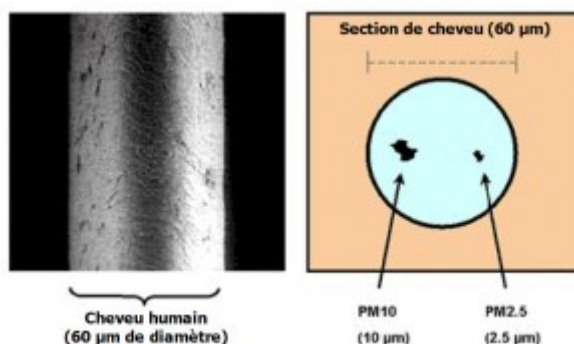
 <h2 style="text-align: center;">Capteur de particules fines SDS011</h2>	
<b>Sommaire :</b>	
I - Présentation du capteur de particules fines SDS011.....	1
II - Mise en œuvre du SDS011 avec un ESP32.....	2
III - Mise en œuvre du SDS011 avec un Raspberry Pi.....	2

## I - Présentation du capteur de particules fines SDS011

Les particules en suspension (notées « PM » en anglais pour « *Particulate matter* ») sont d'une manière générale les fines particules solides portées par l'eau ou solides et/ou liquides portées par l'air. 2 tailles sont définies : **PM10** (10 µm) et **PM2.5** (2.5 µm).



### Valeurs limites pour la protection de la santé humaine :

- Pour les **PM10** (applicables aux concentrations non liées à des événements naturels) :
  - 50 µg/m<sup>3</sup> en moyenne journalière, à ne pas dépasser plus de 35 jours par an ;
  - 40 µg/m<sup>3</sup> en moyenne annuelle.
- Pour les **PM2,5**, il n'y a pas de réglementation. L'Union européenne a fixé son objectif de qualité à 20µg/m<sup>3</sup> en moyenne sur l'année. Le Grenelle de l'environnement souhaitait arriver à 15µg/m<sup>3</sup>. L'Organisation Mondiale de la Santé recommande, elle, une valeur de 10 µg/m<sup>3</sup>.

Ce capteur est basé sur un laser **SDS011** PM2.5/PM10 permettant de tester avec précision et fiabilité la qualité de l'air. Ce laser fiable, rapide et précis mesure le taux (ou la concentration) de particules dans l'air compris entre 0,3 et 10 µm. Le capteur SDS011 est relié à un système numérique via un convertisseur USB vers UART TTL.



## II - Mise en œuvre du SDS011 avec un ESP32

Connexion du **SDS011** à l'**ESP32** :

S.No	SDS011	ESP32 Pin
1	5V	5V
2	GND	GND
3	RX	GPIO17 (TX2)
4	TX	GPIO16 (RX2)

Pour programmer ce capteur, il est nécessaire d'utiliser la librairie **SDS011 sensor Library**. La programmation du **SDS011** se fait de la façon suivante :

```
#include <SDS011.h>
SDS011 mySDS;
mySDS.begin(16, 17); // mySDS.begin(&Serial1, 16, 17);
//HardwareSerial port(2);
//mySDS.begin(&port);
float p10, p25;
int err;
err = mySDS.read(&p25, &p10);
if (!err) {
    Serial.println("PM2.5 (µg/m3) = " + String(p25));
    Serial.println("PM10 (µg/m3) = " + String(p10));
}
else Serial.println("Erreur SDS011");
```

## III - Mise en œuvre du SDS011 avec un Raspberry Pi

Connexion du **SDS011** au **Raspberry Pi** : Le **SDS011** est relié au **Raspberry Pi** via le **port USB**.

Si on veut utiliser ce capteur sur un **Raspberry Pi**, on peut utiliser une librairie écrite en **langage C** : <https://github.com/paulvha/sds011>. On peut aussi utiliser une librairie écrite en **Python** <https://pypi.org/project/sds011/>.

En **langage C**, on pourra utiliser le **main** suivant pour relever les grandeurs **pm10** et **pm25** :

```
#include <stdlib.h>
#include <stdio.h>
#include <unistd.h> // needed for sysconf(int name);
#include <malloc.h>
#include <sys/time.h> // needed for getrusage
#include <sys/resource.h> // needed for getrusage
#include <pthread.h>
#include <limits.h>

#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>

#include "serial.h"
#include "sds011_lib.h"
```

```
int fd = 0xff;           // filepointer
char port[20] = "/dev/ttyUSB2";

int main(int argc, char *argv[])
{
sds011_response_t rep[255];
while (1) {
    system("modprobe usbserial");
    system("modprobe ch341");
    fd = open(port, O_RDWR | O_NOCTTY | O_SYNC);
    if (fd < 0) {
        printf("could not open %s\n", port);
        exit(EXIT_FAILURE);
    }
    configure_interface(fd, B9600);
    set_blocking(fd, 0);
    usleep(10000);
    tcflush(fd,TCIOFLUSH);
    if (Try_Connect(fd) == SDS011_ERROR)
    {
        printf("Error during trying to connect\n");
    }
    else
    {
        printf("Current DeviceID: 0x%04x\n", Get_DeVID());
        if (Query_data(1,0, rep) == SDS011_ERROR) {
            printf("error during getting current display data\n");
        }
        else printf("Main PM2.5: %.2f, PM10: %.2f\n", rep->pm25, rep->pm10);
    }
    if (fd != 0xff) {
        restore_ser(fd);
        close(fd);
    }
    sleep(10);
}
return 0;
}
```