
*PROJET
EXPERIMENTAL
ET NUMERIQUE*

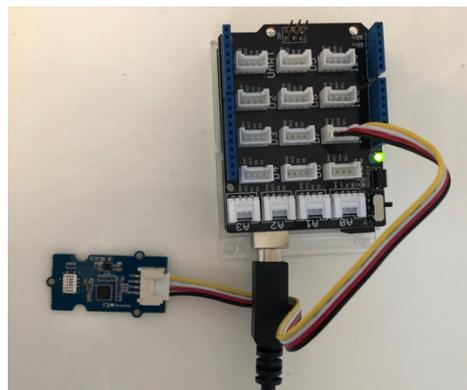
Qualité de l'air

Ce tutoriel permet de créer un dispositif répondant aux trois objectifs du projet expérimental et numérique :

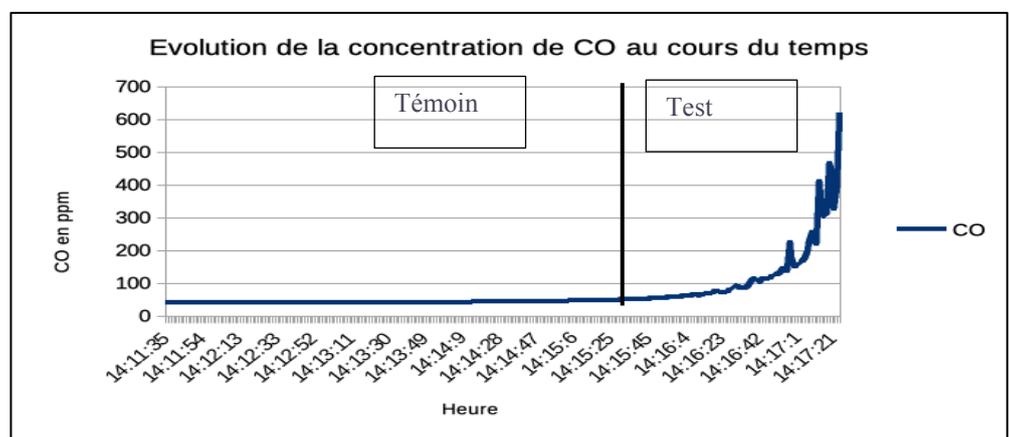
- utilisation d'un capteur éventuellement réalisé en classe ;
- acquisition numérique de données ;
- traitement mathématique, représentation et interprétation de ces données.

Ce dispositif permet de mesurer la **concentration de monoxyde de carbone (CO) dans l'atmosphère en ppm**.

Le CO est un polluant atmosphérique libéré lors des combustions incomplètes (gaz, charbon, fioul ou bois).



Photographies du dispositif



Résultats obtenus





Carte Arduino Uno



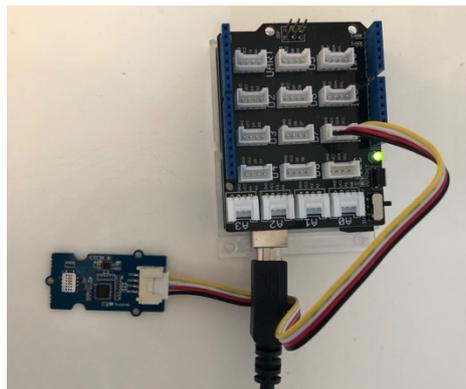
Base Shield Grove



Capteur MultiGaz

Après avoir associé le shield Grove à la carte Arduino Uno, le capteur Multigaz se connecte sur une broche I2C. Respectez l'emplacement.

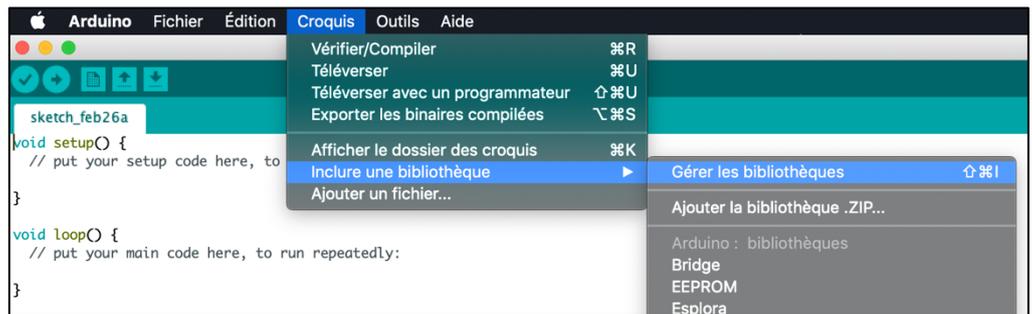
(Temps de chauffage du capteur 10 minutes)



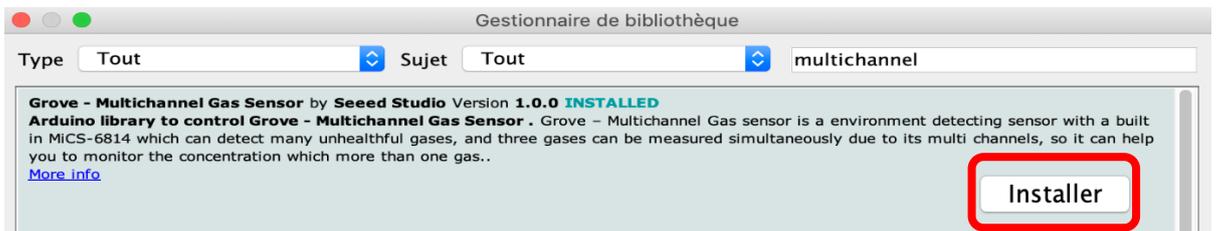
Photographie du montage

Pour faire fonctionner le capteur MultiGaz, vous devez enregistrer deux bibliothèques : **Wire (include)** et **MultichannelGasSensor**.

Pour enregistrer une bibliothèque, allez dans **Croquis > Inclure une bibliothèque > Gérer les bibliothèques**.



Tapez dans la barre de recherche le nom de la bibliothèque souhaitée puis installez-la.



```
CO_simplifie | Arduino 1.8.13
CO_simplifie
/*
  This is a demo to test gas library
  This code is running on Xadow-mainboard, and the I2C slave is Xadow-gas
  There is a ATmega168PA on Xadow-gas, it get sensors output and feed back to master.
  the data is raw ADC value, algorithm should be realized on master.

  please feel free to write email to me if there is any question

  Jacky Zhang, Embedded Software Engineer
  ql.zhang@seeed.cc
  17_mar,2015
*/

#include <Wire.h>
#include "MutichannelGasSensor.h"

void setup() {
  Serial.begin(115200); // start serial for output
  Serial.println("power on!");
  gas.begin(0x04); //the default I2C address of the slave is 0x04
  gas.powerOn();
  Serial.print("Firmware Version = ");
  Serial.println(gas.getVersion());
}

void loop() {
  float c;

  c = gas.measure_CO();
  Serial.print("The concentration of CO is ");
  if (c >= 0) {
    Serial.print(c);
  } else {
    Serial.print("invalid");
  }
  Serial.println(" ppm");

  delay(5000);
}

Enregistrement terminé.
Bibliothèque non valide trouvée dans /Users/cedricmetge/Documents/Arduino/libraries/SDS011_vers : aucun fichier d'en-tête (.h) trouvé dans /U
Bibliothèque non valide trouvée dans /Users/cedricmetge/Documents/Arduino/libraries/DHT_sensor_library : aucun fichier d'en-tête (.h) trouvé
Bibliothèque non valide trouvée dans /Users/cedricmetge/Documents/Arduino/libraries/Adafruit_SGP30_Sensor : aucun fichier d'en-tête (.h) trou
41 Arduino Uno sur /dev/cu.usbmodem14201
```

- Téléchargez le code en entier sur le lien suivant : https://padlet.com/bizik64/code_arduino_capteurs
- Ouvrez le code avec l'IDE d'Arduino,
- Vérifiez puis téléversez le code (attention au port utilisé : **Outils > Port > COM Arduino**).

Récupération des données pour le traitement mathématique

Enseignement
Scientifique
Première
Basé sur Arduino

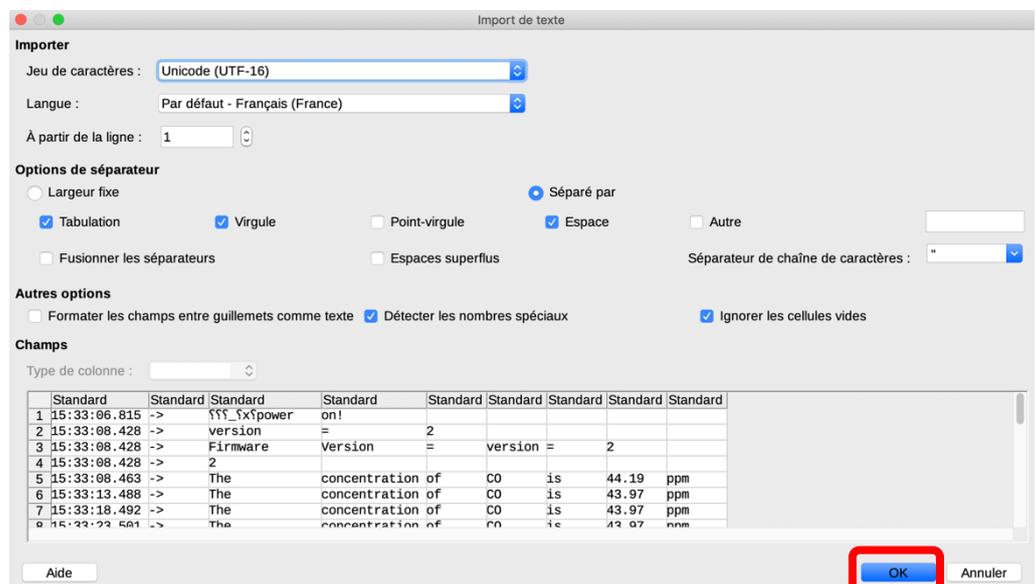
- Ouvrez le moniteur série (**Outils > Moniteur série**),
- Sélectionnez la vitesse de communication baud en accord avec le code : ici **115200 baud**,
- Cochez la case **Afficher l'horodatage**.



- Réalisez vos expériences en suivant les résultats sur le moniteur série,
- Sélectionnez tous vos résultats dans le moniteur série (**CTRL+A**),
- Copiez votre sélection (**CTRL+C**),



- Ouvrir un fichier **LibreCalc**,
- Copiez votre sélection dans la première cellule (**CTRL+V**),
- Validez l'importation des données.

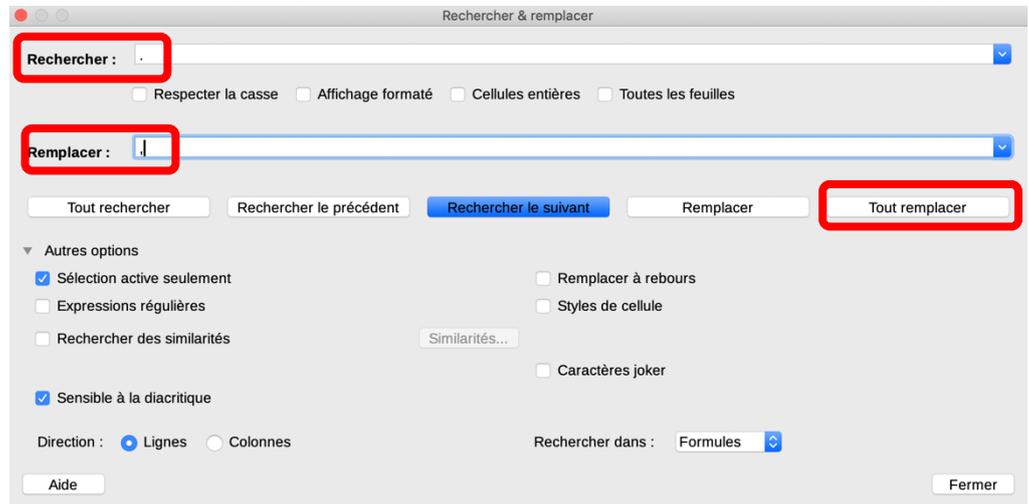


Enseignement
Scientifique
Première

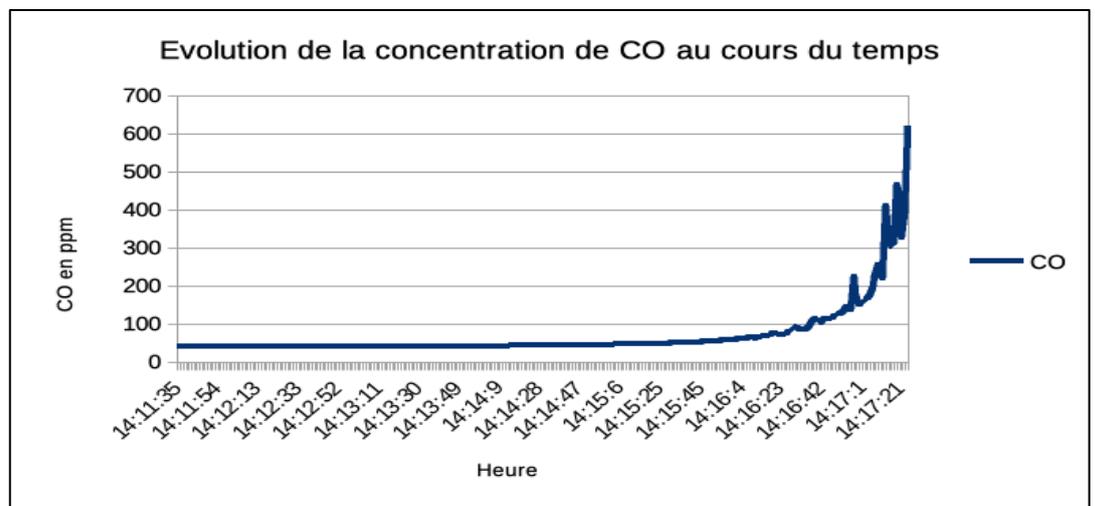
Basé sur Arduino

- Traitez maintenant ce fichier,
- Remplacer les points par des virgules : Edition > Rechercher et remplacer,

Temps	CO
15:33:08,46	44,19
15:33:13,49	43,97
15:33:18,49	43,97
15:33:23,50	43,97
15:33:28,54	43,97
15:33:33,53	43,97
15:33:38,54	43,97
15:33:43,59	44,19
15:33:48,60	43,97
15:33:53,62	43,97
15:33:58,62	43,97
15:34:03,63	43,97
15:34:08,65	43,97
15:34:13,66	43,97
15:34:18,69	43,97
15:34:23,70	43,75
15:34:28,73	43,97
15:34:33,75	43,75
15:34:38,76	43,75
15:34:43,77	43,53
15:34:48,77	43,75
15:34:53,82	43,75
15:34:58,81	43,75
15:35:03,83	43,75
15:35:08,85	43,53
15:35:13,87	43,53
15:35:18,89	43,53
15:35:23,90	43,53
15:35:28,92	43,53
15:35:33,92	43,53
15:35:38,96	43,53
15:35:43,95	43,75
15:35:48,96	43,53
15:35:54,01	43,53
15:35:59,02	43,53
15:36:04,01	43,31
15:36:09,05	43,53
15:36:14,05	43,53
15:36:23,05	43,53
15:36:26,09	43,31
15:36:29,63	43,53



- Construisez le graphique à partir des données.



Résultats obtenus