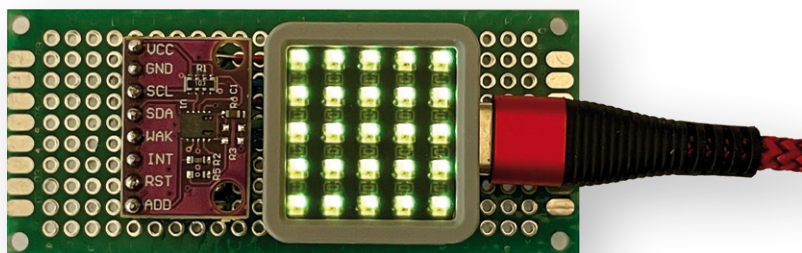


# eCO<sub>2</sub> Telegram bot



moniteur de qualité de l'air avec notification par Telegram

Peter Neufeld (Allemagne)

Déterminer la qualité de l'air intérieur ne nécessite pas toujours des appareils de mesure très précis et coûteux. Un simple indicateur plus un suivi des valeurs limites peuvent s'avérer très utiles si vous souhaitez simplement savoir quand aérer une pièce très fréquentée. Un tel appareil devient encore plus utile lorsque son simple afficheur intégré est complété par une transmission d'alertes vers un compte Telegram sur votre smartphone.

Une technologie de capteur hautement intégrée, un microcontrôleur moderne et un programme BASIC permettent, et sans grande complexité technique, d'utiliser ce détecteur de CO<sub>2</sub>e (équivalent CO<sub>2</sub>) et ses différents modes d'affichage pour indiquer une mauvaise qualité de l'air de quatre manières :

- LED NeoPixel locales comme indicateur visuel de la qualité de l'air
- Interface web pour les appareils locaux avec un navigateur web
- Requête manuelle via l'application de messagerie Telegram
- Message d'alerte Telegram envoyé à un utilisateur spécifique de Telegram

## Matériel

Le circuit simple de la **figure 1** est basé sur un capteur CCS811 eCO2 et un module SoC ESP32. La qualité de l'air est indiquée par une ou plusieurs LED NeoPixel. Un seul interrupteur à bouton-poussoir est connecté à une broche d'entrée. Le capteur et le SoC communiquent via un bus I<sup>2</sup>C à deux fils. L'écran NeoPixel n'utilise qu'une seule ligne de données au niveau du GPIO27, qu'il s'agisse d'une seule LED ou d'une matrice.

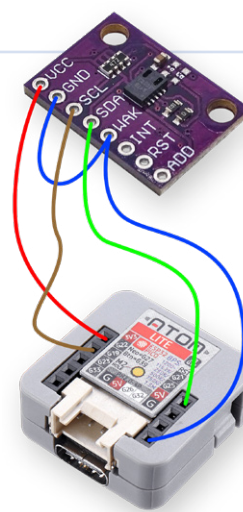
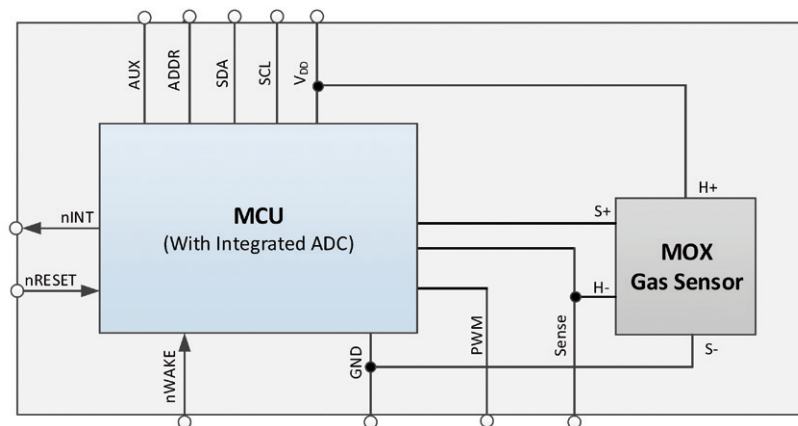


Figure 1. L'« eCO<sub>2</sub> Telegram bot » avec une matrice Atom M5Stack et un CCS811, y compris les circuits internes (source d'image : fiche technique AMS).

Bien que n'importe quel module ESP32 disposant de broches I<sup>2</sup>C convienne à cette tâche, je recommande l'utilisation d'un ATOM Matrix ou d'un ATOM Lite, de M5Stack. En effet, ces modules pratiques combinent un ESP32-PICO-D4 avec une antenne, et soit une matrice NeoPixel, soit une LED NeoPixel, ainsi qu'un bouton poussoir et d'autres composants, le tout dans un petit boîtier de protection compact et élégant. Ainsi, le capteur et le module ESP32 sont presque tout le matériel nécessaire pour construire l'appareil sur une carte perforée ordinaire. Pour ce faire, branchez le CCS811 et l'ATOM dans les connecteurs femelle et mâle appropriés, respectivement, et réalisez les différents branchements nécessaires comme le montre la **figure 2**. La face inférieure de la carte perforée traversante est isolée avec du ruban adhésif.

Le circuit est alimenté via le module ESP32 par une alimentation USB (5 V avec un courant maximal de 500 mA), soit via USB C, le connecteur HY2.0 Grove, ou les prises inférieures.

### Capteur de CO<sub>2</sub> équivalent CCS811

Le capteur CCS811 ne peut pas mesurer directement le taux de CO<sub>2</sub> ! Il calcule le taux équivalent en CO<sub>2</sub> (CO<sub>2</sub>e) en mesurant les VOC (composés organiques volatils totaux), la principale source de ces composés organiques volatils étant l'air expiré par les humains. Ce capteur semi-conducteur à oxyde métallique (MOS) est peu coûteux. Il recherche une valeur relative au « bon air » en déterminant les meilleures conditions d'air sur une longue durée, puis en supposant que le capteur se trouve dans un air frais et non pollué contenant 400 ppm de CO<sub>2</sub> à un moment donné. Mais, le capteur ne stocke pas cette valeur.

En outre, la sensibilité du capteur peut varier dans le temps et dans des conditions environnementales différentes, telles que la température et l'humidité. Ainsi, pour fournir des relevés CO<sub>2</sub>e fiables, le capteur nécessite :

- un temps de déverminage unique de plus de 48 heures et
- une période minimale de rodage d'environ 20 minutes après chaque démarrage à froid.

La fiche technique [1] contient des informations plus détaillées sur le capteur.

### Logiciel

Le programme a été développé avec Annex32, un interpréteur BASIC pour ESP32 [2]. Après avoir installé l'interpréteur dans la mémoire flash du module ESP32 via un programme d'installation et une interface USB série, l'interpréteur et son environnement de développement fonctionnent directement sur l'ESP32. On a seulement besoin d'un navigateur Chrome ou Firefox pour charger, éditer, tester et exécuter (automatiquement) le script BASIC. La version minimale requise d'Annex32 est V1.435, car elle prend en charge le CCS811 et le message Telegram. L'aide en ligne [3] d'Annex32 contient une introduction très utile à cet interpréteur BASIC. Les principales tâches du code BASIC sont les suivantes :

- Initialisation du CCS811 et relevé de la valeur du CO<sub>2</sub>e une fois par seconde.

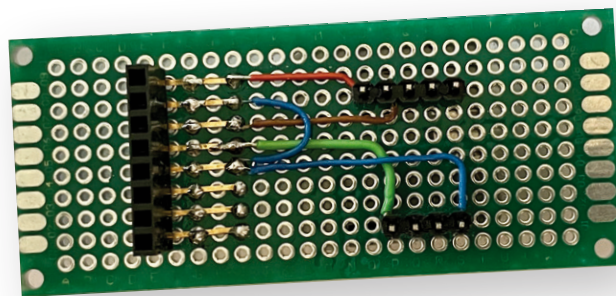
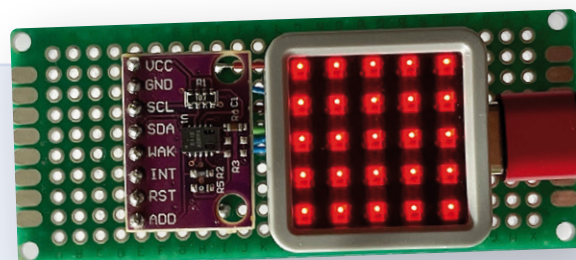


Figure 2. Soudure facile des deux modules sur la carte perforée.

- Classification de l'état de l'air ambiant par couleur VERT, JAUNE ou ROUGE selon le taux du CO<sub>2</sub>e.
- Affichage de cet état par la matrice NeoPixel intégrée ou par la couleur de la LED NeoPixel.
- Une interface web affiche la valeur du CO<sub>2</sub>e et la catégorie dans un navigateur sur le réseau local (W)LAN.
- Il est possible de consulter manuellement l'état et la valeur du CO<sub>2</sub>e via Telegram, car notre programme comprend un bot Telegram et récupère les commandes entrantes de l'utilisateur à partir du serveur Telegram.
- Un message d'alerte Telegram est automatiquement envoyé au dernier chat\_id Telegram lorsque la qualité de l'air est dans la zone rouge (**figure 3**).
- Il est possible d'enregistrer le niveau de référence de la qualité de l'air en appuyant sur le bouton frontal du module Atom ou en envoyant une commande Telegram.



AA

192.168.0.133



## eCO<sub>2</sub> ROOM 1

Time : 16:20:49

eCO<sub>2</sub>: 2028

**Condition: RED**

Figure 3. Message d'avertissement : l'air est vicié !



## Extrait du programme : données sur une page web

```
#####
MAKE_WEBPAGE:
'-----
'create the autorefreshing html page with dynamic
'display values and matching colors
A$ = ""
A$ = A$ + "<H1>eCO2 " + LOCATION$
A$ = A$ + "</H1>"
A$ = A$ + "Time : " + textbox$(T$) + "<br>"
A$ = A$ + "eCO2: " + textbox$(STR_eCO2$) + "<br>"
A$ = A$ + |<span style="color:| + COND_COL$ + |">|
A$ = A$ + "<H1>Condition: " + CONDITION$+ "</H1>"
A$ = A$ + "</span>"
cls
autorefresh 1000
html A$
return
#####
```

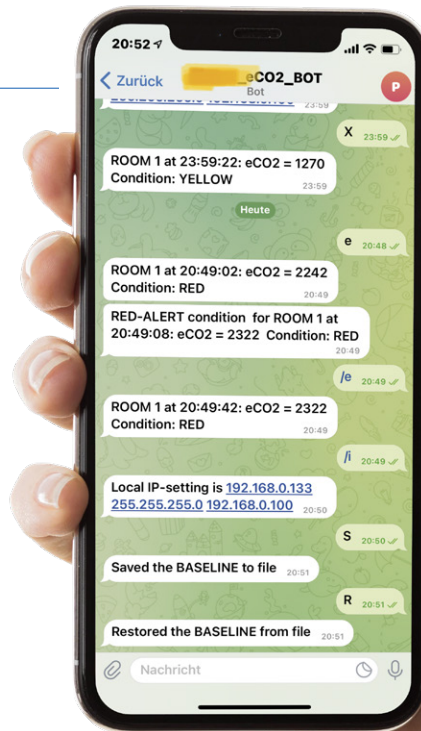


Figure 4. Contrôle via le bot Telegram.

La routine du bot Telegram consulte régulièrement les serveurs Telegram à la recherche de commandes entrantes. Il répond ensuite à ces commandes, comme le montre la **figure 4** :

- > **/e** renvoie la valeur du CO<sub>2</sub>e et la catégorie [VERT | JAUNE | ROUGE].
- > **/s** enregistre la valeur de référence dans */baseline.txt*.
- > **/r** rétablit la valeur de base à partir de */baseline.txt*.
- > **/i** renvoie les paramètres IP locaux du module.
- > [Tout autre caractère] effectue la même chose que **/e**.

## Votre propre jeton Telegram

Afin d'utiliser les fonctionnalités de Telegram dans le programme BASIC, vous devez d'abord créer votre propre bot Telegram en suivant les instructions de BotFather [4] dans votre application Telegram. Vous obtiendrez ainsi votre jeton Telegram personnel et un nom de bot. Attention : vous devez inclure ces deux informations dans le programme BASIC pour définir les variables appropriées.

L'utilisation d'Annex32 BASIC, dont vous pouvez toujours trouver la version actuelle dans le forum Annex RDS [5], a au moins l'avantage d'être facile à lire – même pour les développeurs débutants – et d'être adaptable à vos propres besoins. Cependant, le code [6] est abondamment commenté pour une meilleure compréhension des fonctions. Ce script doit être recopié dans l'éditeur web d'Annex32 et sauvegardé dans le module ESP32 en tant que fichier d'exécution automatique (*/default.bas*). ◀

210566-04

## Des questions, des commentaires ?

Envoyez un courriel à l'auteur (peter.neufeld@gmx.de) ou contactez Elektor (redaction@elektor.fr).



## Produits

- > **M5Stack AtomU ESP32 Development Kit (SKU 20184)**  
<https://elektor.fr/20184>
- > **ESP32-PICO-Kit V4**  
Sans connecteurs : <https://elektor.fr/18423>  
Avec connecteurs : <https://elektor.fr/20323>
- > **Dogan and Ahmet Ibrahim: The Official ESP32 Book (PDF)**  
<https://elektor.fr/18330>

## LIENS

- [1] Fiche technique du CCS811 : <https://bit.ly/2qQKqKu>
- [2] AnnexToolKit 1.1 y compris la version 1.48.22 :  
<https://bit.ly/3Gtf8RS>
- [3] Annex32 WIFI RDS Help Version 1.48.2:  
<https://bit.ly/3Gwhdg9>
- [4] BotFather sur Telegram : <https://t.me/BotFather>
- [5] Annex RDS forum: <https://bit.ly/3VBTRK2>
- [6] Ce projet sur Elektor Labs :  
<https://www.elektormagazine.fr/labs/eco2-telegram>