

---

*PROJET  
EXPERIMENTAL  
ET NUMERIQUE*

*Qualité de l'air*

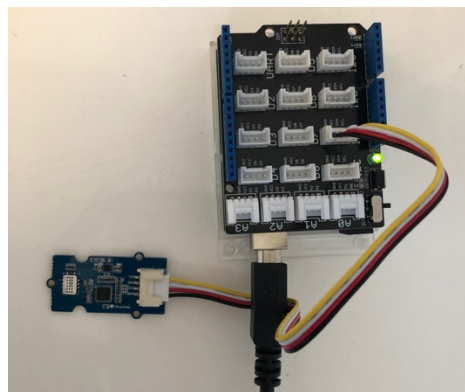
---

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

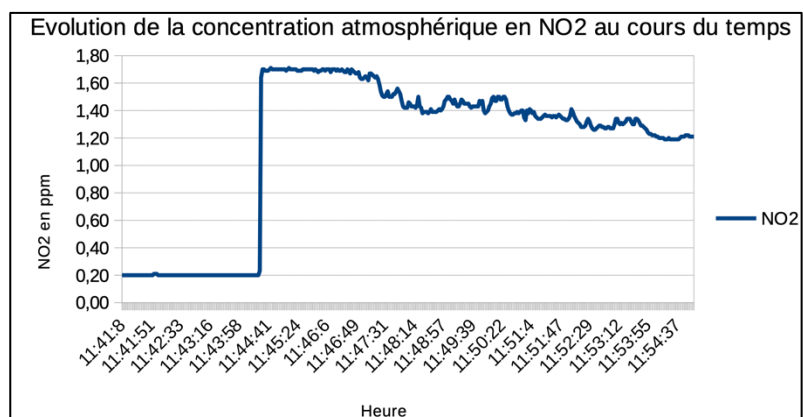
- utilisation d'un capteur éventuellement réalisé en classe ;
- acquisition numérique de données ;
- traitement mathématique, représentation et interprétation de ces données.

Ce dispositif permet de mesurer la **concentration de dioxyde d'azote (NO<sub>2</sub>) dans l'atmosphère en ppm**.

Le NO<sub>2</sub> est un polluant atmosphérique libéré lors de la combustion (chauffage, production d'électricité, moteurs thermiques des véhicules...).



Photographies du dispositif



Résultats obtenus





Carte Arduino Uno



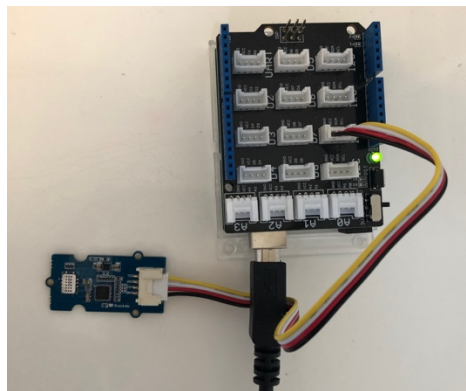
Base Shield Grove



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.

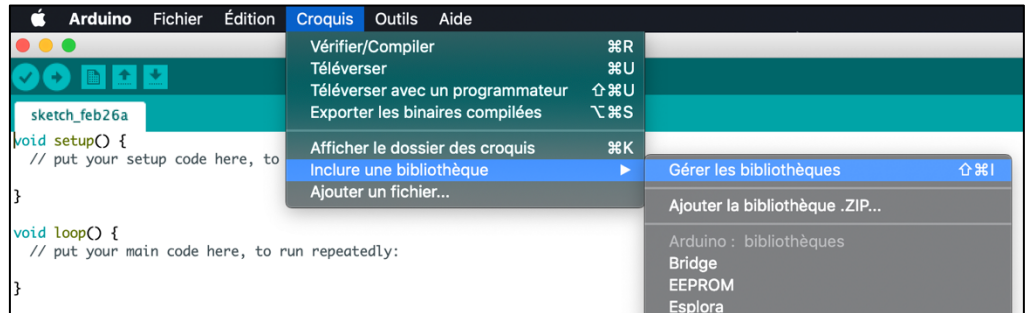
(Temps de chauffage du capteur 10 minutes)



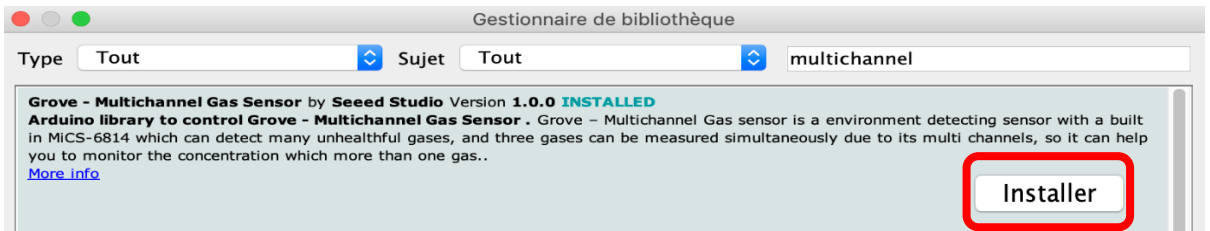
Photographie du montage

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.





```
NO2_simplifie | Arduino 1.8.13
NO2_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
  qi.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_NO2();
  Serial.print("The concentration of NO2 is ");
  if (c >= 0) {
    Serial.print(c);
  } else {
    Serial.print("invalid");
  }
  Serial.println(" ppm");

  delay(1000);
}

Téléversement terminé
Le croquis utilise 8150 octets (25%) de l'espace de stockage de programmes. Le maximum est de 32256 octets.
Les variables globales utilisent 554 octets (27%) de mémoire dynamique, ce qui laisse 1494 octets pour les variables locales. Le maximum est
Bibliothèque non valide trouvée dans /Users/cedricmetae/Documents/Arduino/libraries/esp_sds011 : aucun fichier d'en-tête (.h) trouvé dans /Us
41 Arduino Uno sur /dev/cu.usbmodem14201
```

- Téléchargez le code en entier sur le lien suivant :  
[https://padlet.com/bizik64/code\\_arduino\\_capteurs](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

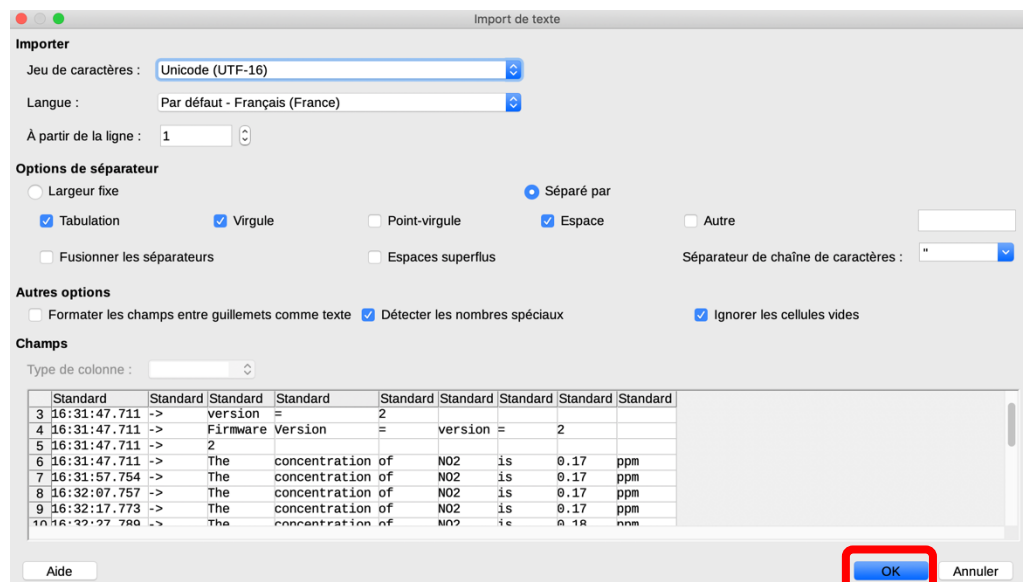
- Ouvrez le moniteur série (**Outils > Moniteur série**),
- Sélectionnez la vitesse de communication baud en accord avec le code : ici **115200 baud**,
- Cochez la case **Afficher l'horodatage**.



- 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.

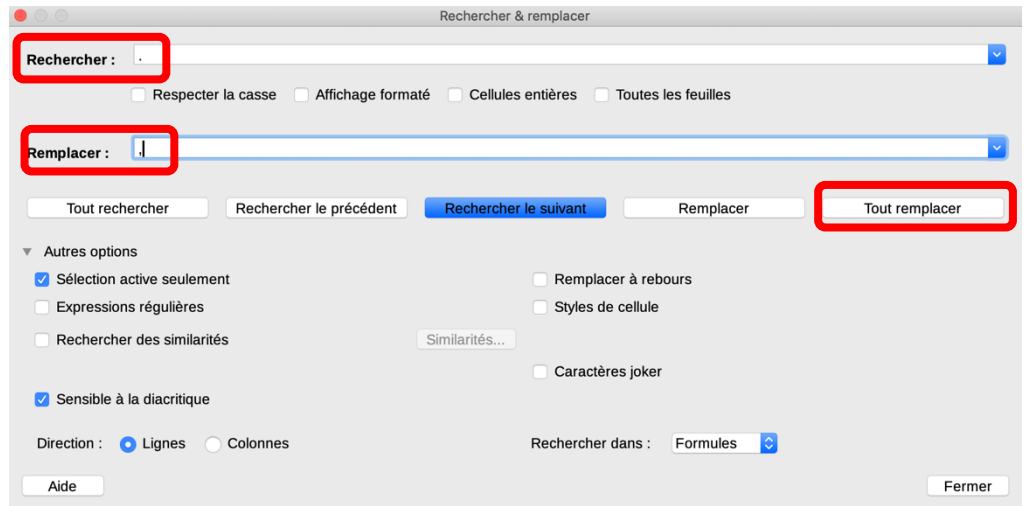


Enseignement  
Scientifique  
Première

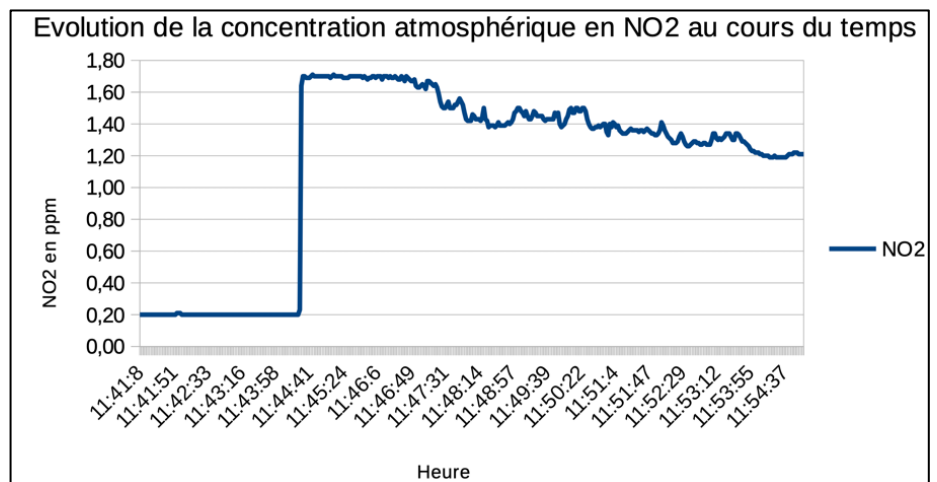
Basé sur Arduino

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

Heure	NO2
16:31:47,71	0,17
16:31:57,75	0,17
16:32:07,76	0,17
16:32:17,77	0,17
16:32:27,79	0,18
16:32:37,80	0,18
16:32:47,82	0,18
16:32:57,84	0,18
16:33:07,84	0,18
16:33:17,87	0,18
16:33:27,89	0,18
16:33:37,90	0,18
16:33:47,91	0,18
16:33:57,92	0,18
16:34:07,92	0,18
16:34:17,95	0,18
16:34:27,98	0,18
16:34:37,97	0,18
16:34:47,99	0,18
16:34:58,02	0,18
16:35:08,04	0,18
16:35:18,02	0,18
16:35:28,05	0,18
16:35:38,05	0,18
16:35:48,07	0,18
16:35:58,09	0,18
16:36:08,09	0,18
16:36:18,11	0,18
16:36:28,14	0,18
16:36:38,15	0,18
16:36:48,18	0,18
16:36:58,16	0,18
16:37:08,20	0,18
16:37:18,22	0,18
16:37:28,21	0,18
16:37:38,23	0,18
16:37:48,27	0,18
16:37:58,28	0,18



- Construisez le graphique à partir des données.



Résultats obtenus