

# Un capteur de CO<sub>2</sub>



## 1. Le contexte

Dans le cadre des mesures sanitaires pour limiter la transmission du SARS-CoV-2, il est conseillé de s'assurer de la bonne aération des locaux. On peut pour cela mesurer le taux de CO<sub>2</sub> qui en est un bon indicateur (on considère qu'une concentration en CO<sub>2</sub> supérieure à 800 ppm est le signe d'une aération insuffisante).

On se propose ici de développer un algorithme et un programme en Python permettant de gérer les mesures données par un capteur de CO<sub>2</sub> de type NDIR (principe de l'absorption dans l'infrarouge non dispersif).



Figure 1 : Capteur  
Sensirion scd30

## 2. Démarche de projet.

Cette activité a été conçue comme un projet filé qui accompagnera la progression au long de l'année de première. Il utilise une carte programmable en MicroPython comme support mais on peut tout à fait travailler avec les élèves sans carte en mimant sur PC les interactions de la carte avec le capteur et leds par de simples affichages dans la console Python. En pratique un seul capteur peut-être construit par le professeur pour illustrer et motiver le projet.

Le mini-projet comporte trois étapes dont la dernière pourra être envisagée en terminale :

- Etape 1 : un devoir maison traité individuellement.

Cette étape proposée en début d'année a pour objectif de consolider l'acquisition des structures algorithmiques de base et la maîtrise de l'environnement Python : boucle de scrutation, affichage selon des seuils définis (structures conditionnelles).

- Etape 2 : travail en binôme. On perfectionne le dispositif en ajoutant un actionneur (Buzzer) et en rendant le code modulaire : on réinvestit ainsi le travail réalisé sur les fonctions.
- Etape 3 : on pourrait prolonger ce mini-projet en classe de terminale en proposant une publication sur une plateforme IOT type ThingSpeak...

## 3. Le matériel

### 1.1 Choix de la carte de programmation :

L'activité a été réalisée sur une carte esp32 Lolin livrée avec un firmware MicroPython :  
<https://projetsdiy.fr/nouvelle-wemos-d32-remplacante-esp32-lolin32-lolin32-lite-pro/>

Si vous utilisez une carte esp32 de base, il faut installer un firmware MicroPython. Un tutoriel pour le réaliser ici :

<https://www.aranacorp.com/fr/programmez-un-esp32-avec-micropython/>

### 1.2 Choix du capteur :

On utilise un capteur de CO2 scd30 de sensirion :

<https://www.sensirion.com/en/environmental-sensors/carbon-dioxide-sensors/carbon-dioxide-sensors-scd30/>



Il peut être intéressant de choisir une version au format grove pour faciliter les connexions.

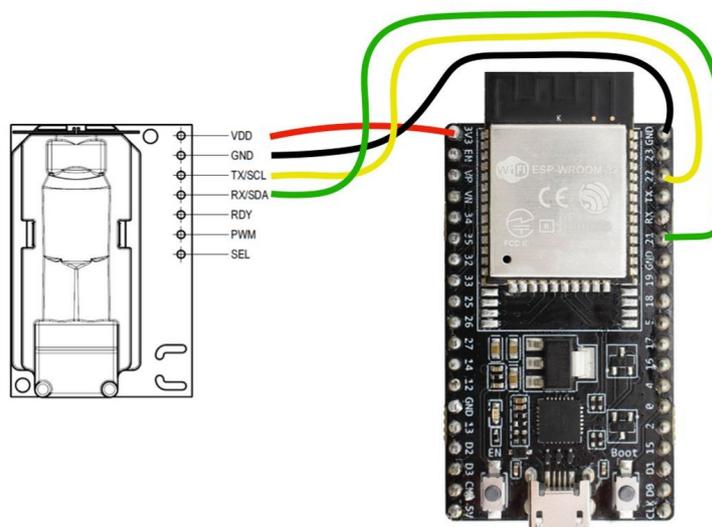
[https://wiki.seeedstudio.com/Grove-CO2\\_Temperature\\_Humidity\\_Sensor-SCD30/](https://wiki.seeedstudio.com/Grove-CO2_Temperature_Humidity_Sensor-SCD30/)



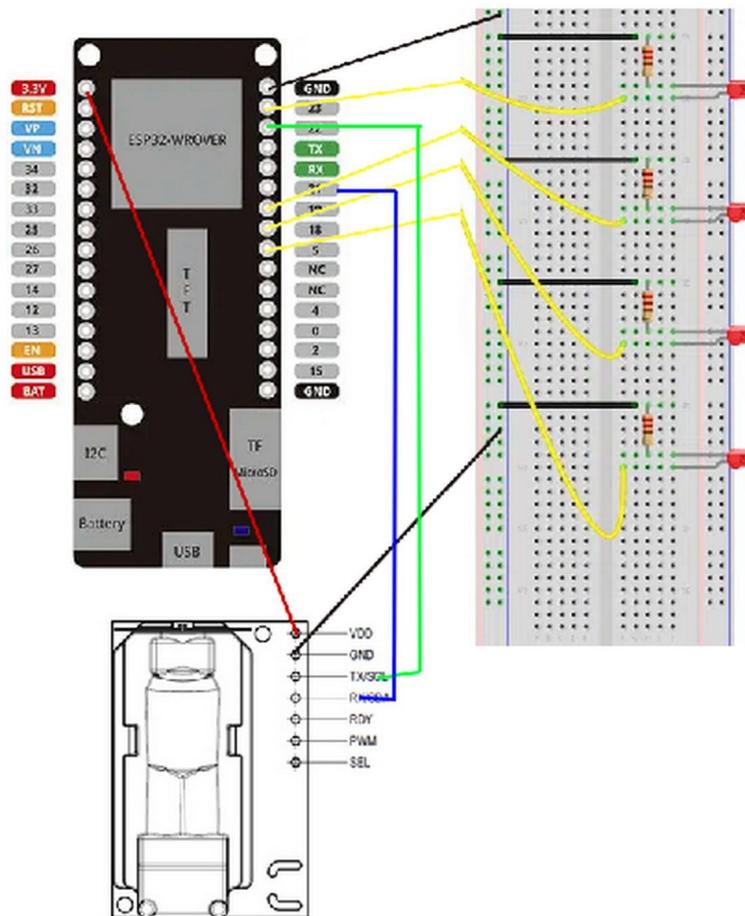
### 1.3 Le câblage :

Les leds (vertes, Jaune, Orange, Rouge) ont été respectivement connectées aux broches 23, 18, 19 et 5 (avec en série des résistances de 220  $\Omega$ , on peut optimiser ceci ...)

Le capteur Sensirion est connecté suivant ce schéma :



Ce qui donne le câblage global suivant :



Pour simplifier ce câblage, on peut utiliser un feu tricolore à 3 leds (on adapte alors les seuils...)



Remarque :

- Pour les deux premières parties, les consignes sont fournies aux élèves, les programmes python 3 qui miment (simulent) le capteur et les leds, les programmes en micro-Python pour l'esp32, des barèmes pour évaluer le travail réalisé.
- Ne pas oublier d'implanter la bibliothèque du capteur scd 30 sur l'esp32 ...
- Pour le Buzzer, on a choisi un « tout ou rien » qui sera branché sur le pin 13 (KY-012)

