BTS Système Numérique
Option : Informatique et réseaux
Projet technique : U62

Lycée Costebelle
83400 HYÈRES

académie Nice

MINISTÈRE DE L'ÉDUCATION NATIONALE

MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE

Liberté • Égalité • Fraternité
RÉPUBLIQUE FRANÇAISE

PROJET COMMUN :

RUCHE CONNECTÉE

Membres de l'équipe projet :
- Florian DEVIGNE
- Antoine DUDUC
- Dorian COLAS
- Loan PÉSERY

Session 2021

SOMMAIRE

INTRODUCTION..... 4
CAHIER DES CHARGES 4
1. Composition et schéma du projet .....
2. Performances envisagées 13
3. Contrainte de développement..... 15
4. Contrainte de qualité
5. Contrainte de fiabilité et de sécurité
6. D
7. D
8. D
9. Budget
DIAGRAMME / ANALYSE UML 5
1. Diagramme de cas d'utilisation
2. Diagramme de séquence
3. Diagramme d'exigence
4. Diagramme de déploiement
5. Diagramme de définition de bloc
ORGANISATION DU TRAVAIL EN ÉQUIPE
1. Répartition des tâches
2. Planning
ÉLÉMENTS NÉCESSAIRES À LA RECETTE DE LA MAQUETTE

INTRODUCTION

Les abeilles subissent une mortalité accrue chaque année, principalement en raison des pesticides présents dans l'environnement, auxquels elles sont particulièrement sensibles des contraintes d'environnement météorologique ou par d'autres espèces d'insectes comme le frelon asiatique.
Le but du projet consiste à développer un système autonome permettant d'équiper facilement chaque ruche d'abeilles classiques en y ajoutant des équipements miniatures pour obtenir la température, l'humidité de la ruche ainsi que son poids. Mais aussi pour obtenir la température, l'humidité et la vitesse du vent près du rucher, nécessaire à l'apiculteur pour évaluer la santé de son essaim et surveiller à distance l'activité de ses ruches. L'ensemble des données collectées peuvent être visualisées à distance à partir d'un navigateur internet classique ou à l'aide d'un smartphone.

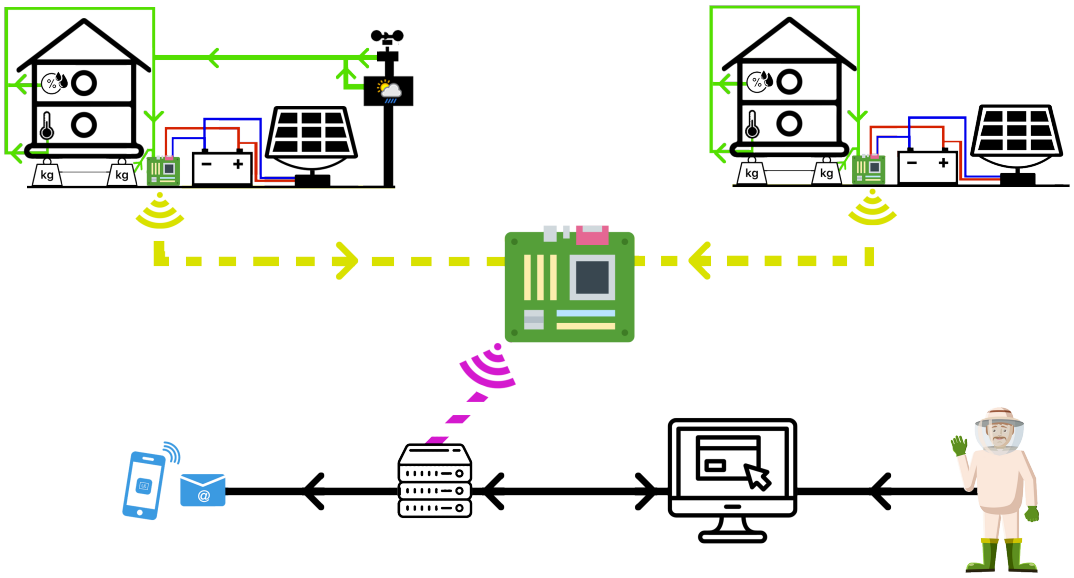
D'autre part, en cas d'alerte critique, et de manière journalière sous forme de récapitulatif de mesures, une notification est envoyée à l'apiculteur par e-mail ou SMS, dépendant du seuil de notification déterminé au préalable par l'apiculteur.

CAHIER DES CHARGES

1. Composition et schéma du projet

Le système qui compose le projet est composé de plusieurs parties :

- Le **site web** permettant la gestion et la visualisation des ruches.
- La **base de données** permettant le stockage et la répartition des données.
- La **passerelle** permettant la communication entre la base de données et les ruches/ station météo.
- La **station météo** permettant l'acquisition de la température, de l'humidité et du vent extérieur de l'environnement des ruches.
- Le système des **ruches** permettant l'acquisition de leur température et humidité interne ainsi que leurs poids.
- La partie gestion de **notifications** permettant à l'apiculteur d'être informé sur l'état de ses ruches.



2. Performances envisagées

Le système doit répondre impérativement aux besoins ci-dessous :

- Système autonome en énergie, sur minimum 1 an.
- Distance maximum de 5 km entre la passerelle et le rucher.
- Installation du système facile et compacte.
- Système totalement étanche et résistant aux intempéries.
- Application web facile à utiliser et sécurisée.
- Il ne doit pas gêner les abeilles ainsi que l'apiculteur.
- Il ne doit pas endommager la ruche lors de l'installation du système.

3. Contrainte de développement

- C++, Python 3.5, Shell.
- HTML, CSS, JavaScript, PHP.
- MySQL, Apache 2.
- Site responsive.

4. Contrainte de qualité

- Facile d'utilisation, intuitif et conviviale.
- Entête de fichier précisant auteur, date de création, de dernière modification, outils de production utilisés.
- Entête de fonctions précisant le rôle de la fonction et l'utilisation des paramètres.
- Tous les commentaires nécessaires à une bonne compréhension du code.

La documentation doit être complète, homogène et non redondante. L'auteur de chaque page est identifiable.
Elle comporte :

- Un dossier d'analyse permettant de déterminer le périmètre du projet, la description complète des solutions retenues, les prototypes des IHM et la planification des différentes étapes du projet ainsi que le cahier de recette.
- Un dossier de conception avec une première partie permettant de définir l'architecture de l'application et les échanges entre chaque module. La deuxième partie regroupe les algorithmes des modules complexes et les fiches de test unitaire permettant de valider chaque partie individuellement.
- Un dossier de réalisation expliquant les technologies utilisées, les points clés du codage sans pour autant le reprendre et les résultats de test unitaire.
- En annexe la nomenclature détaillée des éléments constitutifs avec leur prix et sources d'approvisionnement.
- Un guide utilisateur détaillant le guide de montage, d'utilisation et de réparation.

5. Contraintes de fiabilité et de sécurité

- Une partie du système est destiné à être installé sur un réseau accessible de l'internet, il doit être robuste et sécurisé.
- Tous les appareils installés sur le terrain doivent pouvoir résister aux intempéries.
- Les éléments dans la ruche doivent prendre en compte les abeilles, ne pas les déranger ou perturber leurs sens.
- Les éléments dans la ruche doivent pouvoir résister au quotidien de la ruche, les câbles ne doivent pas pouvoir être coupés par les insectes et ne doivent pas perdre en sensibilité à cause de la cire pouvant les recouvrir.

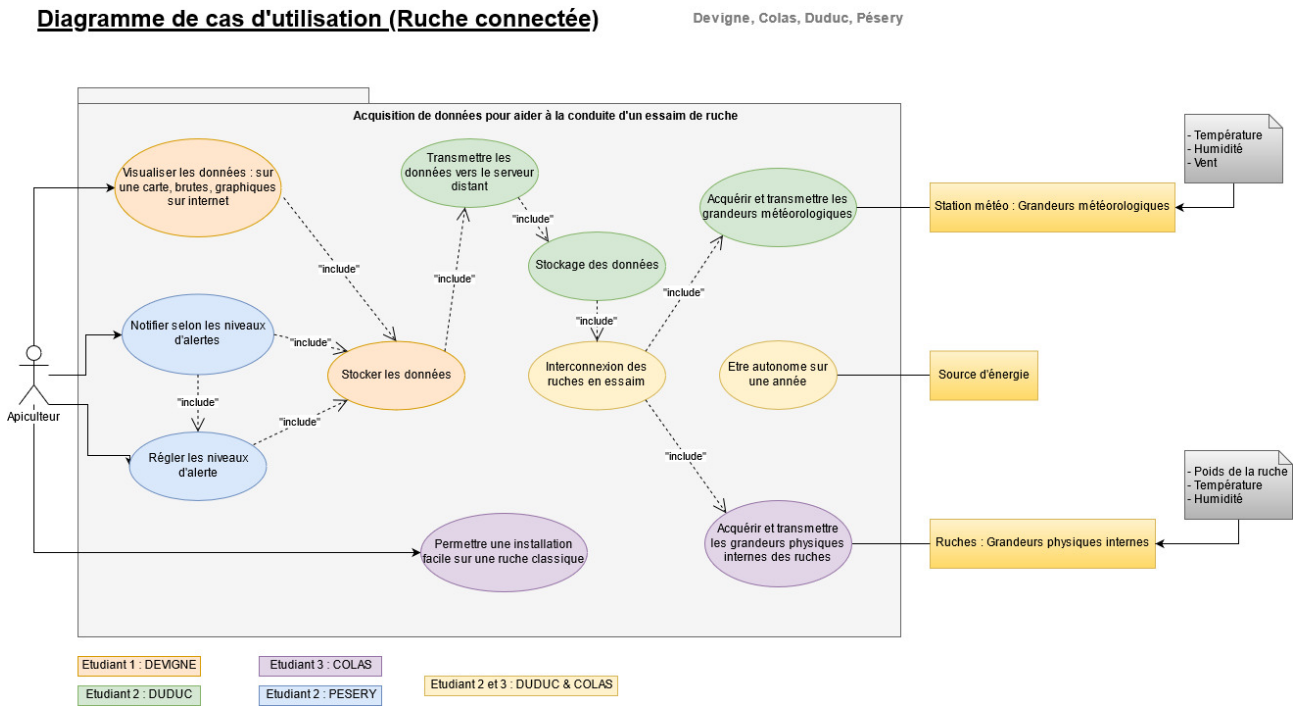
6. Budget

Le budget maximal du coût du système doit être inférieur à 400 € TTC hors abonnement 4G, pour un essaim de deux ruches, l'objectif étant d'améliorer le rendement de la production de miel.

ANALYSE FONCTIONNELLE

I. Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation ci-dessous est une représentation du comportement fonctionnel du système global.



Explication du diagramme :

Dans ce diagramme, nous présentons les cas d'usages mobilisant les différentes fonctions de la ruche connectée. Un code couleur permet de visualiser les différentes parties des étudiants ;

L'apiculteur peut visualiser les données via l'IHM, régler les niveaux d'alertes et doit être notifié selon les niveaux ; il doit également pouvoir installer facilement le système sur une ruche classique.

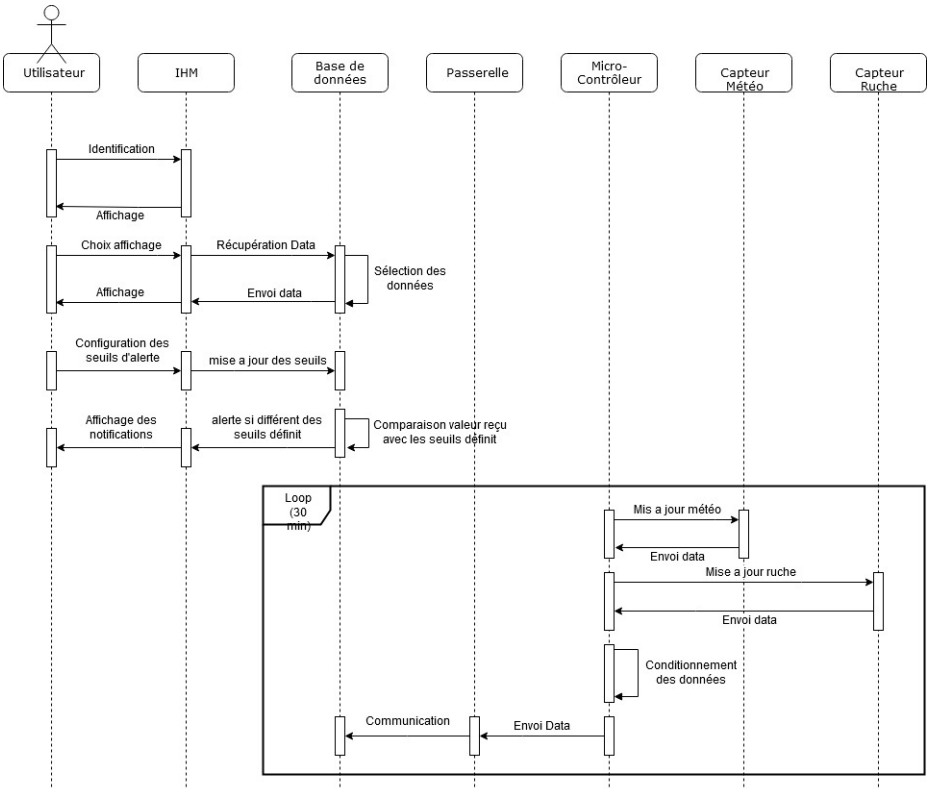
La visualisation des données et les réglages des niveaux d'alertes comprend le stockage des données dans une base de données.

La base de données permet l'interconnexion des ruches en essaim où l'on acquis et transmet les grandeurs physiques interne des ruches et météorologique de la station météo.

La source d'énergie permettant l'alimentation des appareils doit être autonome sur une année.

II. Diagramme de séquence

Le diagramme ci-dessous est une présentation graphique des interactions entre les acteurs et le système.

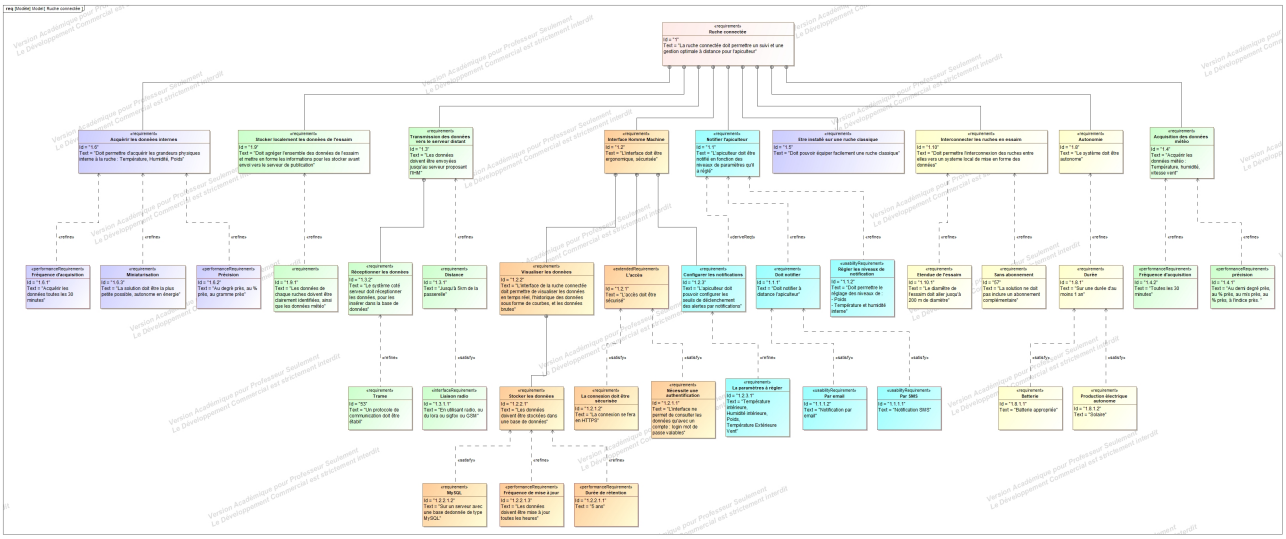


Explication du diagramme :

Sur ce diagramme, on voit que l'utilisateur interagit avec l'IHM (Interface Homme Humain) avec son identification pour commencer, puis avec la visualisation des données et la configuration des seuils d'alerte.
L'IHM quant à elle, interagit avec la base données pour la récupération des données, la mise a jour des seuils et l'envoie des notifications.
La base de données, envoie les données a l'IHM après les avoir reçu de la passerelle qui relais les données envoyé par le micro contrôleur.
Avant d'envoyer les données à la passerelle, le micro contrôleur récupère les données en provenance des capteurs de la station météo et de la ruche avant de les conditionner pour l'envoi.

III. Diagramme d'exigence

Le diagramme d'exigence ci-dessous précise les fonctions à réaliser ainsi que le niveau de performance à atteindre pour le projet.



Explication du diagramme :

1. Site Web et base de données (orange)

Interface ergonomique :

- Visualisation données en temps réel
- Visualisation de l'historique des données sous forme de courbe
- Visualisation des données brutes
- Les données sont stockées dans une base de donnée MySQL, pendant 5 ans, mise a jour des données toutes les heures

Sécurisée :

- Connexion en "https"
- Login, mot de passe valable uniquement

2. Acquérir et transmettre les données (vert, jaune et violet)

Acquisition des données météo :

- Acquérir les données toutes les 30 minutes.
- Acquérir la température extérieure : au demi degré près.
- Acquérir l'humidité extérieure : au % près.
- Acquérir la vitesse du vent : au m/s près.

Acquisition des données des ruches :

- Acquérir les données toutes les 30 minutes.
- Acquérir la température interne : au degré près.
- Acquérir l'humidité interne : au % près.
- Acquérir le poids de la ruche : au gramme près.

Le système doit être autonome en énergie :

- Sur une durée d'au moins un an
- Utiliser une batterie appropriée
- Si besoins, avoir une production d'électrique autonome

L'installation du système :

- Le système doit être facile à installer et à utiliser.
- Le système ne doit pas déranger les abeilles, ni l'utilisateur.
- Le système interne de la ruche doit être le plus petit possible.
- Le système doit résister aux intempéries.

Transmissions des données (passerelle) vers le serveur distant :

- Les données de chaque ruche doivent être clairement identifiées.
- Les données de la station météo doivent être clairement identifiées.
- Interconnexion des ruches entre elles vers un système local (passerelle).
- Utiliser une liaison radio pour communiquer avec les ruches et la station météo.
- Réception des données des ruches et de la station météo, les mettre en forme puis les stocker localement.
- Le diamètre de l'essaim doit aller au maximum jusqu'à 200 mètres de diamètre.
- La passerelle peut être installée à 5 km maximum de l'essaim.
- Les données doivent être envoyées jusqu'au serveur proposant l'IHM.
- La solution ne doit pas inclure un autre abonnement que l'abonnement 4G.
- Côté serveur, réception des données pour les insérer dans la base de données.
- Établir un protocole de communication pour la bonne réception des données côté base de données

3. Notifications - SMS & E-mail (bleu)

L'utilisateur doit pouvoir :

- Modifier les seuils d'alertes depuis le site internet ;
- Recevoir une confirmation de modification des seuils d'alerte ;
- Modifier la façon d'être notifié ;
- Être notifié par SMS si choisi ;
- Être notifié par E-mail si choisi;
- Être notifié par SMS et E-mail si choisi.

Le site internet doit pouvoir :

- Permettre à l'utilisateur la modification des seuils d'alerte concernant : le poids, la température intérieure et extérieure, l'humidité intérieure et extérieur, le vent et le pourcentage des batteries.
- Permettre de gérer chaque ruche indépendamment ;
- Confirmer à l'utilisateur la modification des seuils d'alerte ;
- Envoyer les seuils sélectionnés à la base de données.

Le système d'alerte (OS) doit pouvoir :

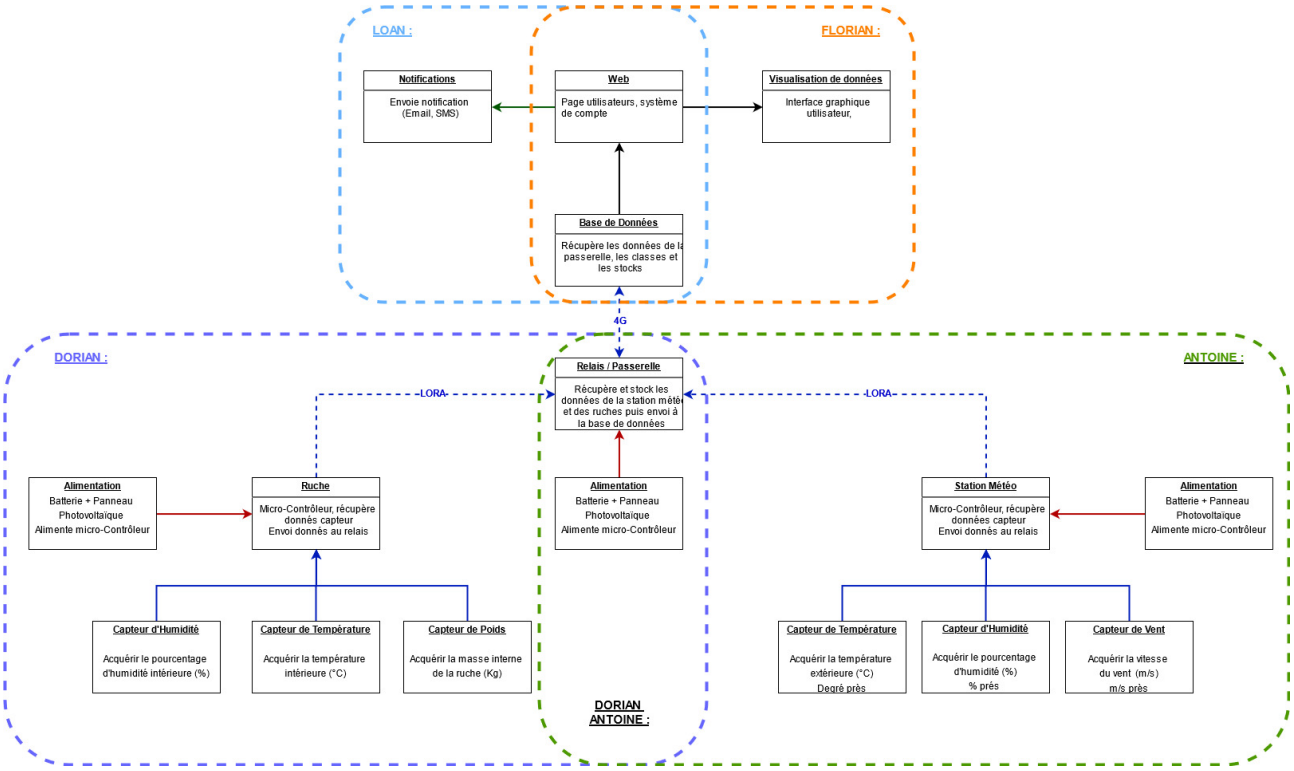
- Récupérer les dernières valeurs de la base de données ;
- Lire les seuils d'alertes définis ;
- Comparer les dernières valeurs récupérées aux seuils définis ;
- Confirmer l'envoi des notifications au système de notification ;

La base de donnée doit pouvoir :

- Enregistrer les seuils sélectionnés sur le site internet ;
- Récupérer et stocker les dernières valeurs reçus depuis la passerelle ;
- Communiquer les dernières valeurs à l'OS;
- Communiquer les valeurs des seuils définis à l'OS.

IV. Diagramme de définition de bloc

Le diagramme de définition des blocs ci-dessous décrit la structure du système.



Explication du diagramme :

Ce diagramme permet de schématiser la position et le rôle des différents étudiants et les interactions entre les éléments qui compose le système.

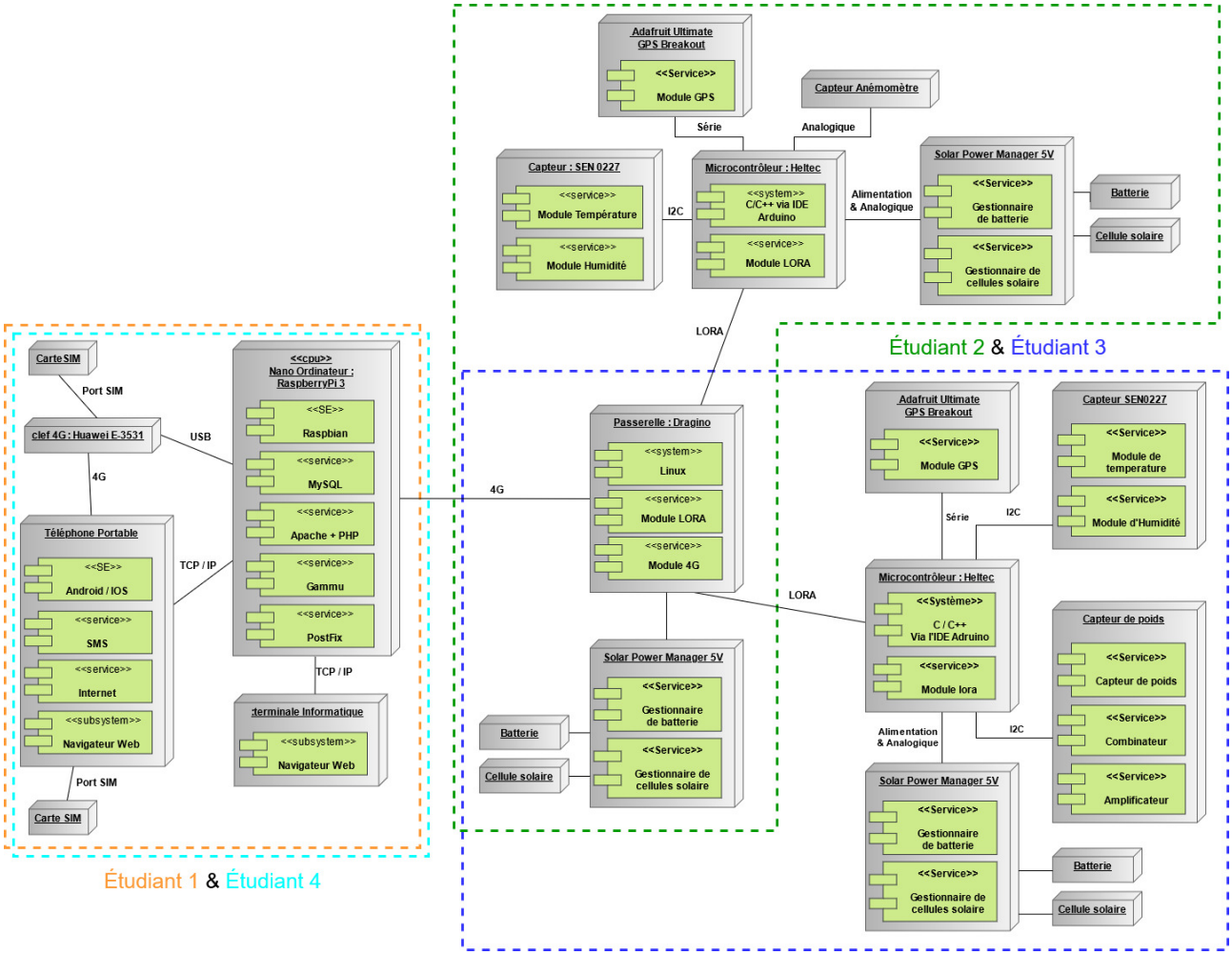
Florian (Étudiant 1) et Loan (Étudiant 4) interagissent tout les deux avec la partie Web et la base de données pour mettre en place l'IHM : Florian pour l'interface graphique du site Web et le système d'utilisateurs et Loan pour la partie notifications.

Dorian (Étudiant 3) et Antoine (Étudiant 2) interagissent tout les deux avec la passerelle afin de stocker les données de leurs capteurs : Dorian pour la ruche et Antoine pour la station météo.

Les étudiants 1 et 4 devront interagir avec les étudiants 2 et 3 afin de récupérer les données des capteurs.

V. Diagramme de déploiement

Le diagramme de déploiement ci-dessous est une représentation de l'utilisation de l'infrastructure physique du système et de la manière dont les composants et les services sont répartis.

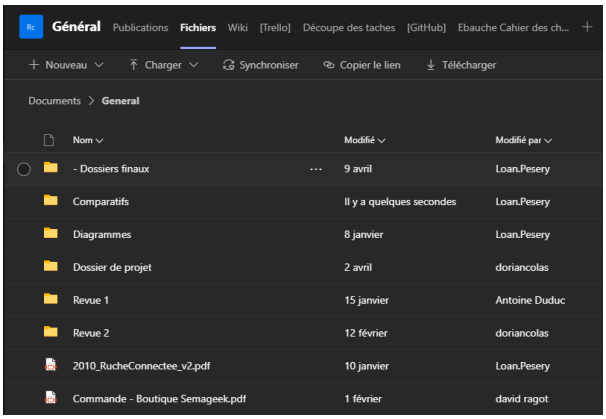


Dans ce diagramme figure les systèmes d'exploitations, les langages et les services installés dans les différents composants physique qui composent l'ensemble du projet ; ainsi que les types de liaisons qui connectes les différents éléments entre eux.
On remarque que les différentes parties des étudiants communiquent toutes avec la passerelle.

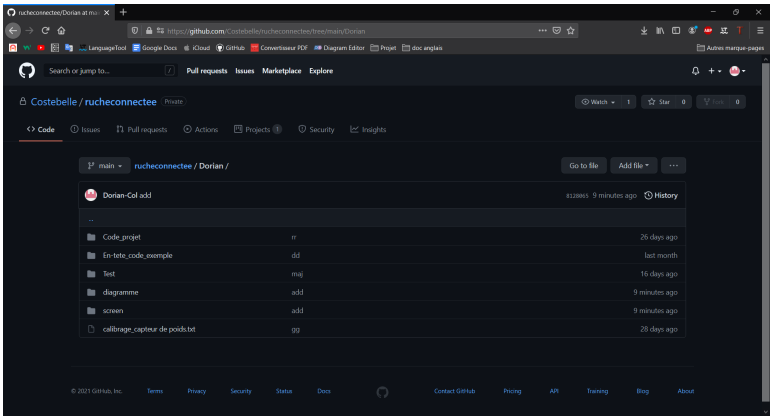
ORGANISATION DU TRAVAIL EN ÉQUIPE

1. Moyen collaboratif

Pour ce projet, nous avons mis en place plusieurs moyens de collaboration et de suivi des avancées du travail. Pour commencer nous avons utilisé "Microsoft Teams" pour stocker tout les fichiers lié au projet.
Nous avons aussi utilisé "Trello" pour réaliser le Kanban qui est un système de planification des taches. Sur notre Kanban un système de couleur permet à chaque



étudiant de différentier ses taches de celle des autres.
Pour ce qui est des scripts, nous avons utilisé "GitHub" qui est un fournisseur d'hébergement Internet pour le développement de logiciels et le contrôle de version à l'aide de Git.



2. Répartition des tâches

Étudiant 1 : Florian DEVIGNE **A FAIRE**
Données et web :
- Mise en oeuvre du serveur (Raspberry Pi)
- Installation de la base de donnée (MySQL)
- Mise en oeuvre d'un site web

Étudiant 2 : Antoine DUDUC
Acquérir et transmettre les données météo :
- Choix des composants :
  - Capteur de température et humidité (étanche)
  - Module GPS
  - Capteur anémomètre
  - Solution de communication
  - Microcontrôleur
- Essais des composants
- Mise en oeuvre des composants

Étudiant 3 : Dorian COLAS
Acquérir et transmettre les données de la ruche :
- Choix des composants :
  - Capteur de température (interne)
  - Capteur d'humidité (interne)
  - Capteur de poids
  - Solution de communication
  - Microcontrôleur
- Test des composants
- Choix d'une installation simple

Étudiant 4 : Loan PÉSERY
Gestion des notification, SMS, e-mail :
- Mise en oeuvre de la page internet "Notifications"
- Installation et configuration de la base de donnée
- Installation et configuration du système d'envoi des SMS
- Installation et configuration du système d'envoi des E-mail
- Réalisation de la mesure du champ électromagnétique lors de l'émission d'un SMS et du principe de transmission et de modulation.

3. Planning

! PLANNING FAIT PAR FLO !

Étudiant 1 :
Florian DEVIGNE

- Analyse du dossier
  05 au 07 Janvier 2021
- Réalisation des diagrammes
  08 au 19 Janvier 2021
- Réalisation du cahier des charges
  14 au 22 Janvier 2021
- Revue 1
  22 Janvier 2021
- Configuration du serveur
  24 au 29 Janvier 2021
- Réalisation des documents pour la revue
  1 au 14 Février 2021
- Configuration de la base de donnée
  1 au 5 Février 2021
- Revue 2
  19 Février 2021
- Recherche de solutions techniques pour le site
  8 au 12 Mars 2021
- Réalisation de la structure du site internet
  15 au 19 Mars 2021
- Réalisation du système de compte
  22 au 26 Mars 2021
- Réalisation de la page principale
  29 Mars - toujours pas fini
- Début de la réalisation du rapport
  06 Avril 2021

Étudiant 2 :
Antoine DUDUC

- Analyse du dossier
  05 au 07 Janvier 2021
- Réalisation des diagrammes & schémas
  08 au 15 Janvier 2021
- Réécriture du cahier des charges
  18 au 19 Janvier 2021
- Organisation Trello / Teams / GitHub
  21 Janvier 2021
- Revue 1
  22 Janvier 2021
- Réalisation comparatif & devis
  25 au 29 Janvier 2021
- Test des différents capteurs
  - Capteur humidité & température (DHT11)
    01 au 02 Février 2021
  - Capteur anémomètre (Adafruit 1733)
    04 au 05 Février 2021
  - Module solaire + batterie + Arduino
    08 au 09 Février 2021
  - Module GPS (Adafruit 746)
    11 au 12 Février 2021
- Réalisation du rapport de conception
  15 au 18 Février 2021
- Revue 2
  19 Février 2021
- Attente des composants
  22 Février au 12 Mars 2021
- Adaptation codes new composants
  15 au 25 Mars 2021
- Test communication LORA
  26 Mars 2021 - toujours pas fini

Étudiant 3 :
Dorian COLAS

Analyse du dossier
05 au 07 Janvier 2021

Réalisation des diagrammes
08 au 19 Janvier 2021

Réalisation du cahier des charges
14 au 22 Janvier 2021

Revue 1
22 Janvier 2021

Réalisation du devis
25 Janvier au 02 Février 2021

Début de test
04 au 05 Février 2021

- Capteur humidité & température (DHT11)
  04 au 05 Février 2021
- Module solaire + batterie + Arduino
  08 au 09 Février 2021
- Module GPS (Adafruit 746)
  11 au 12 Février 2021

Réalisation du rapport de conception
08 au 15 Février 2021

Revue 2
19 Février 2021

Recherche d'une solution d'installation
08 Mars 2021

Tests sur les composants commandé
09 au 23 Mars 2021

Tests de communication LORA
25 Mars au ... 2021

Début de la réalisation du rapport de projet
06 Avril 2021

Étudiant 4 :
Loan PÉSERY

Analyse du dossier
05 au 07 Janvier 2021

Réalisation des diagrammes
08 au 19 Janvier 2021

Réalisation du cahier des charges
14 au 22 Janvier 2021

Revue 1
22 Janvier 2021

Recherches & tests : envoi des e-mails
25 au 29 Janvier 2021

Recherches & tests : envoi des SMS
01 au 05 Février 2021

Réalisation page web : sliders, coches, ...
08 au 19 Février 2021

Revue 2
19 Février 2021

Création & tests base de données
08 au 16 Mars 2021

Mise en place comparaison toutes les min
18 au 25 Mars 2021

Mise en forme de la page web en CSS
26 Mars 2021

Écriture des phrases envoyées pour les alertes
29 Mars 2021

Ajout bouton remise à zéro & optimisations
30 Mars au 04 Avril 2021

Édition des documents finaux
06 au ~~ Avril 2021

ÉLÉMENTS NÉCESSAIRES À LA RECETTE DE LA MAQUETTE

Pour rappel, le projet a pour but de faciliter le suivi des ruches par les apiculteurs.

Pour valider le bon fonctionnement du système, on souhaite avoir un système robuste, sécurisé et autonome sur une année, qui récupère et transmet les grandeurs physiques internes et externes des ruches à la base de données.

Ces données doivent être ensuite mises en forme sur un site internet et consultable par l'apiculteur.
Ce dernier pourra alors choisir si il souhaite être notifié par SMS ou E-mail ou les deux, en fonction des seuils qu'il aura défini.

SOMMAIRE

INTRODUCTION..... 2

SEMAINIER..... 3

ANALYSE FONCTIONNELLE..... 5

I. Diagramme de cas d'utilisation..... 5

II. Diagramme de séquence..... 6

III. Diagramme de définition de bloc..... 7

IV. Diagramme d'exigence..... 8

V. Diagramme de déploiement..... 10

V. Diagramme de classes.....

COMPARAISONS ET SOLUTIONS SÉLECTIONNÉES..... 11

A. Transmissions des données.....

1. Solution de communication.....

2. Solution de passerelle.....

3. Solution d'alimentation.....

B. Acquisition des données.....

1. Choix du microcontrôleur.....

2. Choix du capteur de température et d'humidité.....

3. Choix du capteur anémomètre.....

4. Choix du module GPS.....

5. Solution d'alimentation.....

INTRODUCTION

Les abeilles subissent une mortalité accrue chaque année, elles sont particulièrement sensibles aux contraintes météorologiques.
Par conséquent mon rôle en tant qu'étudiant 2 dans ce projet de ruche connectée consiste à mettre en place une station météo autonome. Elle collecte les différentes données météorologiques du rucher. L'ensemble des données collectées par la station météo sont envoyé à une passerelle puis envoyée à un serveur. Par la suite les données pourront être visualisées à distance par l'apiculteur à partir d'un navigateur internet classique.

Mon travail est séparé en deux partie :

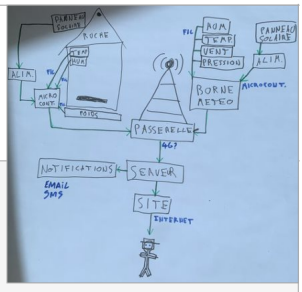
- La première est de mener à bien un système autonome en énergie pour réaliser l'acquisition des informations météo (température, humidité et la vitesse du vent). Et de transmettre les données vers la passerelle effectué dans la seconde partie.

- La seconde partie commune avec l'étudiant 3 Dorian COLAS, est de définir une solution de réception et de transmission des données en réseau avec l'essaim de ruche (une passerelle) vers la base de données du serveur de présentation et les enregistrer. De plus, ce système est encore une fois autonome en énergie. Puis réaliser le développement de cette solution.

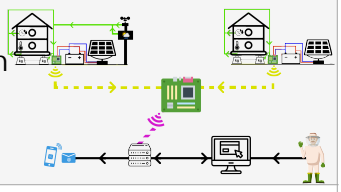
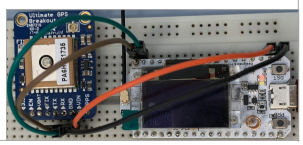
L'apiculteur pourra choisir l'emplacement de la station météo ainsi que de la passerelle en suivant les préconisations utilisateur.

SEMAINIER

Voici ci-dessous mon semainier ou plus couramment appelé carnet de bord, réalisée tout au long du projet :

Semaine 1 : du 05 au 08 Janvier 2021	✳ Analyse du projet ✳ Réalisation du schéma de principe global ✳ Réalisation du diagramme de bloc 
Semaine 2 : du 11 au 15 Janvier 2021	✳ Réalisation du diagramme de séquence ✳ Réalisation de mon schéma de principe ✳ Début de réécriture du cahier des charges ✳ Organisation du Kanban avec l'outil Trello ✳ Commencement diaporama Revue 1
Semaine 3 : du 18 au 22 Janvier 2021	✳ Réalisation du diagramme de définition de bloc, de séquence personnel ✳ Diaporama et cahier des charges de la Revue 1 terminé ✳ Revue 1

Semaine 4 : du 25 au 29 Janvier 2021	✳ Réalisation des comparatifs :
	  ✳ De communications
	  ✳ Des microcontrôleurs
	  ✳ Des différents capteur (température / humidité)
	✳ Commencement du devis

Semaine 5 : du 01 au 05 Février 2021	✳ Rendu du devis V.1 ✳ Test du capteur DHT11 (humidité / température) avec un Arduino ✳ Test du capteur anémomètre ✳ Test du solar power manager + batterie + cellule solaire avec un Arduino
Semaine 6 : du 08 au 12 Février 2021	✳ Réalisation d'un code pour récupérer le pourcentage de batterie ✳ Rendu du devis ✳ Commencement du rapport de conception ✳ Commencement diaporama Revue 2 ✳ Réalisation du schéma de câblage 
Semaine 7 : du 15 au 19 Février 2021	✳ Test module GPS ✳ Diaporama et rapport de conception terminé ✳ Revue 2 

Semaine 8 : du 08 au 12 Mars 2021	✳ Mise en forme des données GPS ✳ Attente des composants
Semaine 9 : du 15 au 19 Mars 2021	✳ Réception des différents composants ✳ Test du capteur d'humidité et température ✳ Organisation des différents code avec l'outil GitHub ✳ Prise en main de l'ESP32
Semaine 10 : du 22 au 26 Mars 2021	✳ Adapter le code du GPS pour l'ESP32 ✳ Adapter le code du capteur anémomètre pour l'ESP32 ✳ Adapter le code du capteur d'humidité et température pour l'ESP32 ✳ Commencer a s'informer sur la configuration de la communication LORA
Semaine 11 : du 29 Mars au 02 Avril 2021	✳ Test communication LORA
Semaine 12 : du 05 au 09 Avril 2021	✳ Confiné : Rédaction du dossier de projet : ✳ Commun

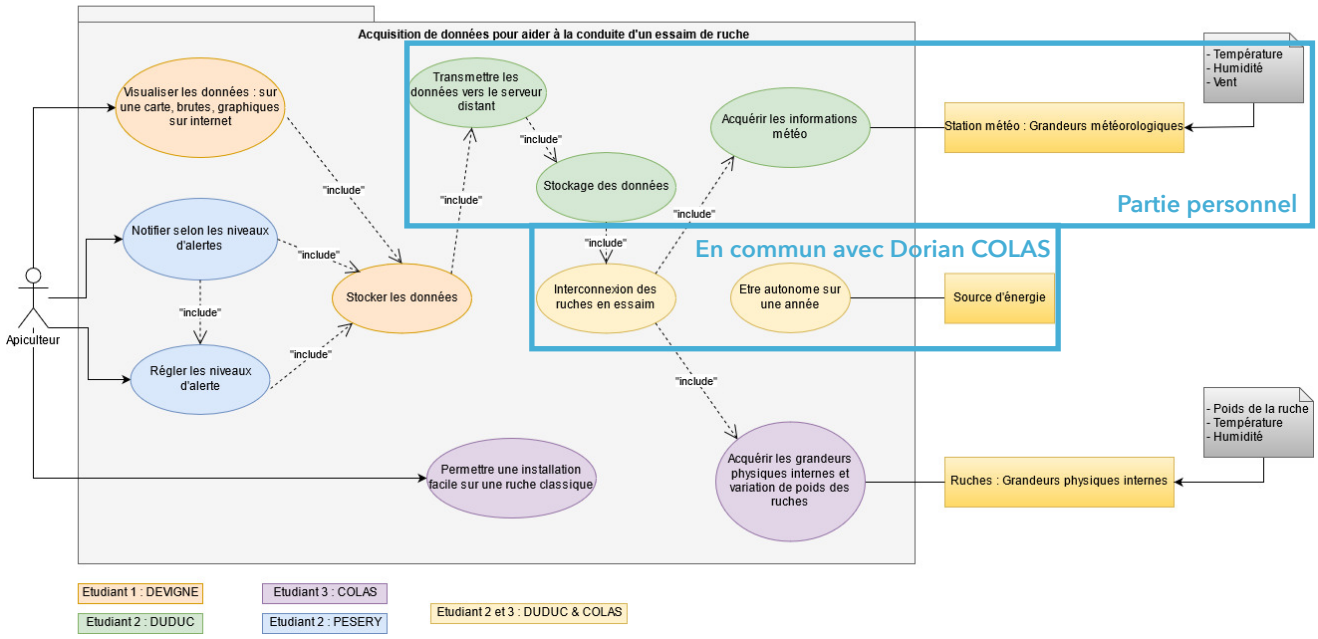
Semaine 13 : du 12 au 16 Avril 2021	✳ Confiné : Rédaction du dossier de projet :
	  ✳ Personnel
	  ✳ Commun

Semaine 14 : du 19 au 23 Avril 2021	
Semaine 15 : du 26 au 30 Avril 2021	
Semaine 16 : du 03 au 07 Mai 2021	

ANALYSE FONCTIONNELLE

I. Diagramme de cas d'utilisation

Voici ci-dessous une représentation du **diagramme de cas d'utilisation** général. Ce diagramme représente les différents comportements entre le système et l'apiculteur. Un code couleur est créer pour différencier les parties des étudiants.

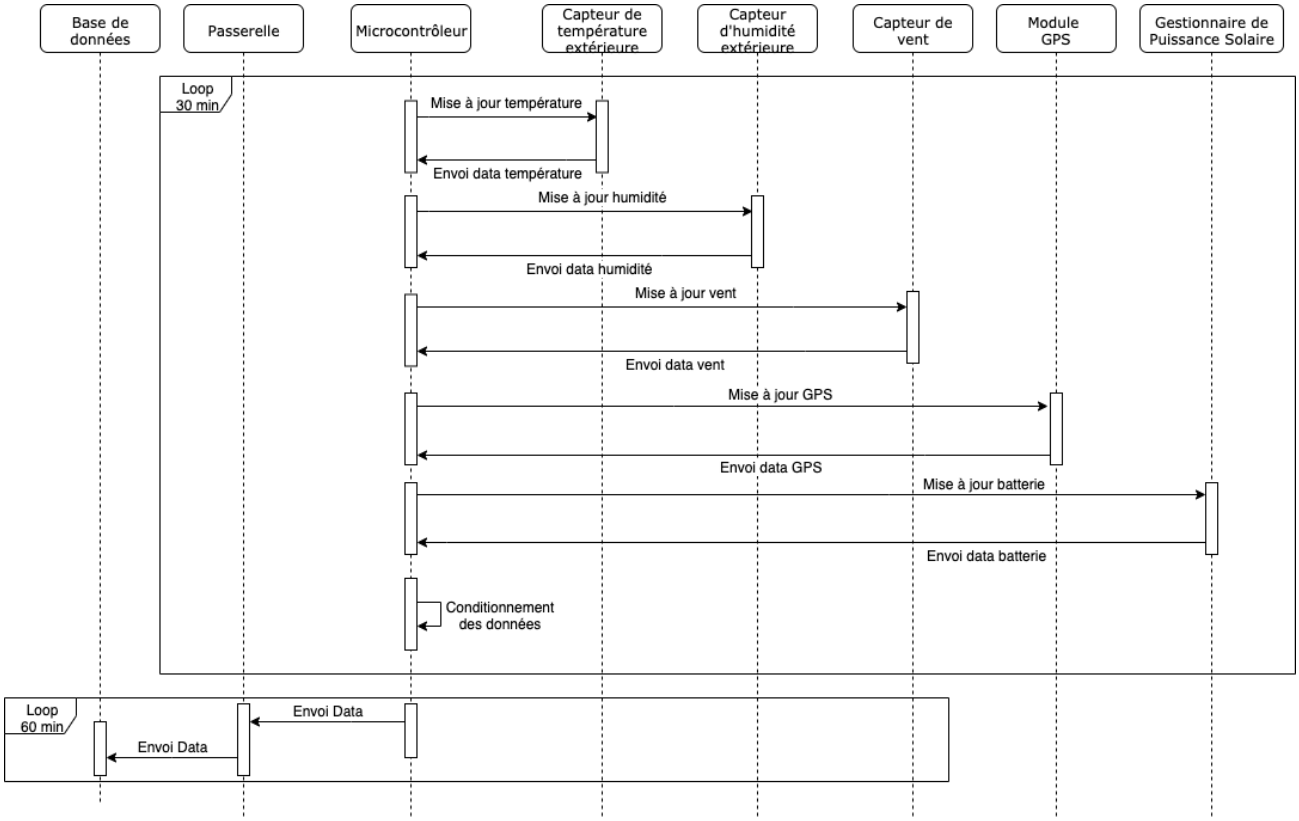


Dans un premier temps, nous nous intéressons a ma partie personnelle, de couleurs vertes. On peut voir que je m'occupe d'obtenir les différentes information météorologique, de stocker localement les informations des ruches et de la station météo avant de les transmettre à la base de données qui est l'endroit ou toute les valeurs sont stockées pour les autres étudiants.

Dans un second temps, on étudie la partie commune avec Dorian COLAS de couleurs jaunes. On peut voir que nous interagissons principalement avec la passerelle qui a pour bute d'interconnecter la station météo, les ruches en essaim à la base de données. Mais on s'occupe aussi de rendre le système totalement autonome sur au moins un an.

II. Diagramme de séquence

Voici ci-dessous une représentation de mon **diagramme de séquence** personnel. Ce diagramme représente les messages transitant entre le système et ses acteurs aux différents instants dans un ordre chronologique.



Voici les différentes ligne de vie de mon diagramme :

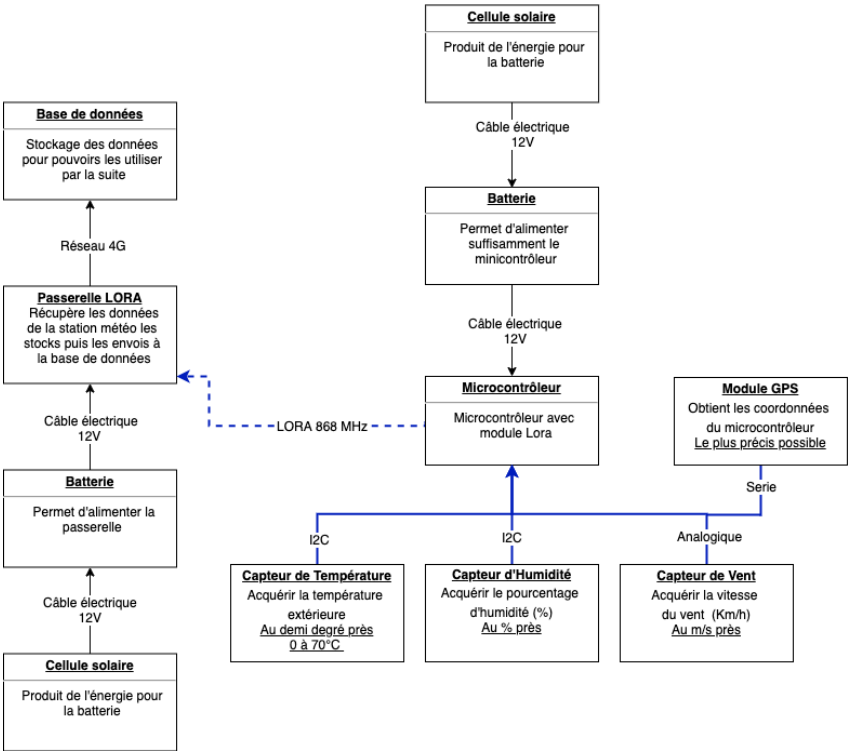
- **Base de données :** ça représente l'emplacement ou les données des capteurs sont stockées et structurées.
- **Passerelle :** c'est l'équipement qui permet de relier le microcontrôleur à la base de données en utilisant deux réseaux différent.
- **Microcontrôleur :** c'est l'équipement qui intègre un circuit intégré pour pouvoir connecter les différents capteurs et interagir avec la passerelle.
- **Capteur de température extérieure :** dispositif qui permet de récupérer la température en degrés Celsius.
- **Capteur d'humidité extérieure :** dispositif qui permet de récupérer le pourcentage d'humidité.
- **Capteur de vent :** dispositif qui permet de récupérer la vitesse du vent en Km/H.
- **Module GPS :** dispositif qui permet de connaitre les coordonné GPS du microcontrôleur
- **Gestionnaire de puissance solaire :** dispositif qui permet de gérer la puissance solaire, la recharge de la batterie et l'alimentation du microcontrôleur.

Le micro contrôleur récupère les données en provenance des capteurs d'humidité, de température et de vent ainsi que les coordonnées du module GPS. Les données conditionnées par le microcontrôleur sont envoyées à la passerelle, cette boucle est réalisé toute les 30 minutes.

Puis la passerelle les envoies à la base de données. Et enfin, la base de données réceptionne et structure les données reçu, ce cycle est réalisé toute les heures.

III. Diagramme de définition de bloc

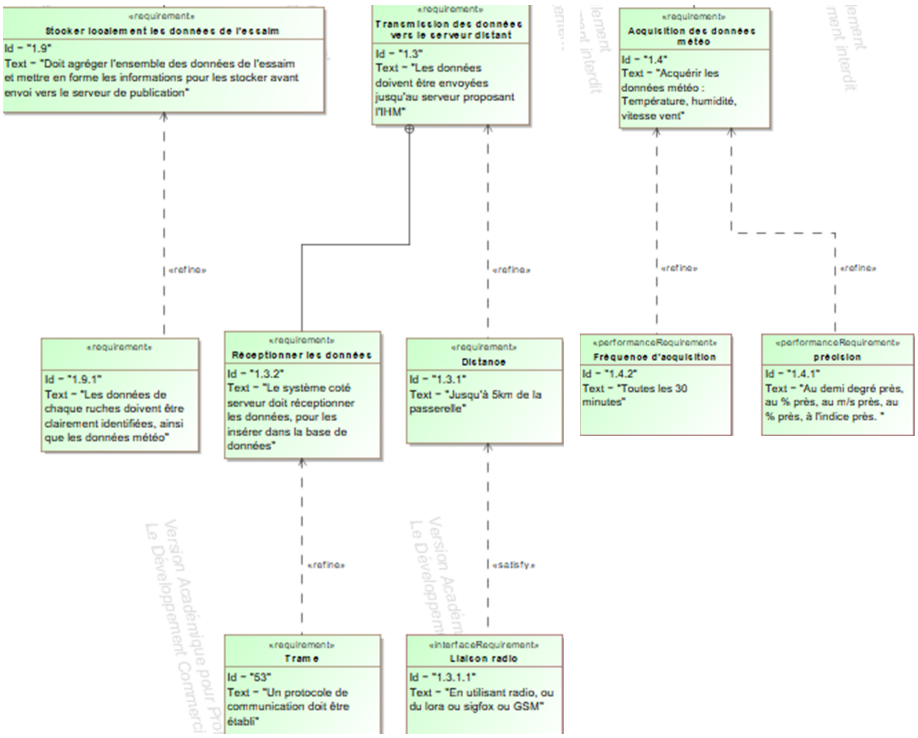
Voici ci-dessous une représentation de mon **diagramme de définition de bloc** personnel. Ce diagramme définie les différents éléments et leurs interactions à l'intérieur du système.



Dans ce diagramme figure les différents blocs de composants relié à la passerelle ou au microcontrôleur.
Le système de ma partie est structuré avec les différents composant relié à la ruche via le microcontrôleur comme les capteurs suivants : humidité, température, poids et GPS ainsi que le panneau solaire et la batterie pour l'autonomie.
Ce microcontrôleur est relié par liaison LORA 868 MHz (norme européenne) à la passerelle qui est alimenté par une batterie et un panneau solaire.

IV. Diagramme d'exigences

En premier lieux, voici ci-dessous en vert une représentation de mon **diagramme d'exigence personnel**. Le diagramme définie les contraintes que doit respecter le système.



Stocker localement les données de l'essaim :

- Réception des données des ruches et de la station météo.
- Mettre en forme les données puis les stocker localement.
- Les données de chaque ruche doivent être clairement identifiées.
- Les données de la station météo doivent être clairement identifiées.
- Envoi des données vers le serveur de publication.

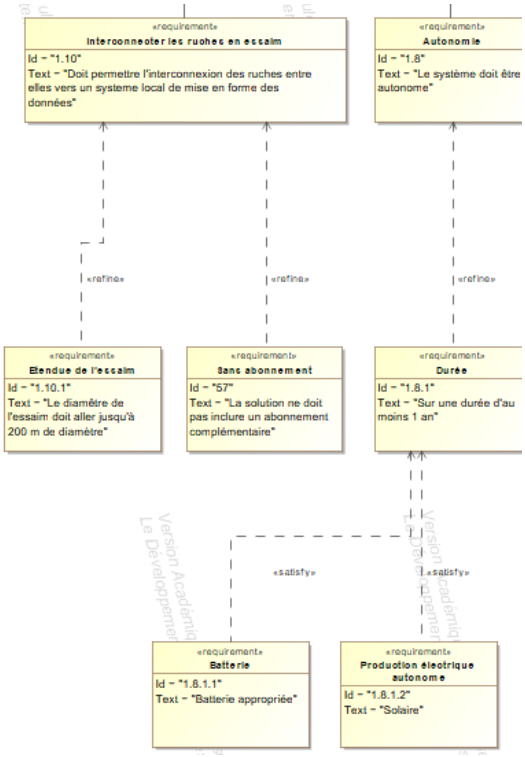
Transmissions des données (passerelle) vers le serveur distant :

- Les données doivent être envoyées jusqu'au serveur proposant l'IHM.
- Côté serveur, il doit réceptionner les données pour les insérer dans la base de données.
- Établir un protocole de communication pour la bonne réception des données côté base de données.
- La passerelle peut être installée à 5 kilomètres maximum de l'essaim.
- La solution de communication doit utiliser une liaison radio ou GSM.

Acquisition des données météo :

- Acquérir les données toutes les 30 minutes.
- Acquérir la température extérieure au demi degré près.
- Acquérir l'humidité extérieure au % près.
- Acquérir la vitesse du vent au m/s près.

En second lieux, voici ci-dessous en jaune une représentation du **diagramme d'exigence commun** avec l'étudiant 3 Dorian COLAS.



Interconnexion des ruches en essaim :

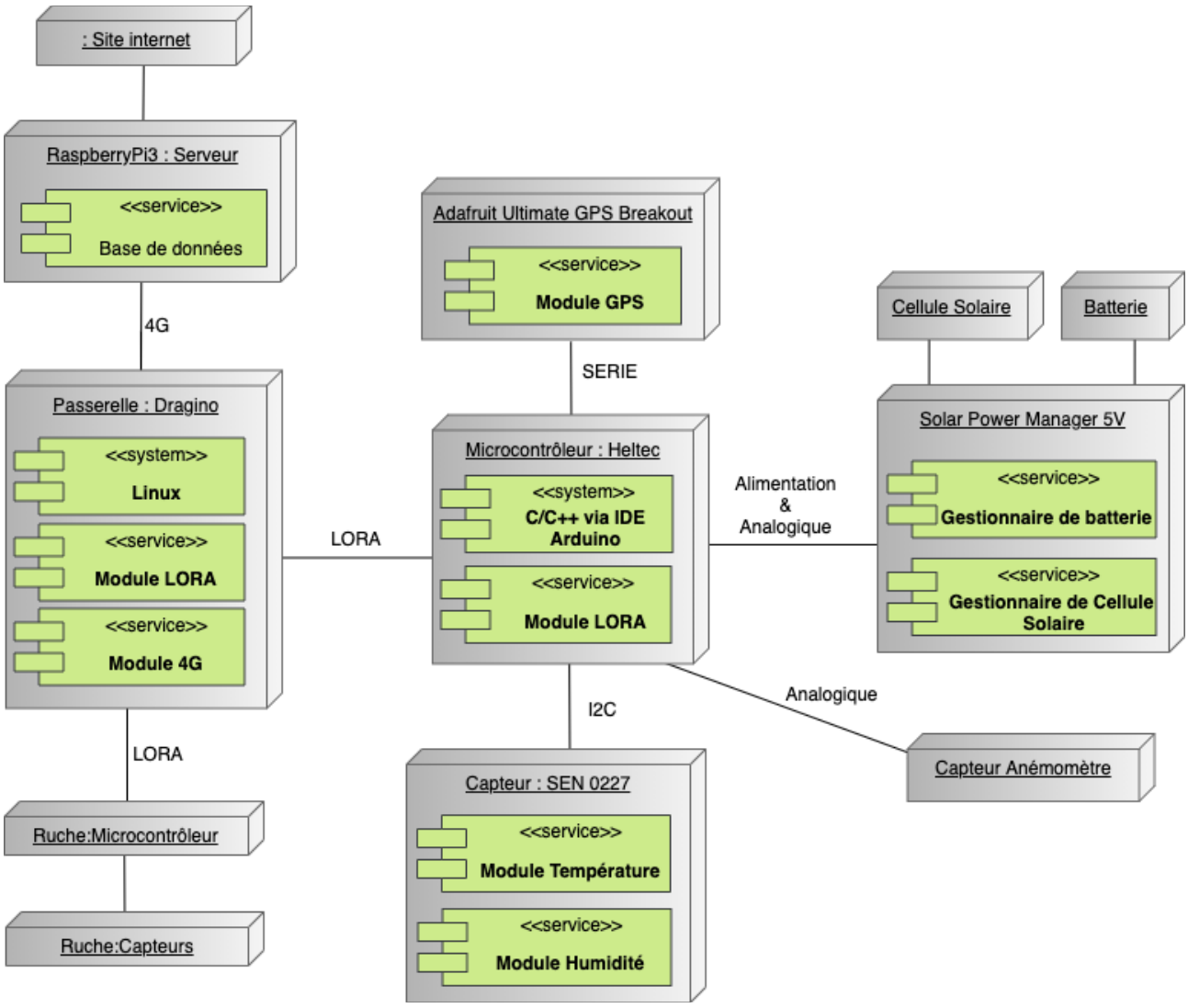
- Le diamètre de l'essaim doit être de 200 mètres au maximum.
- La solution ne doit pas inclure un autre abonnement que l'abonnement 4G.
- Interconnexion des ruches entre elles vers un système local (passerelle).

Le système doit être autonome en énergie :

- Sur une durée d'au moins un an.
- Utiliser une batterie appropriée.
- Avoir une production électrique autonome, par exemple solaire.

V. Diagramme de déploiement

Voici ci-dessous une représentation de mon **diagramme de déploiement** personnel. Ce diagramme définie la disposition des différents matériels et la nature des connexions de communication.



A MODIFIER

Dans ce diagramme figure les différents service, sytème et connexion des matériels. Nous pouvons apercevoir tout les équipements que j'utilise pour ma partie du système.

Il y a aussi les autres matériels nécessaire au bon fonctionnement du système comme le site internet, les capteurs et le microcontrôleur de la ruches.

VI. Diagramme des classes

Voici ci-dessous une représentation de mon **diagramme des classes**. Ce diagramme définie

COMPARAISONS ET SOLUTIONS SÉLECTIONNÉES

A. Transmission des données

Avant toute chose, pour transmettre les données de la station météo à la base de données nous avons faits le choix avec Dorian COLAS d'utiliser l'intermédiaire d'une passerelle.

1. Solution de communication :

Pour sélectionner un protocole de communication du microcontrôleur à la passerelle ainsi que de la passerelle à la base de données nous avons élaboré un tableau comparant les différentes solutions en respectant nos contraintes.

Comparatif des protocoles de communication :

	WIFI	BLUETOOTH	LORA	ZIGBEE	SIGFOX	Z-WAVE	CELLULAIRE	FILAIRE
PORTÉES	+	+	+ + +	+	+ + + +	+	+ + + + +	+ + + + +
ECO ENERGIES	+	+ +	+ + + +	+ + + +	+ + + +	+ + + +	+ +	+ + +
DEBIT	+ + + +	+ + + +	+ +	+ + +	+	+ +	+ + + +	+ + + +
COUT	+ + +	+ + + +	+ +	+ +	+ +	+ +	+ + +	+ + + + +
INTÉGRATION	+ + + +	+ + + +	+ + +	+ + +	+ + +	+ + +	+ + + +	+ + + +
ABONNEMENT	NON	NON	NON	NON	OUI	NON	OUI	NON

Dans un premier temps, nous avons étudié la solution de la station météo à la passerelle.
Pour choisir le protocole de communication on avait différentes contraintes :

- La solution doit permettre d'être disposée dans un rayon de 5 kilomètres maximum dans un milieu urbain.
- Elle doit être la moins énergivore possible.
- Elle doit être sans abonnement.
- Elle doit être facile à intégrer au microcontrôleur.

À la suite de notre comparatif, avec Dorian on a fait le choix d'opter pour la technologie **LORA** afin d'assurer correctement la communication de la station météo à la passerelle. Cette technologie complète parfaitement nos contraintes.

Ce protocole de communication est idéal, il n'y a pas besoin d'abonnement, sa portée est d'environ 5 kilomètres, sa consommation est moindre et il est très simple d'intégrer la solution à un microcontrôleur, pour un Arduino par exemple, il suffit d'un simple Shields ou alors il existe des ESP32 avec un module LORA intégré.

Dans un second temps, nous avons étudié la solution de la passerelle au serveur qui héberge la base de données.
Pour choisir le protocole de communication on avait différentes contraintes :

- La solution doit être en permanence connectée a internet.
- Elle doit couvrir une grande partie de la France.
- Elle doit être la moins énergivore possible.
- Elle peut avoir un abonnement.

À la suite de notre comparatif, avec Dorian on a fait le choix d'opter pour la technologie **CELLULAIRE** afin d'assurer correctement la communication de la passerelle au serveur. Cette technologie complète parfaitement nos contraintes.

Ce protocole de communication est très bien, la couverture du réseau est très étendue, elle fait presque toute la France et avec un abonnement 4G par exemple la passerelle pourra être constamment connectée a internet. Néanmoins, cette solution consomme beaucoup d'énergie contrairement à la technologie LORA.

2. Choix de la passerelle :

Pour sélectionner la passerelle, nous avons pris en compte nos choix précédents. Une passerelle associant liaison LORA et liaison CELLULAIRE (4G). Nous avons donc élaboré un tableau comparant les différentes passerelles.

Comparatif des passerelles LORA :

	Passerelle LORA Dragino LG01-P 868 MHz	Passerelle LORA Dragino LG01-N 868 MHz - EC25-Ew
ECO ENERGIES	+	+
COUT	+ + +	+ +
CONNEXION CELLULAIRE	NON	OUI
PORTEE MAX	+ + + +	+ + + +

À la suite de notre comparatif, nous avons fait le choix d'opter pour la passerelle **LORA Dragino LG01-N 868MHz - EC25 - Ew** afin d'assurer correctement les communications. Néanmoins la passerelle ne pourra pas être autonome en énergie dû à notre budget.

Cette passerelle répond à nos attentes, sa portée est d'environ 5 kilomètres dans un environnement urbain, il y a possibilité de rajouter une carte SIM via un connecteur de carte pour avoir un abonnement cellulaire et la bande de fréquences est de 868 MHz ce qui correspond à la norme européenne. Néanmoins, elle consomme trop d'énergie pour subvenir à ses besoins en autonomie dû à notre budget.

3. Solution d'alimentation :

Pour sélectionner la solution d'alimentation, nous avons commencé par calculer la puissance de la passerelle pour une heure.
Puissance = Ampère x Volt = 0,5A x 12V = 6W. La passerelle consomme 6 Watts pour une heure. La nuit dure environ 12 heures en moyenne, donc 6 x 12 = 72 alors la passerelle consomme environ 72 Watts pour 12 heures. Il faut donc une batterie d'environ 100Watts, un panneau solaire qui pouvant la recharger ainsi qu'un régulateur de puissance et une journée ensoleillée pour la recharger correctement.

Après de nombreuses recherches, malheureusement par faute de budget nous pouvons pas subvenir aux besoins énergétiques de la passerelle. Malgré cela, nous avons quand même élaboré une solution de secours pour pouvoir montrer nos compétences. Notre professeur nous a fourni un régulateur de puissance solaire, une batterie ainsi qu'un panneau solaire qu'il avait en stock au lycée.

Le régulateur de puissance PWM permet la recharge de batterie au plomb via un panneau solaire jusqu'à 10 A en 12 V. Le régulateur utilise également diverses fonctions de protection pour la batterie, le panneau solaire et la sortie, afin d'éviter les surcharges et les décharges ce qui améliore considérablement la sécurité.

Le panneau solaire de 10 Watts délivre une tension de 17,6 V ainsi qu'une intensité de 610 mA. Il est fabriqué en silicium polycristallin et encadré d'aluminium ce qui le rend résistant aux intempéries et à la rouille. Il est idéal pour le régulateur vu au-dessus.

La batterie au plomb rechargeable, à une forte capacité, étanches et sans entretien. Cet accumulateur fournit une tension de 12 V et à une capacité de 7 Ah.

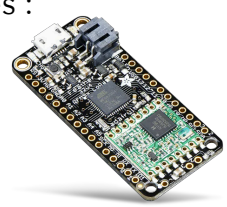
B. Acquisition des données

Avant toute chose, pour réaliser l'acquisition des données de la station météo j'ai créé différents tableaux comparant les composants nécessaires.

1. Choix du microcontrôleur :

Pour choisir quel microcontrôleur commandé, j'ai réalisé un tableau les comparants en fonction de mes besoins :

Pour choisir le microcontrôleur j'ai dû respecter ses différentes contraintes :



- Le microcontrôleur doit être le plus petit possible.
- Il doit être le moins énergivore possible.
- Il doit être disponible sur les sites prescrit par nos professeurs.
- Il doit être le moins cher possible.
- Il doit avoir un module LORA intégré ou la possibilité accueillir un Shields LORA.

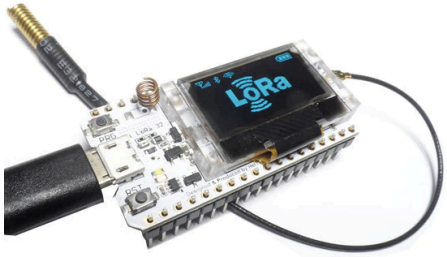
Comparatif des microcontrôleur :

	Arduino Uno	Raspberry Pi 3 B+	ADAFRUIT FEATHER 32u4 RFM95	LILYGO - TTGO T BEAM ESP32 LoRa 868MHz
ECO ENERGIES	+ + + +	+ +	+ + + +	+ + + +
COUT	+ + + +	+ +	+ +	+ + + +
PIN analogique	6	?	10	?
PIN digital	14	?	20	?
LORA intégré	NON	NON	OUI	OUI
Dimensions	74x53x15 mm	86x54x17 mm	51x23x8 mm	100x33x20 mm
GPS intégré	NON	NON	NON	OUI
Disponible sur les sites prescrit	OUI	OUI	OUI	NON
Mémoire Flash	32 kB	Besoin d'une carte SD	32 kB	4 Mo

À la suite de ce comparatif, j'ai fait le choix d'opter pour le microcontrôleur **ADAFRUIT FEATHER 32u4 RFM95** car il répond parfaitement nos contraintes.

Cette passerelle est idéale, le module LORA est déjà intégré, sa consommation en énergie est moindre, il est disponible sur les sites prescrit et il est de petite taille.

Mais malheureusement, nous avons eu un problème avec le vendeur ainsi que la livraison, de ce fait on a changé de microcontrôleur en urgence. Nous avons donc commandé le « **HELTEC WIFI LORA 32** » qui est similaire sauf en plus il y a un écran LCD, un module WIFI, un module Bluetooth et une antenne.



2. Choix du capteur de température et d'humidité :

Pour sélectionner le capteur de température et d'humidité que je vais utiliser pour la station météo j'ai effectué un comparatif. Mais pour me faciliter la tâche j'ai préféré comparer directement des capteurs embarquant la température et l'humidité que choisir deux capteurs différents.
Pour choisir le capteur de température et d'humidité j'ai dû respecter ses différentes contraintes :

- Le capteur de température et d'humidité doit être étanche.
- Il doit être le moins énergivore possible.
- Il doit être précis aux demi-degrés près et au pourcentage près.
- Il doit être le moins cher possible.

Comparatif des capteurs de température et humidité :

	SEN0227	DHT 22	SHT10	DHT 11
ECO ENERGIES	+ + + +	+ + + +	+ + + +	+ + + +
COUT	+ + +	+ + + +	+ +	+ + + +
ETANCHE	OUI	NON	NON	NON
PLAGE DE MESURE	+ + + + +	+ + + + +	+ + + +	+ + +
PRECISION	+ + + +	+ + + +	+ + +	+ + +
DIMENSIONS	73x17 mm	25x15x9 mm	75x14 mm	23x17x9 mm

À la suite de mon comparatif, j'ai fais le choix d'opter pour le capteur **SEN0227** afin d'assurer correctement l'acquisition de la température et de l'humidité au sein de la station météo. Ce capteur remplie parfaitement nos contraintes.

Ce capteur de température et d'humidité est idéal pour moi, sa consommation d'énergie est très faible, un point important il est étanche a un pris abordable, et enfin il a une excellente précision.

3. Choix du capteur anémomètre (vitesse du vent) :

Pour choisir le capteur anémomètre, je n'ai pas eu besoin de faire de comparatif car le lycée en avait déjà un (**Anémomètre Adafruit 1733**) pour l'année précédente. Il m'a suffi de vérifier son fonctionnement et s'il correspondait à mes besoins.

Pour vérifier le capteur anémomètre j'ai étudié ses différentes contraintes :

- L'anémomètre doit être le moins énergivore possible.
- Il doit être précis au mètre par seconde près.
- Il doit être étanche.

À la suite de mes vérifications, j'ai fait le choix de garder le capteur **Anémomètre Adafruit 1733** afin de rester dans le budget total du projet. L'anémomètre est précis et peu énergivore mais malheureusement pas étanche.

4. Choix du module GPS :

Pour choisir le module GPS (Global Positioning System), je n'ai pas eu besoin de faire de comparatif, le lycée en avait déjà un (**Adafruit Ultimate GPS Breakout 746**) prévu pour l'année dernière. Il m'a suffi de vérifier son bon fonctionnement et s'il correspondait à mes besoins.

Pour vérifier le module GPS j'ai étudié ses différentes contraintes :

- Le module GPS doit être le moins énergivore possible.
- Il doit être le plus précis possible.

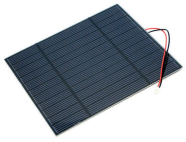
À la suite de mes vérifications, j'ai fait le choix de garder le module GPS **Adafruit Ultimate GPS Breakout 746** afin de rester dans le budget total du projet. Le module GPS complète parfaitement mes contraintes.

5. Solution d'alimentation :

Avant toute chose, pour pouvoir choisir les différents composants qui alimenteront la station météo il faut que je sache quelle est la consommation total.
Pour le microcontrôleur (il peut être alimenté en 3.7V) : le LORA consomme 130 mA +

Au vu de cette consommation, mon professeur avait en stock au lycée un gestionnaire de puissance solaire, une cellule solaire ainsi qu'un accumulateur qui alimentera parfaitement le microcontrôleur HELTEC via cordon USB vers micro USB.

Un gestionnaire de puissance solaire de 5V. C'est un module de gestion d'énergie solaire qui peut fournir un courant de charge allant jusqu'à 900 mA à une batterie de 3,7 V avec un chargeur USB ou un panneau solaire. Le module utilise également diverses fonctions de protection pour la batterie, le panneau solaire et la sortie, ce qui améliore considérablement la sécurité.



Une cellule solaire de 3 Watts délivrant une tension de 5,5 V jusqu'à 540 mA. Une fine couche de résine lui permet d'être utilisé en extérieur.

Un accumulateur lithium-polymère rechargeable qui permet d'ajouter une alimentation autonome, stable et durable à notre station météo. Cet accu fourni une tension de 3,7V et a une capacité de 1800 mAh.

RÉALISATIONS ET CONFIGURATIONS

A. Transmission des données

6. STTT

BTS Système Numérique
Option : Informatique et réseaux
Projet technique : U62

Lycée Costebelle
83400 HYÈRES

Dorian COLAS

PROJET PERSONNEL :
RUCHE CONNECTÉE

Membres de l'équipe projet :
- Florian DEVIGNE
- Antoine DUDUC
- Dorian COLAS
- Loan PÉSERY

Session 2021

SOMMAIRE

Sommaire	2
Introduction	3
Semainier	4
Analyse fonctionnelle	5
1- Diagramme de cas d'utilisation	5
2- Diagramme de séquence	6
3- Diagramme d'exigence	7
4- Diagramme de bloc	8
5- Diagramme de déploiement	9
Réalisation	10

Introduction

Ce projet doit permettre de surveiller la production de miel et le comportement des abeilles. Cela consiste donc à équiper chaque ruche d'abeilles classique facilement et de manière à ne pas gêner les abeilles et l'apiculteur en y ajoutant des capteurs et des équipements miniatures pour obtenir les différentes informations interne ainsi que les données atmosphérique alentour.

L'ensemble des données collectées seront envoyé vers la base de données et disponibles sur internet. Elles pourront être visualisées à distance par l'apiculteur indifféremment à partir d'un navigateur internet classique ou à l'aide d'un smartphone.

D'autre part en cas d'alerte critique, et de manière journalière sous forme de récapitulatif de mesures, une notification sera envoyée à l'apiculteur sur son smartphone récapitulant l'état de chacune des ruches.

Pour ma part, j'ai eu la responsabilité d'acquérir et transmettre les grandeurs physiques interne à la ruche. Mon travail était séparé en deux parties, une partie personnelle et une partie commune à l'étudiant Antoine DUDUC.

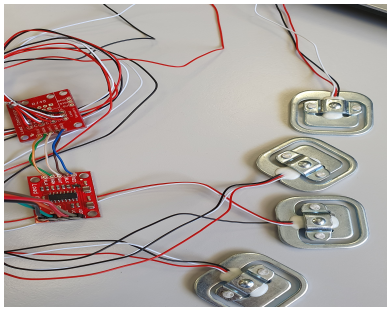
Le travail à réaliser était le suivant :

- Pour la partie personnel : Je devais réaliser l'acquisition des grandeurs physique de la température, de l'humidité et du poids de la ruche. Je devais aussi définir une solution d'installation du système simple et adaptable à toutes les ruches.

- Pour la partie commune : Nous devions définir une solution de transmission des données vers la base de données, mais nous devions aussi rendre ce système autonome en énergie.

SEMAINIER

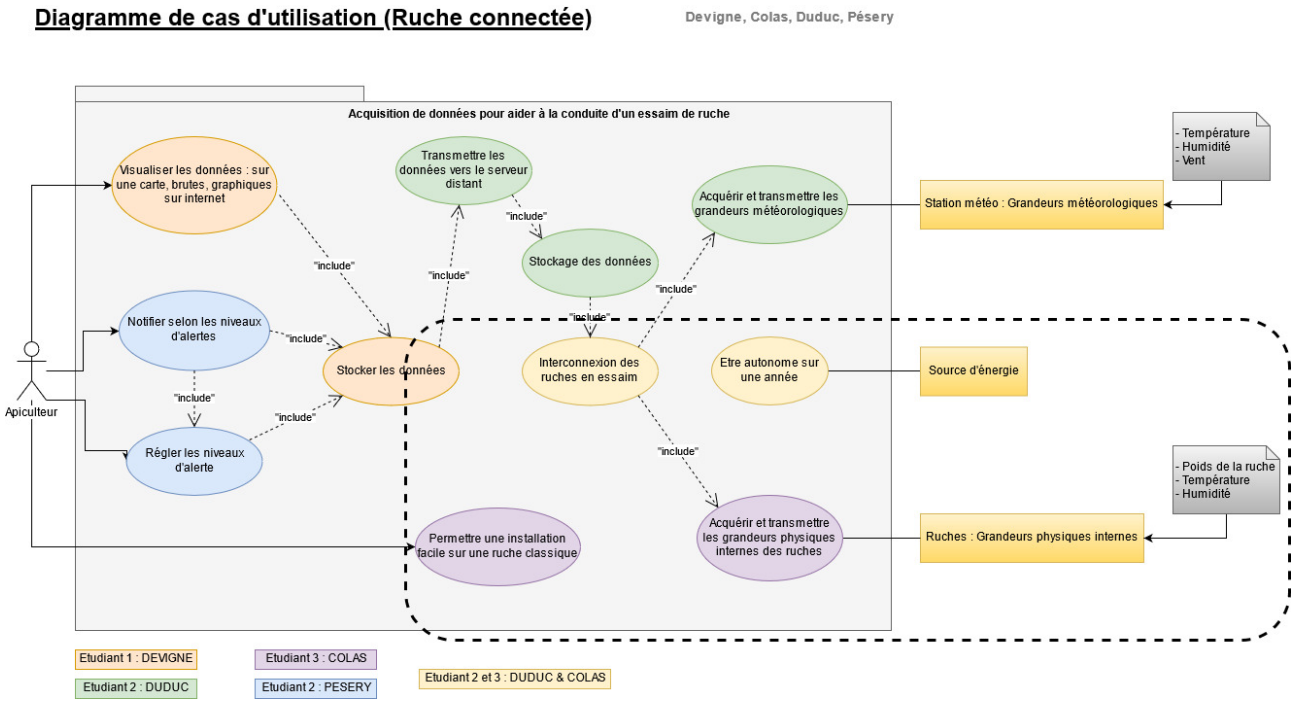
Voici ci-dessous mon carnet de bord réalisée tout au long du projet :

Semaine 1 04 au 08 Janvier 2021	- Analyse du dossier - Début de la réalisation des diagrammes
Semaine 2 11 au 15 Janvier 2021	- réalisation des diagrammes - Début de la réalisation du cahier des charges
Semaine 3 18 au 22 Janvier 2021	- Finalisation des diagrammes et du cahier des charges - Revue 1
Semaine 4 25 au 29 Janvier 2021	- Recherche des composants - Comparaison des composants - Début de la réalisation du devis
Semaine 5 01 au 05 Février 2021	- Finalisation du devis - Début des tests sur les premiers composants
Semaine 6 08 au 12 Février 2021	- Réalisation du rapport de conception
Semaine 7 15 au 19 Février 2021	- finalisation du rapport de conception - Revue 2
Semaine 8 08 au 12 Mars 2021	- Recherche d'une solution d'installation - Recherche sur les composants en attente
Semaine 9 15 au 19 Mars 2021	- Test sur les composants commandé 
Semaine 10 22 au 26 Mars 2021	- Début des tests de communication LORA
Semaine 11 29 Mars au 02 Avril 2021	- Suite des tests de communication LORA
Semaine 12 06 au 09 Avril 2021	- Télétravail : Début de la réalisation du rapport de projet

ANALYSE FONCTIONNELLE

1. Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation ci-dessous est une représentation du comportement fonctionnel d'un système. L'encadré en pointillé entoure ma partie personnel et la partie commun à Antoine.



Explication du diagramme :

Dans ce diagramme, je présente les cas d'usages mobilisant les différentes fonctions de la ruche connectée pour ma partie et celle en commun avec Antoine DUDUC. Il sagit donc de la partie encadré. le code couleur permet de différencier ma parties personnelle de la partie commune.

L'apiculteur doit pouvoir installer facilement le système sur une ruche classique.

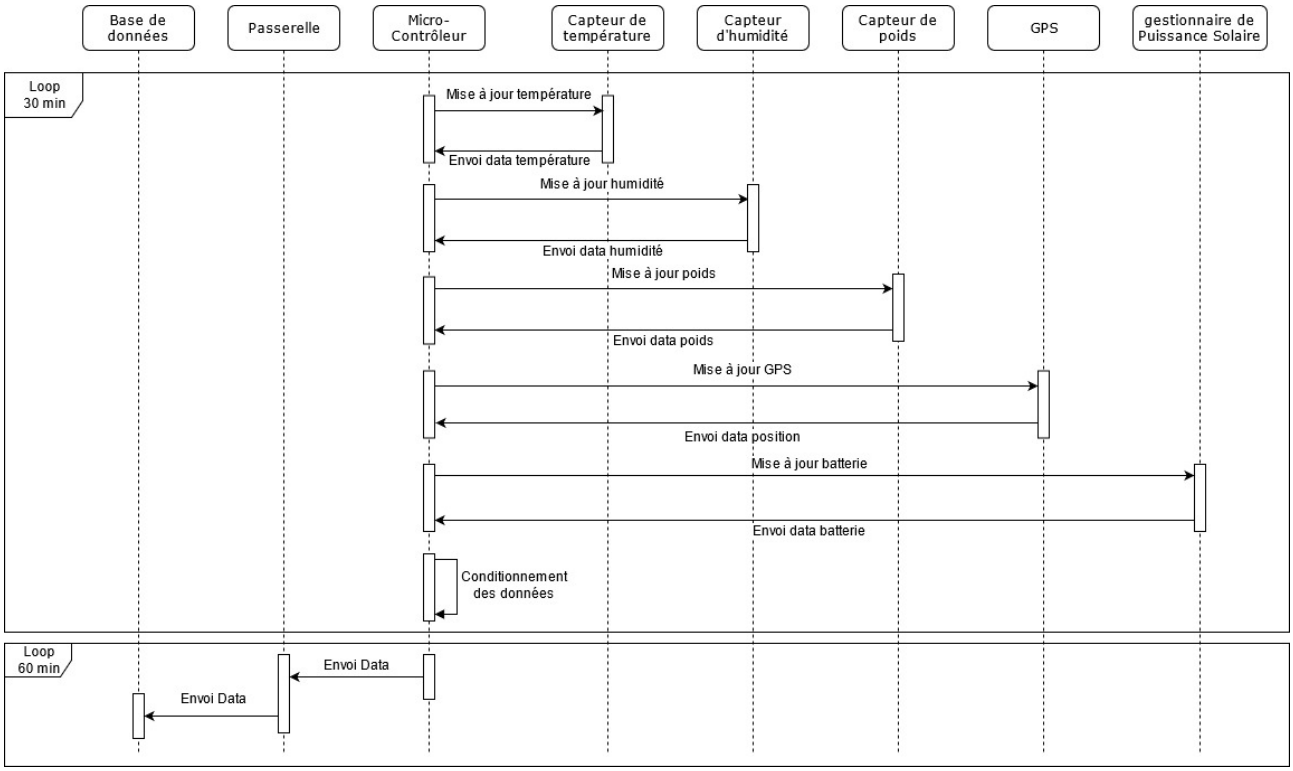
L'interconnexion des ruches en essaim, permet de transmettre les grandeurs physiques interne des ruches à la base de données.

Ces grandeurs physiques interne aux ruches sont acquise par le biais de différents capteurs et transmit via un microcontrôleur.

La source d'énergie permettant l'alimentation des appareils doit être autonome sur une année.

2. Diagramme de séquence

Le diagramme ci-dessous est une présentation graphique des interactions entre les acteurs et le système.



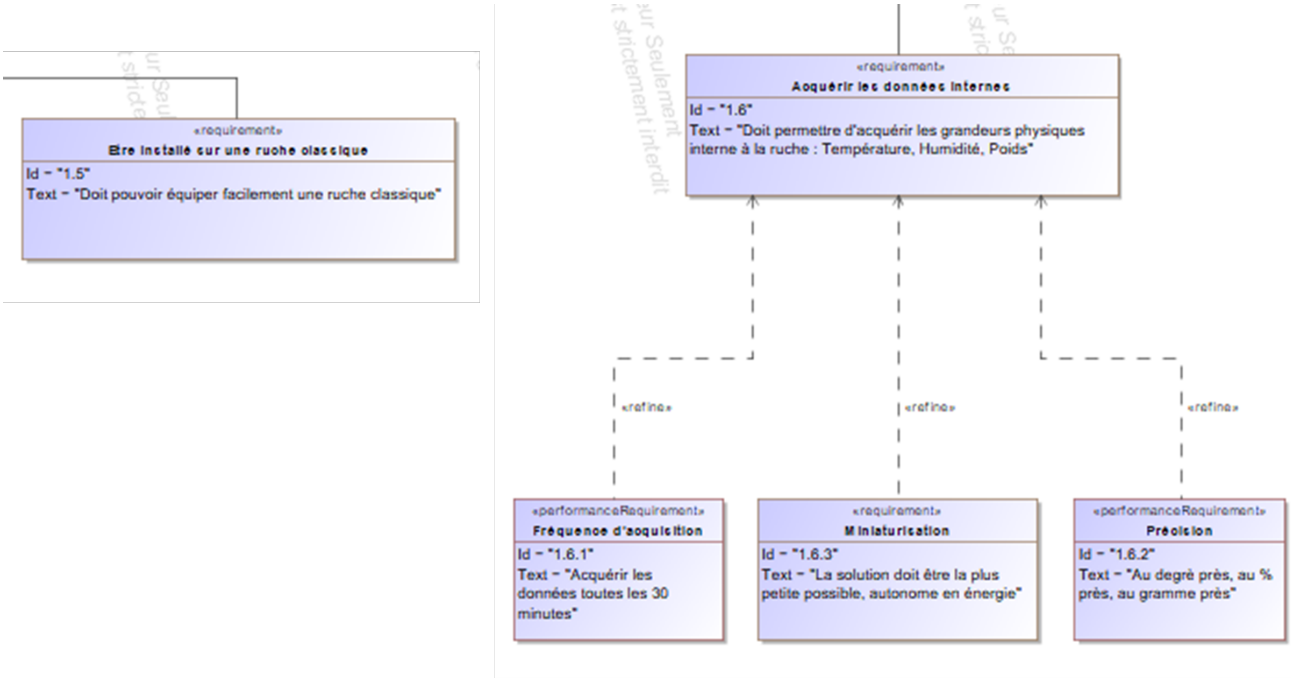
Explication du diagramme :

Le micro-contrôleur est le centre névralgique de ma partie, il interagit avec chaque capteur en récupérant leurs données, mais aussi avec les GPS pour les coordonnées et avec le gestionnaire de Puissance Solaire pour la récupération du niveau de charge de la batterie.
Les interactions s'effectue donc entre le micro-contrôleur et le capteur de température, d'humidité, de poids ainsi qu'avec le GPS et la batterie. Toutes ces données sont conditionnées avant d'être envoyé vers la passerelle puis vers la base de données.

3. Diagramme d'exigence

Les diagrammes d'exigence ci-dessous précise les fonctions à réaliser ainsi que le niveau de performance à atteindre pour les parties qui m'on était attribuée.

- **Diagramme d'exigence de ma partie personnel :**



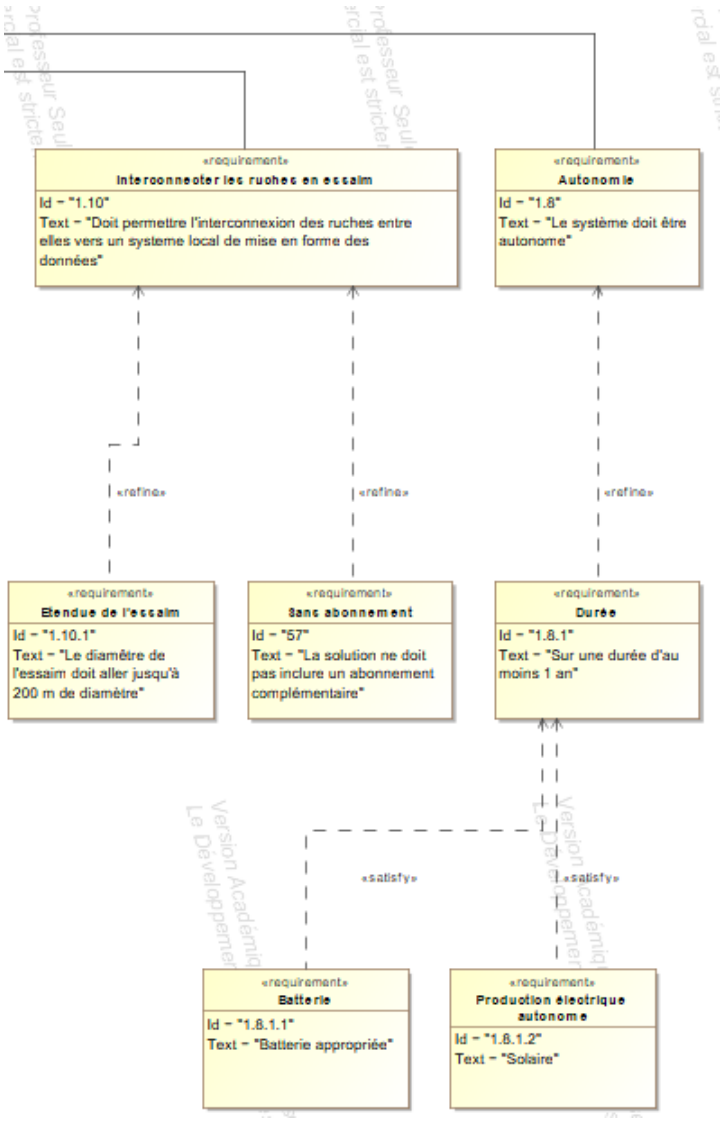
Explication du diagramme :

Sur ce diagramme on peut voir la partie acquérir les données internes qui sont : La température, l'humidité et le poids. Ce diagramme les "refine" donnent des précisions sur :

- La fréquence d'acquisition des données : toutes les 30 minutes.
- La miniaturisation du système : la solution doit être la plus petite possible et autonome en énergie.
- La précision des capteurs :
  - Au degré prés.
  - Au % près.
  - Au gramme près.

J'étais aussi en chargé de l'installation sur la ruche, c'est-à-dire que le système doit être facilement installable sur n'importe quelle ruche classique.

- **Diagramme d'exigence de la partie commune à Antoine DUDUC :**



Ce diagramme montre donc la partie commune avec Antoine. On peut apercevoir la partie principale qui est l'interconnexion des ruches. Sur cette partie il y avait deux contrainte, marqué ici par les "refine" qui sont :

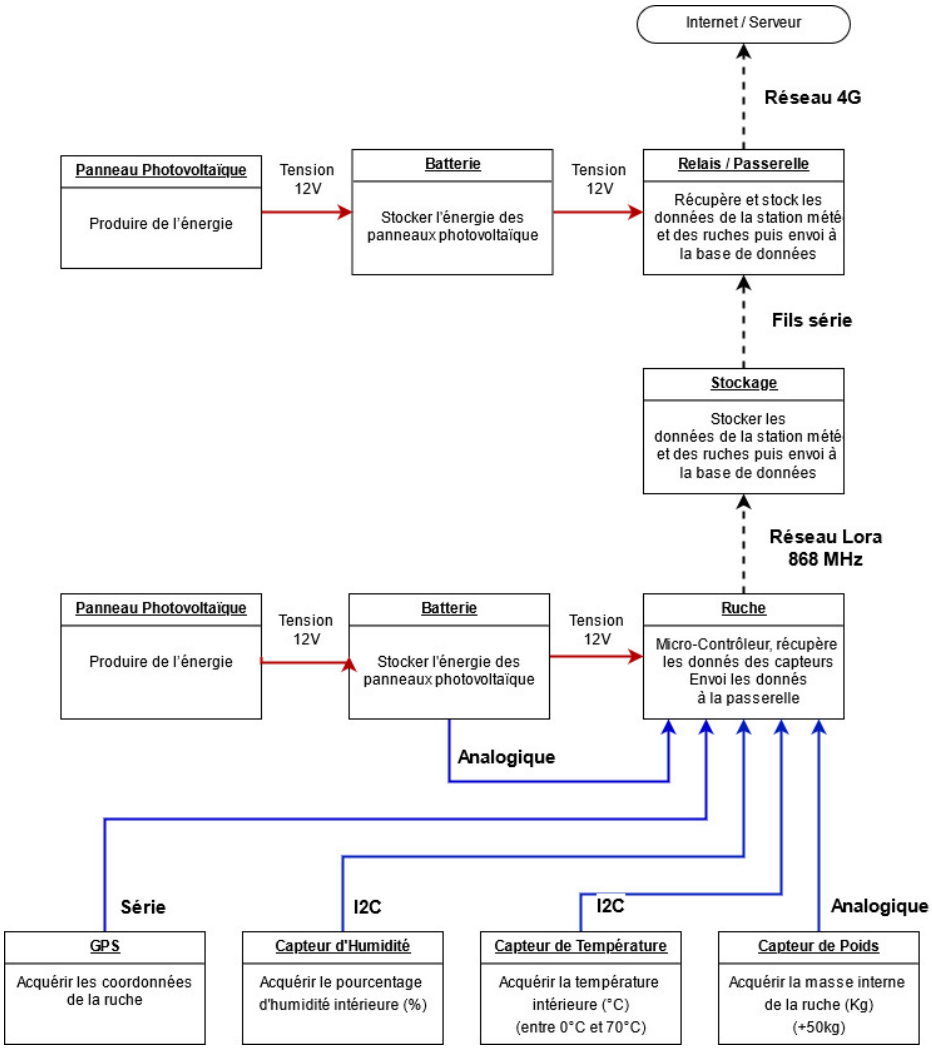
- L'étendu de l'essaim : Essaim devait pouvoir faire 200 mètres de diamètre.
- L'abonnement : la solution ne devait pas inclure d'abonnement complémentaire.

On arrive donc à la seconde partie qui est l'autonomie. Cette partie est simple a expliquer, Le système d'interconnexion des ruches devait être autonome avec une contrainte d'au mois un an. Pour ce faire le diagramme apporte des précisions sur les solution a mettre en oeuvre :

- Une batterie approprié.
- L'énergie solaire pour la production d'électricité autonome.

4. Diagramme de définition des blocs

Le diagramme de définition des blocs ci-dessous décrit la structure du système de ma partie.

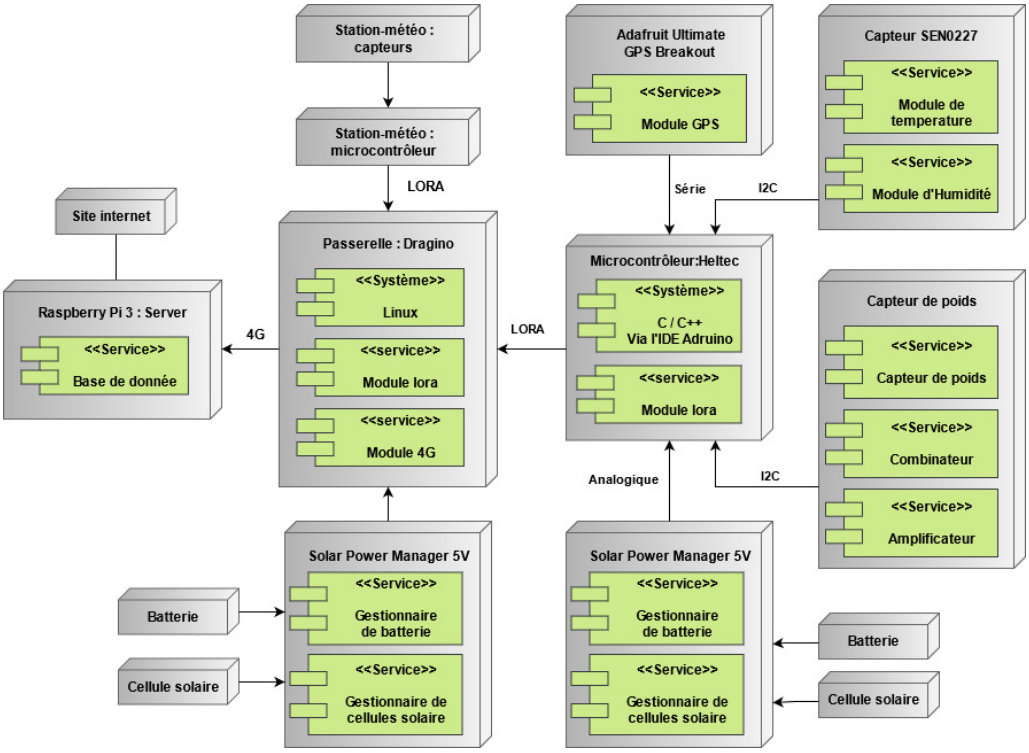


Explication du diagramme :

Le système de ma partie est structuré avec les différents composant relié à la ruche via le microcontrôleur comme les capteurs suivants : humidité, température, poids et GPS ainsi que le panneau solaire et la batterie pour l'autonomie.
Ce microcontrôleur est relié par liaison LORA 868 MHz (norme européenne) à la passerelle qui est alimenté par une batterie et un panneau solaire.
La passerelle transmet ensuite les données via liaison 4G au serveur pour les stocker dans la base de données.

5. Diagramme de déploiement

Le diagramme de déploiement ci-dessous est une représentation de l'utilisation de l'infrastructure physique du système et de la manière dont les composants sont répartis et en relations.



Explication du diagramme :

Dans ce diagramme figure les systèmes d'exploitations, les langages et les services installés dans les différents composants physique qui composent ma partie et celle en commun avec Antoine comme par exemple :

- le capteur de poids qui est un ensemble de quatre capteurs de poids relié par un combinateur qui relie les files des quatre capteurs à un amplificateur qui lui permet la lecture des modifications des résistances, pour avoir des mesures plus précises.
- Le "Solar Power Manager", qui est le gestionnaire de puissance solaire permettant d'utiliser une batterie pour alimenter le microcontrôleur et les capteurs. Il peut aussi accueillir une cellule solaire pour recharger cette batterie.
- La passerelle "Dragino", qui fonctionne avec un système Linux et qui est composée d'un module LORA et d'un module 4G.

Le diagramme représente aussi les types de liaisons qui connectes les différents éléments entre eux. Les principaux sont les liaison LORA et 4G qui permettent le transfère des données via la passerelle. Mais il y en a d'autre comme les liaisons I2C, Analogique et Série.

RÉALISATION ET CONFIGURATION

1. Transmission des données

Pour transmettre les données, nous devions, Antoine DUDUC et moi même, utiliser une passerelle. Pour se faire des recherches sur les différents protocoles de communication. Après plusieurs recherches, nous avons élaborée le comparatifs suivant :

	WIFI	BLUETOOTH	LORA	ZIGBEE	SIGFOX	Z-WAVE	CELLULAIRE	NEUL	FILAIRE
PORTÉES	+ (~50m)	+ (~50m)	+++ (~5km)	+ (~100m)	++++	+ (~30m)	+++++	++++	+++++
ÉCONOMIE ÉNERGIE	+	+++	++++	++++	++++	++++	++	++++	+
DEBIT	++++	++++	++	+++	+	++	++++	+++	++++
COUT	+++	++++	++	++	++	++	+++	++	+
INTÉGRATION	++++	++++	+++	+++	+++	+++	++++	+++	++++
ABONNEMENT	+++++ (Non)	+++++ (Non)	+++ (Au choix)	+++ (Au choix)	+ (oui)	+++ (Au choix)	+ (oui)	+ (oui)	+++++ (non)

La solution sélectionnée pour relier les ruches à la passerelle est : Le LORA

La solution de communication avait pour contrainte :

- Elle devait être disposée dans un rayon de 5 KM maximum autour du ruché.
- Elle devait être la moins énergivore possible.
- Elle ne devait pas avoir besoin d'un abonnement supplémentaire.

Après avoir comparé les différentes solutions, nous avons décidé de sélectionner la solution du LORA pour la communication entre le ruché et la passerelle. Notre choix c'est porté sur cette solution car elle remplissait les contraintes imposée :

- Par sa portée : 5 KM dans un environnement urbain.
- Par sa faible consommation : 125 mA à l'émission, 14 mA à la réception.
- Par son indépendance a un abonnement.

La solution sélectionnée pour relier la passerelle à la base de données est :
Le CELLULAIRE

Pour ce qui est de la communication entre la passerelle est le server ou se trouve la base de données, Nous avons eu plus de mal à faire notre choix. Cependant, Nous avons fait le choix de la communication cellulaire malgré son besoin d'abonnement, car c'est le seule moyen de communication qui couvre une grande partie de la France.

Choix de la passerelle :

Pour choisir la passerelle, nous avons donc pris en compte nos choix précédents, c'est-à-dire qu'il fallait que l'on trouve une passerelle capable de communiquer par liaison LORA et liaison 4G.
Après avoir parcouru touts les sites prescrits, nous avons trouvé une seule passerelle correspondante à nos attentes :

"Passerelle IoT LoRa monocanal Dragino LG01-N 868 MHz / EC25-E w / 4G (EU)"

Cette passerelle est équipée avec :

- Un module LORA 868 MHz pour la norme Européenne.
- Du module "EC25-E w" qui permet d'avoir une liaison 4G grâce à une carte SIM.

Pour rappel cette passerelle permet de récupérer les données des différentes ruches et de la station météo via le module LORA. Elle permet aussi de transmettre ces données au serveur et plus particulièrement à la base de donnée via le module 4G.

2. Acquisition des données

Pour acquérir les données, j'ai dû faire des recherches sur les différents composants nécessaires. Après plusieurs recherches, j'ai élaborée des comparatifs sur ces composants.

À commencer par le microcontrôleur :

	Arduino Uno	Raspberry Pi 3 B+	ADAFRUIT FEATHER 32u4 Rfm95	LILYGO - TTGO T BEAM ESP32 LoRa 868Mhz
Économie énergies	++++	++	++++	++++
Cout	++++	++	++	++++
Pin analogique	6	?	10	?
Pin digital	14	?	20	?
Module LORA inclue	Non	Non	Oui	Oui
Dimensions	74 x 53 x 15 mm	86 x 54 x 17 mm	51 x 23 x 8 mm	100 x 33 x 20 mm
GPS intégré	Non	Non	Non	Oui
Disponible sur les sites préscris	Oui	Oui	Oui	Non
Mémoire Flash	++	Besoins SD	++	+++

Microcontrôleur sélectionné : ADAFRUIT FEATHER 32u4 Rfm95

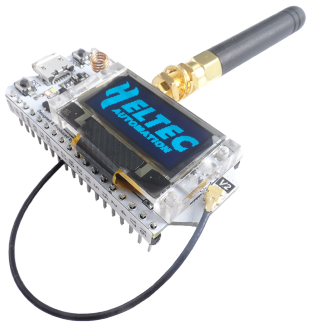
Le microcontrôleur devait respecter les contraintes suivantes :

- Il devait être le plus petit possible.
- Il devait avoir un module LORA intégré ou la possibilité d'en accueillir un, pour établir la communication avec la passerelle.
- Il devait être le moins énergivore possible.
- Il devait être disponible sur les sites prescrit.

Après avoir comparé les différents microcontrôleurs, j'ai décidé de sélectionner L'Adafruit Feather 32u4 Rfm95, car il convenait aux contraintes imposée :

- Par ses dimensions : 51 x 23 x 8 mm.
- Par son module LORA intégré.
- Par sa faible consommation : 120 mA en transmission et 40 mA hors transmission.
- Par sa disponibilité sur les sites prescrit.

Suite à un problème avec le vendeur, nous avons dû réaliser une nouvelle commande et donc changer de microcontrôleur. Nous avons donc trouvé le "WiFi LoRa 32 (V2) Heltec" qui en touts points similaires à l'exception de l'ajout d'un écran LCD et de deux modules : Wi-Fi et Bluetooth. Sa taille quant à elle, diffère que très légèrement : 51 x 25.5 x 10.6 mm.

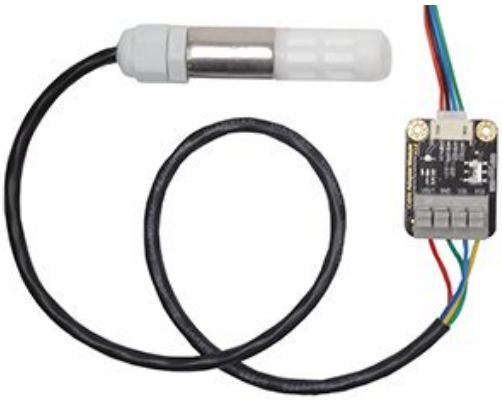


J'ai ensuite cherché un capteur de température et un capteur d'humidité :

	SEN0227	DHT22	SHT10	DHT11
Économie énergies	++++	++++	++++	++++
Cout	+++	++++	++	++++
Étanche	Oui	Non	Non	Non
Plage de mesure	+++++	+++++	++++	++++
Précision	++++	++++	+++	+++
Dimension	73 x 17 mm	25 x 15 x 9 mm	75 x 14 mm	16 x 12 x 7 mm
Capteur d'humidité et de température sélectionné : SEN0227				

Le capteur devait respecter les contraintes suivantes :

- Il devait être le plus petit possible.
- Il devait avoir une précision au demi degrés prés pour la température et au pourcentage prés pour l'humidité.
- Il devait être le moins énergivore possible.
- Il devait être étanche.

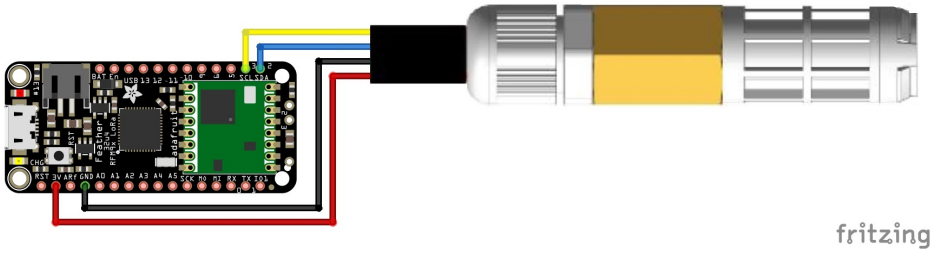


Après avoir fait mes recherches sur les différents sites prescrit, et après avoir comparé plusieurs capteurs, j'ai décidé de sélectionner le capteur SEN0227 car il convenait à mon cahier des charges :

- Par son étanchéité.
- Par sa précision.
- Par sa faible consommation.

Son seul point faible est sa taille qui est un petit peu grande, mais qui reste tout de même respectable. Au vu de la difficulté de trouver un capteur étanche. L'étanchéité du capteur est indispensable du fait de sa proximité avec le miel et la cire.

Voici le schéma de câblage mis en place pour ce capteur :



Le capteur est alimenté en 3.3 V avec le fil rouge et relié à la terre par le fil noir. Pour ce qui est des données, elles transitent via une liaison I2C par les ports SCL (Serial Clock Line) et SDA (Serial Data Line)du microcontrôleur qui sont respectivement les fils jaune et bleu.

Ci-dessous se trouve l'algorithme que j'ai réalisé pour faire fonctionner la partie acquisition de la température et de l'humidité.

test_T_H_esp §

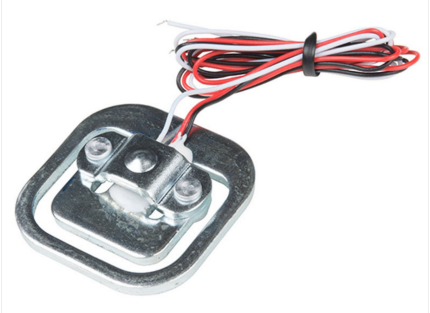
```
1
2  //Inclusion des bibliothèques
3  #include <Arduino.h>
4  #include <heltec.h>
5  #include "DFRobot_SHT20.h"
6
7  DFRobot_SHT20    sht20;
8
9  void setup() {
10   Serial.begin(9600);
11   Heltec.begin(false, true, true);    // OLED désactivé, LoRa activé, Serial activé
12   Wire.begin(SDA_OLED, SCL_OLED);     // Paramétrage des pins
13   sht20.initSHT20();                  // Initialisation du capteur
14   delay(100);
15   sht20.checkSHT20();                 // Check du capteur
16 }
17
18 void loop() {
19   float hum = sht20.readHumidity();          // lecture de l'humidity
20   float temp = sht20.readTemperature();      // lecture de la Température
21   delay(1800000);      // 1 800 000 secondes correspond à 30 minutes
22 }
```

Pour ce qui est du capteur de poids, j'ai réalisé le comparatif suivant :

Capteur :	CZL635-5	SFE RB-Spa-488	RB-Phi-121
Capacité	5 Kg	50 Kg	50 Kg
Dimension	55 x 12.7 x 12.7 mm	35 x 35 mm	25 x 11 mm
Précision	0.05 %	0.1 %	0.2 %
Tension d'excitation	5 V	</= 10 V	5 V
Capacité de surcharge max	150 %	150 %	150 %
Capteur de poids sélectionné : SFE RB-Spa-488			

Le capteur devait respecter les contraintes suivantes :

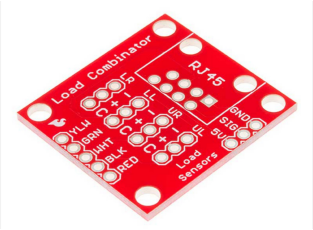
- Il devait supporter le poids de la ruche pleine (100 kg), c'est-a-dire 25 Kg par capteur.
- Il devait avoir une précision d'au moins 1 gramme.
- Il devait être le plus petit possible.



Une fois mes recherches effectuée, et après avoir comparé plusieurs capteurs, j'ai décidé de sélectionner le capteur SFE RB-Spa-488 car il convenait :

- Par sa capacité de 50 Kg : ce qui correspond a une capacité maximum de 200 Kg avec les quatre capteurs.
- Par sa précision : 0.1 gramme prés.
- Par sa taille : 35 x 35 mm, c'est assez petit pour permettre de l'associer à une structure simple à installer et assez petite.

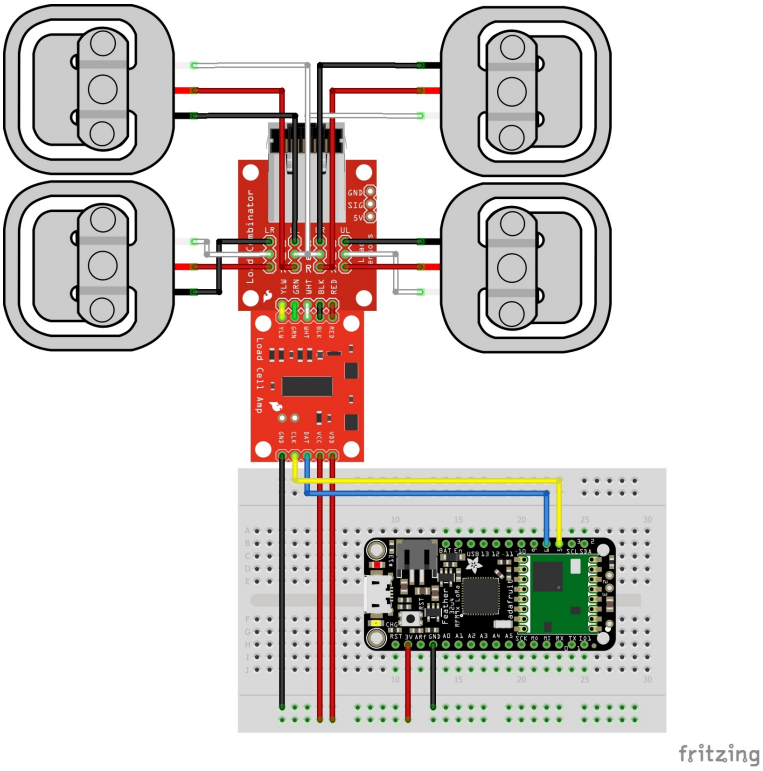
À ces quatre capteurs de poids, j'associe un combinateur qui je le rappelle sert à rassembler les fils des capteurs. J'associe aussi un amplificateur qui sert à améliorer la précision des données combinée.



Combinateur

Amplificateur

Voici le schéma de câblage des capteurs de poids :



Ci-dessous se trouve l'algorithme que j'ai réalisé pour faire fonctionner la partie acquisition du poids.

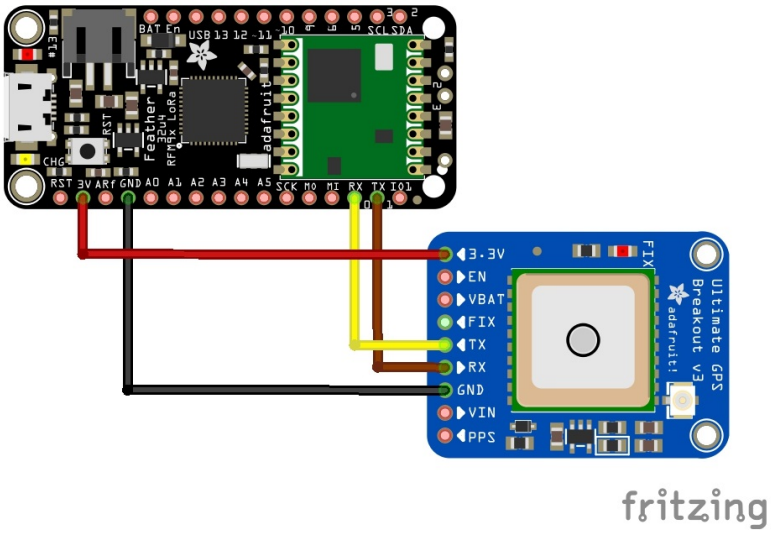
test_capt_poids

```
#include "Arduino.h"
#include "HX711.h"

#define DOUT  2
#define CLK  4

HX711 scale;

float calibration_factor = -22000; //par default -7050

void setup() {

  scale.begin(DOUT, CLK);
  scale.set_scale();
  scale.tare(); //Tarage

  long zero_factor = scale.read_average();
  Serial.print("Zero factor: ");
  Serial.println(zero_factor);
}

void loop() {

  Serial.print("Reading: ");
  Serial.print(scale.get_units(), 1); //récupération des valeurs
  Serial.print(" Kg");
  Serial.println();
  delay(1000);
}
```

Pour le GPS, J'ai pris le module GPS "Adafruit Ultimate GPS Breakout" qui était a disposition dans le matériel de l'établissement.

Pour rappel le module GPS devait servir à avoir les cordonnées des ruches pour connaitre leur emplacement et pour qu'elle apparaisse sur une carte.

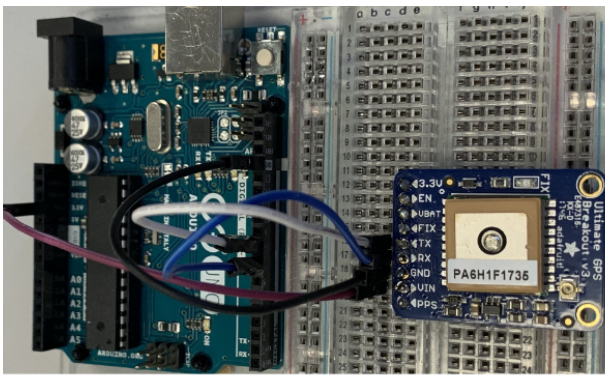
Ci-dessous se trouve le schéma de câblage du module GSP :



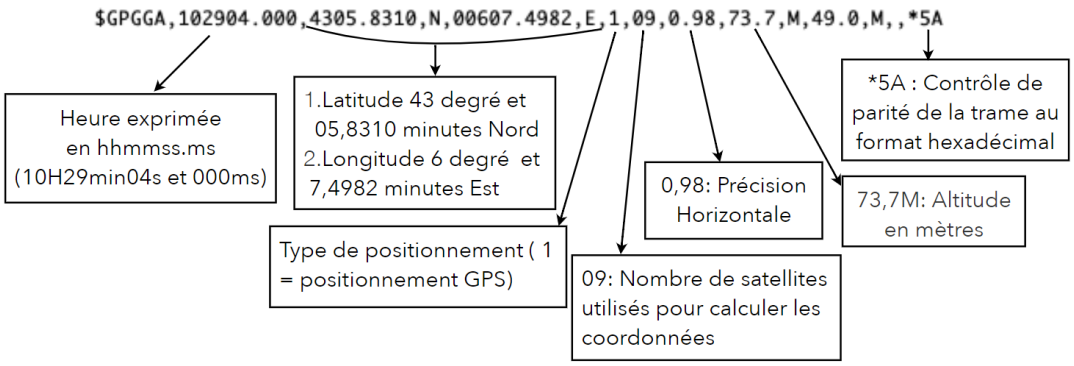
Pour ce module GPS, j'ai réalisée des tests avec un Arduino UNO avants de recevoir le microcontrôleur final.

Après avoir réalisé le câblage ci-contre, j'ai installé la bibliothèque fournie par le constructeur (ADAFRUIT). Puis j'ai testé un exemple fournis par la bibliothèque.

Ce test m'a permis de voir le genre de trame que l'on reçoit, mais aussi d'avoir une approche sur le fonctionnement de ce module pour le projet.



Voici un exemple de trame reçu :



Une fois le microcontrôleur final en ma possession, j'ai réalisé l'algorithme suivant pour utiliser le module GPS.

algorithme GPS

Voyons maintenant le Gestionnaire de puissance solaire :

Pour rappel, J'avais pour objectif de rendre mon système autonome sur une durée de un an minimum. Pour cela il me fallait donc une batterie approprié et un panneau solaire pour la recharger durant la journée.
Pour ce faire j'ai donc décidé de prendre un gestionnaire de Puissance Solaire.

Le "Solar Power Manager 5V" m'as était fournie par mon professeur

- Gestionnaire de Puissance Solaire
- panneau solaire
- batterie
    - analyse
    - description de l'utilisation
    - Schéma fritzing
    - algorithme pourcentage batterie

BTS SN

Spécialité : IR

Lycée Costebelle

83400 – Hyères

Dossier personnel :

RUCHE CONNECTÉE

Étudiant 4 : Notifications

- SMS
- E-mail

Session 2021

SOMMAIRE

Introduction …….. 0
Semainier …….. 0
Diagrammes …….. 00 à 00
Contraintes techniques …….. 0
Solutions choisies & comparaisons ……..…….. 00 à 00
Réalisation & configuration ……. 00 à 00

INTRODUCTION

Que serait une ruche connectée sans système de notification ?

Mon rôle en tant qu'étudiant 4 consiste à mettre en place un système de notification qui permettra de notifier l'apiculteur de l'état de ses ruches et de leur environnement, pour ce faire, les systèmes de notifications choisis sont l'envoi d'SMS et d'E-mail.

Un onglet « notification » réalisé par mes soins sera disponible sur le site internet du projet réalisé par l'étudiant 1, pour permettre à l'apiculteur de régler ses préférences de notifications.

Pour ce faire il me faudra récupérer, stocker et comparer les valeurs reçus des différents capteurs avec les seuils que l'apiculteur aura défini sur le site internet.

Les capteurs en question sont, pour la ruche : température, humidité, poids ; et pour la station météo : température, humidité, vent. Il sera également possible d'être alerté sur le pourcentage de batterie de chaque zone.

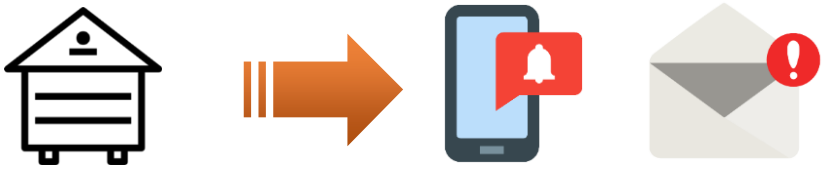
L'apiculteur pourra être notifié :

- Instantanément, si une valeur d'un capteur dépasse le seuil défini.
- De manière journalière, afin d'obtenir un résumé des valeurs minimum, maximum et une moyenne des différents capteurs.

L'apiculteur pourra activer ou désactiver les notifications de chaque capteur indépendamment s'il souhaite ou non être notifié.

Il pourra aussi choisir le système d'envoi de notification de chaque capteur indépendamment.

Des boutons pour mettre les valeurs par défaut conseillées seront aussi à la disposition de l'apiculteur.



CARNET DE BORD

Commenté [Loan Pese1]: Images ?

Dans cette partie nous retrouvons la liste des tâches réalisées durant chaque semaine en partant de l'étude et des recherches en passant par les tests et la réalisation et en finissant par l'édition des documents.

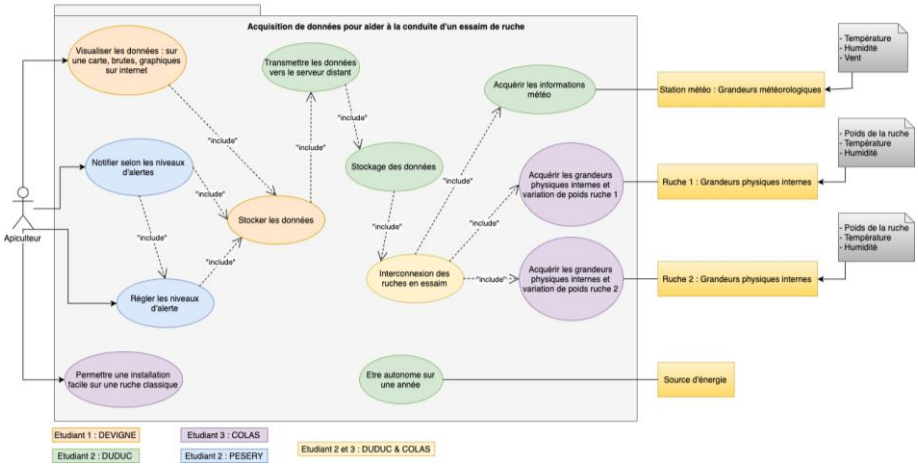
Semaine 1 – 05 au 08 janvier	➢ Analyse du dossier ; ➢ Rassemblement des idées avec l'équipe ; ➢ Début réalisation des diagrammes.
Semaine 2 – 11 au 15 janvier	➢ Finalisation des digrammes ; ➢ Début réalisation du cahier des charges ; ➢ Organisation sur Trello & Gantt.
Semaine 3 – 18 au 22 janvier	➢ Finalisation du cahier des charges ; ➢ Préparation revue n°1 ; ➢ Revue n°1.
Semaine 4 – 25 au 29 janvier	➢ Début des recherches E-mails ; ➢ Premiers tests E-mails ;
Semaine 5 – 01 au 05 février	➢ Début des recherches SMS ; ➢ Premiers tests SMS ;
Semaine 6 – 08 au 12 février	➢ Début réalisation de la page web ; ➢ Premiers tests avec la base de données ; ➢ Mise en place des sliders.
Semaine 7 – 15 au 19 février	➢ Préparation de la revue n°2 ; ➢ Mise en place des checkbox ; ➢ Revue n°2 ; ➢ Rapport de conception.
Semaine 8 – 08 au 12 mars	➢ Ajouts menus déroulant pour choix notifications ; ➢ Finalisation de la base de données.
Semaine 9 – 15 au 19 mars	➢ Début réalisation code pour comparaison ; ➢ Ajout des commandes au démarrage ; ➢ Exécution code comparaison toutes les minutes ;
Semaine 10 – 22 au 26 mars	➢ Finalisation du code de comparaison ; ➢ Amélioration de l'esthétique du site ; ➢ Ajout du code contrôlant les champs vide.
Semaine 11 – 29 mars au 02 avril	➢ Écriture des phrases envoyées pour les alertes ; ➢ Ajout des boutons pour les valeurs par défaut ; ➢ Ajout partie notifications journalières.
Semaine 12 – 06 au 09 avril	➢ Édition des documents finaux.

ANALYSE FONCTIONELLE

Dans cette partie nous retrouvons l'étude fonctionnelle de ma partie Notifications.

1. Diagramme de cas d'utilisation

Le **diagramme de cas d'utilisation** que nous retrouvons ci-dessous est une représentation du comportement fonctionnel du système global.



Dans ce diagramme, nous présentons les cas d'usages mobilisant les différentes fonctions de la ruche connectée.

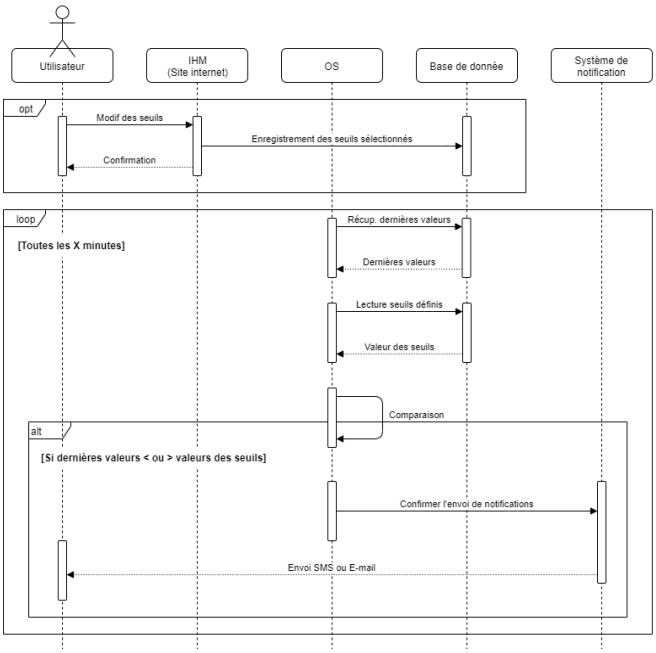
Un code couleur permet de visualiser les différentes parties des étudiants ;

Ma partie est représentée ici par le couleur bleu, on peut donc voir que j'interagi principalement qu'avec l'utilisateur, qui ici est l'apiculteur, mais aussi avec la base de données qui est l'endroit où toutes les valeurs des seuils et des capteurs vont être stockées, très importantes pour pouvoir notifier l'apiculteur sur l'état de ses ruches.

2. Diagrammes de séquences

Les **diagrammes de séquence** que nous retrouvons ci-dessous sont une représentation graphique des interactions entre l'utilisateur et le système, ils montrent comment devra fonctionner le système de notifications.

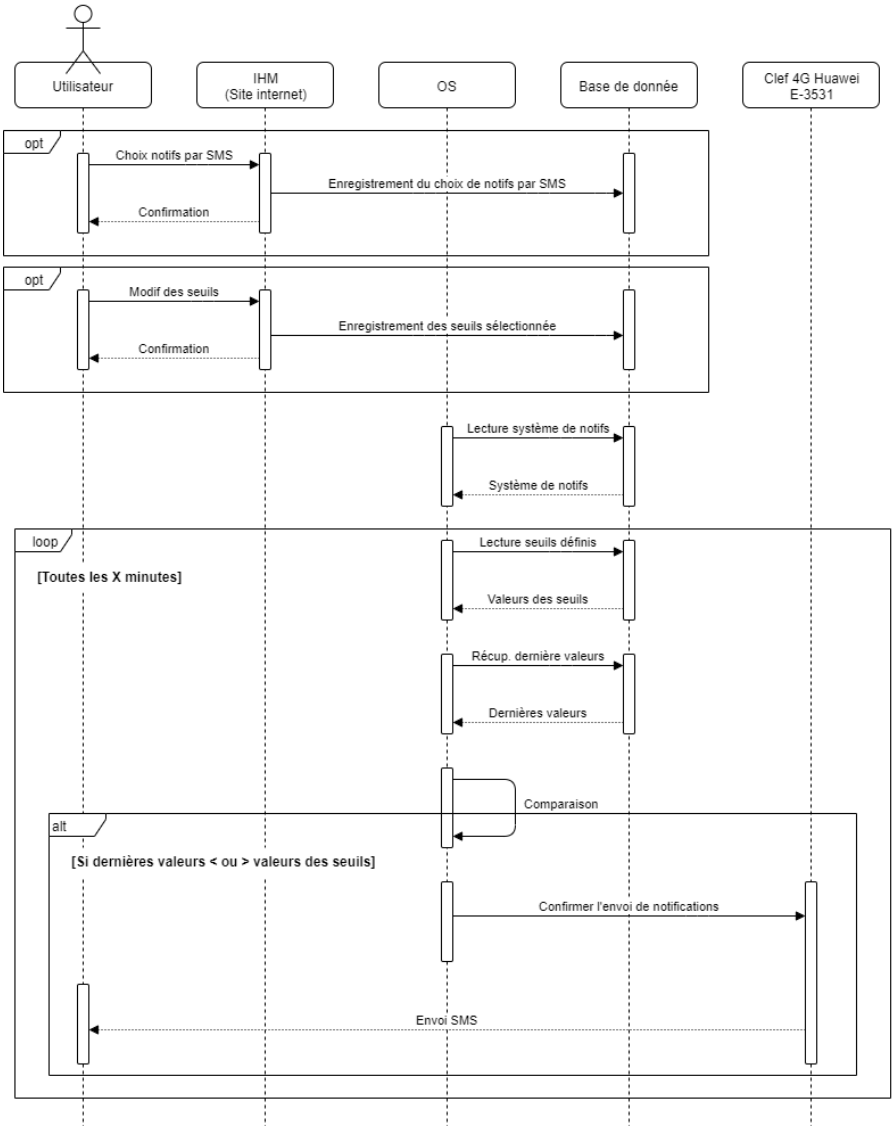
- Description du fonctionnement souhaité d'un point de vue général



Je vous présente ici le diagramme de séquence d'un point de vue global.

- **L'utilisateur** : représente la personne qui utilisera le système, dans le cas de notre projet c'est un apiculteur.
- **IHM** : représente la page internet où figureront les curseurs permettant de régler les seuils d'alertes.
- **OS** : représente le code qui récupèrera les données, les compareras et décidera d'envoyer les notifications ou non à l'apiculteur.
- **Base de données** : représente l'endroit où les données des seuils et des capteurs seront stockées.
- **Système de notification :** représente les différents systèmes d'envoi de notifications : SMS et E-mail.

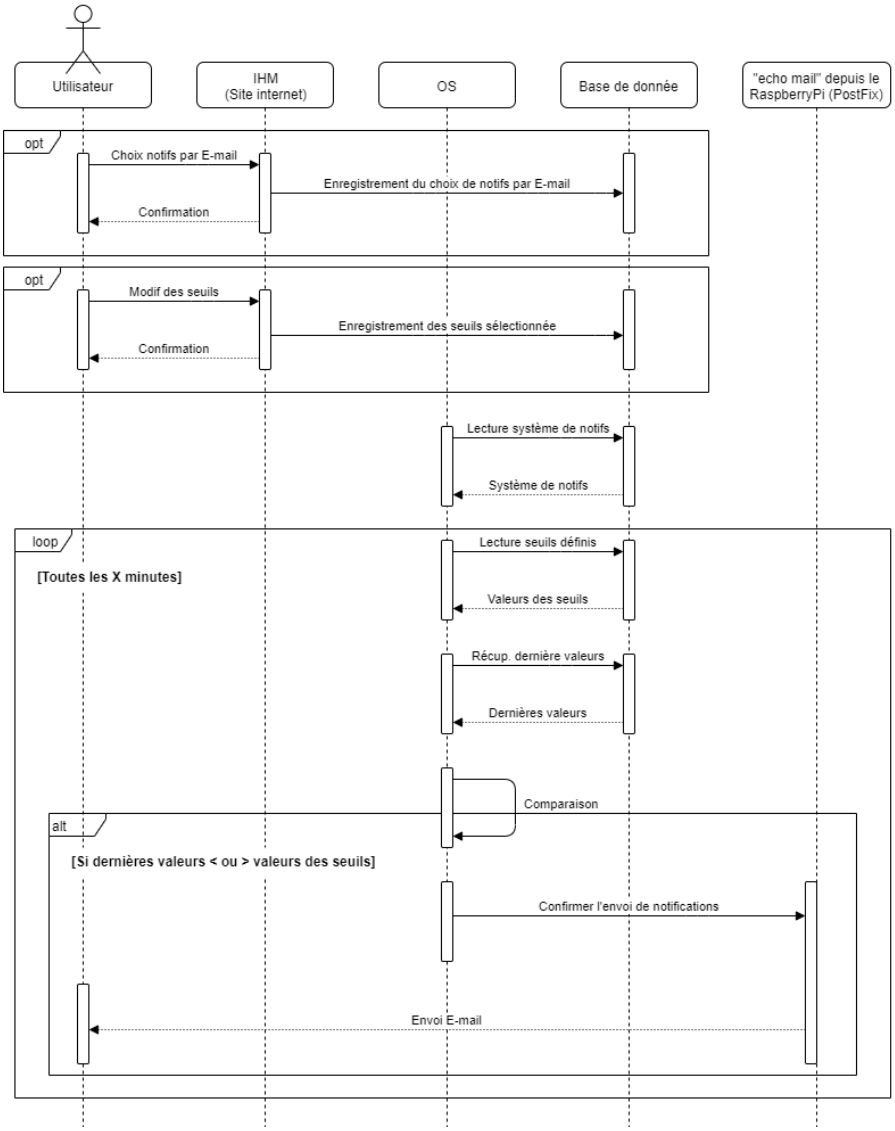
- Description du fonctionnement du système souhaité de notification par SMS



Je vous présente ici le diagramme de séquence concernant le système d'envoi des SMS.

- **Clef 4G Huawei E-3531 :** représente le système physique choisi pour permettre l'envoi des notifications via SMS.

- Description du fonctionnement souhaité du système de notification par E-mail

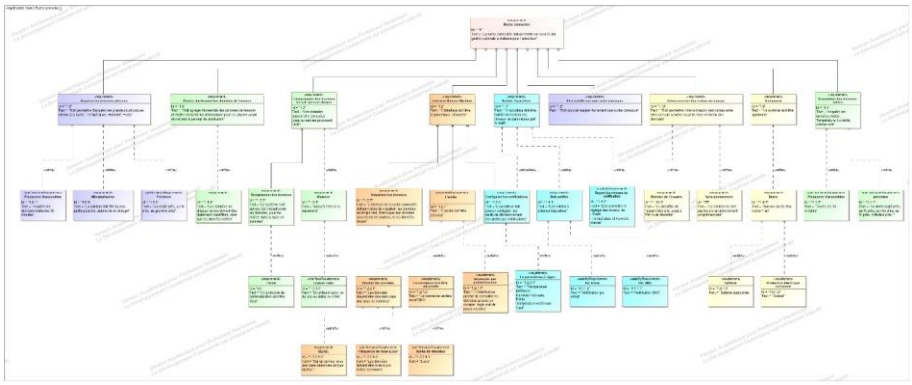


Je vous présente ici le diagramme de séquence concernant le système d'envoi des E-mail.

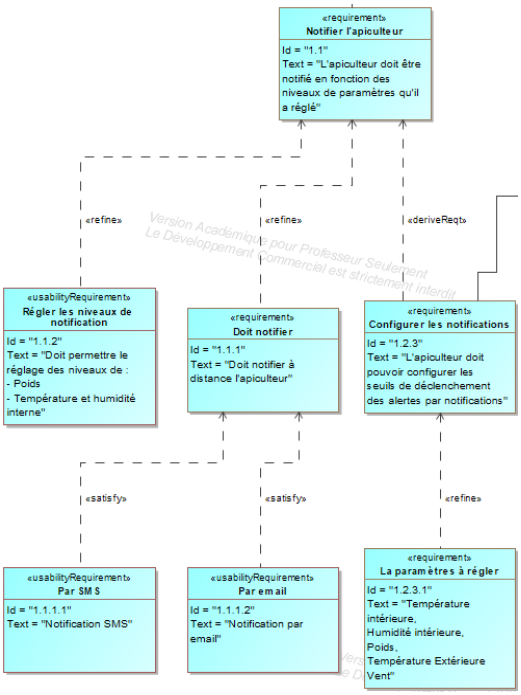
- **« echo mail » :** représente le système choisi qui permettra l'envoi des notifications via E-mail.

3. Diagramme d'exigence

Le **diagramme d'exigence** que nous retrouvons ci-dessous précise les fonctions à réaliser ainsi que le niveau de performance à atteindre pour le projet.



- Zoom la partie de l'étudiant 4



En ce qui concerne ma partie dans ce diagramme :

On voit que l'apiculteur doit pouvoir être notifié selon les seuils d'alertes qu'il aura paramétrés.

Ces alertes interviennent à distance, comme on peut le voir ci-contre, par SMS ou par E-mail.

Les seuils qui peuvent être réglés sont en rapport avec : La température intérieure et extérieure, l'humidité intérieur et extérieure, le poids, et le vent (mais aussi le pourcentage des batteries).

Notre but étant de maintenir les ruches en bonne santé, ce sont ces paramètres qui ont été choisis afin de les maintenir en parfaite santé.

4. Diagramme des classes

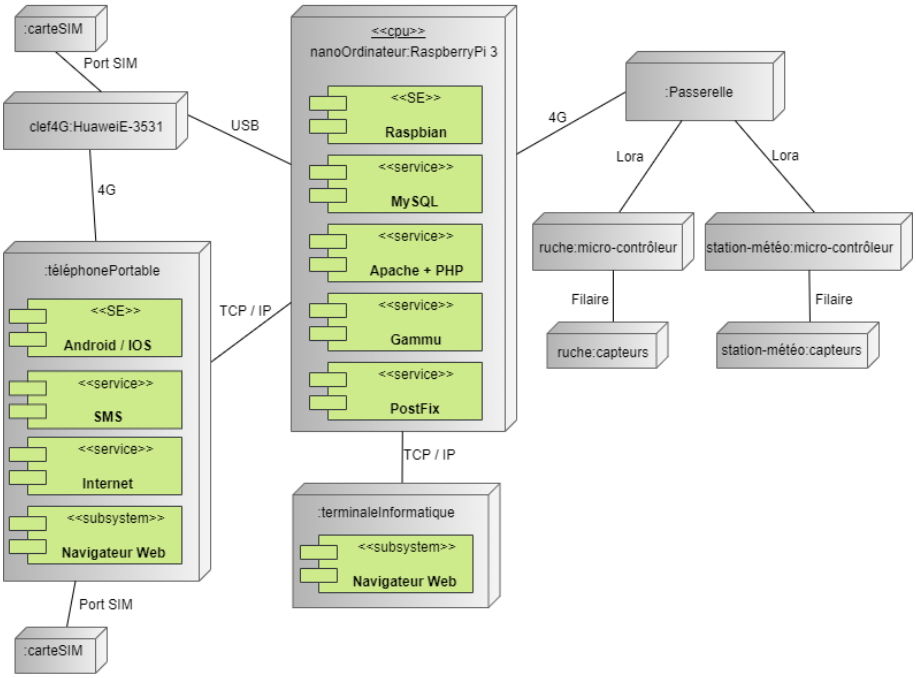
Commenté [Loan Pese2]: Conception logiciel – TITRE
Donne paramètre de la fonction et quoi réalise cette fonction
ENVOYER AU PROF

5. Diagramme de déploiement

Le **diagramme de déploiement** que nous retrouvons ci-dessous représente les matériels physiques et les services qui ont pu leurs être installés.

Ce diagramme est détaillé en fonction de ma partie notification ; les parties des autres membres de l'équipe (disponible dans leur partie personnel) sont ici résumé afin de mieux comprendre le diagramme.

Commenté [LP3]: Pointillés parties Antoine et Dorian

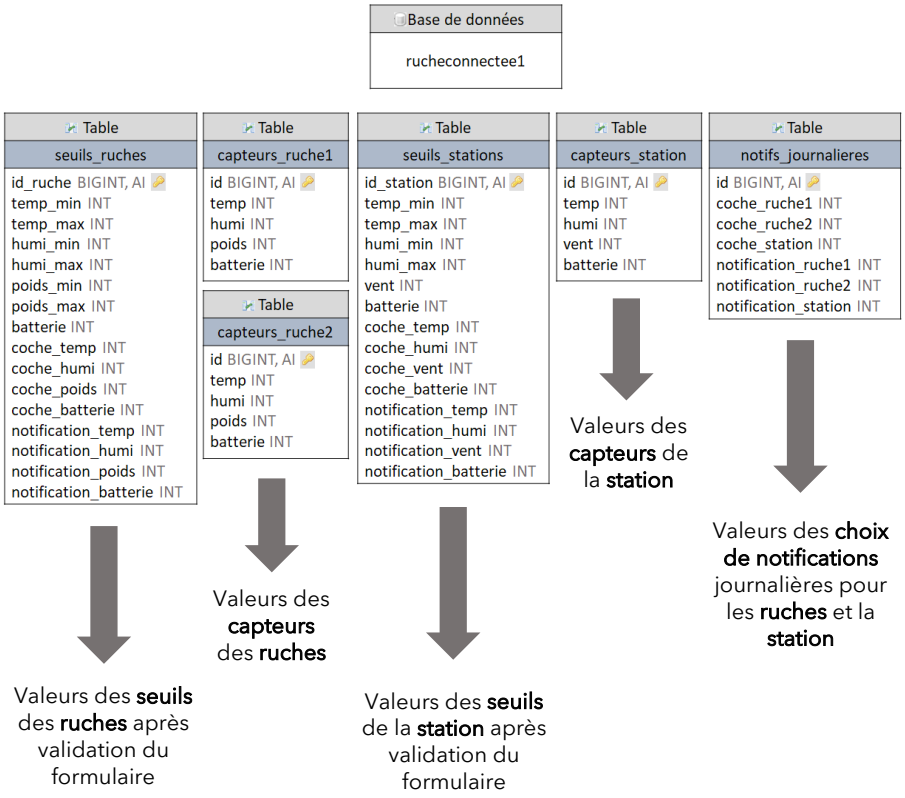


Dans ce diagramme figure le système d'exploitation et les services installés dans le RaspberryPi 3 : Raspbian, MySQL, Apache + PHP, Gammu, PostFix ainsi que le système d'exploitation et les services du téléphone portable : Android / IOS, SMS (carte SIM), Internet (connexion à internet), Navigateur Web (pour consulter le site web).

Nous retrouvons aussi les appareils nécessaires au bon fonctionnement du système d'envoi des notifications : clef 4G Huawei E-3531, carte SIM, Passerelle, …

6. Schéma base de données

Le **schéma de la base de données** que nous retrouvons ci-dessous présente les différentes tables appartenant à la base de données « rucheconnectee1 ».



Dans le cas de notre projet, les tables concernant la ruche 2 ne seront pas utilisées, elles sont présentes ici à titre d'exemple afin de comprendre la logique de la base de données.

Seul les tables de la ruche 1, de la station météo et des notifications journalières seront utilisée dans ce projet.

CONTRAINTES TECHNIQUES

Pour la réalisation de ce projet, des contraintes de **développement**, de **qualité** et de **fiabilité / sécurité** nous sont imposés :

Développement :

- Utilisation d'un RaspberryPi (Raspbian comme OS) ;
- Pour les SMS : (API SMS ou clef 4G) ;
- Pour les E-mail : (Sendmail ou boite de messagerie sur le RaspberryPi) ;
- Pour la partie serveur Web et site Web : HTML/JS/CSS/PHP ;
- Pour la base de données : MySQL, serveur Apache2 ;
- Le site web de paramétrage et de consultation des données doit être responsive.

Qualité :

- Soin au niveau de l'ergonomie des interfaces : faciles d'utilisation, intuitives et conviviales ;
- Pour le code : entête de fichier (auteur, dates, outils de production utilisés), entête de fonctions (rôle de la fonction & utilisation des paramètres), commentaire (compréhension du code).

Fiabilité / sécurité :

- Installation sur un réseau accessible, robuste et sécurisé.

SOLUTIONS CHOISIES & COMPARAISONS

1. Système d'envoi pour les SMS

Concernant l'envoi des SMS, voici les solutions trouvées et celle qui a été retenue :

	smsmode	Free	LabsMobile	Clef 4G Huawei E-3531
Langage	Python	Python	C++ / PHP	PHP avec Gammu
Abonnement	Payant (20 SMS gratuit)	Inclus avec abonnement Free	Payant (SMS de test gratuit)	Carte SIM (tout opérateurs)

La solution retenue est celle de l'utilisation de la **clef 4G Huawei E-3531** : Gratuit, simple et rapide à configurer et à utiliser.

Matériel :

- RaspberryPi 3 ;
- Clef 4G Huawei E-3531 ;
- Carte SIM (tout opérateurs), il faudra désactiver le code PIN de la carte pour simplifier son utilisation.

Logiciel :

- Installation et configuration de Gammu ;
- Paramétrer la clef 4G en tant que modem ;
- Exécution d'un code PHP avec « echo » pour l'envoi des SMS.

Cette méthode conviendra parfaitement pour le système d'envoi des SMS de notre projet car : pas d'hébergement, forfait SMS au choix de l'utilisateur, simple à paramétrer et à utiliser.

2. Système d'envoi pour les E-mail

Concernant l'envoi des E-mail, voici les solutions trouvées et celle qui a été retenue :

	Fonction mail() de PHP	IFTTT	PostFix
Langage	PHP avec Sendmail	Création d'une boucle « Applet »	PHP ("echo mail … »)
Serveur	WampServer	Site web	RaspberryPi

La solution retenue est celle de l'utilisation du serveur de messagerie **PostFix** : Gratuit, simple et rapide à configurer et à utiliser.

<u>Matériel :</u> Aucun

<u>Logiciel :</u>

- Installation du serveur de messagerie PostFix ;
- Configuration des ports SMTP ;
- Création d'une adresse mail « gmail.com » ;
- Exécution d'un code en PHP avec « echo » pour l'envoi des E-mail.

Cette solution conviendra parfaitement pour le système d'envoi des E-mail de notre projet car : serveur de messagerie intégrée au RaspberryPi, personnalisation des messages, simple à paramétrer et à utiliser.

3. Propulsion/hébergement du site Web

Concernant l'hébergement du site Web et des données, voici les solutions trouvées et celle qui a été retenue :

	Apache	NGINX
Architecture	Axée sur les processus	Accès sur les évènements
Vitesse de traitement des données (statiques)	++	++++
Vitesse de traitement des données (dynamique)	++++	++++
Support de l'OS	Unix (tout) + Windows (tout)	Unix (tout) + Windows (partiellement)
Simplicité	++	+++
Flexibilité	++++	++
Sécurité	+++	++++

La solution retenue est celle de l'utilisation de du serveur **Apache**, il s'agit d'une solution gratuite et populaire, elle dispose donc d'une large communauté et d'informations autours d'elle.

De plus, Apache est un service maitrisé par notre équipe et par moi-même grâce notamment aux nombreux TP réalisé en cours. Cela va nous permettre de gagner du temps sur son installation et son utilisation.

Cette solution conviendra parfaitement pour l'hébergement de notre site Web et de nos données car : gratuit et simple à paramétrer et à utiliser.

4. Stockage des données

Commenté [Loan Pese4]: « Fichier plat » ?

Concernant le stockage des données, voici les solutions de base de données trouvées et celle qui a été retenue :

	MySQL	MariaDB	Oracle	PostgreSQL
Documentation	++	+++	+++	++
Difficulté	+	+	+++	++
Popularité	2e	13e	1er	4eme
Coût	Gratuit	Payant	Payant	Gratuit

La solution retenue est celle de l'utilisation de MySQL, il s'agit de la solution gratuite la plus utilisée et populaire et dispose donc d'une large communauté et d'informations autours d'elle.
Cette base de données est aussi celle qui est la plus maitrisée par notre équipe grâce notamment aux nombreux TP réalisés en cours, le temps de travail sera donc pleinement dédié au développement du service.

Cette solution conviendra parfaitement pour le stockage des valeurs capteurs et seuils de notre projet car : gratuit et simple à paramétrer et à utiliser.

5. Langages utilisés

HTML :

- Mettre en forme la page internet ;
- Mettre en forme le formulaire ;
- Afficher les parties textes.

JS :

- Animer les valeurs des sliders ;
- Griser les cases en fonction des checkbox ;
- Afficher les alertes de confirmation de modification ;
- Lancer des fonctions au chargement des pages.

CSS :

- Définir les tailles, la police et les couleurs des différents textes ;
- Définir les positions, les tailles et les couleurs des différents éléments (images, fieldset, div).

PHP :

- Établir la connexion à la base de données ;
- Envoyer les valeurs du formulaire à la base de données ;
- Récupérer les valeurs de la base de données ;
- Utiliser les valeurs de la base de données ;
- Comparer les valeurs de la base de données ;
- Exécuter les codes d'envois des SMS et E-mail.

BASH :

- Configuration de la clef 4G Huawei.

6. Hébergement des données

Concernant le nano ordinateur, voici les solutions trouvées et celle qui a été retenue :

	RaspberryPi 4	RaspberryPi 3 B+	RaspberryPi Zéro
Processeur	++++	++++	++
Mémoire	++++	+++	+
Ethernet	Oui	Oui	Non
WI-FI	Oui	Oui	Non
Bluetooth	Oui	Oui	Non
Coût	++++	++	+

La solution retenue est celle de l'utilisation du **RaspberryPi 3 B+** : il embarque de nombreuses fonctions avec des performances correctes et est à un prix raisonnable.

<u>Caractéristiques :</u>

- Processeur 64 bits à 1.4 GHz ;
- 1 Go de mémoire SDRAM ;
- Wi-Fi 2.4 GHz et 5 GHz, Bluetooth 4.2 ;
- Gigabit Ethernet (300 Mbps en débit maximale) ;
- 4 ports USB 2.0.

Ce nano ordinateur conviendra parfaitement pour accueillir la base de données et les scripts du projet. De plus, le RaspberryPi 3 B+ est le model qui a été utilisé pour s'entrainer par l'ensemble de notre équipe durant notre formation.

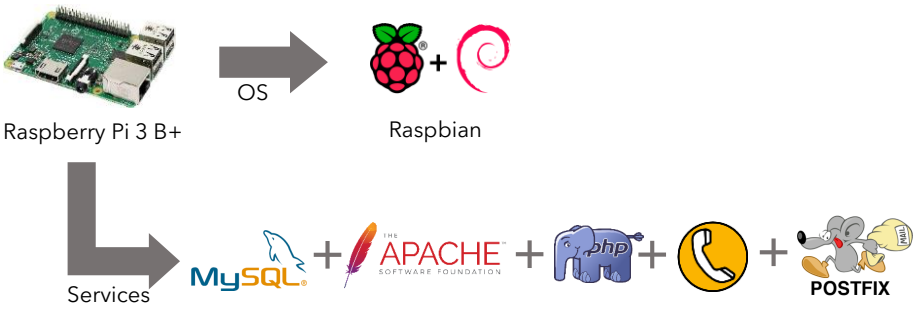
Nous pourrons facilement, grâce à des commandes dans le terminal, y installer et configurer tous les services dont nous allons avoir besoin pour la réalisation de notre projet.

RÉALISATION & CONFIGURATION

1. Présentation du matériel

Voici une liste générale du matériel et de leurs configurations que j'ai pu utiliser pour mener à bien la réalisation de ma partie au sein de ce projet :

Nano ordinateur :



SMS :

Commenté [Loan Pese5]: PRÉCISER le numéro de série, model

2. Réalisation du système de notification par SMS

Dans cette partie figure la configuration nécessaire pour envoyer des SMS avec Gammu à partir de notre RaspberryPi.

Après ça, nous devrons être capable d'envoyer des SMS depuis une ligne de commande exécutée dans le Shell du RaspberryPi.

Matériel :

- RaspberryPi 3 B+
- Clef 4G Huawei E-3531
- Un abonnement et une carte SIM (il ne doit pas y avoir de mot de passe sur la carte SIM).

Configuration de la clef 4G

La clef 4G ne permet pas l'envoi de SMS à partir du Raspberry Pi directement, c'est pourquoi la clef doit être configurée manuellement pour être reconnu en tant que modem et pour pouvoir envoyer des SMS en commande AT (commande sur une seule ligne).

Voici le code provenant de la documentation technique du constructeur :

```
sudo usb_modeswitch -v 12d1 -p 157d -M
'55534243123456780000000000000011062000000100000000000000000000'
```

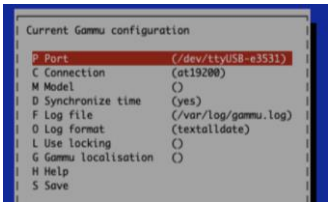
"sudo" permet de lancer les commandes en tant que super utilisateur (root).

La commande sera exécutée au démarrage du RaspberryPi pour éviter tout bug.

```
pi@raspberrypi:~ $ lsusb
Bus 001 Device 010: ID 12d1:1001 Huawei Technologies Co., Ltd. E161/E169/E620/E800 HSDPA Modem
Bus 001 Device 006: ID 0424:ec00 Standard Microsystems Corp. SMSC9512/9514 Fast Ethernet Adapter
Bus 001 Device 005: ID 0424:9514 Standard Microsystems Corp. SMC9514 Hub
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

Gammu

Gammu permet d'envoyer et de recevoir des SMS, il peut fonctionner sous forme de daemon ou de programme à la demande (émission uniquement).



Durant l'installation ont défini :

- Le chemin du port de montage de la clef 4G ;

- Le type de connexion ;

- Le fichier log ;

- La verbosité du fichier log.

Test d'envoi du premier SMS

La commande ci-dessous permet l'envoi d'un SMS :

```
gammu -c /etc/gammurc sendsms TEXT 0612345678 -text "test"
```

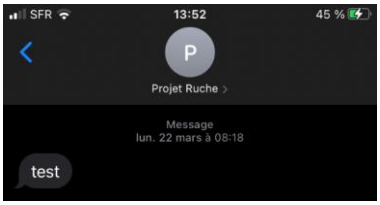
gammu – c /etc/gammurc : Va chercher le fichier de configuration pour obtenir les informations de la clef 4G pour l'utiliser.

sendsms TEXT : permet d'indiquer que l'on veut envoyer un SMS contenant du texte.

0612345678 : est le numéro du destinataire qui va recevoir le message.

-text "test" : Défini le corps du message qui est ici « test ».

Résultat :



Conclusion

Voici un exemple du code en PHP utilisé pour envoyer une notification via SMS :

```
shell_exec ("gammu -c /etc/gammurc sendsms TEXT ".$GLOBALS['numeroTelephone']." -text 'Attention !
    Le ".$GLOBALS['date']." à ".$GLOBALS['heure']." humidité minimale de la [RUCHE 1] a été dépassée !'");
```

Ici l'exemple est une alerte concernant l'humidité de la ruche1.

La base restera la même pour toutes les notifications via SMS, uniquement le corps du message variera.

3. Réalisation du système de notification par E-mail

Dans cette partie figure la configuration nécessaire pour envoyer des E-mail avec l'utilisation du serveur de messagerie PostFix à partir de notre RaspberryPi.

Après cette installation nous devrons être capable d'envoyer des E-mail depuis une ligne de commande exécuté dans le Shell du RaspberryPi vers une adresse mail définie.

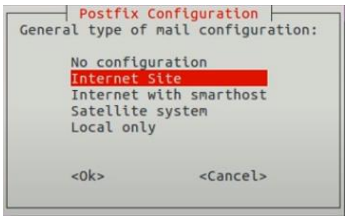
Matériel :

- RaspberryPi 3 B+

POSTFIX

Après avoir installé SMTP server (protocole de communication utilisé pour transférer des courriers électroniques vers les serveurs de messagerie électronique) on installe PostFix (serveur de messagerie électronique, il se charge de la livraison de courriers électroniques).

On configure PostFix pour une utilisation "Internet Site". Pourquoi ?



On finit par l'installation des paquets requis et le certificat ssl.

Configuration

Édition du fichier de configuration /etc/postfix/main.cf : Préciser les valeurs ci-dessous et pourquoi ?

relayhost = ?

smtp_sasl_auth_enable = ?

smtp_sasl_password_maps = ?

smtp_sasl_security_options = ?

BESOIN DU RASPBERRY PI EN CLASSE

Test d'envoi du premier E-mail

La commande ci-dessous permet l'envoi d'un E-mail :

```
echo "test" | mail -s 'Sujet' mail exemple@gmail.com
```

"test" : Défini le corps du message qui est ici « test ».

mail -s : Permet d'appeler la fonction d'envoi du mail.

'Sujet' : Défini le sujet du mail.

mail exemple@gmail.com : Défini l'adresse de destination du mail.

Capture d'écran

Conclusion

Voici un exemple du code en PHP utilisé pour envoyer une notification via E-mail :

```
shell_exec("echo 'Attention ! Le ".$GLOBALS['date']." à ".$GLOBALS['heure']." la température
    minimale de la [RUCHE 1] à été dépassée !' | mail -s 'Ruche Connectée' mail ".$GLOBALS['mail']."");
```

Ici l'exemple est une alerte concernant la température de la ruche1.

La base restera la même pour toutes les notifications via E-mail, uniquement le corps du message variera.

4. Hébergement du site Web et stockage des données

Dans cette partie figure les trois systèmes choisis concernant l'hébergement du site Web et le stockage des données. Chacun ayant des rôles différents :

APACHE

Dans notre configuration Apache est le serveur Web. Son rôle est d'écouter les requêtes émises par les navigateurs (qui demandent des pages web), de chercher la page demandée et de la renvoyer.

PHP

PHP est un langage de script. Il permet, de décrire dans une page web, un affichage dynamique d'information, c'est-à-dire que le texte affiché peut dépendre de variables.

Les instructions PHP sont généralement contenues dans des fichiers d'extension php (exemple.php). Ces fichiers peuvent contenir du HTML, entremêlé avec le code PHP.

Quand un navigateur demande un tel fichier, le serveur Apache exécute les instructions PHP, qui produisent une page HTML. Une fois la page HTML générée, le serveur la renvoie au navigateur, qui ne voit qu'une page HTML.

MYSQL

MySQL est un système de gestion de base de données. Son rôle est de stocker les données, sous forme de tables, et de permettre la manipulation de ces données à travers le langage de requête SQL.

Exemple de requêtes SQL utilisées dans notre projet :

```
"UPDATE seuils_ruches SET temp_min = '$tempMinRuche1' WHERE id_ruche = 1";
```

Exemple d'une ligne de commandes utilisée lors de la création d'une table :

```
CREATE TABLE stations (id_station INT NOT NULL AUTO_INCREMENT primary key,
temp_min INT NOT NULL, temp_max INT NOT NULL, humi_min INT NOT NULL, ...);
```

Conclusion

L'utilisation de ces trois logiciels est nécessaire lorsqu'on veut afficher à l'utilisateur des informations provenant d'une base de données. Cela va m'être très utile pour la réalisation de ma page internet "Notifications".

5. Interface Home Machine

Dans cette partie figure la partie IHM (Interface Homme Machine), c'est-à-dire la partie site Web et plus particulièrement les différentes parties de la page de configuration des alertes.

Nom du paramètre de la section + checkbox (Activé ou Désactivé)

Sélection du système de notification (SMS & E-mail ou SMS ou E-mail)

Sliders de sélection pour les valeurs (minimum, maximum, vitesse et pourcentage)

Bouton permettant la remise par défaut des valeurs des sliders

Bouton de validation des paramètres

Description du CSS

Ici figurent une partie des différentes zones de mon code **style.css**, ce code permet de mettre en forme de manière visuelle les différents éléments de ma page internet **notifications.php** (la totalité de mon code **style.css** disponible en annexe).

Les zones principales :

```
body{
    font-family: monospace;
    font-size: 15px;
}
```

Police et taille de police de toute la page notifications

```
fieldset{
    position: relative;
    float: left;
    width: 420px;
    border-radius: 20px;
    border: 3px outset;
}
```

Position, taille, bordures des 3 grandes parties.

```
div{
    background-color: lightgrey;
    width: 200px;
}
```

Couleur et taille des parties contenant les sliders

Les classes :

```
.iconeResetS{
    width: 150px;
    padding-top: 8px;
    padding-left: 3px;
}
```

Taille et bordures des icônes Reset

```
.titreCategorie{
    font-size: 25px;
    font-weight: bold;
}
```

Taille et style des titres de catégories

```
.trait{
    background-color: black;
    height: 0.5pt;
    width: 100%;
}
```

Trait de séparation entre les catégories

```
.tempMinCss{
    position: relative;
    float: left;
    right: 20px;
    left: 5px;
}
```

Positions et bordures parties Température minimum

Description du PHP

Le langage PHP est présent dans pratiquement tous les codes qui m'ont permis de réaliser ma partie. Sans PHP il me serait impossible d'envoyer des notifications avec les valeurs de la base de données MySQL.

connexion_serveur.php → Permet la connexion à la base de données :

```
$link = mysqli_connect($servername, $username, $password, $database);
```

notifications.php → Permet d'appeler et d'utiliser les valeurs de la base de données :

```
$requeteRuche1 ="SELECT * FROM seuils_ruches WHERE id_ruche = 1";
$resultatRuche1 = $link -> query($requeteRuche1);
$valeursRuche1 = $resultatRuche1 -> fetch_array(MYSQLI_ASSOC);
```

```
<option value="0" <?php echo $notifRuche1TempSelect0; ?> >SMS & E-mail</option>
```

traitement.php → Permet la récupération des valeurs du formulaire pour les envoyer à la base de données :

```
$checkboxTempRuche1 = (isset($_POST['choixTempRuche1'])) ? 1 : 0;
```

```
$requeteRuche1 = "UPDATE seuils_ruches SET temp_min = '$sliderTempMinRuche1', ... WHERE id_ruche = 1";
```

comparaison.php → Permet de récupérer, mettre en variable les valeurs et, avec des "**if**" et "**else**", de les comparer envoyer les notifications :

```
//STATION MÉTÉO
$requeteSeuilStation = "SELECT * FROM seuils_stations WHERE id_station = 1";
$resultatSeuilStation = $link -> query($requeteSeuilStation);
$valeursSeuilStation = $resultatSeuilStation -> fetch_array(MYSQLI_ASSOC);
```

```
$seuilTempMinStation = ($valeursSeuilStation['temp_min']);
```

```
if ($cocheTempStation == 1) {
    // [SMS & E-MAIL]
    if ($notifTempStation == 0) {
        if ($seuilTempMinStation < $capteurTempStation){
            echo '';
        }

        else{
            // sms
            shell_exec ("gammu -c /etc/gammurc sendsms TEXT ".$GLOBALS['numeroTelephone']." -text 'Attention ! ...'");
```

contentChecker.php → Permet de ne pas afficher d'erreur si une variable est vide après un POST :

```
function contentChecker ($toTest) {
    if (!isset($_POST[$toTest])) {
        $toReturn = "";
    }
    else {
        $toReturn = $_POST[$toTest];
    }
    return $toReturn;
}
```

Description du JavaScript

Le langage JavaScript dans mon code **main.js** m'a beaucoup servi pour animer les différentes parties de la page internet **notifications.php**.

```
function changeTempRuche1(){
    if (document.getElementById('choixTempRuche1').checked)
        document.getElementById('sliderTempMinRuche1').disabled=false;

    else
        document.getElementById('sliderTempMinRuche1').disabled=true;
```

Permet de griser les différentes catégories en fonction de la position des checkbox

```
var slider1 = document.getElementById("sliderTempMinRuche1");
var output1 = document.getElementById("valeurTempMinRuche1");

output1.innerHTML = slider1.value;

slider1.oninput = function(){
    output1.innerHTML = this.value;
}
```

Permet d'afficher les valeurs des sliders en direct durant de choix des seuils.

J'ai pu aussi utiliser ce langage pour créer des fonctions de redirection :

```
function chargementPagePrincipale(){
    document.location.href="notifications.php";
}
```

```
<body onload="chargementPagePrincipale()">
```

6. Schéma du réseau

Dans cette partie figure la configuration réseau en rapport avec ma partie concernant le Raspberry Pi :

- Adresse IP → ?
- Masque de sous-réseau → ?

7. Recette des fonctions réalisées

Ci-dessous figure la liste du fonctionnement des différentes codes qui m'ont permis l'élaboration du système de notifications.

● = Description ✓ = Utilisation

connexion_serveur.php :

- Permet la connexion à la base de données MySQL afin de pouvoir modifier les tables et récupérer leurs valeurs.
- ✓ Appelé avec "include" dans les codes nécessitant une connexion à la base de données.

style.css

- Permet de mettre en forme de manière visuelle les différents éléments de la page internet **notifications.php**.
- ✓ Utilisation de classes (class =" …") ou zones principales (body, div, fieldset).

main.js

- Permet d'animer les différentes parties de la page internet **notifications.php** en utilisant différentes fonctions. Permet également le chargement de pages à l'aide de fonctions.
- ✓ Si une checkbox est décochée, alors la partie concernée sera grisé est non utilisable.
- ✓ Les valeurs effectives aux sliders s'affichent en direct sur la page **notification.php.**
- ✓ Appelé avec src="…" à la fin des codes nécessitant l'utilisation de fonctions.

notifications.php

- Page internet utilisée par l'utilisateur pour configurer les alertes. Comprend les formulaires de la ruche 1, ruche 2 (non utilisée, figure comme exemple), station météo et des notifications journalières.
- ✓ Activation ou non des zones avec les checkbox, choix du système de notification avec les menus déroulant, choix des seuils d'alerte avec les sliders et boutons pour utiliser les valeurs par défaut conseillées.
- ✓ Permet aussi, grâce à la récupération des valeurs, de consulter les seuils déjà utilisées.

traitement.php

- Permet l'envoi des données du formulaire de **notifications.php** vers les tables de la base de données **rucheconnectee1**.
- ✓ Les valeurs du formulaire sont récupérées et mise en variables pour ensuite être envoyé à la base de données à l'aide de la commande SQL "UPDATE".

comparaison.php

- Permet la récupération et la comparaison des valeurs des tables pour notifier l'utilisateur en fonction des paramètres qu'il aura choisi sur la page **notifications.php** dans les catégories "Ruche 1" et "Station Météo".
- ✓ Les valeurs des capteurs, des checkbox, des menus déroulants et des sliders sont récupérées et analysés à l'aide de "if" et de "else" afin d'envoyer les messages d'alerte correspondant aux valeurs récupérées.
- ✓ Exemple d'un message d'alerte : Attention ! Le 21 décembre à 21h00 la température minimale de la [RUCHE 1] a été dépassée !
- ✓ Ce code a été ajouté au fichier crontab -e du RaspberryPi afin d'être exécuté toutes les minutes.

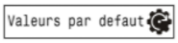
comparaisonJournaliere.php

- Permet la récupération et la comparaison des valeurs des tables pour notifier l'utilisateur en fonction des paramètres qu'il aura choisi sur la page **notifications.php** dans la catégorie "Notifications Journalières".
- ✓ Les valeurs des capteurs, des checkbox et des menus déroulants sont récupérées et analysées à l'aide de "if" et de "else" afin d'envoyer les messages récapitulant les valeurs Journalière atteinte par les différents capteurs durant la journée.

contentChecker.php

- Permet de ne pas afficher d'erreur si une variable est vide après un POST grâce à la variable contentChecker.
- ✓ Lorsque l'on désactive une zone (Température de la Ruche 1 par exemple) les valeurs récupérées sont inexistantes et la page retourne une erreur.
- ✓ Grâce à ce code, si une variable est inexistante à la validation du formulaire elle sera remplacée par une valeur "vide".
- ✓ Ce code est appelé lorsqu'une valeur est mise en variable dans le code **traitement.php**.

defautTempRuche1.php (et autres)

- Lorsque l'utilisateur appui sur ce bouton 

, la zone correspondante à ce dernier affichera les valeurs par défaut conseillées et les enverra à la base de données.
- ✓ Ce bouton sera utilisé par l'utilisateur lorsqu'il souhaitera remettre des valeurs par défaut.
- ✓ Ces valeurs ont été choisi en fonction de mes recherches afin de proposer à l'utilisateur des valeurs "sures" pour garder en bonne santé ses abeilles.

CHAMP ÉLECTROMAGNIÉTIQUE ET PRINCIPE DE TRANSMISSION

1. Mesure du champ électromagnétique lors de l'émission d'un SMS

2. Principe de transmission et de modulation.