

**Rapport BNJmétéo**

Le port / Mosbah / Jalal

Introduction

Une image contenant bâtiment, table, assis, vert

Description générée automatiquementCette station météorologique doit permettre aux utilisateurs d’accéder à ses données météorologiques, via <http://77.131.214.150/>. La station sera équipée de capteurs permettant de réaliser des relevés de données météorologiques (hygrométrie, pluviométrie, luminosité, température, vitesse et direction du vent, pression atmosphérique. La station météorologique devra également pouvoir être installée facilement par n’importe quel utilisateur et à n’importe quel endroit. De plus, la station devra fonctionner en toute autonomie : aucune intervention extérieure ne devra être effectuée.

Notre projet consistera à mettre en place une station météorologique qui collecte des données de capteurs, les stockent sur un serveur dans un BD et les exploitent pour faire des analyses et les affiche. Cet affichage peut se faire à distance via une appli mobile et/ou web

Schéma d’architecture

Notre projet contient les trois modules que nous avons travaillé au cours du semestre.

Avec la présence de l’application web qui marche sous PHP qui est la partie de Jean Michel Bruneau.

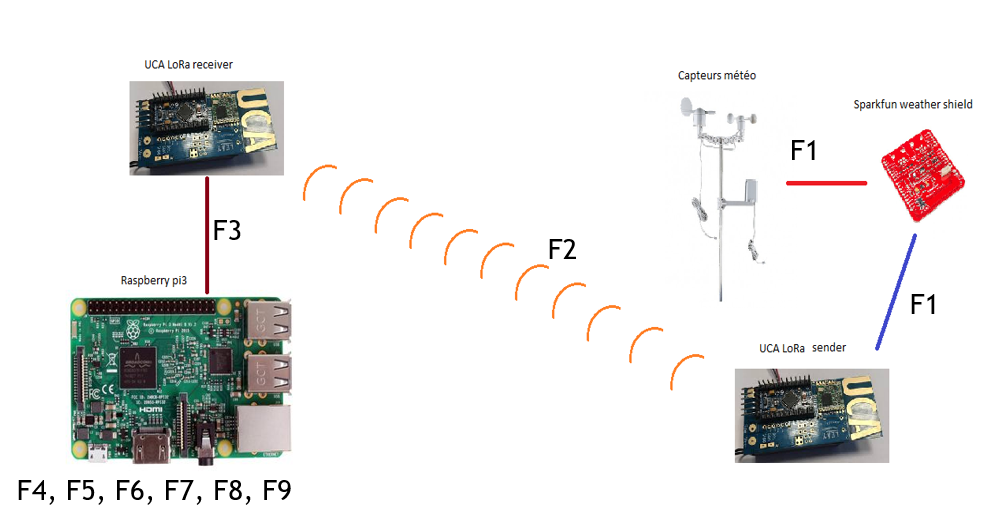
Grâce au module de Madame Belleudy, nous allons pouvoir utiliser les logiciels des cours pour faire communiquer deux cartes Arduino à distance avec le dispositif LoRa.

Pour la partie de Madame Peraldi, nous avons choisi d’envoyer une notification via mail lorsque le vent est trop fort (>20km/h)

Une image contenant capture d’écran

Description générée automatiquement

Architecture matérielle



Fonctionnalités attendues

* F1: Collecte de données par le mât
* F2: Transmission des données en LoRa
* F2’: Réception des données en LoRa et envoi des données au Raspberry® pi3®
* F3: Collecte des données par le Raspberry® pi3® à fréquence variable
* F4: Notification en temps réel lorsque le vent est trop fort (>20km/h)
* F5: Stockage des données dans la base de données
* F6: Application web pour l’affichage des données
* F7: Accès distant au serveur web avec accès sécurisé et certificat
* F8: Application mobile qui va pointer vers le serveur web
* F9: Accès à la webcam en direct

Les capteurs positionnés sur le mât ainsi que sur la carte *sparkfun weather shield* vont relever des données et les transmettre à la carte *UCA LoRa sender.* Les données seront transmises à l’autre carte *LoRa receiver* par le protocole *LoRa.* Ces données sont ensuite transmises au raspberry pi3 qui contient un serveur *apache2* hébergeant l’application et la base de données. Toutes les 30 minutes la raspberry lance une tâche CRON qui va insérer les données dans la base.

Remarque : Les cartes *sparkfun weather shield, UCA LoRa sender* et *UCA LoRa receiver* sont toutes programmable à l’aide de l’IDE Arduino.

Les différentes fonctionnalités et présentation des acteurs

****

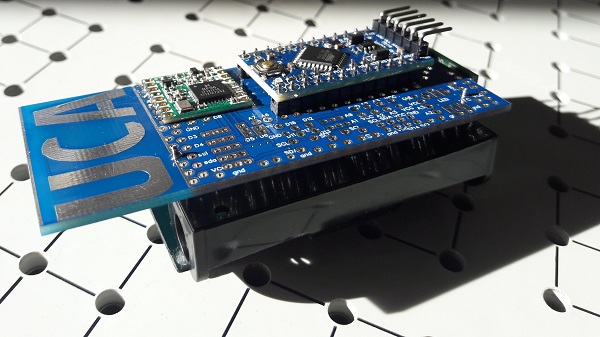
Capteurs météo: Il s’agit d’un mât qui est constitué d’un capteur de vitesse de vent, un capteur d’orientation du vent et d’un capteur de pluviométrie.

Communication RJ45



Sparkfun weather shield: carte electronique programmable avec deux ports rj45, un capteur d’humidite et temperature (ref: si7021), un capteur de pression, (ref: mpl3115a2) et d’un capteur de luminosite (ref: als-pt19).

https://lh5.googleusercontent.com/afN1EhbKD_vbKKFY-DAbWJuO9rX6TBaj3EBMlHc3caATMHpa8Z0QHkuL-RawKqGTPaeQf9dycYruEFbG-g0llR_qX6Zllvp7o26Oc72i3fiqHSkGEXd3ggumgxZ6wJVbfUsO1ijqcommunication filaire



UCA LoRa Sender: Carte contenant un module LoRa et permettant d’envoyer des données.

https://lh4.googleusercontent.com/zo1SSMT9TtucFMmzqauaBlrCZAzfbMcw3UaLrZO2U0UyxuKDwxyr1zCcY89biCmM5tpZ7YiQtTttQTg70dL7h9rhJWZnaYaEeQV07oa3uUqIniVuPnMwl_Z1nNbpZNUz18tS5I1Mhttps://lh4.googleusercontent.com/5KsJLOn902qwPuYw9_qgG45dbY58WbDfBJN-WK2vlgEIVxjFcygHjGChjkBqAO6JM856PokjFVX2HXF_oG2FHYNZ9yw45IYCGOeFhBHmYhKtvTodqhJxAVNpzWd9_ntljRyiWLabCommunication LoRa

UCA LoRa receiver: Carte contenant un module LoRa et permettant de recevoir des données.



Raspberry pi3: Le Raspberry Pi est un [nano-ordinateur](https://fr.wikipedia.org/wiki/Nano-ordinateur) [monocarte](https://fr.wikipedia.org/wiki/Ordinateur_%C3%A0_carte_unique) à [processeur ARM](https://fr.wikipedia.org/wiki/Architecture_ARM) conçu par des professeurs du département informatique de l'[université de Cambridge](https://fr.wikipedia.org/wiki/Universit%C3%A9_de_Cambridge) dans le cadre de la [fondation Raspberry Pi](https://fr.wikipedia.org/wiki/Fondation_Raspberry_Pi) (source: wikipedia).

Communication filaire via un port série

Répartition du travail

Bihane : développement de l'application web et installation de l'environnement de la Raspberry

F1 collecte de données par le mât

[*https://github.com/mapfra/LPIOTIA\_2019\_BNJMeteo/blob/master/meteo/projet%20meteo/code%20arduino.txt*](https://github.com/mapfra/LPIOTIA_2019_BNJMeteo/blob/master/meteo/projet%20meteo/code%20arduino.txt)

F3 collecte des données par le Raspberry à fréquence variable

[*https://github.com/mapfra/LPIOTIA\_2019\_BNJMeteo/blob/master/meteo/projet%20meteo/sparkfun/sparkunIno*](https://github.com/mapfra/LPIOTIA_2019_BNJMeteo/blob/master/meteo/projet%20meteo/sparkfun/sparkunIno)

F4 Notification temps réel de certaines données critiques

[*https://github.com/mapfra/LPIOTIA\_2019\_BNJMeteo/blob/master/meteo/projet%20meteo/code%20python.txt*](https://github.com/mapfra/LPIOTIA_2019_BNJMeteo/blob/master/meteo/projet%20meteo/code%20python.txt)

F5 stockage des données dans la BD

[*https://github.com/mapfra/LPIOTIA\_2019\_BNJMeteo/blob/master/meteo/meteoInsertValues.php*](https://github.com/mapfra/LPIOTIA_2019_BNJMeteo/blob/master/meteo/meteoInsertValues.php)

F6 application web d’affichage des données

[*https://github.com/mapfra/LPIOTIA\_2019\_BNJMeteo/blob/master/meteo/index.php*](https://github.com/mapfra/LPIOTIA_2019_BNJMeteo/blob/master/meteo/index.php)

F7 Accès distant au serveur web avec accès sécurisé et certificat.

[*https://77.131.214.150/*](http://77.131.214.150/)

Jimmy : Programmation de la carte sparkfun weather shield et développement de l'application mobile

F2 Développement du code pour la transmission LoRa

[*https://github.com/mapfra/LPIOTIA\_2019\_BNJMeteo/blob/master/meteo/projet%20meteo/code%20LoRa/LoRaSender.ino*](https://github.com/mapfra/LPIOTIA_2019_BNJMeteo/blob/master/meteo/projet%20meteo/code%20LoRa/LoRaSender.ino)

F2’ Réception des données en Lora et envoi de ces données au RPU en BT

<https://github.com/mapfra/LPIOTIA_2019_BNJMeteo/blob/master/meteo/projet%20meteo/code%20LoRa/LoRaReceiver.ino>

F8 Développement d’une appli mobile et de son infrastructure pointant vers le serveur web

[*https://github.com/mapfra/LPIOTIA\_2019\_BNJMeteo/tree/master/Projetmeteo\_mobile*](https://github.com/mapfra/LPIOTIA_2019_BNJMeteo/tree/master/Projetmeteo_mobile)

F9 Accès à la webcam.

<https://github.com/mapfra/LPIOTIA_2019_BNJMeteo/blob/master/meteo/index.php>

Nassim : Programmation des cartes UCA board

F2 Transmission des données en LoRa grâce aux codes obtenues avec l’aide de

<https://github.com/mapfra/LPIOTIA_2019_BNJMeteo/blob/master/meteo/projet%20meteo/code%20LoRa/LoRaSender.ino>

F2’ Réception des données en Lora et envoi de ces données au RPU en BT

<https://github.com/mapfra/LPIOTIA_2019_BNJMeteo/blob/master/meteo/projet%20meteo/code%20LoRa/LoRaReceiver.ino>

F3 Ecriture de données dans un fichier CSV (code python)

<https://github.com/mapfra/LPIOTIA_2019_BNJMeteo/blob/master/meteo/projet%20meteo/sparkfun/sparkunIno>

F8 Développement d’une architecture mobile pour notre projet, application

<https://github.com/mapfra/LPIOTIA_2019_BNJMeteo/tree/master/Projetmeteo_mobile>

Bilan

* F1: Collecte de données par le mât

Fait

* F2: Transmission des données en LoRa

90% Code Fini, Communication Lora établi mais la transmission des données entre la carte sparkfun et la carte UCA lora impossible (vu avec madame belleudy)

* F2’: Réception des données en LoRa et envoi des données au Raspberry® pi3®

90% Code Fini, Communication Lora établi mais la transmission des données entre la carte sparkfun et la carte UCA lora impossible (vu avec madame belleudy)

* F3: Collecte des données par le Raspberry® pi3® à fréquence variable

Fait

* F4: Notification en temps réel lorsque le vent est trop fort (>20km/h)

Fait

* F5: Stockage des données dans la base de données

Fait

* F6: Application web pour l’affichage des données

Fait

* F7: Accès distant au serveur web avec accès sécurisé et certificat

Fait

* F8: Application mobile qui va pointer vers le serveur web

90% Erreur sur l’application mobile (problème de connectivité entre serveur web et l’application mobile)

* F9: Accès à la webcam

Fait

Erreur Application mobile

Menu Principal de l’application mobile



Erreur de connexion application Mobile

