

Ville intelligente : Station qualité de l'air

Rapport de synthèse de PFE : explique brièvement le projet et son état d'avancement.

PFE 2018 – GROUPE 29
GROUPE ESEO ANGERS

TABLE DES MATIERES

Introduction :	3
Référence du document	3
Présentation du projet.....	3
Alimentation	5
Le régulateur de tension.....	5
Les batteries	6
Les panneaux solaires	6
Autres informations	6
Nœud Luminos	7
Le capteur UV	7
Le capteur de luminosité	8
Bilan de consommation	8
Nœud Thermos.....	9
Le capteur de température	9
Le capteur de pression.....	9
Bilan de consommation	9
Nœud Aeros	10
Bilan de consommation	10
Nœud Precipitos	11
Le gyromètre	11
L'anémomètre.....	11
Le pluviomètre	11
La centrale Nucleus	12
La communication entre NRF24.....	12
La communication MQTT	14
La base de données	15
La communication MQTT	15
L'insertion SQL	15
La récupération de la base de données	15
L'application Android	16
Fonctionnement de l'application mobile.....	16

TABLE DES ILLUSTRATIONS

Figure 1: Architecture matérielle.....	3
Figure 2: Empreinte NRF24LE1	4
Figure 3: Synoptique du bloc d'alimentation	5
Figure 4: Gestion de l'alimentation des capteurs de gaz	6
Figure 5: Synoptique du Luminos	7
Figure 6: Courbe de détermination de l'UV	7
Figure 7: Tableau de correspondance des indices UV	8
Figure 8: Bilan de consommation de Luminos	8
Figure 9: Synoptique de Thermos.....	9
Figure 10: Bilan de consommation de Thermos	9
Figure 11: Synoptique de Aeros.....	10
Figure 12: Bilan de consommation de Aeros	10
Figure 13: Synoptique de Precipitos.....	11
Figure 14: Synoptique de Nucleus	12
Figure 15: Types définis du protocole	12
Figure 16: Trames et échanges du protocole	13
Figure 17: Machine à état du protocole.....	13
Figure 18: Schéma de communication MQTT.....	14

INTRODUCTION :

REFERENCE DU DOCUMENT

Document	Référence
cahier_des_charges.pdf	REF 1
opt3001_datasheet.pdf	REF 2
rapport_pfe_2017.pdf	REF 3
installation_esp8266.txt	REF 4

PRESENTATION DU PROJET

Ce PFE avait pour but la réalisation d'une station de qualité de l'air. L'explication du projet, le cahier des charges précis, les contraintes et les attentes sont défini dans le document **REF 1**.

Pour ce projet, certains composants sont imposés : les NRF24 et l'ESP8266. En effet, ces composants ont la particularité d'avoir un très bon rapport qualité/prix et ils consomment très peu (la consommation énergétique étant une partie majeure de ce projet).

Dans ce document, il est donné un certain nombre d'information afin d'aider au mieux la poursuite de ce projet. Le code source du projet a aussi été commenté pour aider le développement futur.

Avant de débiter l'explication de chaque partie du projet, voici une architecture matérielle de notre système :

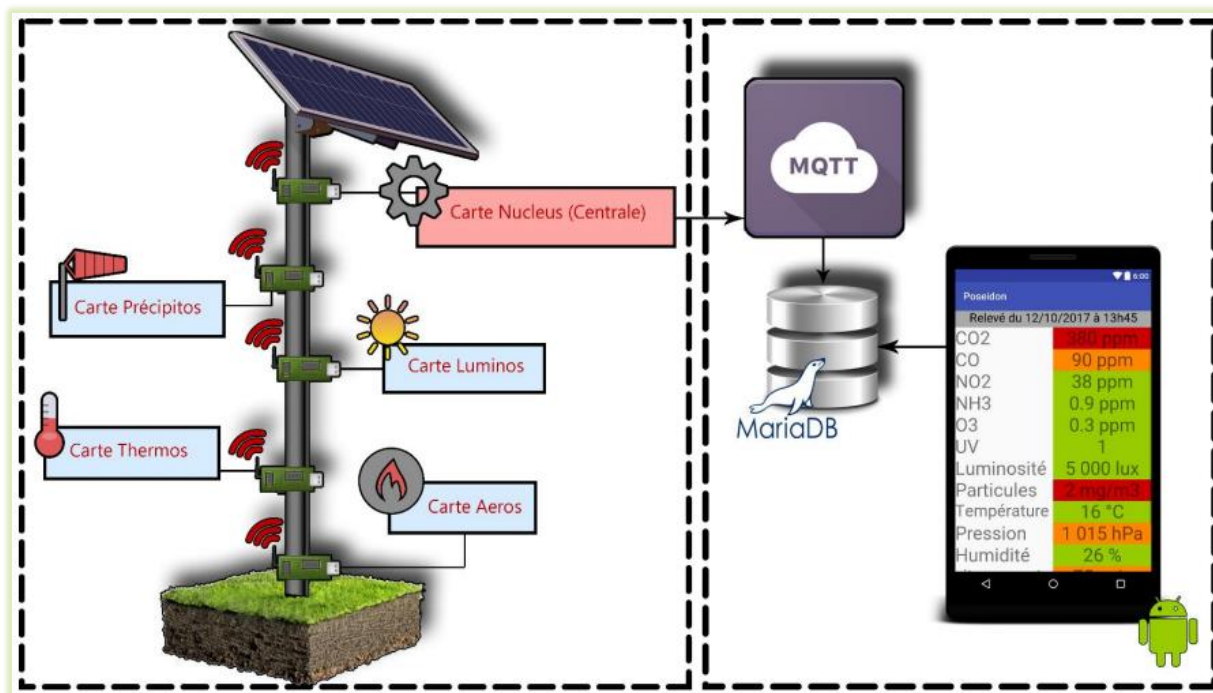


Figure 1: Architecture matérielle

Chaque carte contient un certain nombre de capteurs. Les capteurs utilisés peuvent mesurer un ou plusieurs paramètres. Par la suite, chaque carte va être détaillée, notamment, le nombre de capteurs utilisés, l'avancement de la carte et le fonctionnement des capteurs. Une particularité du microcontrôleur NRF24LE1 présent sur chaque carte est qu'il faut connecter le programmeur à l'envers pour le programmer, c'est-à-dire, ne pas suivre le détrompeur. Il faut que le détrompeur de la fiche femelle soit positionné vers la carte. Voir le schéma ci-dessous :

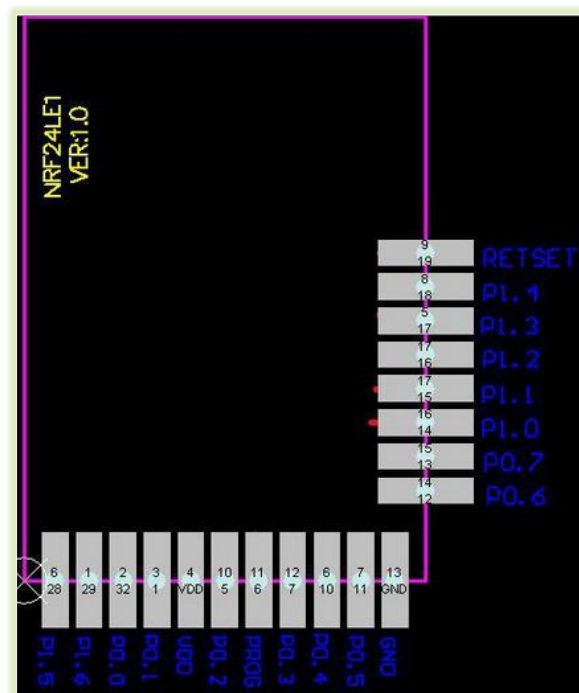


Figure 2: Empreinte NRF24LE1

ALIMENTATION

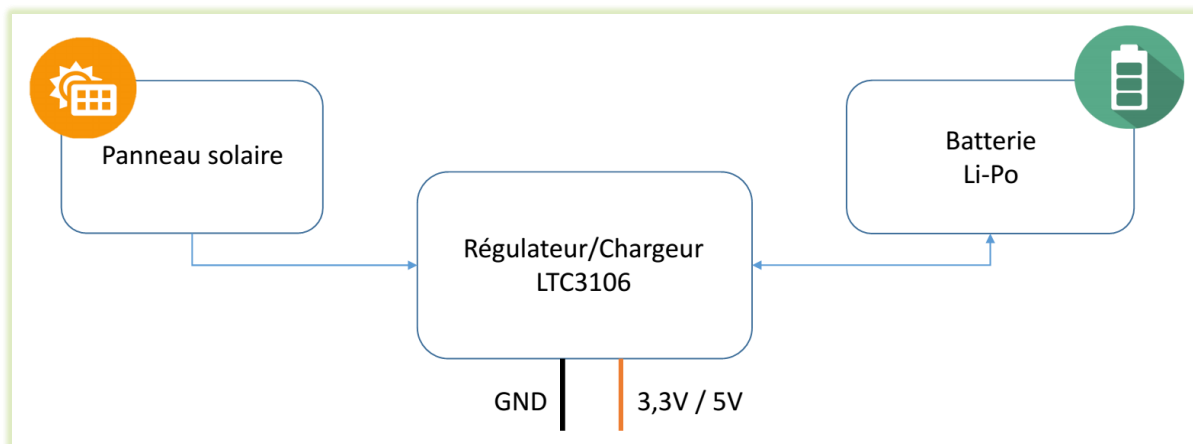


Figure 3: Synoptique du bloc d'alimentation

LE REGULATEUR DE TENSION

Chaque nœud est alimenté à l'aide de panneaux solaires et de batteries. Le tout est régulé à l'aide d'un circuit intégré de chez Linear, le LTC3106. Ce dernier est un régulateur buck/boost permettant de réguler la tension délivrée par le panneau solaire, d'alimenter le circuit et de recharger les batteries. Il offre aussi une protection contre les surcharges et les sur-décharges. Il est possible de choisir plusieurs tensions (1.8V, 2.2V, 3.3V et 5V). Cela est réalisable en câblant les deux broches "OS1" et "OS2". Ci-dessous, le tableau permettant de générer les tensions voulues :

OS1	OS2	OUTPUT VOLTAGE
0	0	1.8V
0	V _{CC}	2.2V
V _{CC}	0	3.3V
V _{CC}	V _{CC}	5V

Afin de fournir la tension appropriée à la batterie lors de l'opération de recharge, il suffit de connecter les broches "PRI", "SS1" et "SS2" à "VCC" et/ou à masse :

PRI	SS1	SS2	V _{STORE} / V _{CAP} OV	V _{STORE} / V _{CAP} UV	BATTERY TYPE
0	0	0	4V	2.78V	Li Carbon
0	0	V _{CC}	2.9V	1.9V	2x Rechargeable NiMH
0	V _{CC}	0	3V	2.15V	Rechargeable Li Coin Cell
0	V _{CC}	V _{CC}	4V	3V	Li Polymer/Graphite
V _{CC}	0	0	4.2V	2.1V	Primary, Non-Rechargeable

LES BATTERIES

Les batteries sélectionnées sont des batteries Lithium-polymère 3.7V/1200mAh et une batterie 3.7V/10Ah. La broche "PGOOD" permet de voir l'état de la batterie. Il est possible de la connecter à une broche du microcontrôleur. Dans le cas où, la broche "ENVSTR" est connectée à "V_{STORE}", la batterie est utilisée comme back-up. Par contre, si elle est connectée à la masse la batterie n'est pas utilisée comme back-up. Le circuit permet d'avoir une fonction MPPC (Max Power Point Controller). Cette fonctionnalité n'a pas été utilisée.

LES PANNEAUX SOLAIRES

Le montage réalisé permet d'obtenir une tension, mais elle n'est pas stable lors des chutes de tension aux bornes du panneau solaire. Afin d'améliorer le montage, il est possible de s'inspirer de ce lien :

<https://www.hackster.io/LuisLabMO/iot-energy-harvester-using-cayenne-86beb6>

Sur ce lien, "VCC" correspond à la tension de sortie et "VCCR" à la broche "VCC" du LTC3106.

Deux types de panneaux solaires ont été sélectionnés :

- Des panneaux 5V/1.5W pour la station centrale et les 3 nœuds (Themos, Precipitos et Luminos).
- Un autre panneau a été sélectionné pour le nœud Aeros, il est de 16cm*16cm et de 6V/4.5W.

AUTRES INFORMATIONS

Vu que les capteurs sur le nœud Aeros sont alimentés en 5V et que le microcontrôleur ne supporte pas une tension supérieure à 3.3V, le régulateur est configuré et va se charger de délivrer une tension de 5V et un LDO est ajouté pour obtenir 3.3V. Il faut savoir que les capteurs de gaz ne proposent pas de sleep mode. Pour pouvoir les contrôler un circuit à base de transistor PMOS et NMOS a été ajouté :

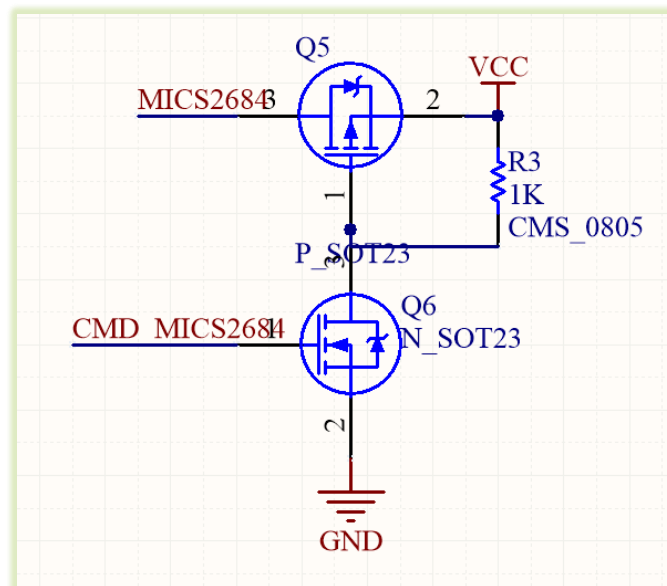


Figure 4: Gestion de l'alimentation des capteurs de gaz

NŒUD LUMINOS

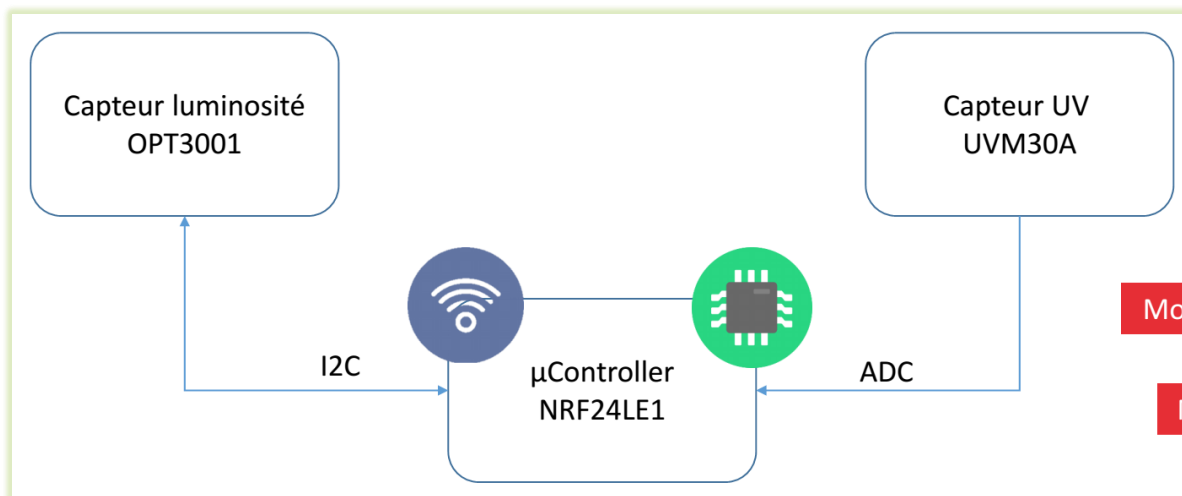


Figure 5: Synoptique du Luminos

Le nœud Luminos correspond au nœud n°1 sur Altium. Il y a deux capteurs sur ce nœud : un capteur de luminosité et un capteur UV.

NB : Avant d'utiliser la carte, il faut revérifier les soudures du composant OPT3001.

LE CAPTEUR UV

Pour trouver la valeur de l'UV, il faut se baser sur ce schéma constructeur :

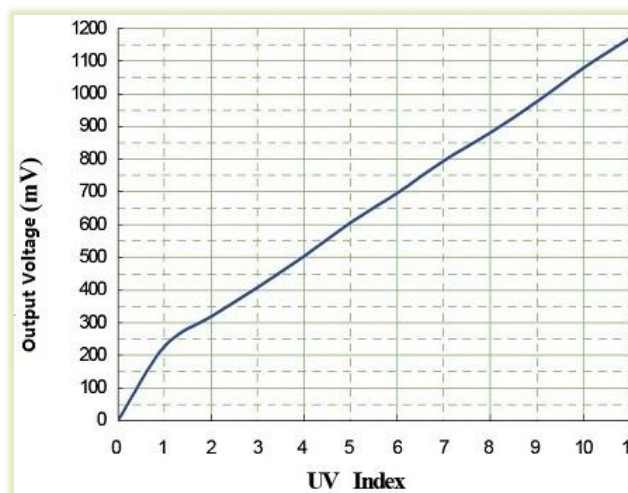


Figure 6: Courbe de détermination de l'UV

紫外线指数 UV Index	0	UV INDEX 1	UV INDEX 2	UV INDEX 3	UV INDEX 4	UV INDEX 5
Vout(mV)	<50	227	318	408	503	606
紫外线指数 UV Index	UV INDEX 6	UV INDEX 7	UV INDEX 8	UV INDEX 9	UV INDEX 10	UV INDEX 11 ⁺
Vout(mV)	696	795	881	976	1079	1170+

Figure 7: Tableau de correspondance des indices UV

LE CAPTEUR DE LUMINOSITE

Le capteur de luminosité (OPT3001) fonctionne en I2C. Avant de l'utiliser il faut le paramétrer en mettant les bits 10:9 du registre "01h" à "01" pour faire un single-shot. L'adresse I2C est déterminée par la broche "ADDR". Sur la carte réalisée, elle correspond à "1000101". Les résultats sont récupérés depuis le registre "00h". La donnée est sur 16 bit. Ensuite, il faut utiliser l'équation suivante et le tableau 9 du document **REF 2** :

$$lux = 0.01 \times (2^{E[3:0]}) \times R[11:0]$$

BILAN DE CONSOMMATION

Composants	Tension	En mesure	Sleep
NRF24LE1	1,9-3,6V	15ma	1,6µA
UV	0,3-5V	0,1mA	60µA
Luminosité	1,6-5,5V	3,5µA	1,8µA
régulateur		4µA	4µA
Total		20mA	70µA

Figure 8: Bilan de consommation de Luminos

NŒUD THERMOS

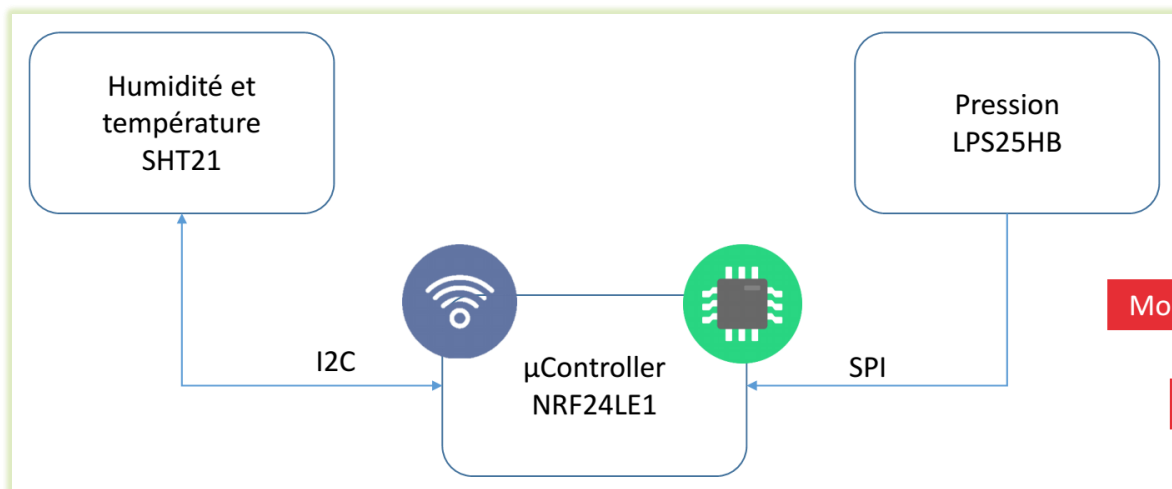


Figure 9: Synoptique de Thermos

Le nœud Thermos correspond au nœud n°2 sur Altium. Il y a deux capteurs sur ce nœud : un capteur de pression et un capteur permettant de relever la température et l'humidité.

NB : Aucun composant n'a été soudé sur cette carte, il y a une erreur sur l'empreinte du capteur LPS25HB.

LE CAPTEUR DE TEMPERATURE

Le capteur de température et d'humidité SHT21 fonctionne en I2C. Il n'y a pas de configuration spécifique pour ce capteur. La mesure est effectuée au moment de la lecture. L'adresse du dispositif est "1000000R/W". Afin de lire la température ou le taux d'humidité, différentes adresses sont utilisées

LE CAPTEUR DE PRESSION

Le second capteur c'est un capteur de pression LPS25HB. Il fonctionne en I2C ou en SPI. Le protocole de communication utilisé est le SPI. Afin d'utiliser le capteur, certains paramètres doivent être configurés. Le résultat de la mesure est récupéré depuis 3 registres : "REF_P_H", "REF_P_L" et "REF_P_XL". Le registre "REF_CONF" (10h) est utilisé pour configurer la résolution. Le single-shot est un mode par défaut, c'est-à-dire, le capteur fait une mesure, puis se met en sleep mode. Afin de lancer une mesure, il faut mettre le bit "PD" du registre "CTRL_REG1" à "1".

BILAN DE CONSOMMATION

NRF24LE1	1,9-3,6V	15ma	0,5µA
Pression	1,7-3,6V	0,025mA	0,5µA
température et humidité	2,1-3,6V	0,33mA	0,4µA
régulateur		4µA	4µA
Total		20mA	6µA

Figure 10: Bilan de consommation de Thermos

NŒUD AEROS

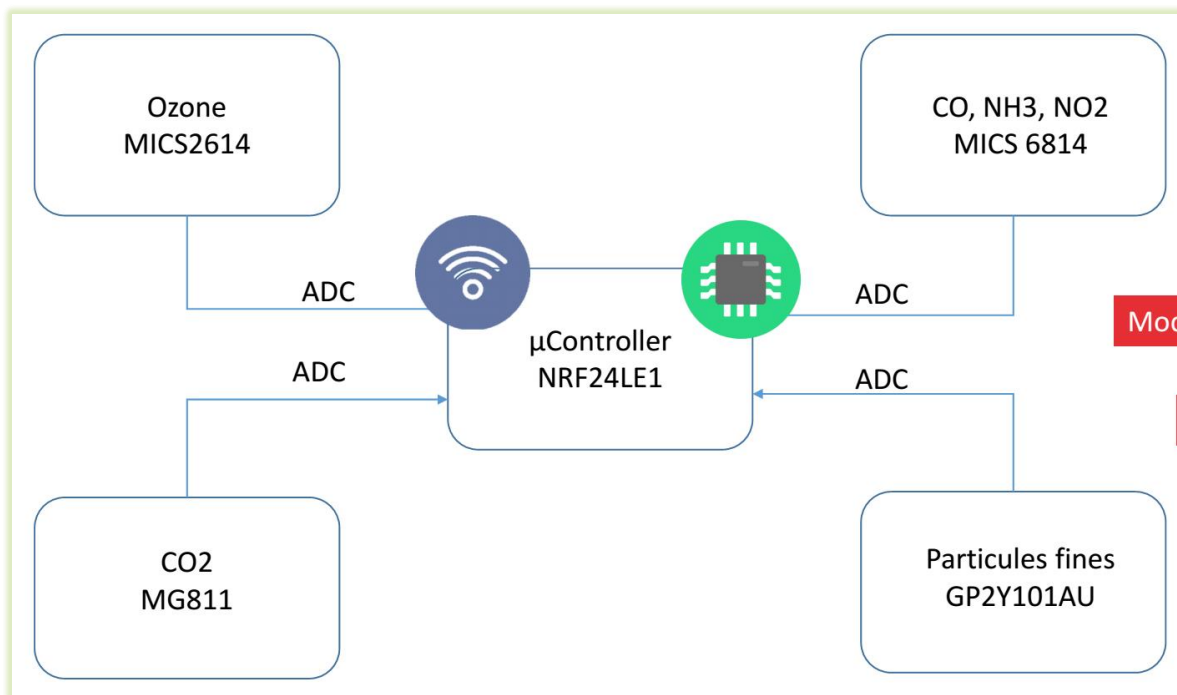


Figure 11: Synoptique de Aeros

Le nœud Aeros correspond au nœud n°3 sur Altium. Il y a plusieurs capteurs mesurant des données provenant de l'air ambiant (Gaz, particules,...). Toutes les valeurs renvoyées sont des tensions à convertir pour trouver la donnée utile. Il y a des capteurs de : CO2, MICS 6484, MICS2684 et de particules fines. Les informations d'utilisation de ces capteurs sont disponibles dans le document **REF 3**.

BILAN DE CONSOMMATION

Composants	Tension	En mesure	Sleep
NRF24LE1	1,9-3,6V	15ma	1,6 µA
Capteur Gaz MICS 2614	4,9-5,1	204mA	
Capteur co2 MG811	6V	200mA	
Capteur Gaz MICS 6814	5V	35mA	
Capteur de particule	0,3-7V	20mA	
régulateur		4µA	4µA
total		480mA	

Figure 12: Bilan de consommation de Aeros

NŒUD PRECIPITOS

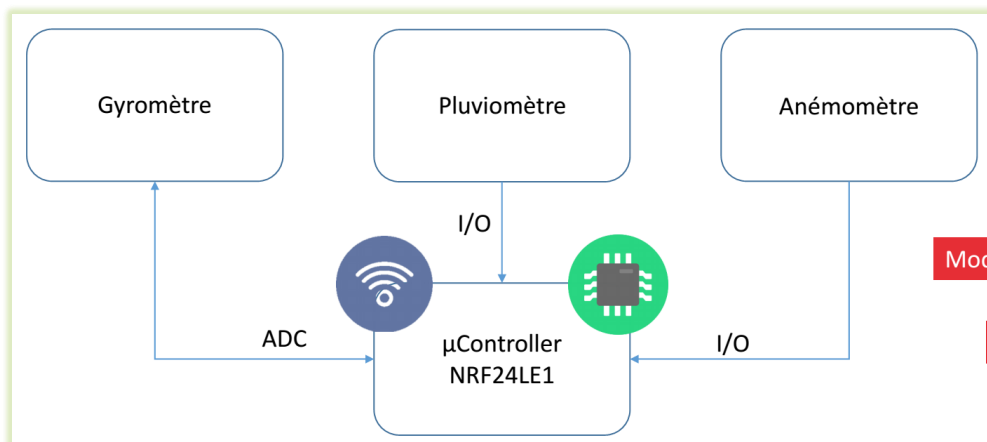


Figure 13: Synoptique de Precipitos

Le nœud Precipitos correspond au nœud n°4 sur Altium. Il y a trois capteurs sur ce nœud : un pluviomètre, un anémomètre et un gyromètre.

LE GYROMETRE

Il renvoie une tension, il fonctionne comme un potentiomètre.

L'ANEMOMETRE

La mesure de la vitesse du vent est réalisée à l'aide du capteur N82NF. Ce capteur est assez simple d'utilisation, en effet, un aimant est fixé à un point de la périphérie de la partie mobile de l'anémomètre. Cet aimant vient ouvrir et fermer un contacteur situé lui sur la partie fixe de l'anémomètre. Ce contacteur va donc se fermer à chaque fois que la girouette aura fait un tour sur elle-même. Pour récupérer la vitesse du vent, il suffit de déclencher un timer au moment où l'on détecte un premier passage, puis de l'arrêter lors du second passage.

Ainsi, avec la mesure du temps obtenu, nous pouvons en déduire la vitesse du vent. Pour cela, nous devons d'abord transformer ce temps en une fréquence, puis multiplier cette fréquence par 2,4 (en recherchant sur internet, il s'avère que si une fréquence de 1Hz est obtenue entre deux passages de l'aimant, cela correspond à une vitesse du vent d'environ 2,4km/h).

LE PLUVIOMETRE

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

Afin de récupérer les précipitations en mm, il suffit de compter le nombre de passages de l'aimant et de le multiplier par 0,2974 (correspond à la quantité d'eau qu'il a plu en millimètres). Afin de ne jamais rater de basculement du godet, il est judicieux de récupérer l'information avec une GPIOTE. Cela permet aussi de gagner du temps processeur.

LA CENTRALE NUCLEUS

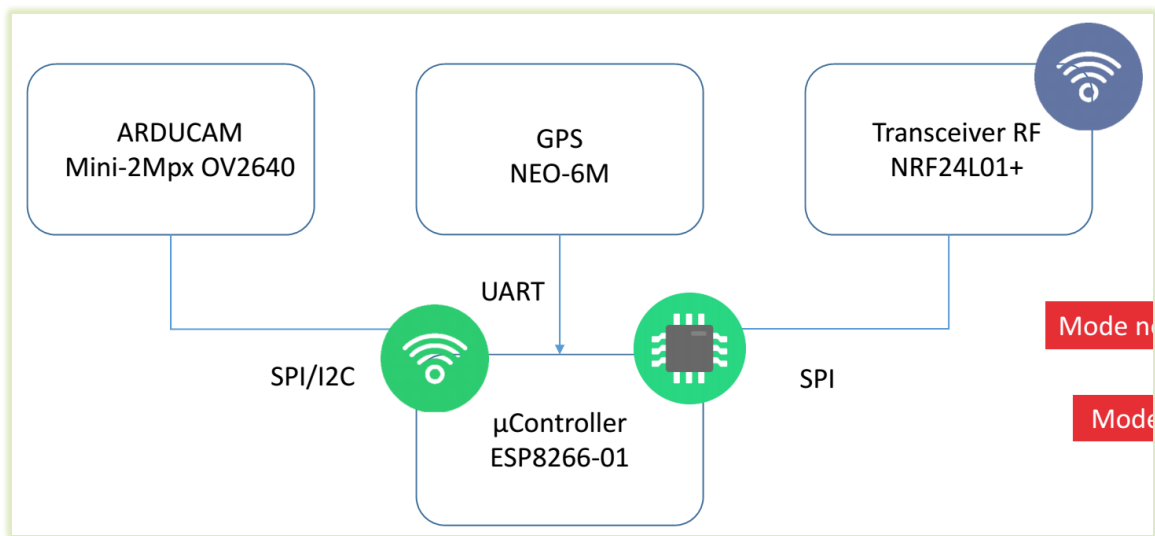


Figure 14: Synoptique de Nucleus

Sur la station centrale il y a l'ESP8226, le module NRF24L01+ et l'Arducam. Le module NRF24L01+ communique en SPI avec l'ESP8226. L'Arducam utilise l'I2C pour être configurée et le SPI pour prendre la photo et la récupérer.

Pour l'installation des logiciels de développement logiciel, se référer au document **REF 4**.

NB : La carte n'a pas été réalisée. L'Arducam et le GPS n'ont pas été développés pour la partie logicielle.

LA COMMUNICATION ENTRE NRF24

Afin de récupérer les informations provenant des nœuds de capteurs, le code s'inspire de la librairie Sming "MySensor". Cet exemple a servi essentiellement pour la configuration du NRF24, car sinon cet exemple implémente un réseau maillé, inutile pour notre projet en point à point. Un protocole "maison" a donc été utilisé :

MSG_TYPE	ID	CARD_TYPE	ID	SENSOR_TYPE	ID
SETUP	0x00	AEROS	0x00	CO	0x00
ASK_SEND_DATA	0x01	LUMINOS	0x01	CO2	0x01
SEND_DATA	0x02	PRECIPITOS	0x02	DIRECTION_VENT	0x02
		THERMOS	0x03	HUMIDITE	0x03
				LUMINOSITE	0x04
				NH3	0x05
				NO2	0x06
				O3	0x07
				PARTICULES	0x08
				PRESSION	0x09
				TEMPERATURE	0x10
				UV	0x11
				VITESSE_VENT	0x12

Figure 15: Types définis du protocole

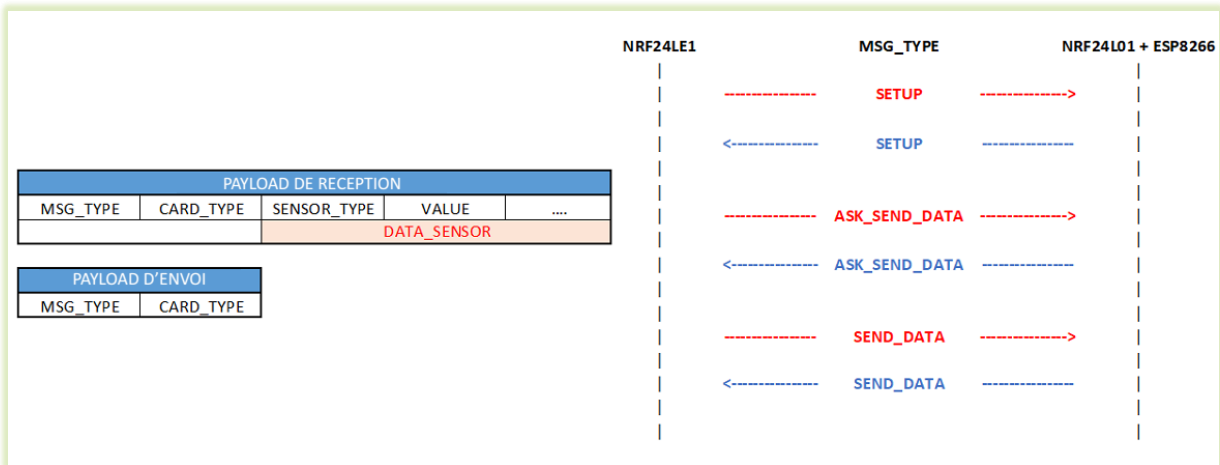


Figure 16: Trames et échanges du protocole

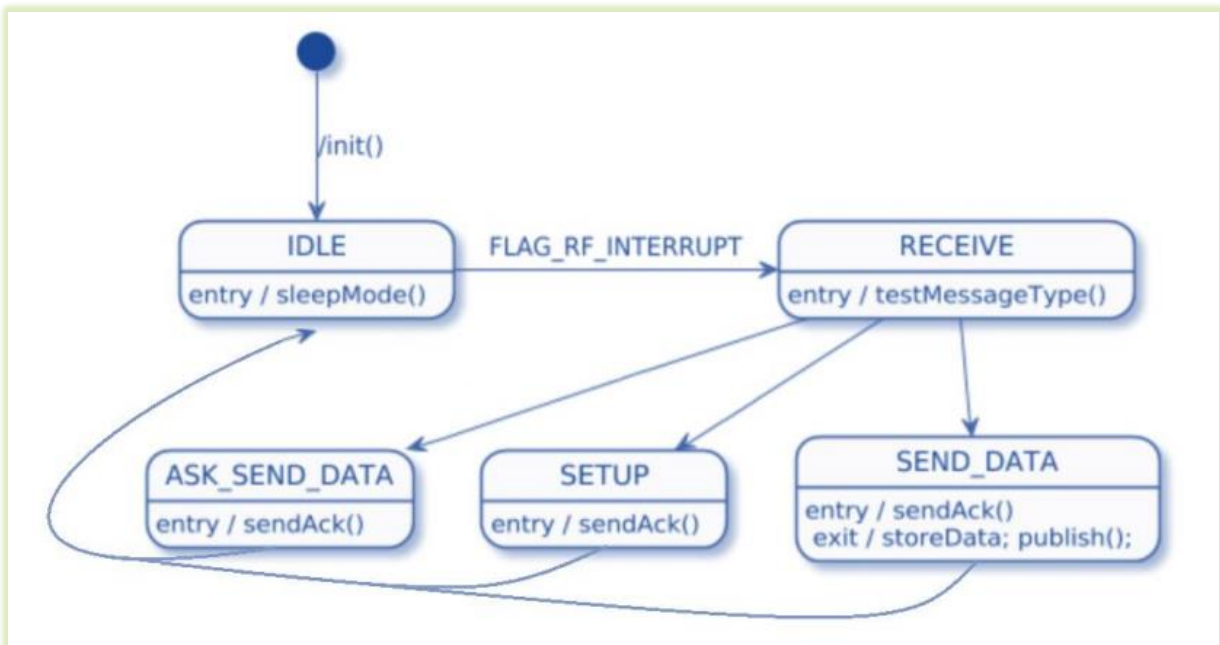


Figure 17: Machine à état du protocole

Le processus de réception est simple, le microcontrôleur est dans un état "IDLE" en "sleep mode". Dès, la réception d'une trame, un flag d'interruption est levé, le microcontrôleur se réveille et teste le message reçu. En fonction du message, le microcontrôleur peut être dans trois états différents et faire différentes opérations : configurer un nœud de capteur, autoriser la réception de données ou stocker les données reçues en mémoire.

Une particularité du code est que lorsqu'un nœud a demandé l'envoi de ses données au microcontrôleur, il est le seul à pouvoir communiquer avec le microcontrôleur jusqu'au transfert et stockage de celles-ci. Ainsi, quand le microcontrôleur arrive dans l'état ASK_SEND_DATA, il autorisera l'interruption RF seulement pour un nœud particulier et ce tant qu'il ne quittera pas l'état SEND_DATA.

NB : Pour tester la communication entre NRF, il est possible d'utiliser la raspberry comme passerelle afin de voir les messages envoyés. Ce lien détaille l'installation logicielle et matérielle :

<https://www.mysensors.org/build/raspberry>

LA COMMUNICATION MQTT

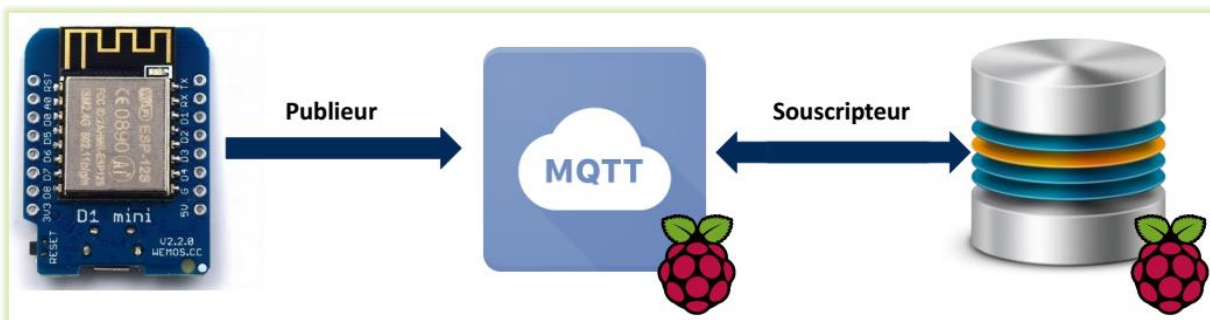


Figure 18: Schéma de communication MQTT

Le principe de MQTT peut se résumer simplement au principe de "boîte aux lettres" en langage C. Une définition à connaître est celle de broker MQTT. Le broker est un serveur (ici hébergé sur une Raspberry) qui reçoit les informations provenant d'un publieur et s'occupe de les renvoyer au(x) souscripteur(s). Pour fonctionner correctement, un publieur et un souscripteur doivent donc être connectés au broker et partager sur le même topic.

L'ESP étant le publieur, il faut donc penser à se connecter au même réseau que le broker et renseigner l'adresse IP du broker, puis choisir un nom de topic. Le code développé dans le projet s'inspire essentiellement du code exemple de Sming "MqttClient_Hello".

NB : Le souscripteur sera détaillé au chapitre suivant. Pour le broker, il faut installer sur une Raspberry un broker Mosquitto. Il est possible de définir le lancement de ce service au démarrage de l'OS pour une raison pratique. Voici un lien permettant d'installer un broker et des clients mqtt pour le test logiciel :

<https://projetsdiy.fr/mosquitto-broker-mqtt-raspberry-pi/>

LA BASE DE DONNEES

Afin de stocker les informations des capteurs, une Raspberry a été utilisé comme serveur de données. Pour installer les serveurs nécessaires au stockage et la récupération de données, il suffit d'installer Apache2, MySQL et PHPMyAdmin. Le lien suivant détaille les installations :

<https://doc.ubuntu-fr.org/lamp>

LA COMMUNICATION MQTT

Comme vu au chapitre précédant, la Raspberry est utilisé en souscripteur. Le programme développé tourne en boucle, en écoutant le port MQTT (:1883) afin de récupérer les informations en provenance du broker. Le programme est développé en langage C utilisant la librairie Paho. Voici le lien d'installation de la librairie et l'exemple dont s'inspire largement le programme :

<https://www.eclipse.org/paho/clients/c/>

<https://www.eclipse.org/paho/files/mqttdoc/MQTTClient/html/subasync.html>

L'INSERTION SQL

Pour faire des insertions SQL dans la base de données, le code s'inspire de cet exemple :

<http://zetcode.com/db/mysqlc/>

LA RECUPERATION DE LA BASE DE DONNEES

Pour extraire les informations de la base de données, un script PHP a été développé.

L'APPLICATION ANDROID

FONCTIONNEMENT DE L'APPLICATION MOBILE

Pour faire fonctionner l'application mobile, il faut qu'un terminal Android soit connecté sur le même réseau que le serveur de données. Après connexion, il faut lancer l'application et aller sur l'écran des relevés. Si tout se passe bien un simple rafraichissement de la page (scroll du doigt vers le bas de l'écran) permet de récupérer les données sur la base de données. Il y a parfois quelques latences et il ne faut pas hésiter à retenter plusieurs fois. Un second écran est propre à la donnée sélectionnée et comprend un graphe. Sur ce graphe, nous avons la représentation de toutes les données présente sur la base de données par rapport au temps.

Il y a des fonctionnalités qui n'ont pas été encore développées :

- L'ajout de filtre sur les graphes (filtres temporels ou limitatifs).
- L'ajout des unités sur les axes des graphes.
- La limitation du nombre de points sur un graphe.

De même, il y a certaines améliorations à apporter :

- L'implémentation d'une donnée avec des chaînes de caractère (pour la direction du vent, NORD, EST, OUEST, SUD). Car pour le moment les constructeurs ne permettent d'afficher que les entiers signés.
- La gestion de la déconnection (intempestive ou non) de la communication est à tester.