

Qualité de l'air : dispositif de mesures de particules fines PM10

Enseignement
Scientifique
Première

Basé sur Arduino

PROJET
EXPERIMENTAL
ET NUMERIQUE

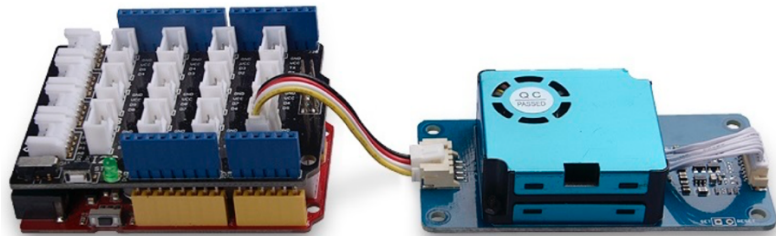
Qualité de l'air

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

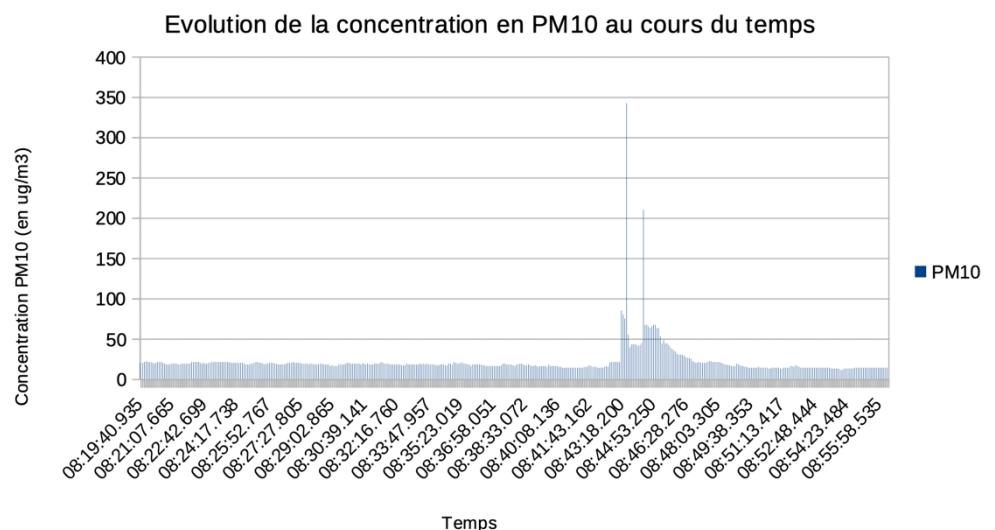
- utilisation d'un capteur éventuellement réalisé en classe ;
- acquisition numérique de données ;
- traitement mathématique, représentation et interprétation de ces données.

Ce dispositif permet de **mesurer la concentration de particules fines de diamètre inférieures à 10 micromètres (PM10) dans l'atmosphère en ppm.**

Les PM10 sont d'origine naturelle ou anthropique (feux de cheminées, fumée de cigarettes, combustion industrielle...)



Photographies du dispositif



Résultats obtenus

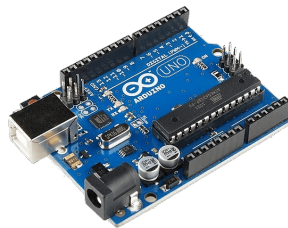


METGE Cédric

Lycée Berthelot, Toulouse
Cedric.metge@ac-toulouse.fr

Composants et montage

Enseignement
Scientifique
Première
Basé sur Arduino



Carte Arduino Uno



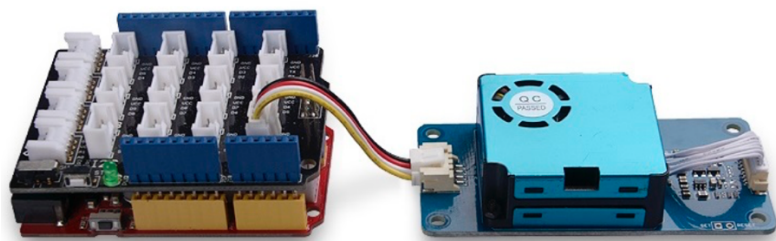
Base Shield Grove



Grove Laser PM Sensor
HM3301

Après avoir associé le shield Grove à la carte Arduino Uno, le capteur HM3301 se connecte sur une broche I2C. Le capteur doit être positionné verticalement.

(Temps de préchauffage du capteur 30 secondes)



Photographie du montage

Enseignement

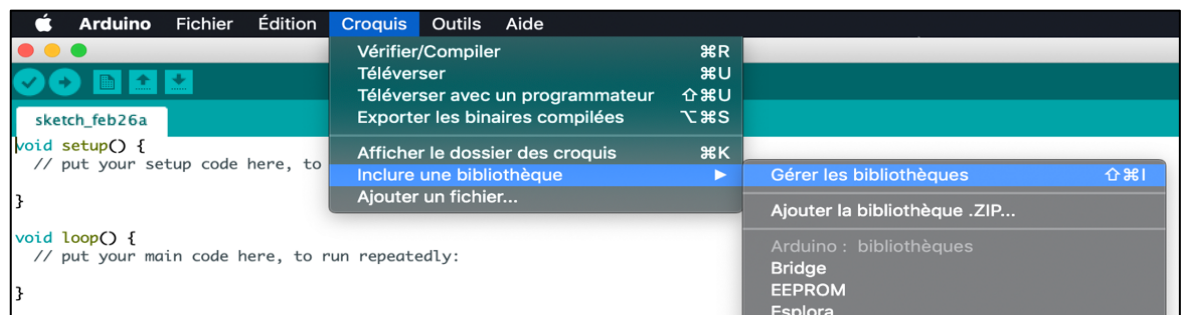
Scientifique

Première

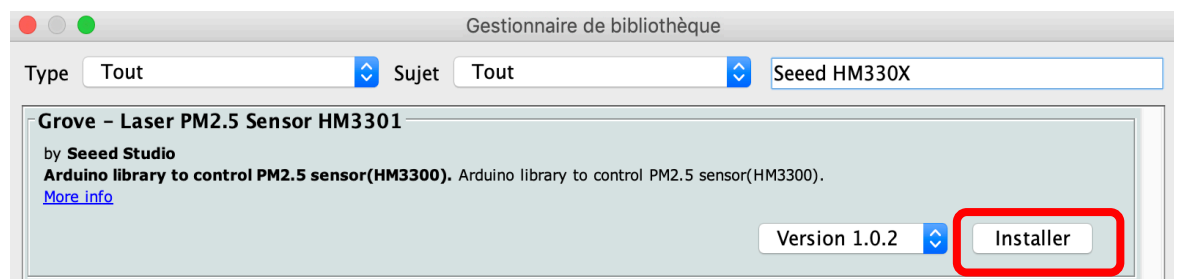
Basé sur Arduino

Pour faire fonctionner le capteur HM3301, vous devez enregistrer une bibliothèque : **Seed_HM330X**.

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



Pour installer la bibliothèque souhaitée, tapez dans la barre de recherche son nom.



Enseignement Scientifique Première Basé sur Arduino

```
PM10 | Arduino 1.8.13
basic_demo.ino
Example for Seeed PM2.5 Sensor(HM300)

Copyright (c) 2018 Seeed Technology Co., Ltd.
Website : www.seeed.cc
Author : downey
Create Time: August 2018
Change Log :

The MIT License (MIT)

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

*/

#include <Seeed_HM330X.h>

#ifdef ARDUINO_SAMD_VARIANT_COMPLIANCE
#define SERIAL_OUTPUT SerialUSB
#else
#define SERIAL_OUTPUT Serial
#endif

HM330X sensor;
uint8_t buf[30];

const char* str[] = {"sensor num: ", "PM1.0 concentration(CF=1,Standard particulate matter,unit:ug/m3): ",
                    "PM2.5 concentration(CF=1,Standard particulate matter,unit:ug/m3): ",
                    "PM10 concentration(CF=1,Standard particulate matter,unit:ug/m3): ",
                    "PM1.0 concentration(Atmospheric environment,unit:ug/m3): ",
                    "PM2.5 concentration(Atmospheric environment,unit:ug/m3): ",
                    "PM10 concentration(Atmospheric environment,unit:ug/m3): ",
                    };

HM330XErrorCode print_result(const char* str, uint16_t value) {
    if (NULL == str) {
        return ERROR_PARAM;
    }
    SERIAL_OUTPUT.print(str);
    SERIAL_OUTPUT.println(value);
    return NO_ERROR;
}

/*parse buf with 29 uint8_t-data*/
HM330XErrorCode parse_result(uint8_t* data,int i) {
    uint16_t value = 0;
    if (NULL == data) {
        return ERROR_PARAM;
    }
    value = (uint16_t) data[i * 2] << 8 | data[i * 2 + 1];
    print_result(str[i - 1], value);
}
```

- Téléchargez le code en entier sur le lien suivant : https://padlet.com/bizik64/code_arduino_capteurs
- Copiez-collez le code dans 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

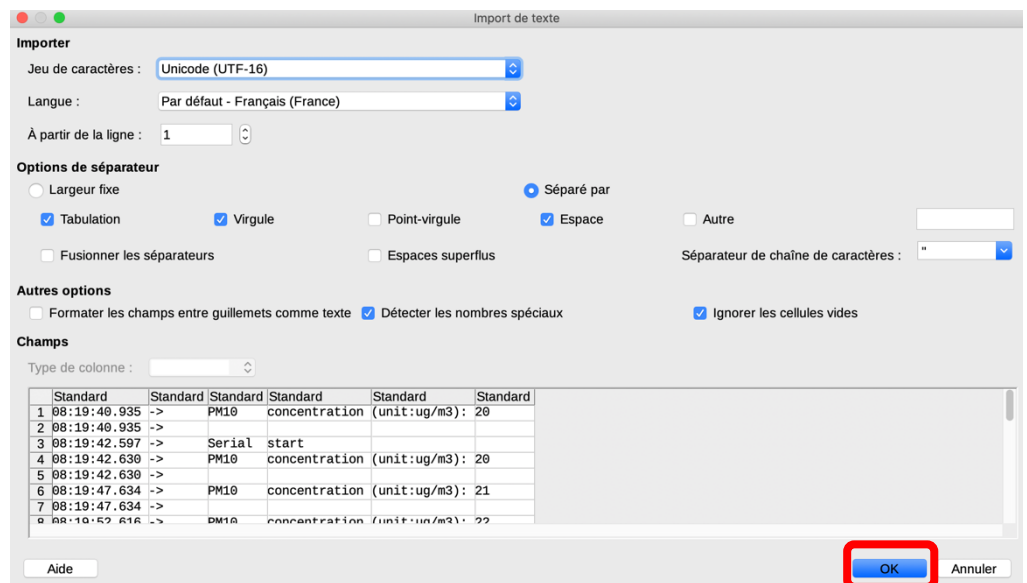
- Ouvrez le moniteur série (**Outils > Moniteur série**),
- Sélectionnez la vitesse de communication baud en accord avec le code : ici **115200 baud**,
- Cochez la case **Afficher l'horodatage**.



- 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.

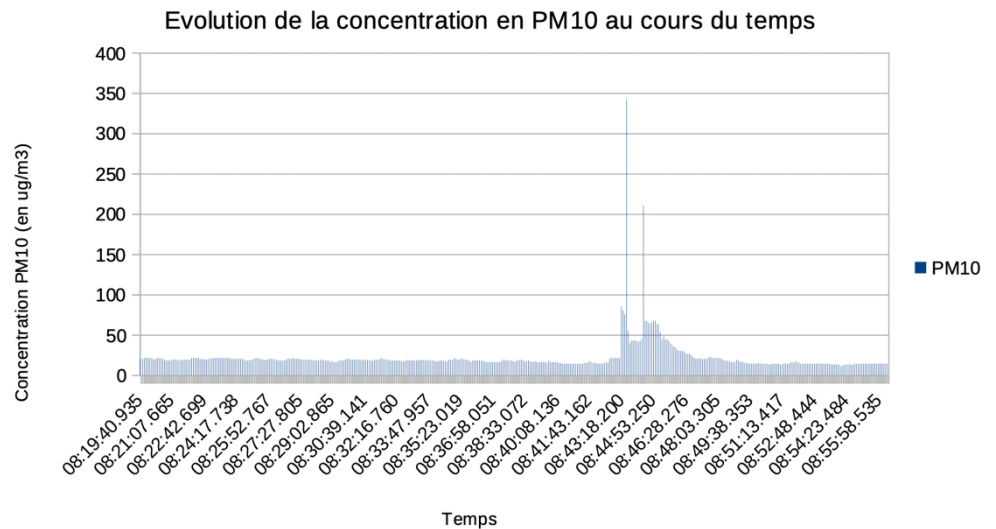


Enseignement
Scientifique
Première

Basé sur Arduino

A	B
Heure	PM10
08:19:40.935	20
08:19:40.935	
08:19:42.597	
08:19:42.630	20
08:19:42.630	
08:19:47.634	21
08:19:47.634	
08:19:52.616	22
08:19:52.616	
08:19:57.634	21
08:19:57.634	
08:20:02.612	21
08:20:02.650	
08:20:07.642	21
08:20:07.642	
08:20:12.631	20
08:20:12.631	
08:20:17.623	19
08:20:17.623	
08:20:22.640	20
08:20:22.640	
08:20:27.644	21
08:20:27.644	
08:20:32.656	21
08:20:32.656	
08:20:37.634	21
08:20:37.634	
08:20:42.635	20
08:20:42.635	
08:20:47.635	19
08:20:47.635	
08:20:52.641	18
08:20:52.641	
08:20:57.640	18
08:20:57.640	
08:21:02.635	18
08:21:02.669	
08:21:07.665	19

- **Traitez** maintenant vos données,
- **Construisez** le graphique à partir des données.



Résultats obtenus