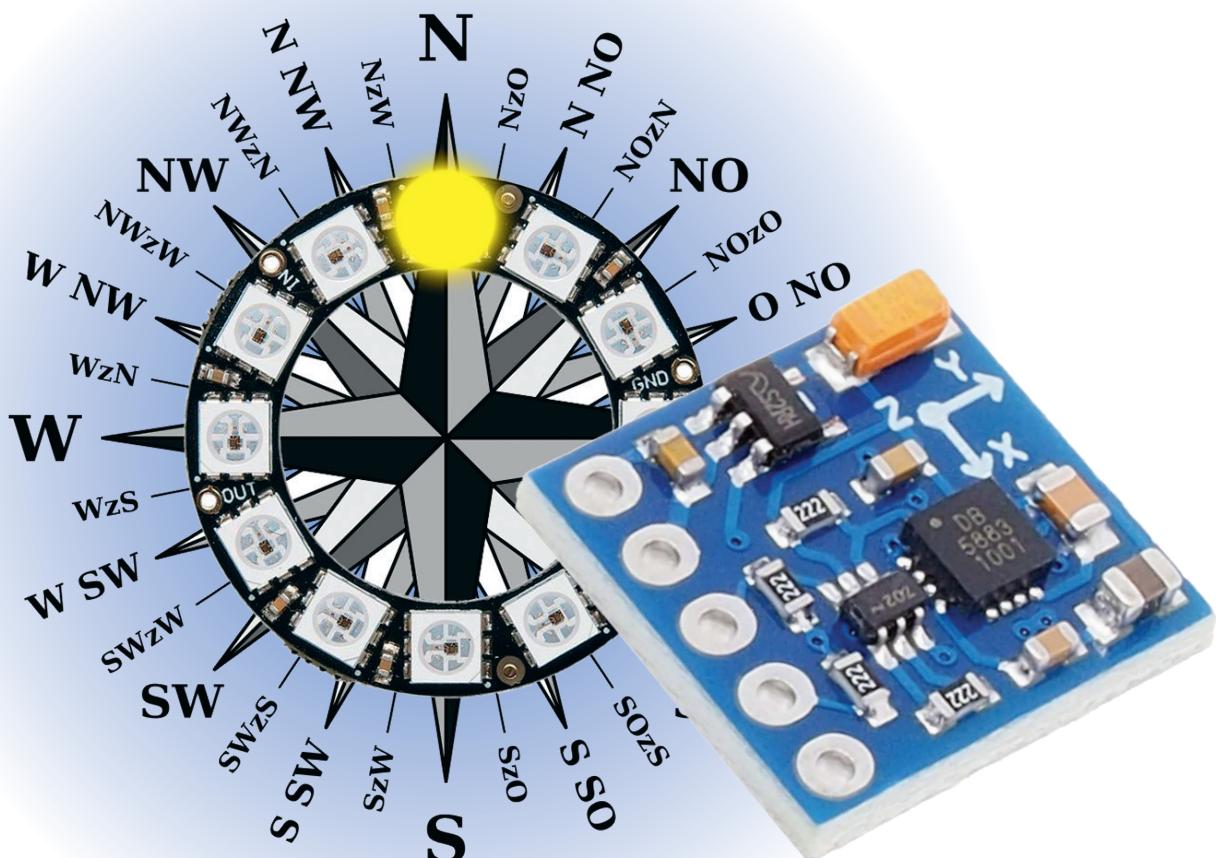


rose des vents

avec le capteur GY-271

Ou pourquoi il faut faire des huit pour calibrer un capteur



Cette magnifique rose des vents est une œuvre de Soeren Gasch, mise dans le domaine public : https://commons.wikimedia.org/wiki/File:Compass_rose_german.svg.

Rolf Hase (Allemagne)

Quoi de plus simple que de réaliser une boussole électronique à partir d'un capteur de champ magnétique sur une carte d'expérimentation, d'un anneau de LED NeoPixel comme rose des vents, et d'un Arduino qui gère le tout avec un logiciel tout fait ? Mais quand ça ne marche pas comme prévu, il ne reste que le « retour aux fondamentaux » pour mettre le doigt sur un défaut surprenant et trouver une solution qui ne l'est pas moins !

Autrefois, lorsqu'on partait faire une balade à vélo en territoire inconnu, il fallait emporter une carte. Aujourd'hui, un ordiphone dans son support sur le guidon suffit, ou, si l'on

veut une image plus grande, une tablette dans le panier avant. Sur mon ordiphone, j'ai installé l'application OsmAnd, qui utilise les cartes d'OpenStreetMap. Mon appareil n'est

malheureusement pas équipé d'une boussole et je dois me contenter de la minuscule flèche de direction du parcours affichée sur l'écran d'OsmAnd qui disparaît lorsque

je m'arrête, me privant de toute possibilité d'orientation. Sur l'écran, la carte se met alors à tourner dans tous les sens, et je dois me déplacer d'une vingtaine de mètres (dans la mauvaise direction, évidemment) pour pouvoir me réorienter correctement. C'est ainsi que j'ai eu l'idée d'emporter avec moi une boussole distincte, bien visible et dépourvue d'une aiguille se mettant à danser à la moindre secousse. C'est en feuilletant d'anciens numéros d'*Elektor* qu'il me vint l'idée d'une solution électronique qui, au moins une génération d'électroniciens plus tard, devrait encore gagner en fiabilité et en précision.

Une solution utilisant un Arduino Nano, un anneau de LED NeoPixel et le module GY-271 est présentée sous [1], avec le schéma du circuit et son code source. Elle me semblait tout à fait satisfaisante, au point que j'ai fait l'acquisition du module GY-271. Comme j'aime avoir plusieurs exemplaires du même composant sous la main, j'en ai commandé trois exemplaires à peu de frais sur l'internet.

Module boussole GY-271

Le GY-271 est un module d'extension conçu pour l'Arduino, et sa rangée de broches est parfaitement compatible avec une carte d'expérimentation ordinaire. Le circuit intégré principal du module est une boussole à trois axes portant la référence HMC5883L ou QMC5883 : ce circuit est souvent présent dans les ordiphones et constitue donc un produit de masse bon marché. La **figure 1** montre le circuit interne de la puce. Les autres composants sur le module sont un régulateur de tension à faible chute de 3,3 V, quelques condensateurs de filtrage et de dérivation et deux adaptateurs de niveau pour l'interface I²C. On peut en déduire que le module peut être alimenté par une source de 3 à 5 V et qu'il communique par un bus I²C avec un microcontrôleur (sur une carte Arduino, Raspberry Pi ou n'importe quelle autre carte « intelligente »). Le protocole de communication est décrit de manière très détaillée dans la documentation d'accompagnement et dans un e-book d'applications gratuit disponible en plusieurs langues.

Comme il n'y a pas de signaux analogiques à traiter, le câblage du circuit se réduit à une connexion, sans se soucier de son alimentation. L'alimentation de l'anneau de LED passe par deux pastilles soudables, le plus et la masse, et un seul signal logique d'horloge suffit aux NeoPixels. Pour gagner de la place, un Arduino Pro Mini a été utilisé,

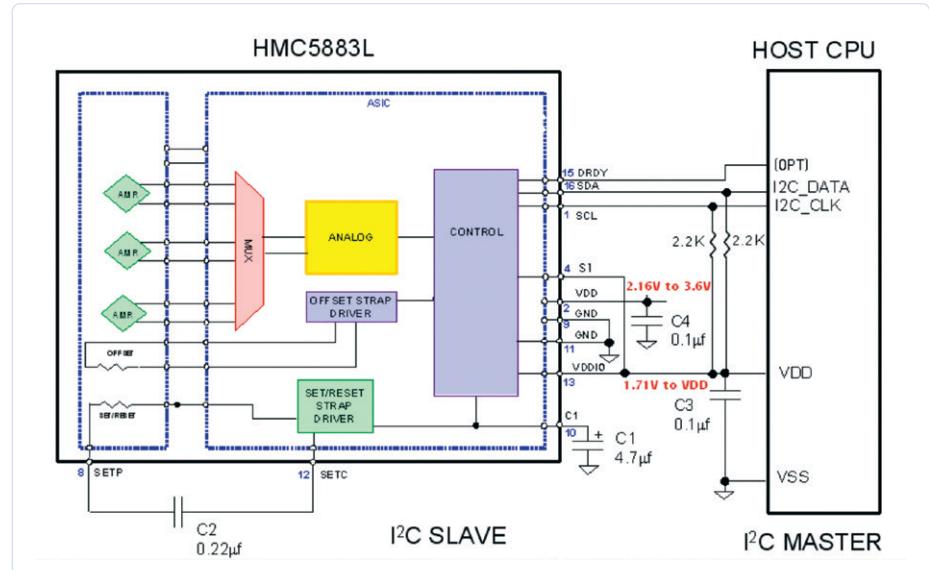


Figure 1 : Circuit interne du capteur boussole HMC5883L (source : fiche technique Honeywell).

et l'EDI Arduino doit être configuré en conséquence. Un adaptateur FTDI-USB-série est nécessaire pour programmer l'Arduino.

Jusqu'ici, tout ne va pas très bien

En théorie, donc, la construction de la boussole et son installation sur mon vélo ne devaient pas présenter de difficulté. Mais dès le premier essai, patatras ! Seul un quadrant a fonctionné, puis l'« aiguille » a fait un saut

de 180°. L'examen du code source (retiré depuis) n'a révélé aucune erreur, alors j'ai mis le problème sur le compte de la médiocre qualité des produits sans marque achetés sur eBay.

En bref, en tant que praticien, il ne me restait plus que la méthode empirique pour découvrir la cause du problème. Les premières expériences ont montré que l'appareil était très sensible aux champs externes, par exemple ceux dans le voisinage

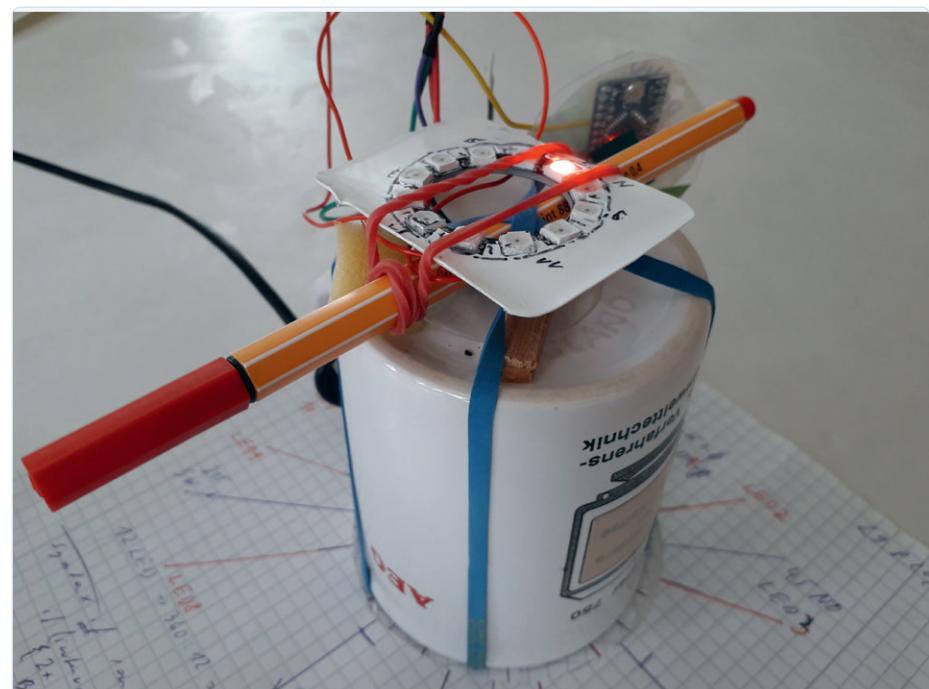
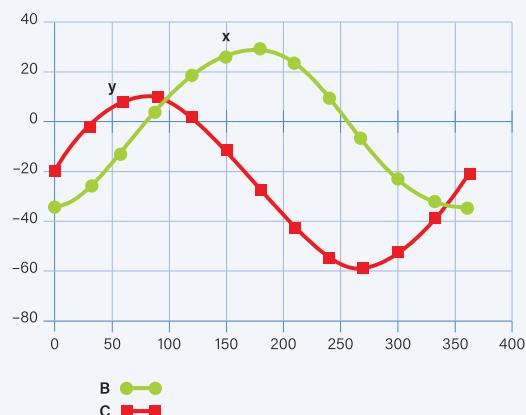


Figure 2 : Un montage expérimental astucieux pour traquer le bogue.



Angle	Valeur x	Valeur y
0	-34	-20
30	-26	-2
60	-14	7
90	3	10
120	19	2
150	26	-11
180	28	-28
210	22	-43
240	9	-55
270	-7	-58
300	-23	-51
330	-32	-38
360	-34	-22

Figure 3 : Les relevés montrent un décalage du zéro, cause du comportement incorrect.



Figure 4 : Voici l'aspect idéal des deux courbes.

de mon ordinateur portable. Comme il n'est pas recommandé d'expérimenter avec un circuit d'essai qui n'est qu'un foisonnement de fils volants, j'ai construit un dispositif expérimental plus robuste, comme le montre la **figure 2**. Voici ce qu'on peut y voir : une tasse à café renversée, garantie magnétiquement saine, supporte la carte d'essai, le capteur (non visible sur cette photo) et l'anneau de LED, tous fixés à son fond à l'aide d'élastiques, en compagnie d'un stylo servant d'indicateur du « nord ». Un petit morceau de bois et une éponge servent à la fois de cale et de coussinet. Les directions des axes x et y sont indiquées sur le côté composants du module GY-271. La pointe rouge du stylo correspond à la direction de la LED 0 et à la direction x négative (!) du capteur, qui est monté avec le côté composants vers le bas. L'appareil n'utilise pas l'axe z du capteur, mais il faut le régler pour que les valeurs correspondantes soient nulles afin de minimiser les perturbations du système. (Ces valeurs, après un petit traitement mathématique, pourraient être utilisées pour mesurer des pentes

montagneuses « hors catégorie »). La tasse repose sur une feuille de papier avec le dessin d'une rose des vents rudimentaire. Un cercle central a été découpé pour pouvoir la faire tourner facilement. Et, mine de rien, ce dispositif a non seulement fourni des résultats reproductibles, mais a également produit un effet « Eurêka ! » remarquable ! J'ai donc tourné la boussole-tasse à café et saisi les valeurs des axes x et y dans la feuille de calcul Excel illustrée à la **figure 3** (valeurs de l'axe x en bleu, axe y en rouge). Si l'on compare ces valeurs avec les valeurs idéales indiquées à la **figure 4**, on constate immédiatement un écart. Les tracés sont ceux d'une courbe sinus et d'une courbe cosinus (ou d'une courbe sinus décalée de 90°). Vieux souvenirs d'école : dans un triangle rectangle, le sinus d'un angle est égal au côté opposé sur l'hypoténuse, le cosinus est égal au côté adjacent sur l'hypoténuse, et la tangente est égale au côté opposé sur le côté adjacent, ou au sinus divisé par le cosinus, l'hypoténuse se simplifiant. Pour déterminer l'angle à partir des côtés, il nous faut la fonction inverse, l'arc tangente, ou arctan.

Les bibliothèques Arduino comprennent une fonction arctan2, qui prend les valeurs x et y comme arguments. Une normalisation à 1 n'est pas nécessaire, mais les amplitudes doivent être égales. Il devient évident que le décalage du zéro est la cause du problème. Cela explique également cette danse bizarre en forme de huit à laquelle on soumet certains ordiphones pour calibrer leur boussole : il est clair que ces ordiphones utilisent des capteurs magnétiques HMC ! J'ai entendu dire que, dans certains cas, cette chorégraphie (ou devrais-je dire « octégraphie » ?) ne fonctionne pas, apparemment à cause d'un décalage du zéro trop grand pour être corrigé. Une procédure analogue permettrait de réaliser un calibrage « automatique » sur l'Arduino, en déduisant les corrections des valeurs maximales et minimales du capteur. Mais l'écriture des valeurs de calibrage vers le capteur nécessiterait une communication bidirectionnelle, donc deux broches d'interface au lieu d'une. Ayant enregistré la totalité de la courbe caractéristique du capteur, nous pouvons calculer le décalage et le reporter dans le programme plutôt que de compliquer le câblage.

Nouveau programme

Un avantage significatif du logiciel original est qu'il inclut tout le code nécessaire pour initialiser et lire la puce HMC. J'ai apporté quelques modifications au code du moniteur, rendu les valeurs de lecture plus faciles à gérer en les divisant par 100, car, comme l'anneau de LED NeoPixel n'a que 12 positions, une précision d'environ ±10 % suffit. Si vous envisagez d'utiliser un anneau NeoPixel plus grand ou un écran graphique, vous devrez adapter ces valeurs brutes à la résolution dont vous avez besoin. Pour piloter les NeoPixels, j'ai dégraissé le code très clair disponible sous [3] et obtenu une

Listage1. Correction du décalage codée en dur.

```
Serial.print("X Value: ");
x = x/100;
x = x+2;
Serial.print(x);
Serial.print(" Y Value: ");
y = y/100;
y = y+20;
Serial.print(y);
Serial.print(" Z Value: ");
z = z/100;
Serial.print(z);
```

version qui s'est avérée convenir pour piloter douze LED. Le **listage 1** montre le fragment de code servant à corriger le décalage du capteur. Le programme Arduino complet, avec les modifications, peut être téléchargé sous [4]. Dans ce croquis, la fonction moniteur est particulièrement intéressante et invite aux expérimentations personnelles.

Essais sur le terrain

La boussole doit être installée dans un boîtier en plastique transparent et montée sur le guidon de la bicyclette. La disposition intérieure n'est pas critique : les différentes couches peuvent être séparées par des feuilles de plastique, toutes fixées à une plaque de base à l'aide de ruban adhésif double face. L'anneau des LED NeoPixel est fixé sur une autre plaque, avec une découpe circulaire intérieure et des dentelures. Les mesures ont été faites avec la LED 0 pointant vers le nord, mais lors des premiers tests avec le logiciel, c'est la direction du sud que l'anneau a indiquée. Pour résoudre ce problème, plutôt que de modifier le logiciel, il était bien plus simple de tourner l'appareil d'un demi-tour pour faire pointer la LED 6 vers l'avant. Maintenant, lorsque la bicyclette est orientée vers le nord, la LED « droit devant » s'allume ; tournez à droite, et la LED suivante à gauche s'allume, indiquant à nouveau la direction du nord, comme il se doit. Malheureusement, il s'est avéré que l'acier du guidon, avec son magnétisme résiduel, perturbait les lectures. Cependant, j'ai découvert à cette occasion que le cadre de mon vélo est fait en aluminium dont, contrairement à l'acier, les propriétés

LA BOUSSOLE SUR UNE BICYCLETTE ÉLECTRIQUE

J'ai réalisé l'alimentation du circuit de la manière suivante. Cachée sous le guidon, j'ai monté une prise ordinaire du type de celles que l'on trouve dans les voitures, reliée par deux mini-fiches (rouge et bleue) à la prise de charge de la batterie de la bicyclette électrique. Il est essentiel de bien isoler ces connexions, car malgré sa faible tension (généralement 36 V), la batterie contient assez d'énergie pour faire pas mal de dégâts. En cas de court-circuit, un câble inadapté brûlera comme un fusible ! L'alimentation de 5 V du circuit est produite par un convertisseur abaisseur USB avec une entrée de 24 V, du monde des camions. Comme les batteries des camions sont souvent chargées à une tension nettement supérieure à leur tension nominale, ces appareils utilisent généralement un convertisseur à découpage MC34063A, qui peut supporter une tension d'entrée allant jusqu'à 40 V. Le convertisseur USB a été démonté pour souder directement sur la sortie 5 V un câble bifilaire avec des fiches, la consommation électrique de la boussole étant assez faible.

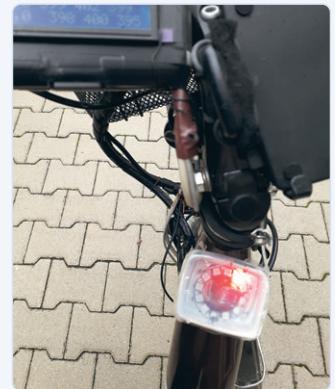


Figure 5 : La boussole assemblée et fixée au cadre de mon vélo.

paramagnétiques ne perturbent pas la boussole. Comme l'appareil est petit, il a été possible de le fixer sur le cadre tout en le gardant dans mon champ de vision (**fig. 5**). Les tests sur le terrain ont montré que l'appareil fonctionnait de manière fiable, le seul problème étant survenu près d'un pont de chemin de fer, probablement à cause du champ dû aux caténaires. Désormais, grâce à cette boussole, je peux m'orienter beaucoup plus facilement en territoire inconnu !

(200597-04)

Des questions, des commentaires ?
Contactez Elektor (redaction@elektor.fr).

Contributeurs

Idée et texte : **Rolf Hase**
Rédaction : **Rolf Gerstendorf**
Traduction : **Helmut Müller**
Mise en page : **Giel Dols**



PRODUITS

- **Coffret Arduino d'Elektor**
www.elektor.fr/elektor-arduino-electronics-bundle
- **Kit de capteurs Arduino**
www.elektor.fr/arduino-sensor-kit



LIENS

- [1] **Module GY-271** : www.az-delivery.de/en/products/gy-271-kompassmodul-kompass-magnet-sensor-fuer-arduino-und-raspberry-pi
- [2] **Guide de démarrage rapide** : GY-271 Kompassmodul Sensor (livre électronique gratuit) :
www.az-delivery.de/products/gy-271-kompassmodul-kompass-magnet-sensor-fuer-arduino-und-raspberry-pi
- [3] **Bibliothèque NeoPixel** : https://github.com/adafruit/Adafruit_NeoPixel
- [4] **Page du projet** : www.elektormagazine.fr/200597-04