
Serres connectées

Projet RICM 4 - Polytech Grenoble



Antoine Boisadam - Oriane Dalle

2016 - 2017

Sommaire

Présentation du projet	3
Outils et matériels utilisés	3
LoRa	3
Node-RED	4
MQTT	4
InfluxDB	4
Grafana	5
Le matériel	5
Fonctionnement	6
Général	6
Envoie des données	7
Traitement des données	8
Evolutions futures	8
Conclusion	9
Annexe : Événements	10
IOT Eclipse Day Grenoble 2017	10
Maker Faire Grenoble 2017	10

Présentation du projet

Pour rentabiliser au mieux une exploitation agricole il est important d'optimiser l'arrosage de ses plantes ainsi que la température ambiante au sein d'une serre. Une serre peut être grande et une exploitation peut être composée de beaucoup de serres qui n'auront pas forcément la même orientation, et donc pas nécessairement les mêmes conditions intérieures. Pour une rentabilité maximale il faudrait donc vérifier plusieurs fois par jour que tout vas bien dans toutes les serres... De plus des événements extérieurs (atmosphériques par exemple) peuvent bloquer l'accès aux serres et donc empêcher de vérifier manuellement les conditions intérieures.

Il existe donc un besoin qui peut être automatisé par des capteurs. Pour y répondre nous avons 2 contraintes :

- Pas de connexion internet
- Pas d'accès à une source électrique dans les serres

Partant de ces observations notre projet a pour but d'installer des capteurs d'humidité et de température du sol et de l'air.

Les mesures seront récupérées par des cartes qui communiquent sans-fil, en LoRa, avec une autre carte "maîtresse" branchée à un ordinateur. Les données sont ensuite formatée et stockée puis finalement accessible pour l'agriculteur via une interface graphique. Les cartes que nous possédons, équipée d'un module **LoRa**, permettent de communiquer avec une distance supérieur à 30 km. Ces cartes et capteurs dans la serres seront alimentés avec un panneau solaire ou une batterie.

Outils et matériels utilisés

1. LoRa

Afin de mettre en place une solution adaptée aux différentes contraintes nous avons utilisé la technologie du LoRa. Le LoRa est une technologie basée sur une communication radio entre des balises (868 MHz en Europe). Celle-ci est de plus en plus utilisé afin de créer des objets/bâtiments connectés. Notamment afin de contrôler la consommation d'eau d'une maison ou l'éclairage d'un bâtiment.



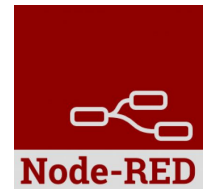
Pour utiliser cette technologie, nous nous sommes servi de deux Nucleo L073RZ (*ST Microelectronics*) muni d'un module LoRa (*Semtech*). La première d'entre elles sert à récupérer les données des différents capteurs et sera placée dans la serre alors que l'autre sera branchée au poste de contrôle et réceptionnera les données envoyés par la première.

Pour programmer avec ces cartes on utilise le compilateur en ligne mbed. Ce dernier nous permet d'éditer du code en C et de le compiler pour notre type de carte. Une fois compilé il ne reste plus qu'à charger le programme sur la carte.

Pour assurer une alimentation de la carte qui restera dans la serre nous équiperons d'un panneau solaire et/ou d'une batterie.

2. Node-RED

Une fois les données envoyées grâce aux cartes, on les récupère et traite grâce à Node-RED. Node-RED est un serveur web qui permet de créer des scénarios de gestion de flux de données d'objet connecté. On peut, de manière claire, décider où doit aller chaque flux de données.



```
03/03/2017 à 12:22:07 node: ca56794e.7aa768
msg.payload: string[11]
▶ "23.899618" ↵

03/03/2017 à 12:22:08 node: ca56794e.7aa768
msg.payload: string[11]
▶ "23.877928" ↵

03/03/2017 à 12:22:09 node: ca56794e.7aa768
msg.payload: string[11]
▶ "24.073235" ↵

03/03/2017 à 12:22:10 node: ca56794e.7aa768
msg.payload: string[11]
▶ "23.877928" ↵
```

*Exemple de l'interface de Node-RED
Ici, on récupère les données en provenance du port serial "/dev/ttyACM0",
elles passent dans la fonction "Extract Data", puis on les affiche.*

3. MQTT

Le protocole MQTT est un protocole de messagerie *publish/subscribe* c'est à dire que des récepteurs peuvent s'inscrire à un certain sujet (topic) et seront averti à chaque message publié sur celui-ci. On utilise ce protocole afin de créer des topics sur l'arrivée des informations concernant les serres, les destinataires peuvent donc récupérer les informations dès que celles-ci arrivent.

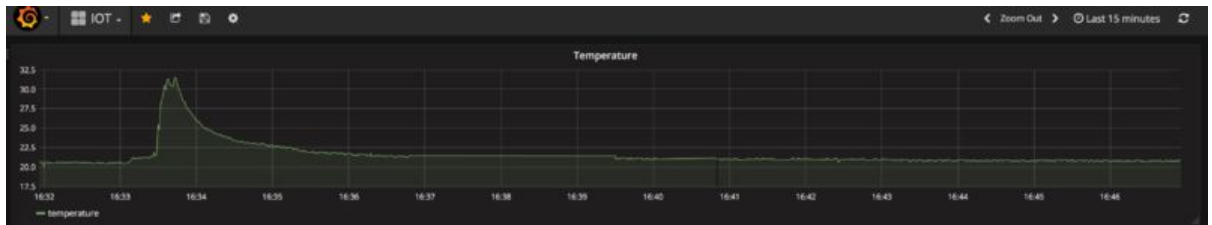
4. InfluxDB

Nous stockons nos informations dans une base de données de type InfluxDB. C'est un type de base de donnée qui enregistre les données en fonction de l'heure où elles arrivent. Elle est conçue et optimisée pour des couples "métrique/date".



5. Grafana

L'affichage des données récupérées est effectué via Grafana. Grafana est une solution libre permettant de réaliser des tableaux de bord depuis des métriques InfluxDB. Grâce à une simple requête, le logiciel va nous créer un graphique qui s'actualisera en temps réel.



Capture d'écran de l'affichage des résultats

6. Le matériel

- 2 cartes Nucleo L073RZ équipées de modules LoRa
- 1 capteur SHT10 : Humidité et température du sol
- 1 capteur DHT11 : Humidité et température de l'air
- 1 panneau solaire pour alimenter la carte LoRa qui sera dans la serre



Carte nucleo L073RZ et son module LoRa + Capteurs + Panneau solaire

Fonctionnement

1. Général

Notre dispositif fonctionne de la manière suivante :

1. La première carte, présente dans la serre, est alimentée par un panneau solaire et/ou une batterie. Elle récupère toutes les informations des capteurs qui lui sont directement reliés.
2. Cette même carte se charge ensuite de compresser (cf. partie [Envoie des données](#)) et d'envoyer les données via le mode de transmission LoRa.
3. Une autre carte, positionnée au centre de contrôle et reliée à un ordinateur, écoute en permanence sur la fréquence 868 MHz. Elle reçoit à un moment les données de la première carte, elle se charge de les décompresser et les affiche.
4. Un serveur Node-RED, lancé sur l'ordinateur, récupère les données que la seconde carte transmet via le port serial (en les affichant) et publie ces données sur un topic MQTT.
5. Un autre scénario Node-RED est notifié lors d'un nouveau message sur MQTT et traite (cf. partie [Traitement des données](#)) puis stocke les informations dans une base de donnée (InfluxDB).
6. Pour finir, cette base de donnée est utilisée pour afficher l'évolution de la température et l'humidité dans Grafana (installé sur l'ordinateur).

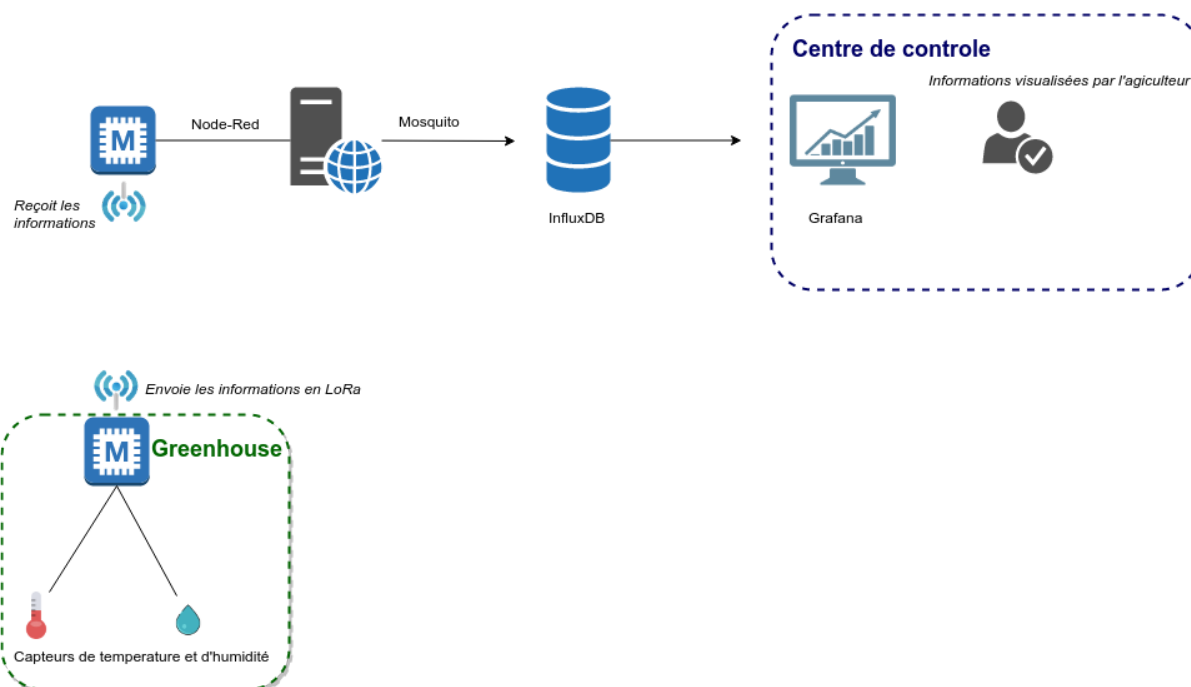


Schéma récapitulatif du dispositif

2. Envoie des données

Une des contraintes du LoRa est qu'il est très coûteux en temps de transmission. De plus, il y a des réglementations nous obligeant à utiliser la bande radio seulement 1% du temps (c'est à dire 36 secondes de transmission/heure, soit en moyenne 1800octets/heure).

Afin de pouvoir récupérer les données des capteurs deux fois par heure, nous avons dû optimiser nos informations et les compresser.

Chacun de nos messages sont capables de transmettre 4 mesures (nombres à virgule), des informations pour l'identification de ces mesures (Température ou humidité, dans l'air ou le sol) ainsi que des informations sur la serre. Ils font 6 octets et sont définis comme ci-dessous :

Mesure 1*	Mesure 2*	Mesure 3*	Mesure 4*	Types des mesures **	Info sur la serre ***	-
8 bits	8 bits	8 bits	8 bits	4 * 2 bits	4 et 2 bits	2 bits

* : Chacune des mesures est encodée sur un octet. Nous avons défini une règle de conversion des *floats* (32 bits) vers ce type de 8 bits :

$$\frac{x + t}{100} \times 255$$

On a x notre mesure, un flottant défini sur:

- [0; 100] dans le cas d'une humidité.
- [-30; 70] dans le cas d'une température.

Et $t = 30$ si et seulement si on converti une température, 0 sinon. Ce t permet de s'assurer d'un résultat de la division compris entre 0 et 1.

On multiplie ce résultat par 255 (Nombre maximum codable sur 8 bits) et on garde la partie entière.

Ce calcul nous garanti une précision de +- 0,2 dans le pire des cas.

** : Pour chaque mesure nous avons 2 bits dédiés au type de la mesure et à l'environnement de celle-ci, voici la table des correspondances :

Bit	Type	Environnement
0	Temperature	Air
1	Humidité	Sol

*** : Pour les informations sur la serre nous avons 2 bits qui nous informe sur la position du capteur (extrémité 1, extrémité 2 ou milieu de la serre) et 4 bits identifiant le numéro de la serre (de 0 à 15). On complète l'octet avec 2 zéros.

3. Traitement des données

Les données “brutes” reçues sont décompressées et traitées, nous identifions alors quels sont les types de chaque mesure et on les stockent dans une base de données en fonction de ces derniers. Notre mode d'identification dans node-RED est :

Le message lu sur le port serial est, par exemple, le suivant :

```
=====  
New Packet  
=====  
Greenhouse number: 1  
Sensors position: 1  
HumiA=37.20  
TempA=21.90  
HumiS=87.00  
TempS=19.10
```

On va donc, en lisant ligne par ligne, identifier les informations à enregistrer en fonction de leur préfixe.

On sauvegarde donc les valeurs dans notre base de données InfluxDB avec les bons types

Evolutions futures

Les modifications pouvant être apportée dans le futur sont :

- L'intégrations de plus de capteurs sur une seule et même carte pour permettre de mettre plusieurs capteurs dans une même serre.
- On peut aussi penser à une compatibilité multi-serres ou bien même d'autre installation de la ferme. Le contrôle du niveau d'eau du réservoir principal est une piste possible, le contrôle de la présence d'un gros animal (prédateur, humain, ..) dans les poulaillers en est une autre.
- On sait que l'on est capable d'automatiser le relevé des données, on pourrait imaginer automatiser l'arrosage, l'ouverture des portes/fenêtres. En d'autres termes, automatiser la régulation de la serre.

Conclusion

Ce projet a été intéressant d'un point de vue technique car il nous a permis de manipuler plusieurs outils différents. Tous les outils étant open-source nous avons pu nous documenter comme il fallait et apprendre rapidement. Nous pouvions voir l'avancé du projet en temps réel car les tâches étaient bien réparties en sous-tâches. Le deuxième avantage étant que l'on a eu la chance de travailler sur un sujet concret avec une utilisation directement derrière. En effet, quand nous avons rencontré les fermiers, ceux-ci nous ont fait part de leur besoin et nous avons pris cela très à cœur d'essayer de les aider.

Nous avons rencontré quelques difficultés avec le compilateur en ligne mbed. En effet son utilisation était parfois un peu aléatoire et un même code compilait à un instant t puis ne compilait plus juste après : plutôt bizarre et un peu rageant.

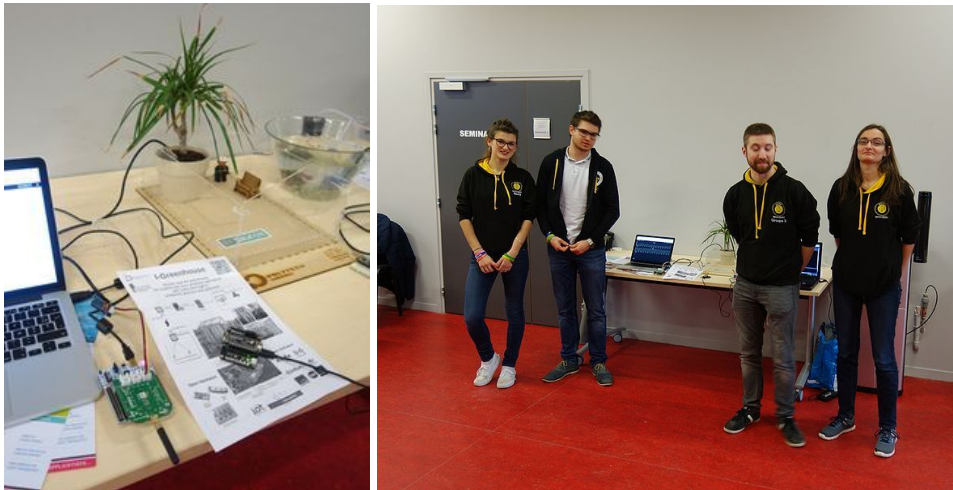
Ce projet nous a donc permis d'apprendre à utiliser des technologies que nous ne connaissions pas forcément et de te pouvoir mener à bien un projet avec une réelle attente derrière.

Annexe : Événements

Durant ce projet nous avons eu la chance de participer à deux événements pour présenter notre projet ainsi que pour échanger avec différentes personnes connaissant les technologies que nous utilisons.

1. IOT Eclipse Day Grenoble 2017

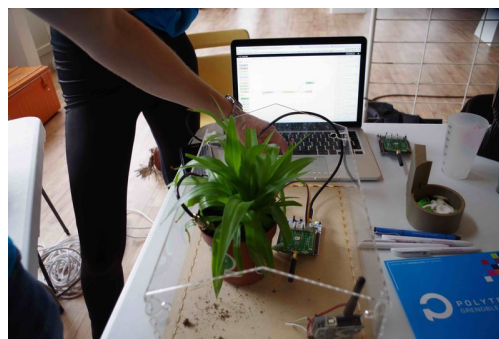
C'est une journée où l'on peut assister à plusieurs conférences sur le thème des objets connectés. Cette journée regroupe donc beaucoup de personnes travaillant sur ce sujet et voulant être au courant des nouvelles technologies de ce domaine. C'était donc pour nous une bonne occasion de voir ce que ce projet pouvait nous apporter en termes de connaissance mais aussi pour parler de notre projet et obtenir des conseils.



Démonstration de notre projet à mi-parcours et une photo de notre binôme ainsi qu'un autre travaillant aussi sur des objets connectés

2. Maker Faire Grenoble 2017

La Maker Faire c'est un salon qui regroupe beaucoup de personnes différentes. Allant des grandes entreprises à des particuliers, qui innove, bricole.. c'est le rendez-vous des techniques et de la créativité, des arts et des sciences, des inventions les plus ambitieuses aux plus amusantes.



Démonstration lors de la Maker Faire