

Rapport de projet : Capteur de Qualité de l'air

ARE 2016



Équipe

Oda BØ, coordinatrice
Dorine CHAKHTOURA, construction physique
Léo LAVILLUNIÈRE, modélisation du support
Victor PERRIN-TURENNE, modélisation électronique
Einat ALTMAN, programmation

Introduction

Si l'on entend souvent parler de réchauffement climatique et de pollution, c'est parce que ce sont des sujets d'inquiétude qui deviennent de plus en plus importants tous les jours. Selon *Le Monde*, la pollution de l'air tue plus de 2 millions de personnes par an en Asie et est aujourd'hui une des dix premières causes de décès dans le monde. Comparé aux niveaux de pollution hallucinants dans beaucoup de villes Asiatiques, la qualité de l'air en France reste relativement faible. Mais les pics de pollutions à Paris (dernier pic le 20 janvier 2016) et dans d'autres grandes villes, peuvent être un vrai danger pour la santé publique. En d'autres termes, la pollution est un sujet important et très actuel, raisons pour lesquelles la création d'un capteur de qualité d'air nous a inspiré. Ainsi, nous envisageons de créer un capteur de gaz qui peut être fixé sur le bord d'une fenêtre pour y mesurer la qualité de l'air au quotidien.

Matériel utilisé

Nous cherchons à créer un capteur transportable et donc assez solide mais petit, et ludique. De plus, étant novices dans le domaine, nous nous sommes dirigés vers quelque chose de simple à programmer. Nous utilisons donc :

- Une [carte Arduino Uno](#), facile à programmer et peu chère.



- Un [Grove Base Shield](#), que l'on couple à la carte Arduino, afin d'avoir des ports Grove et i2c, plus faciles à brancher et plus solides.



- Un [écran LCD i2c backlight](#), facile à brancher et à programmer, il comporte de plus un rétro-éclairage dont on peut changer la couleur, option dont on va se servir pour avertir l'utilisateur lorsque les concentrations dépassent les seuils d'alerte.



- Un [capteur de CO₂](#) en Grove.



- Un [capteur multigaz](#) en Grove, dont on se servira pour capter le CO, le NO2 et le CH4.



- Un [capteur de particules](#) en Grove.



- Un [capteur de température](#) en Grove, afin d'avoir une information supplémentaire étant donné que la température peut avoir une influence sur les concentrations de gaz.



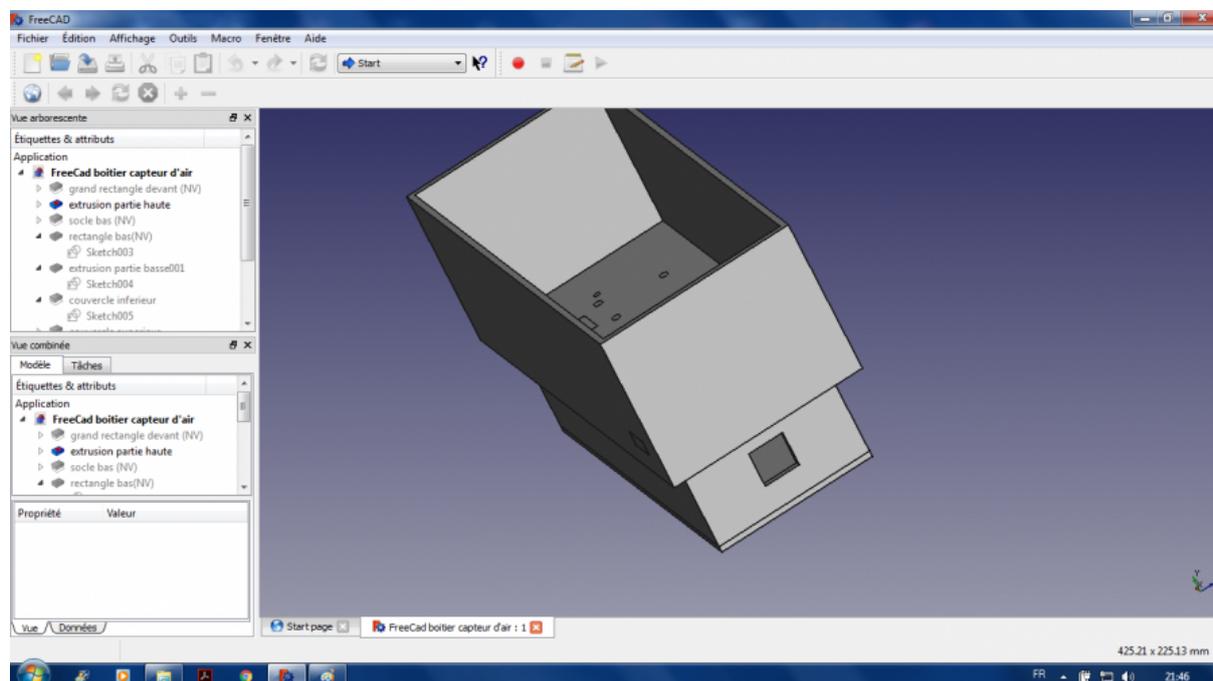
- Un ventilateur pour faire circuler l'air à l'intérieur de la boîte contenant les capteurs, branché directement sur l'alimentation avant la carte en dénudant et soudant les fils.



Chaque élément utilisé fonctionne en 5V, et nous branchons la carte Arduino sur une alimentation secteur. De plus, nous utiliserons des planches de bois, découpées au laser, pour fabriquer la boîte.

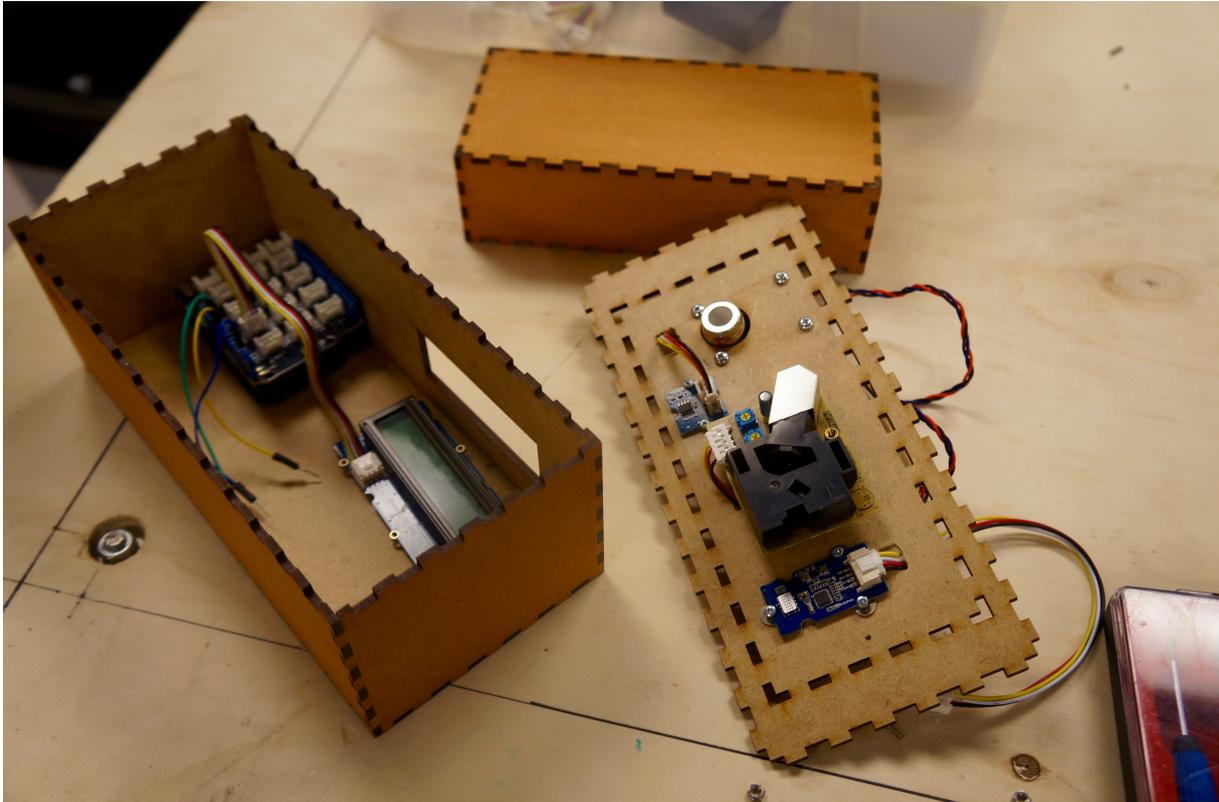
Modélisation de la boîte

Pour la modélisation nous utilisons le logiciel FreeCAD afin d'avoir également une vue d'ensemble en 3D de la boîte.



Nous voulons une boîte petite et transportable. De plus, elle doit être compartimentée : en effet nous voulons séparer les capteurs de la carte Arduino pour qu'une éventuelle chaleur libérée par cette dernière n'influence pas les résultats. Cependant, il est assez difficile de créer une compartimentation avec des planches découpées au laser que l'on va emboîter, c'est pourquoi nous allons plutôt créer une boîte pour l'Arduino et une autre plus petite pour les capteurs qui s'encastre dans la première.

Les capteurs seront fixés à l'aide de vis et d'écrous à la paroi séparant les deux boîtes, et des trous seront aménagés dans les parois pour faire passer les fils, ainsi que pour encastrer le ventilateur et l'écran. De plus, nous aménageons une grille à l'opposé du ventilateur pour l'entrée d'air.



Programme

```
/*
*****
CAPTEUR DE QUALITÉ DE L'AIR ---- ARE 2016
*****
#include <Wire.h>
#include "rgb_lcd.h"
#include <MutichannelGasSensor.h>

#define read_sample_interval (50) //Combien d'echantillons gaz
#define read_sample_times (5) // intervalle entre prise echantillons gaz (ms)

#define minCO2 (0)
#define maxCO2 (1800)
#define bool_pin_CO2 (2)
#define zero_point_voltage_CO2 (0.220) // output capteur quand concentration co2=400PPM
#define reaction_voltage_CO2 (0.02) //voltage drop quand capteur : air -> 1000PPM
#define dc_gain (8.5) //

#define minPart (0)
#define maxPart (25)
```

```

#define minTemp (-40)
#define maxTemp (125)

#define minCO (0)
#define maxCO (229)

#define minNO2 (0)
#define maxNO2 (200)

#define minCH4 (0)
#define maxCH4 (3531)

//max defini par seuil alerte OMS

rgb_lcd lcd;
// variables couleur fond d'ecran
int colorR;
int colorG;
int colorB;

double valCO;
double valNO2;
double valCH4;
double valCO2;
double valTemp;
double valPart;

float cCO;
float cCH4;
float cNO2;

float prctCO;
float prctCH4;
float prctNO2;

float ugPart; // particules en ug - apres conversion depuis prcts/cf
float prctPart;

// cr = capteur
const int crCO2 = 3 ;
const int crPart = 7;
const int crTemp = 2; //entree analog. A2

// le multigaz et ecran sont en i2c et reconnus automatiquement

float CO2Curve[3] = {2.602,zero_point_voltage_CO2,(reaction_voltage_CO2/(2.602-3))};
// courbe approximative CO2 a partir 2 points : log400 et log4000

//*****TEMP*****
float vTemp; //analogRead(crTemp)
int intervalTemp = 1000; //intervalle de temps entre lecture temp
float cTemp; //temp en celsius
int BTemp=3975 ; //necessaire conversion temp en celsius
float resistance;

```

```

//*****PART*****
unsigned long duration;
unsigned long starttime;
unsigned long lowpulseoccupancy = 0; //???
float ratio = 0;
float concentration = 0;

void setup() {

//*****BLOC ECRAN*****
lcd.begin(16, 2);

colorR = 0;
colorG = 0;
colorB = 100;

lcd.setRGB(colorR, colorG, colorB);

lcd.print("Initisation");
lcd.setCursor(0, 1);
lcd.print("en cours");
delay(5000);
lcd.clear ();

//*****fin BLOC ECRAN*****

Serial.begin(9600);

//*****BLOC CO2*****
pinMode(bool_pin_CO2, INPUT);
digitalWrite(bool_pin_CO2, HIGH);

//*****fin BLOC CO2*****

//*****BLOC PART*****
pinMode (crPart, INPUT);
starttime = millis();

//*****fin BLOC PART*****

//*****BLOC MULTIGAZ*****
mutichannelGasSensor.begin(0x04);//the default I2C address of the slave is 0x04

mutichannelGasSensor.powerOn();

//*****fin BLOC MULTIGAZ*****

}
void loop() {

/*****

BLOC CO2

*****/
float prectCO2;

```

```
float pourcentage;
float volts;

volts = MGRead(crCO2);

pourcentage = MGGetPercentage(volts,CO2Curve);

prctCO2 = (pourcentage*1.94*100)/maxCO2;
valCO2 = prctCO2
```

```
/******
```

```
FIN BLOC CO2
```

```
*****/
```

```
/******
```

```
BLOC CO
```

```
*****/
cCO = mutichannelGasSensor.measure_CO();
prctCO = (cCO*100)/maxCO;
valCO = prctCO
```

```
/******
```

```
FIN BLOC CO
```

```
*****/
```

```
/******
```

```
BLOC CH4
```

```
*****/
cCH4 = mutichannelGasSensor.measure_CH4();
prctCH4 = (cCH4*100)/maxCH4;
```

```
valCH4 = prctCH4
```

```
/******
```

```
FIN BLOC CH4
```

```
*****/
```

```
/******
```

```
BLOC NO2
```

```
*****/
cNO2 = mutichannelGasSensor.measure_NO2();
prctNO2 = (cNO2*100)/maxNO2;
```

```

valNO2 = prctNO2

/*****

FIN BLOC NO2

*****/

/*****

BLOC PART

*****/
duration = pulseIn (crPart, LOW);
lowpulseoccupancy = lowpulseoccupancy+duration;

if ((millis()-starttime) > read_sample_interval)
{
    ratio = lowpulseoccupancy/(read_sample_interval*10.0);
    concentration = 1.1*pow(ratio,3)-3.8*pow(ratio,2)+520*ratio+0.62; // d'apres datasheet
    lowpulseoccupancy = 0;
    starttime = millis();
}

ugPart = concentration*3571/500;

prctPart = (ugPart*100)/maxPart;

valPart = prctPart

/*****

FIN BLOC PART

*****/

/*****

BLOC TEMP

*****/

vTemp=analogRead(crTemp);
resistance=(float)(1023-vTemp)*10000/vTemp;
cTemp=1/(log(resistance/10000)/BTemp+1/298.15)-273.15;

delay(intervalTemp);

valTemp = cTemp;

/*****

FIN BLOC TEMP

*****/

```

```
/******
```

BLOC AFFICHAGE

```
*****/
```

```
if (valCO2 >= 100)
{colorR = 50;
colorG = 0;
colorB = 0;
}else{
colorR = 0;
colorG = 50;
colorB = 0;
}
lcd.setRGB(colorR, colorG, colorB);
```

```
lcd.setCursor(0,0);
lcd.print ("CO2 :");
lcd.print (valCO2);
lcd.print ("%");
```

```
delay(1500);
lcd.clear ();
```

```
if (valCO >= 100)
{colorR = 50;
colorG = 0;
colorB = 0;
}else{
colorR = 0;
colorG = 50;
colorB = 0;
}
lcd.setRGB(colorR, colorG, colorB);
```

```
lcd.setCursor(0,0);
lcd.print ("CO :");
lcd.print (valCO);
lcd.print ("%");
```

```
delay(1500);
lcd.clear ();
```

```
if (valNO2 >= 100)
{colorR = 50;
colorG = 0;
colorB = 0;
}else{
colorR = 0;
colorG = 50;
colorB = 0;
}
lcd.setRGB(colorR, colorG, colorB);
```

```
lcd.setCursor(0,0);
lcd.print ("NO2 :");
lcd.print (valNO2);
lcd.print ("%");
```

```
delay(1500);
lcd.clear ();
```

```

if (valCH4 >= 100)
{colorR = 50;
colorG = 0;
colorB = 0;
}else{
colorR = 0;
colorG = 50;
colorB = 0;
}
lcd.setRGB(colorR, colorG, colorB);
lcd.setCursor(0,0);
lcd.print ("CH4 :");
lcd.print (valCH4);
lcd.print ("%");

delay(1500);
lcd.clear ();

if (valPart >= 100)
{colorR = 50;
colorG = 0;
colorB = 0;
}else{
colorR = 0;
colorG = 50;
colorB = 0;
}
lcd.setRGB(colorR, colorG, colorB);
lcd.setCursor(0,0);
lcd.print ("Particules :");
lcd.setCursor(0,1);
lcd.print (valPart);
lcd.print ("%");

delay(1500);
lcd.clear ();

if (valTemp >= 40)
{colorR = 50;
colorG = 0;
colorB = 0;
}else{
colorR = 0;
colorG = 50;
colorB = 0;
}

lcd.setRGB(colorR, colorG, colorB);
lcd.setCursor(0,0);
lcd.print ("Temperature :");
lcd.setCursor(0,1);
lcd.print (valTemp);
lcd.print ("C");

delay(1500);
lcd.clear ();

/*****

```

FIN BLOC AFFICHAGE

```
*****  
} //fin du loop  
  
float MGRead(int crCO2) {  
  int i;  
  float v=0;  
  
  for(i=0;i<read_sample_times;i++){  
    v+=analogRead(crCO2);  
    delay(read_sample_interval);  
  }  
  v = (v/read_sample_times)*5/1024;  
  return v;  
}  
  
float MGGetPercentage(float volts, float *pcurve){  
  if(((volts/dc_gain)>=zero_point_voltage_CO2){  
    return -1;  
  } else {  
    return pow(10, ((volts/dc_gain)-pcurve[1])/pcurve[2]+pcurve[0]);  
  }  
}  
}
```

Produit final et difficultés rencontrées

Notre produit final fonctionne et affiché des valeurs qui sont tout à fait cohérentes avec les valeurs théoriques des différents gaz dans l'air. Cependant, nous avons quelques valeurs incohérentes que nous n'avons pas eu le temps de corriger.

En effet, le manque de temps est une des difficultés que nous avons rencontrées lors de ce projet. Notre matériel est arrivé beaucoup plus tard que ce que l'on avait prévu, et de plus nous avons eu une séance de retard au début de l'année. Ce retard a été rattrapé à la fin du projet, ce qui nous a permis de finir notre projet, mais malgré cela, notre capteur a toujours quelques défauts.

De plus, nous avons eu un problème par rapport à notre carte Arduino, qui finalement n'est pas celle que l'on avait commandée au début. Cela nous a créé des difficultés, surtout au niveau de l'alimentation que nous avons dû changer.

Conclusion

Nous avons obtenu un capteur de gaz fonctionnel, avec un affichage fluide et ludique, dans une boîte transportable. Cependant, nous n'avons pas eu le temps d'effectuer l'étalonnage de nos capteurs en environnement contrôlé, et nous obtenons donc parfois des valeurs incorrectes.