Projet RICM4 - Polytech Grenoble

Serres connectées

Timothée DEPRIESTER

Guillaume BESNARD

Enseignant : Nicolas Palix

PRÉSENTATION

Ce projet a été réalisé pour la ferme “Les Jardins du Coteau” situé à Saint-Cassien près de Voiron. L’objectif de ce projet est d’équiper des serres avec des objets connectés afin de faire des relevés d’information.

Les conditions dans une serre sont beaucoup plus variables qu’en extérieur avec des phénomènes physiques tel que la condensation de l’eau, l’effet de serre, … L’humidité et la température sont donc des choses à surveiller et contrôler afin d’optimiser au mieux la pousse des plantes. De plus, certaines températures et taux d’humidité sont plus propices à la propagation de bactéries ou d’insectes.

Il y a donc un besoin de pouvoir observer en temps réel les conditions climatiques dans les serres avec une courbe de l’évolution. Il faut pouvoir discerner chacune des serres et analyser cette information rapidement.

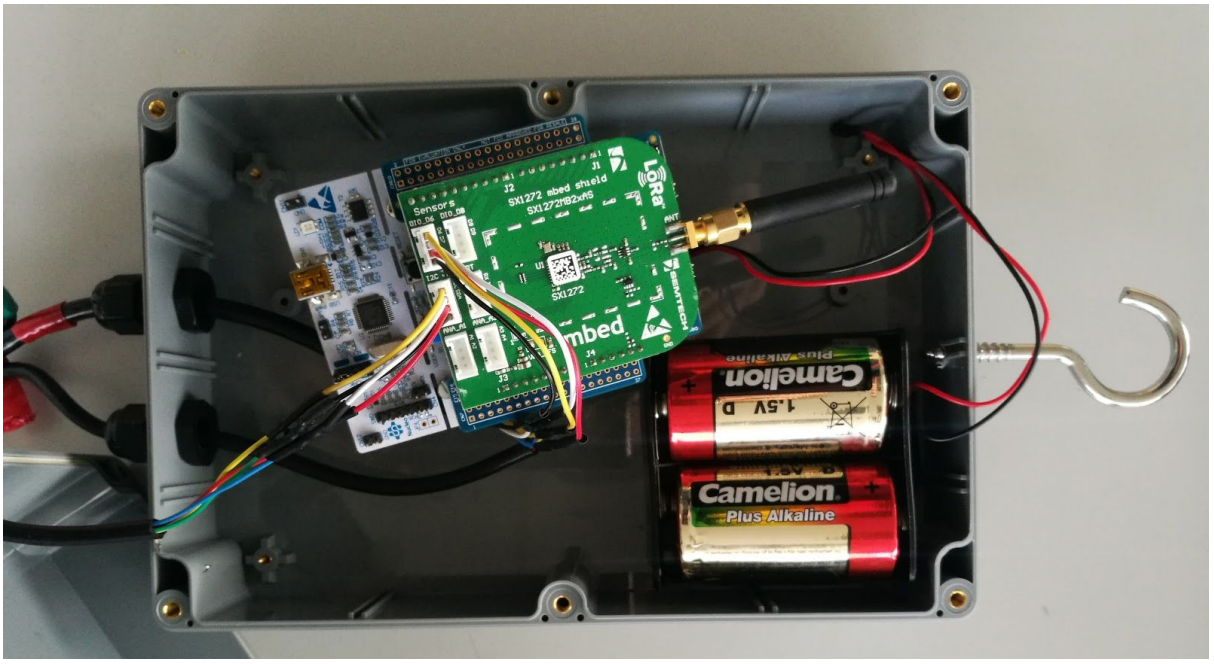
Ces conditions amènent des contraintes supplémentaires sur notre projet, les cartes électronique hébergeant les capteurs doivent être autonome ou du moins avoir une batterie suffisante pour ne pas avoir à faire trop de maintenance. De plus, ces cartes doivent pouvoir résister à des taux d’humidité et des températures élevées.

CHAÎNE D'INFORMATION

L'extrémité de la chaîne d'information est une carte STM32 équipée de multiple capteurs et d'un transmetteur LoRa. Ces cartes sont branchées à une batterie et envoient des paquets à intervalles réguliers (ici 1 heure). Cette technologie transmet sur la fréquence libre 868 MHz (en Europe) et a des avantages comme la possibilité de transmettre sur de la très longue portée et même dans les environnements fermés (bâtiments, ville, …).

Chacune des cartes sera flashée avec un binaire différent : le programme sera le même mais les identifiants varient selon l'emplacement de la carte (numéro de serre, …). Nous verrons dans les évolutions futures un peu plus bas dans le rapport un moyen de contourner ce problème.

Ces capteurs doivent respecter la contrainte d'être facilement déplaçables et résistants, ils sont donc enfermés dans une boite hermétique et facile à mettre en place avec un système de crochet pour pouvoir l'accrocher aisément à la structure de la serre.



Ces paquets sont reçus par le portail qui est lui même relié en USB sur un Raspberry Pi. Nous avons ensuite déployé un système Docker permettant l'automatisation de l'installation des solutions mis en place. Il suffit de lancer une ligne de commande pour que cela soit complètement fonctionnel.

Ces solutions consistent en le lancement de plusieurs **conteneurs**[1] docker comme **Node Red** : notre première brique logiciel. Ce logiciel nous permet de récupérer le flux UDP en sortie de notre gateway LoRa, de le traiter et ensuite de le dispatcher.

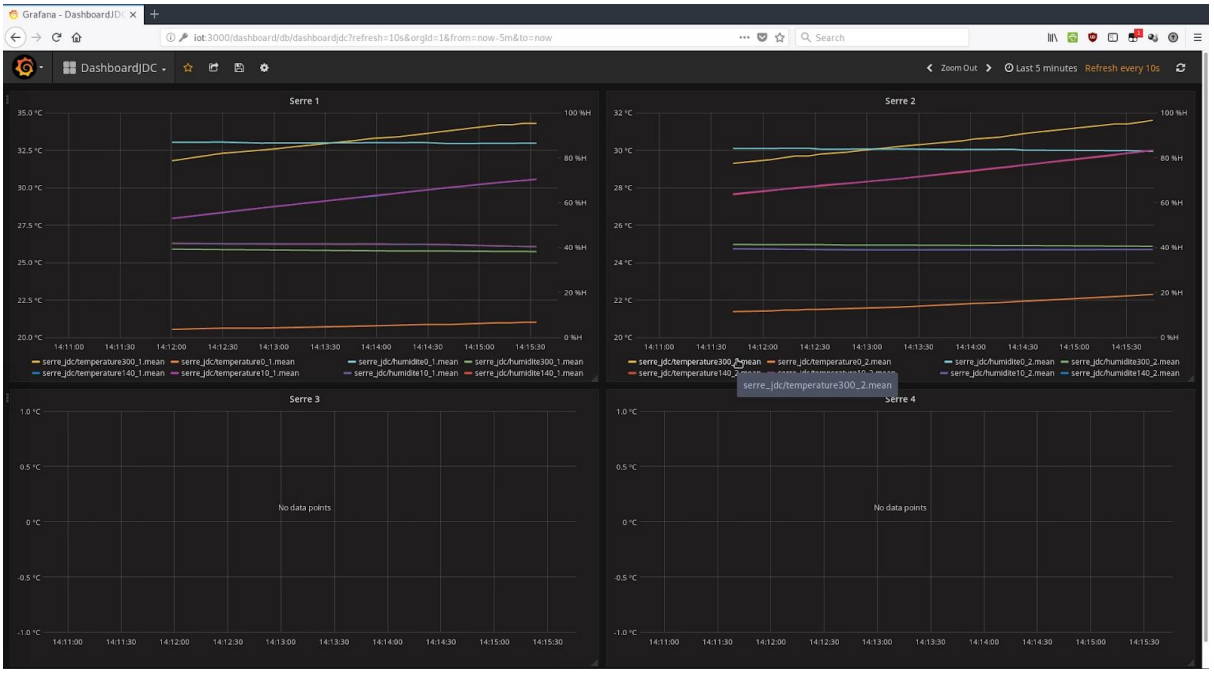
Cette nécessité de dispatcher les paquets vient du fait que nous sommes plusieurs groupe d'étudiants à utiliser cette technologie, il faut donc séparer et réattribuer les paquets. Le protocole d'envoi sera détaillé un peu plus tard dans le rapport. Ces données sont ensuite envoyées via le réseau à l'aide de MQTT[2] (dans notre cas le réseau est interne au Raspberry Pi mais facilement adaptable pour envoyer nos paquets sur le réseau vers un serveur distant).

Ce flux MQTT est donc récupéré par une autre brique logiciel Node Red qui elle va s'occuper du traitement interne au projet de notre groupe (identifiants de la serre, découpage des données renvoyées par les capteurs). Ces données sont ensuite transférées sur le port d'entrée de notre deuxième brique logiciel : InfluxDB. Elle nous permet de stocker les données des capteurs en association avec leur date d'arrivée; elle est très appropriée pour ce cas d'utilisation. Finalement, la dernière brique logiciel : Grafana. Cette dernière permet une mise en place rapide d'une visualisation très adapté sous forme de courbe des données en picorant directement les informations dans la base de données InfluxDB.

---

[1] Environnement plus ou moins isolé du système permettant de lancer une image personnalisé d'une brique logiciel

[2] MQTT (MQ Telemetry Transport) est un protocole de messagerie publish-subscribe basé sur le protocole TCP/IP



Note : *Tous les logiciels utilisés sont open-source et utilisable librement.*

PROTOCOLE ACTUELLE

D'un point de vue radio, voici nos choix d'implémentation :

On utilise une fréquence de 868.3 MHz (configurée sur nos capteurs et la gateway). Notre largeur de bande est de 125 KHz. Nous utilisons actuellement un spreading factor de valeur 7 ainsi qu'une *coding rate* de ⅘ (*Cf.* partie évolution).

En ce qui concerne le protocole de nos trames :

Nous étions 3 groupes de RICM travaillant sur des projets distincts mais utilisant une gateway commune. Il était donc nécessaire de disposer de champs dans nos paquets permettant de distinguer ces groupes. Une autre distinction nécessaire, spécifique à notre projet, doit être de distinguer les différentes serres des Jardins du Coteau.

Ainsi notre premier octet est de la forme

`0x31`, soit

`group` `= 0011 XXXX` (groupe 3) : aiguillage au niveau de la gateway

`id_serre = XXXX 0001` (serre 1) : aiguillage serre

Par exemple, pour notre visite, disposant de 4 boitiers, nos cet octets varié de `0x31` à `0x34`.

Ensuite, disposant de 4 capteurs par boitier, permettant chacun d'acquérir une mesure de température et d'humidité, nous disposons de 8 valeurs par trame. Cette valeur est un `float` stocké sur 4 octets, ajoutant donc 32 octets. Notre trames est donc au final de la forme :

`1 octets id  ||  4 * 8 octets de données`

avec dans l'ordre :

humidité puis température IKS01A2 (300 cm)

humidité puis température SHT31 (140 cm)

humidité puis température SHT31 (10 cm)

humidité puis température SHT10 (0 cm)

(*Cf*. Évolutions pour les améliorations du protocole).

ORGANISATION GIT

Lien gridcad : https://gricad-gitlab.univ-grenoble-alpes.fr/Projet-RICM4/17-18/11

Notre répertoire est composé de trois projets :

- `serres-jdc` : notre projet principal, contenant de la documentation, nos codes embarqués Mbed, nos fichiers de configurations (notamment ceux de la gateway et des conteneurs Docker) ainsi que des sous-modules vers les dépendances (*drivers* Lora).
- `LoRarvest` : contenant les configurations Docker **communes** aux différents groupes RICM travaillant sur la même gateway (Aquaponie et Ruches) ainsi que des sous-modules vers les répertoires des trois groupes.
- `firmware-bin` : contenant les binaires utilisés lors de notre visite aux Jardins du Coteau (triés selon la fréquence d'envoie et le numéro du boîtier)

GUIDE UTILISATEUR

Comment "INFORMATISER" vos serres :

- Commencer par installer les boîtiers dans vos serres en prenant soin de placer les capteurs à la bonne hauteur (sol, air).
- Brancher la gateway au port USB du raspberry puis brancher la raspberry au secteur.
- [Optionnel] Installer le logiciel Bonjour d'Apple (pour Windows et Mac) ou Avahi sous linux.
- Accéder aux données à l'adresse iot:3000 dans votre navigateur internet préféré (Chrome, Firefox, Edge, Opera). A défaut, si Bonjour/Avahi n'est pas installé, vous pouvez y accéder à l'aide de l'adresse du raspberry trouvable facilement sur le panel administrateur de votre box (généralement à l'adresse internet 192.168.0.1).

VISITE

La visite a été pour nous le moment le plus intéressant et gratifiant du projet. Elle nous a permis de voir la concrétisation réel du travail que nous avions fourni. Les boîtiers sont installés dans les serres et relèvent les informations, le Raspberry Pi et en place et fonctionnel. Et nous avons pu visualiser les résultats directement sur l'ordinateur de Vincent Hibeon, le mari de la responsable de l'exploitation.

Il semblait satisfait et nous a envoyé un mail nous remerciant pour le travail et l'installation sur le site.

Elle nous a aussi permis de rencontrer Benoit Marchand ainsi qu'une équipe de communication de STMicroelectronics (ST qui nous a fourni le matériel nécessaire). L'échange fut fructueux, et nous en avons tiré plusieurs pistes d'améliorations. Il est, d'autant plus, toujours intéressant d'avoir un regard extérieur de la part d'un professionnel.

Pour conclure, ce fut motivant au point qu'avec certains IESE nous réfléchissons à pousser le projet et pourquoi pas monter une startup.

EVOLUTIONS FUTURES

Suite à notre visite sur site, plusieurs retour ont mis en valeurs des évolutions possibles et/ou nécessaires :

Benoit Marchand nous a fait remonter que nous devrions utiliser plus d'un octet (4 suggérés) pour nos ID de serres dans le protocole, en vue d'une potentiel forte augmentation. De la même manière, une évolution future souhaitée et de faire remonter une donnée sur le niveau restant des batteries d'un boîtier (quelques octets supplémentaires envisageables).

Une autre idée remontée serait de versionner en plus de nos sources, nos binaires. Par exemple inclure le SHA de notre git dans une version, pour pouvoir directement connaître une version déployé et ainsi rendre plus robuste nos process de déploiement.

Dans la même thématique, une idée intéressante serait de se "débarrasser" de la dépendance de Mbed Compiler pour pouvoir effectuer nos compilations en local, et ainsi pouvoir versionner directement via git, sans passer par le système de Mbed. Le dossier *doc* dans notre répertoire git est un début, permettant la compilation en local avec une *toolchain ARM*. Les projets Mbed peuvent ainsi s'exporter depuis Mbed et se compiler localement. Les problèmes subsistant résident dans l'importation et la recompilation si nécessaire des librairies (*drivers* de cartes et de capteurs).

Un problème restant relativement important est la perte d'environ un paquet sur trois dans les "airs". Des tests plus poussés sur nos configurations de la gateway/carte serait de mises. Même si selon toute vraisemblance le spreading factor minimum de 7 devrait être largement suffisant au vu des distances mise en jeu, il peut être intéressant de tester des variations sur ce paramètre. De la même manière, une largeur de bande plus importante (256 ou 512 KHz) entraîne la perte de tous les paquets. Pourquoi ?

Des tests de stockages seraient bienvenues, notamment pour estimer la longévité de la carte SD sur le raspberry pi (gateway). Il faudrait aussi estimer l'espace mémoire utiliser par les données récoltées stockées dans InfluxDB. Un disque dur externe serait intéressant pour pallier à la faible résistance de la carte SD aux écritures/lectures régulières.

Lors de notre visite au JDC, nous avons flashé chaque carte avec un ID de serre "codé en dur", entraînant donc un binaire par boitier. Il serait intéressant d'implémenter une initialisation de cette ID par la liaison série. Des tests (disponible sur notre git) ont été effectués pour écrire dans l'EEPROM de la carte (mémoire non volatile). Il manque donc la "brique" lecture par le port série pour finaliser cette fonctionnalité.
Un autre groupe (le 12, travaillant sur les serres en aquaponie) a travaillé sur LoraWAN. Ce protocole permettrait, entre autre, de résoudre cette initialisation via un *handshake* lors de l'arrivé d'un émetteur.

Enfin, il est aussi envisageable qu'un groupe continue sur la *data visualisation*. Grafana, pratique de par sa rapidité de mise en place, est il suffisant aux besoins ? Une réflexion pourrait aussi être portée sur la capacité de la solution visuelle à s'adapter automatiquement au nombre de serres connectées : l'arrivée d'une nouvelle serre entraînerait automatiquement la création d'un nouveau graphe.

ANNEXE : ANNIVERSAIRE RICM

Nous avons eu l’occasion de présenter notre projet lors de l'anniversaire de la filière; nous avons pu confronter notre projet à des ingénieurs diplomés et récolter des avis extérieurs.

A L'occasion de cet événement, nous avons réalisé un poster que l’on peut voir en arrière plan.