



Ruches connectées

Systeme de récupération de données par radio

Contexte	2
Le projet	3
- Un système embarqué	3
- Une récupération automatique	3
Les technologies	4
Développement	4
Projet	4
Le matériel	5
Prototype	6
Schéma du montage	6
Envoi de données	6
Réception et inscription dans la base de données	7
Démarrage des serveurs	7
Lecture du port série et envoi dans la base de données	7
Affichage et accès aux données	8
Conclusion	11

Contexte

De nos jours, nous savons que les abeilles domestiques en France sont en danger, dû à l'urbanisation et au non-respect des espaces sauvages. En effet, un grand nombre d'abeilles meurent chaque année, c'est pourquoi des plans de réintroduction dans les villes sont mis en place.

Des ruches sont ainsi placées sur des toits de bâtiments publics, immeubles ou maisons, ce qui aide à protéger la colonie à l'écart de risques naturels et humains.

Cependant, une ruche est un logement qui nécessite une grande supervision de la part de l'apiculteur, tout d'abord pour récolter le miel mais surtout pour mesurer différents facteurs qui pourraient mettre en danger la colonie, comme par exemple la température de la ruche, puisqu'une colonie comme celle-ci est sensible à l'environnement.

C'est pourquoi au cours des projets innovants de la quatrième année d'étude RICM, un projet a été créé afin de pouvoir palier à ce problème. En effet, il consiste à créer un système de supervision à distance afin de venir en aide à l'apiculteur. Il est découpé en deux parties distinctes, la première étant un système embarqué à l'intérieur de la ruche, comprenant des capteur de température, pression et autres, ainsi qu'une carte d'envoi des données par radio. La seconde partie est entièrement informatisée, et permet de traiter les données reçues et de les afficher sur un navigateur internet.

Ce projet était à réaliser dans un délai de 3 mois, utilisant des technologies nous permettant d'utiliser les compétences informatiques apprises jusqu'à maintenant, et à en apprendre de nouvelles par nos propres moyens. Un projet de ce type a été réalisé pour la première fois par des étudiants à l'INP, nous partions ainsi de zéro avec seulement deux cartes de réception et d'émission radio ainsi que de capteurs, que nous verrons plus tard.

Le projet

- Un système embarqué

La première partie de ce projet correspond à la partie qui sera embarquée à l'intérieur de la ruche. Elle est constituée de deux parties :

- La première correspond aux différents capteurs qui peuvent être embarqués, pour l'instant, des capteurs de température, de pression et de poids ont été implémentés dans le reste du projet. Cependant, il est possible d'en rajouter ou d'en enlever facilement.
- La seconde est le traitement de ces données du côté de la ruche. En effet, les données sont envoyées directement sur la première carte radio LoRa, cette dernière les traite et ensuite envoie un paquet de données sur la fréquence 868 MHz contenant ces données.

- Une récupération automatique

La seconde partie correspond à la partie informatique du projet. Elle comprend 4 parties :

- Pour recevoir les informations envoyées par le système embarqué, nous avons connecté par un port série de l'ordinateur (port USB) une seconde carte radio LoRa. Celle-ci récupère tous les paquets arrivant sur la fréquence identique à l'envoi, traite la réception de données, et si le paquet reçu correspond à un envoi de donnée du système embarqué, il le transfère à l'ordinateur via le cordon USB.
- Afin de pouvoir récupérer l'entrée de données via le port série USB, nous avons développé un programme en Java qui se connecte au port, puis attend qu'une donnée arrive par celui-ci. Lorsqu'une donnée est reçue, le programme la met en forme puis l'ajoute au sein d'une base de données locale.
- Cela rejoint donc la troisième partie qui correspond au stockage de données. Afin de les stocker correctement, nous avons décidé de passer par une base de donnée PhpMyAdmin. L'utilisateur ainsi que le mot de passe pour y accéder peut être facilement modifié, ce qui sécurise celle-ci.
- La dernière partie nous donne une représentation visuelle des données reçues. Toujours en local, un serveur HTTP est lancé, et de celui-ci nous

pouvons accéder à une page. Par défaut, celle-ci nous donne la dernière valeur collectée par la carte radio LoRa. Nous avons pu de plus ajouter une option permettant de sélectionner le jour pour lequel nous voulons voir les valeurs. Ainsi, la page affichera toutes les valeurs (dans l'ordre chronologique) capturées dans la journée.

Les technologies

Développement

- NetBeans : IDE développé par Oracle, il nous permet notamment de créer, et modifier des pages HTML et PHP dans notre cas.
- Eclipse : Autre IDE multi-usage, utilisé pour fabriquer des programmes Java.
- Mbed : C'est une plateforme permettant de coder en ARM et offrant des bibliothèques pour les micro-contrôleurs. Les programmes ainsi créés peuvent être compilés afin de les intégrer directement à nos cartes LoRa.
- Tera Term : C'est un émulateur de terminal open source, nous permettant de monitorer les entrées et sorties de cartes connectées en série.

Projet

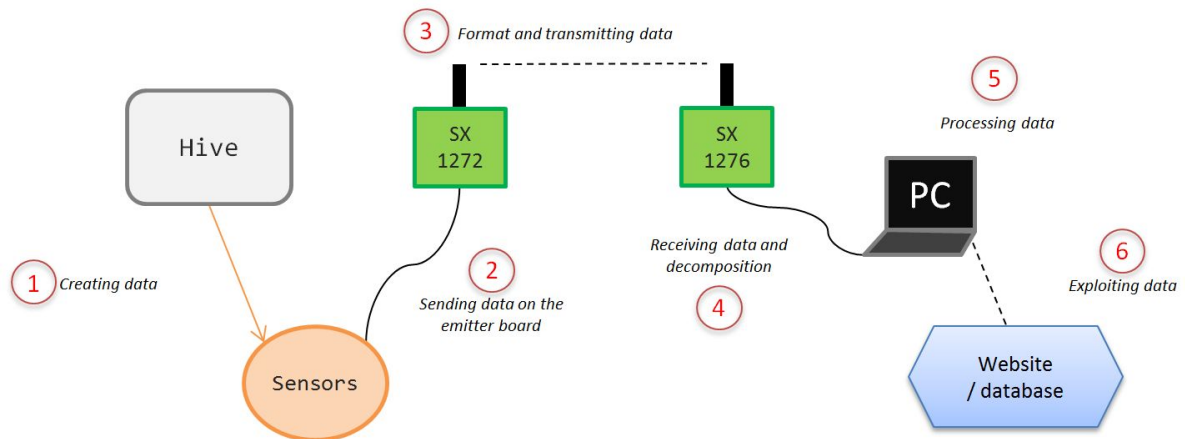
- EasyPhp : EasyPhp est un logiciel de développement ainsi qu'un serveur pouvant héberger des serveurs locaux Web, base de données, etc...
Nous avons utilisé son package phpMyAdmin pour héberger notre propre serveur de base de données local, ainsi qu'un serveur Apache, afin de pouvoir afficher notre interface Web de données.
- LoRa : Cette technologie de transmission par radio nous permet, au travers de cartes d'émission et de réception spécifiques, de pouvoir échanger des messages sur une distance pouvant couvrir jusqu'à 20 kilomètres, ce qui pourrait couvrir toute la métropole grenobloise.
- Java : Le Java est un langage informatique existant depuis de nombreuses dizaines d'années, et souvent utilisé dans les technologies embarquées et pour réaliser des applications. Ici, il permet de créer une application qui tournera en arrière-plan de la machine.

Le matériel

- Cartes Nucleo F411RE et F446RE : Ces cartes programmables possèdent un port série en mini-USB pour une connexion avec un ordinateur. Cette liaison nous permet d'intégrer dans la carte le code précédemment compilé sur Mbed, mais aussi de récupérer les informations renvoyées par la carte, et enfin de l'alimenter en énergie. Ces cartes possèdent également un emplacement pour ajouter un module (voir ci-dessous), ainsi que des Leds et deux boutons utilisateurs programmables.
- Émetteurs-transmetteurs en radiofréquences SX1272 et SX1276 : Ces deux modules de transmission radio utilisent la technologie LoRa vu précédemment, et sont respectivement montées sur les deux cartes Nucleo F411RE et F446RE. Ces deux modules sont identiques, à l'exception qu'ils n'ont pas la même gamme de fréquences; le module SX1272 ne possède qu'une antenne haute fréquence couvrant de 860 à 1000 MHz, tandis que le module SX1276 possède deux antennes, une basse et une haute fréquence, ce qui induit une couverture de 137 à 1020 MHz.
- Capteurs : Les capteurs peuvent être de tous types, du moment qu'ils soient compatibles avec les technologies déjà en place. Il suffit de récupérer les valeurs mesurées grâce aux fonctions présentes dans les bibliothèques associées et à envoyer ces données.

Prototype

Schéma du montage



Envoi de données

Les données sont envoyées automatiquement toutes les heures. La carte émettrice envoie alors une chaîne de caractères contenant les données récoltées par les capteurs au moment de l'envoi.

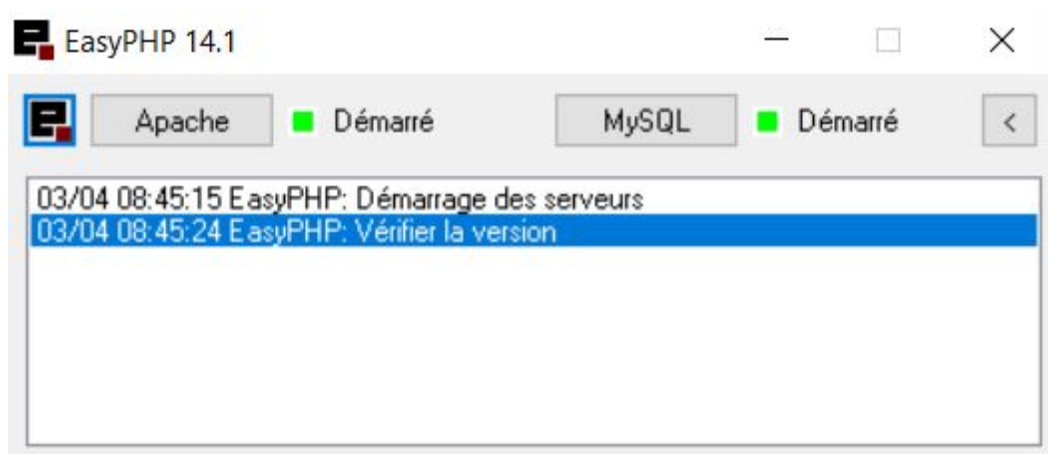
Nous pouvons suivre l'évolution de celle-ci grâce au logiciel Tera-Term présenté précédemment, comme nous pouvons le voir sur la capture d'écran ci-dessous qui affiche les messages de 3 envois de données différents.

```
Sending data...  
...Data sent : 30.5;420.0;3.05  
> OnTxDone  
Sending data...  
...Data sent : 30.5;420.0;3.05  
> OnTxDone  
Sending data...  
...Data sent : 30.5;420.0;3.05  
> OnTxDone
```

Réception et inscription dans la base de données

Démarrage des serveurs

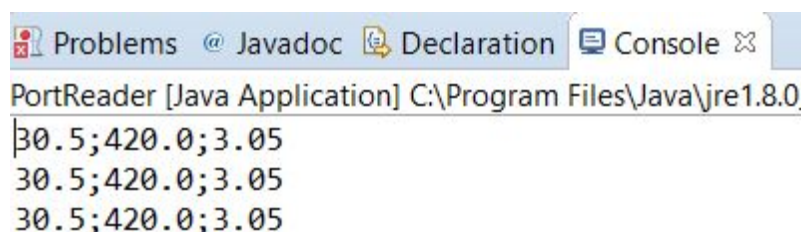
Les serveurs Apache & MySQL sont lancés par le biais du serveur EasyPHP, dont on peut observer le fonctionnement par la petite fenêtre ci-dessous :



Lecture du port série et envoi dans la base de données







































Afin de pouvoir lire les données reçues par la carte réceptrice, il faut tout d'abord installer un programme Java qui tournera en fond afin de se connecter au port série.

Nous pouvons voir sur l'image ci-dessous que l'on peut observer les réceptions grâce à Eclipse, afin de suivre l'évolution des envois radios (ici les 3 envois précédents).



Les données que l'on reçoit sont ensuite envoyées également par le programme Java sur une base de donnée locale appelée "connectedhive" afin de sauvegarder les données de façon permanente. Nous pouvons y accéder facilement grâce au serveur EasyPHP avec l'URL <http://127.0.0.1/> et en navigant jusqu'à l'onglet *Base de données*.

On arrive sur le serveur MySQL local, où l'on retrouve la table suivante :

			Date	Heure	Temperature	Poids	Pression	
<input type="checkbox"/>	 Modifier	 Copier	 Effacer	2017-03-29	10:31:22.000000	35.2	400.5	3.25
<input type="checkbox"/>	 Modifier	 Copier	 Effacer	2017-03-29	10:32:10.000000	35.5	405.5	3.24
<input type="checkbox"/>	 Modifier	 Copier	 Effacer	2017-03-28	19:31:04.000000	32.4	350.4	3.28
<input type="checkbox"/>	 Modifier	 Copier	 Effacer	2017-03-29	14:44:12.000000	23.1	4.2	460
<input type="checkbox"/>	 Modifier	 Copier	 Effacer	2017-03-29	14:59:22.000000	23.1	4.2	460
<input type="checkbox"/>	 Modifier	 Copier	 Effacer	2017-03-29	15:01:56.000000	30.5	420	3.05
<input type="checkbox"/>	 Modifier	 Copier	 Effacer	2017-03-29	15:02:05.000000	30.5	420	3.05
<input type="checkbox"/>	 Modifier	 Copier	 Effacer	2017-03-29	15:05:27.000000	30.5	420	3.05
<input type="checkbox"/>	 Modifier	 Copier	 Effacer	2017-03-29	15:06:13.000000	30.5	420	3.05
<input type="checkbox"/>	 Modifier	 Copier	 Effacer	2017-03-29	15:06:16.000000	30.5	420	3.05
<input type="checkbox"/>	 Modifier	 Copier	 Effacer	2017-03-29	15:06:19.000000	30.5	420	3.05
<input type="checkbox"/>	 Modifier	 Copier	 Effacer	2017-03-29	15:06:29.000000	30.5	420	3.05
<input type="checkbox"/>	 Modifier	 Copier	 Effacer	2017-03-29	15:06:33.000000	30.5	420	3.05

On y retrouve des données sous la forme d'une date et l'heure de la réception, puis les 3 données (température, poids et pression) qui y ont été inscrites. Les 3 envois précédents se retrouvent bien ajoutés à cette table à l'heure exacte.

Affichage et accès aux données

Pour accéder aux données, nous avons mit en place grâce à EasyPHP également un serveur Apache, qui nous permet d'afficher les données plus proprement et aisément qu'avec une base de donnée MySQL classique.

La page que nous avons créée est alors à ajouter dans le dossier que l'on veut, il ne nous manque plus alors qu'à créer un *alias* sur Apache afin d'y accéder. Pour ajouter un alias, il suffit de cliquer sur le bouton "Ajouter un alias" accessible à l'URL : <http://127.0.0.1/home/index.php> et visible ci-dessous :

FICHIERS LOCAUX ?

Nous arrivons sur la page d'ajout d'un alias, où il suffit de remplir la première case par le nom que vous voulez (ici *connectedhive*), et la seconde par le chemin d'accès sur votre ordinateur comme ci-dessous :

AJOUTER UN ALIAS

fermer

1. **Créer votre répertoire** (ex.: C:\weblocal\sites\site1)

2. **Saisir un nom pour l'alias** (ex.: site1)

connectedhive

3. **Saisir le chemin du répertoire créé** (ex.: C:\weblocal\sites\site1)

C:\Users\

Project\connectedhive\

En cas de problème, ouvrir le répertoire '[EasyPHP-DevServer folder]/binaries/conf_files/', effacer ou renommer 'php.ini', renommer la sauvegarde la plus récente 'php.ini' et redémarrer EasyPHP.

J'ai lu l'avertissement ci-dessus » Enregistrer

Une fois l'alias vers le dossier local ajouté, et les pages PHP fournies ajoutées dans ce dossier, la page d'affichage des données est accessible à l'adresse <http://127.0.0.1/connectedhive>, *conectedhive* étant remplacé par l'alias que vous avez choisi plus tôt.

Nous arrivons donc sur la page par défaut, qui nous affiche la dernière valeur enregistrée par notre installation, ici datant du 29-03-2017 à 15:06:33 :

Last measurement to date was the 2017-03-29 at 15:06:33

Temperature was at 30.5 degrees.

Weight was at 420 kg.

Pressure was at $3.05 \cdot 10^5$ Pascals.

Directly choose a date :

1 ▼ 1 ▼ 2017 ▼ Display this day

Nous pouvons voir qu'il y a en bas de page un formulaire permettant de choisir une date. Celui-ci sert à afficher toutes les valeurs d'une date donnée, par exemple si l'on choisit d'afficher les valeurs du 29-03-2017, nous arrivons sur une nouvelle page :

Hive's sensors's measurement for the 2017-3-29

There is a measurement on 2017-03-29 at 15:06:33

Temperature was at 30.5 degrees.

Weight was at 420 kg.

Pressure was at $3.05 * 10^5$ Pascals.

There is a measurement on 2017-03-29 at 15:06:29

Temperature was at 30.5 degrees.

Weight was at 420 kg.

Pressure was at $3.05 * 10^5$ Pascals.

There is a measurement on 2017-03-29 at 15:06:19

Temperature was at 30.5 degrees.

Weight was at 420 kg.

Pressure was at $3.05 * 10^5$ Pascals.

There is a measurement on 2017-03-29 at 15:06:16

Dans le cas où on choisirait une journée sans relevé existant dans la base de données, un message nous l'indiquant est écrit, et nous sommes invités à revenir sur la page d'accueil.

Conclusion

En conclusion, nous avons produit une installation facilement modifiable et modulable, car nous pouvons aisément rajouter de nouveaux capteurs à l'intérieur du système embarqué.

Notre projet peut être également appliqué pour d'autres situations, notamment de la domotique, où n'importe quel projet nécessitant une transmission radio par une technologie LoRa.

Au cours de ce projet, nous avons acquis de l'expérience sur de nouvelles technologies, comme par exemple la transmission par radio LoRa, mais aussi son application dans son langage de programmation. Nous avons pu aussi pour la première fois lier plusieurs langages de programmation différents, ici le C++, Java, PHP et SQL.

Notre projet est disponible sur GitHub à l'adresse suivante :

<http://www.github.com/GwenaelMoreau/connectedhive>