



station météo LoRa à faible puissance

réalisez vous-même une station météo à longue portée

Edward Ringel (États-Unis)

Lorsque mon ancienne station de température et d'humidité à distance a fini par tomber en panne, je l'ai remplacée par un système de ma propre conception : un capteur à l'extérieur et un affichage à l'intérieur tous deux alimentés par batterie et connectés via LoRa. Mettre au point un dispositif fiable capable de supporter l'hiver du nord-est des États-Unis n'a pas été simple ! Mais voyez plutôt...



Figure 1. Capteur enneigé suspendu près de notre voie d'accès.

La fonctionnalité minimale du système était le développement d'un capteur extérieur distant, alimenté par piles (**figure 1**), et d'un affichage intérieur, lui aussi équipé d'un capteur et alimenté par piles. Pendant le développement du projet, j'ai ajouté un serveur web très simple.

Pour ce faire, je voulais utiliser l'environnement Arduino et ses nombreuses bibliothèques de haute qualité. Pour simplifier le projet, je prévoyais d'utiliser des composants sur étagère et qu'on puisse souder à la main. Enfin, la réduction de la consommation d'énergie était une priorité. Imprévue au départ, la climatisation efficace du capteur extérieur s'est avérée étonnamment difficile.

Des considérations générales m'ont permis de déterminer quel matériel / quelles cartes devaient être utilisés – voir plus loin pour en savoir plus et les difficultés rencontrées. Tout d'abord, il faut une station extérieure, une station intérieure et une station de base. Cette dernière collecte les données des deux stations de mesure et les transmet à un ordinateur. Les trois unités ont donc besoin d'au moins une carte microcontrô-

leur et d'une unité radio pour communiquer entre elles. Les deux stations de mesure pour l'extérieur et l'intérieur ont également besoin de capteurs et d'une commande de minuterie, dont il sera question plus loin. Les trois stations sont constituées des « briques de base » suivantes :

- • Station extérieure : Artemis Nano, module radio LoRa, minuterie, capteurs
- • Station intérieure : Artemis Thing Plus, module radio LoRa, minuterie, capteurs, affichage
- • Station de base ou station domotique : Raspberry Pi Pico, Arduino Mini, module radio LoRa

Parlons de consommation

Des mesures de température, d'humidité et de pression barométrique sont effectuées périodiquement. La consommation d'énergie est minimisée en plaçant le dispositif de mesure à distance dans un état de faible consommation ou de veille, en le réveillant, en effectuant et en transmettant les relevés, puis en le ramenant dans un état de faible

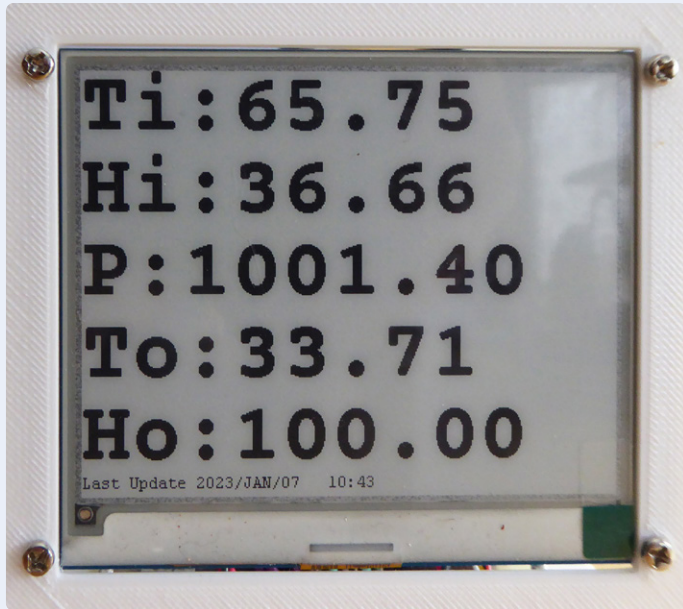


Figure 2. Mon écran ePaper affiche la température (en °F), l'humidité relative et la pression barométrique non corrigée/non normalisée. Les indices « i » et « o » désignent l'intérieur et l'extérieur. Seule la pression extérieure est affichée.

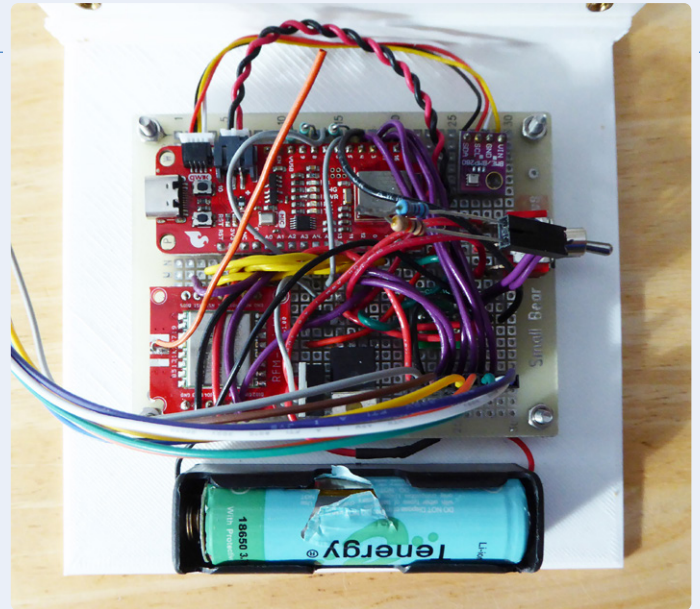


Figure 3. Première itération de la carte d'affichage. Première version du circuit d'affichage, avant la modification de la carte Artemis et l'ajout du TPL5111. Optocoupleurs, MOSFETs, TPL5110, interrupteurs pour la recharge, youpi !

consommation. Le rapport cyclique de la consommation d'énergie est représenté par T_{ON} / T_{TOT} . La consommation d'énergie effective est régie par :

- La consommation de l'appareil lorsqu'il est en marche et qu'il traite des instructions ou des données.
- La fraction du temps pendant laquelle l'appareil est en marche pour collecter et envoyer (ou afficher) des données - la minimisation du cycle d'utilisation affecte de manière significative la consommation globale.
- La consommation de l'appareil entre les mesures. Idéalement, il ne devrait pas y avoir de consommation entre les instants d'échantillonnage, mais ce n'est pas une attente réaliste.

J'ai testé la consommation d'énergie de base de plusieurs cartes microcontrôleurs en les mettant sous tension sans exécuter de programmes ni attacher de périphériques. Voir le **tableau 1** pour les résultats (ceux-ci peuvent ne pas représenter la consommation d'énergie après un réglage fin tel que la désactivation des interfaces périphériques, le ralentissement de l'horloge, etc.) À la lumière de ces résultats, j'ai décidé d'utiliser les cartes Artemis de Sparkfun. L'utilisation de l'énergie a également été un facteur dans la conception de la station

d'affichage intérieure, mais choisir l'écran a été facile : ePaper a la plus faible consommation d'énergie. Cette technologie ne consomme du courant que lors du rafraîchissement de l'affichage (c'est pourquoi le Kindle équipé d'ePaper peut afficher un graphique alors qu'il est inactif et ne consomme pas d'énergie). Le WaveShare dispose d'un bel écran ePaper de 10,7 cm. Les instructions étaient difficiles à suivre et j'ai dû générer quelques polices de caractères, mais comme le montre la **figure 2**, l'écran a bien fonctionné.

J'ai utilisé des capteurs Bosch BME280 pour les mesures. Ces unités mesurent la température, l'humidité et la pression barométrique (T/H/P) et disposent de bibliothèques I²C robustes. Leur consommation de courant est négligeable.

Un mécanisme de minuterie contrôlait le rapport cyclique. La carte microcontrôleur, la radio et le capteur devaient être allumés et éteints - cela n'aurait eu aucun sens d'utiliser une carte microcontrôleur efficace qui consommait 20 µA en veille alors que la radio consommait 20 mA. Plutôt que de coder la mise en veille du microcontrôleur et la mise en veille de la radio, j'allumais et j'éteignais l'ensemble de l'appareil.

Le régulateur 3,3 V de la carte Artemis a une puissance de sortie maximale de 600 mA, suffisante pour alimenter tous les composants de l'affichage intérieur de

même que le capteur extérieur / la radio. Comme je devais couper l'alimentation des cartes, la meilleure solution était une puce de minuterie à très faible consommation. Ces dispositifs possèdent un circuit de minuterie interne, dont l'intervalle est réglé par une simple résistance externe, et s'utilisent avec un microcontrôleur. Lorsque la minuterie se déclenche, le flux de courant est activé. Lorsque le microcontrôleur a terminé sa tâche, il envoie un

Tableau 1. Cartes et consommation

Carte	Consommation électrique (mA)
Teensy 3.5	74,5
Raspberry Pi Pico	20,4
Teensy 4.1	92,6
ESP32 Adafruit Huzzah Feather	125,0
Artemis Thing Plus	8,5
Generic ESP32-WROOM	70,3

Consommation d'énergie de différents microcontrôleurs sous tension, alimentés en 5 V pour V_{in} , mais sans exécution de programme. La consommation d'énergie est régie, non seulement par le microcontrôleur, mais aussi par d'autres composants de la carte (LED, régulateurs de tension, mémoire externe, configuration sans fil, etc.)



message DONE à la minuterie, et le flux de courant est désactivé. J'ai travaillé avec la puce TPL5110 sur une carte maison et sur la carte Adafruit, toutes deux en combinaison avec une première version de la carte d'affichage (figure 3). Mes tribulations avec ce petit truc sont documentées dans [1]. J'ai finalement utilisé un TPL5111 connecté à

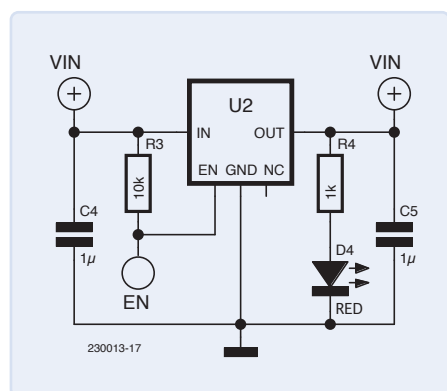


Figure 4. Régulateur de tension Artemis. La broche ENABLE est rappelée au plus par R3 (R1 sur Artemis Nano). Cette résistance doit être retirée. La broche ENABLE est destinée au régulateur de tension de la carte, pas au microcontrôleur.

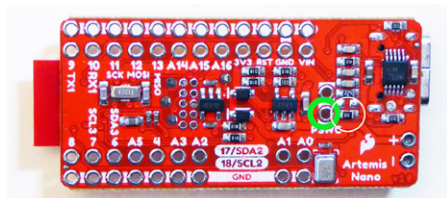


Figure 5. Carte Artemis Nano. R1 est entouré en blanc. Contrairement à la Thing Plus, il n'y a pas de connecteur EN mais un connecteur « PSWC ». La broche ENABLE du TPL5111 doit être reliée au trou entouré en vert une fois que R1 est retiré. Notez que R1 se trouve à l'arrière de la carte et doit être retirée avant le montage.

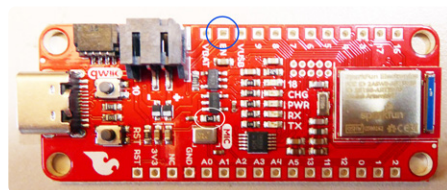


Figure 6. Carte Artemis Thing Plus. R3 est entourée en blanc. La broche ENABLE du TPL5111 peut être reliée directement à la broche EN (entourée en bleu) après que la résistance a été retirée.

la broche d'activation du régulateur 3,3 V sur les cartes Artemis (figure 4). Cette implémentation a nécessité la suppression de la résistance CMS de rappel R1 sur l'Artemis Nano (figure 5), ou R3 sur l'Artemis Thing Plus (figure 6), mais m'a permis de contrôler toute l'alimentation de la carte. Ces résistances rappellent la broche ENABLE du régulateur de tension au plus, sauf si la broche est mise à la masse. Pour finir, les cartes modifiées ont constitué une solution astucieuse, le circuit complet ne consommant que 20 μ A en veille.

Les TPL5110 et TPL5111 sont des puces intéressantes qui méritent d'être étudiées, en particulier dans le domaine des capteurs IdO distants. Les cartes Adafruit sont des implémentations formidables et offrent une excellente flexibilité de conception.

Transmission des données

J'ai choisi les cartes HopeRF LoRa à 915 MHz pour la transmission des données (d'autres pays peuvent exiger une fréquence différente). De nombreuses bibliothèques sont disponibles pour l'environnement

Arduino et simplifient la mise en œuvre. La fréquence porteuse proche du gigahertz permet une meilleure pénétration des murs et des bâtiments et de plus grandes distances de transmission. Les protocoles de transmission sont optimisés pour les petits paquets de données. Ils n'ont pas besoin de signaux d'échange qui prennent du temps ni de surcharge de sécurité. En revanche, la charge de garantir la livraison des données repose sur le programmeur plutôt que sur les protocoles de transmission intégrés.

À ce stade, vous pouvez vous demander comment le capteur et les stations d'affichage communiquent. Après tout, les temps T_{ON} des deux appareils se chevauchent rarement. Si le capteur s'allume et transmet des données, la station d'affichage sera probablement éteinte. La solution a consisté en une troisième station, toujours active, basée sur un Raspberry Pi Pico ne contenant pas de capteurs mais fonctionnant comme un concentrateur et un transmetteur de données, relié à un ordinateur. Je l'appellerai la station domotique. Son seul

Data Flow Diagram

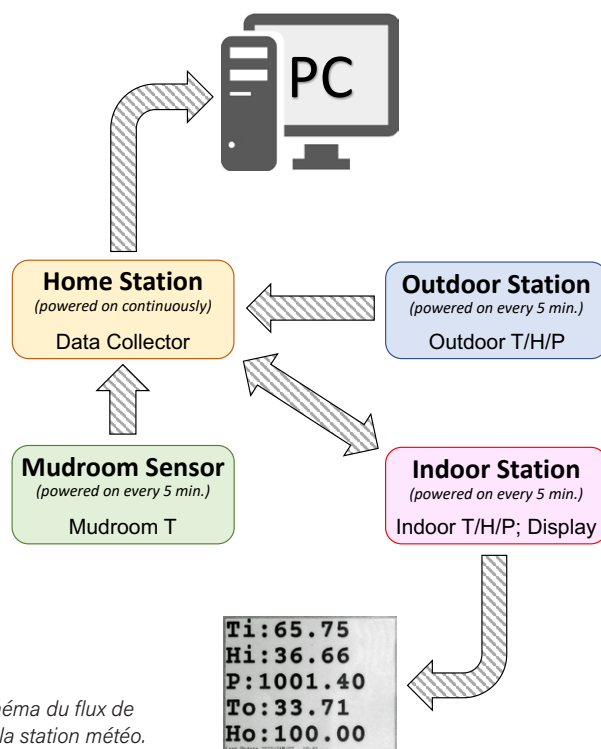


Figure 7. Schéma du flux de données de la station météo.

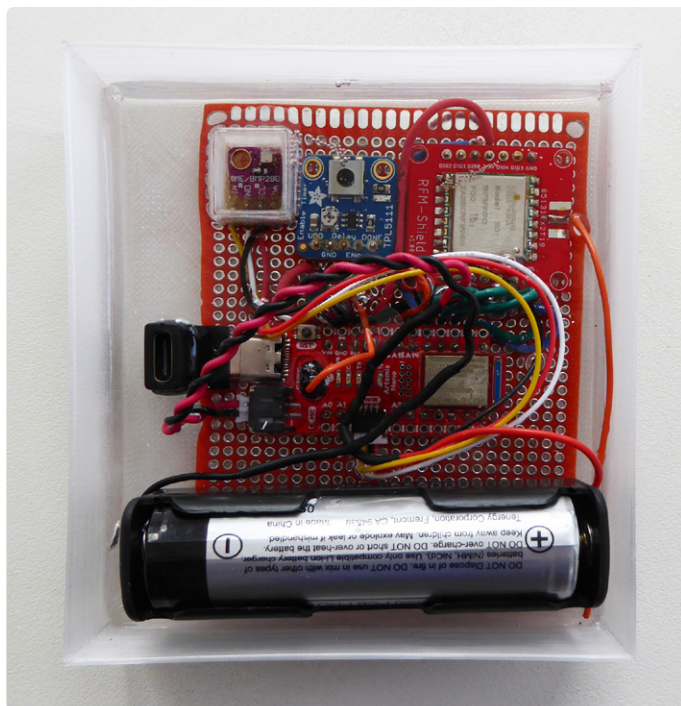


Figure 8. Carte du capteur noyée dans la résine. Notez le petit carré de plastique dans le coin supérieur gauche qui empêche la résine de recouvrir le capteur. Notez également l'adaptateur USB-C à angle droit (à gauche, au milieu) qui permet d'accéder à la carte pour la recharge et les mises à jour logicielles.



Figure 9. Détail du boîtier du capteur. Tous les orifices d'aération sont grillagés pour protéger des attaques biologiques plus que des cyberattaques ;-).

périphérique est la radio et quelques LED pour le débogage. Son régulateur de tension peut supporter un courant de quelques centaines de milliampères, et j'ai alimenté la radio à partir de ce régulateur. Le Pico est alimenté par le port USB d'un PC et communique en série grâce à un programme en Processing. Quelques centaines de lignes de code suffisent pour tracer la pression barométrique et afficher un tableau des relevés sur une page web. La **figure 7** illustre le flux de données comme suit :

1. Le capteur extérieur se réveille toutes les cinq minutes et envoie les relevés de température, d'humidité et de pression (T/H/P) à la station domotique. Il n'y a pas de retour entre la station domotique et le capteur extérieur. Le capteur extérieur s'éteint après la transmission sans confirmation de réception par la station domotique.
2. La station domotique stocke les relevés les plus récents du capteur extérieur en mémoire vive. Les relevés extérieurs sont également envoyés à l'ordinateur pour traitement ultérieur.
3. La station intérieure d'affichage avec son capteur se réveille toutes les 5 minutes. Elle mesure les paramètres intérieurs T/H/P, envoie ces informations à la station domotique et conserve une copie des mesures en mémoire vive.
4. Lorsqu'elle reçoit une demande de mise à jour de la station d'affichage, la station

domotique transfère cette demande à l'ordinateur. Elle envoie alors les observations extérieures les plus récentes à la station intérieure, qui met alors à jour l'affichage et s'éteint.

Au fil du temps, j'ai constaté des blocages répétés de la station domotique avec pour conséquence l'impossibilité de mettre à jour l'affichage et le programme en Processing. Les messages postés sur plusieurs forums d'assistance suggèrent que la puce de l'émetteur-récepteur LoRa SX1276 peut se figer et provoquer ce comportement, bloquant à la fois la radio et le microcontrôleur lorsqu'elle est laissée en marche pendant de longues durées. Ce comportement n'a jamais été constaté avec la station d'affichage ni celle de mesure, qui ont des cycles de mise sous et hors tension fréquents.

La solution pour la Pico a été d'utiliser un second microcontrôleur, un Arduino Mini, comme minuterie intelligente. La sortie du Mini était connectée à la broche d'activation du Pico, ce qui permettait d'éteindre le régulateur de tension pendant 500 ms toutes les heures. Une sortie de la Pico était connectée à la Mini, signalant le moment où le flux de données serait le moins vulnérable aux effets d'une réinitialisation : la réinitialisation du matériel ne se produirait qu'après une mise à jour des données et alors que le Pico ne recevrait pas de nouvelles informations.

Bien que cela puisse sembler être une utilisation extravagante d'un microcontrôleur, les appareils sont si bon marché que cela se justifie. J'encourage la communauté Elektor à me faire part de ses commentaires sur la façon de mieux résoudre ce problème. Deux versions du logiciel de la station domotique sont fournies [2], l'une destinée à être utilisée avec le programme en Processing, l'autre indépendante d'une connexion informatique et ne nécessitant qu'une alimentation électrique telle qu'un vieux chargeur de téléphone avec le connecteur USB approprié. Le code en Processing est encore en chantier et ne prétend pas être une solution complète ni exempte de bogues.

Climatisation

Au début, mon capteur extérieur tombait en panne par intermittence en cas de changements rapides de température et d'humidité, et je soupçonnais la condensation. Après avoir examiné toutes mes soudures et mon câblage et n'avoir trouvé aucun défaut, j'ai décidé d'enrober de résine ma carte extérieure. J'ai imprimé en 3D un boîtier pour mon circuit imprimé et une petite protection à placer autour de la carte du capteur pour qu'elle ne soit pas recouverte par la résine (**figure 8**). Je me suis servi d'un adaptateur USB-C à 90° pour conserver l'accès au connecteur USB-C de la carte Artemis.

J'ai utilisé de la résine époxy marine



Liste des composants

Résistances

R1 = voir texte

R2 à R4 = 1 M Ω ; 1/4 W

R5 = 680 Ω ; 1/4 W

R6 à R8 = 360 Ω ; 1/4 W

Semi-conducteurs

D1 = LED bleue

D2 = LED rouge

D3 = LED jaune

D4 = LED verte

Cartes

2 \times BoB TPL5111 (Adafruit Industries)

1 \times Raspberry Pi Pico

1 \times Artemis Thing Plus (Sparkfun)

1 \times Artemis Nano (Sparkfun)

2 \times cartes de connexion BME280 (ASIN B08DHTGNHR *)

1 \times BoB FTDI basique 3.3 V (Sparkfun)

3 \times Émetteurs-récepteurs LoRa HopeRF RFM95CW 915 MHz certifiés FCC (Anarduino.com)

1 \times Arduino Pro Mini, 3,3 V 8 MHz (voir texte)

Autres

2 \times Connecteurs ST à 4 broches (ASIN B01DUC1M2O* ; voir Conseils de réalisation)

1 \times Écran ePaper 10,7 cm (WaveShare ; ASIN B074NR1SW2*)

3 \times Plaques perforées FR4 double face 7 X 9 cm (ASIN B08F7X8JHV*)

2 \times Batteries rechargeables LiPo18560

2 \times Supports pour les batteries 18650

Kit de résine époxy (ASIN B07TVWTG829*)

Adaptateur USB-C à 90° (ASIN B0BBVWF54L*)

Barrettes et fils selon les besoins

* Les codes ASIN font référence à des numéros d'inventaire Amazon consultables.

transparente bon marché, disponible sur Amazon. L'époxy marine est formulé pour durcir très lentement et donne à l'utilisateur plus de temps pour travailler avec le matériau. Comme je ne recherchais pas un résultat esthétique parfait et que je ne voulais pas que mes idiots de chats soient couverts d'époxy, j'ai placé le récipient sur un chauffe-plats électrique à environ 65° C. En moins d'une heure, l'époxy était à l'épreuve des pattes. J'ai ensuite laissé le durcissement se terminer pendant la nuit à température ambiante (18-20° C). Le circuit imprimé est logé dans un bocal en plastique ventilé peint en blanc (**figure 9**) et les trous de ventilation sont grillagés pour protéger le capteur contre les insectes.

Avant de résiner le circuit, j'ai confirmé, par des tests, que l'époxy ne court-circuiterait pas les fils nus et ne perturberait pas les connexions électriques. Depuis que j'ai résiné le circuit, il n'y a plus eu de pannes intermittentes.

Logiciels et données

Le logiciel pour les trois stations utilise les bibliothèques LoRa et le code WaveShare. Pour les stations intérieure et extérieure, il faut des bibliothèques pour le capteur BME280. Vous pouvez télécharger gratuitement les paquets complets à l'adresse [2]. La charge de codage pour le TPL5111 est triviale. J'ai échangé la simplicité du code et le raccourcissement du T_{ON} contre la garantie de transmission des données. Après la mesure, une courte chaîne est créée qui contient les observations et l'identification de la station d'envoi ; cette chaîne est envoyée deux fois (voir ci-dessous). Les transmissions de la station domotique à la station d'affichage comprennent également des données temporelles afin que l'affichage puisse indiquer l'heure de la dernière mise à jour.

On dirait que le code n'est pas complet. En réalité, l'exécution du code s'arrête lorsque le régulateur de tension est coupé.

Pour cela, l'unité centrale envoie un signal *DONE* à la minuterie, ce qui permet de combler les lacunes du logiciel. À l'exception du code WaveShare, je n'ai pas inclus de bibliothèques dans le téléchargement du logiciel. Celles-ci sont facilement ajoutées à votre environnement par l'EDI Arduino. J'ai utilisé la bibliothèque *Sparkfun BME280* parce qu'il n'y a aucune dépendance de code autre que *Wire.h*. La bibliothèque *Sandeep Mistry LoRa* a fourni toutes les fonctionnalités nécessaires et n'avait aucune dépendance de code autre que *SPI.h*.

Comme la station domotique ne peut pas coordonner les transmissions entre les capteurs et la station d'affichage, les collisions de données et les pertes de données qui en résultent sont inévitables. La chaîne de données est envoyée deux fois avec un délai entre les transmissions pour réduire le risque de données non délivrées. Au global, la perte de quelques points de données par jour est acceptable pour une application non critique.

Au cours du développement, j'ai réalisé que la station domestique pourrait communiquer avec un ordinateur et envoyer les observations météorologiques à une base de données, une page web ou autre. Il s'agit d'un travail en cours. J'ai écrit un serveur de pages web rudimentaire en Processing, un langage dérivé de Java avec une base d'utilisateurs étendue et de nombreuses bibliothèques de qualité. Comme indiqué, le code Processing est fourni, ainsi que les croquis Arduino. Bien qu'il puisse y avoir des similitudes superficielles, il ne s'agit pas d'un projet LoRaWan, et la station d'accueil n'est pas une passerelle LoRa.

Ma configuration surveille les conditions extérieures, les conditions dans ma cuisine (emplacement de l'affichage) et un troisième capteur dans notre vestibule, qui est exposée au risque de gel en hiver. Les données du capteur du vestibule n'apparaissent que sur la page web. Le nombre de capteurs et d'affichages qui peuvent être pris en charge avec cette configuration est limité. Avec l'augmentation du trafic sur le réseau, la perte de données due aux collisions finira par devenir inacceptable. De nombreux facteurs influent sur ce phénomène, notamment la précision des résistances de synchronisation du TPL5111, les facteurs environnementaux et, surtout, la fréquence d'échantillonnage et/ou de

Conseils de réalisation

- 1. Retirer R1 et R3** : Voir les images pour l'identification des résistances R1 et R3 qu'il faut retirer du Nano et de la Thing respectivement. Si vous ne le faites pas, les circuits ne fonctionneront pas.
- 2. Connexion ENABLE sur Artemis Nano** : voir figure 5 pour la connexion ENABLE sur Nano. J'ai pris une photo du bas de la carte. Vous ne devez PAS utiliser la broche à la masse !
- 3. Circuit imprimé principal** : n'importe quel circuit imprimé peut être utilisé, mais les articles référencés étaient pré-étamés, en FR4, de taille adéquate et résistants aux retouches.
- 4. Fréquence radio** : achetez une carte radio légale pour votre pays.
- 5. Connexion au système Qwiic** : le connecteur JST mentionné s'adapte parfaitement au système QWIIC, mais les couleurs NE correspondent PAS à la convention Sparkfun. Pour ces connecteurs, blanc = GND, jaune = 3.3 V, noir = SDA, et rouge = SCL.
- 6. Cartes BME280** : soyez prudent lors de l'achat de cartes BME. Certains vendeurs d'Amazon remplacent (sciemment ou non) les puces BMP280 par des puces BME280. La version BMP est moins chère et semble presque identique, mais les bibliothèques pour BME ne fonctionneront pas et vous n'aurez pas de données d'humidité.
- 7. Notes sur la connexion WaveShare** : l'unité WaveShare est livrée avec un câble facile à utiliser. Violet = BUSY, Blanc = RESET, Vert = DC, Orange = CS, Jaune = CLK, Bleu = DIN (MOSI), Marron = GND, Gris = Vcc. L'unité n'a pas de connexion MISO.
- 8. Intervalle de temps du TPL5111** : l'intervalle de temps du TPL5111 est programmé à l'aide d'un potentiomètre intégré ou d'une résistance externe. Le site web d'Adafruit et la fiche technique du circuit intégré fournissent un tableau des valeurs de résistance pour différents intervalles de temps. Si vous utilisez une résistance externe, vous devez couper une trace au dos de la carte. Si la consommation est vraiment critique, la LED d'activité peut également être retirée du circuit en coupant une trace.
- 9. Connexion de la batterie aux cartes Artemis** : la batterie se connecte à la carte à l'aide d'un câble JST à 2 fils disponible partout.
- 10. Carte d'interface radio** : j'ai utilisé un BoB pour la radio de Diycon.nl (LoRa Node PCB 100 Shield Only pour RFM92/RFM95). Il est facile d'y fixer une antenne filaire ou un connecteur d'antenne.
- 11. Antenne radio** : j'ai utilisé une simple antenne filaire ¼ d'onde. La longueur de fil correcte pour 915 MHz est de 78 mm, soudée au connecteur central de la carte radio.
- 12. Résine** : très salissante. Gants, récipients jetables pour mesurer et mélanger, bâtons de mélange jetables et chiffon. Assurez-vous que la carte fonctionne parfaitement avant de procéder. L'adaptateur USB-C à 90° doit être installé, et la minuterie doit avoir la bonne résistance. Le récipient peu profond doit être légèrement plus grand que la carte et la batterie. Ne pas immerger le BME280 dans l'époxy. J'ai fait une protection autour du capteur (la résine remontera par les trous, protégez donc aussi le dessous). Vous pouvez également connecter le capteur par des fils au connecteur QWIIC et maintenir la carte du capteur au-dessus de la résine pendant qu'elle durcit. Chauffer doucement la résine accélère considérablement le temps de durcissement. Les trois points clés sont que le capteur ne soit pas recouvert de résine, que le connecteur USB-C reste accessible à travers l'adaptateur et qu'à l'exception de la carte du capteur, tous les composants du circuit imprimé soient recouverts.
- 13. Le boîtier** : il y a plusieurs problèmes de conception. Les trous de ventilation doivent être situés et protégés afin que l'eau ne puisse s'infiltrer à l'intérieur du boîtier qu'en contrant la gravité. Deuxièmement, le boîtier doit être réfléchissant ou au moins peint en blanc pour minimiser l'effet de serre. Troisièmement, le boîtier doit être léger. L'inertie thermique d'un boîtier lourd et volumineux rendra vos mesures lentes et imprécises. Enfin, le boîtier doit être protégé d'une manière ou d'une autre contre les insectes.
- 14. Batteries 18560** : les batteries vantant des densités d'énergie impossibles à des prix ridiculement bas sont omniprésentes sur Amazon et eBay. Soyez sceptique face à ces affirmations. Les piles de marque avec de 2500 à 3500 mAh sont un choix sûr !

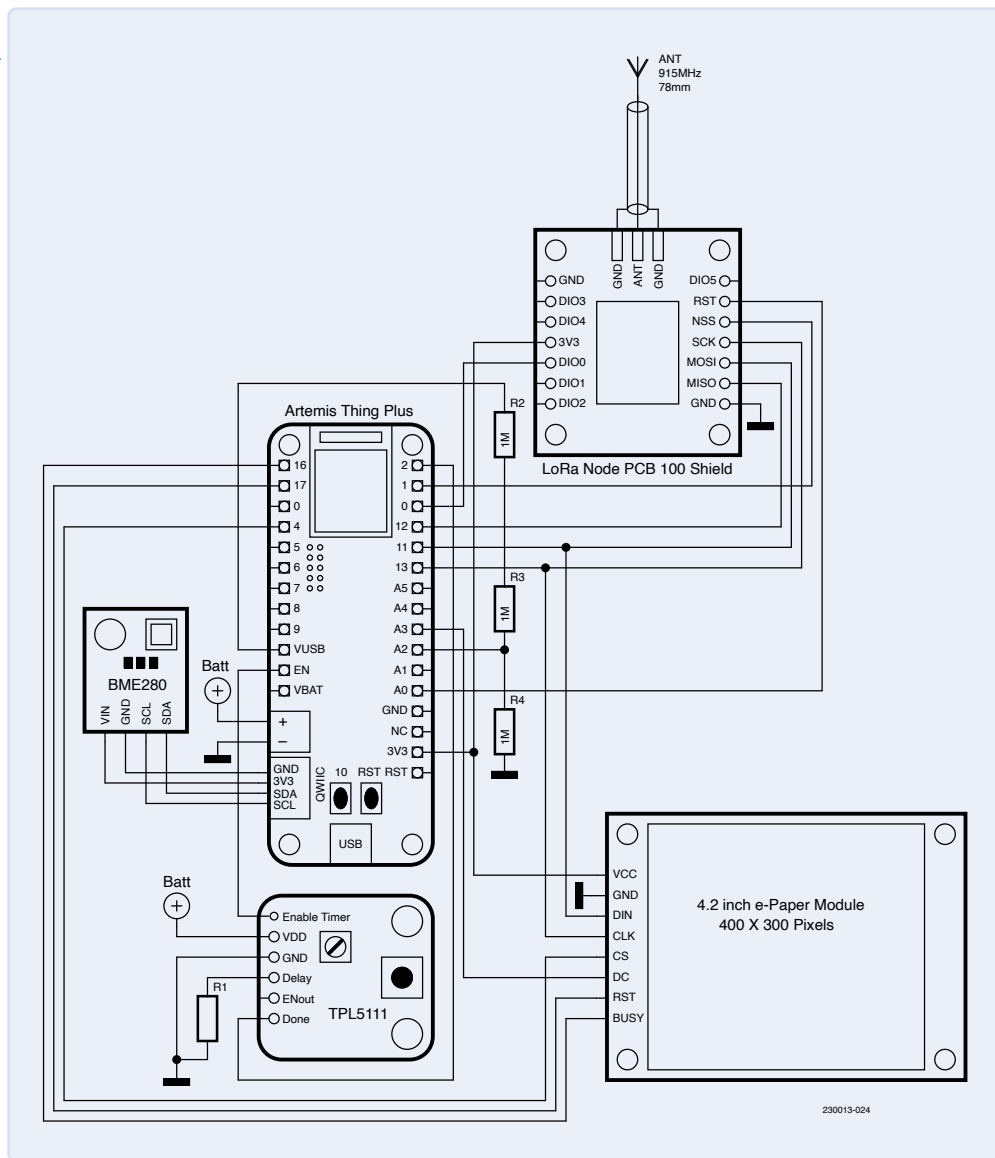


Figure 10. Circuit de la station d'affichage.

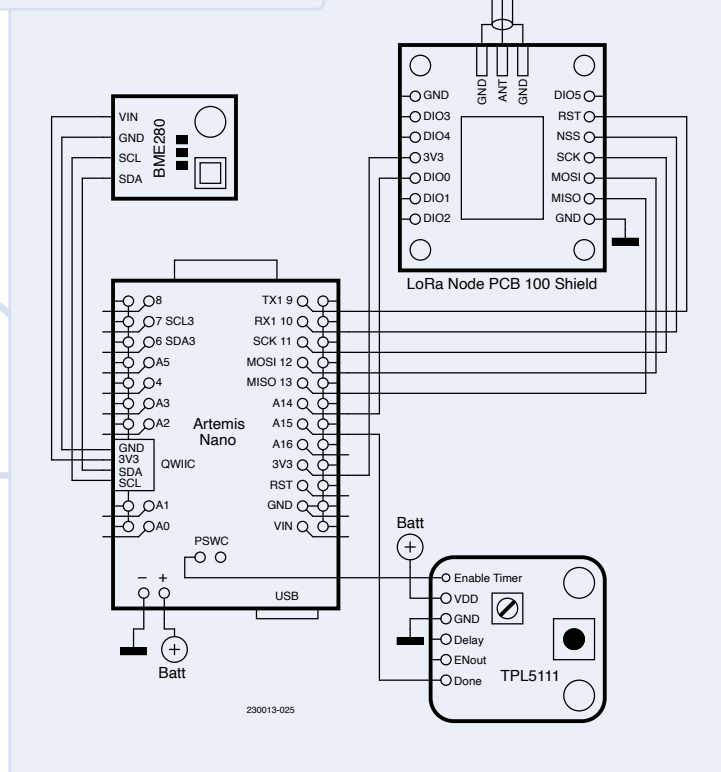


Figure 11. Circuit du capteur distant.



demande de mise à jour. Il n'y a pas de limite au nombre d'affichages n'effectuant pas de demandes de données. Les unités effectuant des demandes de données contribuent au trafic radio. C'est le trafic radio total, et non les transmissions de mesures des capteurs, qui limite le réseau. Le calcul des limites théoriques des différentes configurations est laissé à l'appréciation du lecteur.

Considérations pratiques

Le matériel est câblé à la main. Lisez attentivement l'encadré **Conseils de réalisation**. Les capteurs communiquent via I²C. J'ai utilisé le système Qwiic de Sparkfun, qui est très pratique. Il est facile de câbler un connecteur compatible Qwiic aux petites cartes BME280 disponibles sur Amazon et eBay. Veillez à acheter une BME plutôt qu'une BMP280 si vous souhaitez obtenir des informations sur l'humidité. Le module HoperF a un pas de 2 mm au lieu de 2,54 mm, ce qui nécessite un BoB ou une soudure soignée. J'utilise une simple antenne filaire ¼ d'onde.

J'ai utilisé FontEdit pour créer une police de 36 points pour l'affichage ePaper. Les instructions de WaveShare sont difficiles à comprendre, mais j'ai fini par développer un code fonctionnel. Si vous utilisez un MCU autre que l'Artemis, n'oubliez pas que les tables de polices et le tampon bitmap consomment beaucoup de mémoire vive. J'ai utilisé une batterie rechargeable au lithium 18560 pour l'alimentation. Voir les **Conseils de réalisation** pour des observations sur l'approvisionnement en 18560. D'autres configurations d'alimentation sont possibles, mais les piles alcalines ne sont absolument pas recommandées pour les températures inférieures à -18° C.

Les schémas (voir **figures 10 à 12**) sont indicatifs. Les nouvelles cartes 32 bits sont flexibles, avec de nombreuses broches supportant les interruptions et avec des interfaces SPI, I²C et UART alternatives.

Figure 12. Circuit de la station domotique.

De ce fait, certaines connexions ont été basées sur l'optimisation de la disposition physique de l'ensemble du circuit. Bonne lecture !

VF : Denis Lafourcade — 230013-04

À propos de l'auteur

Ed Ringel est un médecin semi-retraité spécialiste des soins respiratoires et intensifs. Il profite de la nature du Maine avec sa femme, écrit de la science-fiction, fabrique des objets avec son imprimante 3D et développe des projets électroniques avec des microcontrôleurs.

Des questions, des commentaires ?

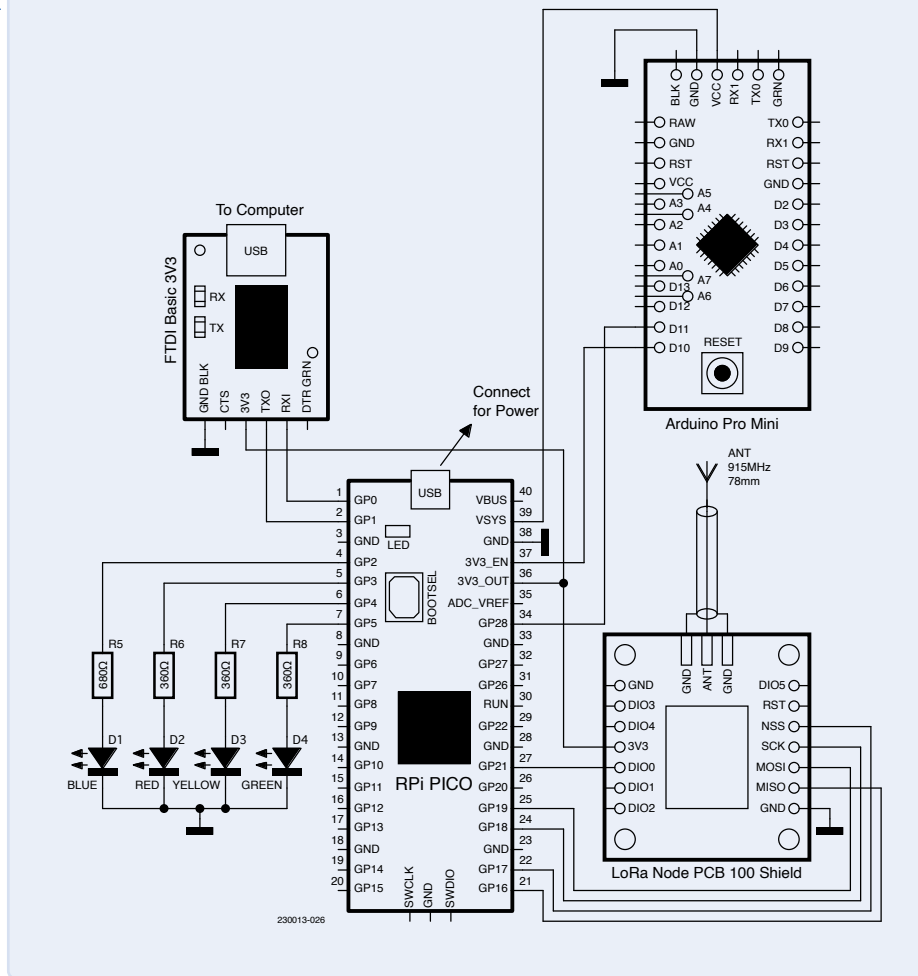
Envoyez un courriel à Elektor à l'adresse redaction@elektor.fr.



Produits

➤ **Kit capteur SparkFun**
<https://elektor.fr/19620>

➤ **Raspberry Pi Pico RP2040**
<https://elektor.fr/19562>



LIENS

[1] Tribulations : <https://forums.adafruit.com/viewtopic.php?p=927526>

[2] Ce projet sur Elektor Labs : <https://elektormagazine.fr/labs/low-power-lora-weather-station>