



UNIVERSITE D'ANTANANARIVO
ECOLE SUPERIEURE POLYTECHNIQUE
D'ANTANANARIVO



MEMOIRE DE FIN D'ETUDES EN VUE DE L'OBTENTION DU DIPLOME DE MASTER

Titre : INGENIEUR

Mention: ELECTRONIQUE

Parcours: INFORMATIQUE APPLIQUEE

**SYSTEME DE DETECTION DE LA POLLUTION DE L'AIR
INTEGRANT LE MODELE INTERNET OF THINGS**

Présenté par :

Mr RANDRIANASOLO JerimanjakaTsiorisoa

N° d'ordre : 005/EN/IA/15

Année universitaire : 2014-2015

MEMOIRE DE FIN D'ETUDES EN VUE DE L'OBTENTION DU DIPLOME DE MASTER

Titre : INGENIEUR

Mention: ELECTRONIQUE

Parcours : INFORMATIQUE APPLIQUEE

**SYSTEME DE DETECTION DE LA POLLUTION DE L'AIR
INTEGRANT LE MODELE INTERNET OF THINGS**

Présenté par :

Mr RANDRIANASOLO JerimanjakaTsiorisoa

Soutenu le : 28 Septembre 2016

Devant le Jury composé de :

- Mr RASTEFANO Elysée ,Président
- Mme RABEHERIMANANA Lyliane Irène , Examineur
- Mr RATSIMBAZAFY Guy Prédon Claude , Examineur
- Mr ANDRIAMANANTSOA Guy Danielson , Examineur

Rapporteur: Mme RAMANANTSIHOARANA Harisoa Nathalie

REMERCIEMENTS

Nous tenons à remercier, en premier lieu, Dieu qui nous a donné sa bénédiction pour nos cinq ans d'étude et ce présent mémoire. Sans Lui nous n'avons pas eu la force et le courage pour la réalisation de ce travail.

Nous tenons également à exprimer notre reconnaissance et nos vifs remerciements à tous ceux qui ont apporté leur contribution dans la réalisation de ce mémoire, en particulier :

Monsieur ANDRIAMANANTSOA Guy Danielson, responsable de la mention électronique et membre de jury.

Monsieur RASTEFANO Elisée, qui a voulu présider la soutenance de ce mémoire malgré ses innombrables occupations.

Les membres de jury :

- Madame RABEHERIMANANA Lyliane Irène
- Monsieur RATSIMBAZAFY Guy Prédon Claude

Madame RAMANANTSIHOARANA Harisoa Nathalie, notre encadreur qui nous a partagé ses connaissances sur le sujet en question. Ses conseils et ses orientations ont été d'une grande aide et nous ont permis de mener à bien nos travaux.

Les enseignants de la Mention Electronique pour les connaissances et les formations qu'ils nous ont prodigué durant ces cinq dernières années.

A nos familles pour les encouragements, moralement ou financièrement ; et à nos amis pour leurs aides de toutes sortes.

RESUME

Ce travail de mémoire présente la conception et la réalisation d'un système de détection de pollution de l'air destiné à une ville comme Antananarivo. Le système suit le modèle « Internet of things (IoT) ». Il est capable de collecter les valeurs des gaz toxiques, de transmettre ces valeurs vers un centre de données en ligne et il est interfacé par un portail web pour les utilisateurs finaux.

La réalisation fait intervenir deux démarches en parallèle: un développement matériel et un développement logiciel. D'une part, le développement matériel a été réalisé par la conception d'un capteur avec wifi qui permet de collecter les données utiles et assure l'envoi vers la base de données en ligne. D'autre part, une application web a été développée avec le langage JavaEE. C'est l'interface utilisateur.

TABLE DES MATIERES

REMERCIEMENTS	i
RESUME.....	ii
TABLE DE MATIERE	iii
LISTE DES ABREVIATIONS	vii
LISTE DES FIGURES	ix
LISTE DES TABLEAUX.....	x
INTRODUCTION GENERALE.....	1
chapitre 1: GENERALITE SUR LA POLLUTION ATMOSPHERIQUE.....	2
1.1. L'atmosphère et l'air	2
1.1.1. Atmosphère terrestre	2
<i>a. La troposphère</i>	2
<i>b. La stratosphère</i>	2
<i>c. La mésosphère</i>	3
<i>d. La thermosphère</i>	3
<i>e. L'ionosphère</i>	3
1.1.2. L'Air.....	4
<i>a. La Composition chimique de l'air</i>	4
<i>b. Rôle de l'air</i>	4
1.1.3. La pollution de l'air.....	5
<i>a. Les principaux polluants et leurs origines</i>	6
<i>b. Les effets de la pollution de l'air</i>	8
1.2. Le capteur [5].....	9
1.2.1. Définition d'un capteur	9
1.2.2. Les caractéristiques d'un capteur	9
1.2.3. Classification des capteurs	10
<i>a. Capteurs actifs</i>	10
<i>b. Capteurs passifs</i>	10
1.2.4. Type de sortie de capteur	10
<i>a. Les Capteurs analogiques</i>	10
<i>b. Capteurs logiques ou capteurs TOR</i>	11
1.2.5. Les différents types de capteur de l'air	11
1.3. Présentation d'un cas existant de gestion de l'air.....	12

1.3.1	Le Programme de gestion de l'air à Ambatovy [6]	12
a.	<i>Problématique</i>	12
b.	<i>Mise en place du système</i>	13
1.4.	Conclusion	13
chapitre 2: LA PRESENTATION DE L'INTERNET DES OBJETS		14
2.1.	Principe de fonctionnement	14
2.2.	Le réseau des capteurs(WSN)	15
2.2.1.	Principe de fonctionnement de WSN	16
2.2.2.	Application des réseaux de capteur	16
2.2.3.	Les Options de communication pour le WSN	17
2.2.4.	Les différents types de technologie sans fil	17
a.	Bluetooth	17
b.	Zigbee	17
c.	<i>Le wireless Fidelity (WIFI)</i>	18
d.	Global System for Mobile communications(GSM)	18
e.	Les Réseaux satellites	18
2.3.	Les plateformes de service	19
2.3.1	Le «Cloud Computing»	19
a.	<i>Définition</i>	20
b.	<i>Principe de fonctionnement</i>	20
2.3.2	Les services Cloud	21
a.	<i>Le logiciel en tant que service :SaaS</i>	21
b.	<i>La plateforme en tant que service : PaaS</i>	22
c.	<i>L'infrastructure en tant que service :IaaS</i>	24
2.4.	L'interface utilisateur pour l'ioT	24
2.4.1	Application mobile	24
2.4.2	Interface Web	25
2.5.	Exemples d'objets connectés	25
2.5.1	Diagnoplant	25
2.5.2	iCropTrak	26
2.5.3	Bioconnect	26
2.6.	Conclusion	26
chapitre 3: PRESENTATION DU PROJET : « Système de détection de pollution de l'air »		27

3.1	Introduction	27
3.2	La partie matérielle	28
3.2.1	Le microcontrôleur Arduino.....	28
3.2.2	Le Capteur de pollution de l'air	29
	<i>a. MQ-135 Gaz sensor</i>	29
	<i>b. Sensibilité de MQ-135</i>	29
	<i>c. Calibration de MQ135</i>	31
3.2.3	L'envoi de données	31
	<i>a. Le principe de socket</i>	32
	<i>b. Le protocole HTTP</i>	32
	<i>c. Le module ESP8266</i>	32
3.3	La partie informatique du système	34
3.3.1.	Le langage de programmation Java EE.....	34
	<i>a. HTTP</i>	35
	<i>b. Enterprise Java Beans (EJB)</i>	35
	<i>c. Servlet</i>	35
	<i>d. java Server Page (JSP)</i>	36
3.3.2.	Le serveur web	36
3.3.3.	Le service Cloud : Google App Engine	36
	<i>a. Fonctionnement du Google App Engine</i>	36
	<i>b. Outils de développement</i>	37
3.3.4.	Le service de stockage utilisé.....	38
	<i>a. La base de données NoSQL</i>	38
	<i>b. le Datastore et le détail de la conception de base de données</i>	39
3.3.5.	Le frontal web	43
3.4	Conclusion.....	44
chapitre 4: REALISATION DU « SYSTEME DE DETECTION DE LA POLLUTION DE L'AIR »		45
4.1	Structure détaillé du projet	45
4.2	La partie électronique du projet.....	46
4.2.1	Les capteurs.....	46
4.2.2	Structure du montage	47
4.2.3	Le Firmware (Le programme à introduire dans l'arduino)	48
4.3	La partie informatique	49

4.3.1.	Accueil	49
4.3.2.	Archive des données.....	50
4.3.3.	Le Notification	51
4.3.4.	Statistique	52
	<i>a.</i> Calcul du pourcentage de gaz non acceptable.....	52
	<i>b.</i> Calcule de pourcentage d'un gaz	52
4.4	Conclusion	53
	CONCLUSION GENERALE	54
	ANNEXE I. CARACTERISTIQUE DES TECHNOLOGIES SANS FIL EXISTANT	55
	ANNEXE II. DATASHEET MQ-135	56
	ANNEXE III. LISTE DES COMANDES AT	59
	ANNEXE IV. LES FONCTIONS IMPORTANTS DE L'ARDUINO.....	63
	ANNEXE V. QUELQUE FONCTION EXISTANT DANS L'APPLICATION WEB	65
	REFERENCES.....	68

LISTE DES ABREVIATIONS

CFC	Chlorofluorocarbures
COV	Composés Organiques Volatils
CSS3	Cascading Style Sheets 3
DEL	diode électroluminescente
FN	fumées noires
GEOS	Geostationary Earth Orbital Satellite
GND	ground
GPS	Global Position System
GPRS	General Packet Radio Service
GSM	Global System for Mobile communications
HAP	Hydrocarbures Aromatiques Polycycliques
HRD	High Replication Datastore
HTML5	Hypertext Markup Language 5
HTTP	Hypertext Transfer Protocol
IaaS	infrastructure as a service
IDE	Integrated Development Environment
IEEE	Institute of Electrical and Electronics Engineers
INRA	Institut national de la recherche agronomique
Ios	iPhone OS
IoT	internet of things
Java EE	Java Enterprise Edition
LEOS	Low Earth Orbital Satellite
NoSQL	Not only SQL
OMS	Organisation Mondiale de la santé
ONE	Office National pour l'Environnement
OS	Operating System
OSI	Open System Interconnexion
PaaS	platform as a service
PC	Personal Computer
PDA	Personal Digital Assistant
PDI	PhotoIonisation Detector

Ps	Poussières
PVC	polyvinyl chloride
p2p	peer to peer
SaaS	software as a service
SDK	Software Development Kit
SGBD	systèmes de gestion de bases de données
SGBDR	systèmes de gestion de bases de données relationnelles
SIM	Subscriber Identity Module
SMS	Short Message Service
SQL	Structured Query Language
TCP	Transmission Control Protocol
TOR	tout ou rien
URL	Uniform Resource Locator
VSAT	Very Small Aperture Terminal
WEP	Wired Equivalent Privacy
WSN	Wireless Sensor Network
WPA	Wi-Fi Protected Access
WPAN	Wireless Personal Area Network
WiFi	Wireless Fidelity
2G	Seconde generation

LISTE DES FIGURES

Figure 1-1: Les différentes couches de l'atmosphère	4
Figure 1-2 Le principe du capteur	9
Figure 2-1 Représentation générale d'un système IoT.....	15
Figure 2-2 Schéma du fonctionnement du réseau de capteur.....	16
Figure 3-1 Architecture du projet.....	27
Figure 3-2 La carte Arduino Uno.....	28
Figure 3-3 Le capteur mq135	29
Figure 3-4 . Caractéristique de la sensibilité du capteur	30
Figure 3-5 La courbe de CO.....	31
Figure 3-6 Le principe client Serveur avec le protocole HTTP	32
Figure 3-7 Le module ESP 8266	33
Figure 3-8 Les fonctionnalités de Java EE	34
Figure 3-9 Fonctionnement de Google App Engine	37
Figure 3-10 Installation de Google App Engine	38
Figure 3-11 Fonctionnement du datastore	40
Figure 3-12 Forme d'information stockée au datastore	40
Figure 3-13 Les entités dans le Datastore	41
Figure 3-14 Les entités pour le système de détection de la pollution atmosphérique.....	42
Figure 3-15 Programme Java EE pour créer une entité	43
Figure 4-1 Structure de projet	45
Figure 4-2 Le circuit d'alimentation pour le projet.....	46
Figure 4-3 Schéma du montage.....	47
Figure 4-4 Organigramme simplifié du firmware de l'Arduino	48
Figure 4-5 La page d'accueil.....	49
Figure 4-6 Courbe d'évolution de CO2.....	50
Figure 4-7 La Page de téléchargement	50
Figure 4-8 Représentation de la page notification CO2.....	51
Figure 4-9 Pourcentage des gaz non acceptable.....	52
Figure 4-10 Présentation de la page statistique.....	53
Figure II-1 Structure and configuration of MQ-135	57
Figure II-2 sensivity characteristics of the MQ-135	58

LISTE DES TABLEAUX

Tableau 1-1. La Composition chimique de l'air	4
Tableau 1-2 Le taux de composition chimique de l'air inspiré et expiré par l'homme	5
Tableau 1-3 Récapitulatif des principaux polluants et leurs origines	6
Tableau 2-1 Les différentes révisions de la norme 82.11	18
Tableau 2-2 Principaux PaaS	23
Tableau 2-3 Tableau récapitulatif des langages et des IDE pour le développement mobile ...	25
Tableau I-1 Caractéristiques de technologie sans fil.....	55
Tableau II-1 Standard work condition	56
Tableau II-2 Environnement condition	56
Tableau II-3 Sensivity caractéristique.....	57
Tableau III-1 La liste des commandes AT	59

INTRODUCTION GENERALE

Il y a déjà quelques années le réchauffement climatique et la pollution atmosphérique devient la première ennemie commune de tous les pays. La population mondiale aperçut les différents dangers et les conséquences de cette perturbation de l'environnement à l'humanité. Alors beaucoup de projets ont été lancés pour la protection de l'environnement dans le monde entier. Cependant, la pollution ne cesse pas de s'augmenter, elle attaque, la vie quotidienne et entraîne le changement climatique et de nouvelles maladies.

Tout comme les autres pays du monde, Madagascar n'a pas pu échapper au fléau moderne de la pollution atmosphérique. Ce phénomène frappe principalement nos grandes villes surtout Antananarivo, cela est dû à l'accroissement rapide de la population, la recherche de meilleur revenu, le développement commerciale et industrielle, la mauvaise infrastructure routière des villes, aux mauvais états du parc automobile en croissance rapide et au développement incontrôlé du transport urbain par les vieilles véhicules.

Ainsi, notre projet s'intitule « Conception d'un système de détection de pollution atmosphérique, intégrant le modèle «IoT». On a choisi l'IoT car c'est l'avenir proche de l'internet. L'IoT est la 4ème révolution industrielle, l'internet des objets représente l'extension d'Internet à des choses (les objets connectés) et à des lieux dans le monde physique. Un objet connecté est un objet électronique connecté sans fil et partage des informations avec un ordinateur, une tablette électronique, un smartphone ou autre appareil. [1]

Cette étude se focalise sur la détection des paramètres polluants de l'air et consiste à informer l'état de l'air. Le système a pour objectif d'aider l'utilisateur ou les acteurs clés de la protection de l'environnement à prendre des décisions en fonction.

Le manuscrit se divise en quatre chapitres. Le premier chapitre présente la généralité sur la pollution atmosphérique, les différents polluants de l'air et leurs effets sur la santé, le capteur de l'air et la présentation d'un cas existant, celui du système de surveillance de la qualité de l'air d'Ambatovy. Le deuxième chapitre présente la généralité sur l'internet des objets, son fonctionnement, le réseau des capteurs et les divers outils pour l'établissement de l'internet des objets. Le troisième chapitre présente le projet «le système de détection de la pollution de l'air ». Et enfin, le quatrième chapitre présente la réalisation du projet.

Chapitre 1. GENERALITE SUR LA POLLUTION

ATMOSPHERIQUE

Avant de concentrer sur les outils nécessaires sur la fabrication du système et de faire l'étude et la réalisation, on doit d'abord connaître l'atmosphère terrestre. Dans ce chapitre on va voir la généralité sur la pollution atmosphérique, en citant les différentes couches constituant l'atmosphère et les compositions chimiques de l'air. On va voir aussi la pollution de l'air et leurs principaux polluants, le capteur de gaz et le cas d'Ambatovy.

1.1. L'atmosphère et l'air

1.1.1. Atmosphère terrestre

L'Atmosphère terrestre s'est formée il y a environ 3,5 milliards d'années. C'est une couche de gaz environ 500km d'épaisseur qui enveloppe la terre. Elle a pour rôle de protéger aux objets célestes qui percutent la Terre. Grâce à elle la vie peut se développer. L'atmosphère comporte plusieurs couches .En partant du sol, on trouve la troposphère, la stratosphère, la mésosphère, thermosphère et l'ionosphère.

a. La troposphère

Zone de l'atmosphère terrestre située entre la surface du globe et une altitude d'environ 8 à 15 kilomètres, selon la latitude et la saison. Elle est plus épaisse à l'équateur qu'aux pôles. La frontière entre la troposphère et la stratosphère s'appelle la tropopause. [2]

b. La stratosphère

Zone de l'atmosphère d'une planète située au-dessus de la troposphère et caractérisée par une faible croissance de la température avec l'altitude. La stratosphère terrestre s'étend, en moyenne, entre 12 et 50 km d'altitude. Elle doit son nom à l'existence de courants essentiellement horizontaux. Elle renferme la quasi-totalité de l'ozone atmosphérique. [2]

c. La mésosphère

Zone de l'atmosphère d'une planète, située entre la stratosphère et la thermosphère et caractérisée par une décroissance de la température avec l'altitude, est située entre 50 km d'altitude, au-dessus de la stratosphère et 90 km d'altitude. C'est une zone de transition entre la Terre et l'Espace [2]. En y pénétrant, pour descendre sur Terre, les météorites, satellites, s'échauffent contre les quelques particules d'air qu'ils rencontrent et sont détruits avant d'atteindre le sol, sauf pour les plus grosses pièces. [2]

d. La thermosphère

Zone de l'atmosphère de la terre située au-dessus de la mésosphère et caractérisée par une forte croissance de la température avec l'altitude. Elle est située entre 85 et 600 km d'altitude. Entre 100 et 150 kilomètres d'altitude, le dioxygène moléculaire absorbe l'ultraviolet solaire de très courtes longueurs d'onde (entre 100 et 200nm). En résulte une augmentation de température avec l'altitude qui oscille entre 300 °C et 1600 °C selon l'activité solaire. Les températures sont élevées, mais la densité de matière est extrêmement faible. [2]

e. L'ionosphère

Caractérisée par la présence de particules chargées, formées par photo-ionisation sous l'effet du rayonnement solaire. La base de l'ionosphère se situe entre 350 et 800 km d'altitude suivant la température à la thermopause, qui est liée à l'activité solaire. L'hélium et l'hydrogène y sont les éléments prépondérants. Elle s'étend jusqu'à la limite extrême de l'atmosphère, soit 50 000 kilomètres. On ne trouve plus là que quelques atomes d'hydrogène. C'est dans cette couche, que la plupart des satellites sont placés en orbite. [2]

La figure 1-1 présente les différentes couches de l'atmosphère terrestre.

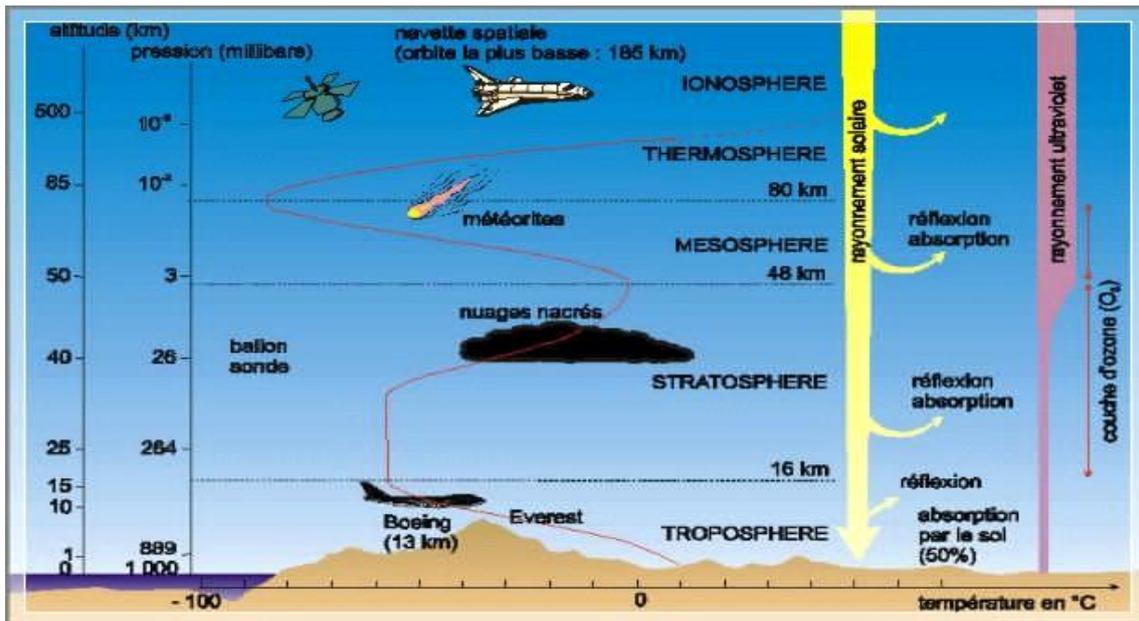


Figure 1-1: Les différentes couches de l'atmosphère [2]

1.1.2. L'Air

L'air est le mélange de gaz constituant l'atmosphère de la Terre. Il est incolore, invisible et inodore.

a. La Composition chimique de l'air

Le tableau ci-dessous montre la composition chimique de l'air pur.

Tableau 1-1. La Composition chimique de l'air [3]

Composition chimique de l'air	
azote	78,10%
oxygène	20,90%
Argon	0,90%
Gaz carbonique	0,03%
Autre gaz, dits rares	0,07 %

b. Rôle de l'air

La respiration des êtres vivants fournit aux organismes le dioxygène nécessaire au fonctionnement des cellules et les débarrasse du dioxyde de carbone qu'ils produisent. Chez l'Homme, l'air pénètre dans l'organisme par le nez ou la bouche et rejoint les poumons par

la trachée. Là, au niveau des alvéoles pulmonaires, un échange gazeux s'effectue à travers une membrane très fine en contact avec des capillaires sanguins. Du dioxygène diffuse dans le sang. Il est ensuite transporté à travers tout le corps par le système sanguin, jusqu'à chacune des cellules. Ces dernières l'utilisent comme énergie pour leur fonctionnement et rejettent du dioxyde de carbone qui est conduit, de nouveau grâce au sang, jusqu'aux alvéoles pulmonaires au niveau desquelles il diffuse dans l'air. En effet il est nécessaire d'inspirer et d'expirer 25 000 fois par jour [3]. Le Tableau 1-2 montre le taux de composition chimique de l'air inspiré et expiré par l'homme.

Tableau 1-2 Le taux de composition chimique de l'air inspiré et expiré par l'homme [3]

	Composition chimique de l'air inspiré	Composition chimique de l'air expiré
N_2	78,10%	78,10%
O_2	20,90%	16,50%
Gaz carbonique	0,3%	4,50%

1.1.3. La pollution de l'air

Quand l'air est modifié par des éléments qui sont nuisibles à notre santé et à notre environnement, on dit qu'il est pollué. La pollution de l'air résulte soit de l'introduction de substances nouvelles, on dit dans ce cas que le polluant est primaire, soit (cas le plus fréquent) d'un changement des concentrations habituelles de substances déjà présentés à l'état naturel et qui peuvent atteindre des valeurs dangereuses, on dit dans ce cas que le polluant est secondaire. Les polluants de l'air sont des agents chimiques, physiques ou biologiques qui affectent à court ou à long terme la santé des êtres vivants et des écosystèmes. Certains d'entre eux (CFC et HCFC, interdits depuis 1987) dégradent la couche d'ozone stratosphérique (« bon » ozone) qui nous protège du rayonnement solaire ultraviolet. Le dioxyde d'azote, l'ozone troposphérique et les particules sont des polluants de l'air extérieur et jouent aussi un rôle dans l'effet de serre.

a. Les principaux polluants et leurs origines

Il existe des polluants naturels et des polluants générés par les activités humaines.

i. Polluants naturels

Ce sont de gaz ou particules, ils sont émis par :

- les volcans, qui envoient dans l'atmosphère d'énormes quantités de gaz et de particules ;
- les plantes qui produisent des pollens, dont certains peuvent entraîner des allergies ;
- la foudre qui émet des oxydes d'azote et de l'ozone ;
- l'érosion qui produit des poussières. Transportées par le vent, elles peuvent parcourir de très longues distances.

ii. Polluants générés par les activités humaines

Il existe un nombre important de polluants. Les particules, les oxydes d'azote, les composés organiques volatils(COV) et l'ozone sont aujourd'hui les polluants les plus préoccupants.

Tableau 1-3 Récapitulatif des principaux polluants et leurs origines

POLLUANTS	ORIGINES
Le dioxyde de soufre : SO ₂	se dégage de la combustion du charbon, du fuel ou du gazole qui contiennent du soufre
L'hydrogène sulfureux : H ₂ S	est lié aux activités industrielles comme la fabrication de pâte à papier ou le raffinage du pétrole.
Les poussières : Ps	Proviennent de l'activité de certaines industries, des chaufferies et des véhicules diesel.
Les fumées noires : FN	sont générées par des phénomènes de combustion industrielle et d'origine

	automobile (diesel).
Le plomb : Pb	résulte de la métallurgie du plomb, celui induit par la circulation automobile est en réduction (essence sans plomb).
L'ozone : O ₃	se crée sous l'action du soleil à partir de l'oxygène, d'oxydes d'azote et d'hydrocarbures.
Les oxydes d'azote : NO ₃	sont principalement émis par les automobiles, les installations de combustion (dont les gazinières dans les cuisines) et les ateliers utilisant l'acide nitrique.
L'ammoniac : NH ₃	provient de l'incinération des ordures ménagères.
Le monoxyde de carbone : CO	est issu de la combustion incomplète de produits carbonés et entre autres, des carburants des véhicules à moteur à explosion.
Gaz carbonique : CO ₂	provient principalement de la combustion des énergies fossiles et de la déforestation.
Les hydrocarbures	sont produits par la combustion incomplète de carburants, le stockage du pétrole, l'utilisation et la fabrication de solvants (peinture, imprimerie...)

Le méthane : CH ₄	Est issu de la fermentation de matières organiques ou la digestion des ruminants
Les Composés Organiques Volatils: COV	Sont issus des activités humaines, principalement l'automobile.
Les chlorofluorocarbures : CFC	ont été largement utilisés dans les circuits de refroidissement ou comme gaz propulseur (aérosols) ou dans certaines mousses isolantes. Leur production et leur utilisation sont interdites.

b. Les effets de la pollution de l'air

Les effets de la pollution atmosphérique sont nombreux et sont tous dangereux, surtout, pour la santé. En effet, l'homme consomme environ 14 000 litres d'air par jours, ce qui veut dire qu'en vivant dans une atmosphère polluée, la santé se détériore rapidement. La liste de maladies que provoquent les polluants est longue : il y a les cancers de poumons, les différentes sortes de problèmes respiratoires, les maladies cardio-vasculaires etc.

En 2012, l'OMS estime que 3.7 millions de morts prématurés ont été provoqués par la pollution de l'air extérieur dans les zones urbaines et rurales.

Outre la pollution de l'air extérieur, la fumée domestique représente un grave risque sanitaire pour environ 3 milliards de personnes qui font cuire leurs aliments et chauffent leur logement à l'aide de combustibles à base de biomasse et de charbon. On estime que la pollution de l'air à l'intérieur des habitations était responsable de 4,3 millions de décès prématurés en 2012. [4]

On associe d'importants risques sanitaires à l'exposition aux particules fines, mais aussi à l'ozone, au dioxyde d'azote et au dioxyde de soufre. Les concentrations de ces molécules sont généralement plus importantes dans les zones urbaines des pays à revenu faible ou intermédiaire. L'ozone est l'un des principaux facteurs de risque de mortalité liée à l'asthme, le dioxyde d'azote et le dioxyde de soufre ayant, quant à eux, une incidence sur l'asthme, les affections bronchiques, les inflammations pulmonaires et l'altération des fonctions pulmonaires.

D'autre part, les effets à long termes que provoquent certains polluants sont aussi de menace contre l'humanité comme la dégradation de la couche d'ozones (provoqués par le

chlorofluorocarbures) et le réchauffement climatique (provoqués par les gaz à effets de serre comme le méthane et le dioxyde de carbone).

Tous ces polluants peuvent être déterminés par des capteurs spécifiques aujourd'hui, ces capteurs sont nombreux et ont tous ses propres méthodes pour détecter les polluants.

1.2. Le capteur

1.2.1. Définition d'un capteur

Un capteur est un dispositif transformant l'état d'une grandeur physique observée en une grandeur utilisable, telle qu'une tension électrique. [5]

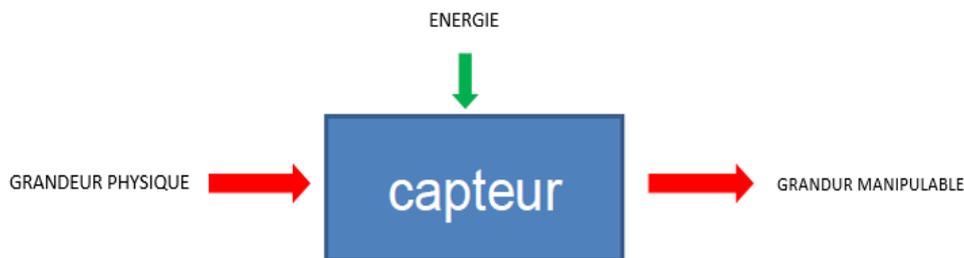


Figure 1-2 Le principe du capteur

1.2.2. Les caractéristiques d'un capteur

- **Etendue de mesure :** Ce sont les valeurs extrêmes pouvant être mesurées par le capteur.[5]
- **Résolution :** C'est la plus petite variation de grandeur mesurable par le capteur. [5]
- **Sensibilité :** C'est la variation du signal de sortie par rapport à la variation du signal d'entrée. [5]
- **Précision :** C'est l'aptitude du capteur à donner une mesure proche de la valeur vraie.[5]
- **Rapidité :** C'est le temps de réaction du capteur. La rapidité est liée à la bande passante.[5]
- **Linéarité :** Représente l'écart de sensibilité sur l'étendue de mesure. [5]

1.2.3. Classification des capteurs

a. Capteurs actifs

Ils ont besoin dans la plupart des cas d'apport d'énergie extérieure pour fonctionner[5]. Ce sont des capteurs modélisables par une impédance[5]. Une variation du phénomène physique étudié (mesuré) engendre une variation de l'impédance[5]. Il faut leur appliquer une tension pour obtenir un signal de sortie.

Exemples : thermistance, photorésistance, potentiomètre, jauge d'extensomètre appelée aussi jauge de contrainte.

b. Capteurs passifs

On parle de capteur passif lorsque le phénomène physique qui est utilisé pour la détermination du mesurande effectue directement la transformation en grandeur électrique [5]. C'est la loi physique elle-même qui relie mesurande et grandeur électrique de sortie [5]. Un capteur passif fonctionne assez souvent en électromoteur et dans ce cas, la grandeur de sortie est une différence de potentiel. Le nombre des lois physiques permettant une telle transformation est évidemment limité, on peut donc recenser facilement les capteurs passifs (dont le nombre est fini). Toutefois, leurs domaines d'application sont très étendus.

Exemple : capteurs de position, potentiomètre, inductance à noyaux mobile, condensateur à armature mobile.

1.2.4. Type de sortie de capteur

Les capteurs et leurs conditionneurs peuvent aussi faire l'objet d'une classification par type de sortie :

a. Les Capteurs analogiques

La sortie est une grandeur électrique dont la valeur est une fonction de la grandeur physique mesurée par le capteur[5]. La sortie peut prendre une infinité de valeurs continues. Le signal des capteurs analogiques peut être du type :

- sortie courant ;
- sortie tension ;

- règle graduée, cadran, jauge (avec une aiguille ou un fluide) ;
- etc.

Exemples : capteur à jauge de contrainte, thermocouple.

La sortie est une séquence d'états logiques qui, en se suivant, forment un nombre. La sortie peut prendre une infinité de valeurs. Le signal des capteurs numériques peut être de type :

- train d'impulsions, avec un nombre précis d'impulsions ou avec une fréquence précise.
- code numérique binaire
- bus de terrain
- etc.

Exemples : codeur rotatif incrémental, codeurs référentiels AA34.

b. Capteurs logiques ou capteurs TOR

La sortie est un état logique que l'on note 1 ou 0 [5]. La sortie peut prendre ces deux valeurs [5]. Le signal des capteurs logiques peut être de type :

- présent/absent dans un circuit
- potentiel, souvent 5 V/0 V
- LED allumée/éteinte
- signal pneumatique (pression normale/forte pression)
- etc.

Exemples : capteurs de fin de course, capteurs de rupture d'un faisceau lumineux, divers capteurs de position.

1.2.5. Les différents types de capteur de l'air

- Catalytic sensors : Utilisé principalement pour détecter les gaz combustibles

Exemples : MC101, MC102, MC109, MC112, MC113, MC119

- Electrochemical Sensors(EC Sensor) : utilisé principalement pour détecter les gaz toxique.

Exemples: ME2-CO, ME3-CO, ME3-O2, DSO115.

- Semi-Conductor Sensors : utilisé principalement pour détecter les gaz toxique et les CFC.

Exemples : MQ2, MQ3, MQ4, MQ5, MQ6, MQ7, MQ135, MQ136, MQ137

- PhotoIonisation Detector (PDI sensor) : utilisé pour les composants volatiles.

Exemples : ESO150, ESO152, DSO150.

1.3. Présentation d'un cas existant de gestion de l'air

A Madagascar, le ministère de l'environnement et l'ONE travaillent ensemble pour la protection de l'environnement. Ils adoptent des lois qui limitent l'émission des polluants chimiques. En effet les automobiles doivent passer en visite technique avant d'être utilisées, les entreprises doivent respecter des normes sur l'état de l'air. Ambatovy est une société utilise des produits chimiques, cependant il présente dans son site un programme qui gère l'environnement.

1.3.1 Le Programme de gestion de l'air à Ambatovy

a. Problématique

Le Programme de gestion de l'air est conçu pour surveiller les sources connues de paramètres d'émission-clés, comme les particules, le dioxyde de soufre, l'oxyde d'azote, l'hydrogène sulfuré et l'ammoniac [6]. À la Mine, les paramètres-clés contrôlés sont les poussières ou les particules en raison de la manutention et du transport continu de grandes quantités de minerai. De même, au Port, la poussière peut résulter du déchargement des marchandises en vrac telles que le calcaire, le soufre et le charbon [6]. Sur le site de l'Usine, le transport et le stockage des produits en vrac peuvent être source de matières particulaires [6]. La suppression de la poussière de ces sources est réalisée par pulvérisation d'eau ou des systèmes de brume [6]. D'autres particules peuvent résulter de la combustion des véhicules de flotte, des générateurs diesel et de la centrale électrique alimentée au charbon [6]. La sélection, la conception et la maintenance des équipements de combustion peuvent aider à réduire les émissions de particules et l'utilisation des dépoussiéreurs dans la centrale électrique peut soutenir le contrôle des émissions de particules [6]. Les processus utilisés à l'Usine pour l'extraction et le raffinage du nickel et du cobalt à partir de la latérite sont

complexes et peuvent également être une source d'émission de dioxyde de soufre, d'oxydes d'azote, d'hydrogène sulfuré et d'ammoniac à faibles niveaux. [6]

La conception des installations a été évaluée au cours d'un processus d'Évaluation d'Impact Environnemental, afin d'assurer la conformité avec les normes internationales de qualité de l'air. L'émission de gaz à effet de serre par Ambatovy a été en partie minimisée par le transport du minerai de la Mine à l'Usine via le Pipeline et pour lequel la gravité fournit une grande partie des besoins en énergie [6]. L'utilisation d'une flotte d'équipement minier moderne réduit davantage les émissions potentielles de gaz à effet de serre [6]. La conception et l'exploitation de l'installation de traitement maximise l'utilisation de l'énergie grâce à l'utilisation efficace des systèmes de vapeur à haute, moyenne et basse pression [6]. En outre, le programme de compensation de la biodiversité, mis en place pour veiller à ce qu'aucune perte nette de biodiversité ne se produise à cause du Projet, peut en partie compenser l'émission de gaz à effet de serre. [6]

b. Mise en place du système

Pour s'assurer que les émissions atmosphériques restent dans les limites permises, un système de surveillance de l'air a été mis en place [6]. Le système utilise une méthode de suivi analytique continu ainsi qu'un système d'échantillonnage passif périodique [6]. Une météorologie en temps réel du site de l'Usine et des données sur la qualité de l'air est accessible par ordinateur en tout temps [6].

Le système de surveillance de la qualité de l'air d'Ambatovy utilise les EC sensors pour les gaz toxique, et les PID sensors pour les composants volatiles.

1.4. Conclusion

L'atmosphère terrestre protège la vie sur la terre. L'air fait partie de l'atmosphère, il est la base de la vie sur terre. Cependant, l'activité humaine a provoqué des polluants qui détruisent de jour en jour l'atmosphère alors que la pollution de l'air provoque des maladies et augmente le taux de mortalité. Grâce au développement de la technologie, aujourd'hui il existe des capteurs de gaz, ces capteurs peuvent collecter les états de gaz polluant. Pour notre projet, on utilise le modèle internet des objets pour communiquer le réseau de capteurs avec les utilisateurs. Le chapitre suivant va présenter l'internet des objets

Chapitre 2. LA PRESENTATION DE L'INTERNET DES OBJETS

L'internet des objets est l'interconnexion nouvelle des objets du quotidien: ils disposent d'une connexion sans fil à internet et peuvent être programmés et pilotés à distance via un ordinateur, une tablette ou un smartphone, qui collectent des données.

2.1. Principe de fonctionnement

On met un ou plusieurs capteurs sur l'objet pour collecter des données et un circuit électronique pour transmettre les données collectées vers un autre appareil à distance qui interprète ces données et qui peut prendre une décision.

L'appareil à distance peut être un logiciel ou une application que l'utilisateur peut commander manuellement. Par exemple : il peut diminuer ou augmenter la température dans la chambre de son bébé, à distance, avec son portable.

Mais l'appareil peut être un dispositif électronique qui, prend tout seul, une décision en fonction des données qu'il reçoit. Par exemple : s'il fait noir, on allume la lampe, ou encore : si le battement du cœur de l'individu est anormal, on l'alerte.

Entre les objets connectés se trouve le système de communication pour que l'ensemble du système fonctionne. On peut utiliser internet, par exemple, mais il se peut aussi que certains objets connectés a son propre réseau, exemple : dans un système de sécurité dans un hôtel, tous les caméras appartiennent à un même réseau local. La Figure 2-1 présente l'architecture d'un objet connecté à l'internet ou le modèle IoT.

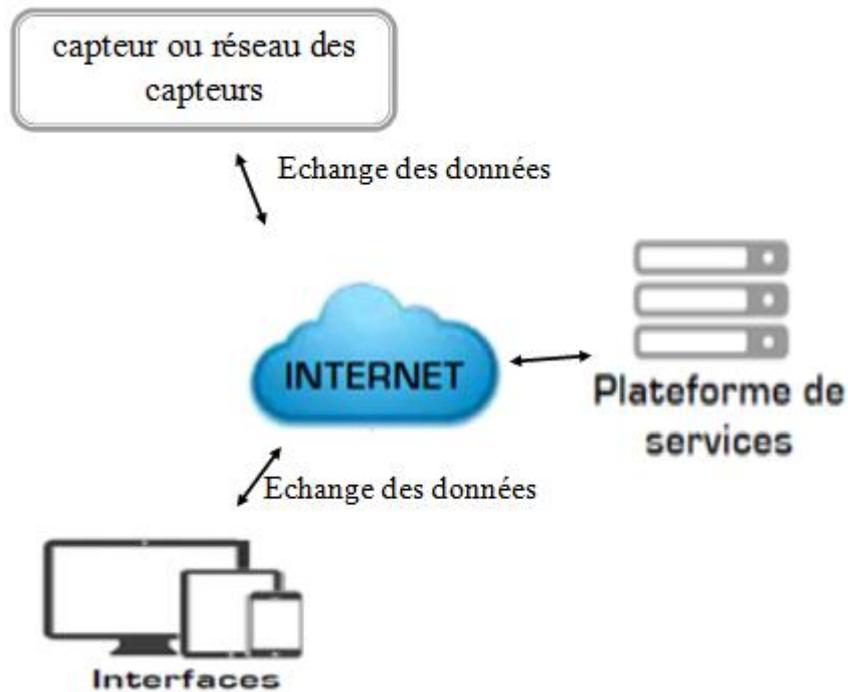


Figure 2-1 Représentation générale d'un système IoT

Le modèle comporte trois couches différentes : réseaux de capteurs, plateformes de service et l'interfaces utilisateurs.

2.2. Le réseau des capteurs(WSN)

C'est la partie plus bas niveau du système. Un réseau de capteur, ou Wireless Sensor Network (WSN) en anglais, est un ensemble de capteurs autonomes à faible coût, interconnectés par un réseau de communications afin de couvrir une zone cible. Cette zone peut être géographique ou délimitée par un système plus ou moins étendu : un ouvrage d'art, un ensemble mécanique, un outillage, un réseau télécommunication.

Les réseaux de capteurs connaissent de multiples applications telles que la métrologie environnementale, urbaine ou industrielle.

En réseau, ces capteurs peuvent interagir entre eux et avec un système externe par le biais de communication sans fil ou filaire.

2.2.1. Principe de fonctionnement de WSN.

Les données captées par les capteurs (nœuds) sont acheminées à un nœud considéré comme un "point de collecte", appelé nœud-puits (ou sink). Ce dernier peut être connecté à l'utilisateur du réseau (via Internet, un satellite ou un autre système). L'utilisateur peut adresser des requêtes aux autres nœuds du réseau, précisant le type de données requises et récolter les données environnementales captées par le biais du nœud puits.

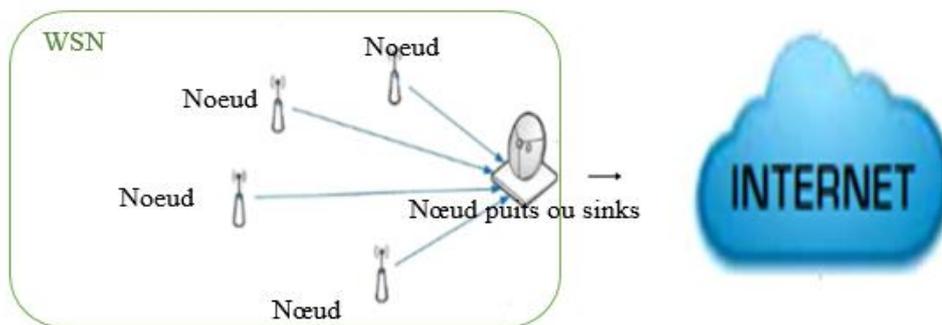


Figure 2-2 Schéma du fonctionnement du réseau de capteur

2.2.2. Application des réseaux de capteur

La diminution de taille et de coût des micro-capteurs, l'élargissement des types de capteurs disponibles et l'évolution des supports de communication sans fil, ont élargi le champ d'application des réseaux de capteurs. Ils s'insèrent notamment dans d'autres systèmes tels que le contrôle et l'automatisation des chaînes de montage. Ils permettent de collecter et de traiter des informations complexes provenant de l'environnement.

Le réseau de capteur est utilisé pour surveiller des phénomènes physiques et biologiques dans différents domaines.

- industriels, techniques et scientifique (monitoring de la température, la pression, l'hygrométrie, la luminosité, etc.).
- écologie et environnement (surveillance des ultraviolets, de la radioactivité, de polluants tels que les HAP, les métaux lourds, ou de l'ozone ou du NO₂ ou encore le CO₂ et d'autres gaz à effet de serre
- santé : suivi des malades, veille éco-épidémiologique et épidémiologique,
- sécurité,

- transports : contrôle du trafic routier urbain, prévention des accidents, optimisations diverses,
- l'automatisation des bâtiments domotique.

2.2.3. Les Options de communication pour le WSN

Les WSN utilisent le réseau sans fil, ce dernier est plus avantageux que le réseau câblé. Le réseau sans fil utilise le canal air pour communiquer c'est-à-dire les ondes hertziennes, les infrarouges ou le laser.

Avantage de réseau sans fil par rapport au réseau câblé :

- Facilité de déploiement ;
- La réduction du temps de déploiement et d'installation ;
- La réduction des coûts d'entretien ;
- L'augmentation de la connectivité ;
- La réduction de l'encombrement ;
- La portabilité et la mobilité.

2.2.4. Les différents types de technologie sans fil

c. Bluetooth

Bluetooth est une technologie de réseau personnel sans fils noté WPAN pour Wireless Personal Area Network. 802.15.4 est un standard pour des applications Bluetooth à bas débit, un protocole de communication créé par l'Institute of Electrical and Electronics Engineers (IEEE). Il est défini dans la deuxième couche de modèle OSI (Open System Interconnexion), il a pour but de laisser la communication entre deux appareils. Le réseau peut fonctionner avec la topologie en étoile ou p2p (peer to peer).

d. Zigbee

Zigbee est un protocole créé par un ensemble de sociétés qui forme alliance ZigBee. Cette norme est défini dans la couche de communication au niveau 3 et supérieur dans le modèle OSI.

e. Le wireless Fidelity (WIFI)

WiFi est le nom donné à la norme IEEE 802.11a, 802.11b, 802.11g, C'est une technologie plus utilisée aujourd'hui pour le réseau local sans fil. Le tableau 2-1 présente les différentes révisions de la norme 802.11 et leur signification.

Tableau 2-1 Les différentes révisions de la norme 802.11 [7]

Standard	Bande de fréquence	Débit	Portée
Wifi A (802.11a)	5GHz	54 Mbit/s	10m
Wifi B (802.11 b)	2.4 GHz	11 Mbit/s	100m
Wifi G (802.11g)	2.4 GHz	54 Mbit/s	100m

f. Global System for Mobile communications(GSM)

Le GSM Shield Arduino relie l'arduino à Internet en utilisant le réseau sans fil GPRS. Il suffit de brancher le module sur la carte Arduino, et brancher une carte SIM d'une couverture GPRS opérateur d'offre et de suivre quelques instructions simples.

g. Les Réseaux satellites

La communication par satellite se fait par ondes radios qui sont des signaux électromagnétiques. Les satellites n'utilisent que l'énergie solaire et des batteries comme source d'énergie. Pour communiquer avec les satellites le client doit avoir une antenne.

Dans le cas d'un satellite géostationnaire, il s'agit en général d'un Very Small Aperture Terminal(VSAT), une parabole de petite taille pouvant faire de 75 à 120 centimètres. Le satellite dispose d'un certain nombre de transpondeur qui lui permettent de recevoir les signaux d'une fréquence donnée afin de les retransmettre vers la terre sur une autre fréquence donnée. Ces transpondeurs sont donc composés d'un couple émetteur/récepteur.

Les satellites géostationnaires utilisent une émission par diffusion, ce qui signifie que le signal envoyé vers un client couvre une énorme zone territoriale. Chaque client dans la zone de couverture peut donc recevoir potentiellement ce signal. De retour sur Terre, le signal est reçu et converti en données numériques par le client (lorsqu'il s'agit d'une transmission de données). Ce client peut être soit un particulier, une entreprise ou le fournisseur d'accès. Dans

le cas d'une transmission TCP par exemple les paquets peuvent donc être lus par un particulier, ou routés vers l'Internet dans le cas du fournisseur d'accès. Les satellites peuvent se situer sur plusieurs types d'orbites en fonction de leur utilisation.

Voici les différents types d'orbites:

- Geostationary Earth Orbital Satellite(GEOS)[8]

Orbite géostationnaire: 35786 km ; Temps aller-retour de l'onde radio: ~260ms Ces types d'orbites sont les plus utilisés dans le domaine de la diffusion vidéo et de l'accès à l'Internet.

- Medium Earth Orbital Satellite(MEOS)[8]

De 2000 à 35000 km Temps aller-retour de l'onde radio: ~ 100ms Les MEOS sont placés entre les satellites à orbite haute et ceux à orbite basse. [8]

- Low Earth Orbital Satellite(LEOS)[8]

De 200 à 2000 km GEOS Temps aller-retour de l'onde radio: inférieur à 10ms. [8]

2.3. Les plateformes de service

On a besoin d'une base de travail pour pouvoir écrire, lire, développer et utiliser un ensemble de logiciels pour faciliter l'interfaçage entre le WSN et l'utilisateur. Les solutions sont nombreuses pour cela mais on va regarder ceux qui respectent le principe de «Cloud Computing» ou « Informatique en nuage » car cette solution est, aujourd'hui, la plus performante du point de vue flexibilité et accessibilité.

Cette couche se charge donc de traiter les données récoltées par les capteurs: les stocker dans une base de données, effectue les calculs nécessaires et renvoie les données demandées par l'utilisateur vers son support.

Pour introduire le sujet de plateforme de service, on doit, d'abord, parler le principe de « Cloud Computing ».

2.3.1 Le «Cloud Computing»

Les 5 générations de l'informatique sont le mainframe, le PC, le client serveur, le web service et le cinquième est le cloud computing.

a. Définition

Le «Cloud Computing» ou Informatique en nuage est une manière de fournir et d'utiliser les aptitudes des systèmes informatiques, qui est basée sur les nuages (cloud en anglais): un parc de machines, d'équipement de réseau et de logiciels maintenus par un fournisseur, que les consommateurs peuvent utiliser en libre-service via un réseau informatique, le plus souvent Internet. Les caractéristiques techniques du nuage ne sont pas connues du consommateur et les services sont payés à l'usage. [9]

Les principaux acteurs de cloud computing sont: Amazon, Citrix, Google, HP, IBM, Intel, Microsoft.

b. Principe de fonctionnement

Un nuage (cloud) est un ensemble de matériel, de raccordements réseau et de logiciels qui fournissent des services sophistiqués que les individus et les collectivités peuvent exploiter à volonté à travers le monde. Le cloud computing est un basculement de tendance: un passage de la production maison d'aptitudes par l'acquisition de matériel et de logiciel, vers l'utilisation d'une aptitude qui est un service fourni par des tiers et mis à disposition via l'Internet.

Les caractéristiques essentielles d'un nuage sont la disponibilité mondiale en libre-service, l'élasticité, l'ouverture, la mutualisation et le paiement à l'usage :

- ressources en libre-service, et adaptation automatique à la demande. La capacité de stockage et la puissance de calcul sont adaptées automatiquement au besoin d'un consommateur. Ce qui contraste avec la technique classique des hébergeurs où le consommateur doit faire une demande écrite à son fournisseur en vue d'obtenir une augmentation de la capacité - demande dont la prise en compte nécessite évidemment un certain temps. En cloud computing la demande est automatique et la réponse est immédiate.
- ouverture. Les services de cloud computing sont mis à disposition sur l'Internet, et utilisent des techniques standardisées qui permettent de s'en servir aussi bien avec un ordinateur qu'un téléphone ou une tablette.
- mutualisation. La mutualisation permet de combiner des ressources hétérogènes (matériel, logiciel, trafic réseau) en vue de servir plusieurs consommateurs à qui les ressources sont automatiquement attribuées. La mutualisation améliore la

scalabilité (monter en charge) et l'élasticité et permet d'adapter automatiquement les ressources aux variations de la demande.

- paiement à l'usage: la quantité de service consommée dans le cloud est mesurée, à des fins de contrôle, d'adaptation des moyens techniques et de facturation.

Les nuages utilisent des technologies telles que la virtualisation du matériel informatique, les grilles, l'architecture orientée services et les services web. Un nuage peut être public, privé ou communautaire.

Un exemple de service grand public fourni en cloud computing, est le jeu à la demande. Il permet de jouer normalement à des jeux vidéo sur son écran d'ordinateur, alors que le ou les logiciels de jeu tournent sur des serveurs à distance, qui renvoient la vidéo de ce qui a été joué en lecture en continu (appelé streaming). Le jeu est hébergé et stocké sur des serveurs, dont l'utilisateur ne connaît pas la localisation ni les caractéristiques.

2.3.2 Les services Cloud

Il existe trois catégories de services qui peuvent être offerts en cloud computing : Le logiciel en tant que service ou Software as a Service(SaaS), la plateforme en tant que service ou Platform as a Service(PaaS) et l'infrastructure en tant que service ou Infrastructure as a Service(IaaS).

a. Le logiciel en tant que service :SaaS

Avec ce type de service, des applications sont mises à la disposition des consommateurs. Les applications peuvent être manipulées à l'aide d'un navigateur web, et le consommateur n'a pas à se soucier d'effectuer des mises à jour, d'ajouter des patches de sécurité et d'assurer la disponibilité du service. Gmail est un exemple de tel service. Il offre au consommateur un service de courrier électronique et le consommateur n'a pas à se soucier de la manière dont le service est fourni. D'autres exemples de logiciels mis à disposition en SaaS sont Google Apps, Office Web Apps ou LotusLive (IBM).

b. La plateforme en tant que service : PaaS

Avec ce type de service, situé juste au-dessus du précédent, le système d'exploitation et les outils d'infrastructure sont sous la responsabilité du fournisseur. Le consommateur a le contrôle des applications et peut ajouter ses propres outils. La situation est analogue à celle de l'hébergement web où le consommateur loue l'exploitation de serveurs sur lesquels les outils nécessaires sont préalablement placés et contrôlés par le fournisseur. La différence étant que les systèmes sont mutualisés et offrent une grande élasticité - capacité de s'adapter automatiquement à la demande, alors que dans une offre classique d'hébergement web l'adaptation fait suite à une demande formelle de la consommation. Le Tableau 2-2 montrant les principaux Paas.

Tableau 2-2 Principaux PaaS [10]

Service	Editeur	Langages	IDE	Bases de données
Amazon Elastic Beanstalk	Amazon Web Services	Containers Docker, Java (Tomcat), .NET, Node.js, PHP, Python, Ruby (Passenger)	Extensions AWS Toolkit pour Eclipse et Microsoft Visual Studio	Amazon RDS, DynamoDB, SimpleDB, SQL Server, Oracle, IDBM DB2, Informix
Google App Engine	Google	Java, Python, PHP 5.4, Go	Google Plugin for Eclipse pour Java	Cloud SQL Datastore
Azure Cloud Services	Microsoft	Java, Node.js, PHP, Python, .NET, Ruby	WebMatrix, Visual Studio + Azure SDK	Data Services Microsoft : Blobs, base SQL, SQL Server, MongoDB
Salesforce1 platform	Salesforce	Langage Apex sur Force.com, Java, Ruby, Node.js, Python sur Heroku1	Outil web pour Force.com, client Heroku toolbelt pour Windows, MacOS X, Linux Debian/Ubuntu	Force.com Database, Heroku Postgres + les les SGBD de l'add-on marketplace Heroku, dont MongoDB, Postgres, MySQL
Cloud Foundry	Cloud Foundry Community	Java, Grails, Play, Spring, Node.js, Ruby on Rails, Sinatra, Go, Erlang, etc.	Plugin Eclipse	MySQL, PostgreSQL, MongoDB

Le service PaaS offre des moyens pour créer une application dans le cloud, l'application est accessible en mode SaaS.

c. L'infrastructure en tant que service :IaaS

C'est le service de plus bas niveau. Il consiste à offrir un accès à un parc informatique virtualisé. Des machines virtuelles sur lesquelles le consommateur peut installer un système d'exploitation et des applications. Le consommateur est ainsi dispensé de l'achat de matériel informatique. Ce service s'apparente aux services d'hébergement classiques de centre de traitement de données, et la tendance est en faveur de services de plus haut niveau, qui fait abstraction de détails techniques.

2.4. L'interface utilisateur pour l'ioT

L'interface utilisateur est un moyen utilisé par l'utilisateur pour communiquer avec un système. Le système iot utilise une application informatique pour gérer les données de capteur et pour l'interface utilisateur. En utilisant le cloud on peut choisir entre une application web et une application mobile.

2.4.1 Application mobile

Une application mobile est un logiciel pouvant être installé sur un « smartphone », ou tablette. Elles sont créées pour fonctionner avec un système d'exploitation spécifique. On distingue plusieurs systèmes d'exploitation spécialement conçus pour mobile. Leur constructeur les améliore toujours et produit plusieurs versions. Il existe des plateformes et des technologies permettant de créer des applications propres à chacun d'eux. Le tableau 2-3 ci-dessous montre les langages et l'environnement de développement pour chaque système d'exploitation.

Tableau 2-3 Tableau récapitulatif des langages et des IDE pour le développement mobile

Système d'exploitation	Propriétaire	Langage de utilisé pour créer une application mobile	Environnement de développement(IDE)
Android	Google	JAVA	Android Studio
Ios	Apple	Objective C	Xcode IDE
Windows phone	Microsoft	C#	Visual Studio
BlackBerry	Research In Motion (RIM)	JAVA	SDK BlackBerry WebWorks+ SDK natif de BlackBerry

2.4.2 Interface Web

On peut interfacier aussi l'utilisateur et le système avec une interface web c'est à dire avec des pages web. Dans ce cas on utilise une application web. Une application web est un logiciel applicatif hébergé sur un serveur et accessible via un navigateur web.

2.5. Exemples d'objets connectés

2.5.1 Diagnoplant

Une suite d'applications de L'INRA (l'Institut national de la recherche agronomique), dédiées à divers produits agricoles : courgette, tomate, vigne, salade... Chacune d'elles permettent d'identifier et diagnostiquer les maladies susceptibles d'abimer telle ou telle culture. Les utilisateurs peuvent détecter les problèmes via des images ou par une liste de noms latins destinés aux plus expérimentés. Chacune des maladies connues est ensuite expliquée et détaillée. [11]

2.5.2 iCropTrak

iCropTrak est une application d'assistance dans la gestion d'une ferme. Celle-ci collecte l'ensemble des données relatives au ferme et aide dans la plantation, la pulvérisation, les engrais, l'irrigation, l'échantillonnage. Fonctionnant en cloud, elle permet « la collaboration et la synchronisation des données de l'ensemble de votre équipe tout en offrant la flexibilité pour mettre en œuvre vos propres formulaires de données » explique la société éditrice sur son site. [11]

2.5.3 Biconnect

C'est une application web, accessible via un ordinateur, un smartphone ou une tablette. Concrètement elle permet de détecter rapidement les complications liées à la chimiothérapie. Les patients peuvent alors alerter le médecin grâce à la collecte de plusieurs données comme la température, le symptôme tels que les frissons ou encore un essoufflement important. [12]

2.6. Conclusion

L'IoT est largement répandu actuellement pour l'usage quotidien. Il simplifie et améliore la vie. Il est présent presque dans tous les domaines et est utilisé pour surveiller la présence des maladies dans les agricultures, pour la gestion d'une ferme, et dans le domaine de médecine humaine. Grâce au développement de la technologie, les outils informatique et électronique pour réaliser l'ioT sont très nombreux aujourd'hui. Le Cloud computing est un outil informatique très récent pour réaliser l'ioT. Nous allons exploiter l'ioT et le Cloud computing pour la réalisation de notre projet: un système de détection de pollution atmosphérique sera présenté dans le chapitre suivant.

Chapitre 3. PRESENTATION DU PROJET : « Système de détection de pollution de l'air »

3.1 Introduction

Le système est donc constitué par un capteur qui récupère et envoie les informations à l'internet vers le google apps engine, le google apps engine, l'offre Paas de google permettant d'utiliser le serveur de google et le service de stockage, et l'interface informatique qui permet à l'utilisateur de manipuler le système. La figure 3-1. suivante montre l'architecture de projet avec les différents éléments qui le constituent.

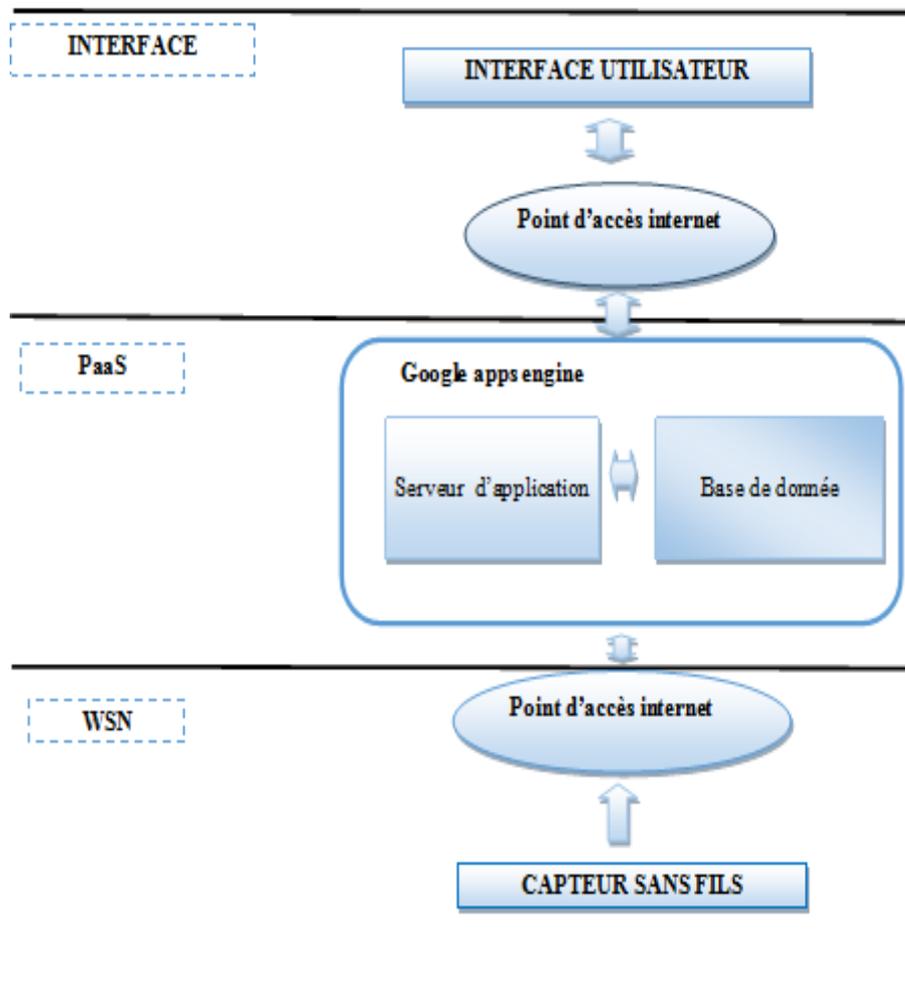


Figure 3-1 Architecture du projet

Malgré ces trois couches, le système peut être divisé en deux grandes parties : la partie matérielle et la partie application.

3.2 La partie matérielle

Dans la partie matérielle on va créer un capteur constitué par un microcontrôleur, une technologie sans fil et des capteurs de pollution de l'air.

3.2.1 Le microcontrôleur Arduino

Une carte arduino est une carte électronique permettant de créer un système électronique, elle repose sur un microcontrôleur associée à des entrées et sorties qui permettent à l'utilisateur de brancher différents types d'éléments externes. Et ensuite il faut gérer tous les éléments branchés sur la carte par un programme en utilisant l'environnement de programmation Arduino (IDE). [13]

On distingue plusieurs types de carte arduino comme l'Arduino uno, arduino mega, elles sont différent les un des autre par sa capacité, et sa puissances de calcule. On les choisi en fonction de nombre et de puissance des matériels utilisés avec lui. La figure 3-2 ci-dessous présente une carte arduino uno. [13]



Figure 3-2 La carte Arduino Uno[13]

3.2.2 Le Capteur de pollution de l'air

Pour connaître la qualité de l'air dans un lieu, on utilise un capteur. Ce capteur doit donc détecter et mesurer le niveau des polluants chimiques.

a. MQ-135 Gaz sensor

Le MQ-135 est un capteur de gaz utilisé pour connaître la qualité de l'air, ce capteur a une large champ de détection ,avec une réponse rapide et haute sensibilité ,il est aussi stable et a une long durée de vie. Un capteur chimique pouvant mesurer des nombreux gaz, il peut être utilisé donc pour mesurer les polluants présents dans l'atmosphère d'une ville. Il est sensible au Gaz carbonique, à l'alcool, au Benzène à l'oxyde d'azote, à l'ammoniac et au monoxyde de carbone. La Figure 3-3 montre le schéma du capteur MQ-135.



Figure 3-3 Le capteur mq135 [14]

Il est composé d'un micro tube en céramique, de Dioxyde d'étain avec une conductivité électrique varie en fonction de la présence de polluant(s), et l'électrode de mesure et de chauffage.

Il existe quatre broches dans le capteur, les pins Vcc pour l'alimentation, ground (GND), A0 et D0 pour envoyer le signal analogique.

b. Sensibilité de MQ-135

La figure 3-3 suivante montre la caractéristique de la sensibilité du capteur. Cette figure est utilisée pour convertir la sortie du capteur à la valeur en partie par million (ppm) correspondant au gaz à détecter.

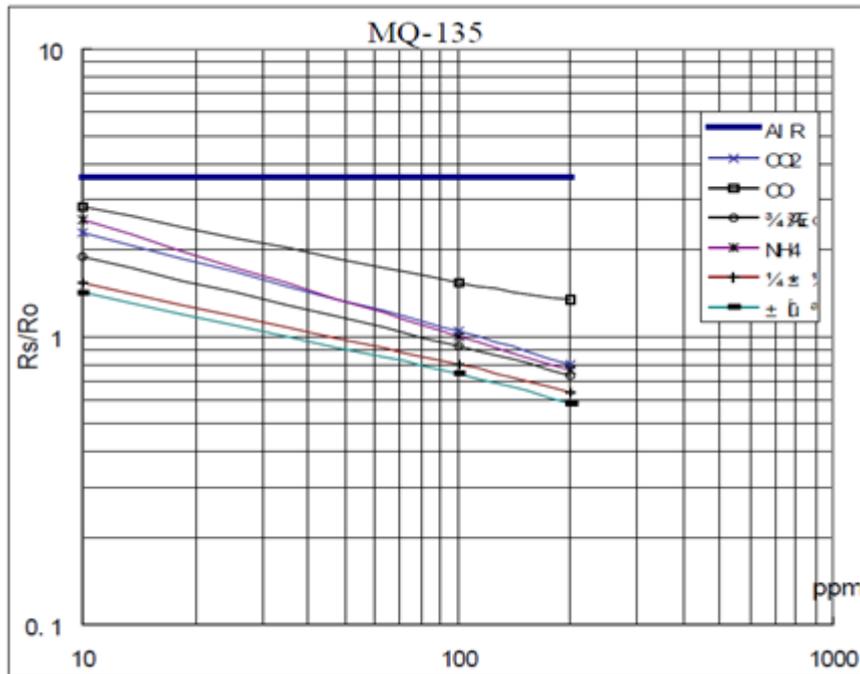


Figure 3-4 . Caractéristique de la sensibilité du capteur [14]

R_0 : Résistance de capteur à l'air pur.

R_s : Résistance de capteur à diffèrent concentration de gaz.

Ces graphes sont des fonctions de puissances avec

$$y = ax^b \tag{3.1}$$

$$ppm = a\left(\frac{R_s}{R_0}\right)^b \tag{3.2}$$

L'unité de mesure utilisé est la ppm, Pour avoir la valeur en ppm, on doit chercher les valeurs de a et b , pour avoir des valeurs plus précis on utilise l'Excel.il suffit de retracer le courbe dans le graphe et ensuite afficher son équation. La figure 3-5 suivante présente la courbe de CO et son équation en utilisant l'Excel.

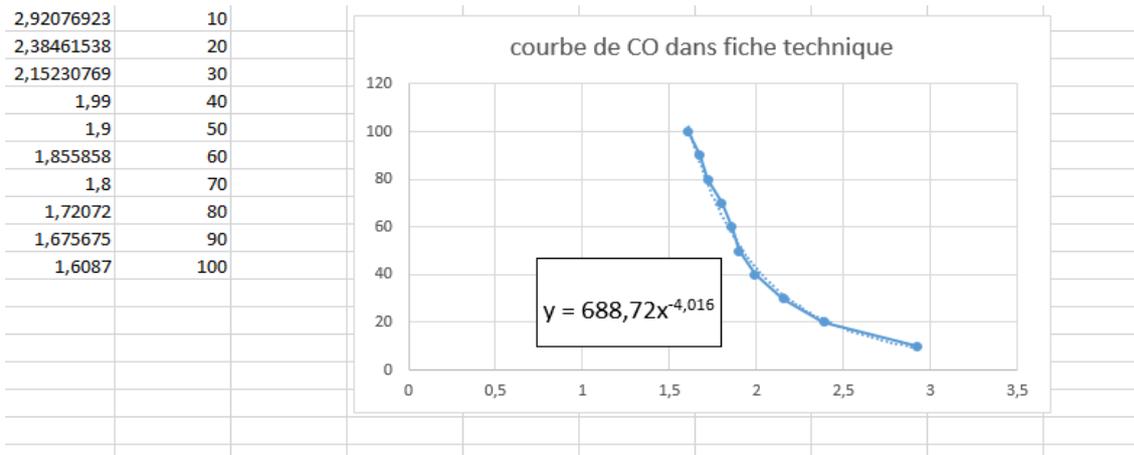


Figure 3-5 La courbe de CO

Alors Pour CO $a = 688.72$ et $b = - 4.016$

c. Calibration de MQ135

Il faut connaître la valeur en ppm de gaz à détecter dans l'air pur. On l'obtient en utilisant l'équation (3.2).

$$R_0 = R_s \sqrt[b]{\frac{a}{ppm_0}} \quad (3.3)$$

ppm_0 : la valeur de gaz à détecter dans l'air pur.

La fiche technique de MQ135 se trouve dans l'Annexe II. On note que 95% des capteurs de pollution de l'air ne détectent que la valeur située au point où ils se situent et le MQ135 en fait partie.

3.2.3 L'envoi de données

Il existe deux moyen pour envoyer des données envoi de donnée par socket ou en utilisant le protocole Hypertext Transfer Protocol (HTTP).

a. Le principe de socket

Les sockets sont basés sur une architecture client/serveur: Le serveur décide d'accepter les demandes de connexion sur un port particulier, tandis que le client décide de se connecter sur le serveur sans pour autant être sûr que cela sera possible. Une fois la connexion établie, les deux protagonistes peuvent communiquer à l'aide de read et write, exactement de la même manière que lorsque l'on lit/écrit dans un fichier binaire.

Cette méthode est la plus utilisée pour envoyer des données mais il existe certaines plateformes de cloud qui n'acceptent pas l'envoi de données par socket.

b. Le protocole HTTP

Le protocole HTTP est inventé par Tim-Berner Lee au début des années 1990, il fonctionne sur le principe "requête-réponse". La figure suivante présente le principe de la communication client-serveur avec le protocole HTTP.

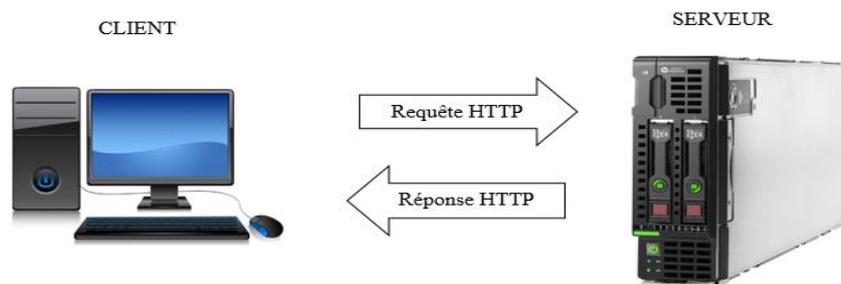


Figure 3-6 Le principe client Serveur avec le protocole HTTP

Il existe deux méthodes permettant d'envoyer les données vers le serveur :

- La méthode Get : envoi de donnée par l'url.
- La méthode POST : envoi de donnée à travers un formulaire.

c. Le module ESP8266

Les deux moyens cités précédemment ne sont que des moyens utilisés côté programmation. Nous ne pouvons pas connecter à l'internet sans le support physique. Parmi les technologies sans fil existant, on peut choisir entre les modules wifi ESP 8266.

Il existe plusieurs versions d'ESP8266. Ce qui les différencie c'est le nombre de pins accessibles, le type d'antenne, ou encore la quantité de mémoire Flash. La figure 3-7 suivante présente le module ESP8266.

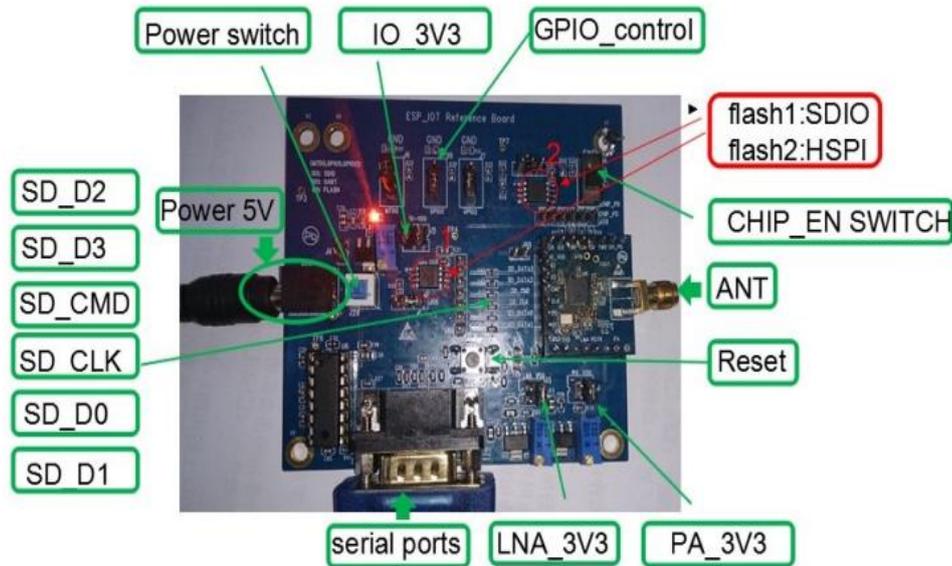


Figure 3-7 Le module ESP 8266 [15]

Les caractéristiques communes à toutes les versions

- Tension de fonctionnement: 3.3v [15]
- Fréquence CPU: 80 MHz [15]
- RAM: 64Ko pour le code, 96Ko pour les données [15]
- Wi-Fi: b/g/n, WEP ou WPA/WPA2. [15]
- Antenne: intégrée. [15]
- Consommation: entre 60mA et 200mA en fonctionnement normal, quelques dizaines de μ A en veille. [15]

On commande l'ESP8266 par la commande AT, la liste des commandes AT est présentée à l'annexe III. Cette commande peut être utilisée dans le code arduino ou en mode console.

3.3 La partie informatique du système

Pour la partie informatique, on utilisera l'offre Paas de google: le Google App Engine, cette offre permettant de bénéficier de nombreuses offres de services web. Les outils utilisés sont le langage de programmation Java EE, le Google App Engine et la base de données datastore.

3.3.1. Le langage de programmation Java EE

La famille Java Enterprise offre la possibilité de construire une application n tiers qui répond à la norme des applications entreprise. [16]

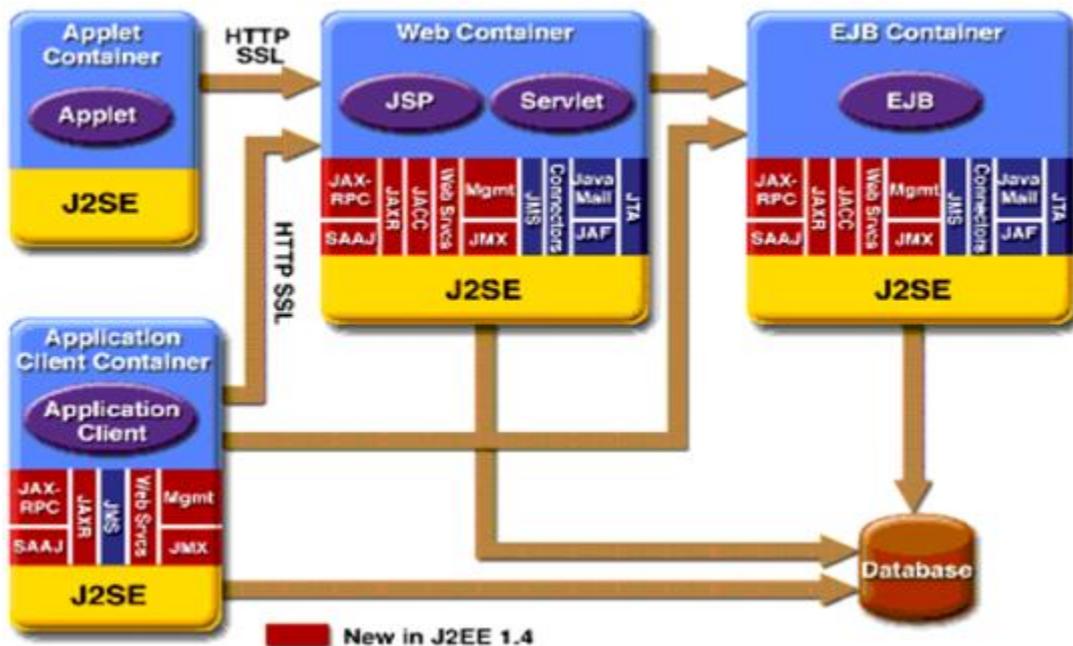


Figure 3-8 Les fonctionnalités de Java EE [16]

Elle est construite à partir de la standard édition. Dans la couche la plus externe, elle se différencie du J2SE par les technologies suivantes : HTTP, EJB, Servlet, JSP. [16]

a. HTTP

Cette norme permet de véhiculer des entités serialisables sur un support de réseau [16]. Elle offre donc la possibilité de relier les modules de chaque conteneur (web, EJB, Application Windows, Applet).[16]

b. Enterprise Java Beans (EJB)

L'EJB est une entité d'application qui tourne dans un serveur d'application [16]. Chaque Beans est indépendant les uns des autres. Ces entités permettent de fournir des services réutilisables par toutes les applications qui utilisent le conteneur.[16]

Il existe trois types d'EJB

- Les EJB Session fournissent un service aux clients [16]. Ils sont accessibles via le réseau (en passant par le protocole RMI) ou localement dans la JVM du client [16]. Les EJB sessions sont les points d'entrée de la couche métier. [16]
- Les EJB Entity représentent les données qui proviennent ou qui alimentent la base de données [16]. Ils ne sont généralement pas accessibles directement au client [16]. Ce dernier doit, traditionnellement, passer par un EJB session pour récupérer des EJB entités. [16]
- Les EJB MessageDriven représentent une file de messages postés par le client et traités par le serveur (ou vice-versa). [16]

c. Servlet

Un servlet est une entité d'application qui implémente la technologie http [16]. Il permet de gérer donc les différentes requêtes d'une application cliente et de fournir la réponse sur canal. [16]

d. Java Server Page (JSP)

Le JSP est un ensemble de script HTML et de scriptlet en Java [16]. Après compilation par le JDK qui se trouve dans le serveur, un JSP devient un servlet. [16]

3.3.2. Le serveur web

Avec Google App Engine, le logiciel est stocké dans le serveur de Google, les Datacenter de Google pour effectuer nos calculs. La différence entre un Datacenter et un serveur classique est qu'on ne loue plus un serveur mais une puissance.

L'intérêt de ce fonctionnement, c'est que nous n'avons plus à nous préoccuper des problèmes physiques des machines. Si un disque dur tombe en panne, ce n'est pas à nous d'aller le remplacer. Nous avons juste besoin de demander à notre site de démarrer sur une autre machine. Si le trafic de notre site augmente, le Datacenter va automatiquement augmenter le nombre de serveurs.

3.3.3. Le service Cloud : Google App Engine

Google App Engine est une des solutions cloud de Google. C'est un service qui permet d'utiliser les mécanismes du "cloud computing" proposé par Google. Il nous permet d'héberger notre site sur les serveurs de Google et de bénéficier d'une puissance qui peut s'adapter au trafic du site.

a. Fonctionnement du Google App Engine

Le rôle d'App Engine est de "masquer" le fonctionnement et la complexité des serveurs de Google. Lorsqu'un visiteur se connecte au site, il arrive sur le répartiteur de charge (load balancer) de Google, qui va chercher un serveur disponible et pas trop chargé pour gérer la demande du visiteur. Si le site a besoin d'accéder à des données, ce qui est fréquent, il fera appel à une autre zone de serveurs appelée DataStore (c'est en quelque sorte la base de données). Tout ceci est résumé dans ce schéma fourni par Google.

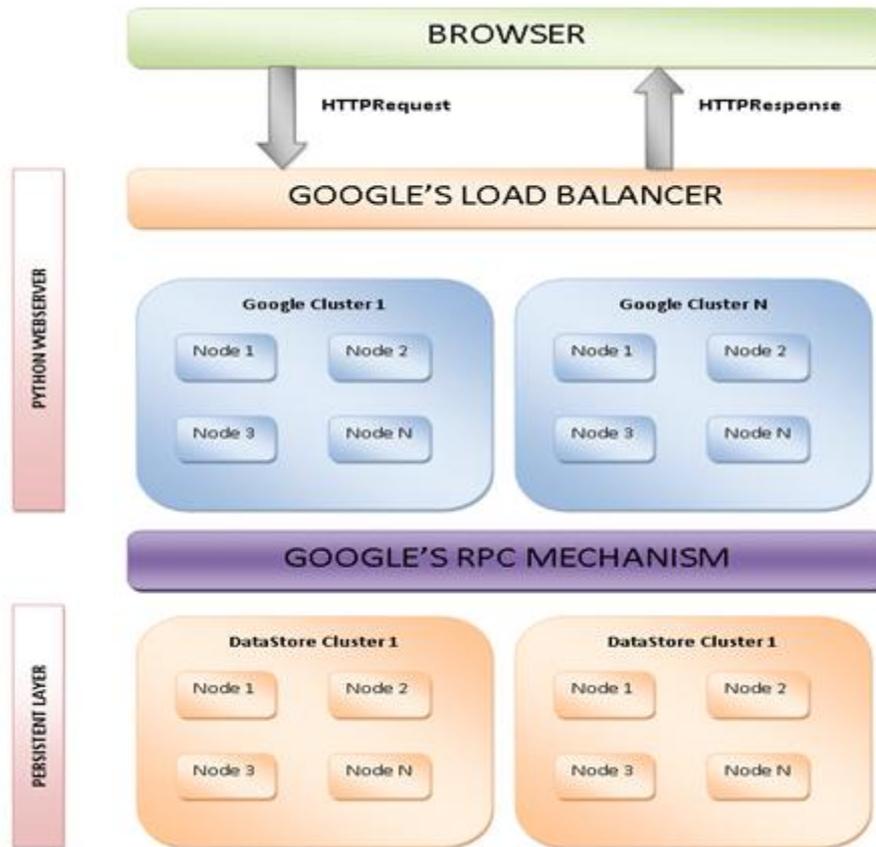


Figure 3-9 Fonctionnement de Google App Engine[17]

b. Outils de développement

Comme on va utiliser le langage JAVAEE pour notre application web, alors, on va utiliser Eclipse pour la phase de développement.

L'API Google App Engine se présente sous forme d'un plugin d'Eclipse. Son installation se fait depuis Eclipse en suivant : Help > Install New Software. Et d'entrer l'url **<http://dl.google.com/eclipse/plugin/x,y>** dans le champ « ou trouver des plugins ». Une fois que c'est fait, les listes de plugins disponibles se chargent.

Enfin, on sélectionne les plugins pour Google App Engine et on les installe. Voici la liste qu'on doit cocher pour utiliser Google App Engine :

- Google Plugin for Eclipse
- SDK > Google App Engine Java SDK
- SDK > Google Web Toolkit SDK

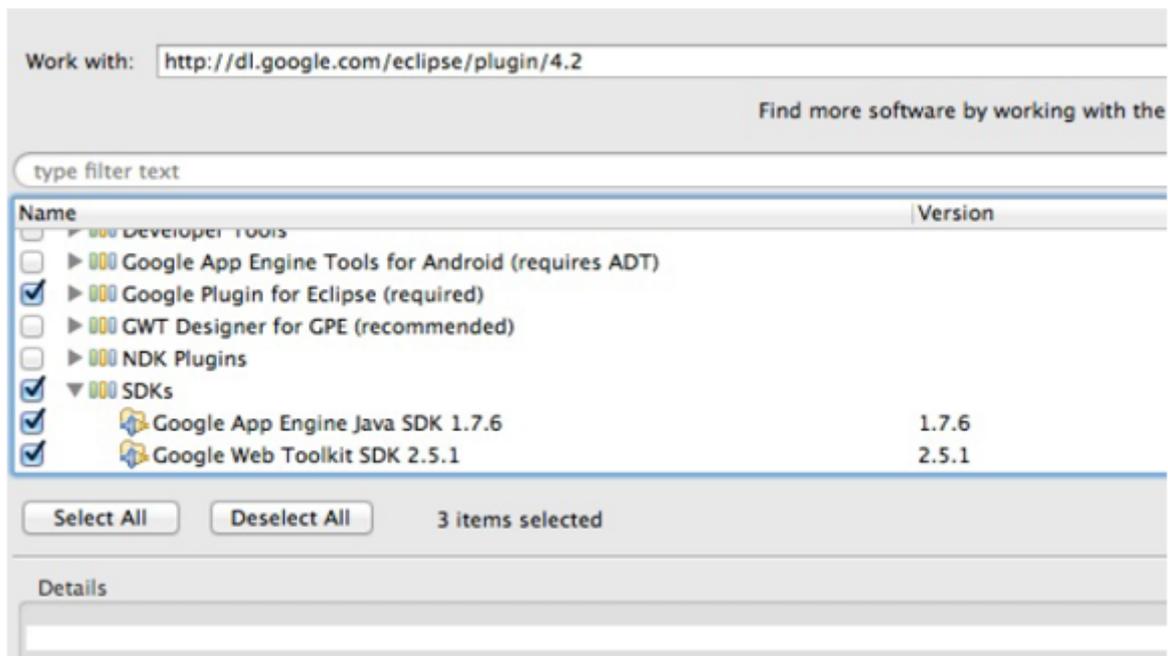


Figure 3-10 Installation de Google App Engine

3.3.4. Le service de stockage utilisé

Le stockage de données est très important dans le développement d'application web pour gérer toutes informations entrantes et sortantes.

Le Datastore est un des services de stockage d'App Engine, c'est une base de données NoSQL. On l'a choisi parce qu'il est idéal pour le stockage en masse ou Big Data. En effet, on va stocker les valeurs prises par le capteur à chaque minute dans notre base de données donc notre volume de données augmente très rapidement qu'il est plus lent d'utiliser Google Cloud SQL pour le traitement et le stockage de données.

a. La base de données NoSQL

NoSQL (Not only SQL en anglais) désigne une catégorie de systèmes de gestion de bases de données (SGBD) qui n'est plus fondée sur l'architecture classique des bases relationnelles. L'unité logique n'est pas une table, et les données ne sont en général pas manipulées avec SQL. [18]

La façon dont la base NoSQL stocke ses données est très simple car il se résume comme un simple tableau associatif, donc chaque données ont une clé et une valeur.

Le système « clé-valeur » comporte quatre modèles de stockage de données à savoir :

- Clé-valeur: Collection de couples (clé, valeur) (key-value pairs) = tableau associatif ;
- Colonne: Modèle clé-valeur permettant de disposer de plusieurs valeurs ;
- Document : Modèle clé-valeur permettant de disposer d'une valeur complexe (Un document), chaque document étant composé de champs et de valeurs associées ;
- Graphe: Gestion de relations multiples entre les objets.

On note que la SGBD NoSQL ne remplace pas les SGBDR mais les complète en palliant leurs faiblesses. Le tableau 3-1 présente la comparaison entre les modèles SGBD NoSQL et SGBDR.

Tableau 3-1 Comparaison entre SGBDR et modèles de SGBD NoSQL [19]

	Data Model ⇄	Performance ⇄	Scalability ⇄	Flexibility ⇄	Complexity ⇄	Functionality ⇄
●	Key-Value Store	high	high	high	none	variable (none)
●	Column-Oriented Store	high	high	moderate	low	minimal
●	Document-Oriented Store	high	variable (high)	high	low	variable (low)
●	Graph Database	variable	variable	high	high	graph theory
	Relational Database	variable	variable	low	moderate	relational algebra

b. Le Datastore et le détail de la conception de base de données

Le Datastore est clairement notre préférence pour stocker nos données sur Google App Engine. Il est rapide, capable de gérer de fortes charges, réparti sur de multiples datacenters, il est basé sur Big Table, la technologie qu'utilise Google pour stocker toutes ses données, notamment les résultats d'indexation des milliards de pages web par le GoogleBot.

i. Fonctionnement du datastore

Le Datastore fonctionne en mode High Replication Datastore (HRD). Ce système est très résilient aux erreurs : il fait en sorte que les données soient copiées sur de multiples datacenters. L'objectif est que les applications puissent continuer à fonctionner même en cas d'évènement catastrophique.

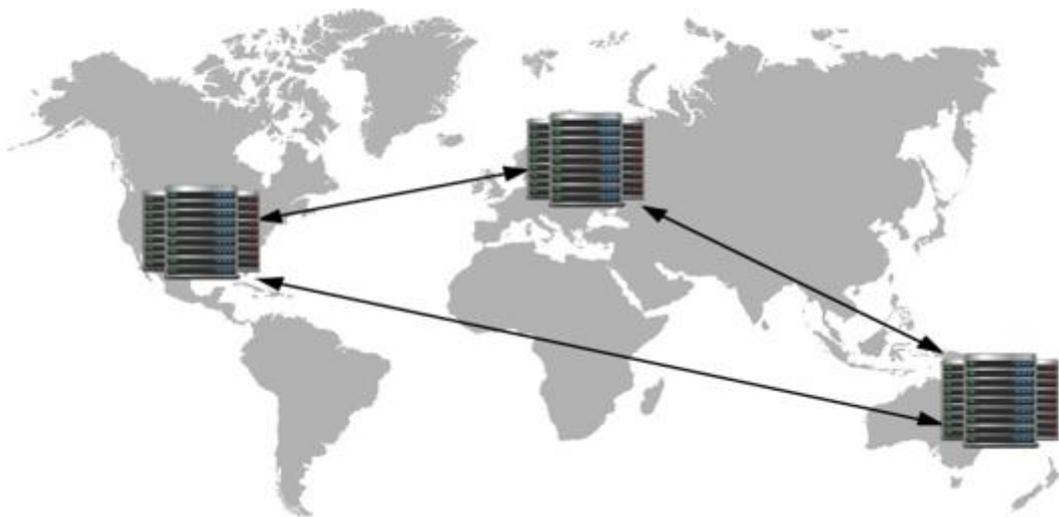


Figure 3-11 Fonctionnement du datastore [17]

Le datastore est une SGBD NoSQL, il est non relationnel. Le Datastore est beaucoup plus basique et de bas niveau qu'une base de données classique. Il est comme un espace géant prêt à accepter n'importe quelles données. On lui envoie le contenu et il le stocke. Quand vous en avez à nouveau besoin, il vous suffit de le lui demander.

Le Datastore n'utilise pas de tables. Il n'y a pas de schéma SQL : on dit que le Datastore est schemaless (sans schéma). On n'a pas besoin de décrire comment sont formées les données avant de les stocker, ou de définir les relations entre table.

Le Datastore peut être utilisé de façon massive, avec beaucoup de données et de requêtes simultanées. Pourtant, à son plus bas niveau, il fonctionne de façon très simple : il accepte des clés et des valeurs. Exactement comme un tableau associatif

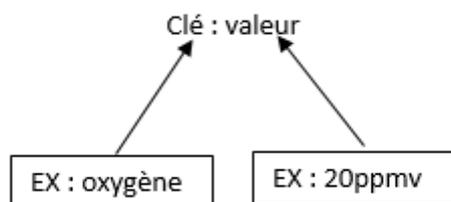


Figure 3-12 Forme d'information stockée au datastore

Le Datastore est capable de stocker beaucoup de paires clé-valeur et comme ce n'est pas structuré d'avoir des millions voire des milliards des paires « clé : valeur », le Datastore a un concept d'entités pour organiser un peu les données stockées.

Une entité est un groupement d'une ou plusieurs paires « clé : valeur »

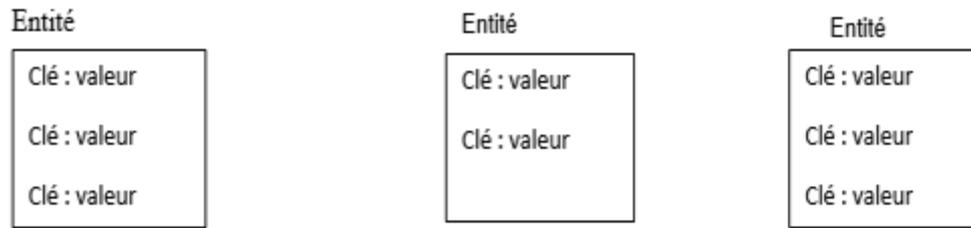


Figure 3-13 Les entités dans le Datastore

Le Datastore est aussi capable d'associer les entités entre elles. Une entité peut avoir une entité parente, qui peut elle-même avoir une entité parente, etc.

ii. Récupération d'une entité

Les entités sont des groupes de données qui vivent dans le Datastore. Il peut y en avoir une très grande quantité. Parfois, certaines entités sont reliées entre elles : elles ont une relation de parenté.

Les données (paires clé-valeur) au sein des entités sont appelées des propriétés. Celles-ci stockent obligatoirement des données basiques, comme du texte ou des nombres.

Pour repérer une entité, il faut leur donner un nom unique pour les repérer. Pour identifier une entité, Google utilise la combinaison de trois informations :

- Un type: c'est le type de données de l'entité
- Un identifiant: il peut s'agir soit d'une chaîne de caractères soit d'un nombre.

On peut le définir ou bien laisser Google générer un nombre par défaut.

- Un parent (l'ancestor path): si l'entité possède un parent, celui-ci aidera à identifier l'entité de façon unique. Si l'entité ne possède pas de parent, on dit que c'est une entité racine (root). Dans ce cas l'ancestor path vaut nul.

iii. Les opérations de base de Datastore

Il y a quatre opérations de base disponibles dans une base de données NoSQL comme le Datastore :

- get() : récupère une entité en fonction de son identifiant
- put() : écrit une entité (ou en écrase une)
- delete() : supprime une entité via son identifiant
- query() : retrouve des entités en fonction de certains critères de recherche

iv. Conception de la base de données utilisé

La base utilisée est donc très simple à présenter puisqu'on va simplement stocker le nom de polluants et sa valeur dans le datastore donc on peut les dessiner comme présent la figure 3.10 ci-dessous.

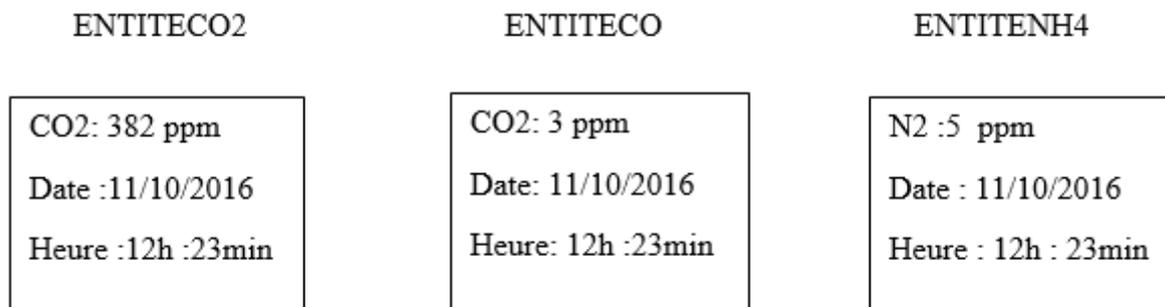


Figure 3-14 Les entités pour le système de détection de la pollution atmosphérique

Pour créer une entité, 3 opérations sont nécessaires :

- Créer l'objet représentant l'entité, de type Entity ;
- Assigner-lui une ou plusieurs propriétés (ce sont les paires clé-valeur)
- Demandez l'écriture de l'entité au Datastore avec la méthode put ()

Les outils pour développer sont déjà intégrés dans le plugin Google App Engine. La figure 3-15 ci-dessus présente le programme pour créer une entité.

```

import java.io.IOException;
import javax.servlet.http.*;

import com.google.appengine.api.datastore.*;
//chargement du APIs datastore
// Récupération de l'objet datastore
|

@SuppressWarnings("serial")
public class EntiteServlet extends HttpServlet {
    public void doGet(HttpServletRequest req, HttpServletResponse resp) throws IOException {
        DatastoreService datastore = DatastoreServiceFactory.getDatastoreService ();
        // Crée l'entité de type (kind) " ENTITECO2"
        Entity ENTITECO2 = new Entity ("ENTITECO2");
        // Assigne des propriétés à l'entité
        ENTITECO2.setProperty("CO2", "382 PPMV");
        ENTITECO2.setProperty ("date", "11/10/2016");
        ENTITECO2.setProperty("heure", "12h:23min");
        // Enregistre l'entité dans le Datastore
        datastore.put (ENTITECO2);

    }
}

```

Figure 3-15 Programme Java EE pour créer une entité

3.3.5. Le frontal web

L'application web comporte un contrôleur chargé d'écouter les appels http faites par le capteur. C'est grâce à cet appel que l'application peut accéder aux valeurs envoyées par le capteur et ensuite, les enregistrer dans la base de données. Les utilisateurs ne doivent pas, savoir l'existence de cette page. Par contre, ils ont le droit de voir tous les autres pages et fonctionnalités de l'application.

On accèdera à l'application par tous les appareils ayant un navigateur web: PC, smartphone, tablette. Notre frontal web doit accepter la norme suivant le besoin. Ainsi il peut afficher la liste des valeurs de gaz en un instant choisit par l'utilisateur. On peut aussi surveiller l'augmentation des valeurs des polluants. Il peut aussi envoyer des notifications lorsqu'il y a des polluants et indiquer les menaces. L'application serait accessible en mode SaaS.

Les outils utilisés pour le développement du frontal web sont :

- HTML5 pour le squelette de la page
- CSS3 et le Framework Bootstrap pour le style.

3.4 Conclusion

Notre système est basé sur l'utilisation des technologies nouvelles. La partie matérielle est basée sur l'arduino, une carte électronique à microcontrôleur qui est très utilisée aujourd'hui dans le domaine de la programmation embarqué, on l'utilise pour établir la relation entre le MQ135 et l'ESP01. Pour la partie logiciel, on utilise le « google apps engine » le Paas de google comme plateforme de service qui est aussi une technologie nouvelle dans le domaine de l'informatique. Le chapitre suivant va présenter la réalisation du système.

Chapitre 4. REALISATION DU « SYSTEME DE DETECTION DE LA POLLUTION DE L’AIR »

4.1. Structure détaillé du projet

On va créer un capteur composé de capteur de gaz MQ135, le module wifi ESP01 et un arduino. Après avoir détecté la valeur des gaz, il suffit de connecter ce capteur à l’internet via un routeur et d’envoyer les données par URL à l’aide de la méthode GET du protocole HTTP. Ensuite les données sont récupérées par le cloud et peuvent être visualisées par les utilisateurs à l’aide d’une interface web. Ce chapitre va présenter la réalisation de ce système. La structure de projet est présentée ci-dessous par la Figure 4-1.

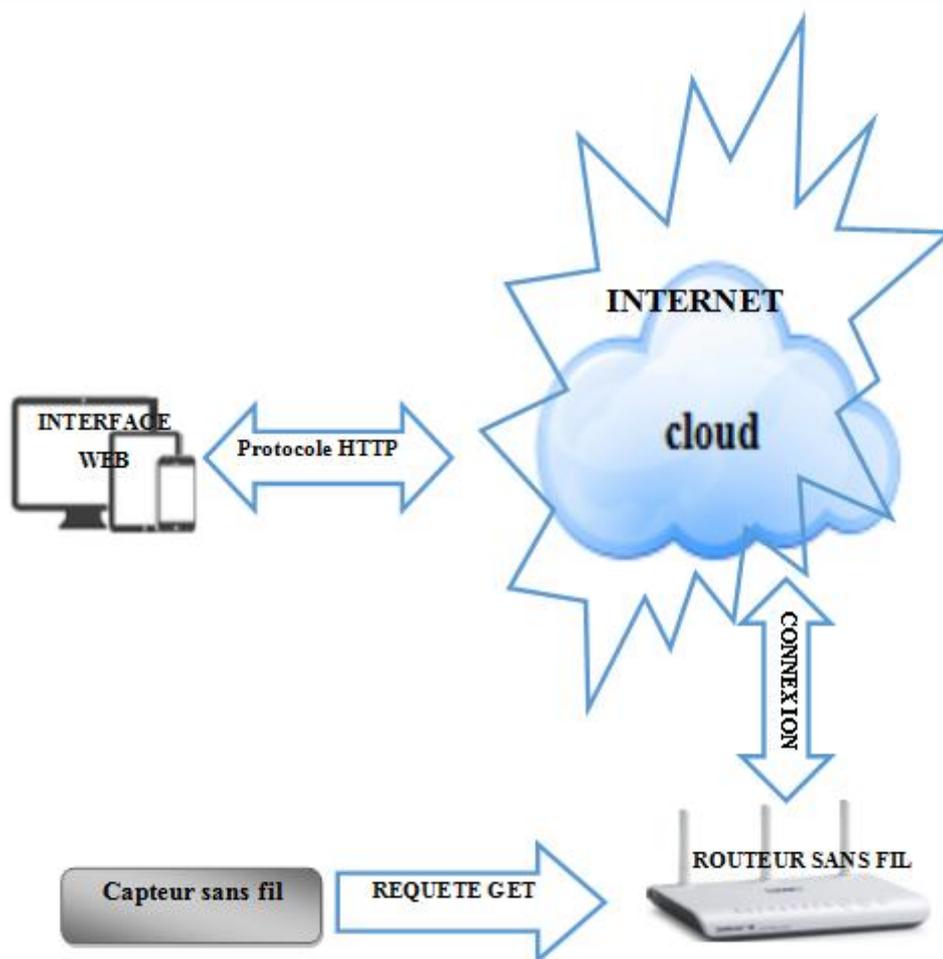


Figure 4-1 Structure de projet

4.2. La partie électronique du projet

4.2.1 Les capteurs

Le capteur électronique détecte les valeurs des gaz toxiques et les gaz qui peuvent devenir toxiques dont l'unité de mesure est en ppm.

Ce capteur est constitué par :

- les capteurs de gaz « MQ135 »

Le projet cible principalement l'atmosphère d'Antananarivo alors il faut utiliser le capteur spécifique aux polluants de l'atmosphère de la ville. Les principaux polluants sont les Gaz carboniques (CO₂), les monoxydes de carbone (CO) émis par les véhicules, et les ammoniacs (NH₃) émis par les ordures. On a utilisé des capteurs MQ135. Les spécifications techniques de ce capteur permettent de détecter les gaz cités auparavant.

- Le module Wifi ESP01 est large et utilisé avec la carte arduino, le module est la responsable de la communication entre l'appareil et le routeur.
- La carte arduino uno est utilisée pour contrôler le MQ135 et l'ESP01.
- Le projet est alimenté par la tension 5V et 3.3 V, le circuit d'alimentation est constitué par un batterie de 12V, un LM1117 pour régler le tension à 3.3v et le LM7805 pour le 5V. La Figure 4-2 suivante présente le circuit d'alimentation.

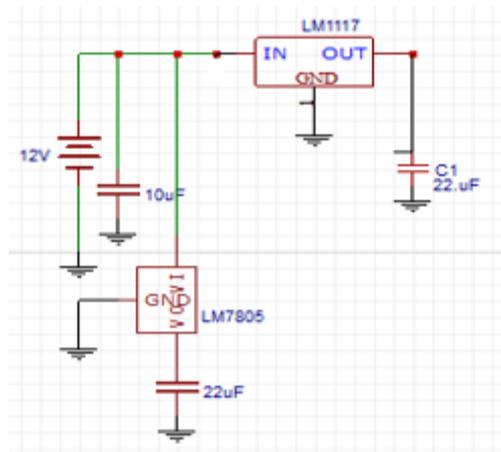


Figure 4-2 Le circuit d'alimentation pour le projet

4.2.2 Structure du montage

Le branchement de toutes les matérielles s'effectuent comme suit :

- Vcc de MQ135 à 5 V, Vcc et CHPD de l'ESP 01 à 3.3 V
- GND vers GND
- Le DOUT de MQ135 vers l'entrée analogique
- Le RX de l'ESP01 vers le pin 11 de l'arduino, et TX vers le pin 10 de l'Arduino.

La figure 4-3 ci-dessous montre le Schéma du montage.

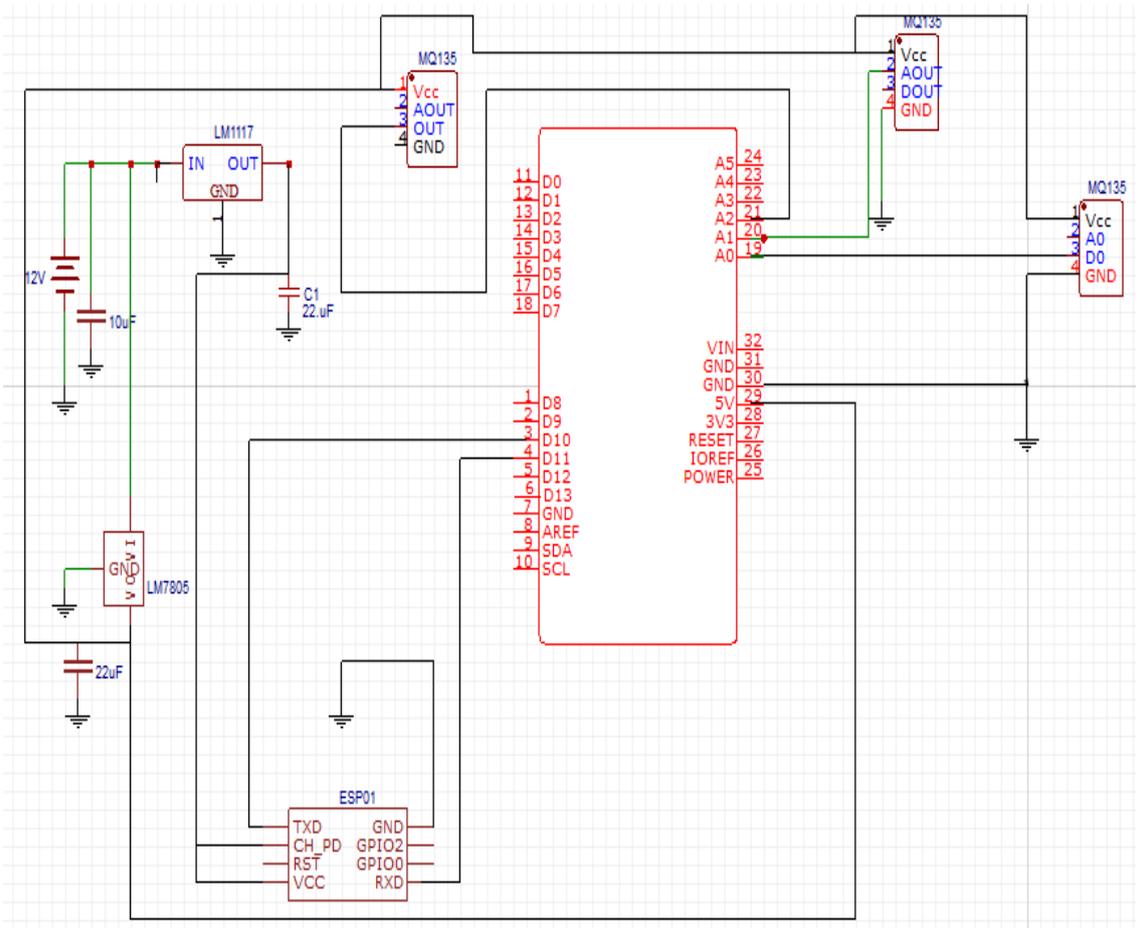


Figure 4-3 Schéma du montage

4.2.3 Le Firmware (Le programme à introduire dans l'arduino)

On utilise les courbes de CO, de CO₂ et de NH₃ dans le Figure 3-3 pour chercher la valeur en ppm, on applique l'équation 3.2.

- La fonction float getResistance (); retourne la résistance de l'équation (3.3).
 - La fonction float getPPM (float parA, float rzero, float parB); de la classe Mq135 nous permet de recevoir la valeur de gaz en ppm. Les paramètres parA est la valeur a et parB est la valeur b de l'équation (3.2). Les valeurs a et b pour chaque gaz sont obtenu en utilisant la puissance de régression. Pour CO : a = 740.75 et b = - 4.201, Pour CO₂ : a = 116.60 b = - 2.769, pour NH₄ : a = 112.88 b = -2.491
 - Le paramètre rzero de fonction getPPM est la valeur R_0 de l'équation (3.3), cette valeur est obtenu en utilisant la fonction suivant : float getRZeroS (float atm, float parA, float parB); de la classe Mq135 avec atm est la valeur de gaz dans l'air pur, atm pour CO₂ = 400.00ppm, pour CO = 0.1 ppm et atm pour NH₄ = 0.15 ppm
 - La fonction void sendValue(String CO₂, String CO, String NH₄) va envoyer les donnée. Les paramètres String CO₂, String CO, String NH₄ sont les valeurs à envoyer.
- Le code complet se trouve dans l'annexe, voici l'organigramme de firmware.

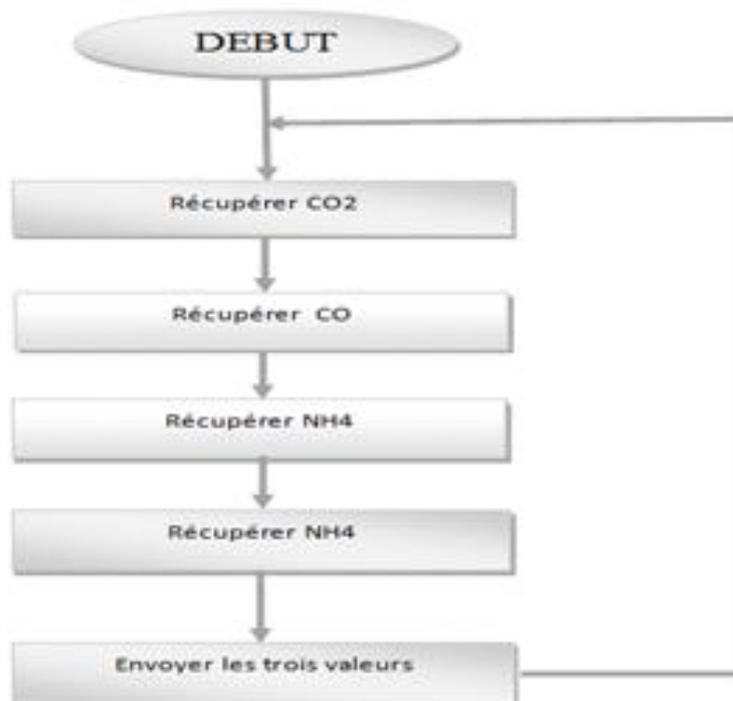


Figure 4-4 Organigramme simplifié du firmware de l'Arduino

4.3. La partie informatique

Les valeurs captées par les capteurs sont envoyées à l'internet, et récupérées par un serveur, ainsi les utilisateurs peuvent visualiser les données dans une interface web. La structure de l'application web se présente comme suit.

On distingue quatre fonctionnalités sur l'application :

4.3.1. Accueil

Cette fonctionnalité affiche les dernières valeurs reçues, la date d'obtention et les seuils ne doit pas être dépassé. La fonction responsable de la récupération est `acc.recevoirAccueil (req)` ; implémentation de cette fonction se trouve à l'annexe. La figure 4-5 présente la page d'accueil de l'interface.



Figure 4-5 La page d'accueil

Avec cette fonctionnalité, on peut voir aussi les courbes d'évolution de la pollution pour chaque gaz en cliquant sur le bouton courbe. La figure 4-6 suivante présente la courbe d'évolution de gaz carbonique.



Figure 4-6 Courbe d'évolution de CO2

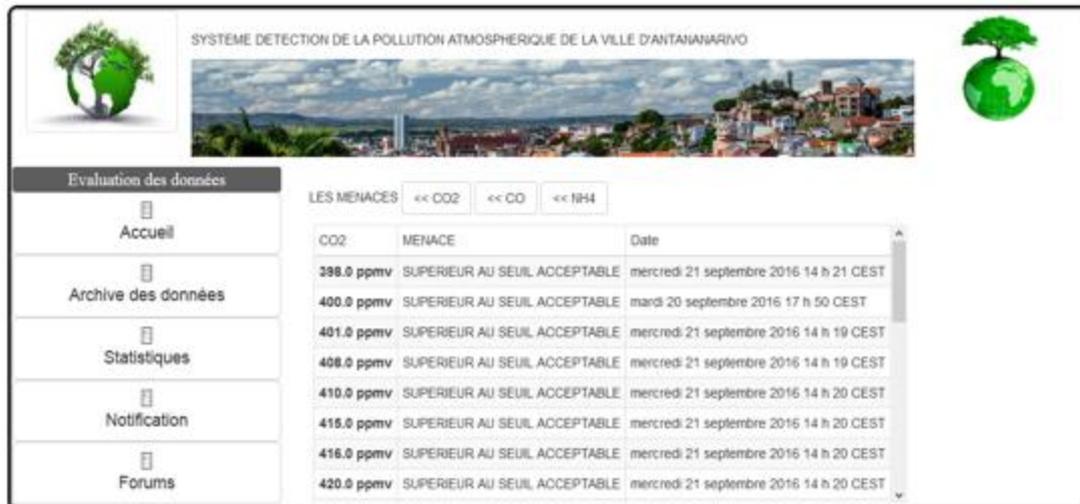
4.3.2. Archive des données

On peut voir les valeurs de gaz dans la base de données, il suffit de cliquer sur le formulaire et remplir les deux champs par des dates. La fonction demande. Récupérer (req); permet de télécharger les données dans la base des données.

Figure 4-7 La Page de téléchargement

4.3.3. Le Notification

Il existe une valeur acceptable pour chaque gaz toxique et si la valeur est supérieure au seuil acceptable, le gaz a un effet sur la santé, dans ce cas il faut alerter les utilisateurs. La notification consiste à afficher des informations à chaque fois que cette valeur est dépassée. La figure 4-8 montre la liste de notification pour CO2.



CO2	MENACE	Date
398.0 ppmv	SUPERIEUR AU SEUIL ACCEPTABLE	mercredi 21 septembre 2016 14 h 21 CEST
400.0 ppmv	SUPERIEUR AU SEUIL ACCEPTABLE	mardi 20 septembre 2016 17 h 50 CEST
401.0 ppmv	SUPERIEUR AU SEUIL ACCEPTABLE	mercredi 21 septembre 2016 14 h 19 CEST
408.0 ppmv	SUPERIEUR AU SEUIL ACCEPTABLE	mercredi 21 septembre 2016 14 h 19 CEST
410.0 ppmv	SUPERIEUR AU SEUIL ACCEPTABLE	mercredi 21 septembre 2016 14 h 20 CEST
415.0 ppmv	SUPERIEUR AU SEUIL ACCEPTABLE	mercredi 21 septembre 2016 14 h 20 CEST
416.0 ppmv	SUPERIEUR AU SEUIL ACCEPTABLE	mercredi 21 septembre 2016 14 h 20 CEST
420.0 ppmv	SUPERIEUR AU SEUIL ACCEPTABLE	mercredi 21 septembre 2016 14 h 20 CEST

Figure 4-8 Représentation de la page notification CO2

Il suffit de cliquer sur le bouton, pour afficher les notifications pour chaque gaz. La liste des remarques avec les dates de réception de la valeur est présentée dans la page notification. Les fonctions affichent les notifications sont :

- void public ListeNotifCO2 (HttpServletRequest req) ; affiche la liste de notifications pour CO2
- void public ListeNotifCO (HttpServletRequest req) ; affiche la liste de notifications pour CO
- void public ListeNotifNH4 (HttpServletRequest req) ; affiche la liste de notifications pour NH3.

4.3.4. Statistique

Trois informations sont présentées sur le menu statistique, premièrement le pourcentage de CO₂, CO, et NH₃ non normal. Le pourcentage des valeurs non acceptable pour chaque gaz est exprimé à l'aide de graphe comme présenté à la figure 4-9 ci-dessous.

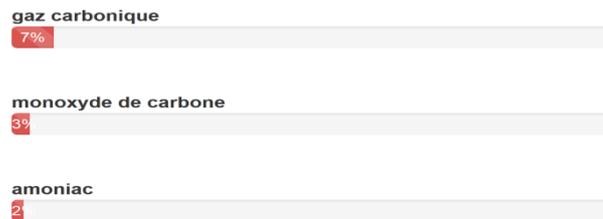


Figure 4-9 Pourcentage des gaz non acceptable

c. Calcul du pourcentage de gaz non acceptable

Condition pour que le gaz est à l'état normal

- CO₂ < 396ppm[20]
- CO < 3ppm[20]
- NH₃ < 5ppm[20]

d. Calcule de pourcentage d'un gaz P_{gaz}

$$P_{gaz} = n * \frac{100}{N} \quad (4.1)$$

n : nombres de valeur non acceptable

N : nombres totales de valeur d'un gaz reçus

Si le pourcentage est égal à 0%, on se trouve dans un endroit bien aéré. Par contre si l'une ou toute la valeur est supérieure à 0% ça implique qu'il y a un moment où l'état du gaz a été changé à cause de passage d'un polluant à côté du capteur, pendant ce temps-là la valeur de l'état de gaz en ppm dépasse la valeur acceptable.

La valeur maximale de chaque gaz en ppm est présentée en bas de la page. Ces valeurs sont automatiquement affichées avec le graphe.

A droite du graphe, on rencontre un formulaire. On peut afficher la statistique entre deux dates en entrant la date de début et la date finale dans le formulaire. La figure 4-10

présente la page Statistiques. Par défaut ces sont les taux des toutes les valeurs des gaz existants dans la base et les valeurs maximales par rapport à toutes les valeurs sont affichés.

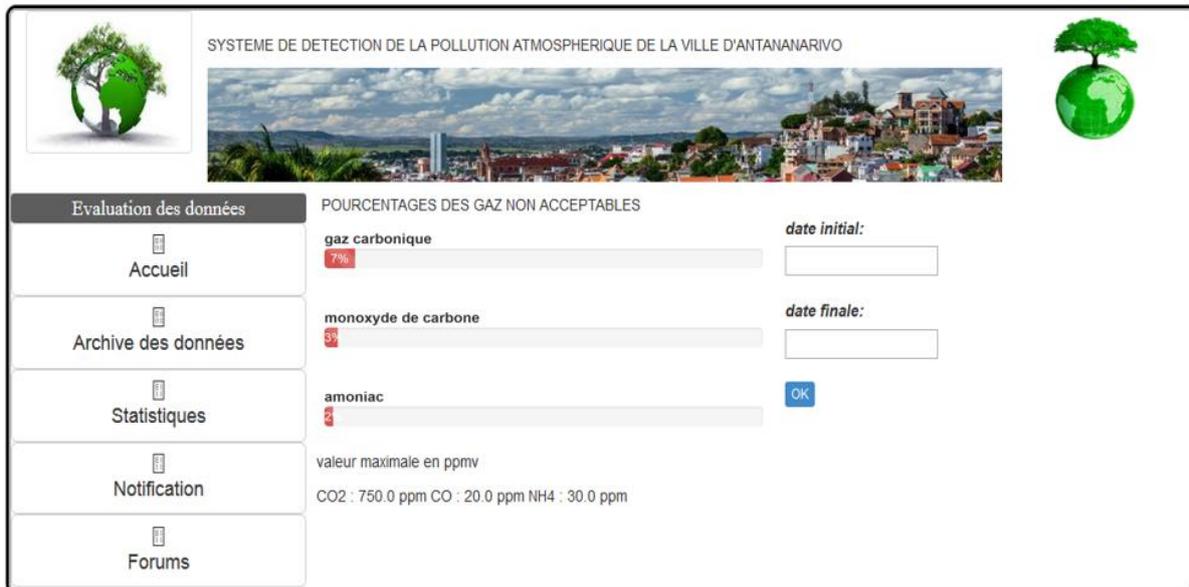


Figure 4-10 Présentation de la page statistique

4.4. Conclusion

Dans cette partie, on s'est surtout focalisé sur la mise en pratique de tous ce qu'on a vus aux chapitres précédents. L'utilisation des composants principaux capteurs MQ135, carte arduino uno et module wifi esp8266 permet de poursuivre à chaque instant donné l'état de l'air dans un lieu précis. La captivité est assurée par les MQ135 et l'envoi par l'ESP8266. La carte arduino uno est utilisée pour mettre l'ensemble en harmonie. On a vu un système complet et fonctionnel mais pour couvrir une ville, il faut utiliser plusieurs capteurs et les mettre sur des points stratégiques pour assurer la fiabilité. Ces capteurs envoient leurs données vers un nœud principal qui effectue un calcul sur ces données et envoie le résultat vers le serveur. Ainsi, même en utilisant plusieurs capteurs, l'étude se résumera toujours à ce que nous avons étudié.

CONCLUSION GENERALE

Après avoir vu l'importance de l'atmosphère et les mauvais effets de la pollution atmosphérique, on constate qu'on a besoin d'un système permettant de détecter la pollution de l'air pour la surveillance de l'état de l'air. On a réalisé un système de détection de la pollution atmosphérique pour résoudre ce problème.

Grâce au développement des technologies d'aujourd'hui, il existe des nombreux outils pour créer un système permettant de détecter les gaz polluants. Le système est réalisé avec le modèle IoT et peut être divisé en deux, le domaine de capteur contrôlé par l'arduino, avec le capteur de gaz mq135 et le module wifi ESP8266. Et l'application web avec la plateforme de service le PaaS de google « le google apps engine ».

Ce mémoire de fin d'études a développé notre connaissance sur l'internet des objets et l'utilisation de cloud computing. Ce capteur sans fil doit se connecter à l'internet afin que toutes personnes puissent visualiser les informations sur l'état de l'air à distance grâce à l'application web. Dans ce projet, On profite la puissance de calcul de cloud de google et sa capacité de mémoire illimité.

Le système de détection de la pollution atmosphérique réalisé fait connaitre le niveau de polluant existant et prévenir les utilisateurs. On préconise une amélioration en ajoutant le module GPS pour localiser le capteur, en augmentant le nombre de polluant à détecter, en multipliant le nombre de capteur et en ajoutant des fonctionnalités sur l'application.

ANNEXE I. CARACTERISTIQUE DES TECHNOLOGIES SANS FIL EXISTANT

Tableau I-1 Caractéristiques de technologie sans fil

	802 .15 .4	Zigbee	Wifi	GSM	Satellite
Porté	10 m	10	100	Infini	Infini
Capacité multihop	Non	Oui	Non	Non	Non
Consommation d'énergie	Faible	Faible	Faible	moyen	Haute
Sécurité	Non	Oui	Oui	Non	Non
Coût (matérielle)	Faible	Faible	Moyen	moyen	Moyen
Coût (service)	Gratuits	Gratuits	Gratuit	Haute	Moyen
Disponibilité	Bien	Bien	Bien	Moyen	Faible
Règlement	Bien	Bien	Bien	Bien	Faible

ANNEXE II. DATASHEET MQ-135

TECHNICAL DATA

MQ-135 GAS SENSOR

FEATURES

Wide detecting scope

Fast response and High sensitivity

Stable and long life

Simple drive circuit

APPLICATION

They are used in air quality control equipments for buildings/offices, are suitable for detecting of NH₃,NO_x, alcohol, Benzene, smoke,CO₂ ,etc.[15]

SPECIFICATIONS

Tableau II-1 Standard work condition [14]

Symbol	Parameter name	Technical condition	Remarks
V _c	Circuit voltage	5V±0.1	AC OR DC
V _H	Heating voltage	5V±0.1	AC OR DC
R _L	Load resistance	can adjust	
R _H	Heater resistance	33Ω±5%	Room Tem
P _H	Heating consumption	less than 800mw	

Tableau II-2 Environnement condition [14]

Symbol	Parameter name	Technical condition	Remarks
T _{ao}	Using Tem	-10□-45□	
T _{as}	Storage Tem	-20□-70□	
R _H	Related humidity	less than 95%Rh	
O ₂	Oxygen concentration	21%(standard condition)Oxygen concentration can affect sensitivity	minimum value is over 2%

Tableau II-3 Sensivity caracteristique [14]

Symbol	Parameter name	Technical parameter	Ramark 2
R_s	Sensing Resistance	30K Ω -200K Ω (100ppm NH ₃)	Detecting concentration scopell 10ppm-300ppm NH ₃ 10ppm-1000ppm Benzene 10ppm-300ppm Alcohol
α (200/50) NH ₃	Concentration Slope rate	≤ 0.65	
Standard Detecting Condition	Temp: 20 \square \pm 2 \square Vc:5V \pm 0.1 Humidity: 65% \pm 5% Vh: 5V \pm 0.1		
Preheat time	Over 24 hour		

D. Structure and configuration, basic measuring circuit

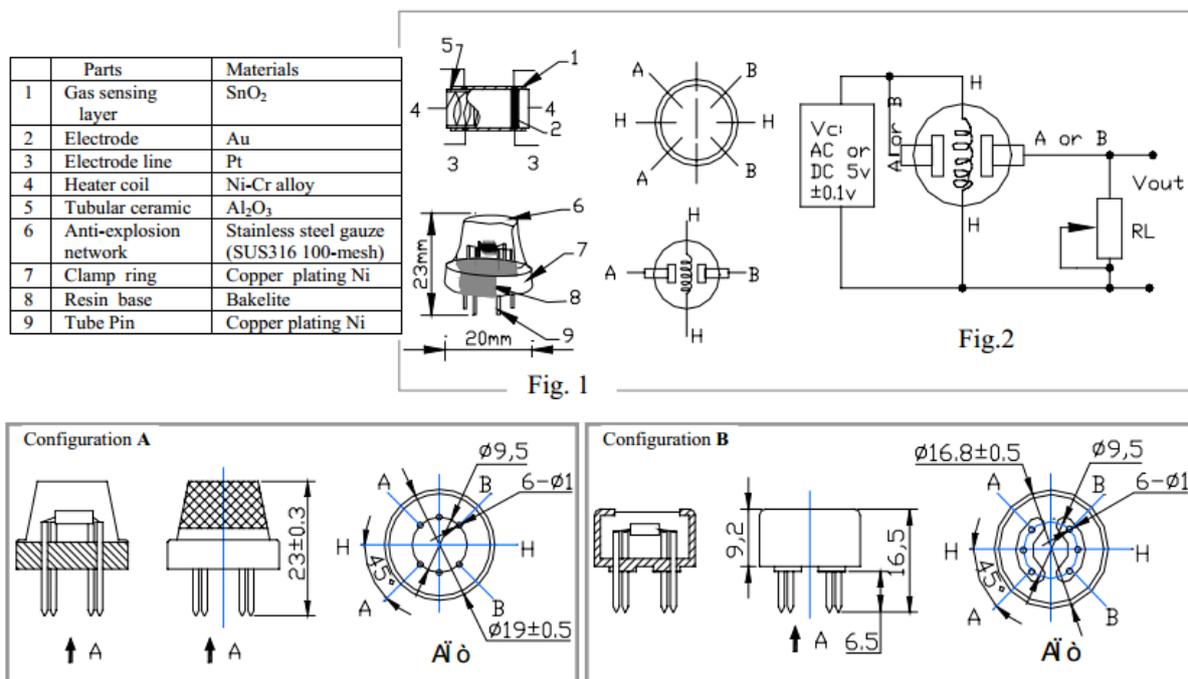


Figure II-1 Structure and configuration of MQ-135 [14]

Structure and configuration of MQ-135 gas sensor is shown as Fig. 1 (Configuration A or B), sensor composed by micro AL₂O₃ ceramic tube, Tin Dioxide (SnO₂) sensitive layer, measuring electrode and heater are fixed into a crust made by plastic and stainless steel net [14]. The heater provides necessary work conditions for work of sensitive components. The enveloped MQ-135 have 6 pin, 4 of them are used to fetch signals, and other 2 are used for providing heating current. Electric parameter measurement circuit is shown as Fig.2

E. Sensitivity characteristic curve

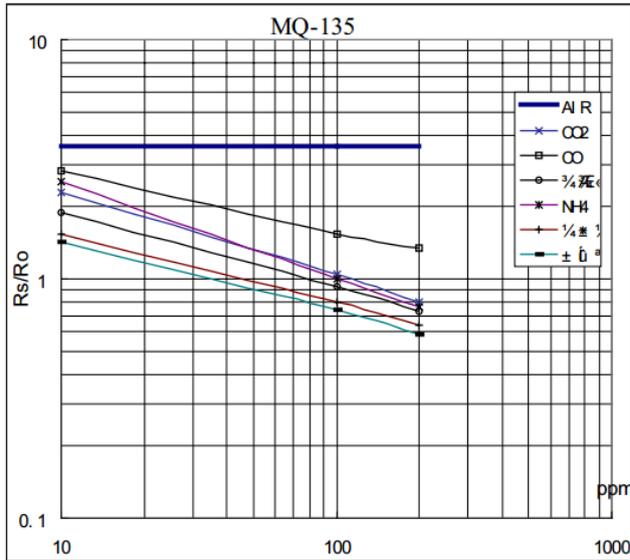


Figure II-2 sensitivity characteristics of the MQ-135 [14]

SENSITIVITY ADJUSTMENT

Resistance value of MQ-135 is different to various kinds and various concentration gases. So, When using this component, sensitivity adjustment is very necessary. We recommend that you calibrate the detector for 100ppm NH₃ or 50ppm Alcohol concentration in air and use a value of Load resistance that (RL) about 20 KΩ (10KΩ to 47 KΩ) [14]. When accurately measuring, the proper alarm point for the gas detector should be determined after considering the temperature and humidity influence.

Fig.3 shows the typical sensitivity characteristics of the MQ-135 for several gases.[22]

in their: Temp: 20 [14]

Humidity: 65% [14]

O₂ concentration 21%

RL=20kΩ [14]

Ro: sensor resistance at 100ppm of [14]

NH₃ in the clean air.

Rs:sensor resistance at various [14]

ANNEXE III. LISTE DES COMANDES AT

Tableau III-1 La liste des commandes AT [15]

Function	AT command	Response
Working	AT	OK
Restart	AT+RST	OK[System Ready, Vendor:www.ai-thinker.com]
Firmware version	AT+GMR	AT+GMR 0018000902 OK
List Access Points	AT+CWLAP	AT+CWLAP +CWLAP:(4,"RocheFortSurLac", -38,"70:62:b8:6f:6d:58",1) +CWLAP:(4,"LiliPad2.4",- 83,"f8:7b:8c:1e:7c:6d",1) OK
Join Access Point	AT+CWJAP? AT+CWJAP="ssid","Password"	
Quit Access Point	AT+CWQAP=? AT+CWQAP	Query OK
Get IP Address	AT+CIFSR	AT+CIFSR 192.168.0.105 OK

Set Parameters of Access Point	AT+CWSAP? AT+CWSAP=<ssid>,<pwd>,<chl>,<ecn> >	Query ssid, pwd chl = channel, ecn = encryption
WiFi Mode	AT+CWMODE? AT+CWMODE=1 AT+CWMODE=2 AT+CWMODE=3	Query STA AP BOTH
Set up TCP or UDP connection	AT+CIPSTART=? (CIPMUX=0) AT+CIPSTART = <type>,<addr>,<port> (CIPMUX=1) AT+CIPSTART= <id><type>,<addr>, <port>	Query id = 0-4, type = TCP/UDP, addr = IP address, port= port
TCP/UDP Connections	AT+ CIPMUX? AT+ CIPMUX=0 AT+ CIPMUX=1	Query Single Multiple
Check join devices' IP	AT+CWLIF	
TCP/IP Connection Status	AT+CIPSTATUS	AT+CIPSTATUS? no this fun
Send TCP/IP data	(CIPMUX=0) AT+CIPSEND=<length>; (CIPMUX=1) AT+CIPSEND= <id>,<length>	
Close TCP / UDP	AT+CIPCLOSE=<id> or AT+CIPCLOSE	

connection		
Set as server	AT+ CIPSERVER= <mode> [, <port>]	mode 0 to close server mode; mode 1 to open; port = port
Set the server timeout	AT+CIPSTO? AT+CIPSTO=<time>	Query <time>0~28800 in seconds
Baud Rate	AT+CIOBAUD? Supported: 9600, 19200, 38400, 74880, 115200, 230400, 460800, 921600	Query AT+CIOBAUD? +CIOBAUD:9600 OK
Check IP address	AT+CIFSR	AT+CIFSR 192.168.0.106 OK
Firmware Upgrade (from Cloud)	AT+CIUPDATE	1. +CIPUPDATE:1 found server 2. +CIPUPDATE:2 connect server 3. +CIPUPDATE:3 got edition 4. +CIPUPDATE:4 start update
Received data	+IPD	(CIPMUX=0): + IPD, <len>: (CIPMUX=1): + IPD, <id>, <len>: <data>
Watchdog Enable	AT+CSYSWDTENABLE	Watchdog, auto restart when program errors occur: enable

Watchdog Disable	AT+CSYSWDTDISABLE	Watchdog, auto restart when program errors occur: disable

ANNEXE IV. LES FONCTIONS IMPORTANTS DE L'ARDUINO

La fonction Retourne la résistance de capteur

```
float Mq135::getResistance() {  
  int val = analogRead(_pin);  
  return ((1023. / (float)val) * 5. - 1.) * RLOAD;  
}
```

La fonction qui récupère la valeur de gaz en ppm

```
float Mq135::getPPM(float parA, float rzero, float parB) {  
  return parA * pow((getResistance() / rzero), -parB);  
}
```

Fonction qui récupère La résistance R0

```
float Mq135::getRZeroS(float atm, float parA, float parB) {  
  return getResistance() * pow((atm / parA), (1. / parB));  
}
```

Envoi des données vers le serveur

```
void sendValue(String CO2, String CO, String NH4) {  
  
  String host = "pollution_detection-1334.appspot.com";  
  
  delay(100);  
  // on va se connecter à un réseau existant, donc on passe en mode station  
  softSerial.println("AT+CWMODE=1");  
  delay(1000);  
  
  // on se connecte au réseau  
  softSerial.println("AT+CWJAP=\"\" + SSID + "\",\"\" + PASS + "\"");
```

```

delay(1000);

// mode "connexions multiples"
softSerial.println("AT+CIPMUX=1");

delay(1000);
// on se connecte à notre serveur en TCP sur le port 80
softSerial.println("AT+CIPSTART=4,\"TCP\", \"" + host + "\",80");

delay(1000);

String request = "GET /hello?valeurCO2=" + CO2 + "&valeurCO=" + CO +
"&valeurNH4=" + NH4 + "HTTP/1.0\r\n";
request += "Host:" + host + "\r\n";

// on donne la taille de la requête qu'on va envoyer, en ajoutant 2 car
// println ajouter 2 caractères à la fin "\r" et "\n"

softSerial.println("AT+CIPSEND=4," + String(request.length() + 2));
delay(500);

// on envoie la requete
softSerial.println(request);
delay(1000);
// on ferme la connexion
softSerial.println("AT+CIPCLOSE=4");
}

```

ANNEXE V. QUELQUE FONCTION EXISTANT DANS L'APPLICATION WEB

Récupération des données à afficher dans la page d'accueil

```
public void recevoirAccueil( HttpServletRequest req){

    DatastoreService data =DatastoreServiceFactory.getDatastoreService();

    Query q = new
    Query("entiteCO2").setAncestor(KeyFactory.createKey("ancestre","ancestre")).addSort(
    "date",SortDirection.DESCENDING);

    List<Entity> resultsCO2 =data.prepare(q).asList(FetchOptions.Builder.withLimit(1));
    req.setAttribute("valeurCO2", resultsCO2);

    Query q1 = new
    Query("entiteCO").setAncestor(KeyFactory.createKey("ancestre","ancestre")).addSort("
    date",SortDirection.DESCENDING);

    List<Entity> resultsCO =data.prepare(q1).asList(FetchOptions.Builder.withLimit(1));
        req.setAttribute("valeurCO", resultsCO);

    Query q2 = new
    Query("entiteNH4").setAncestor(KeyFactory.createKey("ancestre","ancestre")).addSort(
    "date",SortDirection.DESCENDING);

    List<Entity> resultNH4 =data.prepare(q2).asList(FetchOptions.Builder.withLimit(1));
        req.setAttribute("valeurNH4", resultNH4);
}
```

Liste des notifications de CO

```
public void ListeNotifCO(HttpServletRequest req){

    DatastoreService data =DatastoreServiceFactory.getDatastoreService();

    Query q1 = new
    Query("entiteCO").setAncestor(KeyFactory.createKey("ancetre","ancetre")).setFilter(
    new Query.FilterPredicate("co", Query.FilterOperator.GREATER_THAN, 2.0));

    List<Entity> resultsCO =
    data.prepare(q1).asList(FetchOptions.Builder.withDefaults());

    req.setAttribute("valeurCO", resultsCO);

}
```

Liste des Notifications de CO2

```
public void ListeNotifCO2(HttpServletRequest req){

    DatastoreService data =DatastoreServiceFactory.getDatastoreService();

    Query q1 = new
    Query("entiteCO2").setAncestor(KeyFactory.createKey("ancetre","ancetre")).setFilter
    (new Query.FilterPredicate("co", Query.FilterOperator.GREATER_THAN, 390.0));

    List<Entity> resultsCO2 =
    data.prepare(q1).asList(FetchOptions.Builder.withDefaults());

    req.setAttribute("valeurCO2", resultsCO2);

}
```

Liste des notifications de NH3

```
public void ListeNotifNH4(HttpServletRequest req){

    DatastoreService data =DatastoreServiceFactory.getDatastoreService();

    Query q1 = new
    Query("entiteNH4").setAncestor(KeyFactory.createKey("ancestre","ancestre")).setFilter
(new Query.FilterPredicate("co", Query.FilterOperator.GREATER_THAN, 2.0));

    List<Entity> resultsNH4 =
    data.prepare(q1).asList(FetchOptions.Builder.withDefaults());

    req.setAttribute("valeurNH4", resultsNH4);

}
```

REFERENCES

- [1] Internet des objets, https://fr.wikipedia.org/wiki/Internet_des_objets, 10 mai 2016
- [2] Couches atmosphériques, http://www.meteo45.com/couches_atmospheriques.html, 10 mai 2016.
- [3] L'air, atmosphère et la pollution, Arnoldo Mondadori, 1975
- [4] Health impact, http://www.who.int/phe/health_topics/outdoorair/databases/health_impact/fr, 25 juin 2016.
- [5] E435, mesure des grandeurs et capteurs, M1, département électronique, 2013/2014
- [6] Ambatovy | Gestion de l'environnement physique, <http://www.ambatovy.com/docs/?lang=fr&p=9573>, 30 juin 2016
- [7] Les réseaux sans fil, http://www.icriq.com/fr/productique_tfp.html/asset_publisher/MeX1/Content/les-reseaux-sans-fil-fil/maximized, 4 juillet 2016
- [8] Les Réseau Satellites, http://www-igm.univ-mlv.fr/~dr/XPOSE2009/Les_Reseaux_Satellites/principes.html, 4 juillet 2016
- [9] Cloud computing, https://fr.wikipedia.org/wiki/Cloud_computing, 10 juillet 2016
- [10] Comparatif des principaux PaaS du marché; Cloud : comparatif de cinq services de plateforme (PaaS) incontournables – JDN, <http://www.journaldunet.com/solutions/cloud-computing/les-meilleures-plateformes-paas/tableaux-de-synthese.shtml#>, 10 juillet 2016
- [11] applications utiles aux agriculteurs, <http://www.frenchweb.fr/bon-app-5-applications-utiles-aux-agriculteurs/143304>, 15 Aout 2016
- [12] Une application web pour suivre la santé des patients atteints d'un cancer, <https://rcf.fr/actualite/une-application-web-pour-suivre-la-sante-des-patients-atteints-dun-cancer>, 15 Aout 2016
- [13] E559EA, SYSTEME AMBARQUE, M2, département ELECTRONIQUE, 2014/2015
- [14] TECHNICAL DATA MQ135 GAZ SENSOR
- [15] ESP 8266EX : A BEGINNER'S GUIDE, Espressif Systems, 17 septembre 2014
- [16] E556IA, JAVA, M2, département ELECTRONIQUE, 2014/2015
- [17] Mathieu Nebra ; Le datastore; Licence Creative Commons 6 2 ; 031 Aout 2016
- [18] NoSQL, <https://fr.wikipedia.org/wiki/NoSQL>, 20 Aout 2016
- [19] NoSQL Data Modeling Techniques, highlyscalable.wordpress.com/2012/03/01/nosql-data-modeling-techniques, 20 Aout 2016

[20] La dynamique des masses atmospherique, <http://eduscol.education.fr/applied/circula/theme/atmos22.htm>, 23 Septembre 2016

Auteur : RANDRIANASOLO Jerimanjaka Tsiorisoa
Titre : Système de detection de la pollution de l'air integrant le modèle Internet of Things
Nombre de page : 69
Nombre de figure : 31
Nombre de tableau : 11

RESUME:

Cet ouvrage présente la conception d'un système qui détecte la pollution de l'air en suivant le modèle IoT. On a créé un détecteur de gaz avec wifi et une application web. On a utilisé trois mq135, un arduino et le module Esp 01 pour créer le détecteur de gaz sans fil. L'application est développée avec le langage Java EE, html, css, JavaScript et stockée sur le Cloud. Pour utiliser le système il faut connecter le capteur à l'internet pour qu' il puisse envoyer des données vers la base en ligne. L'application web est l'interface entre l'utilisateur et le capteur. Le système a pour rôle d'informer les utilisateurs sur la qualité de l'air.

Mots clés : modèle IoT, detecteur de gaz, application web

Rapporteur: Mme RAMANANTSIHOARANA Harisoa Nathalie

Contact de l'auteur : 0340929649 - tsioriso@gmail.com