



GREAT  
SCOTT!

# construit un système d'alarme LoRa



lien vers  
YouTube

**GreatScott!** (Allemagne) avec  
**Luc Lemmens** (Elektor Labs) pour l'intro et la  
conclusion

Nous présentons ici un projet du youtubeur GreatScott!. Il s'agit de transmettre vers son appartement par LoRa un signal de notification du système anti-intrusion de son garage, à une distance de 600 m. Il utilise deux cartes Nucleo STM32 et deux modules LoRa RFM95 (plus des cartes de connexion Elektor). Le matériel a été parrainé par Elektor, c'est GreatScott! qui a assemblé et testé le matériel.

Tout le monde ne sait pas que *Great Scott!* est une interjection de surprise, voire de consternation[...], populaire dans la seconde moitié du XIX<sup>e</sup> et au début du XX<sup>e</sup> siècle, considérée aujourd'hui comme dépassée (Wikipedia). Certains se souviendront d'avoir entendu l'exclamation *Nom de Zeus* souvent utilisée par le Dr Emmett Lathrop Brown dans *Retour vers le futur*. Eh bien, c'est la traduction de l'interjection *GreatScott!* de l'original anglais dans *Back to the Future*. De nos jours, les électroniciens connaissent tous le canal YT *GreatScott!* lancé en 2013.

## GreatScott! raconte l'histoire de son garage

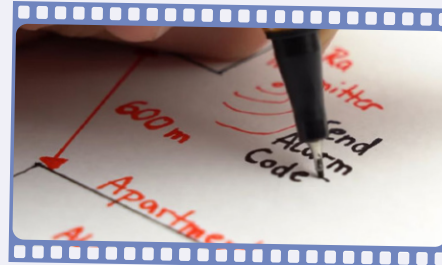
Les deux croquis Arduino nécessaires pour l'extension de l'alarme sont étonnamment simples, grâce aux bibliothèques Arduino RFM95 que vous trouvez sur GitHub ou sur la page de cet article [1]. Voici maintenant le récit par GreatScott! lui-même de l'histoire de son projet.



Mon garage est équipé d'une alarme : barrière lumineuse, lumière stroboscopique, sirène et automate *Controllino*, depuis qu'un lascar est entré par effraction il y a quelques mois. J'ai aussi amélioré le verrouillage du garage, mais le principal problème vient des 600 m qui séparent mon appartement de mon garage. Pour que je puisse être averti aussitôt que quelqu'un déclenche l'alarme, Elektor m'a proposé un ensemble de composants : deux cartes LoRa RFM95, avec des circuits imprimés de connexion adéquats, deux cartes de développement à base de STM32 et une passerelle *LGO2 Dragon* compatible LoRa.



Pour ce projet, j'utilise la technique de communication LoRa pour créer un petit système d'émission-réception qui enclenchera un avertisseur sonore dans mon appartement aussitôt que l'alarme de mon garage se manifestera.



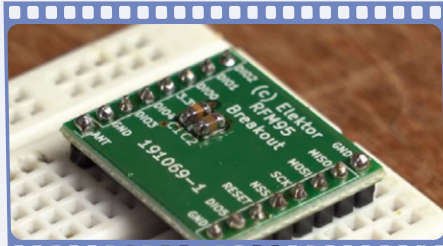
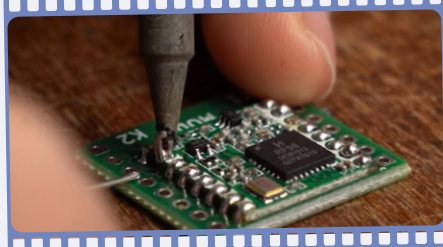
LoRa est l'acronyme de *Long Range* que *Semtech*, son inventeur, décrit comme plateforme sans fil à longue portée et à faible puissance, adoptée par de très nombreux réseaux de l'internet des objets (IOT) dans le monde entier. Cela me semble prometteur pour mon projet puisque je souhaite transmettre très peu de données sur une distance relativement longue. Pour le matériel, j'ai commencé par souder les RFM95 à leurs cartes de connexion, avec quelques collecteurs mâles et des condensateurs de découplage.

regarder la vidéo du projet  
avec l'application Blipper

télécharger  
l'application

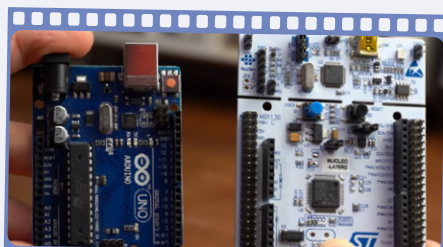


=



Selon la fiche technique, les RFM95 sont des modules émetteurs-récepteurs LoRa à faible puissance : ils peuvent transmettre et recevoir des données modulées LoRa. Ce sont donc les composants clés de mon projet. Ils sont dotés d'une interface SPI. Pour communiquer avec eux, il nous faut donc un microcontrôleur, et du code pour lire et écrire dans leurs registres.

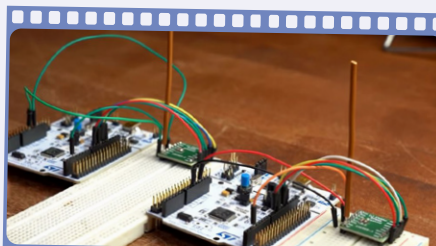
Côté microcontrôleurs, j'ai déballé les deux cartes de prise en main STM32 Nucleo L476RG. Elles ressemblent à des cartes Arduino Uno.



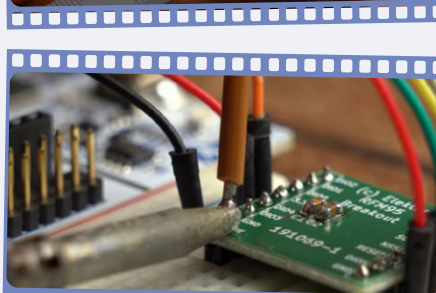
Eh oui, on pourrait utiliser un Arduino pour interagir avec le RFM95, mais l'avantage de la carte STM32 est de fonctionner avec des niveaux logiques de 3,3 V, au lieu de 5 V comme l'Arduino. Le RFM95 fonctionne sous 3,3 V.

Il est temps de connecter nos deux cartes aux deux modules LoRa selon le schéma mentionné dans la bibliothèque Arduino

LoRa que j'utiliserai pour simplifier le codage.

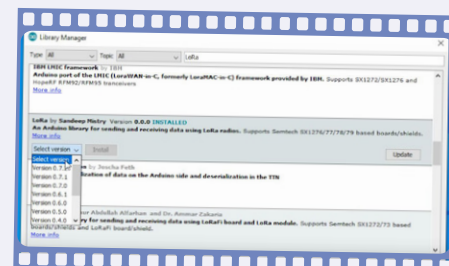
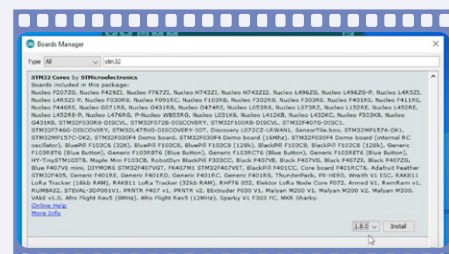
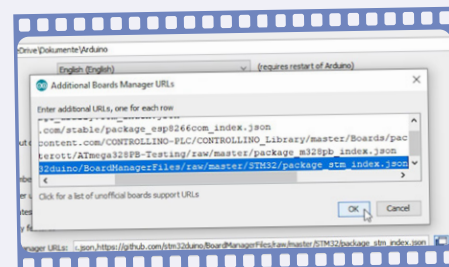


Il nous faut évidemment des antennes pour les cartes LoRa. C'est facile à fabriquer. D'après ces calculs simples que j'ai trouvés dans un article d'Elektor, un morceau de fil d'environ 82 mm fera l'affaire. J'en ai donc coupé deux que j'ai soudés directement à la broche d'antenne des cartes LoRa.

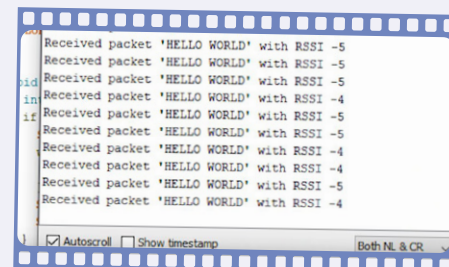


Ça y est, je suis paré pour un test, il ne reste qu'à connecter à mon ordinateur les deux cartes de prise en main.

Après avoir saisi l'URL de la bibliothèque des cartes STM32 et l'avoir installée avec le gestionnaire de cartes (Boards Manager), j'ai également installé la bibliothèque Arduino LoRa.



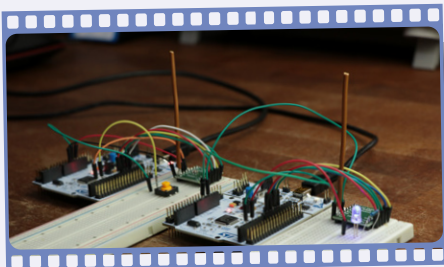
J'ai ensuite pondé un bout de code pour envoyer un simple HELLO WORLD et un autre pour que le récepteur envoie un accusé de réception au Serial Monitor en indiquant la puissance du signal reçu. Une fois les cartes Nucleo STM32 sélectionnées et les codes chargés, nous voyons que le récepteur accuse réception des messages envoyés par l'émetteur. Génial...





... mais pas fini, car l'émetteur doit envoyer un code d'alarme quand il est activé par l'alarme de mon garage, afin que le récepteur, lorsqu'il recevra ce code d'alarme correct, active à son tour un petit MOSFET qui commande une sirène.

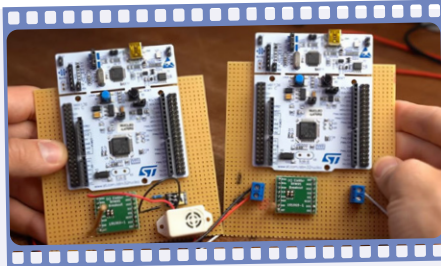
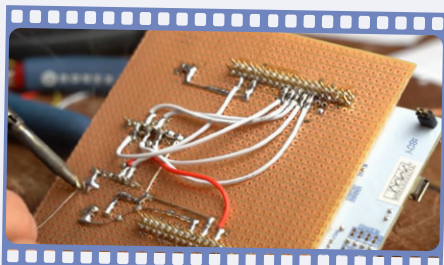
Pour faciliter les essais, j'ajoute à l'émetteur un bouton-poussoir et au récepteur une LED (et sa résistance) qui signalera la bonne réception de la notification d'alarme.



Pour le code de l'émetteur, il m'a suffi de définir un message d'alarme et de valider l'entrée du bouton poussoir pour déclencher la transmission.

Le code pour le récepteur est un peu plus compliqué : le message reçu est stocké dans un tableau pour le comparer au message d'alarme prédéfini. S'ils sont identiques, la LED sera activée.

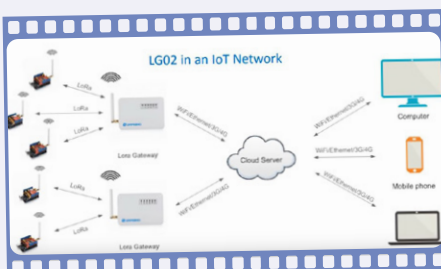
Après avoir téléchargé les deux croquis sur les cartes et effectué quelques tests, j'ai constaté que tout fonctionnait parfaitement. Il était temps de mettre au propre les schémas pour l'émetteur (fig. 1) et le récepteur (fig. 2). Après cela, pour que l'ensemble soit robuste, j'ai soudé et connecté ce petit monde sur deux circuits à pastilles.



Il m'a fallu une petite heure pour souder... quand je me suis rendu compte que j'avais complètement oublié la passerelle LoRa Gateway incluse dans les fournitures d'Elektor. J'ai commencé à douter de son utilité pour mon projet.



En bref, pour cette application simplissime, pas la peine de se démener avec les manuels et les menus du portail : elle reçoit des nœuds LoRa les données qu'elle peut déposer sur un serveur dans le nuage via votre réseau domestique. De là, ces données sont accessibles par d'autres appareils. La passerelle peut aussi envoyer ces données à d'autres nœuds LoRa.

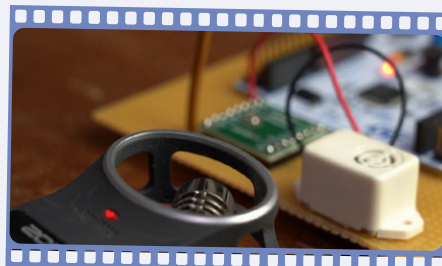


Ce procédé de diffusion de données dans le nuage est parfait pour enregistrer p. ex. de nombreuses données de température ou d'humidité pour une serre ; une telle application justifierait de mobiliser un serveur LoRa WAN, MQTT ou un serveur TCP-IP. Ces termes apparemment compliqués sont intimidants pour quelqu'un comme moi qui ne suis pas intéressé par ce genre de

choses, mais j'ai vu que le manuel d'utilisation donne d'excellentes instructions sur la façon d'utiliser ces trois méthodes, avec en prime des croquis Arduino ou STM32. Mon système est si simple, en revanche, que ces méthodes de dépôt de données sur un serveur ne sont pas justifiées. D'ailleurs, je ne veux pas dépendre de la stabilité d'une connexion à l'internet pour faire fonctionner mon système de notification d'alarme. Toutefois, ma passerelle peut envoyer ou recevoir des messages LoRa directement. Elle offre aussi la possibilité de réagir à des messages LoRa spécifiques au moyen d'un script personnalisé. Ainsi est née l'idée de l'implémenter comme une sorte de central pour mon système. J'ai passé des heures à essayer d'envoyer le bon code d'alarme et réécrit le script shell Linux personnalisé, mais j'ai abandonné, jugeant la complication inutile. Pour l'instant, je n'utilise donc pas la passerelle, mais j'y reviendrai peut-être dans le cadre d'un autre projet.

Mon récepteur est sous tension, il reste à aller au garage pour y connecter l'émetteur à mon automate Controllino, dont j'ai également dû modifier un peu le code.

Dès que j'ai déclenché l'alarme au garage, le signal sonore de la station de réception dans mon appartement s'est fait entendre. Ça maaaaarche !



Si la portée n'est pas satisfaisante, essayez de régler la puissance, le facteur d'étalement et la largeur de bande du système LoRa, sur lesquels vous pourrez vous instruire grâce au GitHub Arduino LoRa.

```
void setup() {
  pinMode(D4, INPUT_PULLDOWN);
  LoRa.setTxPower(20);
  LoRa.setSpreadingFactor(12);
  LoRa.setSignalBandwidth(62.5E3);
  Serial.begin(9600);
  if (!LoRa.begin(868100000)) {
    Serial.println("LoRa module not found");
  }
}
```

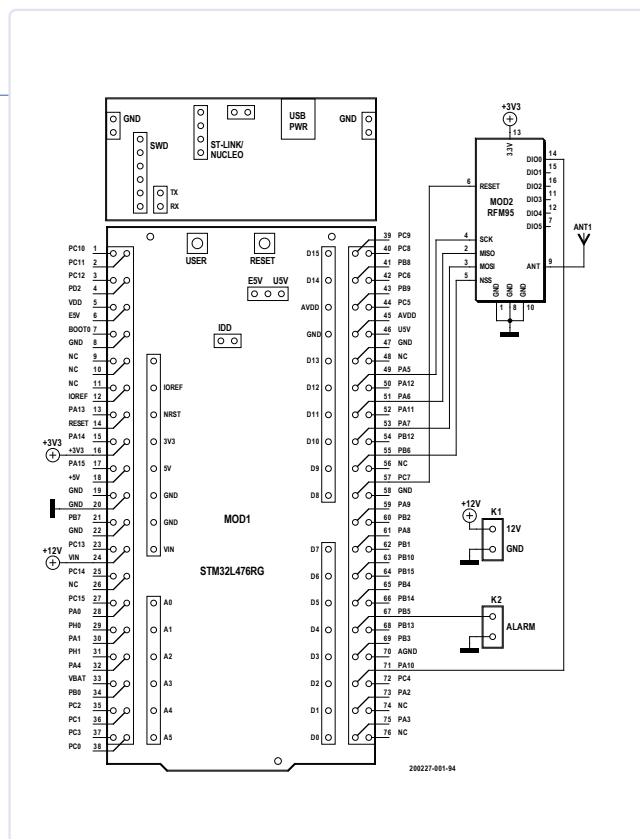


Figure 1. Schéma de l'émetteur.

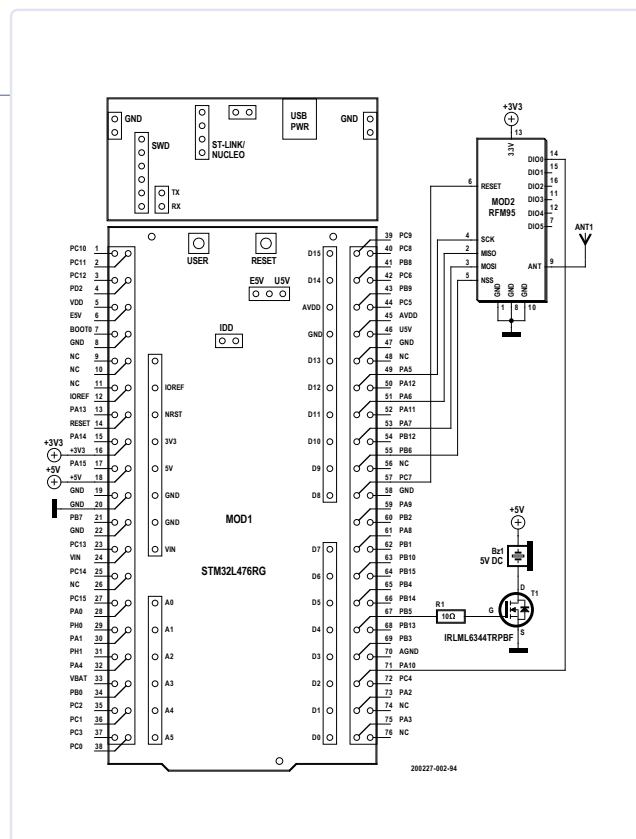



Figure 2. Schéma du récepteur.

## Conclusion (de Luc Lemmens)

GreatScott! montre ici que LoRa est une solution pour une communication simple à longue portée et à faible vitesse. Le FET sur la carte de réception du type SOT-23 (monté sur une carte de connexion) présente des spécifications exceptionnelles pour la logique 3,3 V (5 A max. et seulement 800 mV de tension de seuil !), mais pour cette application un TNO702 (TO-92) fonctionnera aussi.

Si vous souhaitez construire ce projet, les principaux composants sont disponibles dans l'e-shop d'Elektor. Dans l'encadré @ [www.elektor.fr](http://www.elektor.fr), vous noterez que la carte STM Nucleo fait partie d'un kit de démarrage, mais elle peut aussi être achetée séparément. GreatScott! a également produit sur Youtube [2] une vidéo des opérations décrites ici. 

200227-02



## Liste des composants

**Note :** composants requis pour le récepteur ET pour l'émetteur. Il faut donc 2 exemplaires de chaque composant.

Carte STM32 Nucleo L476RG  
Module émetteur-récepteur RFM95 Ultra LoRa  
Carte de connexion Elektor RFM95 191069-1  
C1 = 100 nF, 10%, 10 V, MLCC, X7R, SMD 0805  
C2 = 10 µF, 20%, 10 V, MLCC, X5R, SMD 0805  
2 connecteurs à 8 broches 2,54 mm  
fil de 82 mm (antenne)

**Composants supplémentaires pour l'émetteur :**  
K1, K2 = bornier à vis à 2 voies pour circuit imprimé au pas de 5,08 mm

**Composants supplémentaires pour le récepteur :**  
R1 = 10 Ω  
T1 = IRLML6344TRPBF (carte de connexion optionnelle) ou TNO702 (boîtier TO-92)  
Bz1 = buzzer 5 V<sub>CC</sub>



@ [WWW.ELEKTOR.FR](http://WWW.ELEKTOR.FR)

- **carte STM32 Nucleo L476RG (SKU 17946)**  
[www.elektor.fr/stm32-nucleo-l476rg-board](http://www.elektor.fr/stm32-nucleo-l476rg-board)
- **kit de démarrage Elektor STM32 Nucleo (SKU 19205)**  
[www.elektor.fr/elektor-stm32-nucleo-starter-kit](http://www.elektor.fr/elektor-stm32-nucleo-starter-kit)
- **carte de connexion RFM95 – circuit imprimé nu (191069-1) (SKU 19142)**  
[www.elektor.fr/rfm95-break-out-board-bare-pcb-191069-1](http://www.elektor.fr/rfm95-break-out-board-bare-pcb-191069-1)
- **module émetteur-récepteur RFM95 Ultra LoRa (868/915 MHz) (SKU 18715)**  
[www.elektor.fr/rfm95-ultra-lora-transceiver-module-868-915-mhz](http://www.elektor.fr/rfm95-ultra-lora-transceiver-module-868-915-mhz)
- **passerelle Dragino LG02 Dual Channels LoRa IoT (SKU 18624)**  
[www.elektor.fr/dragino-lg02-dual-channels-lora-iot-gateway](http://www.elektor.fr/dragino-lg02-dual-channels-lora-iot-gateway)

## LIENS

[1] **sketches LoRa Arduino à télécharger :** [www.elektormagazine.fr/200227-01](http://www.elektormagazine.fr/200227-01)

[2] **vidéo de GreatScott!** Transmettre un signal d'alarme avec LoRa (600m) !  
Améliorer le système d'alarme de mon garage : <https://youtu.be/ITzwa1AdrpU>

