

Appareil de mesure de CO2

Auteur : Fabien · **Publié le** 29/04/2021 · 3 vues · 2 téléchargements PDF

Capteurs

MicroContrôleur Esp

L'objectif est de réaliser un capteur de CO2 connecté avec suivi au cours du temps.

Nous nous baserons sur l'utilisation du matériel suivant :

- Esp8266
- Capteur de qualité de l'air MQ135
- Capteur de température et d'humidité DHT22
- Résistance 4,7 kOhms

Vous pouvez également utiliser un capteur CO2 CCS811 et/ou un capteur DHT11.

Étapes du projet

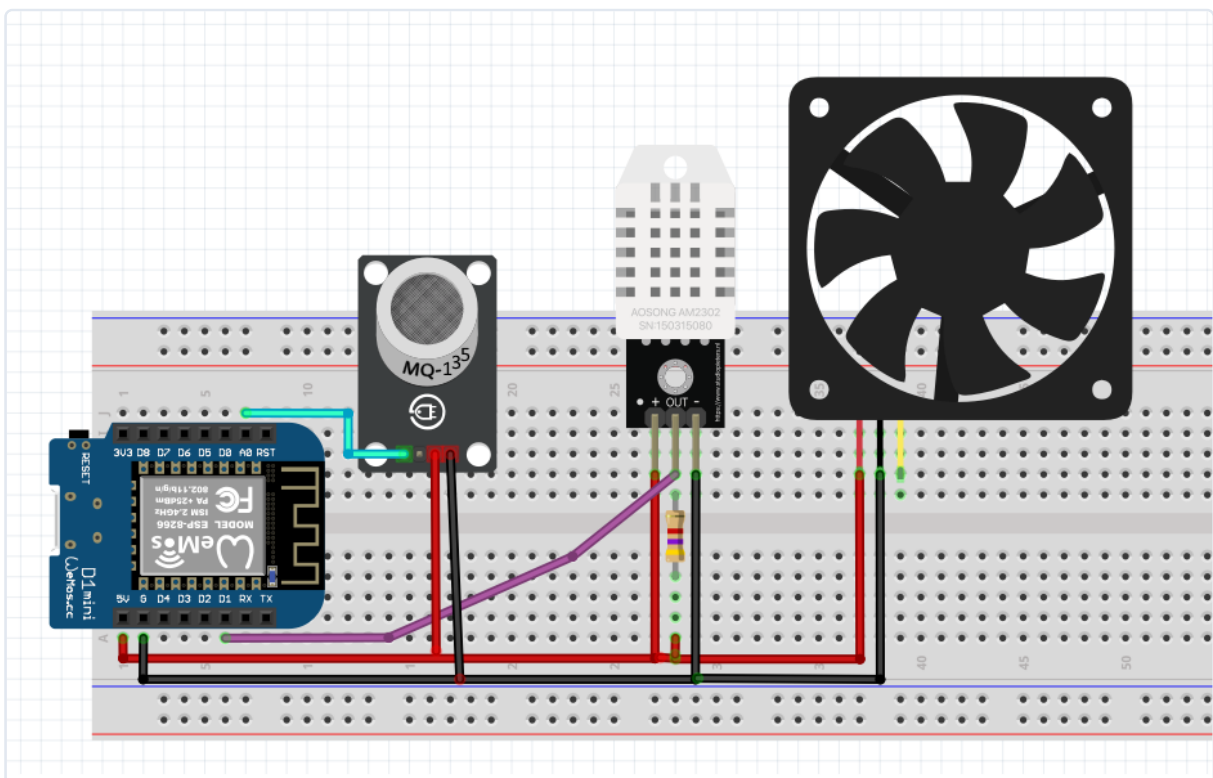
ÉTAPE 1

Le montage

Nous utiliserons l'entrée analogique A0 de notre ESP pour obtenir une valeur plus précise du taux de CO2 et la pin D1 pour lire les valeurs de température et d'humidité.

J'ai utilisé un ventilateur 5V pour assurer la bonne ventilation du capteur et ainsi obtenir une valeur correcte en temps réel.

Attention a bien mettre la résistance 4,7 kOhms à cheval entre la borne positive et la borne de lecture des datas du DHT.



ÉTAPE 2

Le capteur de qualité de l'air MQ-135

Ce capteur ne mesure pas seulement le CO₂ mais aussi l'ammoniac (NH₃), NO_x, l'alcool, le benzène et la fumée. Il est composé d'une résistance chauffante dont la valeur varie avec la quantité de ces gaz dans l'air.

Il va donc nous falloir mesurer la valeur de cette résistance en cours d'utilisation pour que la mesure obtenue soit la plus précise possible.

De plus, sa réponse n'est pas linéaire et varie en fonction de la température et de l'humidité ambiante. Il nous faudra donc adapter la lecture de l'entrée analogique.

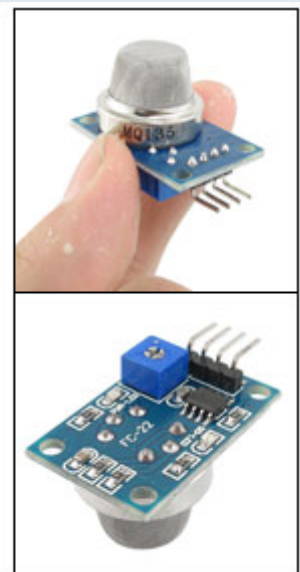
Les mesures correctes sont obtenues d'après les données constructeurs au bout de 24h de chauffe mais d'après mes mesures, j'ai obtenu des valeurs correctes après une heure de chauffe seulement.

Les valeurs sont données en ppm (parties par million) c'est-à-dire en nombre de particules par million de particules. Le taux normal actuellement est d'environ 400 ppm soit 0,04% de la composition de l'air ambiant.

MQ-135



DC 5V 10-1000ppm



ÉTAPE 3

Le code

ArduinoBlocks:

Si vous n'êtes pas un habitué d'ArduinoBlocks, commencez par cliquer sur Ressources en haut à droite de votre écran puis choisissez ArduinoBlocksConnector. Téléchargez la version correspondant à votre ordinateur puis installez le. Pendant ce temps là, vous pouvez vous inscrire sur le site pour réaliser votre projet.

Passons au code. Le capteur ne faisant pas partie de l'immense bibliothèque proposée par ArduinoBlocks il a fallu s'inspirer de celle disponible pour Arduino à l'adresse :

<https://github.com/GeorgK/MQ135>

Vous pouvez directement importer le code dans ArduinoBlocks à partir du fichier **abp** présent dans l'archive Code_Capteur_CO2.7z à télécharger sur cette page.

Vous n'avez pas à vous soucier du code en violet, il permet de faire la lecture des valeurs. La fonction qui nous intéressera vraiment est **GetCorrectedppm** qui permet de récupérer la valeur mesurée et corrigée d'après la température et l'humidité ambiantes.

Nous utiliserons également la fonction **getCorrectedRZero** dans la partie Calibration, de manière à étalonner notre appareil de mesure.

Arduino :

Vous trouverez le code en format **ino** directement dans l'archive Code_Capteur_CO2.7z à télécharger sur cette page. Il est directement généré par le site ArduinoBlocks par contre vous devrez avoir installé les librairies pour programmer les ESP, comme le montre la vidéo ci-dessous :

|

en utilisant les liens : https://dl.espressif.com/dl/package_esp32_index.json, http://arduino.esp8266.com/stable/package_esp8266com_index.json

ÉTAPE 4

L'étalonnage

Comme on l'a vu plus haut le capteur a besoin d'un temps de chauffe afin que nous déterminions la valeur de la résistance de notre capteur.

Pour cela nous allons utiliser ce que l'on appelle la **console**, qui permet de voir les réponses de notre ESP en temps réel (le bouton est placé en haut à droite, à coté du bouton téléverser). Il suffit de cliquer sur **Connecter** (il peut être nécessaire d'attendre que le wifi se connecte).

Nous pouvons voir que le programme contient des constantes de correction des mesures (**CorX**) mais celles qui nous concernent sont **ATMOCO2** et **RZERO**.

La première est le taux normal de CO2 dans l'atmosphère, elle peut se trouver [en cliquant ici](#) pour avoir les valeurs les plus récentes.

La seconde est la résistance propre de votre capteur, dans mon cas 73,89. Elle permet de transformer la mesure réalisée sur le port analogique de votre microcontrôleur en valeur en ppm.

Commencez par corriger la valeur de **ATMOCO2** puis branchez votre montage pour laisser le temps au capteur de chauffer (vous avez le temps de boire des cafés, d'aller manger ou de continuer à parcourir l'ensemble des projets disponible sur fablab66.fr) Vous remarquerez qu'au bout d'un certain temps (environ 1h dans mon cas) les valeurs changent grandement. Mes valeurs sont passées d'environ 400 à 4000, j'ai supposé que mon capteur avait atteint son rythme de croisière. Vous pouvez à présent relever la valeur de **Get R0 Corrected: XX.XX** dans la console. Modifiez alors le 73.89 présent dans le code original par celui-ci puis re-téléversez le code dans votre ESP.

Votre capteur est calibré.

ArduinoBlocks :: Console série



Baudrate: 9600

Connecter

Déconnecter

Nettoyer



Envier

```
577
Get ppm :637.86 Get ppmcorrected: 454.46 Facteur :0.88
Get R0 :63.27 Get R0 Corrected: 71.51
581
Get ppm :637.86 Get ppmcorrected: 456.10 Facteur :0.89
Get R0 :63.27 Get R0 Corrected: 71.41
578
Get ppm :637.86 Get ppmcorrected: 456.10 Facteur :0.89
Get R0 :63.27 Get R0 Corrected: 71.41
574
Get ppm :637.86 Get ppmcorrected: 448.29 Facteur :0.88
Get R0 :63.27 Get R0 Corrected: 71.86
578
Get ppm :637.86 Get ppmcorrected: 449.62 Facteur :0.88
Get R0 :63.27 Get R0 Corrected: 71.78
579
Get ppm :637.86 Get ppmcorrected: 456.10 Facteur :0.89
Get R0 :63.27 Get R0 Corrected: 71.41
577
Get ppm :637.86 Get ppmcorrected: 456.36 Facteur :0.89
Get R0 :63.27 Get R0 Corrected: 71.40
```

Initialiser

> Démarrer Bauds 9600

Fixer ATMOCO2 = 415

Fixer CorA = 0.00035

Fixer CorB = 0.02718

Fixer CorC = 1.39538

Fixer CorD = 0.0018

Fixer RLOAD = 10

Fixer RZERO = 73.89

Fixer ParA = 116.6020682

Fixer ParB = 2.769034857

B Blynk Démarrer

WiFi SSID NomDuReseauWifi

WiFi mot de passe MotDePasseDuWifi

Serveur IP 185 . 133 . 208 . 47 Port 8080

Code (Auth) TokenDonnéParBlynk

Recent trend

Full Record

Growth Rate

Data

Recent Global CO₂ Trend

April 28: 414.59 ppm

April 27: 414.58 ppm

April 26: 414.58 ppm

April 25: 414.57 ppm

April 24: 414.56 ppm

Last Updated: April 29, 2021

ÉTAPE 5

L'appli Blynk de suivi

Commencez par télécharger l'application Blynk sur votre appareil mobile puis suivez la vidéo suivante si vous n'avez jamais créé une telle appli :

I

Pour que votre **Gauge** affiche la valeur en CO2 corrigée dites lui d'afficher la pin virtuelle **V0**.

Pour la température la pin **V2** et pour l'humidité la pin **V3**.

Si vous souhaitez effectuer un suivi continu et un tracé de la courbe d'évolution, il vous faudra insérer un **SuperChart** et en cliquant sur celui-ci vous choisirez **+ Add Datastream** . Une nouvelle ligne **Name** apparaît ainsi que trois petits curseurs sa droite. Cliquez dessus pour pouvoir choisir la pin virtuelle que vous souhaitez suivre ainsi que le style de votre courbe, le nombre de décimales, la couleur ...

Vous pouvez également créer directement un clone de mon appli en cliquant sur le logo **QR-code** lors de la création d'un New Project et en scannant celui fourni ci-dessus.



INSTANTANÉ

SUIVI

TEMPÉRATURE

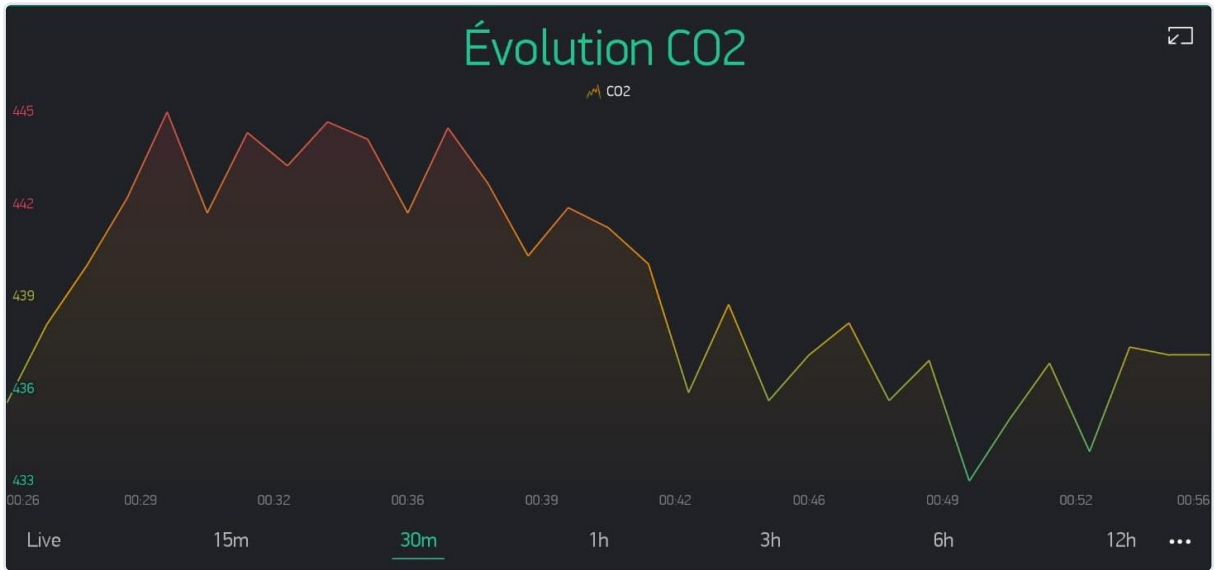


HUMIDITÉ



TAUX CO2

437 ppm





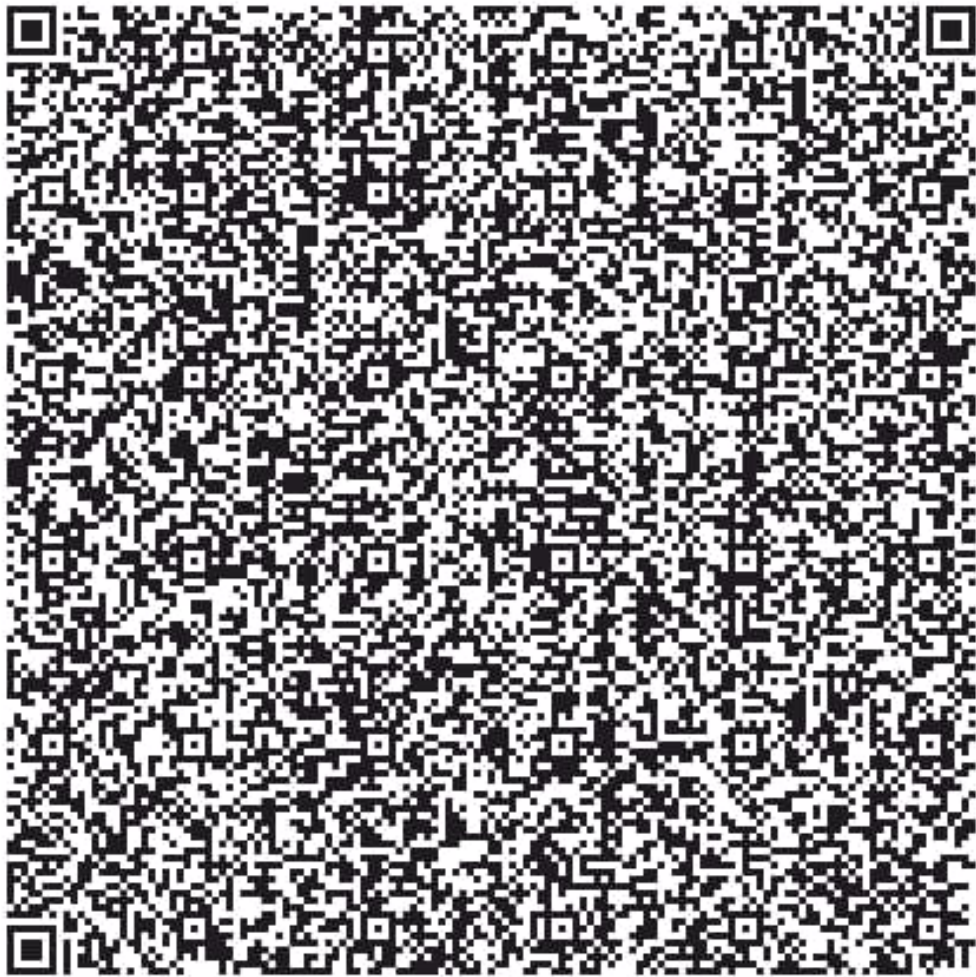
Blynk



New Project



Community



ÉTAPE 6

Le boîtier à imprimer

A venir ...

ÉTAPE 7

Sitographie

<https://projetsdiy.fr/mq135-mesure-qualite-air-polluant-arduino/>

<https://projetsdiy.fr/capteurs-mq-detecter-gaz-polluants-fumees/>

<https://github.com/GeorgK/MQ135/>