

A dark blue vertical bar on the left side of the page. A blue arrow points to the right from the bar, containing the date.

01/02/2017

# Projet PAVIC

Mesure de paramètres  
environnementaux pour la ville  
intelligente

Several thin, curved lines in dark blue and light grey originate from the bottom left and sweep upwards and to the right.

Grégoire de CHASTEIGNER  
Alexandre MEMEINT  
Erwan RICHARD  
Baptiste ROLANDEAU  
Sujeeth SUSINTHIRAN  
Guillaume THIENNOT  
ESEO

## Table des matières

Introduction.....	2
Schéma fonctionnel.....	3
Bloc Capteurs de luminosité/UV .....	4
Bloc Capteurs de température/pression/Humidité .....	7
Bloc Capteurs de gaz/poussières.....	9
Bloc Capteurs météorologique.....	15
Bloc Transmission RF.....	19
Bloc Centrale .....	21
Sources .....	27
Annexes .....	28
Table des Figures .....	28
Schémas électriques.....	28
Etudes des gaz et poussières mesurés .....	31
Machine à état (Centrale) .....	34
Machine à état (Capteurs).....	35

## Introduction

Le projet de fin d'études qui nous été confié avait pour intitulé « Mesure des paramètres environnementaux d'une ville intelligente », dans le but de pouvoir récolter suffisamment de données pour envisager la prévention des épisodes de pollinisation au sein de la ville d'Angers. Le projet était supporté par PAVIC.

Pour cela, les paramètres météorologiques sont mesurés, ainsi que les différents paramètres de l'air, avec les teneurs en gaz polluants et poussières, les paramètres liés à la température et à la luminosité.

Différents blocs constituent le système conçu :

➤ Bloc Capteurs de luminosité/UV

Ce bloc permet la mesure pour des événements visuels liés au taux de luminosité ou encore de l'indice UV permettant une prévention pour les personnes sujettes à de fortes sensibilités au soleil pendant les périodes les plus fortes.

Ce bloc permettra la mesure des données suivantes :

- Mesure de l'indice UV
- Mesure de la luminosité

➤ Bloc Capteurs de température/pression/humidité

Ce bloc permet la mesure pour des événements liés à la température, la pression et l'humidité. Il permet entre autres la prévention avec la récolte des données sur le long terme de pics de température ou encore de fortes précipitations.

Ce bloc permettra la mesure des données suivantes :

- Mesure de la température
- Mesure de la pression
- Mesure de l'humidité

➤ Bloc Capteurs de gaz/poussières

Cela comprend la mesure des gaz présents dans l'air, notamment ceux pouvant être liés à des problèmes respiratoires chez l'être humain. L'objectif du projet est d'apporter une solution à la prévention des épisodes de pollinisation, c'est pour cette raison que les poussières de diamètres inférieures à 10µm sont également mesurées.

Cette fonction « Mesure des taux des gaz et poussières » possède donc les caractéristiques suivantes :

- Mesure des polluants chimiques
- Mesure des poussières PM10
- Transmission des données à la centrale

### ➤ Bloc Capteurs météorologiques

Ce bloc permet la mesure de données météorologiques basiques mais néanmoins essentielles. En effet ces données seront corrélées avec l'ensemble des données mesurées et permettront une meilleure prévention des épisodes de pollinisation. Ce bloc permettra la mesure des données suivantes :

- Mesure vitesse du vent → Anémomètre
- Mesure direction du vent → Girouette
- Mesure précipitations → Pluviomètre

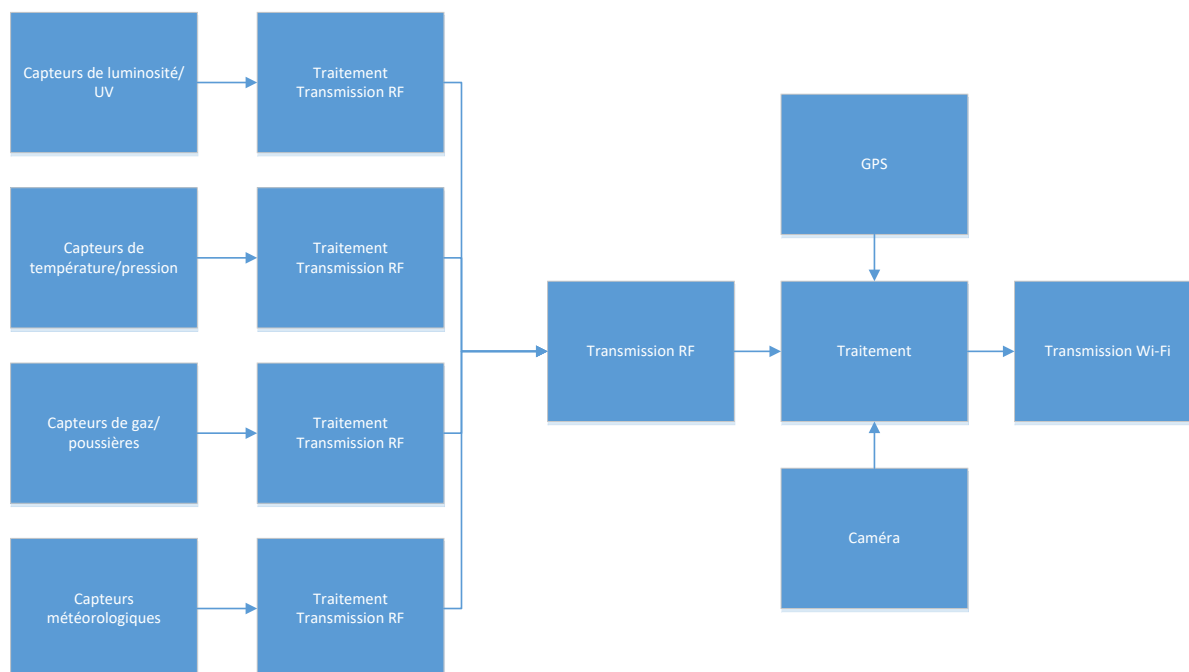
### ➤ Bloc Centrale

Ce bloc est le bloc responsable des autres blocs capteurs dans sa zone proche (< 10-15m). Il doit centraliser toutes les données récupérées, avant de pouvoir les envoyer toutes sur un broker MQTT, de manière intelligente pour permettre un traitement à posteriori avec des méthodes de *deep learning*.

La centrale doit pour cela être capable de donner sa position ou un identifiant, la donnée, le capteur d'où provient la donnée, ainsi que les informations temporelles de la donnée. Elle sera aussi responsable de la coordination des cartes capteurs.

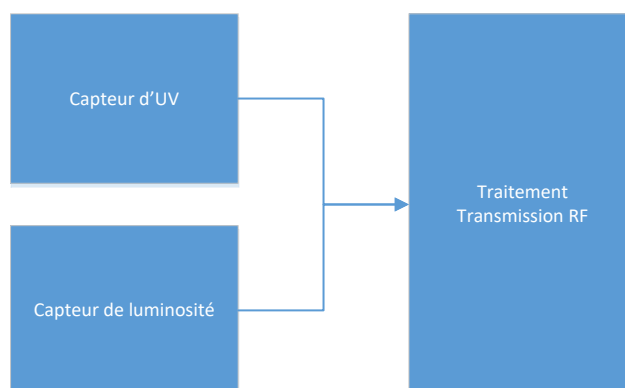
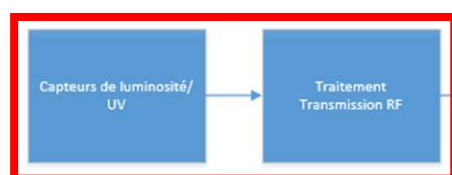
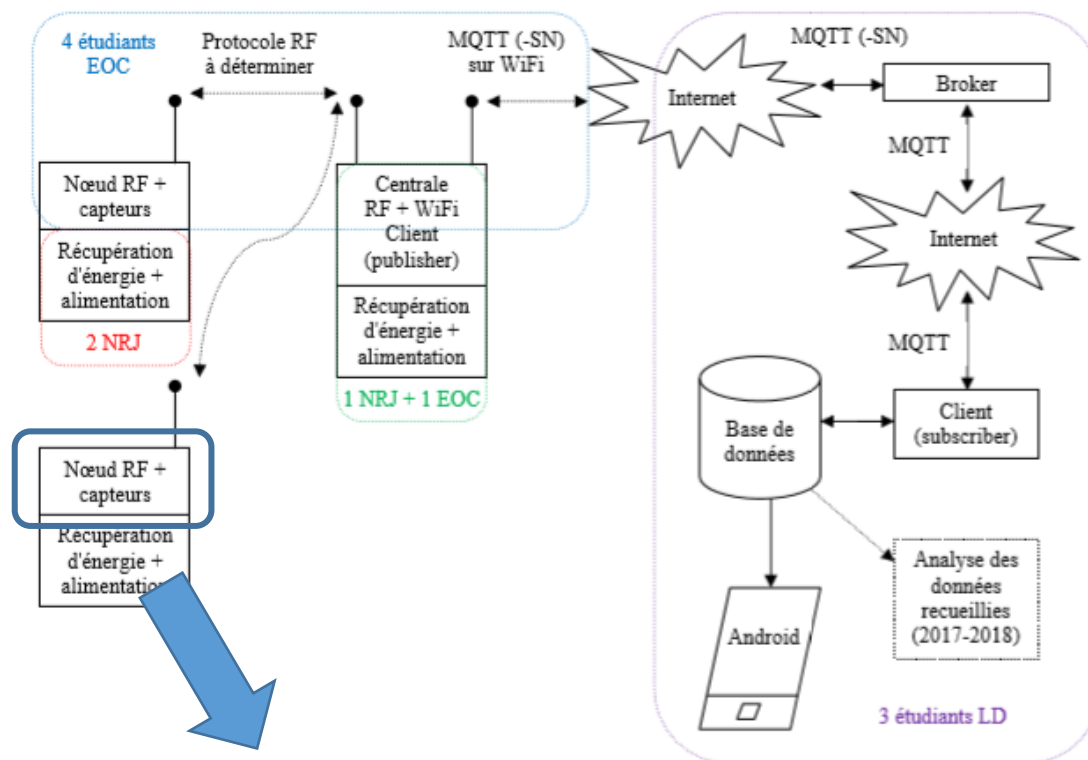
## Schéma fonctionnel

Voici le schéma fonctionnel global de notre projet, dont les différents blocs sont étudiés et détaillés par la suite :



## Bloc Capteurs de luminosité/UV

### Etude fonctionnelle



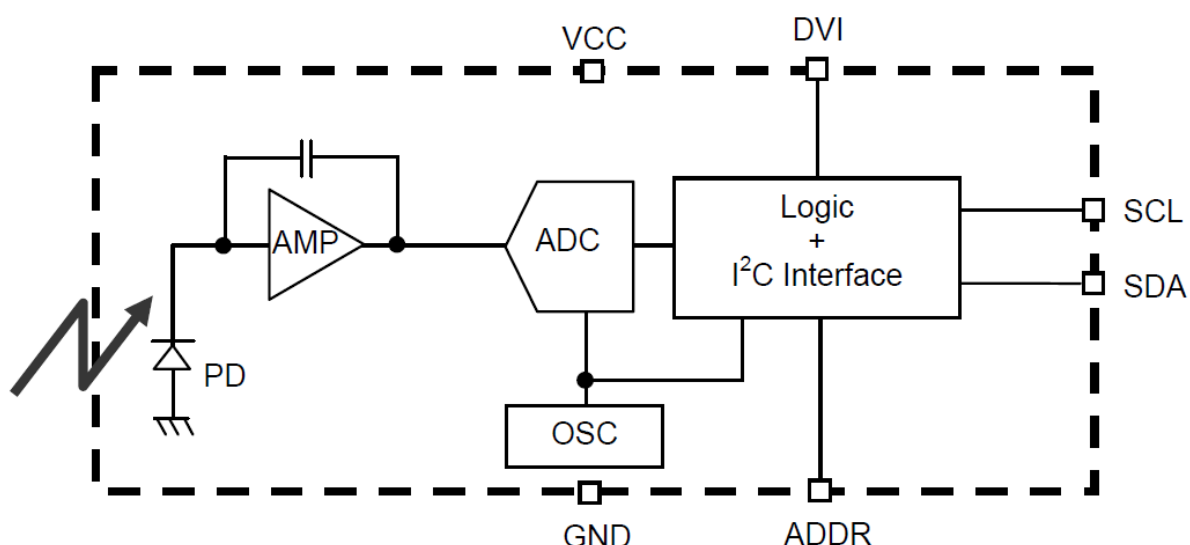
## Etude structurelle

Le schéma structurel de la carte capteur de luminosité et UV est présent en Annexes. Nous détaillons ici chaque fonction afin d'étudier les capteurs correspondants.

### Mesure de la luminosité

La mesure du taux de luminosité est réalisée à l'aide du capteur BH1750. Ce capteur est idéal, car il est notamment utilisé dans le domaine de la téléphonie pour capter et ajuster la lumière d'un téléphone portable en fonction de la luminosité mesurée en haute résolution (1 à 65 535 lx).

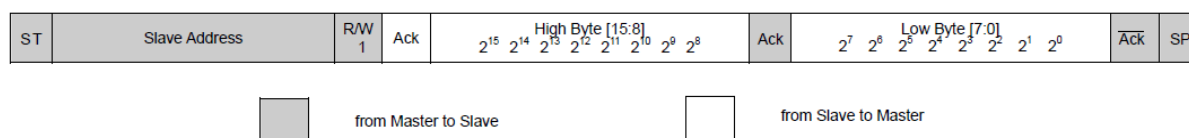
Ce capteur réalise la mesure à l'aide d'une diode via le block diagramme ci-dessous :



Ici nous avons une diode (PD) qui possède un temps de réponse approximatif équivalent à un œil humain. Elle va permettre de mesurer un courant qui va être converti en tension avec l'aide d'une intégration AMP. Cette donnée va être convertie en une donnée 16 bits grâce à un ADC.

Le calcul et le traitement des données sont gérés dans la partie « Logic + I²C Interface » pour permettre de calculer le taux de luminosité et enfin la transmettre sous forme de registre.

La transmission se fera avec le protocole I²C qui est en principe simple d'utilisation.



Le protocole consiste à envoyer dans un premier temps l'adresse de l'esclave qui sera pour notre utilisation « 1011100 » pour récupérer la valeur de la luminosité selon le principe suivant :

Exemple :

High Byte = "1000\_0011"

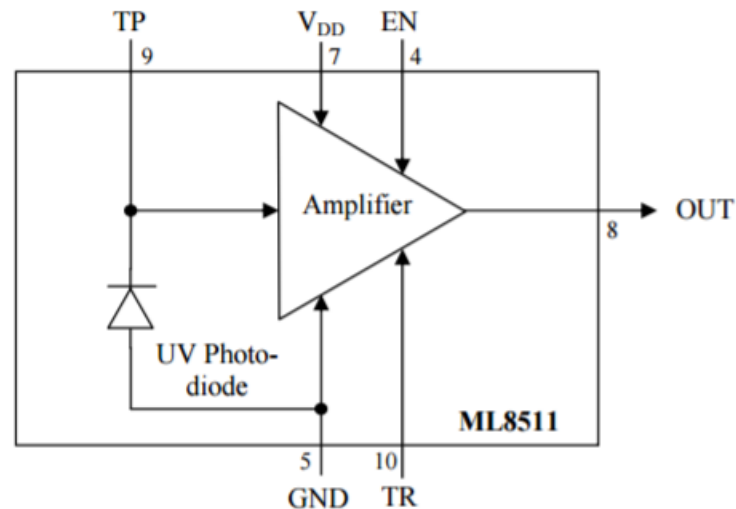
Low Byte = "1001\_0000"

$$\frac{(2^{15} + 2^9 + 2^8 + 2^7 + 2^4)}{1.2} \approx 28\,067 \text{ lx}$$

### Mesure UV

La mesure du taux d'UV est réalisée à l'aide du capteur ML8511. Ce capteur est idéal car il est notamment utilisé dans le domaine de la téléphonie, de la navigation, du jeu, et bien d'autres domaines avec une précision de 0 à 15 d'indice UV.

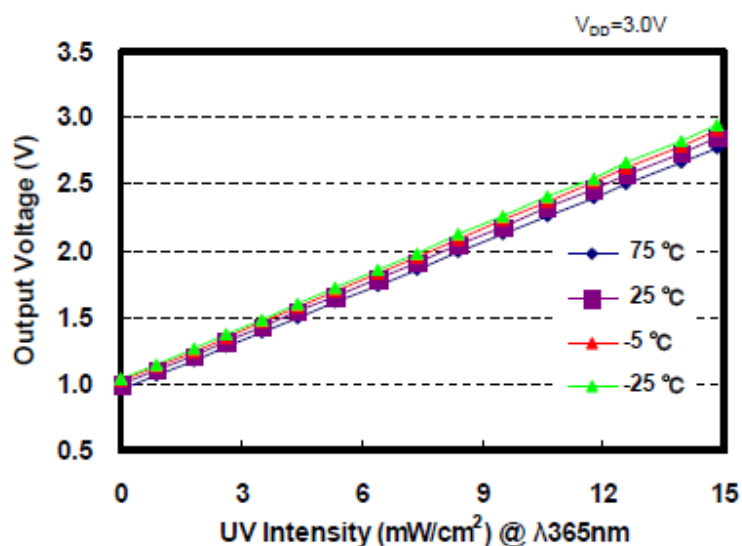
Ce capteur réalise la mesure à l'aide d'une photo diode via le block diagramme ci-dessous :



Ici nous avons une photo diode dont la tension est amplifiée pour la rendre avec un ADC. Différentes broches sont d'ailleurs présentes pour pouvoir tester (TP), ou encore mettre en veille (EN) le module.

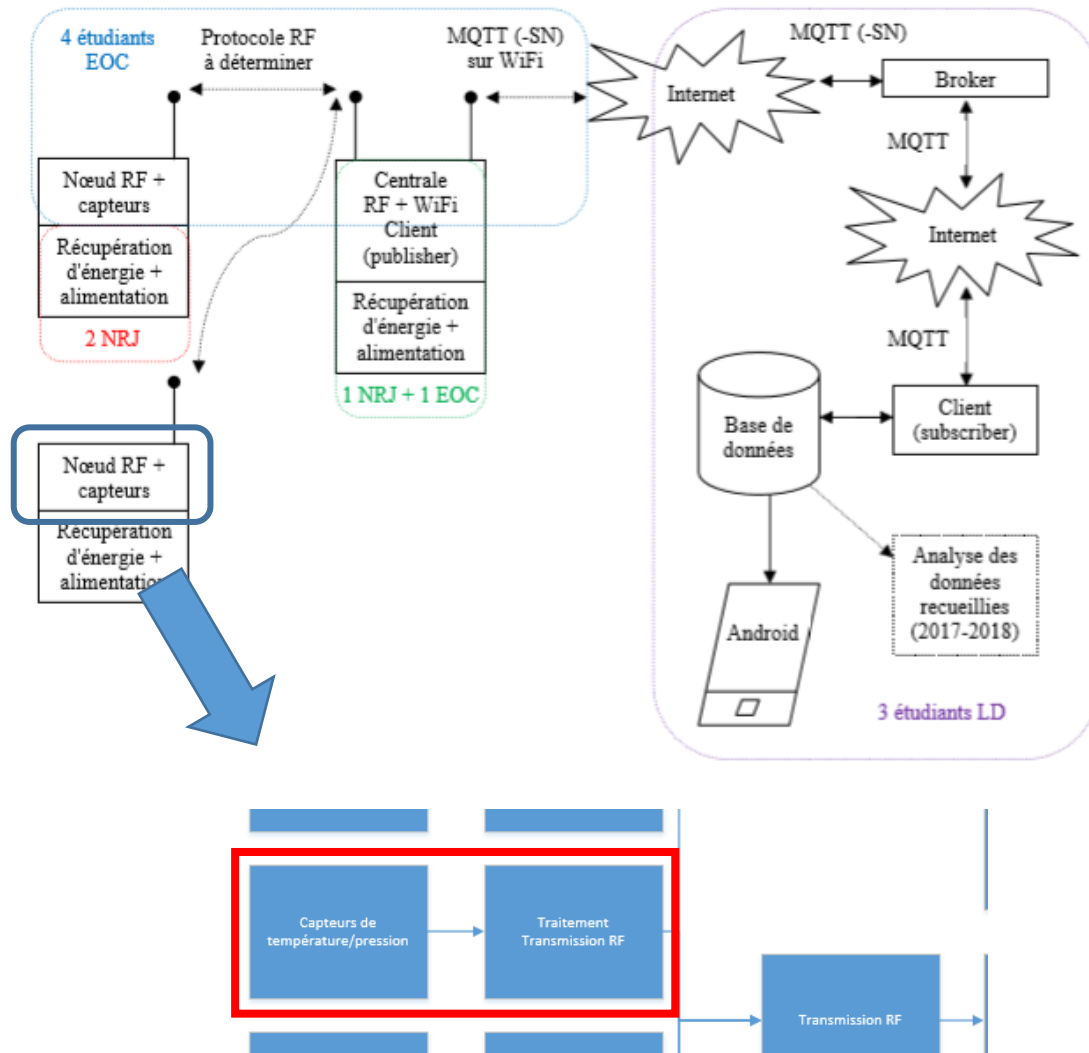
Le calcul et le traitement des données sont effectuées directement avec un ADC, dont la valeur en sortie est à corrélérer à l'aide de la courbe tension ci-dessous.

OUTPUT VOLTAGE- UV INTENSITY CHARACTERISTICS



## Bloc Capteurs de température/pression/Humidité

### Etude fonctionnelle





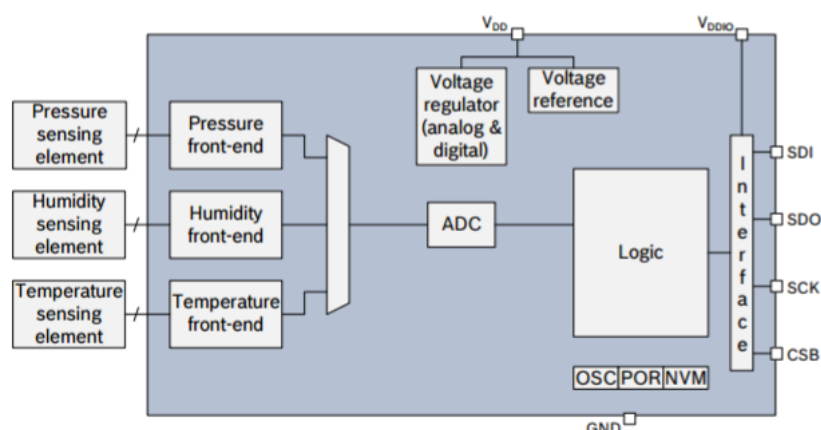
## Etude structurelle

Le schéma structurel de la carte capteur de température, pression et humidité est présent en Annexes. Nous détaillons ici chaque fonction afin d'étudier les capteurs correspondants.

### Mesure de la température/pression/humidité

Les mesures de température, pression et humidité sont réalisées à l'aide du capteur BME280. Ce capteur est idéal car il peut effectuer des relevés d'humidité et de pression de manière indépendante. Il est notamment utilisé dans de nombreuses applications comme la domotique.

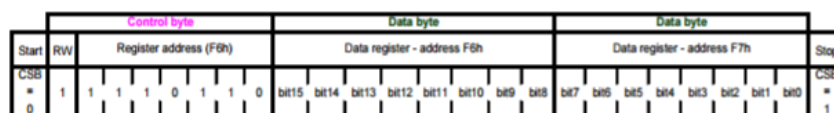
Ce capteur réalise la mesure via le block diagramme ci-dessous :



Ici nous avons trois relevés indépendants qui correspondent aux valeurs de la température, pression et humidité. Ces données vont être converties en données 16 bits avec un ADC.

Le calcul et le traitement des données sont gérés dans la partie « Logic + SPI Interface » pour permettre de calculer le taux de température, pression, humidité et enfin la transmettre sous forme de registre.

La transmission se fera avec le protocole SPI qui est en principe simple d'utilisation.



Le protocole consiste à passer CSB à 0 pour déclencher la lecture et envoyer dans un premier temps le contrôle qui consistera à rentrer l'adresse du registre pour récupérer la valeur souhaitée de température, pression ou encore humidité selon le tableau suivant :

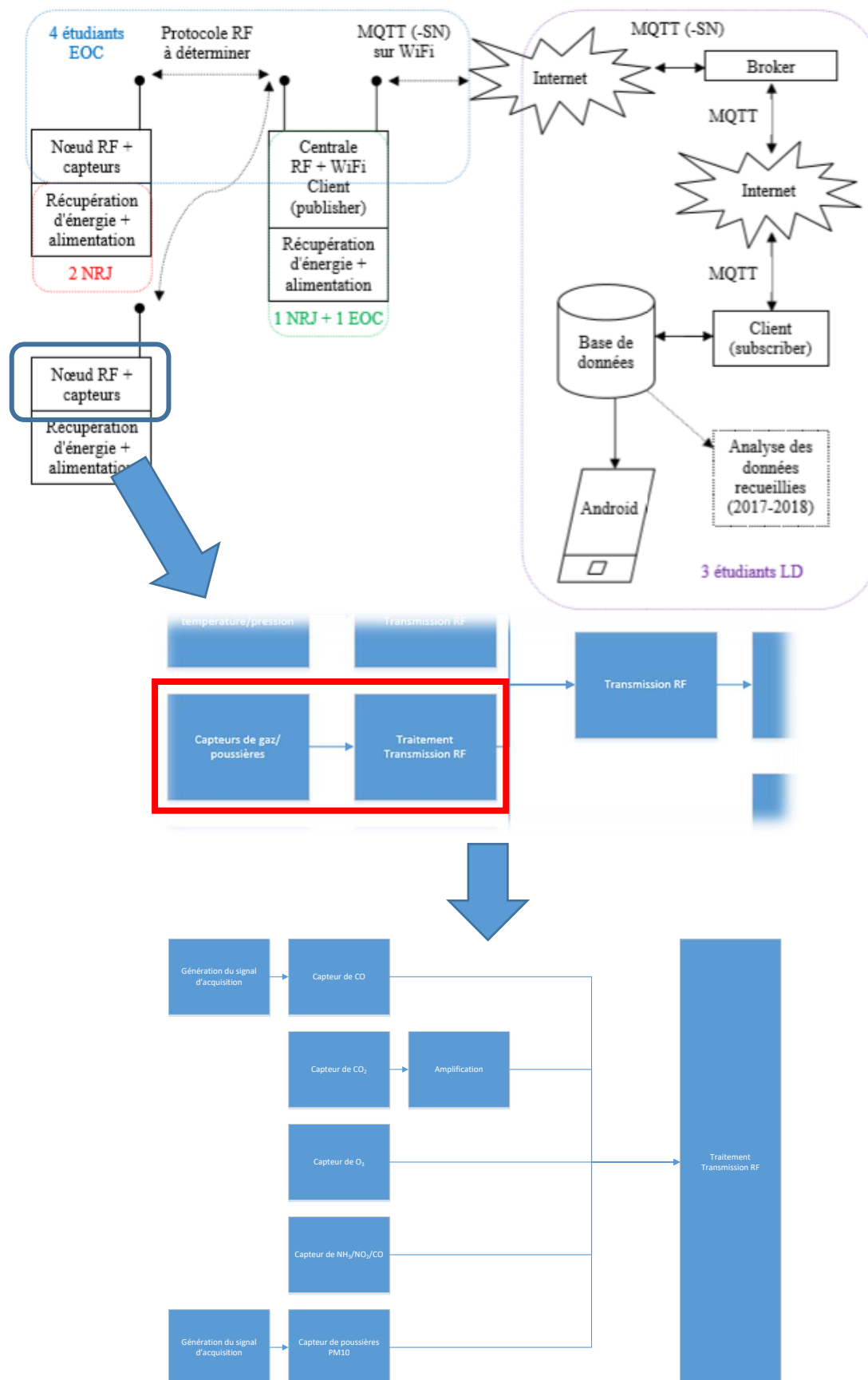
Register Name	Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Reset state				
hum_lsb	0xFE	hum_lsb<7:0>								0x00				
hum_msb	0xFD	hum_msb<7:0>								0x80				
temp_xlsb	0xFC	temp_xlsb<7:4>				0	0	0	0	0x00				
temp_lsb	0xFB	temp_lsb<7:0>								0x00				
temp_msb	0xFA	temp_msb<7:0>								0x80				
press_xlsb	0xF9	press_xlsb<7:4>				0	0	0	0	0x00				
press_lsb	0xF8	press_lsb<7:0>								0x00				
press_msb	0xF7	press_msb<7:0>								0x80				
config	0xF5	t_sb[2:0]			filter[2:0]			spi3w_en[0]		0x00				
ctrl_meas	0xF4	osrs_t[2:0]			osrs_p[2:0]			mode[1:0]		0x00				
status	0xF3					measuring[0]		im_update[0]		0x00				
ctrl_hum	0xF2							osrs_h[2:0]		0x00				
calib26_calib41	0xE1...0xF0	calibration data								individual				
reset	0xE0	reset[7:0]								0x00				
id	0xD0	chip_id[7:0]								0x60				
calib00_calib25	0x88...0xA1	calibration data								individual				

Registers:	Reserved registers do not change	Calibration data	Control registers	Data registers	Status registers	Chip ID	Reset
Type:		read only	read / write	read only	read only	read only	write only

## Bloc Capteurs de gaz/poussières

### Etude fonctionnelle



## Etude structurelle

Le schéma structurel de la carte capteur gaz est présent en Annexes. Nous détaillons ici chaque fonction afin d'étudier les capteurs correspondants.

### Mesure de l'Ozone

La mesure de l'ozone ( $O_3$ ) est effectuée grâce au capteur de SGX Sensortech MiCS 2614. Celui-ci utilise la technologie MOS (Metal Oxide Semiconductor) pour mesurer la quantité du gaz. L'ozone cause une oxydation de la couche sensible du capteur, ce qui augmente sa résistivité proportionnellement au taux d' $O_3$  capté.

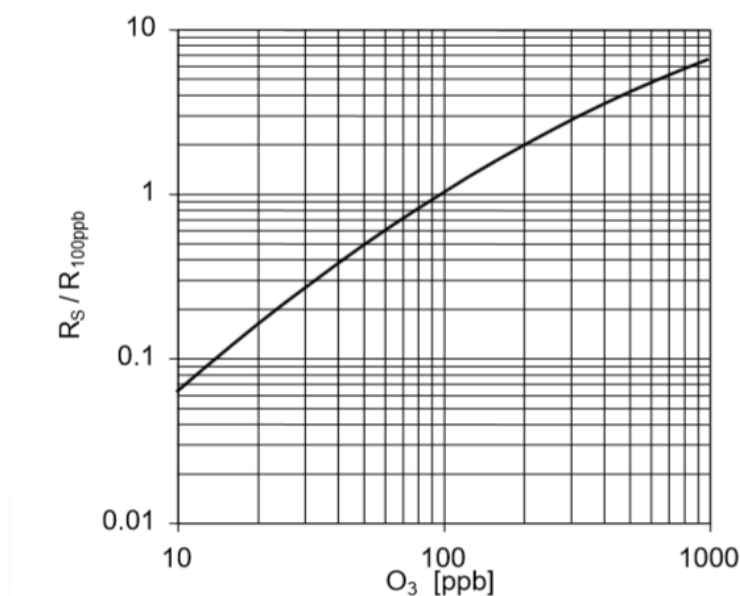


Figure 1: Evolution de la résistivité du capteur selon le taux d'Ozone présent dans l'air

Le capteur nécessite une étape de calibration pour connaître la valeur exacte du taux d'ozone mesuré, ce qui ne peut se faire qu'en présence d'un appareil précédemment calibré dans un pièce où le taux est stable. En revanche, sans calibration on peut toutefois avoir un aperçu de la valeur de ce taux en mesurant la tension en sortie du capteur.

Le MiCS 2614 possède une gamme de mesure allant de 10 à 1000 ppb, ce qui correspond à des taux de présence d'ozone dans l'air allant de 5,1 à 510  $\mu\text{g}/\text{m}^3$ , alors que le seuil de pollution journalier a été défini par les professionnels de santé à 120  $\mu\text{g}/\text{m}^3$ .

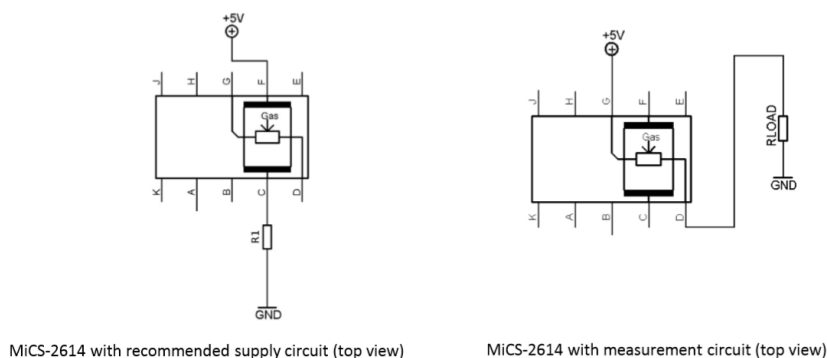


Figure 2: Structure électrique du capteur reprise dans le schéma de la carte

### Mesure du CO

La mesure du taux de monoxyde de carbone est réalisée par deux capteurs différents : le « classique » MQ7 et un capteur utilisant la technologie MOS, le MiCS6814, de la même famille que le MiCS2614 présenté précédemment. La présence de ces 2 capteurs permet à l'utilisateur de se faire une idée sur la précision de mesure des 2 technologies, tout en sachant que la technologie MOS possède l'énorme avantage d'être bien plus économe en énergie que le traditionnel filament chauffé permettant la réaction électrochimique.

### MQ7

Le MQ7 possède la particularité, pour un capteur de la série MQ, de devoir être alimenté de façon cyclique par 2 tensions différentes lors de la mesure, afin de chauffer le filament à haute puis basse température afin de récupérer le taux de gaz dans l'air.

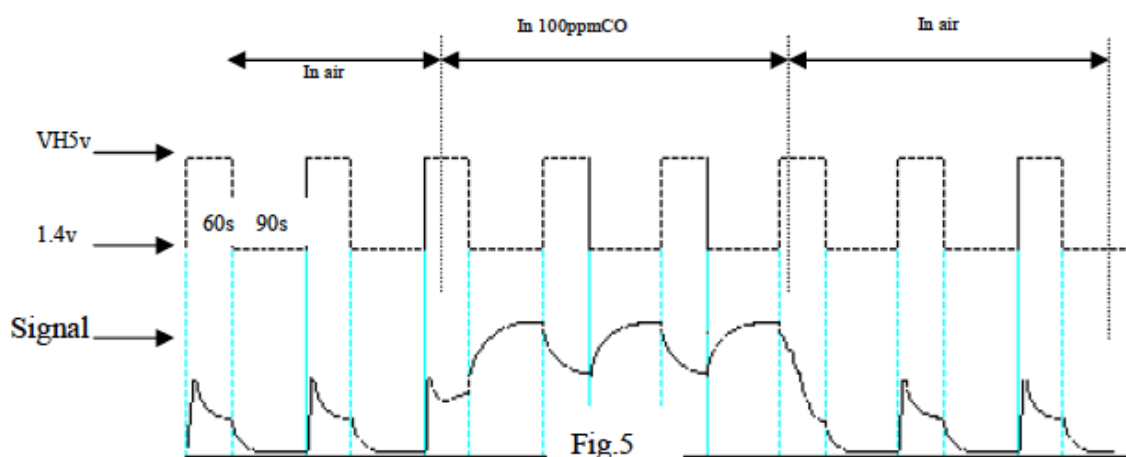


Figure 3: Conditionnement de l'alimentation du MQ7

Comme le capteur nécessite beaucoup de courant pour fonctionner, une structure à base de relais commandé par un transistor a été mise au point afin de générer les 2 tensions différentes.

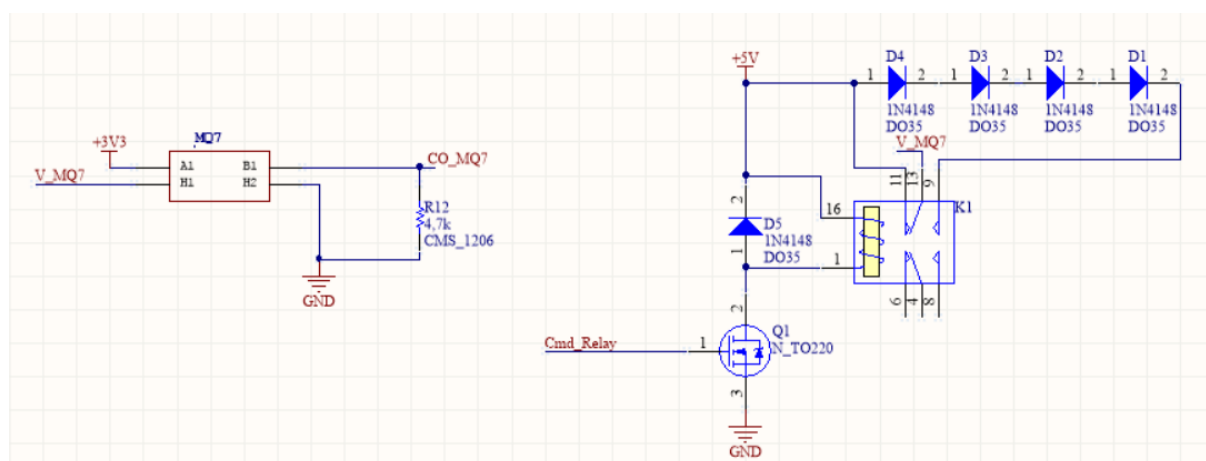


Figure 4: Structure de commande de l'alimentation du MQ7

Ainsi, en fonction de l'état logique de la sortie *Cmd\_relay*, le transistor alimente ou non la bobine interne du relais, qui lui assigne en sortie une tension de 5V ou 1,4V (à l'aide des 4 diodes en série).

### MiCS 6814

L'étude du MiCS6814 est détaillée dans la partie suivante.

### Mesure des gaz polluants industriels

La mesure des taux de polluants industriels et agricoles, tel que le dioxyde d'azote, l'ammoniac ou le monoxyde de carbone, est réalisée par le MiCS 6814, capteur issu de la même famille que le MiCS 2614, et utilisant donc la même technologie basée sur le MOS. D'autres gaz sont identifiables par le capteur, comme l'éthanol, l'hydrogène ou le méthane, mais les valeurs mesurées sont insuffisantes pour être exploitables.

Le capteur possède 3 sorties, correspondant à 3 comportements différents afin de mesurer les gaz oxydés, les gaz réduits et l'ammoniac. La couche sensible aux gaz oxydés permet de mesurer le taux de dioxyde d'azote et d'hydrogène notamment, alors que celle sensible aux gaz réduits réagit au monoxyde de carbone ainsi qu'à l'éthanol.

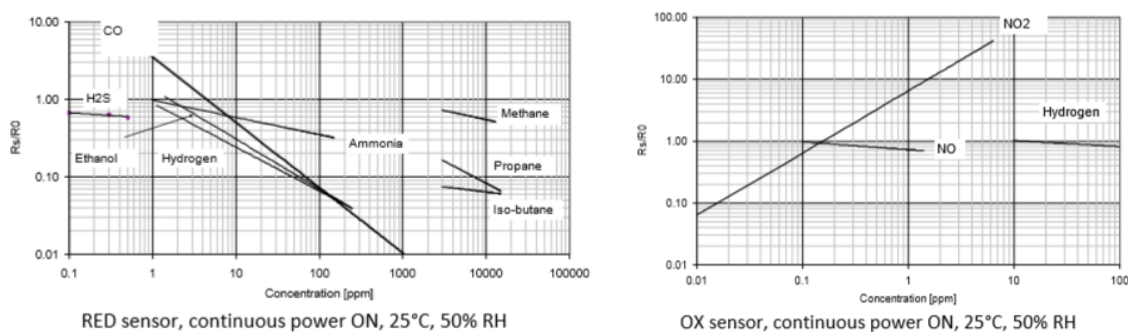


Figure 5: Sensibilité des différentes couches du capteur

Le MiCS 6814 est capable de mesurer des taux de CO entre 1 et 1000ppm, de NH<sub>3</sub> entre 1 et 500ppm, et de NO<sub>2</sub> entre 0.05 et 10ppm. Comme pour le capteur d'ozone, celui-ci nécessite une calibration en milieu stable pour connaître précisément les valeurs des taux de gaz captés, d'autant plus que les 3 couches sensibles composant le capteur sont réactives à plusieurs gaz.

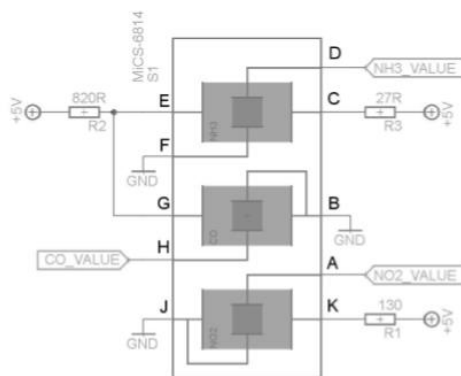


Figure 6: Structure recommandée pour le MiCS 6814

### Mesure du CO<sub>2</sub>

La mesure du taux de dioxyde de carbone est réalisée à l'aide du capteur MG811, dont la valeur de sortie est amplifiée afin de la rendre plus facilement exploitable.

Ce capteur réalise la mesure à l'aide d'une réaction chimique qui se produit lorsque le CO<sub>2</sub> est détecté par un filament chauffé. Il réagit à la manière d'une pile, et fournit une tension à ses bornes qui diminue lorsque la teneur en CO<sub>2</sub> augmente dans l'air. Typiquement, on retrouve une différence de potentiel comprise entre 200 et 600mV pour une teneur en CO<sub>2</sub> de 10000 à 400ppm, l'air étant considéré comme « pur » lorsque le taux de CO<sub>2</sub> est inférieur à 400ppm. La tension maximale générée par le capteur est donc de l'ordre de 600mV, ce qui est très faible.

De plus, la sortie du capteur suit une loi logarithmique lorsque le taux de CO<sub>2</sub> est supérieur à 400ppm, de telle sorte que (avec C<sub>CO2</sub> représentant le taux de CO<sub>2</sub>) :

$$V_s = V_0 + \Delta V_s / (\log_{10} 400 - \log_{10} 1000) * (\log_{10} C_{CO2} - \log_{10} 400)$$

$$\text{Where } \Delta V_s = \text{sensor output@400ppm} - \text{sensor output@1000ppm}$$

D'après les retours d'expérience,  $\Delta V_s$  vaut généralement 100mV (dépend du capteur). L'amplification de la tension de sortie par un facteur de 5,5 est donc intéressante pour retrouver en l'entrée de l'ADC une tension comprise entre 0,5V et 3,3V.

Compte tenu de sa faible impédance, il est conseillé d'ajouter un AOP possédant une très forte impédance en entrée afin de ne pas influencer sur la tension de sortie du capteur. Pour cette raison, un TL081 sert à la fois d'amplificateur et de suiveur grâce à son impédance d'entrée de 1TΩ théorique et au montage associé.

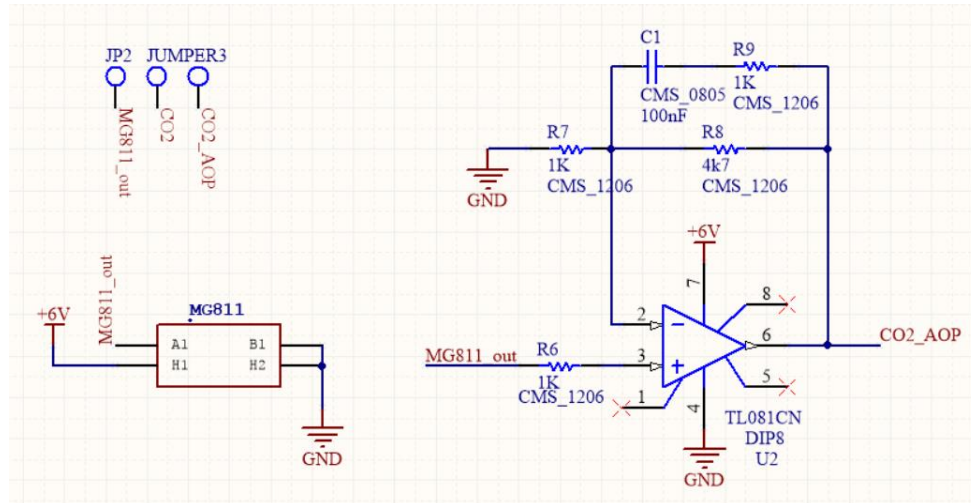


Figure 7 : Structure du capteur de CO<sub>2</sub>

Le gain apporté par celui-ci est paramétré par les valeurs de R7 et R8, avec la formule suivante :

$$V_{out} = V_{in} * (1 + \frac{R8}{R7})$$

Il vaut ici 5,7.

La présence d'un sélecteur permet de choisir la valeur à relier à l'entrée de l'ADC, entre la valeur brute du capteur et celle amplifiée. On note également la présence d'un filtre passe-bas RC en sortie du TL081, permettant de lisser la tension.

### Mesure du taux de poussières PM10

Le taux de particules fines présentes dans l'air est mesuré à l'aide du capteur GP2Y101AU de Sharp. Celui-ci présente la particularité d'être sensible aux poussières de diamètre inférieur à  $10\mu\text{m}$ , tel que la fumée de cigarette ou le pollen. Il possède un émetteur infrarouge ainsi qu'une photodiode, qui lui permettent de « compter » le nombre de particules fines présentes dans l'air. Le filtre passe-bas situé en sortie du montage permet d'approximer cette valeur, qui est donc la plus représentative possible du taux de poussières présentes.

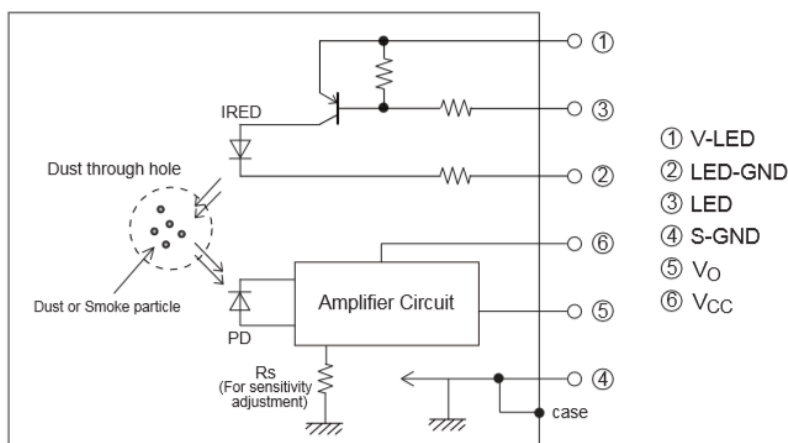


Figure 8: Principe de fonctionnement du capteur de poussières

Le capteur se commande en appliquant un signal PWM sur la broche reliée à un transistor interne pilotant la LED, qui fournit une faisceau IR permettant au système de compter les particules fines.

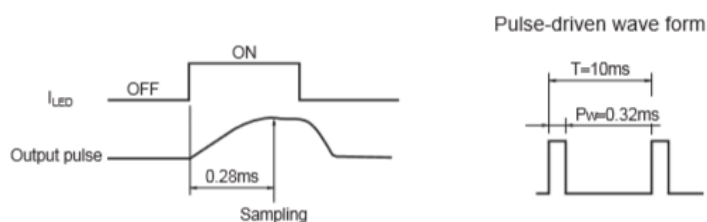


Figure 9: PWM à générer afin de commander le capteur

La tension analogique présente en sortie du montage est directement représentative de la densité de poussière mesurée.

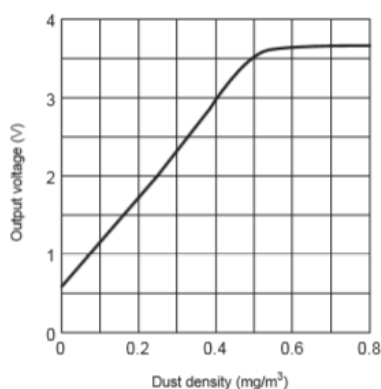
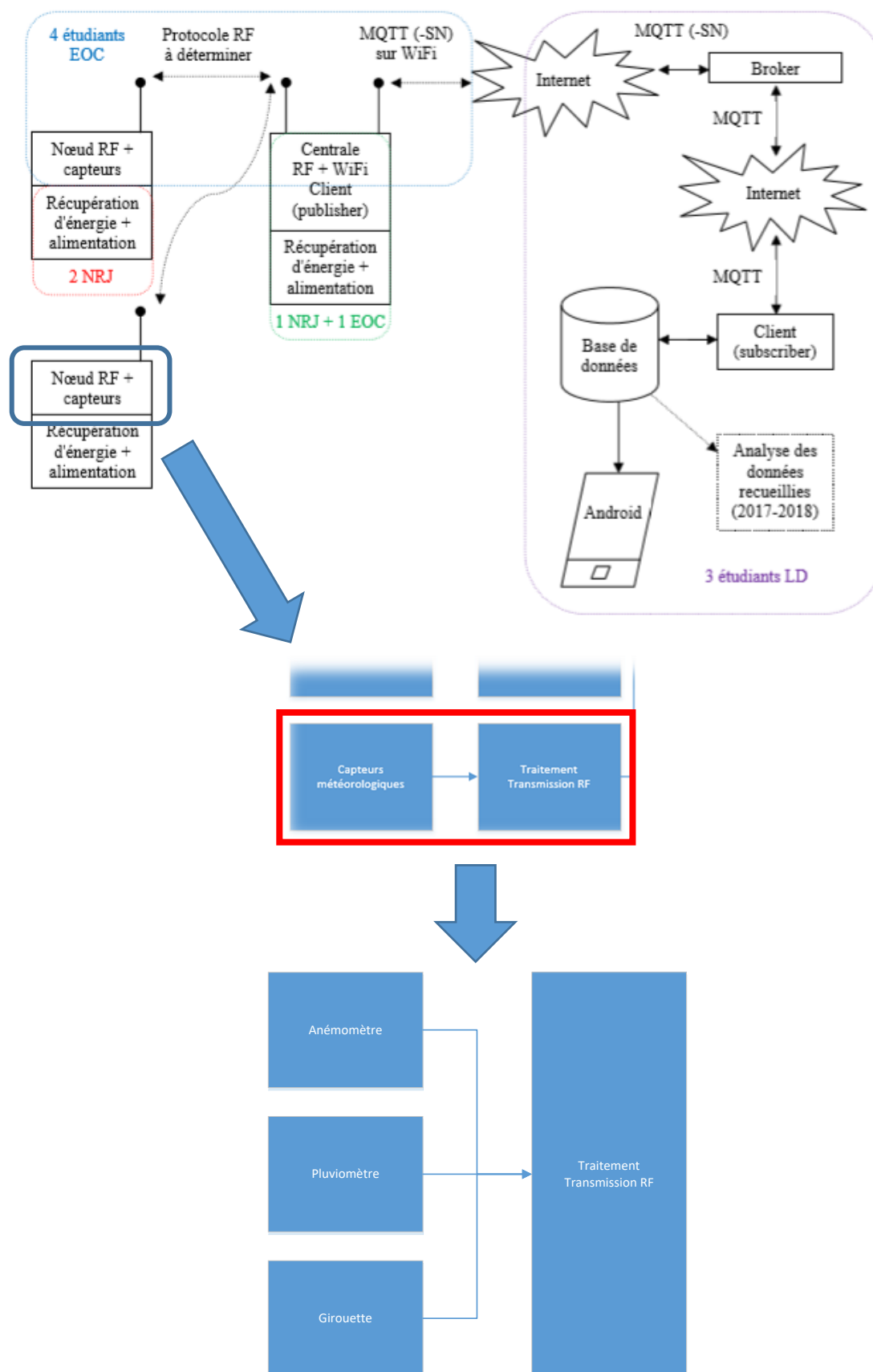


Figure 10: Evolution de la tension de sortie en fonction du taux de poussières

## Bloc Capteurs météorologique

### Etude fonctionnelle



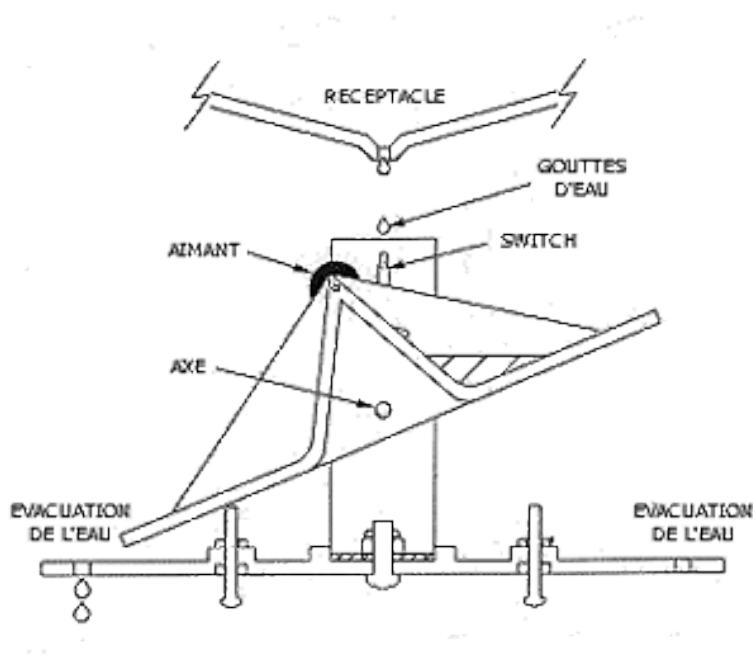


## Etude structurelle

Les capteurs météorologiques se connectent via un adaptateur sur la carte capteurs. Les schémas structurels de ceux-ci sont disponibles en Annexes. Nous détaillons ici chaque capteur associé afin d'en comprendre le fonctionnement et le déroulement de la mesure.

### Pluviomètre

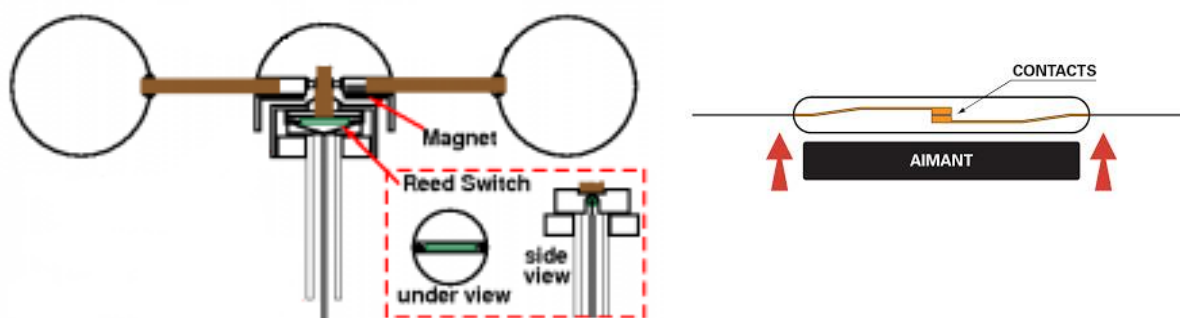
Le pluviomètre qui permet de connaître les précipitations en millimètres est réalisé par le capteur N77NF. Ce capteur fonctionne un peu sur le même principe que celui de l'anémomètre. Il est constitué d'une partie mobile, constituée de deux réservoirs avec un aimant fixé entre ceux-ci. D'une partie fixe sur laquelle se trouve un contacteur et d'une dernière partie qui vient collecter l'eau de pluie. Lorsqu'il pleut l'un des godets se remplit d'eau et lorsqu'il devient trop lourd, il bascule pour se vider entraînant le passage de l'aimant devant le contacteur.



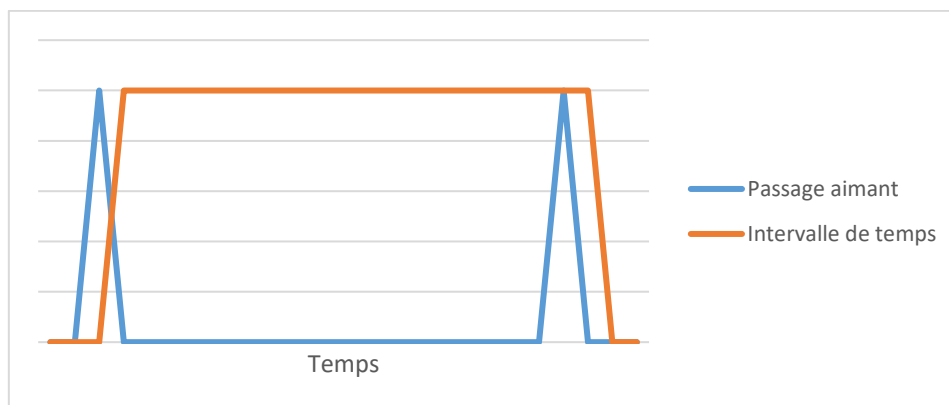
Afin de récupérer les précipitations en mm il suffit de compter ce nombre de passages de l'aimant et de le multiplier par 0,2974. Ce chiffre correspond à la quantité d'eau qu'il a plu en millimètres. Dans notre cas, nous avons décidé afin de ne jamais rater de basculement du godet de récupérer l'information avec une GPIOTE. Cela nous permet ainsi de gagner du temps processeur.

### Anémomètre

La mesure de la vitesse du vent est réalisée à l'aide du capteur N82NF. Ce capteur est assez simple d'utilisation, en effet un aimant est fixé à un point de la périphérie de la partie mobile de l'anémomètre. Cet aimant vient ouvrir et fermer un contacteur situé lui sur la partie fixe de l'anémomètre. Ce contacteur va donc se fermer à chaque fois que la girouette aura fait un tour sur elle-même.



Pour récupérer la vitesse du vent, il suffit de déclencher un timer au moment où l'on détecte un premier passage puis de l'arrêter lors du second passage. Avec le temps ainsi obtenu nous pouvons en déduire la vitesse du vent. Pour cela, nous devons d'abord transformer ce temps en une fréquence puis de multiplier cette fréquence par 2,4. En effet lors de nos recherches sur internet, il s'est avéré que si l'on obtient une fréquence de 1Hz entre deux passages de l'aimant, cela correspondait à une vitesse du vent de environ 2,4km/h.

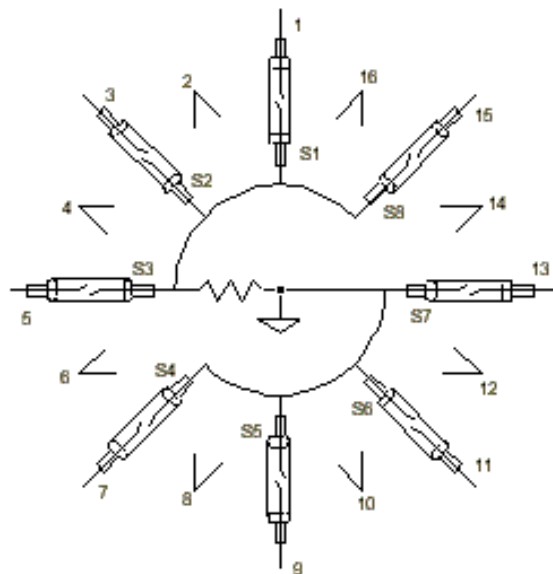


Afin d'obtenir une mesure fiable de la vitesse du vent et ainsi éviter la mesure d'une bourrasque, il est préférable d'effectuer une série de mesures et d'en calculer la moyenne. Dans notre cas nous souhaitons effectuer une mesure toute les secondes 10 fois dans le but d'obtenir une vitesse du vent cohérente.

Il faut aussi prendre en compte dans la conception de l'algorithme de mesure le cas où aucune détection ne se passe, soit parce que la vitesse du vent est nulle soit parce que le capteur est défectueux. Nous avons donc eu l'idée d'intégrer un délai d'une seconde au-delà duquel si aucun événement ne s'est produit, on considère la vitesse du vent comme nulle. Ceci dans le but d'éviter de bloquer l'exécution du programme dans une boucle potentiellement infinie.

### Girouette

La girouette, permet, elle de donner la direction du vent. Cette fonction est assurée par le capteur N81NF. Le principe de celle-ci est un peu différent car à chaque position de la girouette correspondra une valeur de résistance précise. Ce capteur est donc équipé de huit contacteurs et de huit résistances qui seront utilisés en mode pont diviseur de tension.

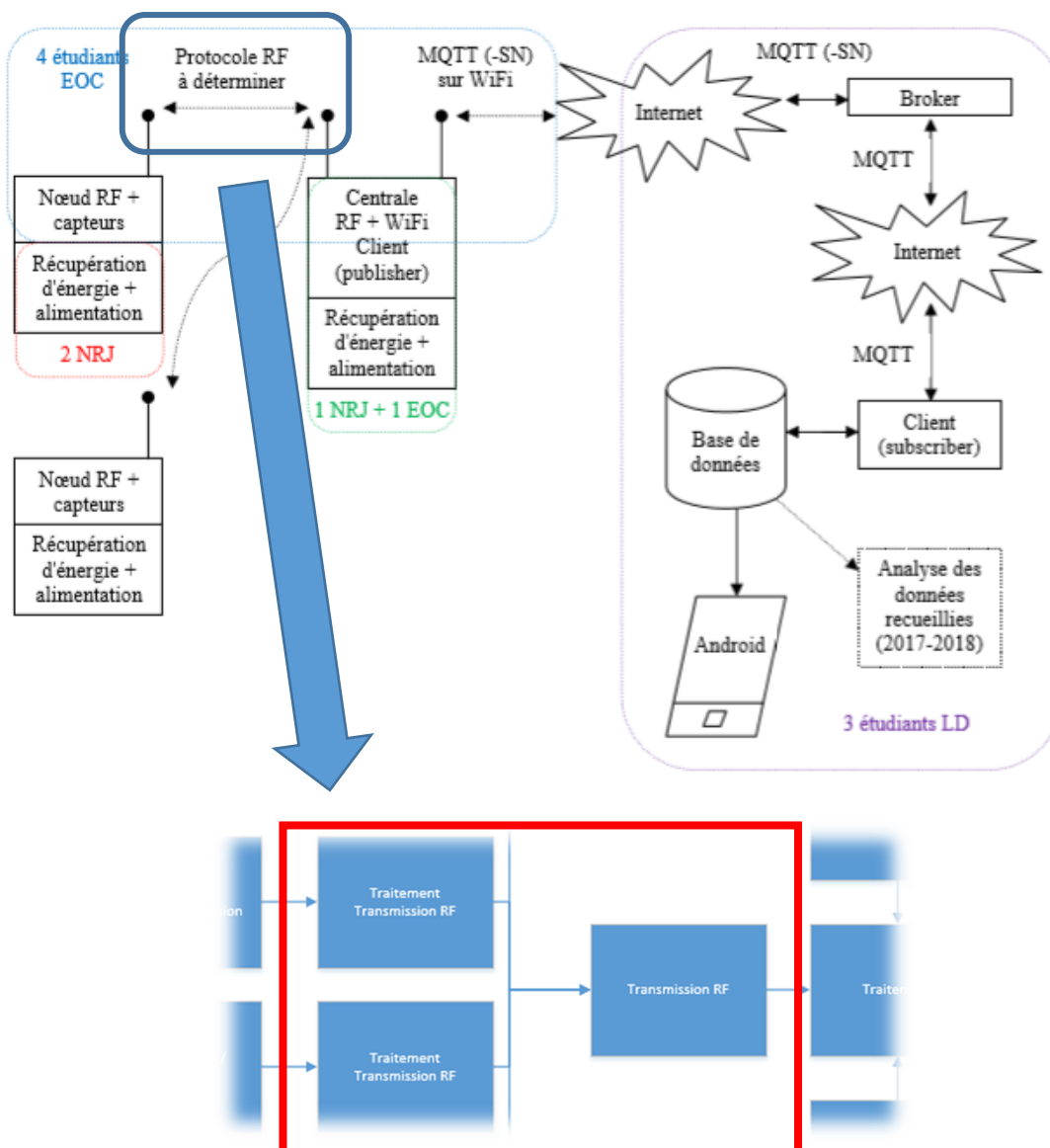


Du côté du microcontrôleur, on sera connecté à une entrée analogique et on viendra lire la tension avec l'ADC qui nous redonnera une valeur numérique. Une fonction permettra ensuite de donner la direction du vent en fonction de cette valeur ADC. Ces valeurs sont répertoriées dans le tableau ci-dessous.

Direction	Valeur de l'ADC
Nord	485
Nord Nord-Est	405
Nord-Est	460
Est Nord-Est	80
Est	97
Est Sud-Est	64
Sud-Est	185
Sud Sud-Est	123
Sud	284
Sud Sud-Ouest	245
Sud-Ouest	626
Ouest Sud-Ouest	598
Ouest	945
Ouest Nord-Ouest	821
Nord-Ouest	884
Nord Nord-Ouest	703

## Bloc Transmission RF

### Etude fonctionnelle



La transmission des données recueillies par les différents capteurs à la centrale est effectuée à l'aide du NRF51822, circuit qui possède une interface RF basse consommation à 2.4GHz. C'est également ce circuit qui réalise l'acquisition des différentes valeurs ainsi que la commande des signaux PWM nécessaires.

Afin de faire communiquer le module NRF51822 et le module NRF24L01, nous avons dû importer dans le SDK du NRF51822 une librairie spécifique permettant d'inclure le mode Enhanced ShockBurst (ESB). Ce protocole permet un transfert de données bidirectionnel automatique, et permet aussi de gérer la perte de paquets ainsi que leur retransmission si nécessaire.

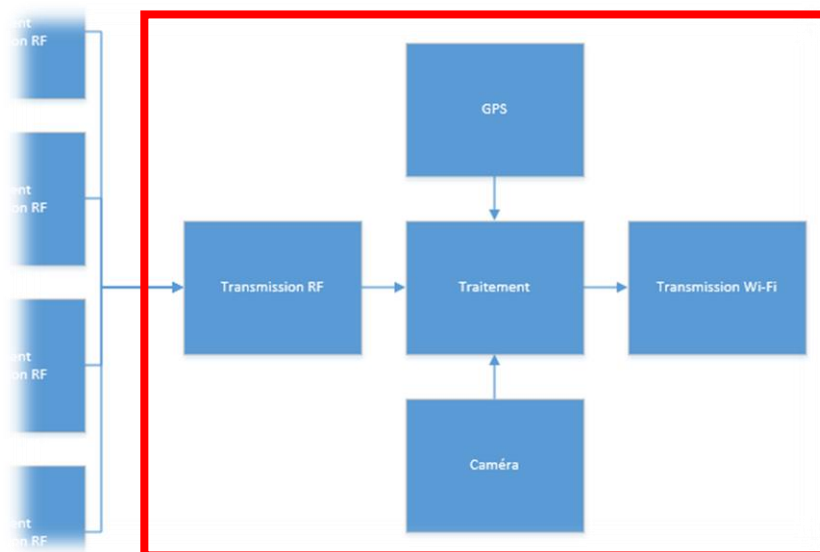
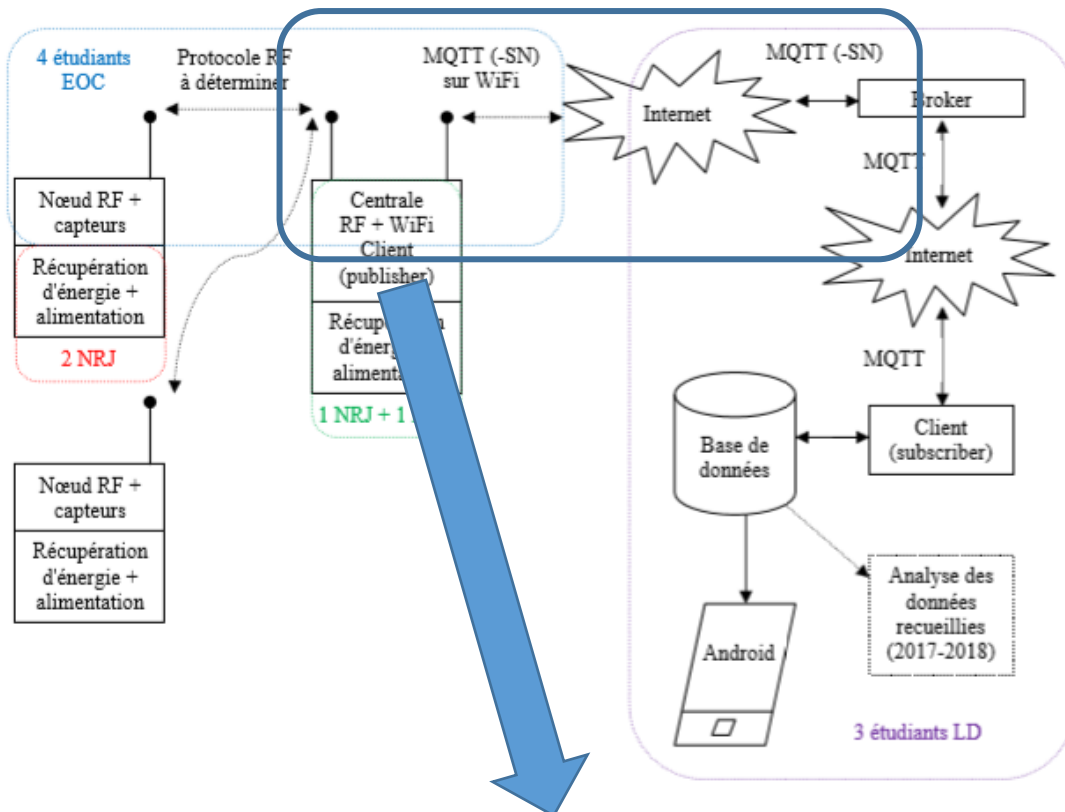
Plus de détails concernant l'intégration de la librairie ESB dans le SDK ainsi qu'un code exemple sont fournis dans le fichier `Integration_transmission_reception.txt` contenu dans le dossier ESB.

Au cours de nos recherches et de notre projet, nous avons remarqués qu'aucun tutoriel facile de compréhension n'existait quant à l'installation des différents drivers permettant de programmer le NRF51822 sous Eclipse. Il nous a aussi été difficile de créer de nouveau projets, c'est pourquoi nous avons faits nos propres tutoriels que nous incluons dans l'archive finale.

Pour installer tous les drivers du NRF51822 il faut suivre le tutoriel `Installation_drivers_NRF51822.txt` contenu dans le dossier `NRF_install`. Enfin le tutoriel permettant de créer un nouveau projet propre est contenu dans le fichier `Creation_nouveau_projet.txt`.

## Bloc Centrale

### Etude fonctionnelle



La carte centrale est constituée de 4 éléments :

- Un nRF24L01
- Une Arducam
- Un module GPS
- Une D1 mini faisant office de microcontrôleur

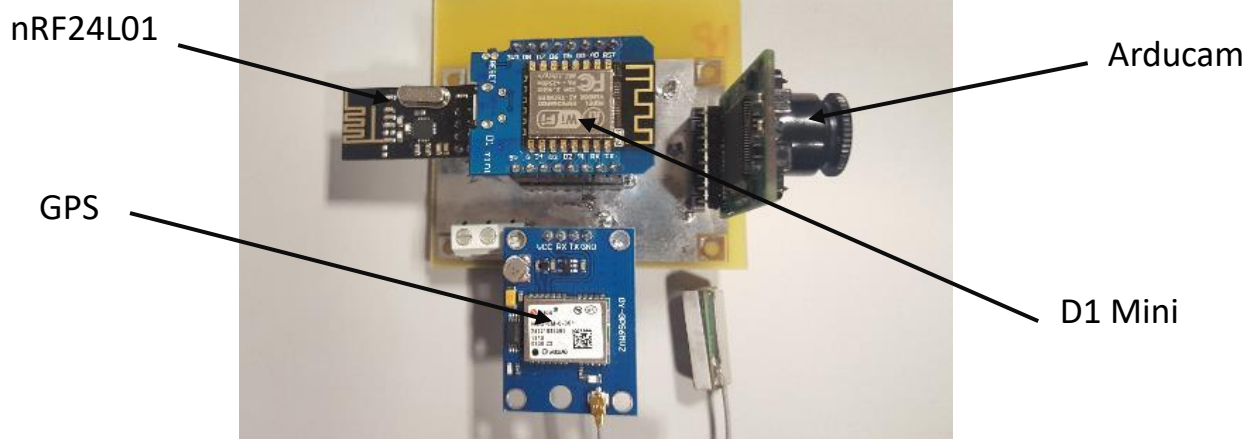


Figure 11: Carte Centrale

### Etude de consommation

Après avoir fait quelques recherches sur les composants, nous sommes arrivés à 2 solutions :

- Soit la D1 mini avec le nRF24L01
- Soit le nRF51822 et l'ESP01

Nom	Conso max (mA)	Conso min (µA)
D1 mini	Pic à 280	20
nRF24L01	19	2
nRF51822	17.4	4.2
ESP01	Pic à 200	10

Notre choix s'est porté sur ces composants car nous voulions consommer le moins possible tout en ayant un module WiFi (malheureusement énergivore) pour pouvoir envoyer les données via le WifiLib de la ville d'Angers. Nous avons choisi la D1 mini car elle comptait le bon nombre d'entrées/sorties dont on a besoin pour les différents modules. Sa mémoire flash de 4 Mo a également été un point positif car cela nous donne la possibilité d'enregistrer une photo sans avoir recours à une flash externe (demandant plus de PINs au microcontrôleur, et consommant souvent beaucoup plus).

Nous avons trouvé d'autres composants, notamment les modules de chez ATMEL, les ATWILC3000 et ATWILC1000, faisant à la fois BLE et Wifi, cependant la consommation de ces modules était vraiment très élevée, 300mA pour le Wifi et jusqu'à 100mA pour le BLE. Ce choix aurait été intéressant en cas d'utilisation de BLE en liaison RF.

Concernant l'alimentation de notre carte, elle se fait via la D1 mini lorsqu'elle est, pour les tests et le debug, branchée au PC ou via le bornier 3 broches, par du 5V.

Après le routage, des modifications ont été rajoutées. Des résistances de 2,2k ont été rajoutées pour la liaison SPI et une résistance de 10k a été rajoutée entre la pin D0 et le reset de la D1 (cette modification a pour but de permettre la veille prolongée. Une diode serait plus sûre, mais la résistance est suffisante d'après la communauté).

On trouve une liaison SPI entre la D1 mini et le nRFL2401 et la caméra. On configure la caméra au départ par I2C. Le module GPS est relié en UART à la D1.

#### nRF24L01

Ce module sert d'interface entre la D1 mini et les capteurs en mettant en place une liaison RF propriétaire de chez Nordic Semiconductor. Les 2 modules communiquent en SPI.

A l'heure actuelle, la liaison RF entre le nRF24L01 et les nRF51822 ne fonctionne pas. Ces derniers modules sont plus anciens que le nRF24L01 et des adaptations de code doivent être faites du côté nRF51822. Le problème peut venir d'ici, nous pouvons également supposer qu'au vu des problèmes que nous avons rencontrés avec Sming, celui-ci pourrait venir du code mal configuré du côté de la centrale. Le problème aurait pu venir du SPI mais les bits de statuts se mettent correctement à jour.

Pour la configuration du nRF24L01, il est nécessaire d'utiliser des adresses d'émission et de réception qu'il faut configurer avec les autres nRF avec qui il veut communiquer et de configurer les pins CE et CSN de façon logicielle.

```
10 //
11 **** NRF24L01 pins connection: ****
12 * VCC      3.3v
13 * SCK      GPIO14
14 * MOSI     GPIO13
15 * MISO     GPIO12
16 * CE       GPIO4 (changeable)
17 * CSN      GPIO5 (changeable)
18 * GND      GND
19 */
20 RF24 radio(0, 15); // Set up nRF24L01 radio on SPI bus plus pins 0 (CE) & 15 (CSN)
21
22 // Radio pipe addresses for the 2 nodes to communicate.
23 const uint64_t pipes[2] = { 0xF0F0F0F0E1LL, 0xF0F0F0F0D2LL };
```

La méthode « begin » du fichier RF24 permet quant à elle :

- de configurer le débit

```
661 | setDataRate( RF24_1MBPS );
```

- de configurer le canal

```
679 | setChannel(76);
```



## Arducam

Nous avons choisi d'utiliser un module OV2640 qui est une caméra 2MP suffisant pour notre application. Le but étant de prendre une photo 1 ou 2 fois par jour pour ensuite faire un timelapse. On configure la caméra via l'I2C et ensuite les photos sont transmises via le SPI.

```
8 ArduCamCommand::ArduCamCommand(ArduCAM *CAM) {
9     debugf("ArduCamCommand Instantiating");
10    myCAM = CAM;
11    imgSize = OV2640_320x240;
12    imgType = JPEG;
13 }
```

Un constructeur de la classe Arducam permet de spécifier la taille de l'image désirée, ainsi que le type, JPEG ou BMP.

Pour pouvoir récupérer l'image de l'Arducam, il faut que la D1 mini soit connectée au WiFi, typiquement le WiFi IoT de l'ESEO. On la récupère via un serveur WEB en appuyant sur le bouton d'acquisition. Cela est plus simple au début pour les tests de la caméra. Par la suite, la caméra prendra la photo toute seule et elle sera enregistré dans la flash.

## GPS

Il s'agit d'un module GPS NEO-6M. Il est peu cher mais peu précis surtout en dynamique. Nous l'utilisons en statique, cela ne pose pas de problèmes, le but étant d'avoir une idée de la localisation. L'UART utilisé pour le GPS est également la liaison UART utilisée pour le debug. Pour faire fonctionner le GPS, il faut donc flasher la D1 mini, puis la débrancher du PC et enfin brancher le module.

Il est possible de paramétrer, ou debugger, le module GPS à l'aide du logiciel du constructeur **u-center\_v8.23**. Il suffit juste d'utiliser une perf UART (les problèmes de compatibilités avec Windows 8 ou 10 ont été résolu avec le [driver non officiel](#) de la perf).

## Software centrale

Nous développons ici plus en détail les spécificités du code.

Nous utilisons donc le SDK « Sming » pour ESP8266 développé par la communauté sur GitHub. L'idée est de pouvoir utiliser la syntaxe Arduino avec l'ESP8266. Le nombre d'exemples fonctionnant sous Arduino étant conséquent, cela devait nous faire gagner un temps précieux dans le développement. Cependant, Sming est un outil en développement constant, et l'on n'est pas assuré que tous ce qui a été développé fonctionne correctement, ou que les exemples donnés sont encore à jour.

Il s'agit donc de programmation en C++, *avec certainement un RTOS*. L'**initialisation** se fait à travers une fonction init(), changeable dans SmingFramework/appinit/user\_main.cpp (cf code ci-dessous).

```
extern "C" void __attribute__((weak)) user_init(void)
{
    system_timer_reinit();
    uart_div_modify(UART_ID_0, UART_CLK_FREQ / 115200);
    cpp_core_initialize();
    System.initialize();
#ifdef ENABLE_GDB
    gdbstub_init();
#endif
    init(); // User code init
}
```

Il est donc possible de changer ce nom à cet endroit. Nous avons ensuite décidé de créer une boucle nommée `loop()`, qui est appelée après un certain temps, décidé au préalable avec la fonction : `procTimer.initializeUs(1000*1000,loop).start();`

Cette fonction de l'objet `Timer` est appelée dans l'initialisation, et appelle la fonction `loop()` toute les ms dans cette configuration. Cette fonction est importante, car un watchdog est présent, et n'hésite pas à faire un reset en cas de présence d'une fonction qui prend trop de temps.

L'un des intérêts d'utiliser l'ESP8266 est la présence d'une **veille prolongée** « deepsleep », ne consommant presque rien (de l'ordre de quelques  $\mu A$  d'après la datasheet, nous n'avons pas fait de mesures en réel). Le principe de cette veille est d'éteindre l'ensemble du cœur, à l'exception d'un horloge interne, qui déclenche une interruption en sortie à la fin d'un compte à rebours défini au préalable. Le but de cette interruption est de déclencher un *reset* du processeur (d'où notre ajout d'une résistance entre D0 et le reset).

Il s'agit donc d'une veille plus bricolée qu'autre chose : ainsi, le programme reprend du début. C'est à prendre en compte dans l'écriture du code (garder les valeurs de variables dans la mémoire flash et non pas la RAM, être capable de savoir s'il s'agit d'une première mise en marche ou d'un retour de veille).

Dernière chose sur cette veille : elle ne se déclenche pas immédiatement après l'appel de la fonction

```
system_deep_sleep();
```

Mais dans l'état IDLE de l'OS embarqué (lorsqu'aucune fonction n'est en train d'être exécuté, donc par exemple lorsque `procTimer.initializeUs()` ne fait plus que compter).

Nous avons fonctionné selon la **machine à état** [en Annexes](#).

Nous avons choisi de ne pas mettre d'IHM de contrôle pour la centrale, afin de réduire la consommation. Ainsi, l'initialisation de la centrale se fait automatiquement.

On attend donc que la position de la centrale, donné par le module GPS (message sous la forme « `hh:mm:ss $GPGLL,lat,(NS),lon,(EW),hhmmss.ss,(AV),x*75` »), soit valide. La date devait à l'origine être récupéré avec le GPS (message `$GPZDA`), mais cette solution était peu fiable car la RTC interne à l'ESP n'était pas utilisable. Nous sommes finalement partis sur un système de récupération ponctuelle de la date sur le web (explications ci-après). Comme dit précédemment, la récupération de la position par UART pose problème si l'USB est branché, ainsi que si le module n'est pas dans un endroit correctement dégagé.

Pour la suite, on demande successivement à chaque capteur de nous donner les valeurs mesurées. On garde en mémoire la présence d'une réponse après un certain timeout. Les conditions de changement d'états sont encore à définir dans le code, après un accord sur les messages qui seront échangés entre les capteurs et la centrale.

Autre point, la caméra ne prenant pas de photos à chaque fois, on peut passer l'état « prise de photo », selon l'heure.

On entre par la suite dans l'état d'envoi de données. Il s'agit d'une publication « publish » sur un *broker* MQTT. A priori, c'est dans cet état que l'on devrait se connecter au WiFi. La connexion sur le WiFi de WiFiLib n'a pas encore été prévu. Nous n'avons pas reçu de réponses à nos questions par l'entreprise sur la manière dont serait faite la connexion. En effet, le principe de WiFiLib est de fournir gratuitement du WiFi, en échange de la visualisation de pub à la connexion (donc peu approprié à notre utilisation).

On n'envoie que les données dont nous avons reçu une réponse (inutile de renvoyer les anciennes données, il est préférable de ne rien envoyer). Le *topic* contenant le message se fait automatiquement, d'après la forme définie dans la fonction `majPath(char* capteur)` :

```
void majPath(char* capteur)
{
    sprintf(cheminMQTT, "PAVIC/%d/%d/%d/%d/%d/%d/%d/%d/%s",
lat,lon,annee,mois,jour,heure,minute,seconde,capteur);
}
```

On vérifie ensuite l'heure actuelle en envoyant une requête http vers un site quelconque, en vérifiant au préalable qu'il nous renvoie bien l'heure actualisée à chaque fois. L'entête de la réponse est censé contenir l'heure (en réalité, ce n'est pas toujours le cas). Sming s'occupe automatiquement de remplir certain champ dans sa classe clientHttp, il suffit juste de récupérer les informations voulus.

A partir de ces informations, on décide immédiatement si la prochaine fois on prend une photo ou non, puis combien de temps nous attendons en mode deepsleep (actuellement, nous nous réveillons toutes les demi-heures, sans exceptions). La veille durera donc moins de 30 minutes.

Rétrospectivement, nous aurions fait certaines choses autrement. Pour améliorer la centrale nous conseillerons plutôt :

- De créer un module IHM à connecter en UART à la centrale. On pourrait grâce à ce module entrer les informations comme la position géographique (obtenu à l'aide d'appareils plus fiable que le NEO-6M), ou simplement un nom de quartier, de rue, etc.
- Récupérer l'heure à l'aide d'une demande à un serveur de temps, plus propre qu'une requête http (et de nombreux exemples sur Arduino existent)
- Choisir des modules de communications RF similaire sur la centrale et les capteurs
- Le chemin MQTT serait à revoir (ordre des sous-topic)
- Une solution à partir de STM32L series en maître avec un module WiFi ESP-01 pourrait ne pas consommer beaucoup plus, et l'on rencontrerait moins de problèmes de développement, car plateforme de développement mieux connue à l'ESEO. Cependant l'avantage actuel de la solution est que les modules de chez Nordic Semiconductor sont peu chers contrairement aux STM qui seront un peu plus chers.

## Sources

Ressources sur les gaz et polluants présents dans l'air :

[www.cespharm.fr](http://www.cespharm.fr)

<http://www.airpaca.org/article/polluants-surveilles>

<http://www.ecologs.org/air/pollution-atmospherique-les-principaux-polluants.html#31>

<http://www.airparif.asso.fr/reglementation/normes-francaises>

Etude du MQ7 :

<http://thesis.jmsaavedra.com/prototypes/technology/mq-7-carbon-monoxide-sensor-breakout/>

Etude du MG811 :

<http://sandboxelectronics.com/?p=147>

Etude du capteur de poussières SHARP GP2Y1010AU :

<http://lantaukwcounter.blogspot.fr/2016/01/shinyei-and-sharp-dust-sensors-looking.html>

Etude de la consommation des capteurs :

<http://www.nordicsemi.com/eng/Products/2.4GHz-RF/nRF24L01>

<https://www.nordicsemi.com/eng/Products/Bluetooth-low-energy/nRF51822>

<https://espressif.com>

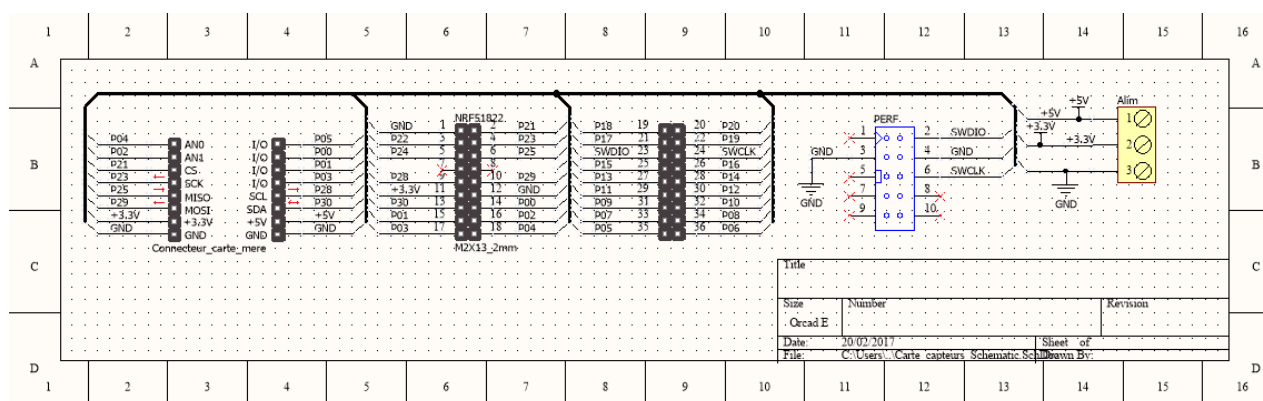
## Annexes

### Table des Figures

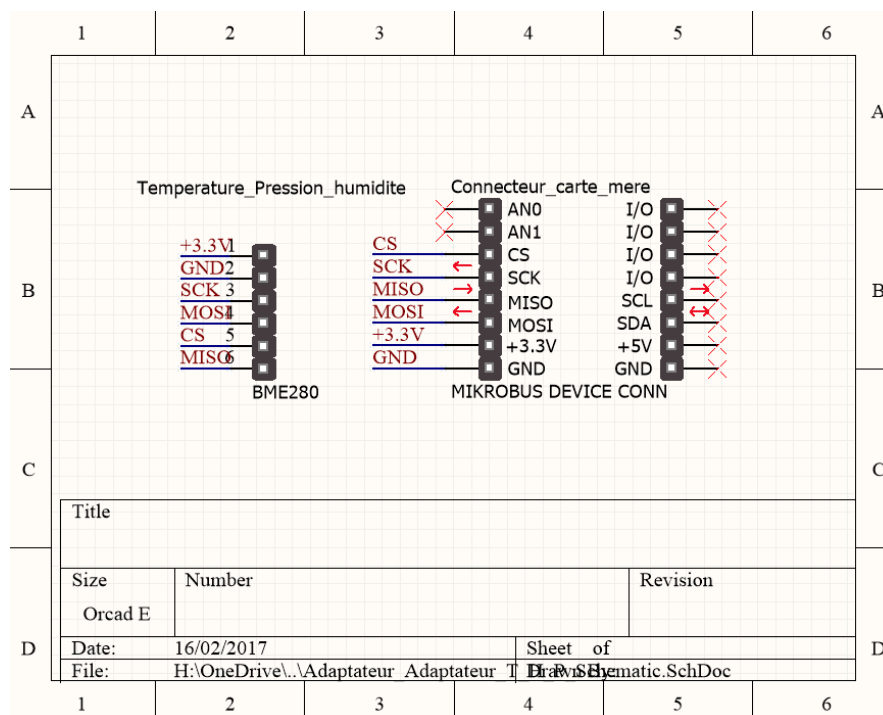
Figure 1: Evolution de la résistivité du capteur selon le taux d'Ozone présent dans l'air .....	10
Figure 2: Structure électrique du capteur reprise dans le schéma de la carte .....	10
Figure 3: Conditionnement de l'alimentation du MQ7 .....	11
Figure 4: Structure de commande de l'alimentation du MQ7 .....	11
Figure 5: Sensibilité des différentes couches du capteur.....	12
Figure 6: Structure recommandée pour le MiCS 6814.....	12
Figure 7 : Structure du capteur de CO <sub>2</sub> .....	13
Figure 8: Principe de fonctionnement du capteur de poussières .....	14
Figure 9: PWM à générer afin de commander le capteur.....	14
Figure 10: Evolution de la tension de sortie en fonction du taux de poussières .....	14
Figure 11: Carte Centrale .....	22
Figure 12: Infographie comparative sur la taille des différentes particules fines.....	33
Figure 13 : Machine à état centrale .....	34
Figure 14 Machine à état capteurs* .....	35

### Schémas électriques

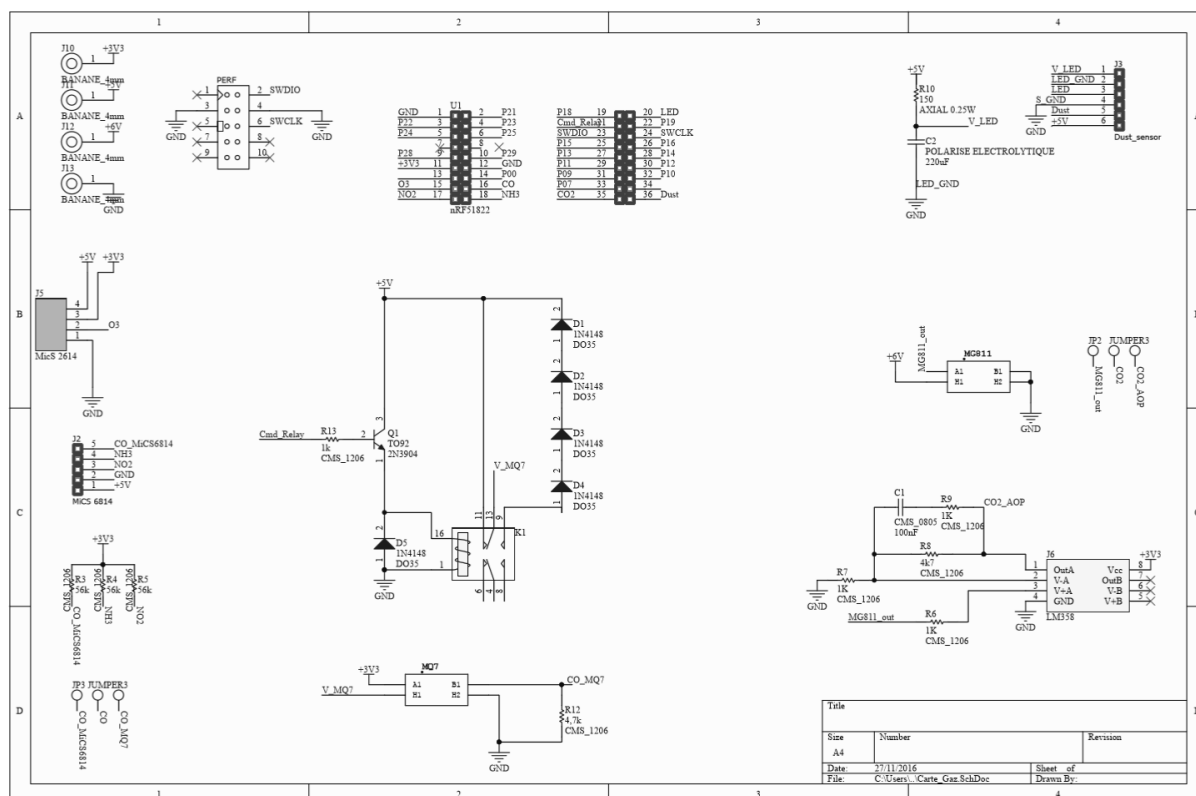
#### Schéma électrique de la carte Capteurs



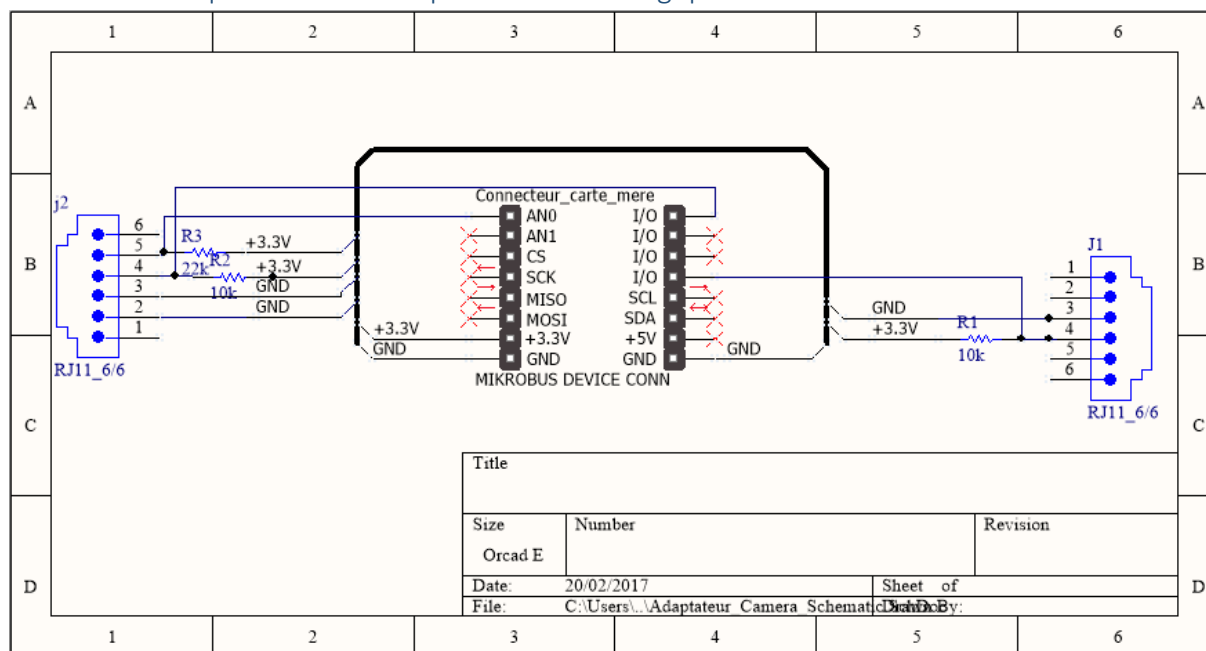
## Schéma électrique de la carte Capteurs Température/Pression



## Schéma électrique de la carte Capteurs Gaz/Poussières



## Schéma électrique de la carte Capteurs Météorologiques



## Etudes des gaz et poussières mesurés

### O<sub>3</sub> - Ozone

#### *Où le trouve-t-on ?*

L'ozone n'est pas directement rejeté par une source de pollution, mais se forme par une réaction chimique réalisée par les rayons UV, à partir de polluants comme les oxydes d'azote NO<sub>x</sub> et les Composés Organiques Volatils (COV).

#### *Effets sur la santé*

Augmente le seuil de sensibilité bronchique aux pollens notamment, peut provoquer toux, inconfort thoracique, essoufflement, irritations nasale et oculaire.

#### *Effets sur l'environnement*

L'ozone a des effets néfastes sur la végétation et perturbe la croissance de certaines espèces. Il entraîne une baisse des rendements des cultures, et participe au phénomène des pluies acides et à l'effet de serre. Enfin, il dégrade certains matériaux (caoutchouc par exemple).

### CO - Monoxyde de Carbone

#### *Où le trouve-t-on ?*

Gaz toxique, incolore et inodore, présent surtout en intérieur dans les pièces ayant un chauffage mal entretenu (résulte d'une mauvaise combustion). Il se diffuse très vite dans l'environnement.

#### *Effets sur la santé*

Responsable de nombreuses morts accidentelles (200 cas/an en France) lors d'intoxications aiguës, il agit comme un gaz asphyxiant qui se fixe très rapidement sur l'hémoglobine. L'intoxication à faible dose possède des effets peu spécifiques (maux de tête, fatigue, vertiges, troubles digestifs), alors que les enfants et personnes âgées sont les plus sensibles au CO, tout comme les insuffisants respiratoires et coronariens.

#### *Effets sur l'environnement*

Participe au mécanisme de production de l'ozone troposphérique et contribue à l'effet de serre en se transformant en dioxyde de carbone (CO<sub>2</sub>).

### NO<sub>2</sub> - Dioxyde d'azote

#### *Où le trouve-t-on ?*

On le trouve surtout dans les foyers, car il est émis par les cuisinières, radiateurs et poêles à gaz, les cheminées à foyer ouvert et la fumée de tabac. Mais il est également émis lors de la combustion à haute température de carburants et combustibles fossiles, et donc présent dans les échappements des voitures. Le pot catalytique a permis une diminution des émissions des véhicules à essence depuis 1993, mais l'effet reste encore peu perceptible du fait de l'augmentation du parc automobile (alors que les véhicules diesel davantage de NO<sub>x</sub>...).

#### *Effets sur la santé*

Irritation des yeux et voies respiratoires, augmentation de la sensibilité des bronches et aggravation de l'asthme chez les asthmatiques. Chez l'enfant, il favorise les infections pulmonaires. Pas de risque cancérigène lié à l'exposition au dioxyde d'azote.

#### *Effets sur l'environnement*

Le NO<sub>2</sub> participe aux phénomènes de pluies acides, ainsi qu'à la formation de l'ozone troposphérique, la dégradation de la couche d'ozone et à l'effet de serre.



## NH<sub>3</sub> - Ammoniac

### *Où le trouve-t-on ?*

L'ammoniac (NH<sub>3</sub>) est un gaz incolore à odeur piquante, plus léger que l'air. Il se liquéfie facilement. Soluble dans l'eau, il produit l'ammoniaque. Sous forme gazeuse, l'ammoniac est utilisé par l'industrie pour la fabrication d'engrais, d'explosifs et de polymères. Il est principalement émis dans l'atmosphère des activités industrielles et la fertilisation agricoles avec l'application des engrais.

### *Effets sur la santé*

À l'état gazeux comme à l'état liquide, cette substance toxique et irritante provoque des douleurs et des brûlures.

## CO<sub>2</sub> - Dioxyde de Carbone

### *Où le trouve-t-on ?*

Le dioxyde de carbone (CO<sub>2</sub>) un gaz à effet de serre émis par la combustion de produit organique et des énergies fossiles. Dans les conditions normales de température et de pression le dioxyde de carbone est un gaz incolore, inodore et à la saveur piquante communément appelée gaz carbonique ou encore anhydride carbonique.

Il est présent dans l'atmosphère dans une proportion approximativement égale à 0,0375 % en volume, dans cette décennie (années 2000), soit 375 ppmv (parties par million en volume). Mais elle augmente rapidement, d'environ 2 ppmv/an, de par les activités humaines de consommation des combustibles fossiles : charbon, pétrole, gaz.

### *Effets sur la santé*

Notre système respiratoire et circulatoire est très sensible au CO<sub>2</sub> : Une augmentation minime de la concentration en CO<sub>2</sub> de l'air inspiré accélère quasi-immédiatement le débit respiratoire qui est normalement de 7 litres/minute (sous 0,03% de CO<sub>2</sub> dans l'air inspiré), et qui passe à 26 litres/minutes (pour 5% de CO<sub>2</sub> dans l'air inspiré).

Le CO<sub>2</sub> est un toxique pour l'homme à des doses relativement faibles et pour des temps d'exposition de quelques minutes seulement : 1000 ppm (0,1%) est la valeur maximale admise pour le dimensionnement des systèmes de conditionnement de l'air, à l'intérieur des bâtiments et maisons d'habitation :

- À partir de 0,1 %, (1000 ppm), le CO<sub>2</sub> devient un des facteurs d'asthme ou du syndrome des bâtiments.
- Au-dessus de 0,5% (5000 ppm), la valeur maximale d'exposition professionnelle retenue dans la plupart des pays, et la valeur maximale admise pour le dimensionnement des appareillages d'air conditionné dans les avions sont dépassées.
- trois fois ce taux (1,5%, ou 15000 ppm) est la valeur maximale d'exposition professionnelle sur une durée maximale de 10 minutes.
- Au-delà de 4% de CO<sub>2</sub> dans l'air (40000 ppm) le seuil des effets irréversibles sur la santé est atteint (c'est le seuil qui justifie une évacuation immédiate de locaux). À partir de 10% et d'une exposition dépassant 10 minutes, sans une action médicale de réanimation, c'est la mort.

## PM10 - Matières Particulaires <10µm

### Où le trouve-t-on ?

La matière particulaire (MP) est émise par la fumée et les poussières. Elle peut contenir différents types de contaminants (fumée de tabac, fibres, spores de moisissures, bio contaminants, allergènes, produits de combustion, poussières naturelles, pollens, moisissures, fumées, particules fines des gaz d'échappement automobiles...) ayant chacun une action potentielle spécifique sur la santé.

### Effets sur la santé

La fraction PM10 désigne les particules en suspension de 10 micromètres de diamètre et moins capable d'entrer dans la cavité nasale. PM 2.5 désigne les particules de 2,5 micromètres de diamètre et moins capable de pénétrer les bronches et les poumons. Ainsi les effets des particules dépendent de leur taille. Elles sont mises en cause dans l'apparition de pathologies respiratoires à court terme (inflammation, allergies respiratoires, asthme) ou long terme (cancer, broncho-pathologie chronique obstructive - BPCO). Les particules aériennes peuvent véhiculer des bactéries et des virus et favoriser la dissémination des maladies infectieuses. Par ailleurs, une interaction entre les particules et les allergènes peut survenir et amplifier la réaction allergique.

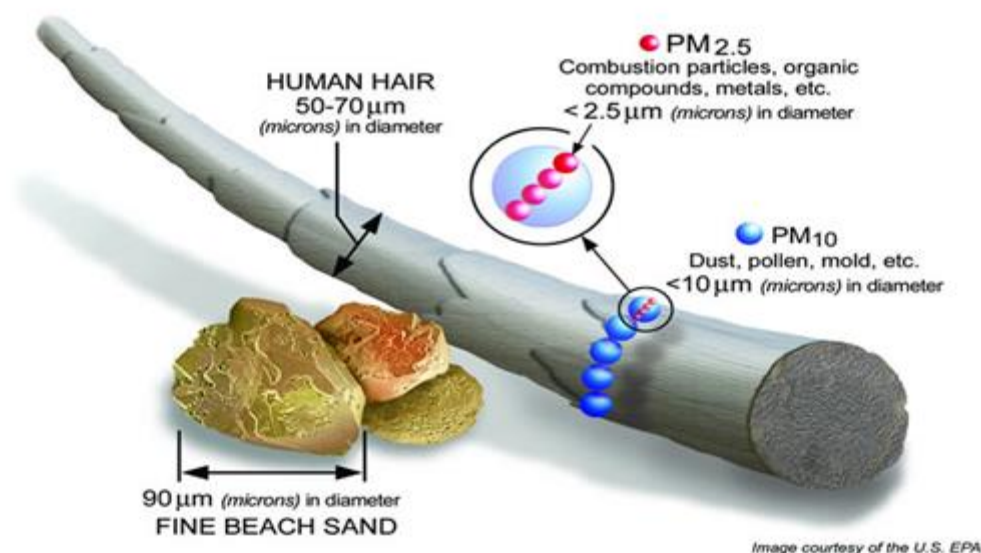


Figure 12: Infographie comparative sur la taille des différentes particules fines

Seuils sanitaires limites selon le Code de l'environnement en vigueur en France

Polluants	Seuil horaire	Seuil journalier
Ozone (O <sup>3</sup> )	180µg/m <sup>3</sup>	120µg/m <sup>3</sup>
Monoxyde de carbone (CO)		10 000 µg/m <sup>3</sup>
Dioxyde d'azote (NO <sub>2</sub> )	200µg/m <sup>3</sup>	40µg/m <sup>3</sup>
Ammoniac (NH <sup>3</sup> )		35 mg/m <sup>3</sup>
Dioxyde de carbone (CO <sub>2</sub> )		5 000 ppm
Particules fines PM10		80µg/m <sup>3</sup>

## Machine à état (Centrale)

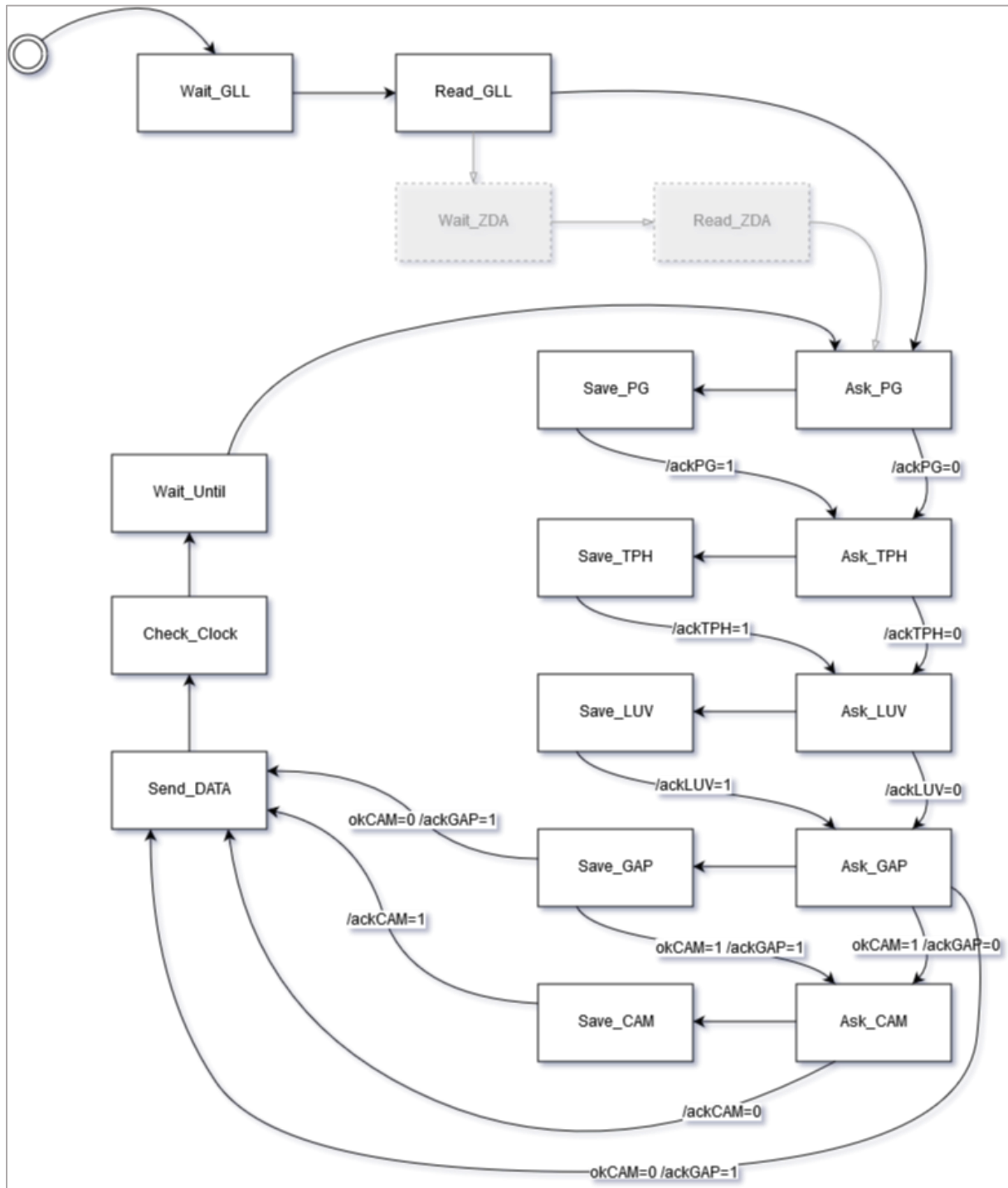


Figure 13 : Machine à état centrale

## Machine à état (Capteurs)

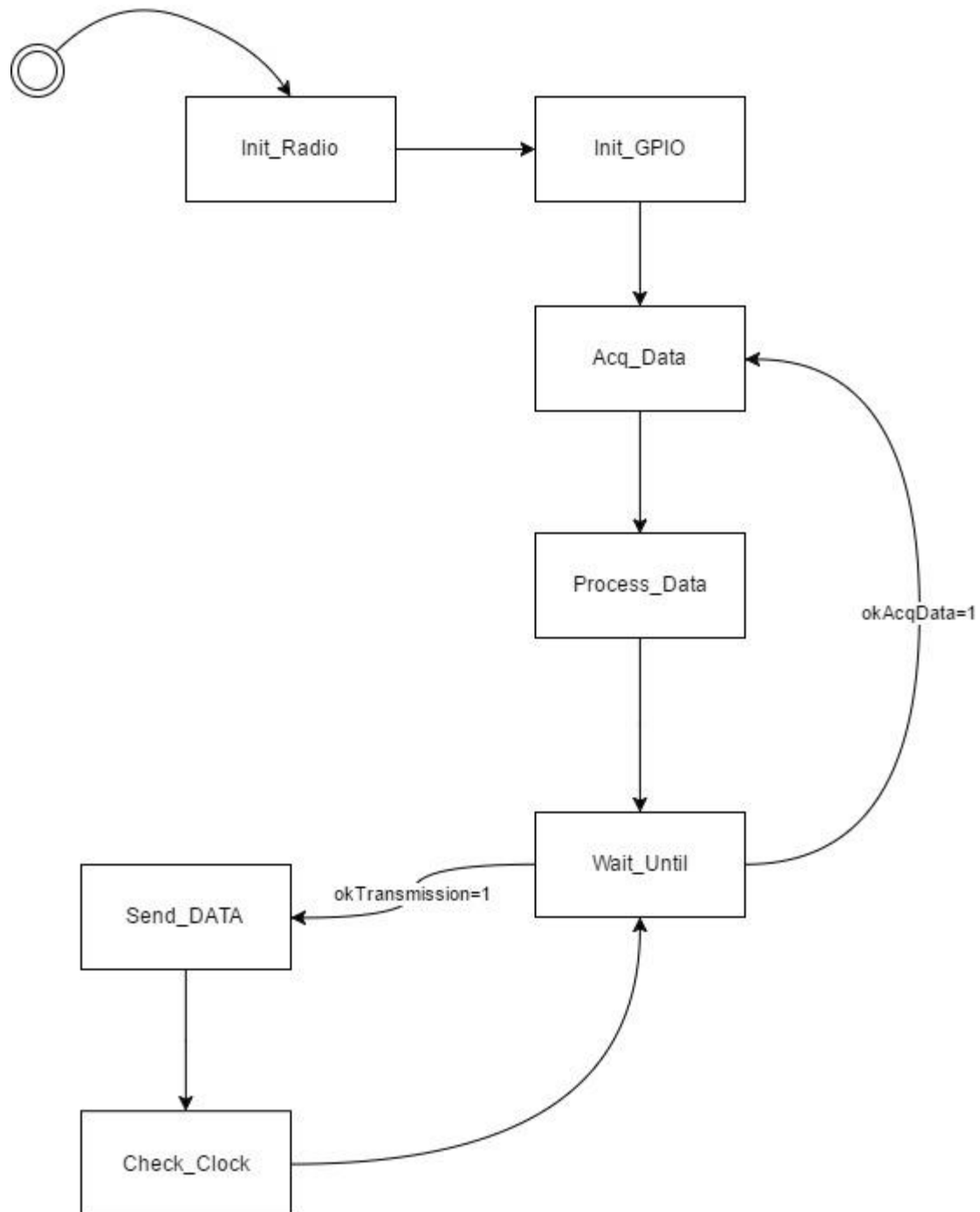


Figure 14 Machine à état capteurs\*

\*Cette machine à état peut varier en fonction du capteur mais sa structure reste semblable