

# détecteur de CO<sub>2</sub> : CO<sub>2</sub> Guard

Une solution « maison »  
pour surveiller la qualité de l'air

Mathias Claußen, Ton Giesberts et Luc Lemmens (Elektor)  
Conception Florian Schäffer (Allemagne)

Les concentrations élevées de CO<sub>2</sub> peuvent causer des problèmes dans les locaux mal aérés. Ce détecteur de CO<sub>2</sub> (CO<sub>2</sub> Guard) offre une protection dans ce domaine. Construit avec des composants à trous traversants, il avertit les utilisateurs à l'aide de trois LED et d'une alarme sonore lorsqu'il est temps de renouveler l'air. Les fichiers Gerber du circuit imprimé et les fichiers de conception de la face avant du boîtier peuvent être téléchargés gratuitement.



Le CO<sub>2</sub> Guard est un projet à réaliser vous-même pour la surveillance et l'évaluation de la qualité de l'air intérieur en mesurant la concentration de dioxyde de carbone (Indoor Air Quality, IAQ). Le dispositif est basé sur un capteur infrarouge non dispersif (NDIR). Il indique la qualité de l'air selon cinq plages, sur une échelle composée de trois LED. En outre, une alarme retentit si la concentration en CO<sub>2</sub> est trop élevée. Il est possible de transférer les valeurs mesurées par Wi-Fi (valeur de CO<sub>2</sub> et température) vers la plateforme d'analyse Ido ThingSpeak et d'analyser les données graphiquement.

La qualité de l'air dépend de nombreux facteurs, notamment la concentration en CO<sub>2</sub> dans une pièce, qui est ressentie presque immédiatement. Un taux trop élevé de ce gaz provoque de la fatigue, une baisse de la concentration, voire des nausées, sans parler des effets négatifs durables causés par son excès dans votre environnement. Ce dernier cas est (espérons-le) peu susceptible de se produire dans une maison ou un bureau, mais le détecteur CO<sub>2</sub> Guard peut vous alerter au bon moment de la nécessité de ventiler votre local.

## Circuit

Le cœur du schéma illustré à la **figure 1** est le circuit MOD2, Wemos D1 Mini, qui est un module Wi-Fi basé sur le SoC ESP8266EX avec une mémoire de programme flash de 4 Mo. Il collecte les données d'un capteur de CO<sub>2</sub> MH-Z19C (MOD1) via un logiciel UART, pilote trois LED pour indiquer les niveaux de CO<sub>2</sub> (LED 1... 3), et contrôle le buzzer BZ1, qui sert d'alarme sonore lorsque le niveau de CO<sub>2</sub> est excessif. Le CO<sub>2</sub> Guard est alimenté via le connecteur micro-USB du module Wemos, par un adaptateur secteur standard de 5 V CC. Cette tension d'alimentation est acheminée vers la broche 9 du MOD2 via une diode et un fusible internes, et vers un régulateur de tension 3,3 V embarqué. La sortie de ce régulateur à faible chute de tension (LDO) est utilisée pour l'alimentation interne du module Wemos lui-même et est également connectée à la broche 8 du module. La tension de 3,3 V peut être utilisée pour alimenter un petit ventilateur connecté à K1, comme nous le verrons plus tard. La sortie 5 V du circuit MOD2 sert à alimenter le capteur de CO<sub>2</sub> en plus d'une LED à l'intérieur du commutateur S1 servant à indiquer que la connexion Wi-Fi est active. Elle peut



également être connectée à K1 (via R6) comme source d'alimentation alternative pour le ventilateur. Le bouton-poussoir S1 peut déclencher le capteur de CO<sub>2</sub> pour lancer l'étalonnage automatique.

### Alimenter le ventilateur

Dans ce projet, le ventilateur n'est pas utilisé pour le refroidissement, mais pour assurer un flux d'air autour du capteur. Bien qu'il semble évident d'alimenter un ventilateur de 5 V à sa tension nominale, si le débit d'air maximal n'est pas le critère de conception le plus important — comme dans ce projet — il est recommandé de minimiser le bruit de cet appareil en réduisant le débit d'air via la tension d'alimentation. Avec notre CO<sub>2</sub> Guard, il existe quelques options à cet égard.

Le rail d'alimentation de 3,3 V du module Wemos est régulé par un régulateur LDO dans un boîtier SOT23-5. Le courant de sortie maximal de ce régulateur est limité à 150 mA grâce à son boîtier. L'ESP8266 du module Wemos D1 Mini lui-même utilise une partie importante du courant de sortie. Une surcharge du LDO peut entraîner une instabilité de l'ESP8266 et donc des crashes aléatoires. Pour éviter ce cas, vous pouvez connecter le ventilateur au 5 V via une résistance (R6, l'empreinte sur le circuit imprimé correspond à une résistance de 1 W). Cela permet de réduire encore plus le flux d'air et, surtout, le bruit. Notez que les ventilateurs fonctionnant à une tension inférieure à celle pour laquelle ils sont conçus ont tendance à avoir des problèmes au démarrage ou à ne pas démarrer du tout. Si le ventilateur doit être directement connecté au 5 V, vous pouvez insérer un pontage à la place de la résistance R6.

### Autre matériel

Le buzzer est également alimenté en 5 V. Le niveau de sortie haut des E/S du D1 Mini est de 3,3 V. Pour désactiver le buzzer, le diviseur de tension R4/R7 est donc connecté à la base du transistor PNP T1. Certains buzzers sont inductifs et susceptibles de produire des pics de tension, notamment lors de la mise hors tension. La diode D1 les empêche d'endommager le transistor. Le circuit imprimé comporte des empreintes pour les buzzers de 6,5 mm et 7,62 mm.

Nous avons utilisé un interrupteur doté d'un indicateur optique sous la forme d'un anneau bleu pour S1. Aucune résistance externe n'est nécessaire pour la LED à l'intérieur de l'interrupteur. Les broches marquées + et - sur le circuit imprimé (PCB), à côté de celles de l'interrupteur, sont les deux connexions pour la LED. Quatre fils sont reliés au connecteur à quatre voies S1 (ou soudés directement sur le circuit imprimé). Bien sûr, vous pouvez aussi utiliser un interrupteur et une LED séparément. R5 définit alors le courant pour la LED et la conception du panneau avant doit être modifiée pour tenir compte de cet indicateur optique. Avec l'interrupteur proposé dans la liste des composants, la résistance R5 n'est pas requise et doit être remplacée par un pontage.

### Circuit imprimé

La conception et les fichiers Gerber du circuit imprimé illustrés à la **figure 2** peuvent être téléchargés à partir du référentiel GitHub [1] de ce projet. Avec les fichiers Gerber, vous pouvez commander la carte auprès de votre fabricant préféré. Tous les composants sont à trous

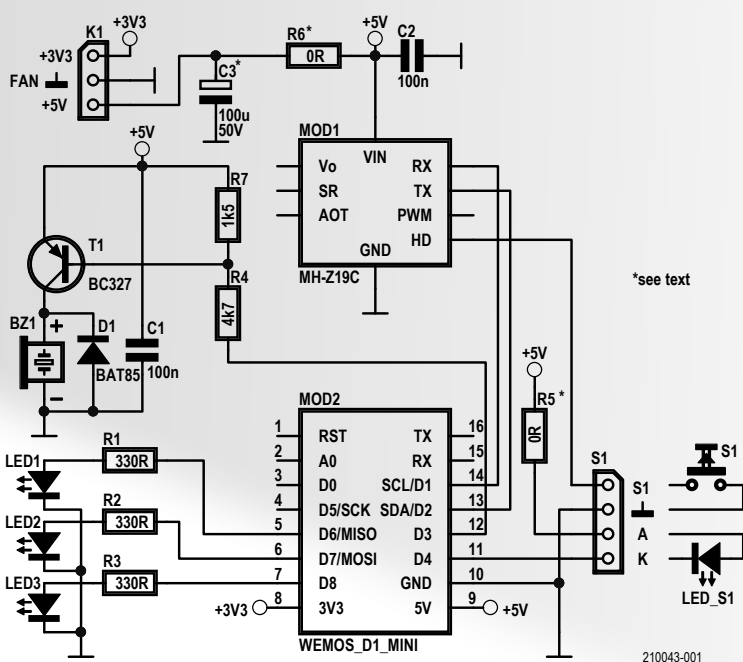


Figure 1. Schéma du CO<sub>2</sub> Guard.

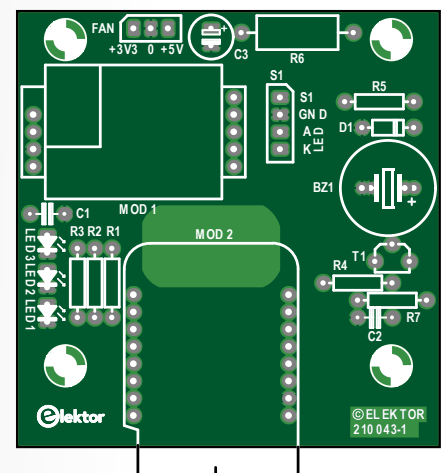


Figure 2. Circuit imprimé su CO<sub>2</sub> Guard.

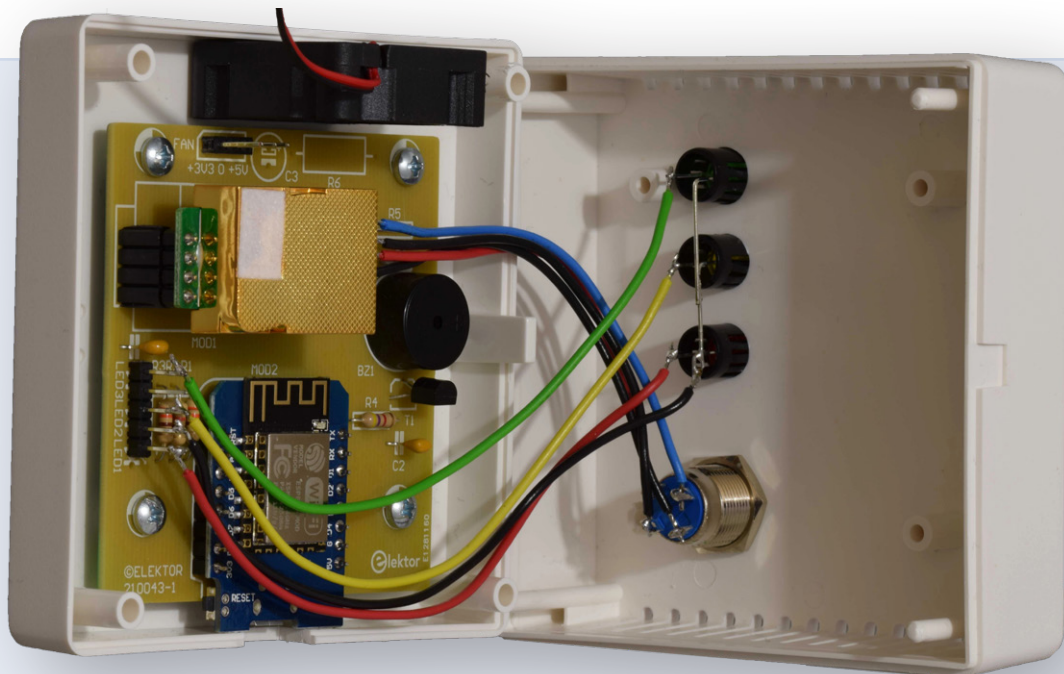


Figure 3. Ventilateur monté.

traversants et le circuit imprimé peut être à une seule face ; la couche supérieure est seulement constituée de cuivre pour éviter les coûts additionnels des circuits imprimés monocouche. Ainsi, même pour les makers débutants, le soudage ne sera pas trop difficile en suivant les directives données. Et avec toutes les traces sur la couche inférieure de ce circuit, il est relativement facile de dessouder les composants en cas d'erreur. Commencez par assembler les composants les plus petits, les résistances et la diode, puis les condensateurs, le transistor, le buzzer et les connecteurs (supports pour le capteur de CO<sub>2</sub> MOD1, connecteurs optionnels pour l'interrupteur et les LED). Pour monter le module D1 Mini (MOD2), soudez d'abord les deux connecteurs mâles à huit broches sur le côté du module où se trouvent le connecteur USB et la plupart des composants (voir **figure 3**). Il est recommandé de ne pas utiliser de supports de circuits (sockets) sur le circuit pour MOD2, car le circuit subit une certaine force lors du branchement et du débranchement du connecteur USB. Il est préférable de souder D1 Mini sur le circuit. De plus, l'utilisation de supports de circuits pourrait limiter le flux d'air à travers les trous de ventilation du boîtier.

Lorsque le circuit imprimé est fixé à l'aide de quatre vis sur la moitié inférieure du boîtier et le ventilateur collé (voir ci-dessous), le module du capteur peut être branché dans les supports correspondants. **Ne touchez jamais le dessus du capteur**, comme indiqué dans la fiche technique [2] : « Veuillez éviter la pression de son enceinte en plastique doré, quelle que soit la direction, pendant la soudure, l'installation et l'utilisation. En particulier, ne touchez jamais l'entrée d'air (filtre à poussière blanche). »

Les supports à quatre et cinq broches du capteur seront probablement fournis sous la forme d'un support SIL à 10 broches. Coupez ce bloc au niveau de la cinquième broche pour obtenir deux supports distincts à quatre et cinq broches.

## Préparer le boîtier

Commencez par utiliser une copie du schéma du panneau pour les positionner correctement dans les quatre trous (trois LEDs et l'interrupteur), et utilisez un poinçon pour marquer les quatre centres. Sur la page de ce projet sur Elektor Labs [3], vous trouverez des conseils et astuces pour éviter d'endommager le boîtier lors du perçage de plus

gros trous dans le panneau avant en plastique. Avec ce type de boîtier, faites surtout attention aux longues broches qui servent à le fermer ; elles risquent de se casser si elles sont mal manipulées !

Une entaille doit être faite sur un côté de la moitié inférieure du boîtier pour accéder au connecteur micro-USB du module D1 Mini, qui sert d'alimentation pour le CO<sub>2</sub> Guard. Assurez-vous que l'entaille est suffisamment grande pour la prise USB de votre alimentation. Placez le mini module D1 sur le circuit imprimé pour marquer l'emplacement correct. À ce stade, il n'est pas encore nécessaire de souder le module avec les connecteurs mâles correspondants. Avec le module placé sur le circuit imprimé et ce dernier posé correctement sur les vis de fixation (vous pouvez utiliser deux vis temporaires), la position de l'encoche peut être repérée. Elle est légèrement décentrée.

Vous n'avez pas besoin de faire de trous pour le flux d'air du ventilateur ; celui-ci est collé sur la face inférieure du boîtier avec de la super colle ou de la colle époxy (**figure 3**). Pour éviter toute résonance, le ventilateur ne doit pas toucher le couvercle de l'enceinte. Le ventilateur doit souffler l'air hors du boîtier. Avant de le monter, vérifiez la direction du courant d'air en le connectant à une alimentation de 3,3 V à 5 V ou vérifiez s'il y a une flèche sur le côté du cadre du ventilateur indiquant la direction du flux. Faites attention à la polarité des fils. Une mauvaise polarité endommagera le ventilateur. (Les modèles modernes comportent souvent des circuits à l'intérieur).

## Connexion des composants du panneau avant

Pour connecter les LED, quatre fils suffisent. La connexion à la masse entre le circuit imprimé et les trois LED sur la face avant peut être commune, il suffit de connecter les trois cathodes des LED et d'utiliser un seul fil.

Assurez-vous qu'aucun fil ne touche l'antenne du circuit MOD2 lorsque le couvercle du boîtier est finalement déposé sur la partie inférieure. Gardez-les aussi loin que possible du module Wemos, mais ne les allongez pas plus que nécessaire. Les fils connectés à l'interrupteur doivent avoir une longueur de 8,5 cm seulement et ceux connectés aux LED d'environ 12 cm. Avec cette longueur, vous pouvez enlever le haut du boîtier et le placer sur le côté, vers le bas, tout en gardant tous les fils connectés. Pas besoin de vis pour fermer le boîtier ; quatre



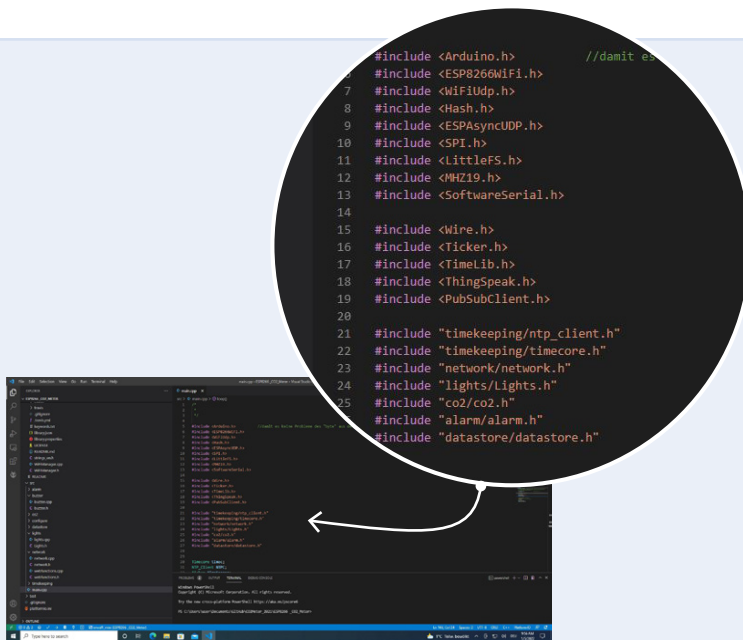


Figure 4. Code chargé dans l'IDE PlatformIO.

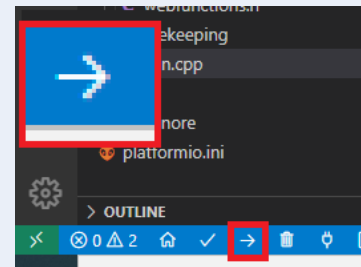


Figure 5. Bouton pour le téléversement du micrologiciel.

longues broches en plastique du couvercle glissent dans quatre longs bossages de montage.

Bien entendu, n'hésitez pas à créer votre propre design de panneau avant. Vous pouvez par exemple utiliser une LED bleue et un interrupteur séparés. Mais faites attention à l'emplacement des LED et de l'interrupteur ; ils ne doivent pas être situés trop près des modules.

## Téléversement du micrologiciel sur Wemos

Le micrologiciel de l'ESP8266 sur le module Wemos est créé à l'aide du framework Arduino, vous avez donc besoin du sketch et d'un ensemble de bibliothèques. Pour le développement du code de CO<sub>2</sub> Guard, nous utilisons PlatformIO et Visual Studio Code. Les bibliothèques, le projet et les paramètres de téléversement sont combinés afin de faciliter la compilation du code.

Pour installer Visual Studio Code et PlatformIO, vous pouvez suivre le guide [4] fourni par PlatformIO. Après l'installation, vous devez utiliser le code source du dépôt GitHub d'Elektor. Si vous voulez en savoir plus sur la manière d'obtenir le code, regardez la vidéo de notre collègue Clemens Valens [5]. De plus, si vous souhaitez en savoir plus sur GitHub et l'utilisation des dépôts, visionnez le webinaire Elektor sur GitHub [6]. Une fois que vous avez le code et que vous l'ouvrez avec l'IDE PlatformIO, vous devriez voir ce qui est reproduit sur la **figure 4**. Connectez Wemos à votre ordinateur et appuyez sur le bouton de téléversement (**figure 5**). PlatformIO téléchargera toutes les bibliothèques requises, les outils, les packages de gestion de la carte et commencera à effectuer des compilations, puis des téléversements. À ce stade, PlatformIO a établi et lancé le micrologiciel. Comme pour la plupart des projets basés sur l'ESP8266, il y a une deuxième étape : comme le micrologiciel offre une configuration via un serveur Web, nous devons également créer les pages Web sur l'ESP8266. Cliquez dans Visual Studio Code sur l'icône de la petite fourmi et vous devriez voir apparaître une liste de tâches (**figure 6**). Cliquez d'abord sur *Build Filesystem Image*, puis sur *Upload Filesystem Image*. Cela permettra d'écrire le contenu du serveur Web sur l'ESP8266. Si vous obtenez des erreurs du type «*can't access COM port*», assurez-vous que vous n'avez pas de terminal série ouvert ce qui bloquerait le port. Ensuite, le micrologiciel est prêt à être configuré et utilisé.

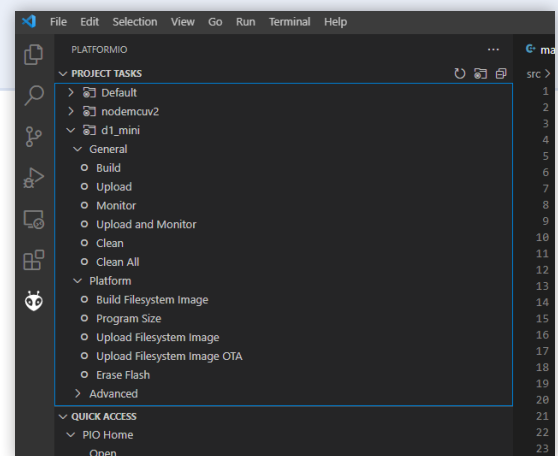


Figure 6. L'icône de la petite tête de fourmi montre plus d'options.

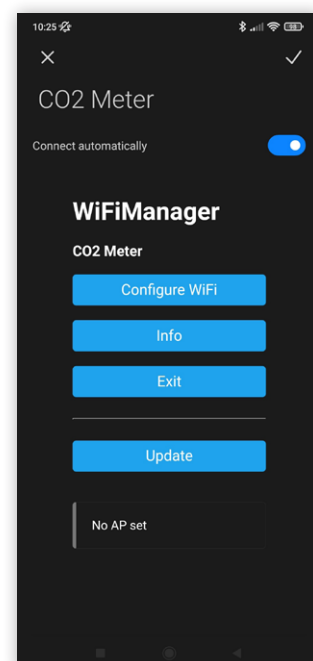


Figure 7. Paramètres du Wi-Fi.



Menu
Main page
Time settings
ThingSpeak
MQTT settings
WiFi settings
Restart

CO2

Current Value  
936 ppm

Notes

Up to 500 characters for your notes

Update Notes

Figure 8. Interface web du CO<sub>2</sub> Guard.

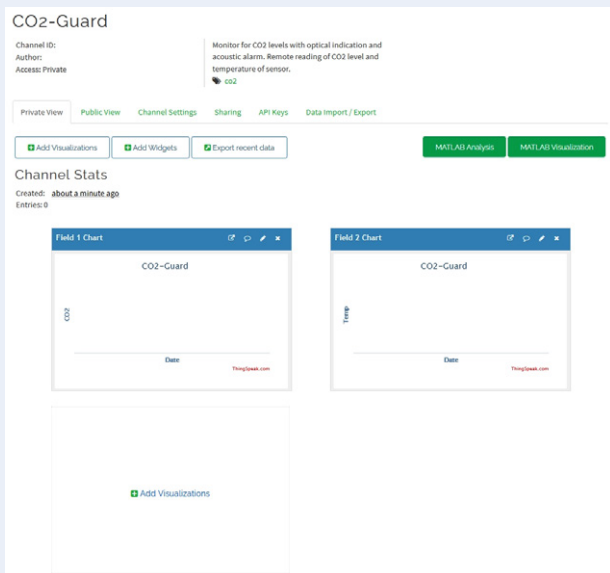


Figure 10. Nouveau canal ThingSpeak généré.

ThingSpeak

Channels Apps Devices Support

New Channel

Name

CO2\_Guard

Description

Monitor for CO2 levels with optical indication and acoustic alarm. Remote reading of CO2 level and temperature of sensor.

Field 1

CO2

Field 2

Temp

Field 3

Field 4

Field 5

Field 6

Field 7

Field 8

Metadata

Tags

CO2

Link to External Site

http://

Link to GitHub

https://github.com/

Elevation

Show Channel Location

Latitude

0.0

Longitude

0.0

Show Video

YouTube

Vimeo

Video URL

http://

Show Status

Save Channel

Help

Channels store all the data that a ThingSpeak application collects. Each channel includes eight fields that can hold any type of data, plus three fields for location data and one for status data. Once you collect data in a channel, you can use ThingSpeak apps to analyze and visualize it.

Channel Settings

- Percentage complete:** Calculated based on data entered into the various fields of a channel. Enter the name, description, location, URL, video, and tags to complete your channel.
- Channel Name:** Enter a unique name for the ThingSpeak channel.
- Description:** Enter a description of the ThingSpeak channel.
- Field:** Check the box to enable the field, and enter a field name. Each ThingSpeak channel can have up to 8 fields.
- Metadata:** Enter information about channel data, including JSON, XML, or CSV data.
- Tags:** Enter keywords that identify the channel. Separate tags with commas.
- Link to External Site:** If you have a website that contains information about your ThingSpeak channel, specify the URL.
- Show Channel Location:**
  - Latitude:** Specify the latitude position in decimal degrees. For example, the latitude of the city of London is 51.5072.
  - Longitude:** Specify the longitude position in decimal degrees. For example, the longitude of the city of London is 0.1278.
  - Elevation:** Specify the elevation position meters. For example, the elevation of the city of London is 35.032.
- Video URL:** If you have a YouTube® or Vimeo® video that displays your channel information, specify the full path of the video URL.
- Link to GitHub:** If you store your ThingSpeak code on GitHub®, specify the GitHub repository URL.

Using the Channel

You can get data into a channel from a device, website, or another ThingSpeak channel. You can then visualize data and transform it using ThingSpeak Apps.

See Get Started with ThingSpeak® for an example of measuring dew point from a weather station that acquires data from an Arduino® device.

Learn More

Figure 9. Paramètres de ThingSpeak.

Menu
Main page
Time settings
ThingSpeak
MQTT settings
WiFi settings
Restart

ThingSpeak Settings

Upload Enabled

True

Upload interval

10

Minutes

Write API Key

www.elektor.com

Figure 11. Paramètres de ThingSpeak.

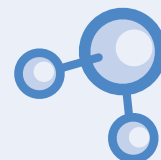
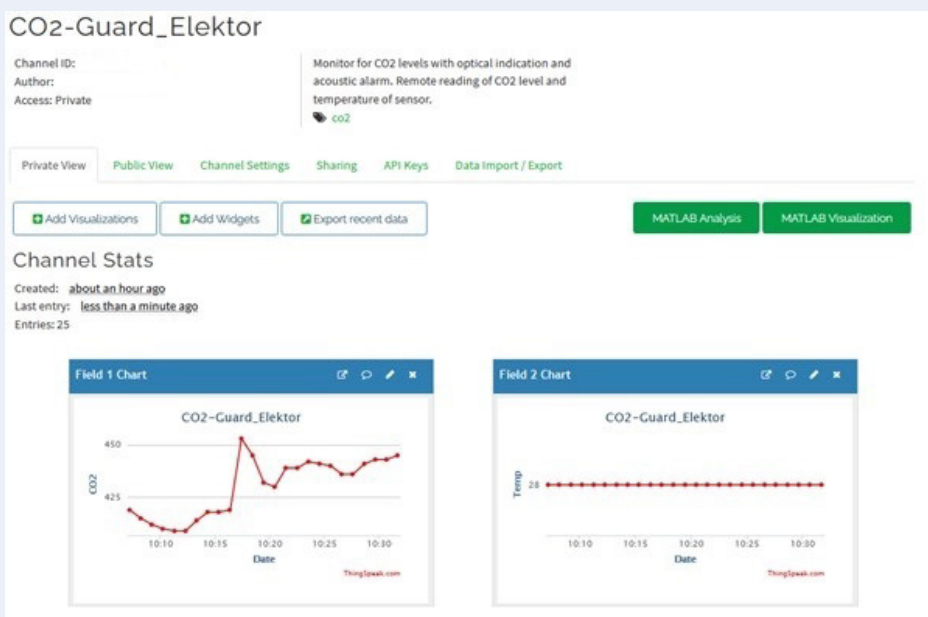


Figure 12. Données reçues.



## Connectivité Wi-Fi et Cloud

Si votre ESP8266 sur le module Wemos ne parvient pas à se connecter à un réseau Wi-Fi, il lancera un point d'accès appelé « CO<sub>2</sub> Meter ». Connectez-vous à ce point et ouvrez un navigateur pointant vers <http://192.168.4.1>. Vous devriez voir l'interface de configuration WiFi Manager (**figure 7**). Cliquez sur Configure WiFi, sélectionnez votre réseau et appliquez les paramètres requis avant de cliquer sur Save. L'ESP8266 tentera de se connecter à votre réseau. S'il peut établir une connexion, vous pouvez utiliser votre réseau <http://co2guard.local> pour accéder à l'interface Web (**figure 8**). À partir de là, vous pouvez commencer à configurer le CO<sub>2</sub> Guard.

Si vous le souhaitez, vous pouvez connecter le CO<sub>2</sub> Guard à une solution de stockage de données Web comme ThingSpeak ou un courtier MQTT, comme celui utilisé dans Node-RED. Avec ce dernier, vous pouvez automatiser encore plus le détecteur, par exemple en envoyant un message à votre téléphone portable si le CO<sub>2</sub> dépasse un certain niveau. Pour démarrer avec ThingSpeak, nous vous proposons un guide rapide de la configuration du compte.

## ThingSpeak : création d'un compte MathWorks

Si vous souhaitez uniquement surveiller la concentration actuelle en CO<sub>2</sub> dans votre pièce, vous n'avez pas besoin d'une connexion Wi-Fi et certainement pas d'un accès à un compte ThingSpeak. Les LED (et dans les cas extrêmes, le buzzer) donnent alors une indication claire de la qualité de l'air, du moins en ce qui concerne le CO<sub>2</sub>. Mais si vous souhaitez, par exemple, enregistrer l'évolution de ces mesures et analyser les données au cours du temps, ou pouvoir visualiser de loin la concentration actuelle, alors la connexion à ThingSpeak via le Wi-Fi est un choix évident. Vous pouvez également suivre la température, qui est mesurée par le module de capteur, mais qui n'est pas affichée sur le CO<sub>2</sub> Guard lui-même.

Pour créer un canal en ligne pour vos mesures, accédez au site Web de ThingSpeak [7] et choisissez *Get Started For Free*. Sur la page suivante, vous pouvez créer un compte MathWorks (si vous n'en avez pas déjà un). Ensuite, créez un nouveau canal et entrez les données requises comme indiqué dans la **figure 9**. Cliquez sur *Save Channel*. Un aperçu du canal qui vient d'être créé est présenté comme illustré à la **figure 10**.

Dans l'onglet *API Keys*, nous devons copier la valeur *Write API Key* pour une utilisation ultérieure. Gardez-la privée et ne la montrez à personne. Accédez à l'interface web du CO<sub>2</sub> Guard et sélectionnez l'option *ThingSpeak* dans le menu latéral. Une page contenant les paramètres de ThingSpeak s'affiche. Saisissez l'*API Key*, sélectionnez un intervalle de téléchargement et activez ThingSpeak (**figure 11**). Vous revenez maintenant sur le site ThingSpeak de MathWorks. Dans le menu *Channels*, choisissez *My Channels*, puis l'onglet *Private View*. Les premières mesures devraient apparaître au bout de quelque temps, comme dans la **figure 12**.

Pour partager vos données avec d'autres personnes, vous pouvez ajouter des utilisateurs individuels avec une autorisation de visualisation. Vous pouvez également rendre le canal public pour que vos données soient accessibles. Les utilisateurs individuels recevront un courrier électronique à l'adresse e-mail saisie dans l'onglet *Sharing*. Le partage des données est limité à trois autres utilisateurs si vous

utilisez un plan gratuit (non payant) pour ThingSpeak. Pour plus de détails sur le partage des données, vous pouvez consulter le centre d'aide de MathWorks [8].

## Connexion MQTT

MQTT est un moyen flexible de distribuer vos données à d'autres systèmes qui peuvent les stocker ou les traiter. Vous n'avez pas besoin d'un service externe pour collecter les données ou les traiter, et vous pouvez conserver les données dans votre propre réseau. Un des outils qui offre une connectivité MQTT est Node-RED (par exemple, sur un Raspberry Pi). Nous avons couvert beaucoup d'installations pour Node-RED dans le passé, s'il existe un guide d'installation rapide pour votre Raspberry Pi sur le site web de Node-RED [9]. Pour MQTT, dans l'interface web du CO<sub>2</sub> Guard, vous pouvez indiquer un serveur et une rubrique où les données doivent être publiées (**figure 13**). Les données elles-mêmes sont transférées sous la forme d'une chaîne JSON, comportant les objets « CO<sub>2</sub>Value » et « Temp », présentant la valeur du CO<sub>2</sub> en ppm et la température en °C. Si vous souhaitez en savoir plus sur Node-RED en utilisant un Raspberry Pi, consultez le livre de Dogan Ibrahim, *Programming with Node-RED* (Elektor 2020) [10].

## Fichiers de conception de circuits imprimés KiCad

À l'origine, le circuit imprimé a été conçu dans Altium, et les fichiers Gerber fournis sont basés sur cette démarche éprouvée. La conception est assez simple, comme vous pouvez le voir sur les schémas. Comme le schéma n'est pas très complexe (**figure 1**), il a été reproduit dans KiCad. Outre le schéma, le circuit imprimé doit aussi être transféré. Le projet ayant été entamé dans la dernière version stable

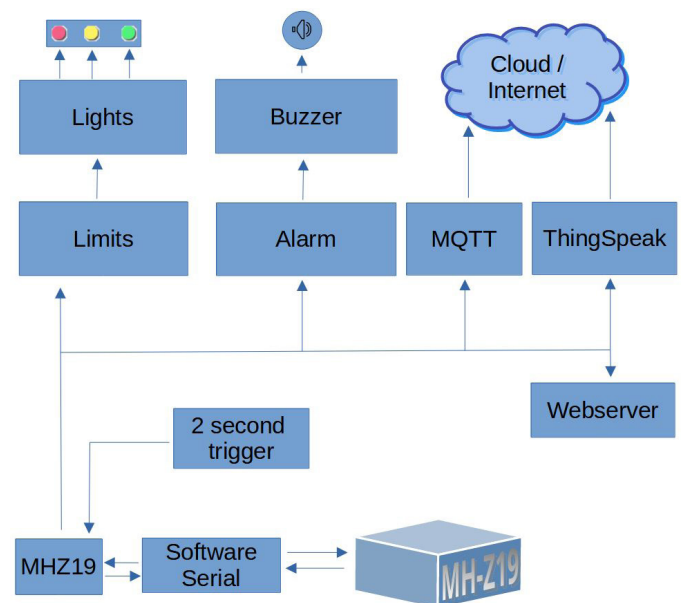


Figure 13. Paramètres MQTT.




de KiCad 5, il n'y avait pas d'importation native (intégrée) pour les cartes Altium. L'astuce est d'utiliser les fichiers Gerber fournis et de les importer dans KiCad. Le visualiseur Gerber de KiCad est capable de générer des fichiers de carte que vous pouvez utiliser dans l'outil de dessin. Il fournit non seulement les contours de la carte, mais aussi les empreintes et les trous de montage à utiliser pour le placement des composants. Cette méthode peut être utilisée pour importer vos projets dans KiCad, mais il est préférable d'utiliser la version 6 de KiCad et l'importation native de cartes Altium si vous en avez des fichiers. Mais pourquoi faire cette conversion ? Tout d'abord, cela permet à d'autres personnes de travailler directement avec les fichiers KiCad et de les modifier si elles le souhaitent. De plus, avec certains services de production de PCB, il suffit de glisser-déposer un fichier de carte KiCad pour que votre carte soit fabriquée (par exemple, chez Aisler) [11]. Vous n'avez pas besoin d'effectuer l'exportation vers des fichiers Gerber et de perçage pour générer un fichier ZIP. Cependant, si votre service PCB a besoin de ces fichiers, vous pouvez toujours les générer dans KiCad selon les spécifications requises par votre service.

La combinaison de KiCad et FreeCad est une solution très efficace si vous souhaitez concevoir votre propre boîtier pour le circuit imprimé, comme le montre le projet de Noël [12]. Imprimer un boîtier est pour certains plus pratique que de percer des trous dans un autre préfabriqué. Si vous avez chez vous une imprimante 3D inutilisée, pourquoi ne pas vous en servir ? De plus, cela fera un composant de moins à commander. Au moment de la rédaction de cet article, le boîtier imprimable en 3D est en cours de conception. En attendant, si vous voulez utiliser un boîtier, vous devez percer quelques trous.

## Construire le vôtre

Construire un détecteur de CO<sub>2</sub> pour votre maison ou votre espace de travail n'est pas si difficile, et comme vous le savez, de nombreux dispositifs de mesure de CO<sub>2</sub> ont été développés sous la forme de

projets « maison » et commerciaux. Le capteur MH-Z19 utilisé dans ce projet constitue un outil de mesure abordable et fiable. L'ESP8266 est un microcontrôleur Wi-Fi éprouvé qui peut non seulement être programmé avec le framework Arduino, mais qui prend également en charge le Home Assistant.

Vous voulez construire le vôtre ? Le capteur vous coûtera environ 30 € (en Europe). L'interrupteur Wemod D1 mini, le circuit imprimé, les LED, le boîtier, le ventilateur et les frais d'expédition s'ajouteront à la facture. Le coût total des composants s'élèvera donc à environ 60 €. D'autre part, la surveillance de la concentration en CO<sub>2</sub> dans vos pièces présente de nombreux avantages. Connaissant la concentration en CO<sub>2</sub>, vous pouvez vous assurer que le renouvellement de l'air est suffisant. 

210043-04

## Des questions, des commentaires ?

Contactez Elektor ([redaction@elektor.fr](mailto:redaction@elektor.fr)).

## Contributeurs

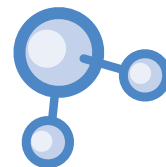
Conception originale : **Florian Schäffer**

Conception et texte : **Mathias Claußen, Ton Giesberts, Luc Lemmens**

Rédaction : **Jens Nickel, C. J. Abate**

Traduction : **Asma Adhimi**

Mise en page : **Harmen Heida**



## Logiciel

Certains aspects du logiciel initialement fourni par le développeur sur la plate-forme Elektor Labs pourraient être améliorés. Par exemple, les identifiants codés en dur pour votre réseau Wi-Fi ne fonctionnent que si vous ne changez jamais le nom du réseau ou le mot de passe auquel l'appareil est connecté. De plus, garder vos identifiants quelque part dans des fichiers de code n'est pas le moyen le plus sûr de les stocker. Il en va de même pour l'API Key ThingSpeak. Si vous souhaitez modifier cette clé, vous devez recompiler tout le code et le télécharger sur l'appareil.

Elektor Lab n'a gardé dans le code (sous forme de `#defines` dans `/configure/configure.h`) que l'affectation des broches, ainsi que les seuils de CO<sub>2</sub> pour les trois LED et le buzzer. Dans le fichier `/configure/configure.h`, vous avez tous les paramètres codés en dur à un seul emplacement et vous pouvez les modifier si nécessaire. Bien sûr, après modification, vous devrez recompiler le micrologiciel et le télécharger sur l'ESP8266.

Dans le logiciel, le flux de données est assez simple, comme vous pouvez le voir sur l'image. Toutes les deux secondes, les valeurs des capteurs sont vérifiées et renvoyées aux différents modules de code. Pour la surveillance locale du niveau de CO<sub>2</sub>, nous avons la classe `limits`, qui contrôlera finalement nos trois LED, et nous avons la classe `alarm`, qui fera sonner le buzzer si un certain niveau de CO<sub>2</sub> est dépassé. Ces classes sont créées de manière générique, ce qui permet de changer facilement la disposition des LED et la façon d'afficher les valeurs. Il en va de même pour le buzzer. Si vous souhaitez utiliser un buzzer actif ou même un module sonore, cette classe est suffisamment flexible pour le faire. Pour la connectivité avec le cloud, nous avons utilisé, dans la mesure du possible, des bibliothèques prédéfinies, notamment pour ThingSpeak. Cela simplifie considérablement l'échange de données avec ce service par rapport à la station météorologique ESP32 [13] que nous avons publiée [14] il y a quelques années.



## LISTE DES COMPOSANTS

### Résistances

R1, R2, R3 = 330 Ω, 0,25 W, 5 %

R4 = 4,7 kΩ, 0,25 W, 5 %

R5 = 0 Ω, connecter un fil, voir le texte

R6 = 0 Ω, connecter un fil pour 5 V ou laisser ouvert, voir le texte

R7 = 1,5 kΩ, 0,25 W, 5 %

### Condensateurs

C1, C2 = 100 nF, 50 V, 10 %, X7R, espacement des fils 5 mm

C3 = 100 µF, 25 V, optionnel, non monté, voir texte

### Semi-conducteurs

D1 = BAT85, DO-35

T1 = BC327, PNP, TO-92

LED1 = LED rouge, 5 mm

LED2 = LED jaune, 5 mm

LED3 = LED verte, 5 mm

### Divers

MOD1 = connecteur 1x10 SIL, femelle, divisé en 1x4 et 1x5 (voir Texte)

MOD1 = MH-Z19C, version 400-5000 ppm, broches en bas

MOD2 = Wemos D1 mini avec 2x connecteur 1x8 SIL

S1 = interrupteur avec LED intégrée (marche momentanée, auto-réinitialisation, cercle bleu)

Boîtier PP73BL, Supertronic

BZ1 = buzzer CC, 5 V, pas 5 or 7,62 mm, diam. 14 mm max tube thermorétractable, I.D 2.4 mm, 20 cm

Boîtier de LED, pour LED 5 mm (LED1..3), noire

Ventilateur 5 V CC, 30x30x10,5 mm

Connecteur SIL mâle à 12 broches

Circuit imprimé 210043-1 v2.0

### Fil fin pour connecter les LED et l'interrupteur :

fil torsadé, 0,25 mm<sup>2</sup>, noir (2x8,5+12 cm)

fil torsadé, 0,25 mm<sup>2</sup>, rouge (8,5+12 cm)

fil torsadé, 0,25 mm<sup>2</sup>, jaune (12 cm)

fil torsadé, 0,25 mm<sup>2</sup>, vert (12 cm)

fil torsadé, 0,25 mm<sup>2</sup>, bleu (8,5 cm)



## PRODUITS

- **WeMos D1 mini Pro – ESP8266 based WiFi Module** [www.elektor.fr/19185](http://www.elektor.fr/19185)
- **PCB V2.1 with Components (excluding WeMos D1 and MH-Z19)** [www.elektormagazine.de/aisler/co2guard](http://www.elektormagazine.de/aisler/co2guard)
- **H. Henrik Skovgaard, IoT Home Hacks with ESP8266** [www.elektor.fr/19158](http://www.elektor.fr/19158)
- **D. Ibrahim, Programming with Node-RED (E-book)** [www.elektor.fr/19225](http://www.elektor.fr/19225)



## LIENS

- [1] Référentiel GitHub d'Elektor: <https://github.com/ElektorLabs/210043-CO2-Guard>
- [2] Fiche technique de Winsen MHZ-19: [www.winsen-sensor.com/d/files/infrared-gas-sensor/mh-z19b-co2-ver1\\_0.pdf](http://www.winsen-sensor.com/d/files/infrared-gas-sensor/mh-z19b-co2-ver1_0.pdf)
- [3] « CO2 Guard » Elektor-Labs.com: [www.elektormagazine.fr/labs/co2-guard](http://www.elektormagazine.fr/labs/co2-guard)
- [4] Guide d'installation de PlatformIO : <https://platformio.org/install/ide?install=vscode>
- [5] « Comment coder (télécharger) ou cloner depuis GitHub » Elektor TV: [www.youtube.com/watch?v=X5e3xQBeqf8](http://www.youtube.com/watch?v=X5e3xQBeqf8)
- [6] « Webinaire GitHub : GitHub pour les débutants, » Elektor TV: [www.youtube.com/watch?v=j\\_LgvVhBdwQ](http://www.youtube.com/watch?v=j_LgvVhBdwQ)
- [7] MathWorks ThingSpeak: <http://thingspeak.com>
- [8] Aide MathWorks ThingSpeak: <https://bit.ly/ChannelPropertiesMATLAB>
- [9] Guide d'installation de Node-RED : <https://nodered.org/docs/getting-started/raspberrypi>
- [10] D. Ibrahim, Programming with Node-RED, Elektor 2020: [www.elektor.fr/programming-with-node-red-e-book](http://www.elektor.fr/programming-with-node-red-e-book)
- [11] AISLER B.V: <http://aisler.net>
- [12] M. Claußen, « DIY Christmas Fireplace: A 3D Puzzle with PCBs, LEDs, and Raspberry Pi Pico » ElektorMagazine.com: [www.elektormagazine.com/articles/diy-christmas-fireplace-3d-puzzle-pcbs-leds-raspberry-pi-pico](http://www.elektormagazine.com/articles/diy-christmas-fireplace-3d-puzzle-pcbs-leds-raspberry-pi-pico)
- [13] « ESP32 Weatherstation » Elektor-Labs.com: [www.elektormagazine.fr/labs/esp32-weather-station-180468](http://www.elektormagazine.fr/labs/esp32-weather-station-180468)
- [14] R. Aarts, « station météo à ESP32 » ElektorMag 1-2/2019: [www.elektormagazine.fr/magazine/elektor-71/42320](http://www.elektormagazine.fr/magazine/elektor-71/42320)