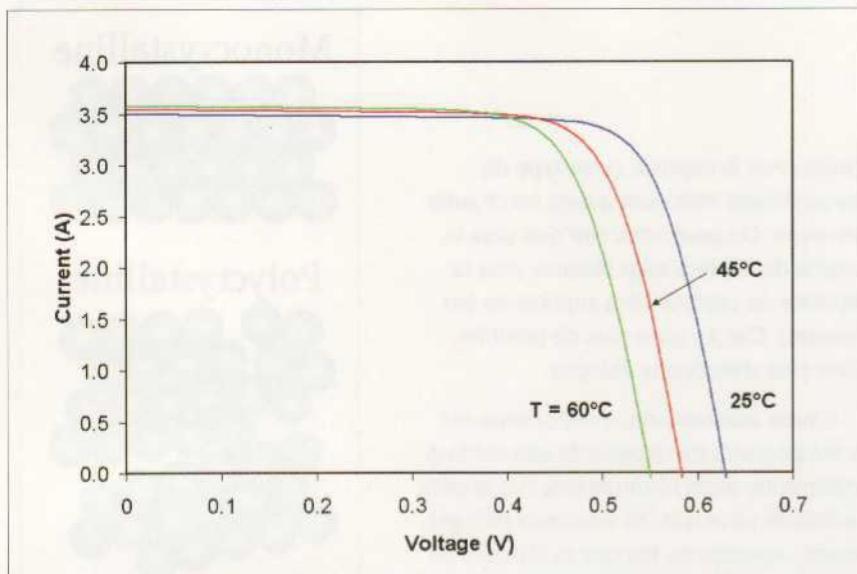


Fig. 3 : Courbes caractéristiques d'une cellule photovoltaïque pour plusieurs températures (source Wikipédia - Squirmymcphée - CC BY-SA 3.0).

3. COURBE CARACTÉRISTIQUE (INTENSITÉ EN FONCTION DE LA TENSION)

C'est maintenant que commence vraiment la prise de tête, car là où un générateur « classique » (pile, batterie, alimentation de PC, etc.) réagit de manière linéaire, un capteur photovoltaïque n'a clairement pas une réponse linéaire...

La première chose que l'on remarque c'est qu'au-delà d'une certaine valeur de tension, l'intensité s'effondre... littéralement !

3.1 Gestion de la puissance PWM (Pulse Width Modulation)

Une manière de gérer cette courbe non linéaire est de choisir un capteur photovoltaïque, ou d'associer des capteurs, pour fournir une tension supérieure à la tension nécessaire, de cette manière on s'assure d'avoir toujours un courant de sortie non nul. La gestion de la puissance se fera simplement en ajustant la tension de sortie grâce à un système PWM, en français : modulation de largeur d'impulsion.

Pour faire simple, on peut réduire ce genre de système à un condensateur en série avec un interrupteur. L'algorithme est très simple, on fixe une tension de consigne, si la tension aux bornes du condensateur est inférieure alors on ferme l'interrupteur, si au contraire la tension est supérieure alors on ouvre l'interrupteur.

Très simple et très facile ce système permet d'utiliser correctement un capteur solaire. Cependant, il n'est clairement pas optimum, car il « coupe » la courbe à une certaine tension. De ce fait, on ne peut pas accéder à la partie de la courbe contenant le point de puissance maximale. En général, le rendement est compris entre 70 et 85 %.

3.2 Gestion de la puissance MPPT (Maximum Power Point Tracker)

Pour optimiser l'utilisation d'un panneau solaire et en tirer le maximum de puissance possible, il faut à la fois maximiser la tension ET l'intensité. Pour ce faire, il existe plusieurs algorithmes : P&O (perturber et observer), incrément de la conductance (INC), mesure d'une fraction de courant de court-circuit (FCC), mesure d'une fraction de tension de circuit ouvert (FCO) ainsi que l'utilisation de la logique floue (LF). Pour éviter de vous noyer sous des pages d'explications, je ne vais détailler que le plus simple, mais cependant très efficace, l'algorithme P&O.

Pour que ces algorithmes puissent fonctionner, il faut un dispositif permettant d'adapter l'impédance entre le panneau et la batterie. Ce rôle sera joué par un convertisseur boost. Sans trop rentrer dans les détails, il s'agit d'un petit circuit électronique capable de générer une tension plus élevée. Le but est alors de jouer sur le transistor de ce circuit afin de trouver la tension de sortie qui correspond à la puissance maximale du panneau.

L'algorithme P&O est très simple, on joue sur le rapport cyclique du circuit boost, on mesure la puissance fournie par le capteur, si on constate une augmentation par rapport à la mesure précédente, alors on est sur la bonne voie et on continue à modifier le rapport cyclique dans ce sens. Dans le cas contraire, cela signifie qu'on s'éloigne du point de puissance maximale et qu'il faut donc faire varier le rapport cyclique dans l'autre sens.

Ce système est plus complexe que le PWM, mais permet d'atteindre un rendement de l'ordre de 95 %.

3.3 Petit récapitulatif

Pour finir, voici quelques points importants à prendre en compte lors de l'utilisation de capteurs photovoltaïques :

- choisissez le capteur qui correspond à vos besoins, s'il sera en plein soleil optez pour un monocristallin, s'il ne sera exposé qu'à une faible intensité ou à une lumière diffuse, préférez un amorphe ;
- l'intensité électrique fournie est proportionnelle à l'intensité lumineuse reçue, il faut donc que les rayons lumineux frappent votre capteur de manière la plus perpendiculaire possible et que votre capteur soit bien propre ;
- la plupart des capteurs fonctionnent grâce aux infrarouges, ne placez pas de vitre devant, car le verre absorbe beaucoup dans ce spectre-là ;

- la tension de sortie chute avec l'augmentation de la température, laissez-les respirer, un petit flux d'air ne peut pas leur faire de mal, si vos capteurs sont destinés à rester en plein soleil et qu'ils sont d'assez petite taille, pourquoi ne pas trouver un vieux radiateur en alu (récupéré dans un vieux PC) et lui coller dessus ?
- la courbe courant-tension n'est pas du tout linéaire, brancher votre montage directement sur une cellule photovoltaïque est une mauvaise idée. Utilisez un convertisseur DC-DC pour charger une batterie ainsi que pour pouvoir extraire le maximum de puissance du capteur.

Après ce petit tour d'horizon, j'espère que vous aurez une idée plus claire du fonctionnement de ces capteurs et de comment les utiliser. Ça m'étonne toujours de voir ces petits objets du quotidien, qui nous paraissent si simples, si anodins, mais qui pourtant sont d'une complexité folle. Gratter la surface pour comprendre ce qui se cache en dessous est toujours très enrichissant, souvent prise de tête, mais très enrichissant ! **PJ**

RÉFÉRENCES

- Dopage des semi-conducteurs :
[https://fr.wikipedia.org/wiki/Dopage_\(semi-conducteur\)](https://fr.wikipedia.org/wiki/Dopage_(semi-conducteur))
- Jonction P-N : https://fr.wikipedia.org/wiki/Jonction_P-N
- Généralité sur les cellules photovoltaïques :
https://fr.wikipedia.org/wiki/Cellule_photovolta%C3%A9que
- Effet de la température :
http://photovoltaque.guidenr.fr/III_3_effet-temperature-module-photovoltaque.php
https://en.wikipedia.org/wiki/Theory_of_solar_cells#Cell_temperature
- Cristallisation :
https://en.wikipedia.org/wiki/Crystalline_silicon#Mono-silicon
- Réponse spectrale :
<http://www.photovoltaque.info/L-effet-photovoltaque.html>
- Courbe caractéristique :
<https://www.energieplus-lesite.be/index.php?id=16696>
- Convertisseur boost :
https://fr.wikipedia.org/wiki/Convertisseur_Boost
- Article sur les MPPT :
https://www.cder.dz/download/Art17-3_9.pdf



SCSI2SD OU COMMENT REEMPLACER UN DISQUE SCSI PAR UNE CARTE SD

Denis Bodor



Les supports de stockage magnéto-mécaniques et les équipements associés comme les cassettes, les disquettes et les disques durs ont une durée de vie limitée. Non seulement leur capacité de rétention de l'information se dégrade au fil du temps, mais ce sont également les parties mécaniques qui s'usent et finissent par lâcher. À cela s'ajoute, de plus, l'évolution de la technologie elle-même qui rend obsolètes ces supports et conduit à un arrêt de leur production et donc, inéluctablement, à leur raréfaction. Ainsi, simplement remplacer un disque dur dans une machine ancienne n'est souvent pas l'idée la plus judicieuse, ni la plus économique.

Nous nous intéresserons ici principalement aux disques durs et en particulier ceux utilisant une interface SCSI. Ceci nous place donc dans une tranche de temps vers la fin des années 80 et dans une catégorie de machines destinées à supporter un stockage interne performant. À cette époque et avec l'arrivée de l'IBM PC XT équipé d'un gigantesque disque dur de 10 Mo, la notion de standard informatique n'était pas celle que l'on connaît aujourd'hui.

L'XT et ses clones utilisaient une technologie et une interface pour les disques durs que beaucoup (dont moi) désignaient sous l'acronyme MFM. Il ne s'agissait en réalité absolument pas d'un standard, mais d'un système d'encodage des données (*Modified Frequency Modulation*), également utilisé pour les disquettes. Le fait est que le « modèle » IBM utilisait un disque et une interface développés par Seagate (alors Shugart Technology) appelés ST-412, qui utilisait précisément cette technologie, et le terme s'est popularisé.

ST-412 est une variante de ST-506 (disque et interface) et l'ensemble a rapidement été adopté par les autres constructeurs. C'est ainsi que cela se passait à l'époque, IBM donnait le ton et les autres suivaient la musique. C'est ce qu'on appelle un standard de facto. Il s'est passé exactement la même chose avec le successeur de l'XT, l'IBM PC/AT, son processeur 80286 et son interface ATA

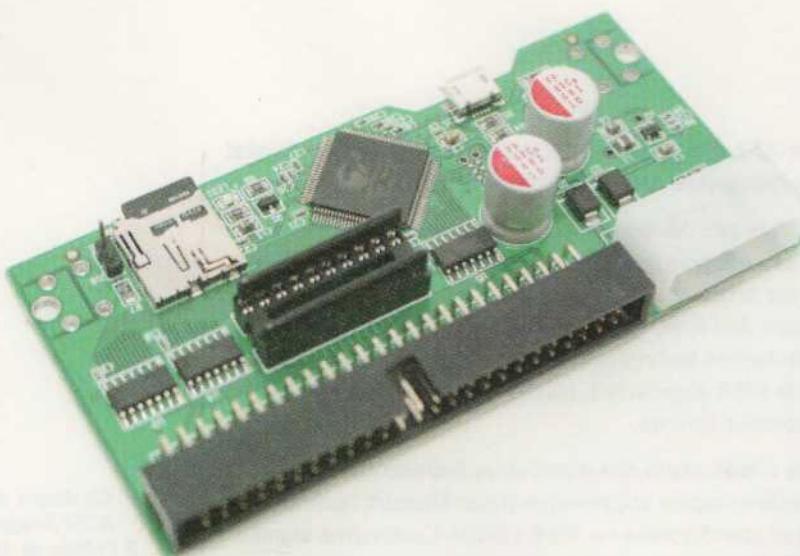
(pour *AT bus Attachment*) développée par Western Digital sous l'acronyme IDE pour *Integrated Drive Electronics*.

ATA et ses déclinaisons parallèles sont intimement liés au bus de l'époque, ISA, et une rétrocompatibilité imposée par le marché (et la peur de la plupart des acteurs majeurs de l'époque), ce qui impliqua un certain nombre de limitations technologiques. Sans entrer dans le détail, plus le débit augmentait, plus le bus était chargé et le processeur sollicité.

Ce n'était pas le cas d'une autre technologie, également développée originellement par Shugart Technology, et standardisée en 1986 : SCSI. L'acronyme signifie *Small Computer System Interface* et se prononce normalement « scuzzy », et non « haisséhaissé » même si cela était très courant à l'époque dans les pays francophones. SCSI ne se limite pas aux disques durs, ni aux machines « exotiques ». Il permet d'interfacer également des lecteurs optiques, des unités de sauvegarde, des scanners et même des imprimantes. C'est un bus plus générique, plus performant, plus intelligent (moins demandeur de l'attention du processeur) et... plus cher. On remarquera avec amusement le « Small » (« petit ») de l'acronyme et à quel point cette notion est toute relative, en particulier au regard du matériel utilisé actuellement...

Ce disque dur SCSI Seagate ST1480n de 426 Mo est la raison pour laquelle je me suis penché sur le SCSI2SD. Après des années de bons et loyaux services, il a commencé à rendre l'âme, menaçant d'emporter avec lui d'autres éléments de la machine où il se trouvait : une NeXTstation Turbo de 1992.





Il existe plusieurs déclinaisons et modèles du SCSI2SD. Celle-ci est une version 5.0a fabriquée et vendue sous licence par iTead. Moins coûteuse que la plus récente V6, c'est également celle qui est compatible avec le plus grand nombre d'ordinateurs et d'échantillonneurs vintage.

Dans les années 90, les machines utilisant SCSI en standard sont le plus souvent des stations de travail (NeXT, SGI, Sun, etc.), les machines Apple haut de gamme, mais aussi les PC. Je me souviens distinctement qu'à l'époque, « SCSI » était synonyme de « performance » dans ce domaine, à la fois pour les accès disque, mais également pour le nombre de périphériques qu'il était possible de connecter : 7 disques par interface, formant une chaîne où chacun d'eux devait avoir un ID entre 0 et 7 (0 et 15 avec le Wide SCSI). À la même époque, les PC avec interface IDE ne pouvaient accepter que 4 disques, 2 par bus, un maître et un esclave (cela vous rappelle des souvenirs ?).

Dans les grandes lignes, SCSI se divise en trois déclinaisons historiques : SCSI-1 à SCSI-3. Alors que les deux premières sont relativement similaires en termes de connectique, avec un câble nappe de 50 broches (IDC50) ou un câble Amphenol C50 ressemblant à un connecteur pour imprimante parallèle (également appelé Centronics SCSI ou CN-50), SCSI-3 se caractérise par un connecteur de 68 broches. SCSI-3 est également connu sous le nom de Ultra SCSI. Post SCSI-3, on parle généralement de Ultra2, Ultra-3, Ultra-320, Ultra-640, etc., référençant ainsi soit la génération, soit le débit en mégaoctets par seconde. SCSI évolua, plus tard en SAS (*Serial Attached SCSI*), pendant série du SCSI parallèle comme l'est le SATA (*Serial ATA*) pour l'ATA/IDE parallèle.

À noter qu'Apple en ce temps avait déjà cette tendance quasi-obsessionnelle à ne pas faire comme tout le monde et proposait donc, sur ses ordinateurs, un

connecteur SCSI au format DB25 (similaire à un port parallèle). Un certain nombre de constructeurs ont alors tout naturellement suivi le mouvement en proposant des périphériques (disque externe par exemple) avec ce type de connecteur. Enfin, il est également important de préciser qu'un format Micro DB50, version compressée du CN-50, était également utilisé, en particulier vers la fin de l'ère SCSI-2, et ne doit pas être confondu avec le MD68, identique, mais comportant 68 broches.

Nous avons donc des générations de périphériques qui se divisent entre « l'avant SCSI-2 » et « l'après SCSI-2 », tout en ayant une grande compatibilité entre les standards. Rien ne vous empêche de connecter un disque SCSI-2 à un contrôleur Ultra SCSI de toute dernière génération et d'accéder aux données, sous réserve d'adapter les connecteurs, de respecter la longueur maximum des câbles et, bien entendu, de vous satisfaire du débit correspondant. Les adaptateurs permettant presque toutes les combinaisons sont disponibles et relativement faciles à trouver, encore aujourd'hui.

1. LE PROBLÈME 25 ANS APRÈS

Les disques durs IDE (ou E-IDE) comme SCSI et SCSI-2 ne sont plus fabriqués depuis longtemps. Après une période suivant l'arrivée du SATA début 2000 où la transition faisait baisser les prix, tout allait encore pour le mieux. Il n'était pas difficile de trouver des

disques provenant de fins de stock ou de machines peu utilisées, mais le matériel s'usait petit à petit...

Les possesseurs de machines fonctionnant avec des disques IDE (PC, Amiga 1200, Macintosh Quadra, iMac, etc.) avaient d'ores et déjà une parade : les cartes CompactFlash. En effet, ce support de stockage possède un contrôleur ATA et dispose d'un mode IDE permettant à une machine de le voir, tout simplement, comme un disque dur. Inutile donc d'investir dans un matériel très coûteux pour remplacer un disque par une carte CompactFlash. Tout ce qu'il vous faut c'est un adaptateur sans le moindre composant électrique qui vous coûtera trois francs six sous (voir article sur FreeDOS dans *Hackable n°18*). En résumé, une carte CompactFlash peut fonctionner comme un disque dur IDE pour peu que l'on adapte simplement la connectique.

Pour le SCSI en revanche, il en va tout autrement, car il n'existe pas d'astuce similaire. On est donc généralement obligé de se tourner vers le matériel d'origine et ceci soulève un certain nombre de problèmes, en particulier avec les disques durs. En effet, il est très facile de dénicher un disque Ultra SCSI d'une capacité d'une dizaine de Go ou plus, pour une vingtaine d'euros. Plus vous avancez en génération, plus les capacités augmentent, plus le coût baisse, mais plus les problèmes de compatibilité se font jour avec les machines, qu'il s'agisse d'un vieux Mac ou d'une station de travail Silicon Graphics, NeXT, Sun ou Digital Equipment Corp.

Il faut donc impérativement s'orienter vers des disques contemporains aux machines en question et là, un problème se pose, car il y a une véritable pénurie. Une simple recherche pour un disque SCSI ou SCSI-2 (non Ultra) sur eBay, remonte des résultats très insatisfaisants. Non seulement beaucoup d'annonces concernent du matériel SCSI beaucoup trop récent, mais l'équipement adéquat, une fois enfin trouvé, est vendu à des tarifs tout à fait prohibitifs dépassant allègrement les 200€. Bien entendu, qui dit « rareté » dit bien souvent « arnaque » et un grand nombre d'annonces proposent du matériel, certes compatible, mais déjà largement utilisé (souvent dans des configurations serveurs) et décrit comme « Vintage, non testé, voir photo ». Voici quelques exemples :

- Fujitsu M2949SYU 9100 Mo : 230€,
- HP Quantum Atlas 9 Go : 445€,
- Digital RZ28 2 Go (testé) : 245€,
- HP A4909A 18 Go : 200€,
- Seagate ST11200N 1,2 Go : 230€.

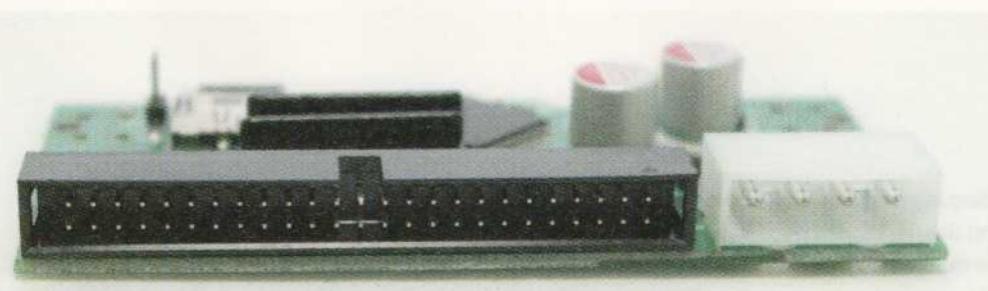
Je suis même tombé sur une annonce proposant un disque SCSI-2 SSD de 4 Go à presque 300€ n'étant en réalité rien d'autre qu'une jolie boîte contenant l'adaptateur dont il est question dans cet article, couplé à une carte SD de 4G.

Quand bien même vous décideriez de jeter une telle somme d'argent par la fenêtre, vous recevrez finalement un matériel usé et donc susceptible de devenir défectueux sans doute aussi rapidement que le disque déjà présent dans votre ordinateur vintage. Il existe une règle implicite lorsqu'on parle de disque dur, celle consistant à ne surtout pas se demander s'il va vous lâcher, mais simplement *quand* il va vous lâcher. Le remplacement d'une pièce ancienne par une autre pièce ancienne n'est donc pas une solution. Ceci reviendrait à changer les pneus de votre voiture de collection en les remplaçant par des pneus lisses d'un autre véhicule ayant déjà des dizaines de milliers de kilomètres au compteur...

La solution idéale pour redonner un « coup de jeune » à une machine vintage consisterait à lui fournir un support de stockage neuf, de préférence plus durable que l'original et plus économique. Le problème cependant réside dans le fait qu'il n'existe pas de solution équivalente à l'astuce de la CompactFlash pour le SCSI.



À première vue, on pourrait penser qu'il s'agit ici d'une interface IDE, mais ce connecteur IDC50 possède 10 broches de plus et utilise un standard totalement différent, concurrent en son temps, le SCSI.



Il ne reste alors que trois options : un disque SCSI neuf (introuvable), un adaptateur SCSI/SATA (inexistant) ou un adaptateur simulant un disque et stockant les données sur un support comme une carte SD ou microSD. Cette dernière solution existe et s'appelle SCSI2SD.

2. LA SOLUTION : SCSI2SD

Le matériel dont je vais parler est l'œuvre de Michael McMaster qui a constaté il y a une demi-douzaine d'années, comme nous venons de le faire, que les machines anciennes utilisant une interface SCSI avaient de plus en plus de mal à voir leur disque remplacé. Il décida donc de relever ce défi afin d'équiper l'Apple Macintosh LC III qu'il était en train de rénover.

Cette première version de ce qu'il décida d'appeler le SCSI2SD fut commercialisée fin 2013, mais depuis, plusieurs versions se sont succédé, améliorant le concept toujours plus avant, jusqu'à aboutir aux deux versions disponibles aujourd'hui, la V5/V5.1, robuste et éprouvée et la V6, plus performante, mais également plus coûteuse, plus gourmande en énergie et pouvant tantôt poser des problèmes de compatibilité avec les machines les plus anciennes (SCSI-1).

Ce qui a donc commencé comme un émulateur matériel de disque Quantum ProDrive LPS270 à destination d'un Macintosh est maintenant une solution complète, générique et stable permettant de simuler un disque SCSI à destination de nombreuses machines. Plus de 40 ordinateurs et 25 modèles d'échantillonneurs (*samplers*) sont présents dans la liste de compatibilité de la V5, avec quelques noms remarquables comme Mac SE/30, Amiga 3000, Atari Falcon, NeXTStation, SGI Indy, HP 9000, Roland JS-30, AKAI S5000 ou encore Yamaha A5000.

La version 6 n'est actuellement plus disponible sur le site de Michael (codesrc.com), mais peut être achetée en ligne sur la boutique Inertial Computing (USA). Cependant, mon choix s'est porté sur la version 5 fabriquée sous licence et distribuée par iTead

(itead.cc) pour quelques 60€. Ce prix peut sembler dissuasif au premier regard, mais dès lors qu'on considère les alternatives, on se rend rapidement compte que l'investissement est bien plus rentable et durable qu'il n'y paraît.

Précisons également qu'il semble exister des productions de ce matériel, réalisées sans l'accord de Michael McMaster et basées sur les toutes premières versions dont le *design* était diffusé sous licence GPL. Comme vous le savez sans doute, la licence GPL oblige à diffuser les modifications apportées à un travail original et il semblerait que certains constructeurs n'aient pas respecté cette obligation en produisant et vendant des clones dérivés. Suite à cela, Michael a donc décidé que les versions suivantes ne seraient plus aussi « ouvertes » et que seules les sources du firmware et des outils seraient toujours sous licence GPL, mais non plus les fichiers de conception matérielle (schémas, circuit, etc.).

Le SCSI2SD produit par iTead est plus petit que l'original et se présente comme un circuit de 5 cm par 10 cm équipé d'un côté d'un connecteur pour une nappe SCSI 50 broches et d'un connecteur Molex d'alimentation, et de l'autre d'une prise USB micro B, d'un emplacement pour une carte microSD et d'un connecteur pour une led notifiant de l'activité disque. L'ensemble est construit

autour d'une puce Cypress CY8C5267AXI. Ce composant est plus qu'un simple microcontrôleur et est présenté par le constructeur comme un « système sur puce » programmable ou PSoC. Il intègre un cœur ARM Cortex-M3, de la flash, de la RAM et de nombreux périphériques. Son travail est à la fois de passer pour un disque SCSI pour la machine hôte, mais également de fournir l'accès aux données placées sur une carte microSD (jusqu'à 64 Go dans le cas de la version 5).

Le SCSI2SD est alimenté exclusivement en 5V par la machine hôte et sa configuration se fait via le port USB micro B, qui sert également à mettre à jour le firmware. Cette configuration peut supporter jusqu'à 4 périphériques SCSI se voyant attribuer un ID au choix et pouvant être :

- un disque dur,
- un support amovible,
- un CD-ROM,
- une disquette 3,5",
- ou un disque magnéto-optique.

En pratique, le matériel en version 5 (ou 5.1) sera généralement utilisé pour simuler un ou plusieurs disques durs. En effet, l'émulation des supports repose sur une complète simulation qui, en d'autres termes, se limite à l'accès en SCSI. Le SCSI2SD se contente de créer des périphériques virtuels et c'est à la charge de la machine hôte d'accéder, lire et écrire les données. Ceci signifie donc que pour émuler un CD-ROM, il faut accéder en SCSI pour fournir les données. Ce fonctionne-

ment est différent avec la version 6 qui offre un certain nombre de facilités à ce niveau en permettant un accès aux données sur le support microSD directement via la connexion USB. Cependant, la liste de compatibilités vérifiées de la version 6 est bien trop courte à mon goût.

Pour ma part donc, bien que la possibilité d'émuler toutes sortes de supports puisse paraître séduisante, je considère cela comme totalement annexe. Il est encore possible de trouver un lecteur CD-ROM externe SCSI d'occasion à un prix raisonnable (~40€), tout comme il est possible de trouver un lecteur qu'on placera soi-même dans un boîtier externe. Ce type de boîtier ne fait que fournir l'alimentation et les connecteurs externes, on peut donc également se satisfaire du lecteur seul (~20€ en cherchant bien) en le branchant directement et temporairement sur une nappe SCSI. L'un des principaux avantages du SCSI est de justement fonctionner sur plusieurs types de machines, un lecteur CD-ROM externe est donc également un investissement polyvalent si vous disposez de plusieurs ordinateurs vintages.

3. CONFIGURATION DU SCSI2SD

Pour configurer le SCSI2SD, vous devrez le connecter en USB à un PC (Windows ou GNU/Linux) ou un Mac et utiliser l'outil fourni par Michael McMaster, **scsi2sd-util**. Celui-ci est téléchargeable directement sur <http://www.codesrc.com/files/scsi2sd/latest/>,

pour GNU/Linux (64 bits), Windows (64 et 32 bits) et pour macOS. Si votre système n'est pas dans la liste proposée (typiquement un GNU/Linux 32 bits), vous devrez recompiler l'outil via une procédure relativement longue.

scsi2sd-util est un outil multiplateforme utilisant wxWidgets. Il est donc parfaitement identique sur les trois systèmes directement supportés. Dans le

Le SCSI2SD V5/V5.1 est construit autour d'un microcontrôleur Cypress de la famille PSoC 5LP, intégrant un processeur ARM Cortex-M3 à 80 Mhz. Ce composant est chargé de l'émulation de l'interface SCSI, de l'accès aux données sur microSD et de la gestion du port USB permettant la configuration de l'ensemble.





cadre de cet article, je n'ai pas procédé au test sur Mac (mon vieillissant MacBook a cédé sa place à un Lenovo E480 flambant neuf). L'utilisation sous Windows 10 n'a posé aucun problème, mais une manipulation supplémentaire est nécessaire sous GNU/Linux afin d'éviter de devoir lancer l'outil via `sudo`.

Vous devrez ajouter un fichier dans `/etc/udev/rules.d/ scsi2sd.rules` par exemple, contenant les lignes suivantes :

```
# HIDAPI/libusb
SUBSYSTEM=="usb", ATTRS{idVendor}=="04b4", \
ATTRS{idProduct}=="1337", MODE="0666"
SUBSYSTEM=="usb", ATTRS{idVendor}=="04b4", \
ATTRS{idProduct}=="b71d", MODE="0666"

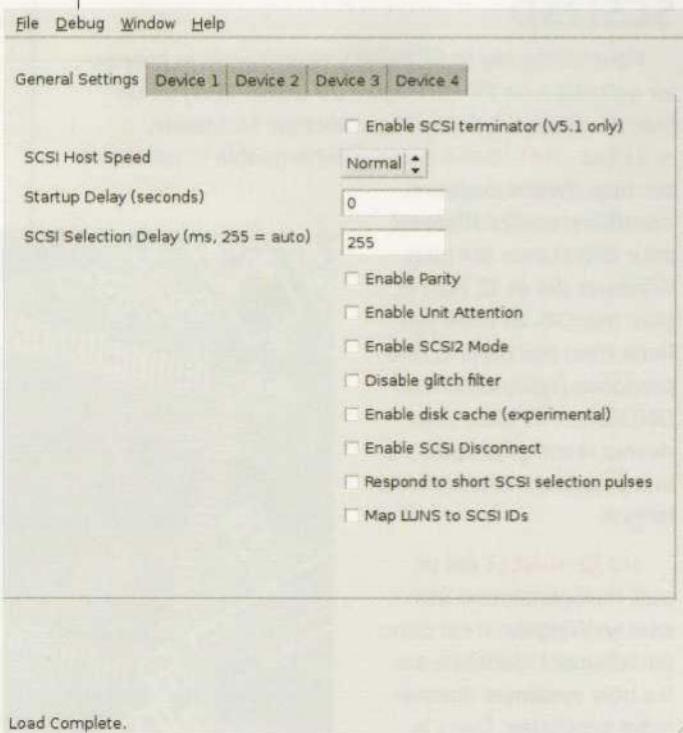
# HIDAPI/hidraw
KERNEL=="hidraw*", ATTRS{busnum}=="1", \
ATTRS{idVendor}=="04b4", ATTRS{idProduct}=="1337", MODE="0666"
KERNEL=="hidraw*", ATTRS{busnum}=="1", \
ATTRS{idVendor}=="04b4", ATTRS{idProduct}=="b71d", MODE="0666"
```

Ceci fait, rechargez la configuration, avec `sudo udevadm control --reload` par exemple, puis branchez votre SCSI2SD. Normalement, celui-ci sera accessible avec `scsi2sd-util` sans avoir besoin des permissions super-utilisateur. Le fait que plusieurs identifiants de vendeur et produits USB (VID/PID) soient utilisés provient du fait que le

microcontrôleur Cypress du SCSI2SD redémarre dans un mode particulier pour la mise à jour du firmware et apparaît comme un périphérique différent.

Cette mise à jour est la première chose que vous devrez faire après le lancement du programme. Vous pourrez récupérer la dernière version en date à l'adresse donnée précédemment, dans le répertoire **firmware**. À l'heure actuelle, il s'agit du fichier **firmware_bundle-v4.8.00.scsi2sd** (avril 2018). Au lancement de `scsi2sd-util`, vous verrez apparaître deux fenêtres : l'interface de configuration et celle de *log* vous informant du déroulement des opérations.

Celle-ci devrait vous afficher quelque chose comme ceci :



L'outil de configuration du SCSI2SD est disponible pour Windows, macOS et GNU/Linux. C'est avec cet utilitaire que vous définirez ce que vous souhaitez émuler et comment, puis chargerez cette configuration dans le SCSI2SD.

```
<file> 16:16:39: SCSI2SD Ready, firmware version 4.4 16:16:39: SD Capacity
(512-byte sectors): 2012160 SD CSD Register: 002d00321b5983d67efbf80164000e5
SD CID Register: 02544d5344303147289711ba8e0078bd <file>
```

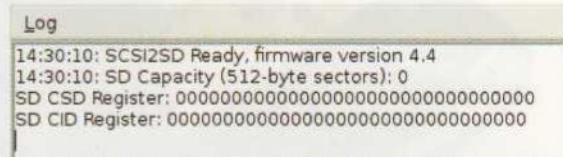
La première ligne nous indique la version actuelle du firmware. Ici, nous sommes très en retard et un certain nombre de fonctionnalités ont été ajoutées ou optimisées depuis cette révision datant de 2015. Nous allons devoir mettre à jour. La seconde ligne nous affiche la capacité de la carte microSD placée sur le SCSI2SD, en nombre de secteurs de 512 octets (1 Go ici). Enfin, les deux lignes qui suivent affichent le contenu de deux des registres de la carte accessible en lecture seule. Ceci démontre que la carte est utilisable et apporte d'éventuelles informations utiles sur son identité, sa configuration et ses performances.

La procédure de mise à jour du firmware est relativement simple, il suffit de passer par le menu *File* et *Upgrade firmware*, chercher et sélectionner le bon fichier et valider. Suite à l'opération, il est recommandé de quitter l'utilitaire, déconnecter et reconnecter la carte. Dès le nouveau lancement de **scsi2sd-util**, vous devriez voir apparaître le bon numéro de version dans la fenêtre d'activité.

Notez qu'il est très important d'avoir une correspondance entre la version de **scsi2sd-util** et celle du firmware. Dans le cas contraire, vous risquez d'avoir des problèmes ou, au mieux, de voir certaines options de configuration être sans effet.

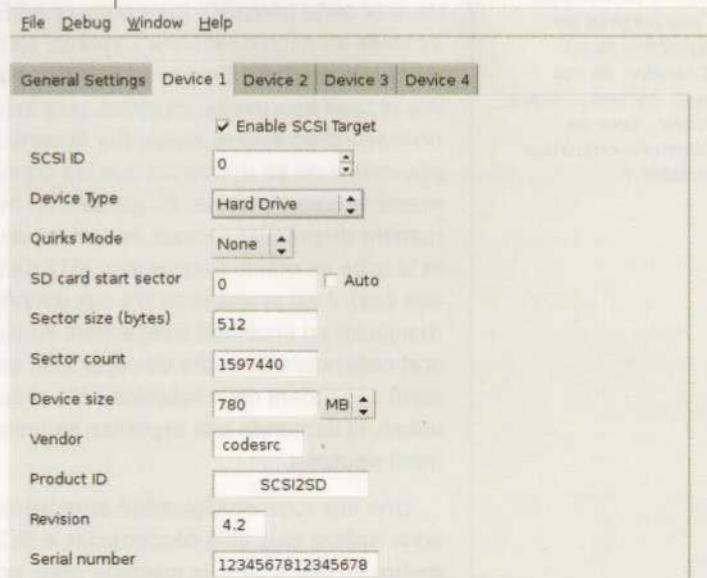
La configuration en elle-même est relativement simple puisque ceci se résume généralement à utiliser les paramètres par défaut sur l'onglet *General Setting* et à activer le premier périphérique (onglet *Device 1*) en spécifiant une taille en Ko, Mo ou Go pour le disque à émuler. Il est également possible de spécifier un nom de vendeur et de produit ainsi que d'autres paramètres pour affiner l'émulation. Ceci est utile, par exemple, pour certains Macintosh n'acceptant que des disques Seagate comme unité de démarrage.

Vous pouvez également choisir d'émuler plusieurs disques (jusqu'à 4) via les trois onglets restants et l'outil vérifiera alors que les zones des données ne se chevauchent pas (mais pas si elles dépassent la taille de la

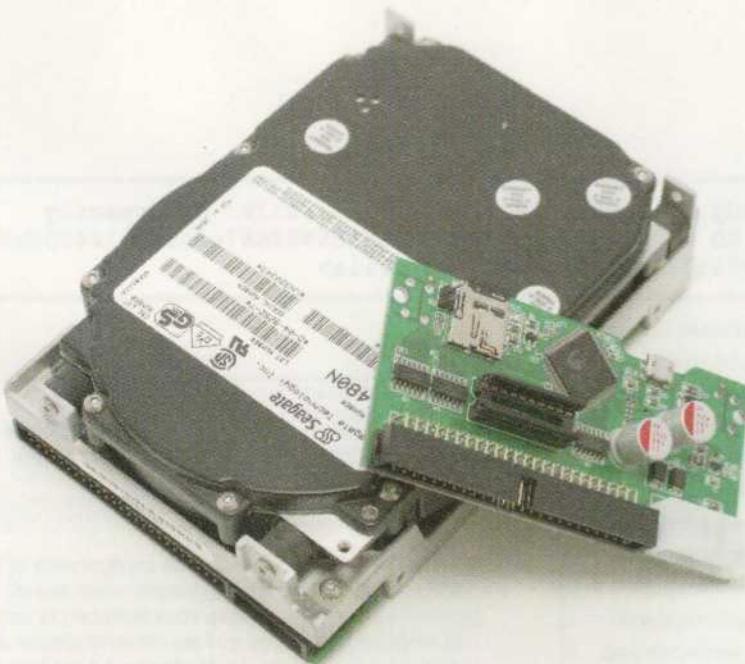


Dès le lancement de l'utilitaire de configuration et si le SCSI2SD est correctement détecté, vous devriez voir apparaître cette petite fenêtre vous indiquant la version du firmware utilisé ainsi que les caractéristiques de la carte microSD insérée. Ici, la version 4.4 est bien trop ancienne et une mise à jour vers 4.8 est indispensable.

Le SCSI2SD permet, sur la base d'une seule carte microSD, de configurer jusqu'à 4 disques à simuler. Généralement et en fonction de la taille maximum de disque supportée par la machine hôte, on se contentera souvent d'émuler un seul disque.



Load Complete.



Quelques 25 années séparent ces deux périphériques, avec un écart technologie considérable. Qui aurait pensé, lorsque ce disque dur a été fabriqué, qu'un espace de stockage d'une poignée de gigaoctets serait considéré, de nos jours, comme presque risible... pour un téléphone-ordinateur portable ?

carte microSD). Si vous ne voulez pas vous amuser avec les numéros de secteurs (*SD card start sector*), cochez simplement *Auto* et les disques virtuels s'enchaîneront.

Notez que la configuration n'est pas enregistrée dans la carte microSD, mais dans une EEPROM intégrée au microcontrôleur Cypress. Celle-ci peut être lue/récupérée depuis le SCSI2SD via le menu *File* et *Load from device*, modifiée, puis inscrite à nouveau avec *Save to device*. Sur la carte microSD elle-même ne se trouveront que les données que la machine hôte y placera. En gardant en mémoire le numéro du premier secteur, le nombre de secteurs et la taille en octets d'un secteur (512 dans la plupart des cas), il est possible de lire ces données et les manipuler en accédant directement au support. Ceci sort cependant du cadre de cet article, est totalement dépendant de la machine hôte et du système utilisé, et demande une expertise technique relativement poussée.

Une fois votre configuration enregistrée, il ne vous restera plus qu'à déconnecter le SCSI2SD et le mettre en place dans la machine cible, pour lui faire croire qu'il s'agit effectivement d'un disque dur. Un avantage annexe très intéressant, au-delà de l'économie et de la robustesse de la solution, réside dans le support utilisé. En effet, rien en vous empêche de prévoir plusieurs microSD de même modèle et

ainsi de basculer d'un système à un autre, faire des sauvegardes ou tout simplement prévoir une version vous permettant d'expérimenter sans risquer de perdre des données. Le tout pour le prix d'une carte qui, avec une capacité de 1 ou 2 Go, est accessible à toutes les bourses.

Cependant, il faut bien l'avouer, comme beaucoup j'ai tout simplement remplacé un disque mourant par cet ensemble, qui n'a pas bougé depuis. Et je pense que je vais faire de même avec un vieux Mac SE/30 me suppliant de prendre soin de lui et de lui offrir une seconde jeunesse. Le duo SCSI2SD et carte microSD de quelque gigaoctets revenant à moins de 70€, je pense qu'il s'agit là, tout simplement, de la mise à jour ultime pour un ordinateur de collection en SCSI.

4. UN VRAI BONHEUR

J'ai initialement acquis un premier exemplaire de ce matériel en raison d'un énorme problème avec ma NeXTstation Turbo dont je vous avais parlé dans le numéro 4 (pfiouuu déjà ?!). En effet, celle-ci a décidé subitement de ne plus démarrer et une investigation poussée m'a permis d'isoler le problème : le disque dur st1480n (420 Mo). Celui-ci possédait une diode Transil (ou TVS en anglais), chargée de protéger des surtensions. Seul problème, quand ce genre de composant fait son travail, il met en cours-circuit l'alimentation et la machine refuse totalement de démarrer.

LISEZ **HACKABLE**
~ MAGAZINE ~ EN LIGNE !



...CELUI D'AUJOURD'HUI ET CEUX D'HIER...
...LE BIMESTRIEL ET LES HORS-SÉRIES !

À CONSULTER SUR :
connect.ed-diamond.com



Le retrait de la diode permettait un démarrage de la machine, tout comme le retrait du disque lui-même, mais c'était un premier signe de la dégradation définitive de la configuration. Le disque était en fin de vie et mieux valait le remplacer que de risquer davantage de dégâts.

Le passage d'un disque au SCSI2SD impliquait la réinstallation de tout le système, une nouvelle configuration et une réinstallation des éléments personnalisés (compatibilité Y2K, client DHCP, serveur SSH, etc.) qui avaient fait l'objet de l'article dans le magazine, mais c'était mieux que d'avoir ainsi une épée de Damoclès suspendue sur ce qui est maintenant une pièce de collection. Changer un disque est facile, trouver une nouvelle alimentation pour la machine ou pire, une carte mère, c'est une tout autre histoire. On ne prend pas de risque quand il existe si peu d'ordinateurs de ce type en état de marche dans le monde.

La plupart des machines s'accommoderont d'un SCSI2SD en lieu et place d'un disque dur SCSI sans le moindre problème. Une machine NeXT cependant nécessitera une petite manipulation supplémentaire dans le cas d'une réinstallation du système sur un tel support. Ceci n'est pas un problème physique, ni même directement lié au système NeXTstep/OpenStep, mais en relation avec l'outil de formatage (création du système de fichiers UFS). En installant le système, vous obtenez une erreur « *cylinder group too large* » liée à la manière dont sont organisées les données sur le disque. Sans entrer inutilement dans le détail, le problème provient du calcul du nombre de cylindres formant un groupe et ceci est une conséquence de la géométrie du disque émulé.

La version 4.2 du firmware règle ce problème en supportant la configuration de la géométrie du disque, mais ceci n'est pas accessible directement depuis l'interface de configuration. Pour corriger le problème, il faut exporter la configuration dans un fichier XML via **File** et **Save to file** et éditer le fichier à la main pour ajuster les paramètres **sectorsPerTrack** et **headsPerCylinder** de, respectivement, **63** et

255, à **139** et **4**. On enregistrera ensuite le fichier modifié pour le charger avec **File** et **Open file**, et l'enregistrer dans le SCSI2SD avec **Save to device**. Cette modification n'entraînera pas un changement de taille du disque simulé, mais satisfira l'outil **mkfs** chargé de le formater en UFS avant l'installation. Il s'agit là d'une spécificité NeXT et il en existe quelques autres pour certaines plateformes (comme l'outil Apple HD SC qui n'acceptera que des disques s'annonçant comme étant des Seagate ST225N).

En cas de problème, mieux vaut en premier lieu consulter le wiki du créateur du SCSI2SD (www.codesrc.com/mediawiki) et se plier d'une petite recherche en ligne. Toutes les machines annoncées comme compatibles avec cette solution ont été testées par les utilisateurs et les éventuelles astuces sont publiées. Gardez cependant à l'esprit qu'il s'agit d'une solution de rétocomputing et qu'en tant que telle, elle n'est sans doute pas parfaite. Mais c'est toujours mieux qu'un disque dur mourant ou à 10 fois le prix auquel vous avez acquis la machine à restaurer.

Finalement, mon seul regret concernant l'installation d'un SCSI2SD est la disparition de ce délicieux bruit de disque qui démarre, accélère et se stabilise à la mise en route de la machine, et bien sûr son opposé lors de l'arrêt. J'aurai bien laissé le disque d'origine (alimenté, mais déconnecté), mais il n'y avait malheureusement pas la place... **DB**

ACTUELLEMENT DISPONIBLE GNU/LINUX MAGAZINE HORS-SÉRIE N°97

LES HORS-SÉRIES CHANGENT DE FORMULE !



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



LEROY MERLIN PRÉSENTE

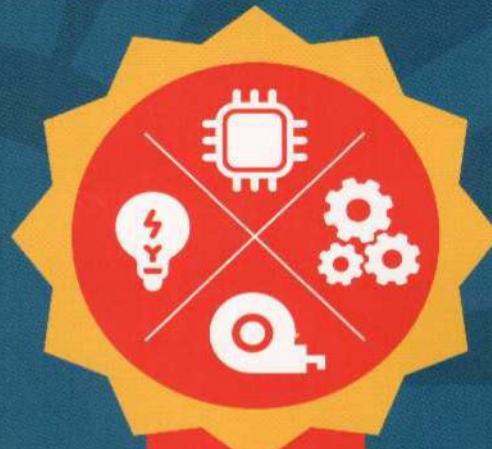
Maker Faire® Paris

DU 9 AU 11 NOVEMBRE 2018 - CITÉ DES SCIENCES ET DE L'INDUSTRIE

APPEL AUX MAKERS

NOUS RECHERCHONS
BRICOLEURS ★ MAKERS
DESIGNERS ★ CODEURS
GAMERS ★ ARTISTES
HACKERS ★ CREATEURS
INVENTEURS ★ YOUTUBERS
INGENIEURS ★ RÊVEURS

EN PARTENARIAT AVEC
DESIGNSPARK PRÉSENTÉ PAR 



INSCRIPTIONS JUSQU'AU 14 JUILLET 2018 SUR LE SITE
PARIS.MAKERFAIRE.COM

cité
sciences
et industrie