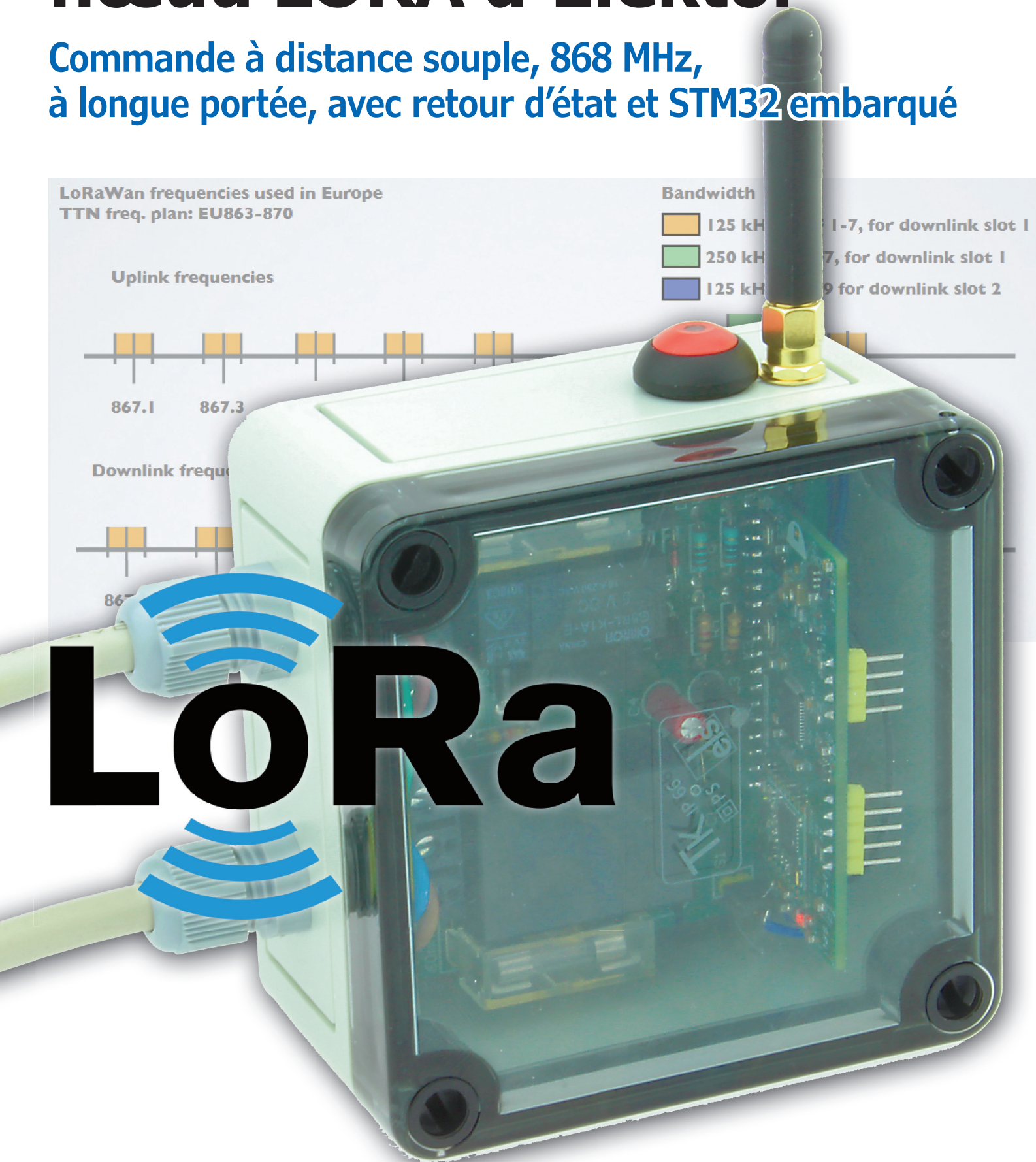


# nœud LORA d'Elektor

Commande à distance souple, 868 MHz,  
à longue portée, avec retour d'état et STM32 embarqué



**Luc Lemmens & Mathias Claussen** (Elektor Labs)

Convaincu de la puissance combinée du logiciel et du matériel ouverts, Elektor montre ici que LoRa n'est pas réservé aux professionnels. Cet article raconte l'histoire de la ténacité d'un concepteur face à des défis ardu. Il présente un nœud LoRa, facile à reproduire, pour une commande à distance avec retour d'état. La portée est de 10 à 20 fois celle du WiFi grand public ou des communications à faible puissance (LPR) dans la bande Industrie-Science-Médecine (ISM) 433 MHz.

### INFOS SUR LE PROJET



LoRa  
Arduino  
STM32



débutant  
→ **connaiss**eur  
expert



env. 4 h



UART USB,  
EDI Arduino,  
outils de labo



±150 € sans coffret

Nous partagerons ici le produit final et un bilan de qualité, mais aussi tous les problèmes résolus ainsi que les outils utilisés pour mener à son terme un projet ambitieux. Le récit aurait pu commencer comme ceci : « *ce nœud LoRa est né de l'idée d'un circuit imprimé compact alimenté par des piles ou des batteries rechargeables et embarquant un micro-contrôleur (µC) courant tel que le STM32 à boîtier LQFP48 ainsi qu'un module LoRa de type RFM95, le tout casé dans un coffret disponible chez l'un des plus gros distributeurs et quant au compilateur STM32, nous...* » **STOOOP !** Reprenons tout depuis le début, mais en mode *staccato*.

C'est le coffret qui détermine tout : volume disponible et source d'énergie, c'est donc par lui qu'il faut commencer. Idée initiale : alimentation par piles non rechargeables. Abandonnée, trop de piles partout. Recherche d'une solution à batteries rechargeables, faciles à remplacer. Objectif : nœud de petite taille, étanche, facile à maintenir, fonctionnel au labo, mais tout aussi bien en plein air, et capable de transmettre des données d'environnement réel. Autres sujets de débats : la durée de vie des batteries et la liste des capteurs supportés, embarqués ou externes. On imagine de préférence une plateforme souple et utile pour explorer LoRa et ultérieurement LoRaWAN, pour un usage d'abord académique, dans l'ambiance feutrée d'un

labo, mais capable aussi de commander à distance une fonction marche/arrêt dans n'importe quel environnement difficile. Autrement dit : la quadrature du cercle entre souplesse, consommation et encombrement.

Cet article décrit les trois éléments suivants :

- **Nœud LoRa**, carte n° 180516-1, la plateforme expérimentale LoRaWAN, équipée pour le LoRa local seulement. Il en faut deux pour le projet.
- **Bouton LoRa**, carte n° 180666-2.

- **Interrupteur de puissance LoRa** pour courant alternatif, carte n° 180666-1.

Ensemble, ces trois éléments constituent non seulement une commande à distance avec une portée de 10 à 20 fois celle d'une liaison grand public à 2,4 GHz, mais aussi, en projet, une plateforme de développement avec LoRaWAN.

C'est parti pour l'élément principal, le nœud LoRa !

## Caractéristiques (abrégées)

### Nœud Lora

- Plateforme expérimentale LoRaWAN avec équipement minimum
- Alimentation par batteries Li-Ion
- Batteries remplaçables par l'utilisateur
- Microcontrôleur STM32F072C8T6TR ARM Cortex-M0
- SPI F-RAM ou Flash (facultatif)
- Co-processeur de cryptage (facultatif)
- Module GPS (facultatif)
- Interface USB (facultatif)

### Interrupteur LoRa pour courant alternatif (esclave)

- Retour de l'état du relais
- Pouvoir de coupure 5 A (1000 W à 230 V~; 500 W à 115 V~)

### Bouton LoRa (maître)

- Faible consommation
- Afficheur OLED (facultatif)
- Antenne en hélice
- Portée vers l'esclave LoRa : 100 à 500 m



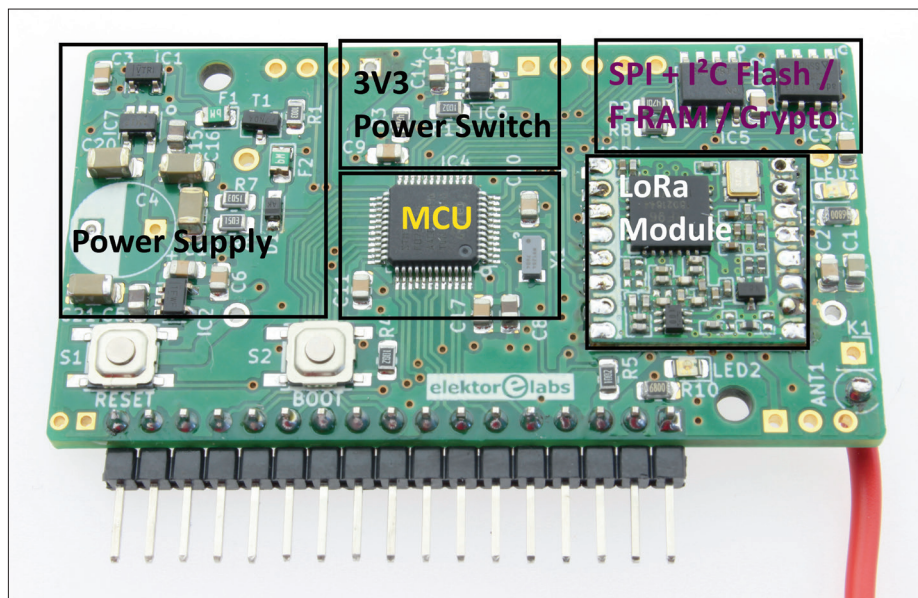


Figure 1 : La carte de gestion du nœud LoRa avec les espaces alloués aux blocs fonctionnels. En fait, ce que vous voyez, c'est une plateforme expérimentale du nœud LoRaWAN avec un équipement minimum.

### Au tout début

Le circuit imprimé a été conçu avec KiCad, un outil dont la popularité augmente, aussi chez les pros. Si vous voulez vous y mettre, Elektor vous propose un excellent livre [1].

Après un faux départ avec un *autre* processeur et des tâtonnements, nous avons conçu une carte pour le STM32F072 en boîtier LQFP48, profitant ainsi de sa compatibilité avec l'EDI Arduino. Beaucoup de bibliothèques Arduino bien connues sont utilisables. Faute de modèle de quartz convenant à notre carte, nous nous sommes rabattus sur les options d'horloge interne et la valeur par défaut de 8 MHz, afin de réduire la consommation, mais prêts à passer à 48 MHz si nécessaire.

### Approche modulaire

La version précoce de la carte du nœud LoRa d'Elektor (**fig. 1**) n'est pas seulement la base du présent projet, elle servira aussi pour des projets dérivés à publier ultérieurement. C'est pourquoi nous allons commencer par faire le tour des composants et autres options susceptibles de se retrouver sur la carte. Les lecteurs chevronnés suivront en même temps sur le schéma (**fig. 2**).

Afin de respecter certaines exigences, nous nous en sommes tenus à la modularité. Si vous implantiez tous les compo-

sants, vous obtiendriez un système LoRaWAN avec GPS, coprocesseur de cryptage et mémoire SPI F-RAM ou Flash. Impressionnant, mais pas indispensable. Les composants laissés de côté réduisent le coût de la version de base ; inutile de payer pour des fonctions que vous n'utiliserez pas. Le brochage des SPI F-RAM et des SPI Flash est standard, vous pouvez donc utiliser votre module favori.

Idem pour le coprocesseur de cryptage ATECC608a. Vous pouvez aussi vous contenter d'une puce I²C-EEPROM ou F-RAM : vous disposerez d'un système souple et configurable selon son emploi. Le microcontrôleur STM32F072 à boîtier LQFP48 présent sur la carte est une puce Cortex-M0 avec 64 ko de mémoire flash et 16 ko de RAM, avec possibilité d'utiliser l'USB pour les mises à jour du progiciel et vos propres applications.

Le brochage de tous les membres de la famille STM32 est compatible ; vous pouvez donc changer de modèle sans modifier la carte. Ce peut être le STM32L072 à faible consommation, ou bien le STM32L151 pour sa puissance de calcul et sa mémoire Flash. Même le STM32F103, qu'on trouve habituellement sur la carte *Bluepill*, peut y prendre place.

L'encadré *Caractéristiques (abrégées)* est une (tentative de) démonstration de la souplesse d'utilisation de la carte.

### Conception du circuit imprimé et défis

Après les premiers prototypes, des ajustements s'imposaient. Pour le module GPS, il a fallu améliorer la réception. Ajouter une petite antenne ? Il faut alors tenir compte du plan de masse. Le module GPS exige une taille minimum du circuit imprimé pour que l'antenne fonctionne. Pas solution satisfaisante en vue pour améliorer la qualité de la réception.

Après la version 1 le module Quectel L96 GPS a été remplacé par le module vendu par Elektor [2] moins cher et plus facile à connecter. Modification consécutive des connecteurs utilisés, de sorte que tout est maintenant au pas de 2,54 mm, compatible avec une carte d'expérimentation.

Nous avons ensuite le module LoRa RFM95W qui ne se passe pas d'antenne. Pour l'encombrement, nous voulions une antenne maison hélicoïdale pour 868 MHz. Pour améliorer la réception, une antenne extérieure est une option aussi. La leçon à retenir : la conception est un compromis entre encombrement, fonctionnalité et performances radio.

Les batteries Li-Ion au format 10440 (AAA), comme on en trouve dans les e-cigarettes, sont faciles à remplacer par l'utilisateur et à recharger hors l'appareil par un chargeur approprié. Pour les protéger d'une décharge profonde, nous avons inclus une puce de surveillance de tension qui coupe le régulateur à faible chute de tension (LDO) de l'alimentation 3,3 V.

Un connecteur femelle aux broches de 2 mm permet de connecter des paquets de batteries LiPo utilisées sur les drones. Plats, de petite taille, équipés d'un connecteur Molex 510005 ou JST, ils ont un ratio capacité/volume intéressant. Avantages : souplesse d'utilisation, batteries standard, faciles à trouver. La protection contre l'inversion de courant a été réalisée avec un *interrupteur de charge*. Nous avons fini par utiliser la *diode idéale* MAX40200, qui coupe l'alimentation dès qu'un certain seuil de courant inverse est atteint. Une seconde, sur l'alimentation du module GPS (K4), permet au µC de couper l'alimentation de ce module pour ménager les batteries. Si vous n'utilisez pas ce module, vous pouvez utiliser cette ligne d'alimentation 3,3 V pour commander la mise sous tension d'un circuit extérieur.

Il est en principe possible d'alimenter le système par des batteries NiMH, mais

au prix d'un circuit supplémentaire, ici un convertisseur continu-continu élévateur. Comme ce module consommera

en permanence, cela aura un effet sur le temps de fonctionnement, et il nous faudra aussi revoir la protection de

sous-tension.

Nos défis majeurs semblent maintenant se concentrer dans le logiciel.

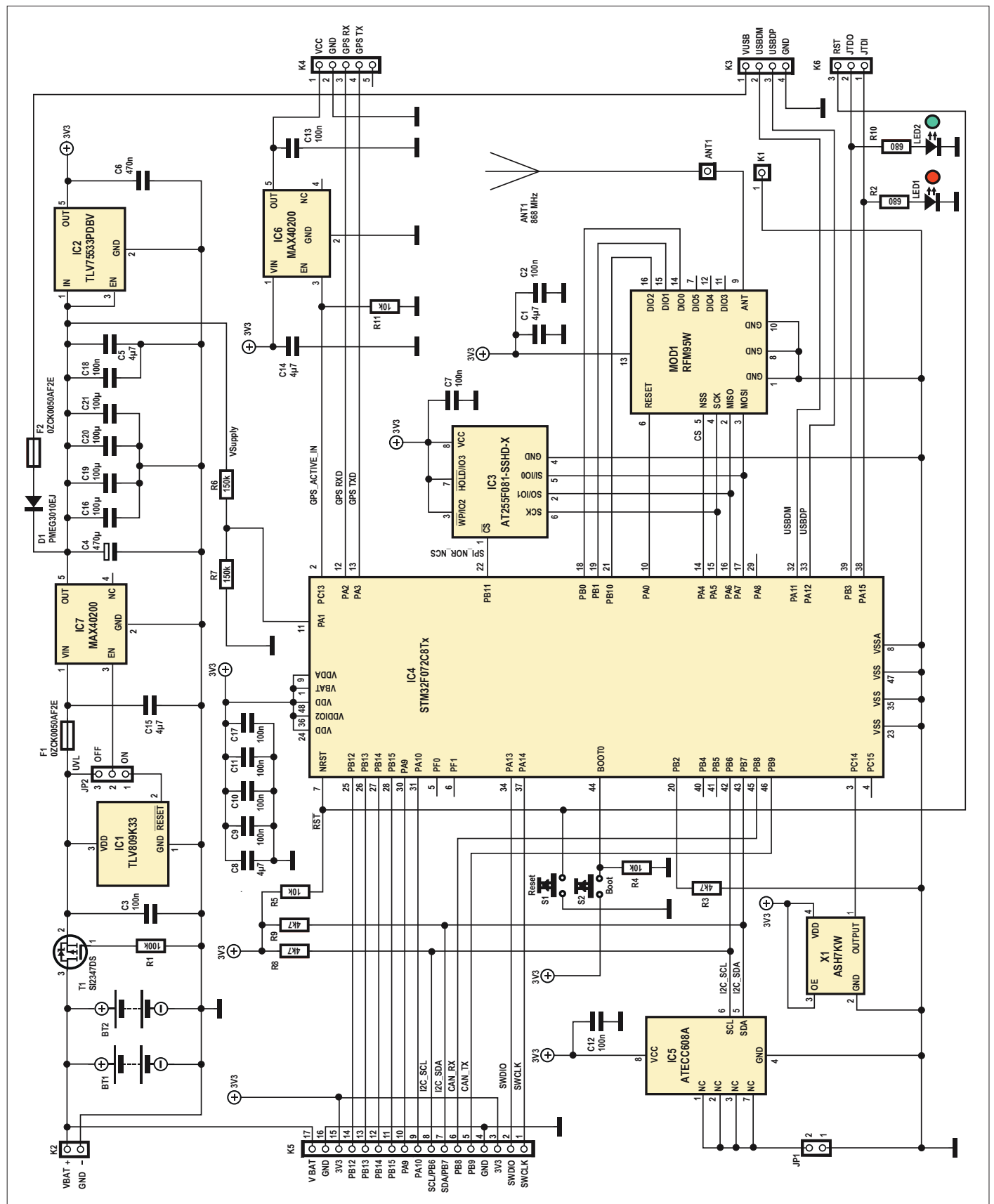


Figure 2 : Schéma de la carte de gestion du nœud LoRa. Ceci est fondamentalement la plateforme expérimentale du nœud LoRaWan équipée d'une série de puces spécifiques.

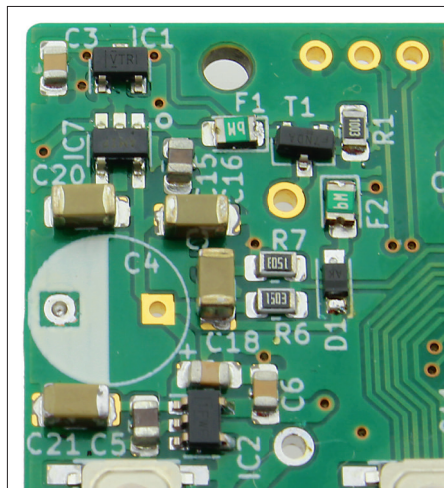


Figure 3 : L'espace de la carte réservé au condensateur électrolytique C4, l'un des composants encombrants.

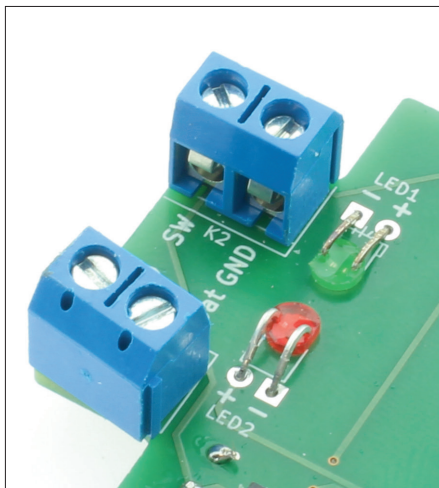


Figure 5 : Les LED sont montées de manière inhabituelle sur la carte bouton LoRa.

Aucunement intimidés, nous tentons la communication de point à point en mode RAW proposée par la bibliothèque LMIC. Problèmes au niveau des compo-

sants, avec un condensateur de 470  $\mu\text{F}$  monté en amont du circuit de protection de sous-tension IC1. Si la tension tombe en dessous de 2,97 V, ce circuit

coupe l'interrupteur HS IC7, ce qui interrompt l'alimentation pendant 20 ms. Si le condensateur tampon est insuffisant, les courants de démarrage élevés perturbent le contrôleur de sous-tension (UVLO) et sa tension d'entrée tombe en dessous de 2,97 V, ce qui provoque une oscillation permanente de conduction-blocage du régulateur LDO. Problème résolu par le *gros* électrolytique C4 (**fig. 3**). Ne l'ajoutez que si les deux condensateurs solides à céramique multicouche (MLCC) de 100  $\mu\text{F}$  vous paraissent hors de prix. Les feuilles de caractéristiques montrent aussi qu'il n'y a pas de FET ou de diode qui empêcherait les batteries d'être chargées par la tension fournie par l'USB. La MAX40200 coupe l'alimentation s'il circule un courant inverse. C'est également le cas si la tension USB est supérieure à la tension fournie par les batteries du système.

Les deux LED indiquent l'état de la carte fourni par le  $\mu\text{C}$  sur ses ports PB3 et PA15, soit 'JTDO' et 'JTDI' sur le connecteur K6.

### Logiciel : les fondations

Le logiciel du système a beau déterminer les fonctions du matériel dans des proportions toujours croissantes, certains n'y voient rien d'autre qu'un peu de magie noire à flasher dans une puce. Comment se fait-il que le temps passé sur le logiciel l'emporte à ce point sur celui passé sur le matériel ?

Vous pouvez programmer la carte avec votre EDI Arduino. Pour cela, le projet *stm32duino* [3] est d'une grande utilité, car il apporte le support d'une gamme étendue de  $\mu\text{C}$  et de cartes STM32.

Y ajouter dans les règles la plateforme expérimentale du nœud LoRa d'Elektor est la première tâche administrative : éditer quelques fichiers du noyau Arduino, ajouter la définition de la carte. Après quelques erreurs et comportements imprévus du logiciel, le canevas fonctionne, nous pouvons écrire les premières lignes de code.

D'habitude on ne se préoccupe guère de la façon dont le  $\mu\text{C}$  gère son horloge. Sur les puces AVR, on avait un quartz, et voilà. Les  $\mu\text{C}$  récents ont davantage d'options pour l'horloge principale (PLL / FFL ou oscillateurs RC à haute fréquence internes). Pour commencer, la stabilité des sources internes suffira. La réduction de la fréquence d'horloge en cours de fonctionnement permet de réduire la consommation, tandis qu'une fréquence élevée augmente la puissance

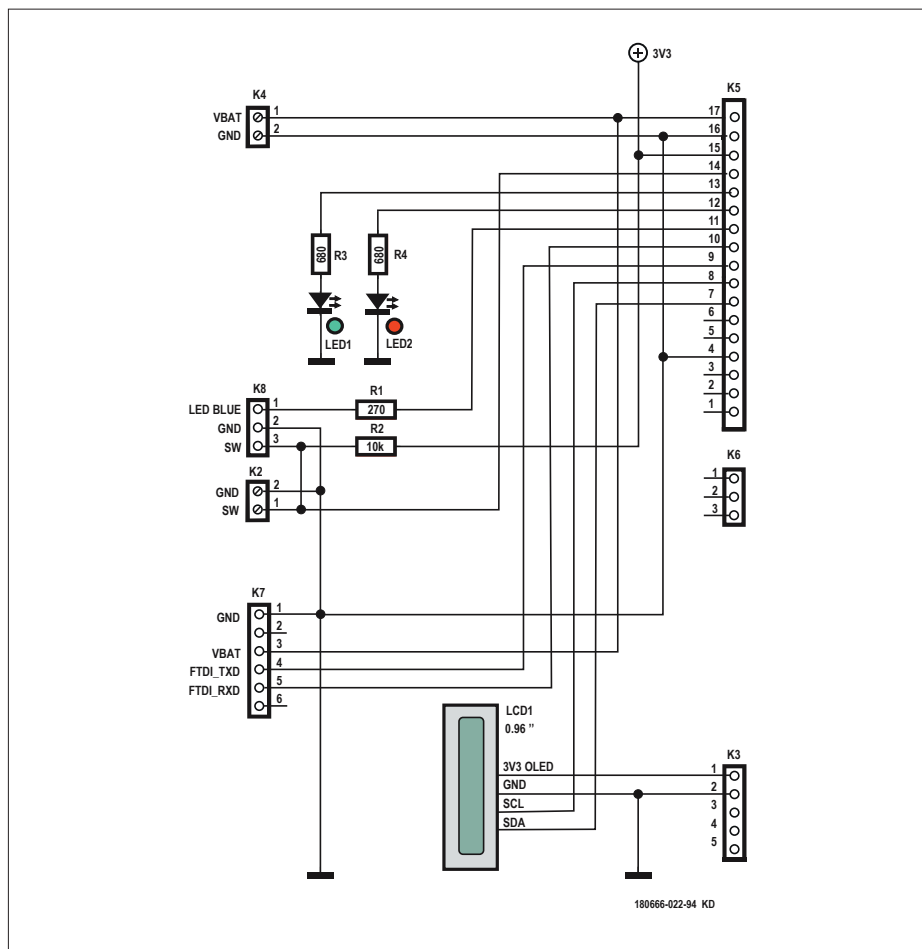


Figure 4 : Schéma du bouton LoRa. C'est la partie «émission» ou «maître» du nœud LoRa d'Elektor. L'afficheur OLED est facultatif et son usage devrait être subordonné à une réflexion approfondie sur le bilan énergétique !



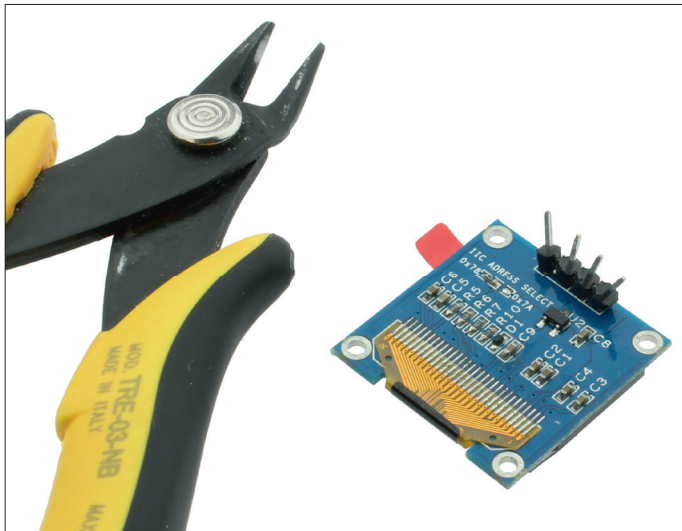


Figure 6 : Utilisez des pinces bien coupantes pour tronquer le plastique entre les broches.

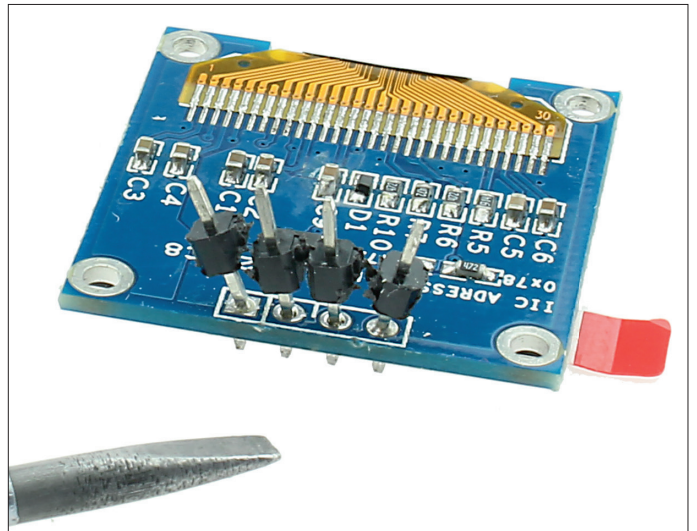


Figure 7 : Avec la pointe d'un tournevis, dégagez et enlevez les morceaux de plastique.

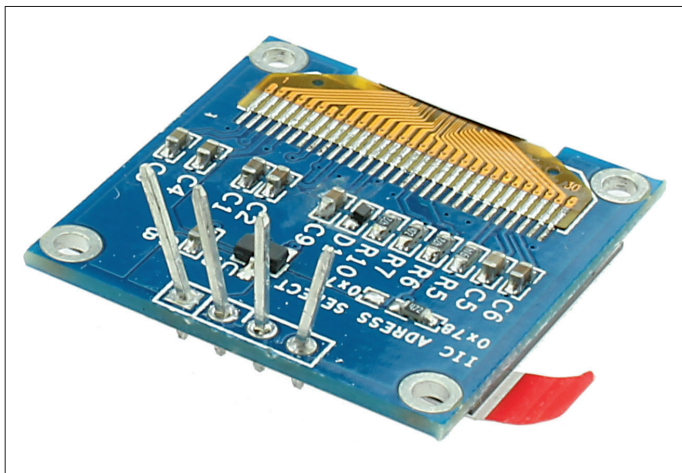


Figure 8: Vous avez maintenant une rangée de broches dénudées.

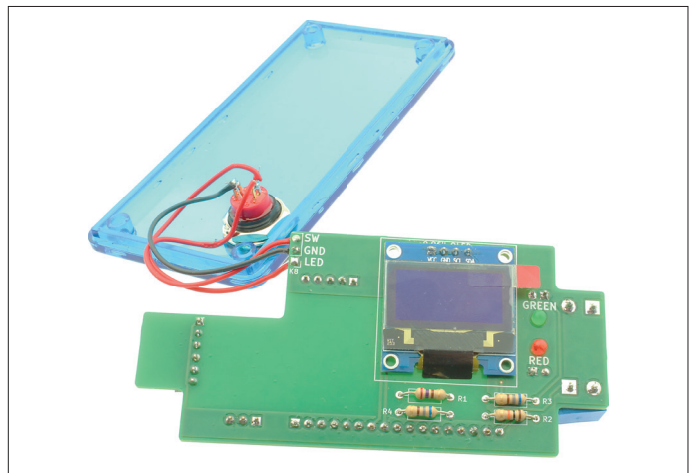


Figure 9 : La carte bouton LoRa au complet.

de calcul. La valeur par défaut de 8 MHz pour le cœur et les périphériques limite la consommation, mais reste suffisante pour la pile LoRa.

Pour l'horloge en temps réel, nous avons ajouté un oscillateur à 32768 kHz (X1) facultatif. Pour un  $\mu A$  de courant en plus, on a des UART cadencés par quartz ainsi qu'une horloge en temps réel capable de réveiller le  $\mu C$  après un délai donné.

### Le bouton LoRa

Ceci est un simple organe de saisie, qui transmet son état par radio ; c'est le second composant du projet, le bouton LoRa (**fig. 4**) constitué d'un circuit imprimé ne comportant que quelques

résistances, une paire de LED, des borniers à vis et des barrettes mâles et femelles. Ces composants sont essentiels pour la version de base du nœud LoRa d'Elektor. L'afficheur OLED de la boutique Elektor [**3**] (composant : LCD1) est facultatif. Attention à l'assemblage, notamment pour les LED, retournées et insérées dans des trous du circuit, de sorte qu'elles émergent du côté inférieur de la carte (**fig. 5**).

Pour l'afficheur OLED, il faut s'y prendre avec précaution, car il va falloir soit dessouder complètement sa barrette de broches, soit la bricoler en enlevant le plastique autour des broches. Pour cela, commencez par couper le plastique entre

les broches (**fig. 6**), puis dégagez les morceaux un à un avec la pointe d'un tournevis (**fig. 7**). Si tout va bien, les broches sont alors dénudées (**fig. 8**).

On peut alors embrocher l'afficheur à plat dans la position prévue, suivi des barrettes à broches, des barrettes coudées et des borniers à vis. Cela termine l'assemblage, sauf le vrai bouton, bien sûr (**fig. 9**).

### Interlude: une antenne 868 MHz maison

Pour émettre et recevoir à 868 MHz, le bouton LoRa doit être équipé d'une antenne. Une antenne quart d'onde 'Marconi' en fil de cuivre émaillé de 1 mm

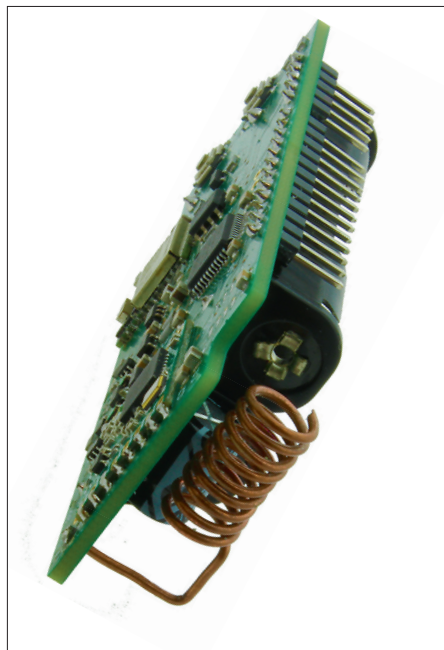


Figure 10 : Le point chaud de l'antenne quart d'onde 868 MHz maison est soudé sur la pastille ANT de la carte (ici, un des premiers prototypes). Remarquez la forme hélicoïdale de l'antenne.

est une bonne solution, peu coûteuse. Avec un bon plan de masse, le fil doit avoir une longueur théorique dans le vide de 8,635 cm, soit 8,2 cm en pratique. Cela est dû au coefficient de célérité de l'air, qui est de 0,95 [4]. Si l'espace disponible le permet, le fil peut être droit, équivalent à une antenne fouet, ou bobiné avec 1 mm entre les spires.

Dans les deux cas, le fil est soudé sur la pastille ANT. (fig. 10).

### L'interrupteur esclave LoRa pour courant alternatif

Comme exemple d'organe de sortie asservi par le nœud LoRa d'Elektor, aussi à l'écoute du bouton LoRa, nous avons un interrupteur de puissance pour appareils alimentés par le secteur, jusqu'à 1 kW. L'état de l'interrupteur (ouvert/fermé) est lu par le nœud LoRa à travers un coupleur optique. La carte embarque aussi un convertisseur 100-230 V~ - 5V=, pour l'alimentation directe par le secteur du nœud LoRa et de l'interrupteur esclave.

La véritable intelligence de l'interrupteur LoRa esclave réside dans le matériel et le logiciel du nœud LoRa, lequel se présente sous la forme d'une carte embrochable. L'interrupteur est toutefois doté des fonctions suivantes :

- rail d'alimentation 5 V=
- interrupteur de puissance 115-230 V~, 1 kW max.
- retour de l'état de l'interrupteur
- bouton-poussoir local marche/arrêt

Voyons les composants principaux de l'interrupteur LoRa esclave et leur assemblage sur le schéma (fig. 11).

**Alimentation.** Solution compacte : Un convertisseur alternatif/continu 5 V/2 W de chez *Mean Well*, sûr et relativement

bon marché. La documentation ne prescrit aucun fusible côté alternatif, mais nous avons ajouté F1 par précaution. La résistance NTC (à coefficient de température négatif) R1 limite la pointe de courant au démarrage, qui peut être élevée pour ce genre d'appareil. C1, C2 et le filtre de mode commun L1 en sortie du convertisseur peuvent paraître exagérés, parce que les affirmations de *Mean Well* au sujet de l'élimination des perturbations électromagnétiques ne nous ont pas convaincus.

**Commutation de charge.** Les relais bistables de la famille G5RL-K1 de *Omron* ont deux bobines, l'une pour l'ouverture, l'autre pour la fermeture. Une seule impulsion sur l'une d'entre elles suffit pour faire changer le relais d'état. Les contacts supportent 16 A, mais pour éviter de faire fondre les pistes du circuit imprimé le fusible F2 ne laissera passer que 5 A .

**Retour de l'état du relais.** En théorie, le  $\mu$ C sur la carte du nœud LoRa devrait garder en mémoire la dernière commande envoyée au relais et savoir s'il est ouvert ou fermé. Pour être sûr, un retour de son état n'est pas un luxe, ne serait-ce que parce que l'état d'un relais bistable est indéterminé après l'initialisation. Le circuit autour de l'optocoupleur IC1 n'indique que l'état de RE1 et la présence du secteur sur le bornier K2. Il ne dit rien sur la charge ni sur la puissance consommée, dont il faut s'assurer par ailleurs.

**Commande manuelle.** Le but est de commuter une charge à (grande) distance en appuyant sur un bouton, mais si vous êtes à proximité du récepteur, il est souhaitable de pouvoir la commuter manuellement sans avoir besoin du bouton LoRa, ni du nœud LoRa. De plus, en cas de perte de la connexion LoRa, quelle qu'en soit la raison, cette fonction viendra à point.

**Assemblage et installation.** Le soudage de la carte pose peu de problèmes. Sauf le filtre de mode commun et (peut-être) le connecteur K3, c'est une promenade de santé avec rien que des composants traversants. Les pastilles pour K3 sont de bonne taille et même L1 ne posera pas de problème.

K3 n'utilise que sept des 16 broches. Il a deux options, décrites plus loin, qui toutes les deux supposent

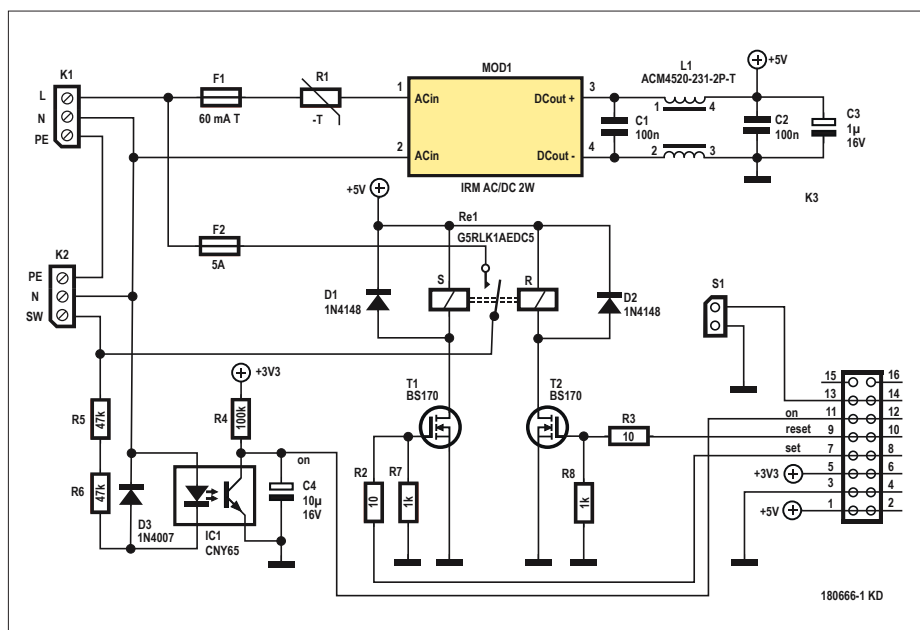


Figure 11 : Schéma de l'interrupteur LoRa. Ce n'est pas un élément passif : il renvoie l'état du relais au nœud LoRa.



## LISTE DES COMPOSANTS

### Pour la plateforme expérimentale LoRaWAN

Note: 2 exemplaires requis,  
1 pour l'interrupteur, 1 pour le bouton

#### Résistances

R1 = 100 k $\Omega$ , couche épaisse, 5%, 0,1 W, 150 V  
R2, R10 = 680  $\Omega$ , couche épaisse, 5%, 0,1 W, 150 V  
R3, R8, R9 = 4,7 k $\Omega$ , couche épaisse, 5%, 0,1 W, 150 V  
R4, R5, R11 = 10 k $\Omega$ , couche épaisse, 5%, 0,1 W, 150 V  
R6, R7 = 150 k $\Omega$ , couche épaisse, 5%, 0,1 W, 150 V

#### Condensateurs

C1, C5, C8, C14, C15 = 4,7  $\mu$ F, 16 V, X7R, SMD 0805  
C2, C3, C7, C9, C10, C11, C12, C17, C18 = 100 nF, 50 V, X7R, SMD 0805  
C4 = voir texte  
C6 = 470 nF, 25 V, X7R, SMD 0805  
C16 = omis  
C20 = omis  
C19, C21 = 100  $\mu$ F, 10 V, SMD 1206 [3216 Metric],  $\pm$  20%, X5R

#### Semi-conducteurs

D1 = PMEG3010EJ, 115, diode, 30 V, 1 A

LED1 = rouge, faible consommation, SMD 0805

LED2 = verte, faible consommation SMD 0805

T1 = SI2347DS, MOSFET, p-channel, 5 A, 30 V, 0,033  $\Omega$

IC1 = TLV809K33DBVR, surveillance de tension

IC2 = TLV75533PDBVR, 3V3 LDO, 500 mA, low- $I_{OQ}$ , SOT-23-5

IC3 = (facultatif) espace pour AT25SF081-SSHD-T 8 Mbit flash

IC4 = microcontrôleur STM32F072C8T6TR ARM Cortex-M0

IC5 = (facultatif) espace pour ATECC608A-SSHDA-B, CryptoAuthentification

IC6, IC7 = MAX40200AUK+T contrôleur 'diode idéale', 1 voie, 1 A

MOD1 = RFM95W-868S2 LoRa transceiver, e-choppe Elektor SKU 18715

#### Divers

X1 = 32,768 kHz module oscillateur, SMD, 3,2 mm x 1,5 mm

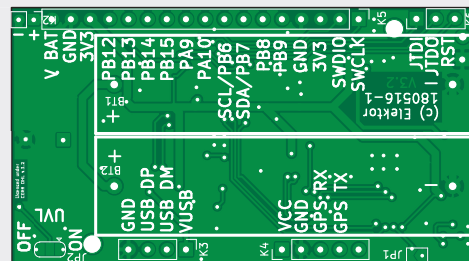
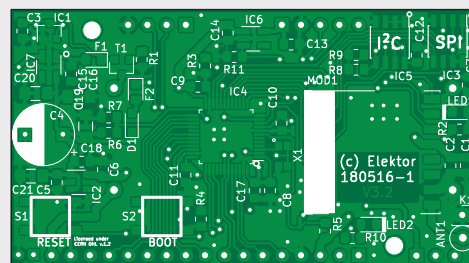
(Abracon ASH7KW-32.768KHZ-L-T)

S1, S2 = bouton-poussoir à retour tactile, 12 V, 50 mA (Multicomp TM-5531-Q-T/R)

Bt1, Bt2 = (facultatif) espace pour support de batteries AAA à broches pour circuit imprimé (Multicomp MP000341)

F1, F2 = 500 mA PPTC fusible réarmable (Bel Fuse OZCK0050FF2E)

K1 = broche pour circuit imprimé



K2 = connecteur à 2 broches

K3 = connecteur à 4 broches SIL

K4 = connecteur à 5 broches

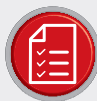
K5 = connecteur à 17 broches

K6 = connecteur à 3 broches

K5, K6, K3, K4 = connecteur à 40 broches SIL

ANT1 = antenne en fil émaillé 1 mm, longueur 8,2 cm

circuit imprimé 180516 V3.1  
de l'e-choppe Elektor



## LISTE DES COMPOSANTS

### Liste des composants

Pour la carte interrupteur LoRa

#### Résistances

R1 = NTC, 120  $\Omega$ , Epcos type B57236S0121M000  
R2, R3 = 10  $\Omega$ , carbone, 5%, 0,25 W, 250 V  
R4 = 100 k $\Omega$ , carbone, 5%, 0,25 W, 250 V  
R5, R6 = 47 k $\Omega$ , carbone, 5%, 0,25 W, 250 V  
R7, R8 = 1 k $\Omega$ , carbone, 5%, 0,25 W, 250 V

#### Inductance

L1 = ACM4520V-231-2P-T filtre de mode commun (Farnell # 2455201)

#### Condensateurs

C1, C2 = 100 nF, 50 V, X7R, pas 5,08 mm  
C3 = 1  $\mu$ F, 16 V, radial, 5 mm  
C4 = 10  $\mu$ F, 16 V, radial, 5 mm

#### Semi-conducteurs

D1, D2 = 1N4148  
D3 = 1N4007  
T1, T2 = BS170

IC1 = CNY65 optocoupleur

#### Divers

RE1 = G5RLK1AEDC5 relais de puissance bistable, à deux bobines, 5 V, 16 A, SPST (Omron)

K1, K2 = bornier à vis à 3 points, pas 7,62 mm, 500 V

K3 = connecteur femelle à 2 rangées, 16 contacts, pas 2,54 mm, série WR-PHD, type CMS (Würth # 610316243021)

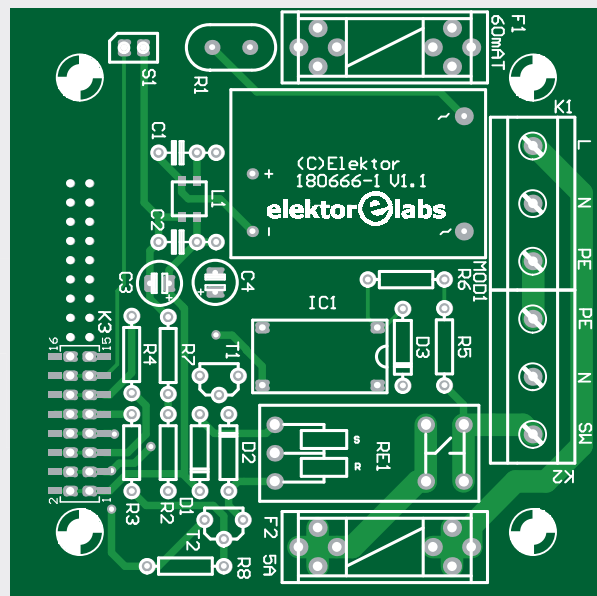
S1 = bouton-poussoir, classe IP68 (Alcoswitch PB6B2FM3M1CAL07)

MOD1 = alimentation secteur encartable 5 V, 200 mA (Mean Well IRM-02-5)

F1, F2 = porte-fusible encartable 5x20 mm

F1 = fusible 60 mA, 20 mm

F2 = fusible 5AT, 20 mm



circuit imprimé 180666-1 V1.1

e-choppe Elektor

coffret HAMMOND 1591-ATBU (Ice Blue)





## LISTE DES COMPOSANTS

### Pour la carte bouton LoRa

#### Résistances

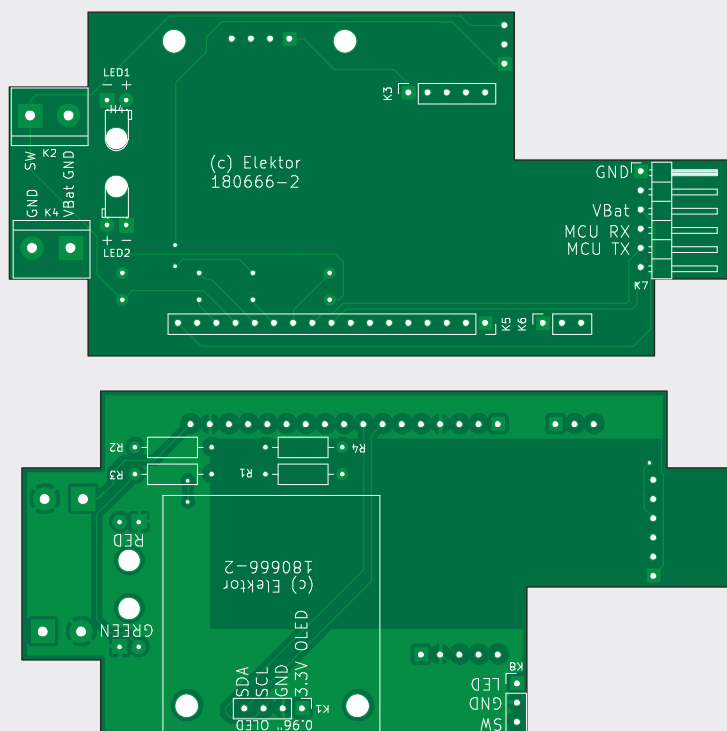
R1 = 270  $\Omega$ , couche carbone, 5%, 0,25 W, 250 V  
R2 = 10 k $\Omega$ , couche carbone, 5%, 0,25 W, 250 V  
R3, R4 = 680  $\Omega$ , couche carbone, 5%, 0,25 W, 250 V

#### Semi-conducteurs

LED2 = rouge, 3 mm  
LED1 = verte, 3 mm

#### Divers

K2, K4 = bornier à vis à 2 bornes, pas 5,08 mm, 630 V  
K3 = barrette sécable à 1 rangée de 5 broches verticales  
K5 = barrette sécable à 1 rangée de 17 broches verticales (voir texte)  
K6 = barrette sécable à 1 rangée de 3 broches verticales  
interrupteur à LED bleue intégrée (facultatif)  
circuit imprimé 180666-2 V1.2  
e-choppe Elektor



coffret Spelsberg  
type TK PS 94 x 94 x 57 mm, IP66

que la carte du nœud LoRa possède en position K5 un connecteur soudé au pas de 2,54 mm. Les sept broches des positions 11 à 17 (voir la carte 180516-1) suffisent, mais K3 peut aussi être une barrette de 17 broches (tronquée) occupant la totalité de K5.

Le connecteur (*Würth Elektronik*) identifié K3 dans la liste des composants est moins facile à trouver qu'une barrette. Nous l'avons néanmoins utilisé à cause de son profil bas grâce auquel deux cartes, montées à angle droit, tiennent

dans le coffret choisi. Tant pis pour la deuxième rangée de contacts inutilisée, faute de connecteur à profil assez bas avec une seule rangée.

L'espace sur la carte pour K3 permet d'utiliser aussi des barrettes à broches traversantes avec un coffret plus profond.

Vous pouvez aussi vous passer de K3 et souder directement les broches coudées du nœud LoRa sur le circuit imprimé, mais il sera impossible dès lors de séparer les deux cartes.

Le bouton-poussoir S1 est monté sur le coffret et relié au circuit imprimé par deux fils.

### Installation du logiciel

Le paquet de logiciel complet du nœud LoRa d'Elektor est disponible à l'endroit prévu [6]. Le fichier d'archive contient aussi une procédure d'installation omise ici faute de place.

Pour développer du logiciel pour la carte nœud LoRa vous pouvez choisir entre la méthode rude de la programmation en C et l'utilisation du STMCubeIDE, ou

### Liens

- [1] Livre KiCAD Like a Pro : [www.elektor.com/kicad-like-a-pro](http://www.elektor.com/kicad-like-a-pro)
- [2] Module GPS de l'e-choppe Elektor : [www.elektor.com/open-smart-gps-serial-gps-module-for-arduino-apm2-5-flight-control](http://www.elektor.com/open-smart-gps-serial-gps-module-for-arduino-apm2-5-flight-control)
- [3] Projet stm32duino : [https://github.com/stm32duino/Arduino\\_Core\\_STM32](https://github.com/stm32duino/Arduino_Core_STM32)
- [4] Coefficient de célérité : [https://en.wikipedia.org/wiki/Velocity\\_factor](https://en.wikipedia.org/wiki/Velocity_factor)
- [5] Programmeur STM32Cube : [www.st.com/en/development-tools/stm32cubeprog.html](http://www.st.com/en/development-tools/stm32cubeprog.html)
- [6] Logiciel du projet : [www.elektormagazine.com/180666-01](http://www.elektormagazine.com/180666-01)
- [7] Ajouter des cartes STM32 : <https://github.com/stm32duino/wiki/wiki/Getting-Started>

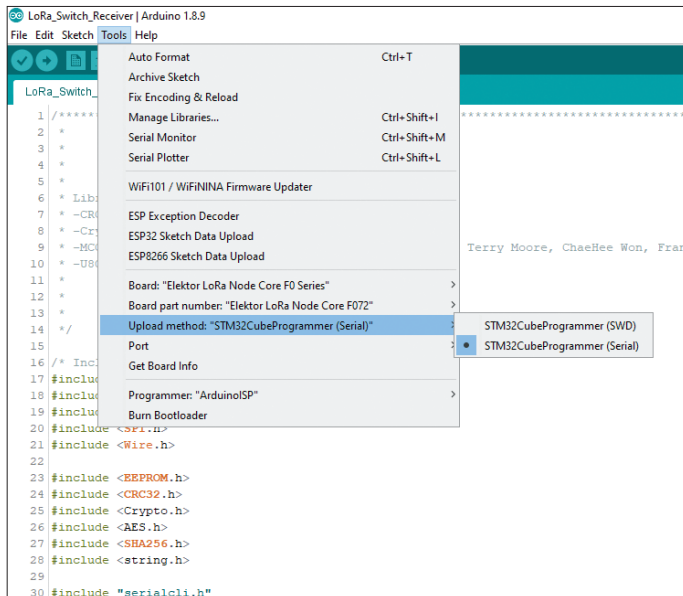


Figure 12 : Observez et copiez ces réglages de configuration de la carte.

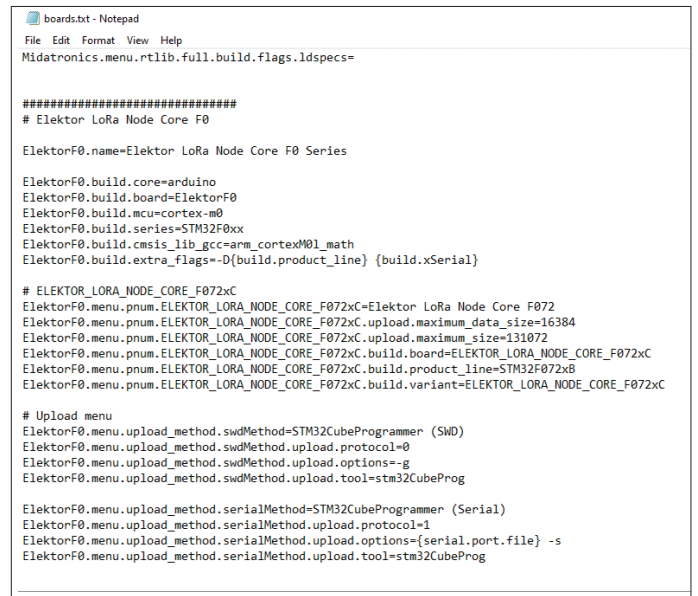


Figure 13 : Elektor a obtenu l'approbation du nœud LoRa (sur la *LoRaWAN Node Experimental Platform*) et son admission dans la famille STM32Arduino. Le logiciel du projet est bon pour le service, nos lecteurs n'auront pas à se préoccuper pas de cette procédure administrative.

bien vous faciliter la tâche avec l'EDI Arduino. Pour cette dernière, il vous faut télécharger l'EDI Arduino ainsi que le noyau STM32 Arduino. Dans l'EDI Arduino, ajoutez l'URL suivant dans le gestionnaire de cartes supplémentaires (Fichier → Préférences → Paramètres) : [https://github.com/stm32duino/BoardManagerFiles/raw/master/STM32/package\\_stm\\_index.json](https://github.com/stm32duino/BoardManagerFiles/raw/master/STM32/package_stm_index.json) et installez les cartes STM2 en suivant les instructions [7]. Vous avez maintenant le noyau de STM32, mais pour le nœud LoRa d'Elektor, il faut encore ajouter quelques petites choses à votre système de fichiers local. Sous Windows, allez à :

`%localappdata%\Arduino15\packages\STM32\hardware\stm32\1.7.0`

et ajoutez à la fin de «Boards.txt» le contenu du fichier *Add\_To\_Board.txt* que nous avons préparé pour vous. Copiez également le répertoire :

`ELEKTOR_LORA_NODE_CORE_F072xC`

dans le répertoire «Versions». À ce stade, vous êtes opérationnel et prêt à compiler le logiciel pour la carte. À la fin, il vous faudra définir la configuration de la carte sur la **figure 12**. Maintenant, vous êtes prêt à compiler du logiciel pour la carte. Pour effectuer

un téléchargement vers la carte, il faut installer le STM32CubeProgrammer [5]. Si vous voulez maintenant télécharger du code, mettez la carte en mode *bootloader* en appuyant sur le bouton *Boot* puis sur le bouton *Reset*. Vous voilà prêt pour un nouveau téléchargement de progiciel vers la carte.

### À votre service

Au début de cet article, nous annonçons que cette histoire serait racontée dans un ordre à peu près chronologique. Depuis la mi-décembre 2019, les définitions de la carte de la plateforme expérimentale LoRaWAN d'Elektor ont été intégrées

au dépôt Git officiel STM32Arduino, en respectant les conditions assez strictes fixées par le comité *STM32duino* (**fig. 13**, merci, les gars !).

En conséquence, vous n'avez plus besoin de faire vous-même aucune des modifs décrites ci-dessus, vous n'avez plus qu'à choisir la carte nœud LoRa d'Elektor dans la liste des cartes approuvées. C'est parti ! Gageons qu'on entendra encore parler ici du nœud LoRa d'Elektor. ◀

(180666-02 Version française : Helmut Muller)



## WWW.ELEKTOR.FR

- Nœud LoRa d'Elektor, 180516 V3.1  
[www.elektor.fr/180516-1](http://www.elektor.fr/180516-1)
- Carte Bouton LoRa, 180666-2 v 1.2  
[www.elektor.fr/180666-2](http://www.elektor.fr/180666-2)
- Carte Interrupteur de courant alternatif LoRa, 180666 1 V 1.1.  
[www.elektor.fr/180666-1](http://www.elektor.fr/180666-1)
- RFM95W-868S2 Émetteur-récepteur LoRa  
[www.elektor.fr/18715](http://www.elektor.fr/18715)
- Livre KiCAD Like a Pro  
[www.elektor.fr/kicad-like-a-pro](http://www.elektor.fr/kicad-like-a-pro)
- Module GPS (facultatif)  
[www.elektor.fr/open-smart-gps-serial-gps-module-for-arduino-apm2-5-flight-control](http://www.elektor.fr/open-smart-gps-serial-gps-module-for-arduino-apm2-5-flight-control)