

Projet Centrale De Mesure De Pollution Intérieure

Par Maxime Pouleyn, Léo Berteloot, Romain Guilbert et Achraf Dabach

Sommaires

Centrale De Mesure De Pollution Intérieur

- Présentation du Projet Général
 - Ma partie
 - Problèmes rencontrés
 - Conclusion
-

● Présentation du Projet Général

Commande de la CUD pour le service d'éducation au développement durable



Objectifs de mesure :

Concentration en particules fines

Concentration en COV (Composés Volatiles)

Concentration en CO₂

Température

Humidité

Avec GPS pour l'installation de plusieurs centrales sur un bâtiment

● Présentation du Projet Général



OMS 50.000 morts/ans soit 10 fois plus que les accidents de la route

● Présentation du Projet Général

The diagram illustrates the relationship between different types of pollutants and their health impacts. On the left, four categories of pollutants are listed in colored boxes: "Allergènes et matières particulières" (orange), "Produits chimiques et odeurs" (red), "Bactéries et virus" (green), and "Véhicules et polluants industriels" (light blue). On the right, the "Effet sur la santé" (Health Effects) are detailed under two timeframes: "Court terme" (Short term) and "Long terme" (Long term). The "Court terme" effects include reactions like allergic rhinitis, eye infections, nose and throat irritation, bronchitis, pneumonia, headaches, nausea, respiratory difficulties, skin reactions (eczema), and asthma attacks. The "Long terme" effects include chronic respiratory diseases, lung cancer, cardiovascular diseases, brain and nerve damage, and internal organ damage (liver and kidneys).

Allergènes et matières particulières

Produits chimiques et odeurs

Bactéries et virus

Véhicules et polluants industriels

Effet sur la santé

Court terme

- Réactions allergiques
- Infections oculaires (conjonctivite)
- Irritation du nez et de la gorge
- Bronchite
- Pneumonie
- Maux de tête et nausées
- Difficultés respiratoires
- Réactions cutanées (eczéma)
- Crises d'asthme

Long terme

- Maladies respiratoires chroniques
- Cancer des poumons
- Maladies cardio-vasculaires
- Atteinte du cerveau et des nerfs
- Atteinte des organes internes (foie et reins)

● Présentation du Projet Général

Réaliser une centrale de mesure de pollution intérieure mesurant les paramètres suivants :

Les locaux professionnels, en particulier les bâtiments industriels, recèlent deux catégories de polluants de l'air intérieur :

- **les polluants gazeux** : odeurs, vapeurs de produits chimiques, composés organiques volatils (COV), etc. ;
- **les polluants sous forme de particules** : poussières, aérosols, substances biologiques (virus, bactéries, spores), etc.

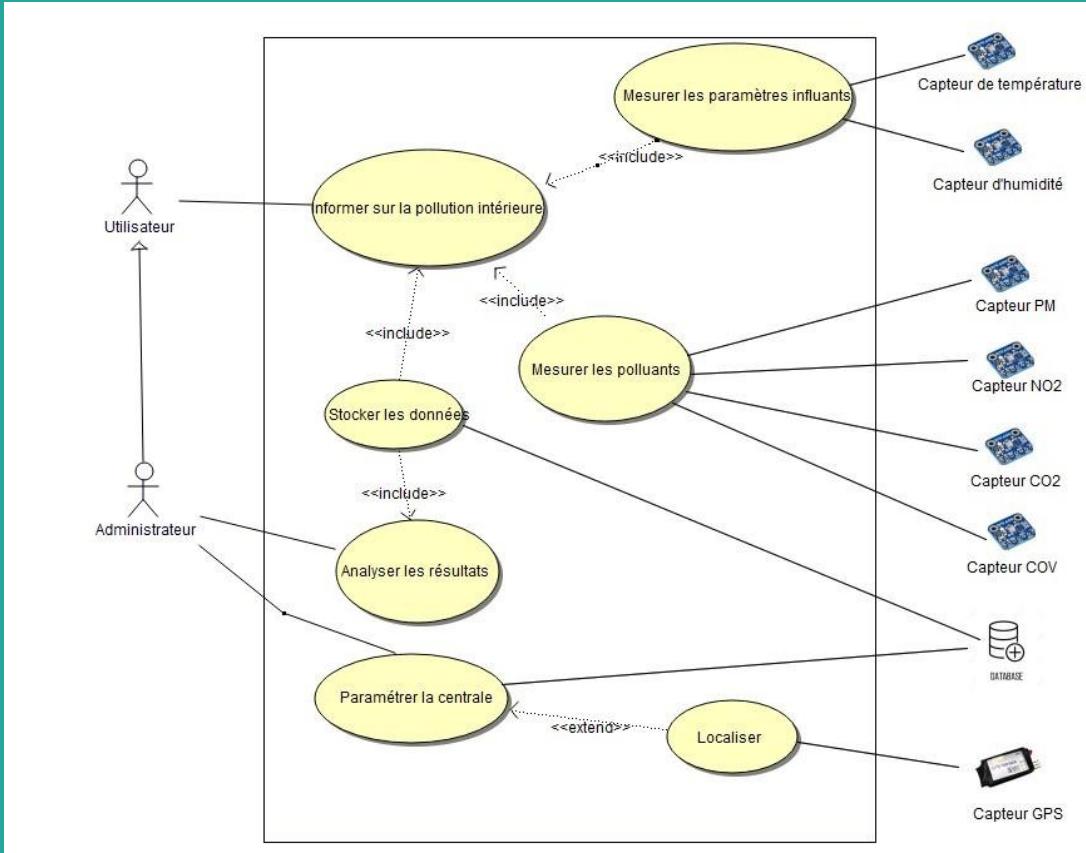
- Concentration en particules fines
- Concentration COV (composés volatiles)
- Concentration CO₂ (dioxyde de carbone)
- Mesure de température
- Taux d'humidité
- GPS

● Répartition des tâches

Taches à faire	Maxime	Romain	Achraf	Léo
Capteur de température	x			
Capteur d'humidité		x		
Capteur de particule		x		
Capteur de COV			x	
Capteur de CO2			x	
Capteur GPS	x			
Interface Administrateur	x			
Interface par défaut		x		
Interface Historique			x	
Liaison Arduino-Raspberry	x	x	x	x
Base de donnée	x	x	x	x
NodeJS				x

● Analyse UML :

Diagramme
de cas
d'utilisation:



● Ma partie :

Description	Quantité	Prix
Raspberry Pi 3B+	1	75€
Arduino uno	1	20€
Capteur SGP 30 (COV eCO2)	1	17.50€
Câbles	1	3.20€
	Total	115.7€

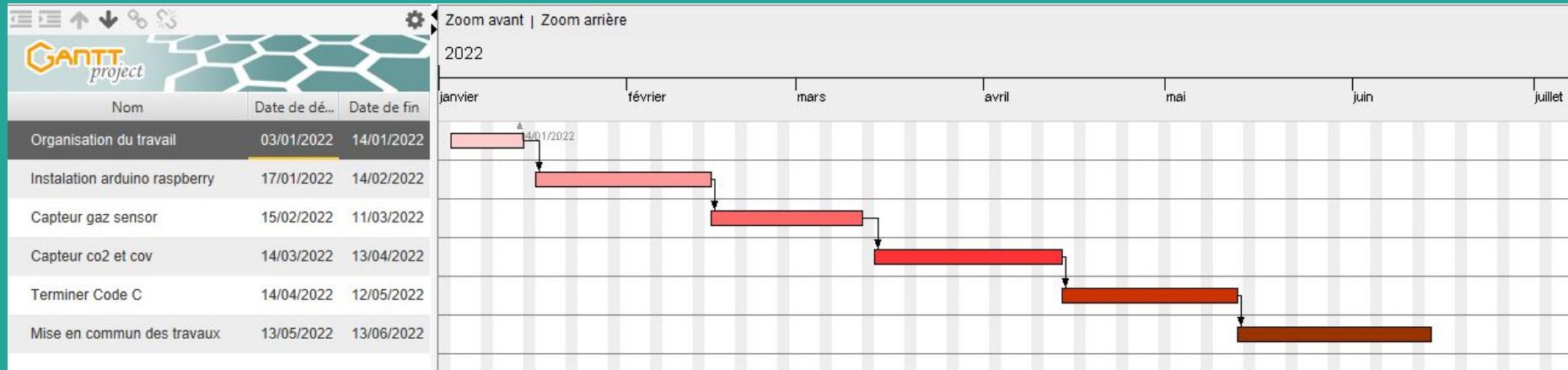
Qu'est ce que j'ai fait
depuis le début?

Les différentes tâches que j'ai à réaliser :

● Ma partie :

- Permettre la communication entre l'arduino et le raspberry
- Récupérer les valeurs du capteur COV (composés volatiles) et les envoyées au raspberry et les stockées
- Récupérer les valeurs du capteur CO2 (dioxyde de carbone) et les envoyées au raspberry et les stockées
- Envoyer toutes les données dans un fichier Json

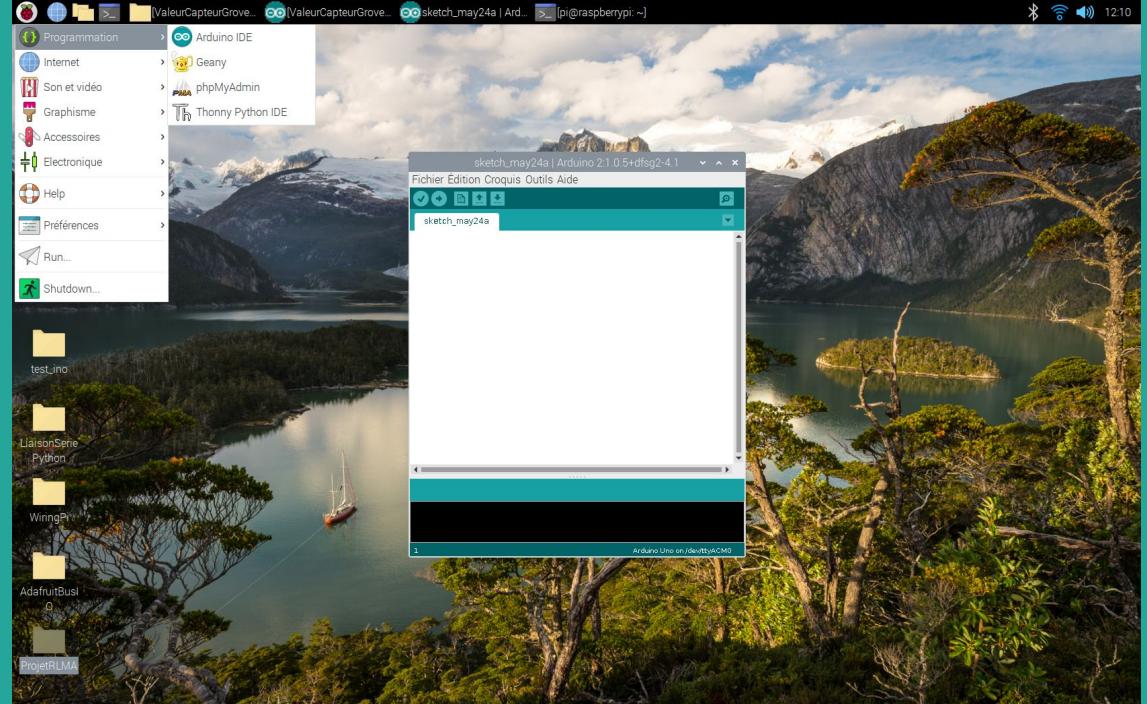
● Ma partie :



● Ma partie :



Branchement de la raspberry



Installation arduino

- Ma partie :

Grove - Gas Sensor(MQ5)



● Ma partie :



```
ValeurCapteurGroveGasMQ5_COV

void setup () {
    Serial.begin(9600);
}

void loop(){
    float sensor_volt;
    float sensorValue;
    const int MonCapteur = 0;

    sensorValue = analogRead(MonCapteur);
    sensor_volt = sensorValue/1023*5.0;

    Serial.print("cov=");
    Serial.print(sensorValue);
    Serial.print(";");
    // Serial.print(" sensor_volt = ");
    // Serial.print(sensor_volt);
    //Serial.println(" V");
    delay (3000);

}
```

● Ma partie :

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4 #include <errno.h>
5 #include <wiringPi.h>
6 #include <wiringSerial.h>
7
8 int main()
9 {
10     FILE *f;
11     int temperature=0;
12     int humidite=0;
13     int pm=0;
14     int no2=0;
15     float cov=0.0;
16     int fd;
17     char c[15];
18 //char type[10];
19     float valeur=0.0;
20     int i;
21     int co2=0;
22     float latitude=0.0;
23     float longitude=0.0;
24
25
26
27     fd=serialOpen("/dev/ttyACM0",9600);
28
29     if(fd<0)
30     {
31         fprintf(stderr,"Unable to open serial device: %s\n",strerror(errno));
32         return 1;
33     }
34
35     for (++)
36     {
37         i=0;
38         do
39         {
40             c[i++]= serialGetchar(fd);
41         }while(c[i-1]!=';');
42         c[i-1]=0;
43         if(strncmp(c,"cov=",4)==0)
44         {
45             valeur = atof(&c[4]);
46             printf("Valeur : %f\n",valeur);
47             cov=valeur;
48         }
49         f = fopen("test.json", "w+");
50
51         if(f==NULL){
```

```
Com.c ✘ EcritureFichierJson.c ✘ test.json ✘ data.json ✘ accueil.php ✘ receivelastData.php ✘ index.php ✘ functions.php ✘
48     }
49     f = fopen("test.json", "w+");
50
51     if(f==NULL){
52         printf("Erreur lors de l'ouverture d'un fichier");
53         //exit(1);
54     }
55     else
56     {
57
58         fprintf(f, "\n\"temperature\": %d,\n\"humidite\": %d,\n\"pm\": %d,\n\"no2\": %d,\n\"cov\": %f,\n\"co2\": %d,\n\"gps\": { \n\"latitude\": %f,\n\"longitude\": %f }\n");
59         fprintf(f, "\nframe : %s\n",c);
60         fflush(stdout);
61     }
62
63
64
65     //float gps;
66
67
68
69
70     printf("temperature : ");
71     scanf("%d", &temperature);
72     printf("humidite : ");
73     scanf("%d", &humidite);
74     printf("pm : ");
75     scanf("%d", &pm);
76     printf("no2 : ");
77     scanf("%d", &no2);
78     printf("cov : ");
79     scanf("%d", &cov);
80     printf("co2 : ");
81     scanf("%d", &co2);
82     printf("latitude : ");
83     scanf("%f", &latitude);
84     printf("longitude : ");
85     scanf("%f", &longitude);
86
87     fprintf(f, "\n\"temperature\": %d,\n\"humidite\": %d,\n\"pm\": %d,\n\"no2\": %d,\n\"cov\": %f,\n\"co2\": %d,\n\"gps\": { \n\"latitude\": %f,\n\"longitude\": %f }\n",temperature,humidite,pm,no2,cov,co2,latitude,longitude);
88     fclose(f);
89 }
```

● Ma partie :



test.json - /home/pi/... [pi@raspberrypi: ~]

test.json - /home/pi/Desktop/ProjetRLMA - Geany

Hier Éditer Rechercher Affichage Document Projet Construire Outils Aide

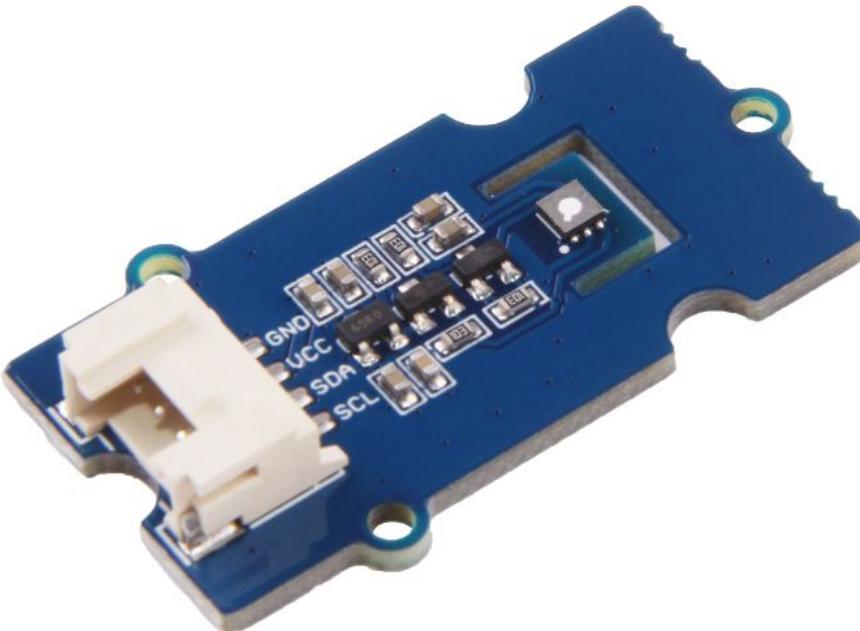
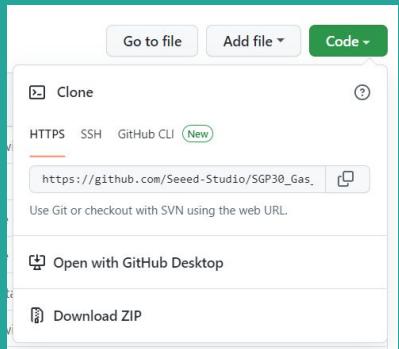
Com.c ✘ EcritureFichierJson.c ✘ test.json ✘ data.json ✘ accueil.php ✘ receivelastData.php ✘ index.php ✘ functions.php ✘

```
1  ↳ {  
2      "temperature": 0,  
3      "humidite": 0,  
4      "pm": 0,  
5      "no2": 0,  
6      "cov": 144.000000,  
7      "co2": 0,  
8      "gps": {  
9          "latitude": "0.000000",  
10         "longitude" : "0.000000"  
11     }  
12 }
```

● Ma partie :

Grove - VOC and eCO₂ Gas Sensor(SGP30)

Bibliothèque :



● Ma partie :

Paramètre	Signal	Valeur
Tension d'utilisation	VOC	0 ppb à 60000 ppb
Plage de sortie	CO2	400 ppm à 60000 ppm

- Ma partie :

CO₂:

Blanc / Très bon : Jusqu'à 800 ppm

Jaune / Bon : 800-1200 ppm

Orange / Moyen : 1200-1400 ppm

Rouge / Mauvais : 1400-2000 ppm.

5000 ppm : Valeur Limite d'Exposition Professionnelle (VLEP) établie en France par l'INRS (Institut National de la Recherche Scientifique) pour une journée de travail de 8h.

- Ma partie :

cov:

Blanc / Très bon : 0-150 ppb

Jaune / Bon : 150-300 ppb

Orange / Moyen : 300-450 ppb

Rouge / Mauvais : 450-600 ppb.

● Ma partie :

```
void loop() {
    s16 err = 0;
    u16 tvoc_ppb, co2_eq_ppm;
    err = sgp_measure_iaq_blocking_read(&tvoc_ppb, &co2_eq_ppm);
    if (err == STATUS_OK) {
        // serial.print("TVOC Concentration:");
        Serial.print(tvoc_ppb);
        Serial.print(":");
        // Serial.println(ppb, "%");
        // Serial.print("CO2eq Concentration:");
        Serial.println(co2_eq_ppm);
        // Serial.println(ppm, "%");
    } else {
        Serial.println("error reading IAQ values\n");
    }
    delay(2000);
    //store_baseline();
}
```

Téléversement terminé

Taille binaire du croquis : 7 508 octets (d'un max de 32 256 octets)

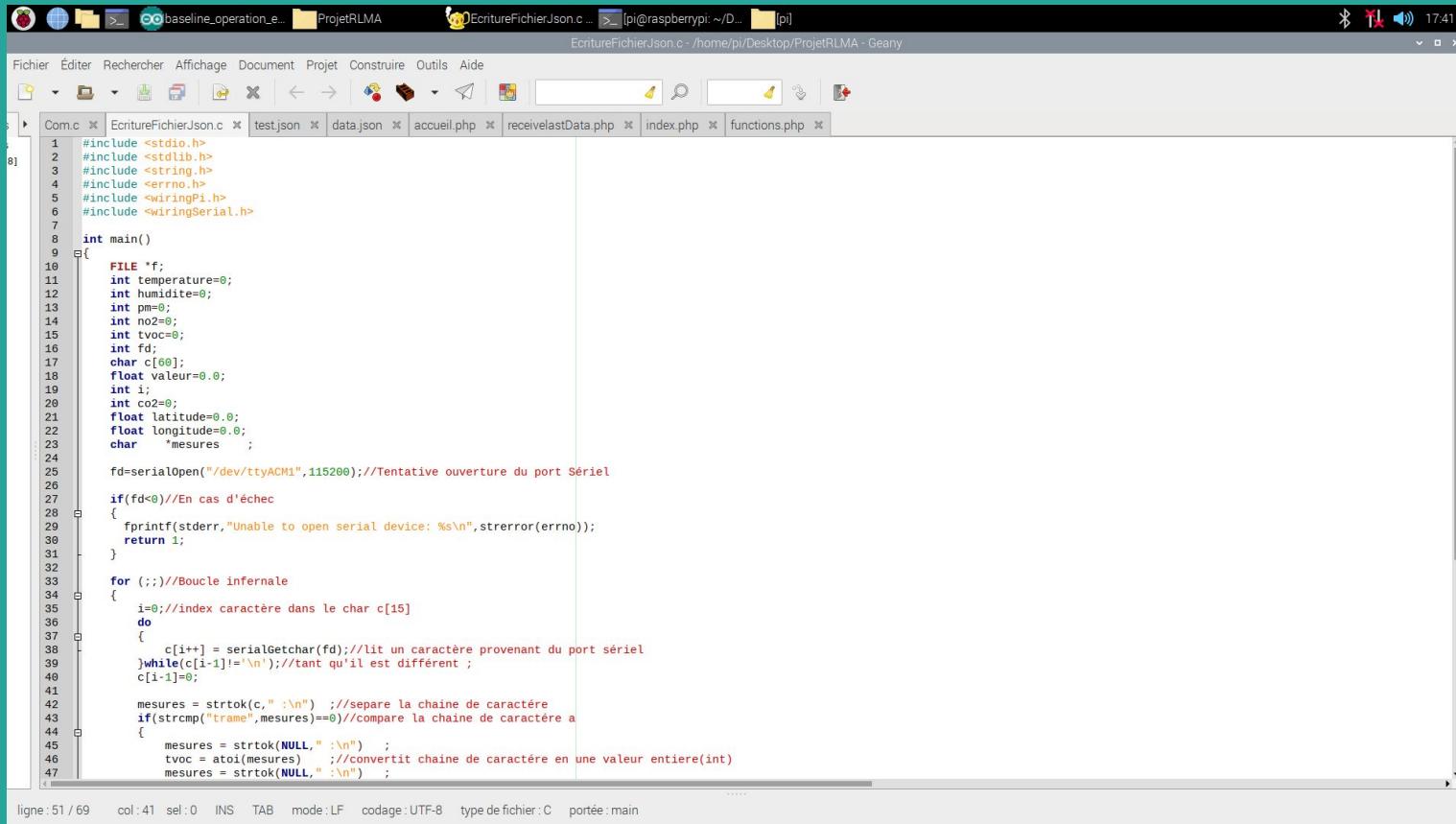
● Ma partie :

```
serial start!!
get ram signal!
FF..FF..FF..FF..FF..!!!
Warning: no baseline value in EEPROM
0:400
0:400
0:400
0:400
0:400
0:400
```

ch
()

Défilement automatique Nouvelle ligne 115200 baud

● Ma partie :



The screenshot shows a Geany IDE window with the following details:

- Title Bar:** EcritureFichierJson.c [pi@raspberrypi:~/D...]
- Menu Bar:** Fichier, Éditer, Rechercher, Affichage, Document, Projet, Construire, Outils, Aide
- Toolbar:** Includes icons for file operations like Open, Save, Print, and a search icon.
- Status Bar:** ligne : 51 / 69 col : 41 sel : 0 INS TAB mode : LF codage : UTF-8 type de fichier : C portée : main

Code Content:

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4 #include <errno.h>
5 #include <wiringPi.h>
6 #include <wiringSerial.h>
7
8 int main()
9 {
10     FILE *f;
11     int temperature=0;
12     int humidite=0;
13     int pm=0;
14     int no2=0;
15     int tvoc=0;
16     int fd;
17     char c[16];
18     float valeur=0.0;
19     int i;
20     int co2=0;
21     float latitude=0.0;
22     float longitude=0.0;
23     char    "mesures" ;
24
25     fd=serialOpen("/dev/ttyACM1",115200);//Tentative ouverture du port Sériel
26
27     if(fd<0)//En cas d'échec
28     {
29         fprintf(stderr,"Unable to open serial device: %s\n",strerror(errno));
30         return 1;
31     }
32
33     for (");//Boucle infernale
34     {
35         i=0;//index caractère dans le char c[15]
36         do
37         {
38             c[i++]= serialGetchar(fd);//lit un caractère provenant du port série
39            }while(c[i-1]!='\n');//tant qu'il est différent ;
40             c[i-1]=0;
41
42             mesures = strtok(c, " :\n") ;//separe la chaîne de caractère
43             if(strcmp("trame",mesures)==0)//compare la chaîne de caractère à
44             {
45                 mesures = strtok(NULL, " :\n") ;
46                 mesures = atoi(mesures) ;//convertit chaîne de caractère en une valeur entière(int)
47                 mesures = strtok(NULL, " :\n") ;
```

● Ma partie :



EcritureFichierJson.c [pi@raspberrypi: ~/D... [pi]

EcritureFichierJson.c - /home/pi/Desktop/ProjetRLMA - Geany

Fichier Éditer Rechercher Affichage Document Projet Construire Outils Aide

Com.c ✘ EcritureFichierJson.c ✘ test.json ✘ data.json ✘ accueil.php ✘ receivelastData.php ✘ index.php ✘ functions.php ✘

```
24
25     fd=serialOpen("/dev/ttyACM1",115200);//Tentative ouverture du port Sériel
26
27     if(fd<0)//En cas d'échec
28     {
29         fprintf(stderr,"Unable to open serial device: %s\n",strerror(errno));
30         return 1;
31     }
32
33     for (");//Boucle infernale
34     {
35         i=0;//index caractère dans le char c[15]
36         do
37         {
38             c[i++] = serialGetchar(fd);//lit un caractère provenant du port série
39             }while(c[i-1]!='\n');//tant qu'il est différent ;
40             c[i-1]=0;
41
42             mesures = strtok(c, " :\n") ;//separe la chaîne de caractère
43             if(strcmp("trame",mesures)==0)//compare la chaîne de caractère a
44             {
45                 mesures = strtok(NULL, " :\n") ;
46                 co1 = atoi(mesures) ;//convertit chaîne de caractère en une valeur entière(int)
47                 mesures = strtok(NULL, " :\n") ;
48                 co2 = atoi(mesures) ;//convertit chaîne de caractère en une valeur entière(int)
49                 printf("tvoc = %d \t co2 = %d\n",tvoc,co2);//affichage des valeurs
50
51                 f = fopen("test.json", "w+");
52
53                 if(f==NULL){
54                     printf("Erreur lors de l'ouverture d'un fichier");
55                     //exit(1);
56                 }
57                 else
58                 {
59                     fprintf(f, "{\n    \"temperature\": %d,\n    \"humidite\": %d,\n    \"pm\": %d,\n    \"no2\": %d,\n    \"tvoc\": %d,\n    \"co2\": %d,\n    \"gps\": {\n        \"latitude\": \"%f\",\n        \"longitude\": \"%f\"\n    }\n} ", temperature, humidite, pm, no2, tvoc, co2, latitude, longitude);
60                     fclose(f);
61                 }
62                 fflush(stdout);
63             }
64
65             fclose(f);
66         }
67
68     }
```

ligne : 51 / 69 col : 41 sel : 0 INS TAB mode : LF codage : UTF-8 type de fichier : C portée : main

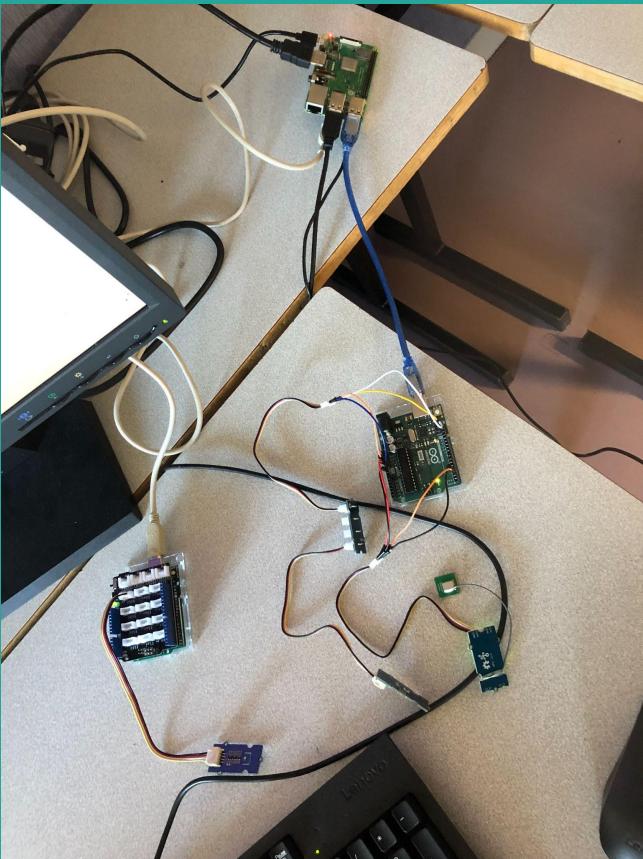
● Ma partie :

```
pi@raspberrypi:~/Desktop/ProjetRLMA $ ./a.out
*tvoc = 0          co2 = 400
tvoc = 21         co2 = 400
tvoc = 32         co2 = 400
tvoc = 59         co2 = 425
tvoc = 64         co2 = 522
tvoc = 99         co2 = 646
^C
pi@raspberrypi:~/Desktop/ProjetRLMA $
```

The screenshot shows a Windows desktop interface. At the top, there is a taskbar with icons for a Raspberry Pi, a globe, a folder, and a terminal window titled 'baseline_operation_e...'. To the right of the taskbar, the file path 'ProjetRLMA' is visible. Below the taskbar is a menu bar with French options: Fichier, Éditer, Rechercher, Affichage, Document, Projet, Construire, Outils. Under the 'Fichier' menu, there are icons for creating files and folders, opening, saving, and deleting. The main area contains two tabs: 'Com.c' and 'EcritureFichierJson.c'. The 'EcritureFichierJson.c' tab is active, showing the following JSON code:

```
1  {
2   "temperature": 0,
3   "humidite": 0,
4   "pm": 0,
5   "no2": 0,
6   "tvoc": 99,
7   "co2": 646,
8   "gps": {
9     "latitude": "0.000000",
10    "longitude": "0.000000"
11  }
12 }
```

● Ma partie :



● Problèmes rencontrés :

- Problèmes liés au chemin des bibliothèques pour utiliser les capteurs
- Problèmes liés à la version de node js
- Problèmes liés à mongo db et la version de raspbian OS

- Conclusion :

Où sommes nous arrivés dans le projet ?

Ce que j'ai retenue et appris :

- L'organisation et le travail en équipe
- Gérer les moments de baisse de motivation
- Rester patient et focus lors de potentielle problème

- Conclusion :

Possibles ajouts dans le futur ?

1. -Suggestions de masques à porter en fonction de la qualité de l'air
2. -Led rouge si la qualités de l'air est mauvaise
3. Imprimé un boîtier 3D pour contenir le raspberry ainsi que les capteurs