



un couteau suisse (2)

le matériel et le logiciel

Gilles Brocard (France)

Après avoir abordé le protocole LoRa dans la première partie, nous allons entrer dans le détail de la réalisation d'un émetteur-récepteur LoRa à partir d'un petit module du commerce. Piloté par un Arduino Nano et un programme entièrement dédié au paramétrage et à l'émission/réception LoRa, le montage final constitue un véritable couteau suisse LoRa.

Le petit circuit imprimé décrit ici réunit un module LoRa, une carte Arduino Nano, un régulateur 5 V, quelques LED et des bornes à vis pour connecter aisément toute sorte de capteurs, de commandes, un écran OLED I²C ou des LED supplémentaires. Ce montage est d'une grande flexibilité, due en partie à son programme. Il constitue un véritable couteau suisse LoRa.

Le schéma (**figure 1**) se résume aux liaisons entre l'Arduino Nano et le module E220-900M30S avec un régulateur de tension et une LED confirmant la présence du +5 V et deux autres LED indiquant une émission (TX) ou une réception (RX).

K2 permet de connecter d'autres composants au circuit. Les ports A0 à A5 sont des entrées analogiques ou E/S numériques suivant le paramétrage. En plus, A4 et A5 peuvent être configurés en bus I²C pour piloter des capteurs en I²C ou encore un petit afficheur OLED 128 × 64 pixels. A6 et A7 sont deux entrées uniquement analogiques. Les entrées analogiques du Nano ont une résolution de 10 bits. La borne +5 V peut fonctionner en entrée ou en sortie.

La sortie DIO2 du module LoRa est reliée à l'entrée TX de l'aiguillage d'antenne et à la grille de T1, qui sert d'inverseur logique. Son drain commande l'entrée RX de ce même aiguilleur d'antenne.

LED1 (blanche) signale une émission (TX). La LED2 (verte) indique la réception (RX) d'un signal LoRa valide, mais sans renseignement sur son contenu. En complément, le programme active pendant une seconde le port A2 si le message attendu est détecté. Cela permet facilement de transformer le montage en une télécommande à grande portée.

Alimentation

La tension d'entrée sur K3 est de 24 V maximum. Il est aussi possible d'alimenter le montage directement en 5 V par K4. Dans les deux cas, l'alimentation doit être capable de fournir jusqu'à 750 mA.

Un dissipateur pour IC1 est optionnel, puisqu'en mode LoRa le rapport cyclique émission/repos est tellement petit (1% maxi) que la dissipation moyenne reste toujours très faible. En réalité, on peut presque aller jusqu'à 35 V à l'entrée. En effet, même avec 750 mA consommé pour un signal de sortie de 30 dB, la puissance à dissiper n'atteint même pas 0,25 W (en supposant que le circuit soit au repos le reste du temps).

Le circuit imprimé

Le circuit imprimé (**figure 2**) mesure 87 mm × 50 mm et peut se loger dans la plupart des boîtiers de blindage (optionnel). La sortie antenne est faite avec un connecteur SMA, parfaitement adaptée à cette fréquence proche du gigahertz (ce qui n'est pas le cas des BNC). Le câblage est facile. Les LED, les résistances et les condensateurs de 100 nF et de 1 µF sont en CMS 1206, facile à souder. Les autres composants sont traversants. Si votre platine à des trous non métallisés (**figure 3**), n'oubliez pas les straps, les quelques traversées du plan de masse et ceux pour les liaisons de D4 et D5 entre les deux faces du circuit.

Si vous souhaitez réaliser des liaisons avec de grandes portées (10 km et plus), il est conseillé de placer le circuit imprimé dans un petit boîtier étanche, directement au pied d'une antenne placée en hauteur (2 m du sol est un optimum), afin de minimiser les pertes dues au câble HF. Une autre solution consiste à remplacer l'Arduino Nano par un ESP32. Ainsi, les messages pourront transiter en Wi-Fi, avant d'être relayés en LoRa. Les modifications à apporter au programme sont mineures, une version adaptée pourra être fournie.

Un commutateur de mode TX/RX peut être connecté entre les contacts D2 et GND du bornier K1. Il permet de changer, à tout moment, le mode de fonctionnement du module en émission (ouvert) ou en réception (fermé). Si le montage n'est utilisé que dans le mode RX, pontez les contacts 1 et 2 de K1. En mode TX fixe, laissez les contacts ouverts.

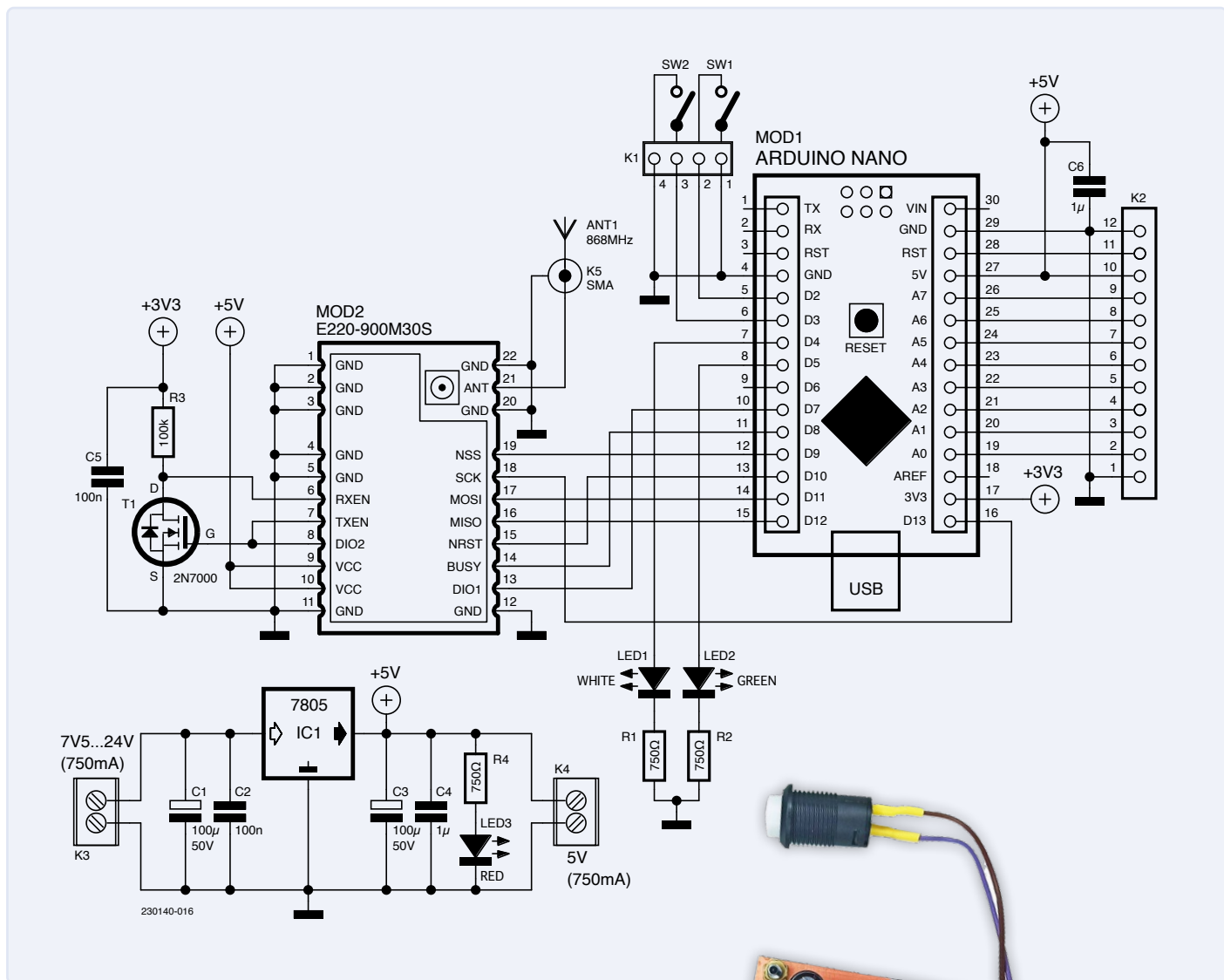


Figure 1. Le schéma du couteau suisse LoRa.

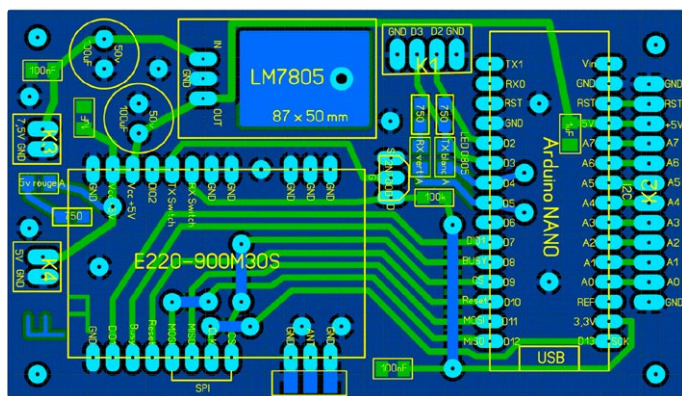


Figure 2. Les fichiers du circuit imprimé sont disponibles avec le code source du programme sur la page internet de cet article [1].

Le programme

Le programme du Nano [1] vous permet de modifier facilement les paramètres LoRa pour toute sorte d'expérimentation et de tentative de grande portée d'une transmission LoRa. Il y a un record à battre... Le logiciel permet également le basculement à chaud entre l'envoi

(TX) et la réception (RX) de messages. La connexion de nombreux capteurs analogiques ou numériques (I²C) est également possible. Voir l'encart « Présentation rapide du programme de gestion du LLCC68 » pour savoir comment le programme est organisé.

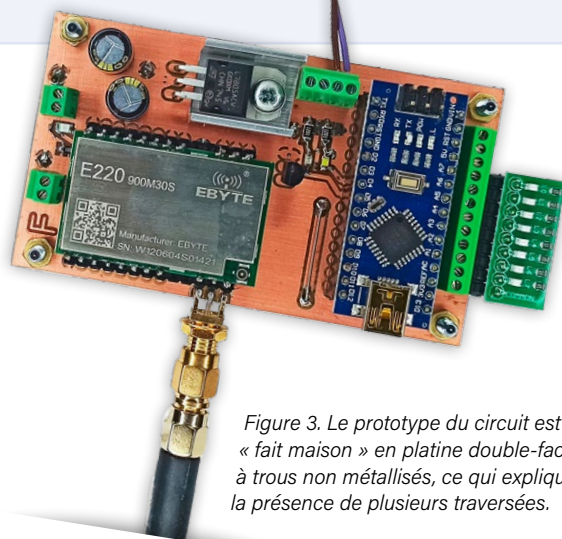


Figure 3. Le prototype du circuit est « fait maison » en platine double-face à trous non métallisés, ce qui explique la présence de plusieurs traversées.

Optimisation LoRa

Le LLCC68 permet d'informer l'émetteur de la qualité de réception du signal pour qu'il puisse adapter les paramètres d'émission (puissance, SF, CR, etc.). C'est l'un des moyens d'ajustement utilisé par la couche LoRaWAN.

- Le RSSI (*Received Signal Strength Indication*) est la valeur de la puissance du signal reçu que le récepteur peut renvoyer à l'émetteur. Ce dernier peut ensuite modifier les paramètres d'émission, augmenter ou réduire sa puissance d'émission afin d'augmenter l'autonomie tout en maintenant un niveau suffisant de transmission.
- La sensibilité. Le LLCC68 peut mesurer le niveau de réception minimum lui permettant d'extraire un signal sans erreur. Cette valeur est très faible dans le cas du LLCC68, puisqu'elle peut atteindre $-129\mu\text{dBm}$ en fonction des paramètres choisis.
- Le SNR (*Signal-to-Noise Ratio*) est le rapport entre la valeur de la puissance du signal reçu et le niveau du bruit ambiant que le LLCC68 mesure en dehors des périodes de réception de données. Dans le cas de LoRa, le niveau du bruit peut être très largement supérieur à celui du signal (SNR négatif).
- La fonction *GetStats* fournit le nombre

total de paquets reçus, le nombre de ceux reçus avec erreur et le nombre de ceux reçus avec erreur d'entête. La fonction *ResetStats* réinitialise toutes les statistiques.

La lecture des 106 pages de la fiche technique n'est pas indispensable, mais très utile pour approfondir certains sujets traités dans cet article. Nous vous conseillons également de lire le remarquable document rédigé par les enseignants du Lycée Dorian [3]. Il traite de la couche physique LoRa, mais aussi du LoRaWAN et de la démodulation LoRa pour laquelle Semtech ne fournit aucune info.

Les principaux paramètres LoRa

Tous les paramètres LoRa sont rassemblés entre les lignes 83 et 111 :

Ligne 83 : la bascule pour le fonctionnement en LoRa.

Ligne 84 : le paramétrage de FS, BW, CR et LDRO.

Ligne 99 : la configuration du paquet envoyé est paramétrée à l'aide de neuf octets, les trois derniers sont communs avec la modulation FSK.

Pour chacune de ces trois lignes de commande (lignes 83, 84 et 99), le premier octet n'est pas un octet de paramétrage, mais l'*opcode* (la commande) qui indique au LLCC68 ce qu'il doit faire des octets suivants. Il y a 41 opcodes en total. Consultez la fiche technique pour plus de détails. Vous y trouverez également les 36 adresses des registres de configuration.

SF, le facteur d'étalement

La sensibilité de la partie analogique de la réception du signal ne peut être modifiée que par le gain du préamplificateur d'entrée, qui est paramétrable (voir ligne 154 du programme). Mais, comme nous l'avons vu, plus on augmente la valeur de SF, plus le débit binaire diminue. Or l'efficacité du traitement numérique de reconstitution du message étant inversement proportionnelle au débit binaire, la sensibilité numérique augmente donc avec la valeur de SF (ainsi que

le temps de transmission). Cela s'ajoute à la sensibilité de la partie analogique, permettant de grandes valeurs de la sensibilité, et donc des portées plus importantes.

SF correspond aussi au nombre de bits transmis pendant la durée T_s , c'est-à-dire par symbole.

Les temps de transmission d'un symbole T_s et d'un message de 24 octets sont présentés dans le **tableau 1**. Semtech fournit sur son site un calculateur (**figure 4**, [2]) qui permet une évaluation précise des durées de transmission, de la consommation et du budget de liaison. N'hésitez pas à l'utiliser, car bien qu'il ne soit pas fait spécifiquement pour le LLCC68 (mais pour les SX1272 à SX1277), les résultats sont quasiment identiques. L'augmentation d'une valeur de SF à la suivante double presque la durée de transmission, comme on peut le constater sur le tableau 1.

La valeur du débit de transmission binaire (*bit rate*) est R_b et est exprimée par la formule :

$$R_b = SF \cdot (BW / 2^{SF}) \cdot 4/(4+CR) \text{ (bits/s)} \quad (1)$$

Puisque SF est égal au nombre de bits par symbole, on peut en déduire la vitesse en baud R_s (*symbol rate*) :

$$R_s = R_b / SF \text{ (baud)}$$

Comme les trois valeurs BW, SF et CR sont accessibles, on paramètre facilement la vitesse de transmission que nous souhaitons.

CR, le taux de codage

LoRa permet d'augmenter la robustesse du codage en augmentant la redondance du signal. Pour CR = 1, le taux de redondance est de 4/5, ce qui veut dire que cinq bits sont utilisés pour en coder quatre.

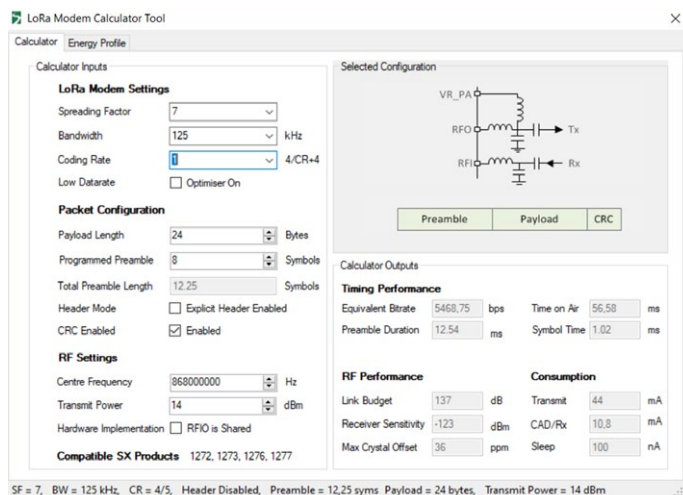


Figure 4. Le calculateur modem LoRa de Semtech existe aussi en version en-ligne.

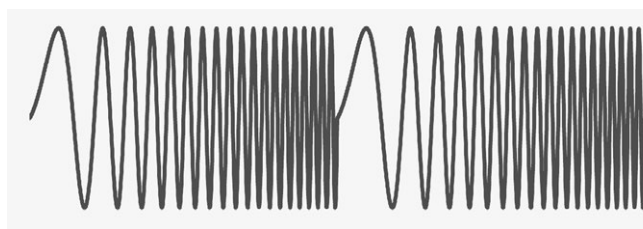


Figure 5. Représentation de deux chirps (avec une modulation augmentée pour l'illustration).

Tableau 1. Les durées de transmission en fonction du facteur d'étalement SF.

SF	R_b (bit/s)	R_s (baud)	Durée préambule (ms)	T_s (ms)	Durée transmission (ms)
5	15,625	3,125	3,135	0,256	16,37
6	9,375	1,562,5	6,27	0,51	30,85
7	5,468,75	781,3	12,54	1,02	56,58
8	3,125	390,6	25,09	2,05	102,91
9	1,757,8	195,3	50,18	4,10	185,34
10	976,5	97,7	100,35	8,19	370,69
11	537,11	48,8	200,70	16,38	659,46

Ces valeurs sont sans entête. La présence d'un entête augmentera les durées. Ici, $CR = 1$, $BW = 125$ kHz, un message (payload) contient 24 octets et un préambule consiste en $8 + 4,25 = 12,25$ symboles. $T_s = 2^{SF}/BW$.

Le facteur d'étalement SF correspond également au nombre de bits transmis pendant la durée T_s . Les valeurs R_b et R_s du tableau sont celles obtenues avec le calculateur Semtech données pour être compatibles avec les produits SX1272 à 1277. Bien que le fondeur ne le précise pas, elles sont également compatibles avec le LLCC68. Remarque : les délais de transmission R_b d'un bit du payload (message utile) et d'un bit du préambule ne sont pas identiques.

Pour $CR = 4$, le taux est de 4/8, soit huit bits pour en coder quatre. Puisque tout avantage à une contrepartie, augmenter CR fait diminuer R_b (voir équation 1) et, par conséquent, augmente la durée de transmission d'un message.

BW, la bande passante

La largeur de bande BW est aussi appelée encombrement spectral du signal modulé. La variation d'un chirp (figure 5) qui s'étend de $F - df$ à $F + df$ détermine la bande passante occupée. Puisque, selon l'équation 1, R_b est proportionnel à BW, on peut réduire la durée de transmission d'un message en choisissant une valeur plus grande pour BW. Contrairement aux autres paramètres, il n'y a pratiquement aucune contrepartie à cette augmentation, hormis l'encombrement de la bande de fréquence.

L'optimisation des données pour les faibles débits

Compte tenu de la durée potentiellement longue du paquet à des valeurs de SF élevées (10 ou 11), l'option d'optimisation des données pour les faibles débits (LDRO, *Low Data Rate Optimize*) doit être activée. Elle améliore la fiabilité de la transmission vis-à-vis des variations de fréquence pendant la durée de la transmission et de la réception du paquet. Son utilisation est obligatoire lorsque la durée du symbole T_s dépasse 16 ms. Par exemple, on peut voir au tableau 1 que si $SF = 11$, LDRO doit être activé, car $T_s > 16$ ms. L'émetteur et le récepteur doivent utiliser la même valeur pour LDRO.

Présentation rapide du programme

Le programme de 48 Ko comporte 571 lignes, dont une grande partie de commentaires très détaillés (en français). Ils renvoient aux numéros de pages de la fiche technique du LLCC68. Le programme est structuré simplement afin de pouvoir être adapté à tout type d'application.

Ligne 1 : le programme inclut la bibliothèque *SPI.h*. Elle fait partie de la dotation initiale de l'EDI Arduino.

Aucune installation de bibliothèque n'est à prévoir.

Lignes 5 à 15 : deux tableaux, l'un pour stocker le message à émettre (TX), le second pour stocker le message reçu (RX).

Lignes 17 à 28 : regroupement de l'initialisation des bornes d'entrées et de sorties pour faciliter l'adaptation à tout type d'application.

Lignes 31 à 48 : déclaration des variables nécessaires pour les calculs internes du programme et les adresses des registres du LLCC68 utilisées par le programme.

Lignes 50 à 56 : déclaration des variables du paramétrage général du LLCC68.

Lignes 57 à 81 : déclaration des variables dédiées à la modulation FSK (les trois derniers paramètres du paquet sont communs avec LoRa).

Lignes 83 à 111 : déclaration des variables pour la modulation LoRa.

Lignes 113 à 170 : déclaration des variables pour le paramétrage de toutes les fonctions RF du LLCC68

Ligne 172 : la fonction Arduino obligatoire *setup*

Ligne 197 : l'autre fonction Arduino obligatoire *loop*. Elle se divise en deux parties. Le

commutateur de mode RX/TX détermine quelle partie est exécutée.

Lignes 232 à 277 : cinq fonctions de correction de certaines limites du LLCC68 (voir chapitre 15 de la fiche technique)

Lignes 278 à 311 : les fonctions *Message*, *busy* et *RX_Wait*, liés au mode RX. Dans la fonction *RX_Wait*, une boucle infinie attend la réception d'un message LoRa valide. Le commutateur de mode permet de sortir de la boucle et de passer du mode RX au mode TX.

Lignes 313 à 355 : *TXRX_Setup*, le paramétrage commun aux deux modes RX et TX, puis *TX_Setup* pour les paramétrages du mode TX et *RX_Setup* pour les paramétrages du mode RX

Ligne 356 : la fonction *TX_Send* qui permet le

lancement du mode TX et démarre l'émission d'un message.

Ligne 371 : la fonction *RX_Read* qui lance le mode RX et place le programme dans une boucle infinie en attendant un message LoRa valide.

Ligne 411 : la fonction *Fsk_Setup* qui paramètre le mode FSK.

Ligne 433 : la fonction *Lora_Setup* qui paramètre le mode LoRa.

Lignes 454 à 483 : regroupent les fonctions de *commande SPI* puis *Status* qui permet de connaître l'état du LLCC68.

Lignes 484 à 571 : regroupent les fonctions de mesure *RSSI*, *SNR* et les *statistiques* des erreurs de transmission et du nombre de messages reçus. Ils peuvent être réémis pour optimiser la liaison.

Vitesse de transmission

La vitesse de transmission d'un message en LoRa et la portée de la transmission est la résultante des valeurs de SF, CR et BW et LDRO, mais également du choix d'avoir un entête (option *explicite*) ou pas (option *implicite*) et de la longueur de l'entête, lorsqu'il est présent.

Retour sur les modules E220-900T

Les logiciels destinés au pilotage des modules E220-900T 22 ou 30 présentés dans la première partie [5] ne permettent pas le paramétrage LoRa comme vous pouvez le faire maintenant. Ils ajustent automatiquement les paramètres qui vont bien pour obtenir la vitesse de transmission désirée. C'est fort dommage, car chaque arrangement a ses avantages et ses inconvénients. Il est donc intéressant de pouvoir choisir. C'est la raison qui nous a fait sélectionner le module E220-900M30S qui vous donne, via sa liaison SPI, toute latitude dans son paramétrage.

Utilisation du logiciel

Toutes les commandes et les paramétrages, hormis celles qui sont spécifiques à votre capteur, sont rassemblés en début de programme et sont abondamment commentés. D'autre part, pendant l'exécution du programme, vous pouvez suivre pas à pas son déroulement, sur le moniteur série de votre PC, car le programme comporte un grand nombre de `serial.print`. Voir aussi l'encart **Présentation rapide du programme** pour plus de détails.

Envoyer un message (TX)

En tout début de programme, ligne 6, il y a un tableau `write_buffer` qui, au moment du lancement du programme, doit contenir votre message. Pour cela, remplacez les codes ASCII des lignes 7 et 8 par votre message (127 octets maxi en ASCII). Si votre message est une valeur numérique (p. ex. issue de la mesure d'un capteur), il vous faudra la convertir en ASCII avant de l'inscrire dans le tableau.

Respectez le format du tableau. Ne changez pas les deux premiers octets : 0x0E est la commande (*opcode*) nécessaire au LLCC68 pour écrire dans le tampon ; 0x80 est l'adresse du début de la partie TX du tampon. Toutefois, il faut modifier la valeur du troisième octet, qui est le nombre total d'octets de votre message. Elle sera utile lors de la récupération du message au moment de la réception. Attention, toutes les valeurs sont en hexadécimale.

Suivant la longueur de votre message, vous pourrez être amené à modifier la valeur de plusieurs paramètres : longueur maximale du message (ligne 99, octet 4 et 7), mais également les lignes 125 et 126

pour la réception. Ligne 122, vous pourrez changer la répartition du Buffer entre réception et émission et les paramètres spécifiques à LoRa (lignes 83 à 112).

Mettez l'interrupteur SW_1 en position TX (ouvert), puis relancez le programme. Votre message est ensuite envoyé périodiquement, espacé d'un délai préprogrammé (ligne 229). Il faudra, vraisemblablement, réajuster le rapport temps d'occupation/temps de repos pour qu'il ne dépasse pas 1 % (contrainte de la réglementation française pour l'émission LoRa).

Recevoir un message (RX)


Positionnez l'interrupteur SW_1 en position RX (fermé), puis relancez le programme. Il va alors attendre la réception d'un message LoRa valide qui sera automatiquement placé dans le tableau `recevoir_buffer` (ligne 14). Il vous suffira de le lire.

Une autre manière de faire consiste à modifier la fonction `RX_Read`, ligne 371, qui permet de récupérer un message reçu par la boucle, ligne 381 à 396. Il contient une fonction de reconnaissance (optionnelle) de la lettre « A » qui déclenche la mise à 1 du drapeau `Message_Valide`. C'est un exemple qui allume une LED de la carte, mais qui peut faire une télécommande.

Quelques idées d'application

La balise GPS pour animal, personne, drone ou autre objet est un exemple d'application de notre couteau suisse LoRa. Mais, le programme que nous avons réalisé est très versatile et il permet un grand nombre d'autres applications, tel que :

- La centralisation de données (humidité du sol, ensoleillement, etc.) d'une exploitation agricole ou d'une station météo.
- Une centrale domotique étendue à un jardin, un parc, plusieurs maisons, une rue entière ou même un village.
- La surveillance (incendie, sécurité) ou la télémétrie d'une zone étendue ou isolée (piscine, terrain de sport, unité industrielle, etc.).
- Une centrale de distribution de l'heure, des phases de la lune et celles des principales planètes vers des horloges réceptrices réparties dans une maison ou un monument plus important (musée, mairie, etc.) ou dans tout le quartier d'une ville.

Il y en a bien plus encore, mais il est probable que d'autres idées germent déjà dans votre imagination... 

230140-B-04

La norme européenne

Vous pouvez émettre avec le protocole LoRa sur la bande 868 MHz en respectant la réglementation européenne [4]. En voici le résumé :

1. Ne pas émettre en dehors de la bande des 868 MHz.
2. Ne pas utiliser la bande passante $BW = 500 \text{ kHz}$.
3. Ne pas dépasser 14 dBm (25 mW) de puissance rayonnée dans l'air. Cela peut

correspondre à beaucoup plus en sortie du module suivant les pertes entre le module et l'antenne.

4. Ne pas émettre pendant plus de 1% du temps (rapport cyclique). Cela veut dire que si le temps de transmission d'un message est de 0,1 seconde, nous ne pouvons effectuer une nouvelle émission que toutes les $0,1 \times 100 = 10$ secondes, soit six fois par minute (ce qui suffit à de nombreuses applications).

Prenons un exemple. Si, comme l'auteur, votre compagnon est un chien Beagle, très fugueur, dont vous souhaitez connaître les coordonnées GPS afin de pouvoir le retrouver lors de ses fugues, un collier contenant un récepteur GPS, un émetteur LoRa et une petite batterie suffit. Il vous renvoie six fois par minute la position du chien dans un rayon de 10 km en campagne. L'ensemble consomme peu d'énergie et on a un système léger, petit et avec une grande autonomie.



À propos de l'auteur

Gilles Brocard est ingénieur-conseil à la retraite et formateur LTspice. Il est également spécialiste de la conception de convertisseurs (SMPS).

Questions ou commentaires ?

Envoyez un courriel à l'auteur (brocard.gilles.b26@gmail.com), ou contactez Elektor (redaction@elektor.fr).



Produits

- > **Radio Logicielle HackRF One Great Scott Gadgets (1 MHz à 6 GHz) (SKU 18306)**
www.elektor.fr/18306
- > **CircuitMess Chatter – DIY LoRa Communicator (SKU 20407)**
www.elektor.fr/20407
- > **Claus Kühnel, Develop and Operate Your LoRaWAN IoT Nodes, Elektor 2023 (SKU 20147)**
www.elektor.fr/20147

LIENS

- [1] Code source et circuit imprimé : <https://www.elektormagazine.fr/230140-B-04>
- [2] LoRa modem calculator tool : <https://lora-developers.semtech.com/build/tools/calculator>
- [3] Lycée Dorian: « Caractérisation de l'interface radio LoRa d'un réseau de communication LoRaWAN », [eduscol.education.fr : https://www.elektormagazine.fr/lycee-dorian](https://www.elektormagazine.fr/lycee-dorian)
- [4] Réglementation officielle radio pour la France: https://www.arcep.fr/uploads/tx_gsavis/21-1589.pdf
- [5] Première partie : <https://www.elektormagazine.fr/230140-04>

LES R&S® ESSENTIALS OSCILLOSCOPE NOUVELLE GÉNÉRATION : EVOLVED FOR MORE CHALLENGES

Précision facilitée.



NOUVEAUX oscilloscopes R&S®MXO 5

La série R&S®MXO 5 fournit une technologie révolutionnaire d'oscilloscopes pour accélérer votre compréhension et vos tests de systèmes électroniques.

Avec des modèles quatre et huit voies, les spécifications de la série R&S®MXO 5 impressionnent, faisant que l'instrument dépasse les autres choix de l'industrie.

De plus, les oscilloscopes de la série R&S®MXO 5 sont la quintessence de la technologie de pointe en fournissant des résultats rapides et précis. Avec une technologie personnalisée et des fonctionnalités révolutionnaires, ces oscilloscopes sont les outils parfaits pour la compréhension des comportements de circuits.

En savoir plus :
www.rohde-schwarz.com/product/mxo5

ROHDE & SCHWARZ
Make ideas real

