

LoRa™

échange débit contre portée

Robert Lacoste (Chaville)

Dans mon précédent article [1], je vous ai expliqué que seuls trois paramètres déterminent la sensibilité d'un récepteur radio. Pour mémoire, ce sont respectivement la qualité de l'électronique, les performances de la modulation utilisée, et surtout le débit de la liaison. Ici je passe de la théorie à la pratique : la solution LoRa développée par Semtech Semiconductors. Pour ceux que cela intéresse, sachez d'ores et déjà que je continuerai dans le même esprit dans le prochain article, avec LoRaWAN, un protocole standardisé utilisant évidemment LoRa...

LPWA ?

Démarrons par quelques généralités. Si vous ne vivez pas dans une grotte, vous savez sûrement que l'on nous promet des dizaines de milliards d'objets connectés dans les prochaines années. Imaginons que vous deviez développer un tel objet connecté, bien sûr sans fil. Comment le relier au grand Internet ? Pour faire simple, trois grandes approches existent (fig. 1).

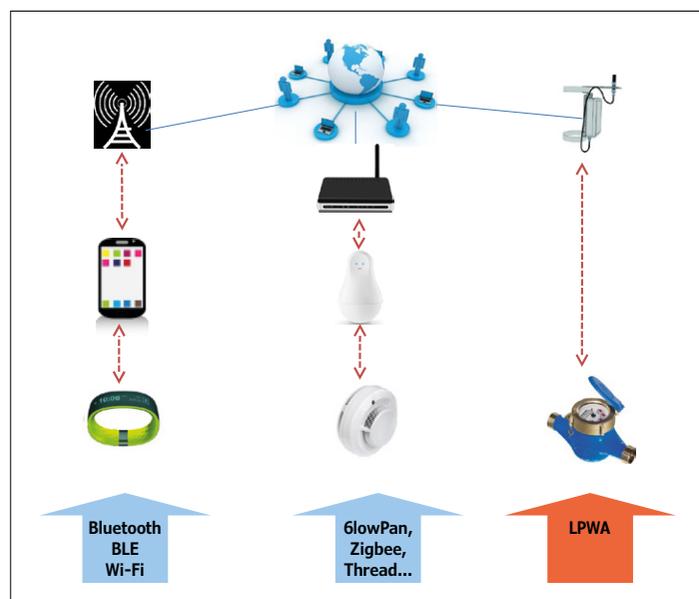


Figure 1. Les trois grandes familles de réseaux pour raccorder un dispositif connecté à l'internet.

La première solution, probablement la plus utilisée aujourd'hui, consiste à utiliser une passerelle de communication qu'on a tous dans notre poche : l'ordiphone. Votre objet connecté peut en effet se connecter par radio à celui-ci, typiquement via Bluetooth, Bluetooth Low Energy ou Wi-Fi, et l'ordiphone assure la connexion à l'internet. Les avantages sont la facilité, la faible consommation et le très bas coût, mais il faut un ordiphone.

La seconde possibilité consiste à remplacer l'ordiphone par une passerelle fixe, installée à votre domicile. Votre « box internet » peut faire l'affaire si votre objet est doté d'une interface Wi-Fi, sinon une passerelle spéciale sera nécessaire. C'est le domaine des protocoles sans fil de type « domotique » comme Zigbee, Zwave, 6lowPan ou Thread. C'est à très bas coût et sans ordiphone, mais installer des passerelles est toujours pénible. De plus, pour des projets un peu ambitieux, le nombre de passerelles nécessaires croît très vite, car ces solutions ont des portées assez faibles.

La troisième voie consiste à éviter toute passerelle locale, en adoptant une technologie à très longue portée. En effet, couvrir une grande distance permet de faire communiquer votre dispositif directement avec une passerelle qui rayonne sur tout un pâté de maisons, voire une ville entière. Traditionnellement cette solution nécessitait d'utiliser un réseau cellulaire (2G ou 3G par ex.), d'où des coûts et des consommations non négligeables. Ces dernières années, plusieurs nouvelles solutions ont toutefois changé la donne. Baptisées du sobriquet de LPWAN (*Low Power Wide Area Network*, réseau étendu à faible consommation), ces réseaux atteignent des portées qui se chiffrent en kilomètres, tout en étant peu énergivores.

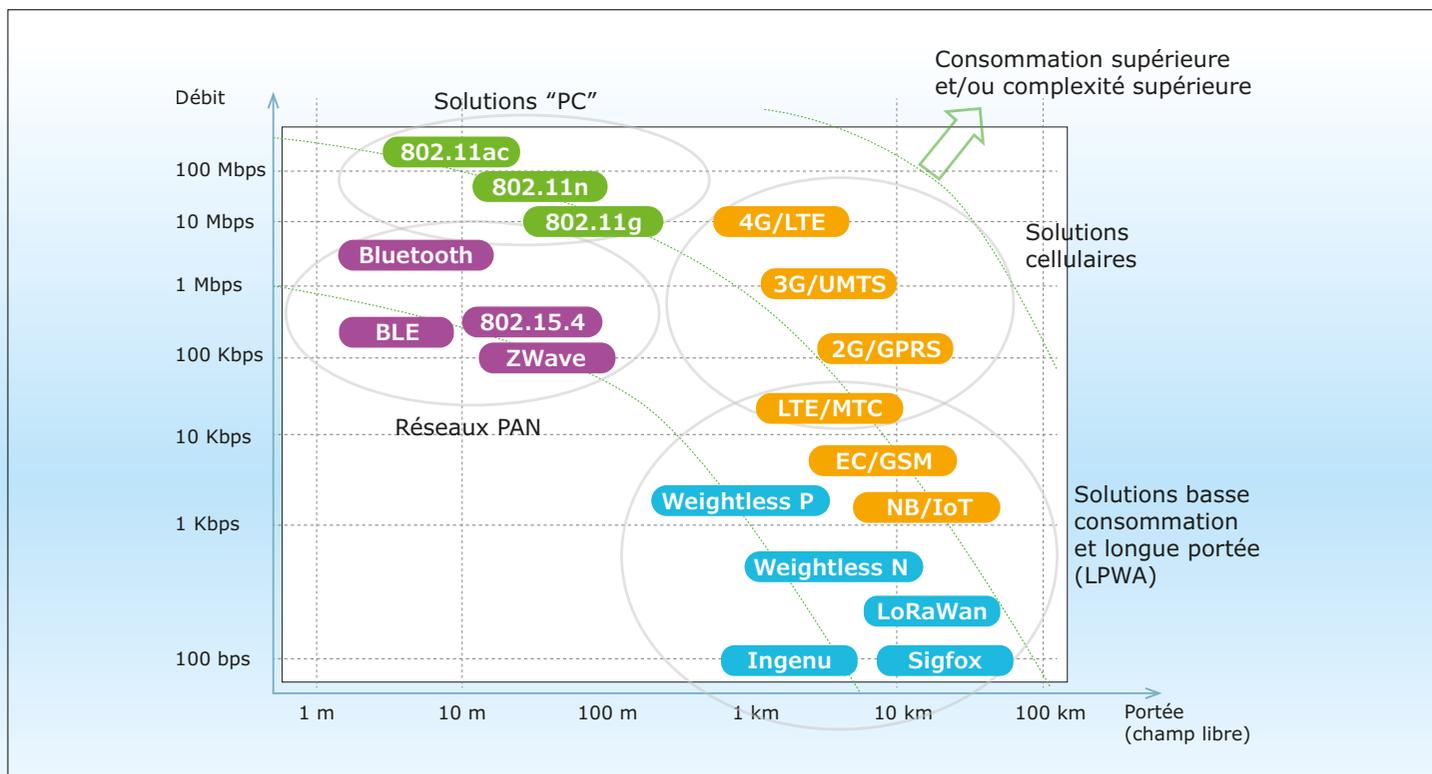


Figure 2. Un panorama global des technologies radio. Les solutions LPWA troquent débit contre portée.

Réduire le débit !

L'idée générale du LPWA est simple. Comme l'illustre la **figure 2**, les technologies radio classiques (Wi-Fi, Bluetooth, etc.) privilégient des débits très élevés et ont donc une portée réduite. Le réseau LPWA répond à d'autres exigences : il doit porter beaucoup plus loin, mais ne pas consommer plus, ce qui a pour conséquence une réduction drastique du débit de la liaison. Le réseau étendu LoRa est une solution parmi d'autres : Sigfox, Ingenu, Weightless, etc. Notez que l'on observe cette tendance chez les opérateurs de téléphonie mobile également : même s'ils augmentent de manière farouche les débits en 4G pour vous permettre de regarder des vidéos en HD sur votre ordiphone, ils déploient aussi des réseaux de type NB-IoT (*Narrow Band Internet of Things*) qui privilégient justement des débits nettement plus faibles pour le monde de l'Internet des Objets.

Est-il si simple de réduire considérablement le débit d'une liaison radio ? La réponse est non. Imaginez que vous avez une transmission d'un débit de 10 kbps qui utilise une modulation classique de type FSK (*Frequency Shift Keying*, modulation par déplacement de fréquence). Cette modulation augmente ou baisse la fréquence de l'émetteur pour transmettre respectivement un 0 ou un 1, typiquement sur une plage de l'ordre de ± 10 kHz ici. Si vous souhaitez réduire le débit de la liaison, disons à 100 bps, il faut, pour que la modulation conserve les mêmes caractéristiques, réduire aussi ce déplacement de fréquence à ± 100 Hz. C'est là que le bât blesse. Cela implique que le récepteur soit très stable en fréquence. Par ex. si la fréquence de la porteuse est de 868 MHz, ± 100 Hz représente $\pm 0,1$ ppm (partie par million), bien plus faible que la dérive de

n'importe quel quartz classique. Cette technique, baptisée à bande ultra-étroite, existe, c'est celle utilisée par Sigfox, mais elle nécessite de sacrées ruses du côté du récepteur.

La solution LoRa

LoRa est une solution alternative à ce problème, elle permet de réduire énormément le débit d'une liaison sans pour autant nécessiter de quartz très précis. Pour la petite histoire, LoRa a été inventée et développée en 2009 par une jeune pousse française, Cycleo, rachetée en 2013 par le fabricant de semi-conducteurs Semtech. Avec LoRa, la plage de modulation est toujours large (typiquement de 125 à 500 kHz), mais les bits à transmettre sont encodés avec une technique qui réduit le débit sans modifier la largeur de modulation. Cette modulation met bien sûr également à profit des techniques avancées (correction d'erreur, etc.) pour améliorer autant que possible la sensibilité du récepteur.

Pour les lecteurs inquiets, je me permets de proposer une relecture de mon précédent article : à débit binaire égal, la sensibilité d'un récepteur est la même que l'on utilise une modulation à bande ultra-étroite ou très large (pour une même complexité du récepteur), donc cette approche a tout son sens...

Une solution à large bande comme LoRa offre plusieurs avantages : la possibilité d'utiliser un quartz ordinaire, mais aussi une meilleure insensibilité aux parasites (en général à bande étroite) et une très grande flexibilité. Évidemment il y a des inconvénients. Tout d'abord le récepteur nécessite des traitements numériques complexes. Heureusement il y a des circuits intégrés spécialisés pour cela (j'y reviens plus bas). L'autre

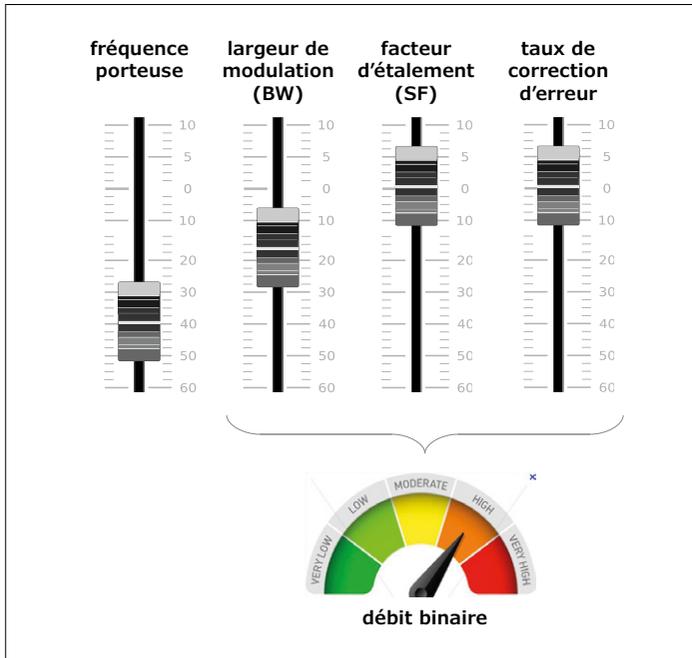


Figure 3. LoRa permet de régler indépendamment quatre paramètres qui ont tous un impact sur la portée et le débit.

inconvenient potentiel est un risque de saturation plus rapide du canal radio, mais pour l’instant on n’en est pas encore là.

CSS, vous avez dit CSS ?

Rentrons dans les détails. LoRa est ce qu’on appelle une couche physique. C’est-à-dire que c’est juste une méthode d’encodage de la suite de bits qui constitue le message, pour l’envoi dans les airs, et avant de procéder au traitement inverse du côté du récepteur. Lora ne se charge pas des fonctions nécessaires pour l’application, par ex. le codage de l’adresse de l’émetteur et des destinataires, les acquittements éventuels..., c’est le rôle d’un protocole de plus haut niveau (LoRaWAN en est un). Comme je l’ai annoncé, l’une des forces de LoRa est sa très grande flexibilité. Tout d’abord la fréquence de la porteuse peut

être choisie dans la plage de 137 MHz à 1020 MHz, avec les composants actuellement disponibles. Cependant, en Europe, la plupart des applications utilisent la bande 868 MHz. L’utilisateur peut également régler librement trois paramètres qui influencent tous le débit binaire (fig. 3). Tout d’abord la **largeur de modulation**, notée BW (*Band Width*), peut être sélectionnée entre 7,8 kHz et 500 kHz. Une bande plus large donne bien sûr un débit plus rapide. Ensuite un paramètre étrange nommé **facteur d’étalement** (SF, *Spreading Factor*) peut être réglé entre 6 et 12. Augmenter ce nombre de 1 réduit le débit de moitié. Enfin il est possible de sélectionner un **code de correction d’erreur** plus ou moins efficace ; il ajoute de 0 à 4 bits de correction d’erreur tous les 4 bits utiles transmis. C’est bien beau tout cela, mais comment ça marche ? Il y a là un problème. Les détails de la modulation LoRa ne sont pas officiellement divulgués par Semtech, qui nous dit juste que LoRa utilise une modulation particulière appelée CSS (*Chirp Spread Spectrum*) [2]. Si vous regardez avec un analyseur de spectre classique l’allure d’un signal LoRa, vous serez déçu : on observe une sorte de plateau qui occupe une largeur de bande BW autour de la fréquence de la porteuse choisie, rien de plus. Plus précisément, à un instant donné le signal est une porteuse pure, mais sa fréquence évolue dans le temps un peu comme en FM. Pour y comprendre quelque chose, il faut utiliser un analyseur de spectre sophistiqué qui permet de mesurer et d’afficher la fréquence du signal au cours du temps. Fort heureusement mon labo dispose d’un joli analyseur de spectre en temps réel Tektronix RSA5106 avec ce genre de fonction. Regardez la **figure 4** pour voir le résultat. Dans l’analyse de fréquence en fonction du temps, on voit que la fréquence est en permanence modifiée, avec une vitesse de balayage constante. Par contre il y a des « sauts » dans cette variation, ce sont eux qui encodent les bits transmis. Maintenant vous avez compris pourquoi on parle d’*étalement de spectre à modulation de fréquence* (CSS) : si l’on revient à des fréquences audio et qu’on applique à un haut-parleur ce signal avec de telles rampes en fréquence, on entend un son ressemblant à *chiiiiirp-chiiiiirp...* Les Français appellent plutôt ça un signal wobulé (*woooaab-woooaab* au lieu de *chiiiiirp-chirrrrr...*), mais c’est un peu la même chose !

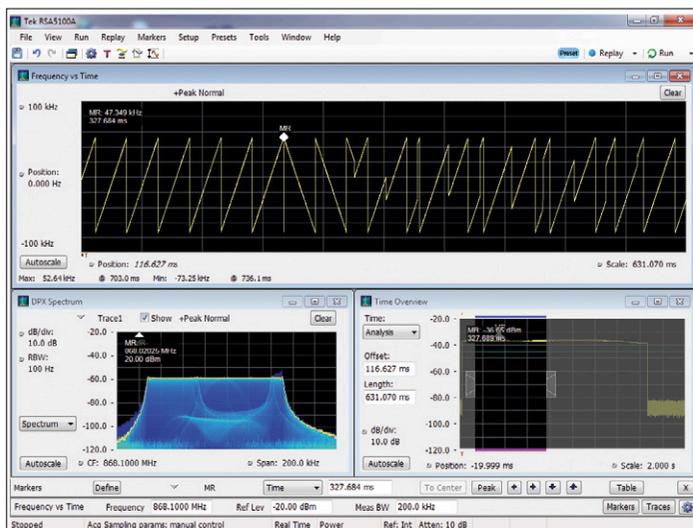


Figure 4. Une trame LoRa visualisée avec un analyseur de spectre en temps réel, ici un Tektronix RSA5106.

Bien que les détails de la modulation LoRa ne soient pas officiellement divulgués, nombre de personnes ont creusé le sujet. La **figure 5** montre les grandes lignes de l’encodage utilisé avec les paramètres suivants pour l’exemple : BW = 125 kHz, SF = 12, code correcteur 5/4. Avec BW = 125 kHz, la largeur du balayage en fréquence est bien sûr de 125 kHz, c’est l’amplitude sur l’axe des ordonnées du graphe *fréquence=f(t)*. Ensuite on remarque que chaque trame radio commence par un préambule constitué de rampes complètes, suivi de quelques rampes « inversées » (la fréquence réduit au cours du temps au lieu d’augmenter). Ces rampes inversées sont le signal de synchronisation qui indique au récepteur le début du message effectif.

Comment sont codés les bits à transmettre ? Tout d’abord, les bits du message sont groupés par paquets, qu’on appelle des symboles. Comme on a sélectionné SF = 12, ces bits sont regroupés 12 par 12. Un symbole est donc un nombre de 0 à 4095 ($2^{12} - 1$), et va être transmis en une seule fois. Sur le graphe, on remarque qu’il y a un saut exactement

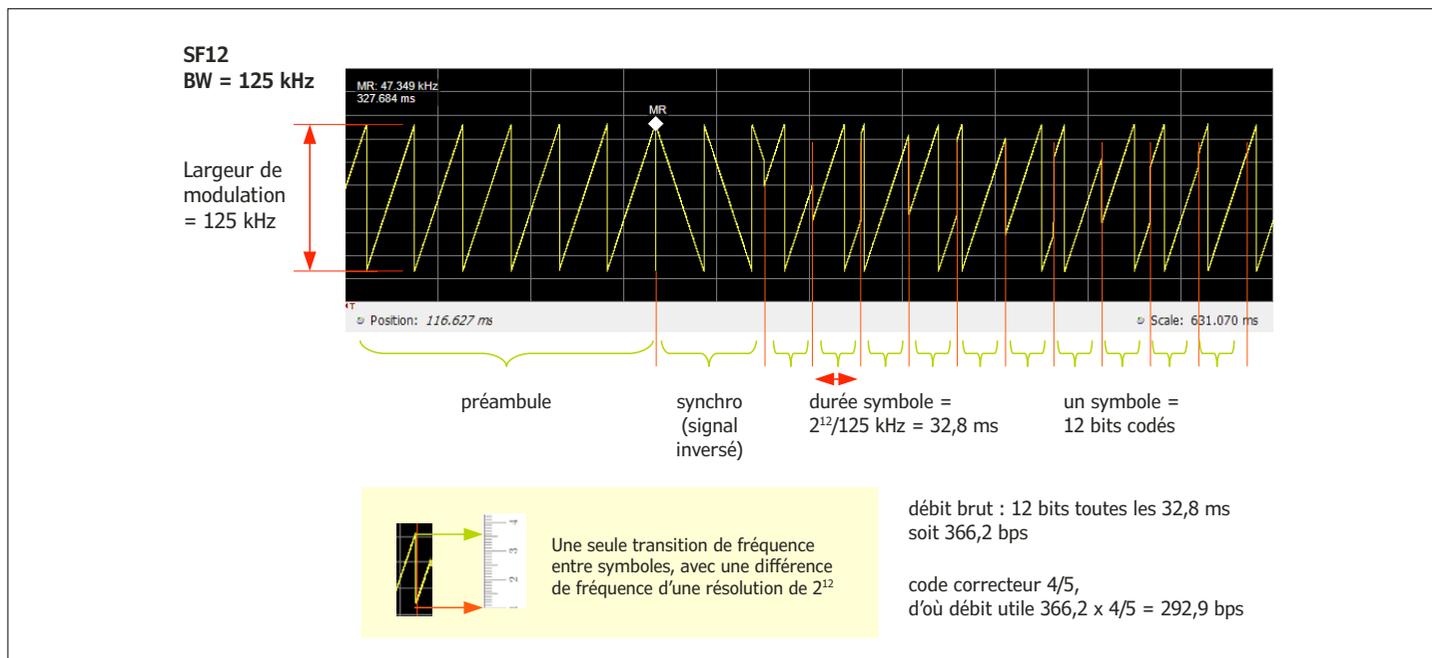


Figure 5. LoRa décrypté !

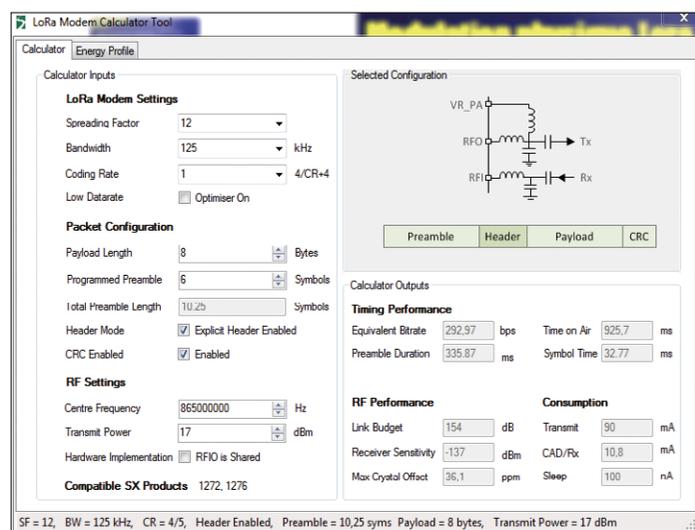
toutes les 32,8 ms dans la séquence de balayage de fréquence. Cela correspond à la durée en LoRa d'un symbole avec $BW = 125 \text{ kHz}$ et $SF = 12$; cette durée est calculée avec la formule $2^{12}/125.000 \text{ Hz} = 0,0328 \text{ s}$. Dans ce mode, LoRa transmet un symbole, soit 12 bits d'un coup, toutes les 32,8 ms, soit 366,2 bps si vous faites le calcul. Ajoutez-y un bit de correction d'erreur tous les 4 bits et vous obtiendrez un débit utile de 292,9 bps, ce qui est bien le débit de LoRa dans cette configuration.

Vous vous demandez peut-être comment un symbole de 12 bits est codé en un saut de fréquence ? C'est simple, du moins sur le papier : il suffit de décomposer la plage de variation de fréquence de 125 kHz en $2^{12} = 4096$, soit des pas de $125.000/4096 = 30,52 \text{ Hz}$. Si le symbole de 12 bits vaut 1, LoRa fait un saut de 30,52 Hz ; pour un symbole égal à 2, le saut est d'une fréquence deux fois supérieure, et ainsi de suite jusqu'à la valeur 4095. Bon, évidemment c'est plus complexe que cela, car la loi d'encodage est plus subtile, mais vous avez compris l'idée. Sachez qu'une rétroanalyse complète de LoRa est maintenant disponible sur la toile, voir [3]

Outil logiciel...

Le calcul du débit binaire effectif d'une liaison LoRa en fonction des différents paramètres n'est donc pas très simple. Heureusement Semtech met à disposition une petite application Windows qui se charge des calculs [4]. Regardez la **figure 6** où j'ai sélectionné les mêmes paramètres que dans mon exemple ci-dessus ($BW = 125 \text{ kHz}$, $SF = 12$, codage 5/4). L'outil indique que le débit binaire effectif sera de 292 bps, comme prévu. Ouf ! L'application calcule également la durée d'émission d'une trame avec le nombre voulu d'octets utiles, en ajoutant les préambules et autres octets d'encapsulation. Les ordres de grandeur sont intéressants : pour huit octets utiles avec ces paramètres, la durée d'émission sera de 925 ms.

Ceci permet de toucher du doigt une limitation de toutes les solutions à bas débit comme LoRa : en Europe, la réglementation impose des limites au temps d'émission dans les bandes « libres » pour laisser de la place aux copains. En particulier, dans la bande 868-868,6 MHz utilisée principalement par LoRa, cette limite est égale à 1 % du temps, calculée sur une heure glissante. 1 % de 3600 s fait 36 s, il n'est donc possible de n'envoyer au maximum qu'une petite quarantaine de messages de 8 octets par heure. Oubliez donc les rêves de transfert de gros fichiers de données, les solutions LPWA visent l'envoi de petits messages peu fréquents. Sinon la seule solution consiste à augmenter le débit, mais là on perd en portée... Pour finir, l'application Windows fournit également une estimation de la sensibilité du récepteur avec les paramètres sélectionnés.

Figure 6. Copie d'écran du logiciel *LoRa Modem Calculator* fourni par Semtech, une aide appréciable...

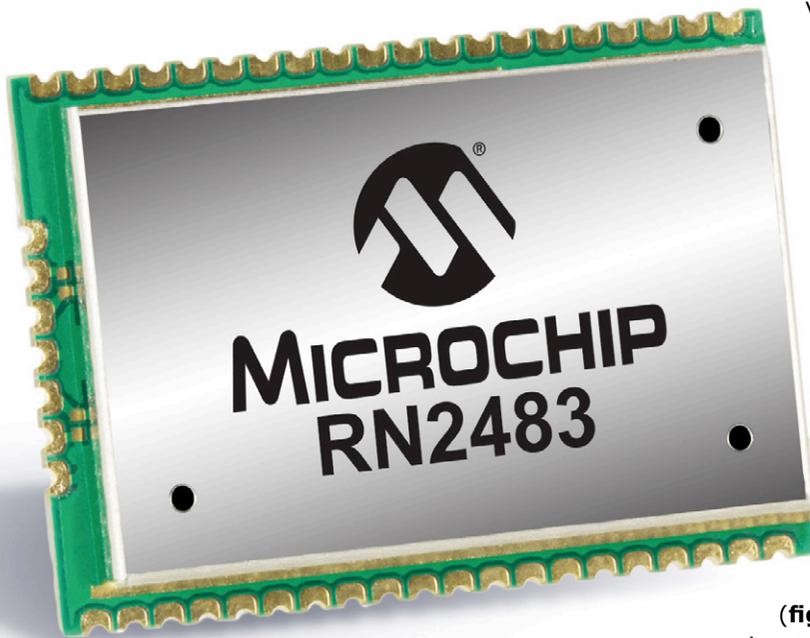


Figure 7. Un exemple de module LoRa.

tionnés. Regardez de nouveau la figure 6, elle est ici de l'ordre de -137 dBm. Avec un émetteur d'une puissance de 25 mW (soit 14 dBm), cela représente un bilan de liaison de 151 dB. Ce chiffre ne vous impressionne pas ? Et si je vous dis que cela correspond théoriquement à une portée de plus de 900 km en champ libre. Toujours pas ? Sur le terrain, il y a bien entendu des pertes, des obstacles et des réflexions, mais des portées de plusieurs dizaines de kilomètres sont tout à fait envisageables avec des antennes bien choisies et pas trop mal placées...

Et le matériel ?

Comment utiliser LoRa dans votre prochain projet ? Tout d'abord vous pouvez soit n'utiliser que la couche physique LoRa et interconnecter vos dispositifs en point à point, soit avoir recours à LoRaWAN comme je vous l'expliquerai dans mon prochain article. Ensuite sur le plan matériel, vous avez deux options : une puce ou un module. Pour les puces, le choix est réduit, car le seul fabricant de circuits intégrés radio qui supportent LoRa est Semtech. La situation changera courant 2017 grâce à des accords de licence prévus avec STM et Microchip, donc sur-

veillez la presse... Pour le moment, furetez sur semtech.com et téléchargez la documentation du SX1276 [5], c'est le composant le plus complet de la gamme LoRa ; il couvre toute la gamme de fréquences de 137 MHz à 1040 MHz. À noter qu'une version un peu moins chère est suffisante pour le 868 MHz : le SX1272. Vous verrez que les composants externes nécessaires sont peu nombreux : un quartz, quelques composants passifs, un connecteur d'antenne et bien sûr un microcontrôleur pour le piloter (en général un petit 32 bits de type Cortex M0). Pour le logiciel, vous trouverez de quoi démarrer en *open source* en cherchant « SX1276 » sur le site www.github.com.

L'autre solution, plus simple, consiste à mettre en œuvre un module prêt à l'emploi qui contient un circuit intégré qui supporte LoRa, un microcontrôleur préprogrammé et tous les composants auxiliaires. De très nombreux modules de ce genre sont maintenant disponibles, comme le RN2483 de Microchip

(figure 7), il coûte moins de 15 €. Il suffit de le raccorder à votre microcontrôleur principal via une liaison UART et de lui envoyer quelques commandes ASCII pour être connecté en LoRa. Ah, il faudra aussi une antenne... Pour les aficionados du format Arduino, vous trouverez aussi un *shield* supportant un SX1276 et la tripaille nécessaire, compatible avec l'environnement de développement Mbed d'ARM (SX1276MB1xAS [7]). Elektor propose également dans son e-shoppe un HAT pour RPi [8]. Aucune raison donc pour attendre !

Pour conclure

Les réseaux LPWA, et LoRa en particulier, permettent comme vous l'avez compris d'établir des liaisons par radio sur de longues distances et à faible coût, avec comme principale contrainte un débit de données très lent, ce qui limite donc le flot de messages. Mais cela répond quand même à de très nombreuses applications !

Dans le prochain article, je vous présenterai LoRaWAN, le protocole qui s'appuie sur LoRa et qui a été développé par l'alliance éponyme. Le fait que cette Alliance LoRa regroupe déjà plus de 200 *petites* sociétés dont Cisco, IBM, ARM, Orange, Bouygues Telecom, KPN, Proximus, ST Microelectronics, Microchip et quelques autres est probablement un bon signe pour l'avenir.

(160236)

Cet article a été publié dans la revue Circuit Cellar (n°313, août 2016).

Liens et références

- [1] *Hors Circuits : bruit et sensibilité des récepteurs*, Elektor, 12/2016 : www.elektormagazine.fr/160307
- [2] AN1200.22, *LoRa Modulation Basics* : www.semtech.com/images/datasheet/an1200.22.pdf
- [3] *Reversing Lora*, Mark Knight / Bastille networks : <https://archive.org/details/ReversingLora>
- [4] Outil de calcul, *LoRa Calculator: fast evaluation of link budget, time on air and energy consumption* : www.semtech.com/wireless-rf/rf-transceivers/sx1272/
- [5] Émetteur-récepteur LoRa SX1276 : www.semtech.com/wireless-rf/rf-transceivers/sx1276
- [6] Module sans fil LoRa RN2483 : www.microchip.com/RN2483
- [7] Shield LoRa SX1276MB1xAS : developer.mbed.org/components/SX1276MB1xAS/
- [8] HAT LoRa pour RPi : www.elektor.fr/dragino-lora-gps-hat-for-raspberry-pi