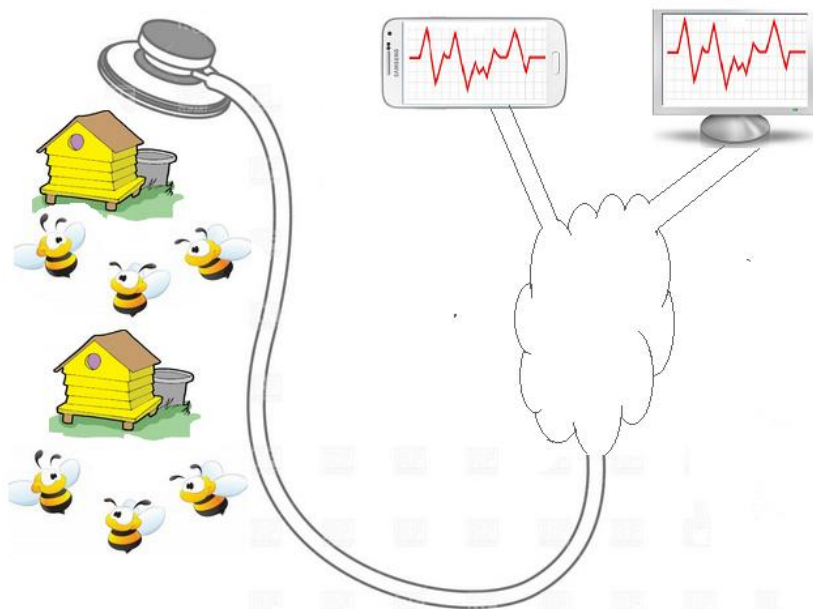




Institut TIC – Filière télécom
Projet de semestre - printemps 2014

Beemon II

Monitoring de ruche



Rapport

*Ecole d'ingénieurs et d'architectes de Fribourg :
Boulevard de Pérolles 80 - 1705 Fribourg
Section : Télécommunications
Etudiants : Gerstein Liam T3
Mandataire : Francis Saucy
Responsables : Buntschu François / Gachet Daniel*




 EIA-FR	Projet de semestre 6 Monitoring de ruches 2	Cahier des charges Version 3
---	--	---------------------------------

Table des matières

1	Introduction	5
2	Structure du rapport	6
3	Contexte	7
4	Objectifs et contraintes	9
5	Analyse	10
5.1	<i>Bluetooth Low Energy</i>	10
5.1.1	Fonctionnement général	10
5.1.2	Pile de protocoles BLE	17
5.1.3	GAP	18
5.1.4	Présentation du BLE112	19
5.2	Besoins	20
5.2.1	Cas d'utilisation	20
5.2.2	Moyens de supervision	22
5.3	Interfaces	31
5.3.1	Interfaces possibles	31
5.3.2	Analyse critérielle des possibilités	31
5.3.3	Interfaces nécessaires	32
5.3.4	Offre en interfaces du BLE112	33
5.4	Architectures matérielles	34
5.5	Architectures communications	38
5.6	Composants du SSI	41
5.7	Commande de matériel	42
6	Spécification du SSI	43
6.1	Utilisation	43
6.2	Systèmes BEEMONS	45
6.3	Fonctionnement	46
7	Conception	49
7.1	Architecture matérielle	49
7.2	Communication	51
7.2.1	Chemins dans le système d'informations Beemon	51
7.3	Software	52

	Projet de semestre 6 Monitoring de ruches 2	Cahier des charges Version 3
---	--	---------------------------------

7.3.1	Bases de l'application	52
7.3.2	Mise en service et configuration.....	57
7.3.3	Collecte et dépôt/alerte	61
8	Réalisation prototype fonctionnel.....	67
8.1	Matériel	67
8.2	Environnement de développement.....	67
8.3	Programmation du BLE112	68
8.4	Débogage du BLE112	68
8.5	Software	69
8.5.1	Base de l'application.....	69
8.5.2	Mise en service et configuration.....	75
8.5.3	Collecte et dépôt	76
9	Test et validation	81
10	Conclusion	83
10.1	Problèmes rencontrés	83
10.2	Perspectives	86
10.3	Conclusion	87
11	Tableau des figures.....	Erreur ! Signet non défini.
12	Glossaire.....	90
13	Bibliographie	Erreur ! Signet non défini.
14	Référence	88
15	Contenu du CD-ROM	92
16	Bibliographie	Erreur ! Signet non défini.

 EIA-FR	Projet de semestre 6 Monitoring de ruches 2	Cahier des charges Version 3
---	--	---------------------------------

Historique des versions			
Version	Date	Modifications	Qui
V1	11.3.14	-Création document -Création du plan d'analyse et rédaction de celle-ci	Gerstein Liam
V2	27.3.14	-Refonte du plan d'analyse -Mise des tableaux « les valeurs à surveillées » dans l'annexe -Mise de la réflexion sur les problèmes dans l'annexe	Gerstein Liam
V3	1.4.14	-Changement dans l'agencement des chapitres consacrés aux interfaces. -Création des chapitres sur le BLE et l'architectures de communications	Gerstein Liam
V4	13.4.14	-Nouveau changement dans l'agencement des chapitres -Création du squelette de la conception, réalisation. -Finalisation du document	Gerstein Liam

1 Introduction

La surmortalité des abeilles est un danger pour l'équilibre de la faune et de la flore mondiale. En effet, elles représentent environ neuf pollinisateurs sur 10 en Europe. Leur disparition engendrerait de forts déséquilibres dans notre écosystème. Bien que plusieurs causes soient probablement responsables de ce phénomène, il est difficile de comprendre leur interaction avec le monde de l'apiculture.

Ce projet est la continuité de celui entamé le semestre passé. Il a pour but principal de réaliser un système de surveillance de ruche à distance.

Il est destiné principalement aux apiculteurs. Une surveillance à distance leur permettrait de gagner du temps en évitant de devoir se déplacer inutilement, de mieux organiser leur calendrier de production, et finalement de pouvoir réagir rapidement en cas de problème. Il doit également être un outil de recherche scientifique afin de mieux comprendre le phénomène d'effondrement des colonies.

Il doit donc offrir un spectre de fonctionnalités minimal pour satisfaire les besoins de l'apiculteur mais être également le plus modulable et extensible possible pour réagir aux besoins, parfois nouveaux, de la recherche scientifique.

Le schéma ci-dessous illustre une vue d'ensemble du système. Il est décomposé en trois parties principales.

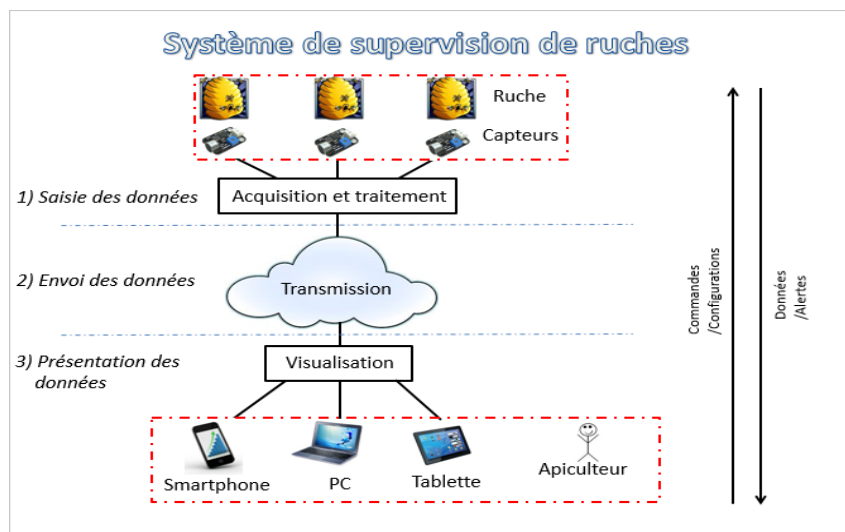


Figure 1: Contexte système

Le premier projet s'est concentré essentiellement sur la partie centrale, celle qui lie l'exploitation à l'utilisateur. Il en résulte que le **GSM** est le plus approprié et intéressant. Il couvre la quasi-totalité de

la Suisse et offre un accès simple à internet. Cependant il est suspecté de désorienter les abeilles, qui s'égarer et meurent.

Il a également permis de définir dans les grandes lignes l'architecture et la technologie de communication pour l'ensemble des systèmes de la partie « **saisie de données** ». Ce second projet a pour but principal de réaliser un système qui collecte des mesures via différents capteurs et transmet ces dernières à un autre système via **Bluetooth® Low energy** (BLE).

2 Structure du rapport

Le rapport présente la structure générale suivante.

Analyse

- Général du fonctionnement du Bluetooth Low Energy, ainsi que des possibilités offertes par le module BLE112
- Une analyse sur les différents services que l'ensemble du système peut proposer
- Des besoins matériels pour la récolte de données provenant d'une ruche ainsi que des moyens d'interaction.
- D'une étude sur les interfaces possibles pour collecter ces valeurs et proposer des interactions sur la ruche
- Une étude l'architecture matérielle du SSI pour déterminer quel forme prendra la carte
- Architecture hardware et software pour le SSI et le SSC pour savoir comment vont communiquer le SSI et le SSC
- Une décision sur l'implémentation du SSI.

Spécification

- Des cas d'utilisations
- Le comportement général du SSI-SSC

Conception

- Des concepts de base qui vont être appliqués.
- Présenter de la façon dont on va les utiliser pour construire notre application

Réalisation

- Explication sur la réalisation des modules élémentaires.
- Explication de la construction de l'application

Test et validation

Test général des modules et fonctionnalités

Conclusion

Explication des problèmes rencontrés ainsi que les perspectives possible pour le projet. Et enfin une conclusion personnel

3 Contexte

Le premier projet a permis de définir une architecture pour les systèmes matériels ayant pour tâche la surveillance de l'exploitation. Cette architecture tient compte des différentes dispositions de ruche existantes ainsi que de la forme générale d'une grande exploitation apicole pour apporter une solution pratique, modulable et extensible. Ainsi elle répond aux besoins des amateurs et des professionnels. Ci-dessous le **premier jet** de cette architecture.

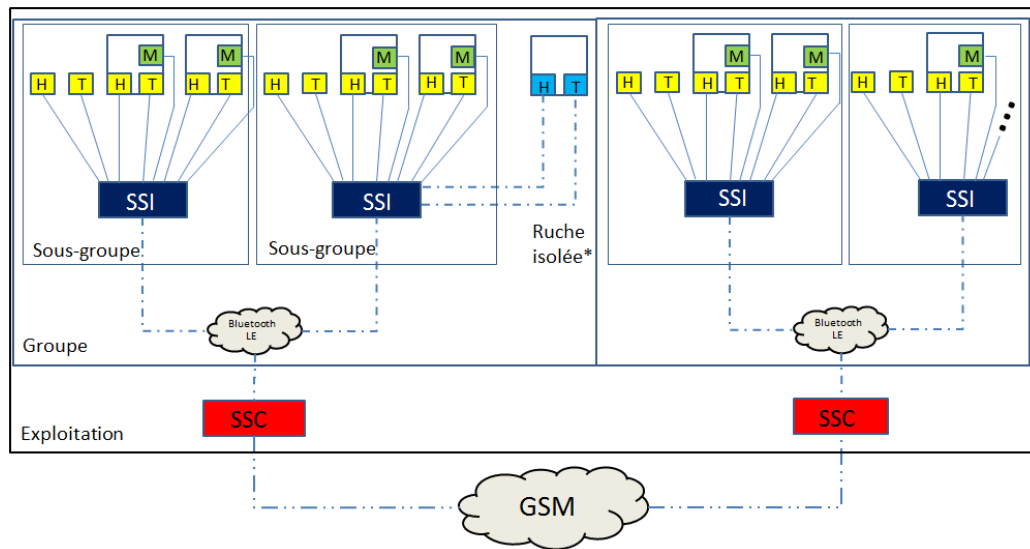


Figure 2: Architecture des systèmes matériels de supervision

Cette illustration contient l'emplacement de nos systèmes ainsi que les technologies de communications qui seront utilisées pour les relier.

SSI : Système de supervision intermédiaire → S'occupe d'une ruche (isolée) ou de plusieurs ruches (sous-groupe).

SSC : Système de supervision central → S'occupe d'un groupe de ruches, qui lui-même contient des sous-groupes.

Carrés jaunes et bleus : Capteur filaire et capteur bluetooth individuel (CBI)

Carrés verts : moteur, chauffage

Cette architecture en tant que telle est susceptible de changer encore en cours d'analyse. Cependant il est possible de définir les rôles minimums suivants :

SSI : Collecte des valeurs sur une ou plusieurs ruches, stocke les données temporairement avant de les envoyer au SSC. Permet de piloter une porte, un chauffage. S'il y a un problème le SSI le notifie au SSC et active un processus de résolution (enclenchement du chauffage). On peut aussi envisager d'utiliser un SSI comme relais BLE à un autre SSI qui est hors de portée du SSC.

SSC : Stocke toutes les valeurs et événements fournis par les SSI d'un groupe de ruches. Il peut piloter les SSI. Sert de passerelle entre l'exploitation et l'utilisateur.

*L'illustration suggère que pour une ruche isolée il est envisageable d'avoir des capteurs Bluetooth® LE individuel (CBI). Ceci n'est pour le moment absolument pas fixé.

Voici à quoi pourrait ressembler ce dispositif :

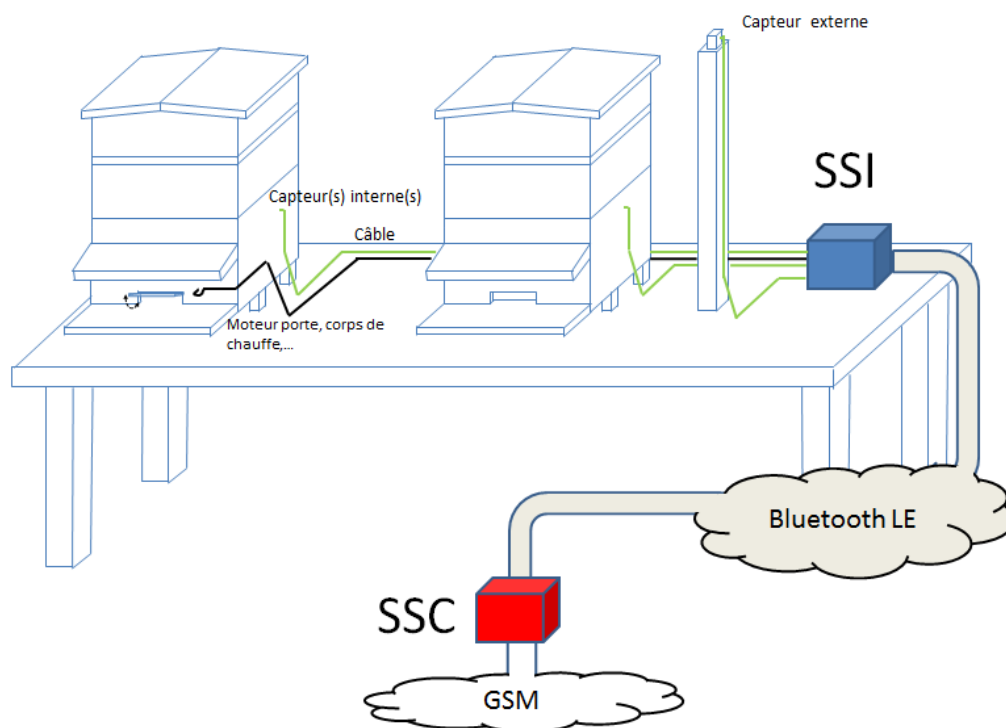



Figure 3: Vue d'artiste de l'installation

 EIA-FR	Projet de semestre 6 Monitoring de ruches 2	Cahier des charges Version 3
---	--	---------------------------------

4 Objectifs et contraintes

Contraintes

Toutes les contraintes relatives au PF et au PCI sont obligatoires et notées PF & PCI

Obligatoires (PF & PCI)

Contraintes géographiques : Dispositions, distances, distributions dans un groupe

Contraintes pratiques : Extensibilité en nombre et en type de capteurs, modularité du système, mise en service, modifications, configurations.

Optionnelles (PCI)

Contraintes électriques : Consommations électriques de chaque composant (SSC, SSI, Moteur, Chauffage)

Contraintes mécaniques : emplacement et fixation des capteurs, moteurs, câblages internes

Contraintes météorologiques : protection des systèmes contre la pluie, température.

Biocompatibilité : Neutralité des matériaux susceptibles d'être en contact direct avec les abeilles.

Objectifs

Obligatoires (PF & PCI)

Architecture hardware et software pour le système SSI : Types de capteurs, forme et composition de la carte du SSI, communication avec les périphériques, câblages et alimentations des moteurs et corps de chauffe, etc.

Spécification de l'ensemble des systèmes internes et du SSI : Cas d'utilisations, acquisition des valeurs, détection et résolution de problèmes, rôles et relations des systèmes, communication entre les systèmes, configurations.

Réalisation d'un prototype fonctionnel : à base d'une carte préfabriquée respectant les contraintes (PF & PCI) et objectifs mentionnés ci-dessus.

Optionnelles (PCI)

Réalisation d'une carte industrialisable: Conception, réalisation d'un prototype et finalement validation du schéma électronique de la carte.

Réalisation d'une maquette mécanique du dispositif : fixations et câblages des composants, forme et dimension des boîtiers de fixations.

5 Analyse

5.1 Bluetooth Low Energy

Cette partie d'analyse se repose en partie sur le travail effectué par **M.Elgueloui Mohammed et Cornaz Gabriel**¹ et pour l'autre d'information venant du Web. Elle doit permettre de cerner les grandes lignes de cette technologie pour pouvoir dans un premier temps en tenir compte dans l'analyse de l'architecture software et de fournir une bonne base de compréhension pour la conception.

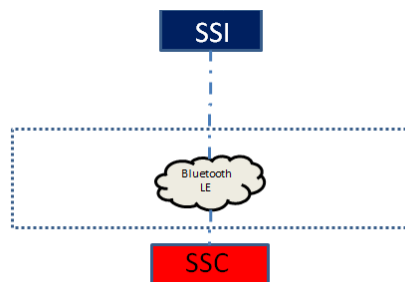


Figure 4 Etude Bluetooth LE

5.1.1 Fonctionnement général

Architecture

Le BLE définit 4 rôles principaux. Il y a encore 2 variantes qui seront expliquées plus bas. Il introduit le concept de paquet d'avertissement et du rôle d'avertisseur qui offre la possibilité aux dispositifs esclaves d'envoyer des paquets au maître sans qu'il soit connecté à lui. En principe ces paquets sont utilisés pour notifier au maître qu'un esclave souhaite établir une connexion avec lui, mais offre également le moyen de transmettre des données en mode non connecté mais sans garantie.

Advertiser : Diffuse des paquets d'advertissing mais n'est pas capable d'en recevoir (non-connecté).

Scanner : Ecoute les paquets d'advertissing envoyés par l'advertisser. Il peut essayer de se connecter à l'advertiser une fois un paquet reçu (non-connecté).

Maître : Dispositif connecté à un ou plusieurs esclaves (connecté)

Slave : Dispositif connecté au maître. Ne peut avoir qu'un maître à la fois (connecté)

Le BLE a introduit 2 nouveaux noms de rôle qui chacun regroupe respectivement advertiser/slave et scanner/master

Peripheral = Advertiser/Slave : Diffuse des paquets d'advertissing. Se connecte au maître et devient esclave

Central = Scanner/Master : Ecoute les paquets d'advertissing envoyés par un autre dispositif, initialise la connexion et sert de maître dans celle-ci

¹ M., Cornaz G. Elgueloui. *Projet de semestre 5 Bluetooth low energy*. 26 Janvier 2014. CD :
..\Documentation technique\CornazElgueloui\rapport_final.pdf P.16 à 18

Voici comment la topologie change lorsqu'un advertiseur demande au scanner d'initialiser une connexion

T1 : Un **advertiser** envoie des paquets d'**advertising** en broadcast

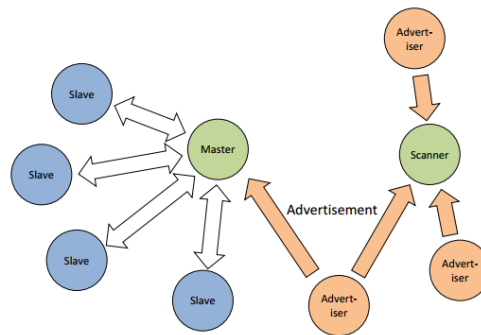


Figure 5 Topologie BLE 1²

T2 : Le **master/Scanner** qui a reçu l'advertising initie la connexion avec l'**advertisser**

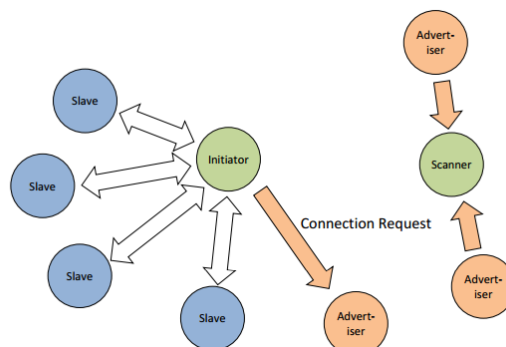


Figure 6 Topologie BLE 2

T3 : La connexion est établie et l'**advertisser** devient un **slave**

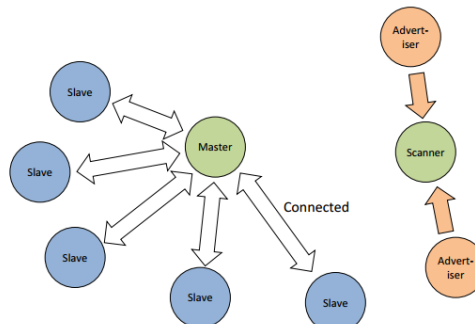


Figure 7 Topologie BLE 3

A cela s'ajoute 2 variantes, l'une découlant de l'advertiser et l'autre du scanner

Broadcaster

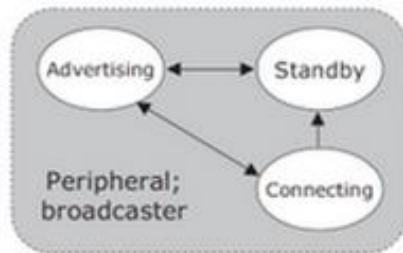


Figure 8 Broadcaster³

Un broadcaster ne fait que diffusé des paquet advertissing et ne se connecte jamais

Observer

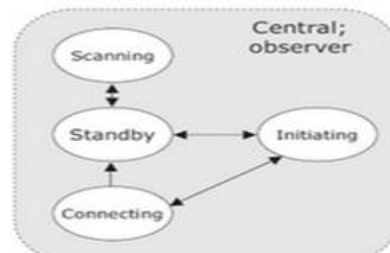


Figure 9 et Observer

Un observer ne fait qu'écoute des paquet advertissing et n'initialise jamais de connexion.

.Machine d'état d'un module BLE

La machine d'état de la **couche de liaison** est la suivante et suffit à implémenter **tous les rôles**. Nous y retrouvons nos 4 rôles BLE : Advertiser, Scanner, Master ,Slave

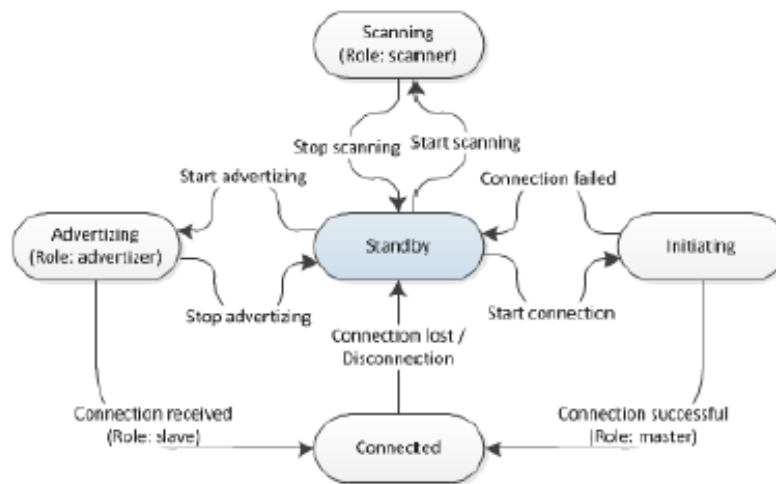


Figure 10 Machine d'état du module BLE

Ci-dessous une figure représenter les différents états possibles vis-à-vis de l'état de la connexion (connecté, non connecté).

³ csr. *Bluetooth 4.0: Low Energy*. s.d. <http://chapters.comsoc.org/vancouver/BTLER3.pdf> P.18 (accès le Avril 2014).

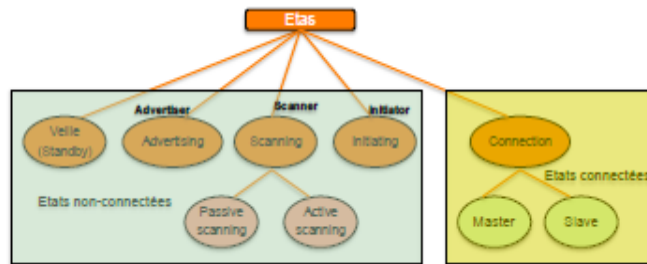


Figure 11 :Etats possibles du BLE

StandBy : Etat dans lequel le dispositif n'utilise pas le module BLE. Mais peut faire autre chose

Advertisng : Etat dans lequel le dispositif esclave envoie des paquets « advertisng » périodiquement au maître. Il y a différents modes d'advertissng.

Scanning : Etat dans lequel le maître écoute s'il y a des messages d'advertissng. Une fois détecté, il peut demander plus d'informations avant d'initialiser une connexion (**active scanning**) ou non (**passive scanning**)

Initialising : Etat dans lequel le maître initialise la connexion auprès de l'esclave. Cet état est généré lorsque qu'un message d'advertissng a été détecté dans l'état « scanning ».

Connected : Etat dans lequel le maître et l'esclave sont connectés.

Etablissement et maintien de connexion, transfert de données

On va maintenant s'intéresser au paramètre qui régit la procédure de connexion, le maintien d'une connexion et le concept introduit pour le transfert de données. Pour cela on va analyser une procédure standard. Les variantes seront exposées plus loin.

1) Etablissement de connexion

Le peripheral (en mode advertiser) envoie des paquets d'advertising au Central (en mode scanner). Ce message contient notamment l'adresse de l'advertiser. Le maître scan et détecte le message puis initialise la connexion. Une fois la connexion établie le **peripherale devient slave** et le **central devient maître** de la communication

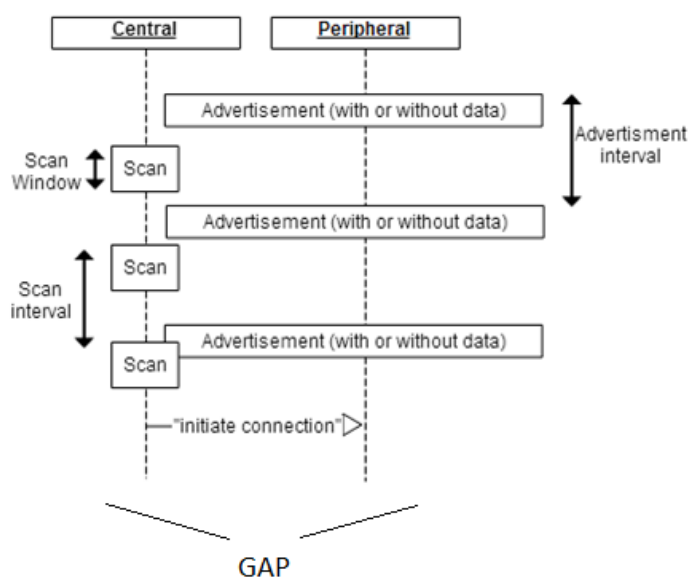


Figure 12 Scanning et advertising⁴

On peut régler les intervalles d'émission de paquets d'advertising et les intervalles des fenêtres de scan. La configuration dépend notamment des besoins de l'application, de la consommation d'énergie et du nombre de peripherals. **Plus les intervalles sont petit plus l'application est réactive mais plus elle consomme**

Interval min-max d'advertising	20 ms-10,24 s
Interval fenêtre scan min-max	40 ms-10,24s

Tableau 1 Intervalles d'advertising et fenêtres de scan possibles

⁴ <http://support.connectblue.com/>. Bluetooth Low Energy Serial Port Adapter - Getting Started. 15 Novembre 2013.

<http://support.connectblue.com/display/PRODBTSPA/Bluetooth+Low+Energy+Serial+Port+Adapter+-+Getting+Started>.

Les paquets d'advertising peuvent contenir des données d'application d'une longueur max de 31 Byte, certains sont laisser libre pour contenir un message. Il est donc possible pour un peripheral d'envoyer des données en mode non connecté.

On reviendra en détail sur les procédures possible et configuration d'intervalle dans la section GAP

Un réseau BLE peut avoir un nombre illimité de peripheral qui peuvent demander au serveur une connexion. Mais le nombre de connexions simultanée (mode connecté) **est limité**.

Le module BLE 112 qui utilise le Chipset Bluetooth Low Energy de Texas Instruments permet d'avoir 8 esclaves connectés simultanément au master. Cependant il y a une différence entre ce que Bluegiga annonce sur son site et ce qu'il y a dans la dernière version de la datasheet, qui limite à 4 le nombre de connexions simultanée.

2) Maintien de connexion

Il est possible pour un maitre et un esclave de rester connecté même s'il y a pas de données à transfert et en ne dépensant que très peu d'énergie.

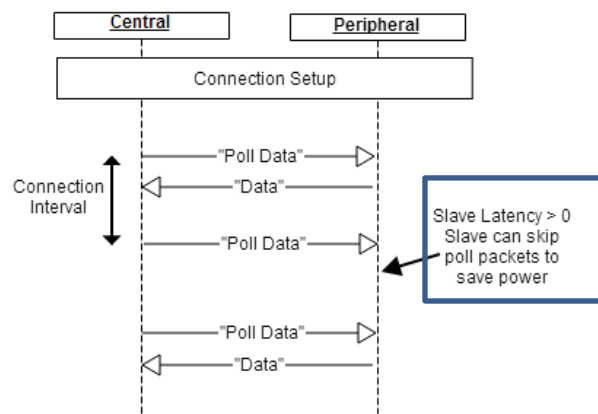


Figure 13 Maintien de la connexion

Pour maintenir la connexion le master envoie régulièrement des paquet vide « poll data » pour savoir si l'esclave a des données à transmettre. Quand l'esclave reçoit un « poll data » cela déclenche un événement appelé « **événement de connexion** ».

L'esclave peut être configuré pour ignorer un certain nombre de ces événements et ainsi économie de l'énergie. Le nombre d'événement qu'il peut ignorer est définie par le paramètre « slave latency ».

Interval de connexion min-max	7.5 ms-4s
Nombre d'événement ignorable (slave latency)	Entre 0 et 500
Time_out de la communication s'il y a pas d'activié	Entre 100 ms et 32s

Tableau 2 : Paramètres de maintien d'une connexion

3) Transfert de données

Chaque dispositif Bluetooth LE dispose d'une base de données nommées GATT. Elle définit une structure pour les données et des méthodes y donnant accès en local ou à distance. Le transfert de données se fait via le protocole ATT qui permet d'écrire et de lire dans la base de données distante.

Ce qu'il faut retenir c'est que les 2 dispositifs qui sont connectés sont les 2 à la fois client et server Gatt.

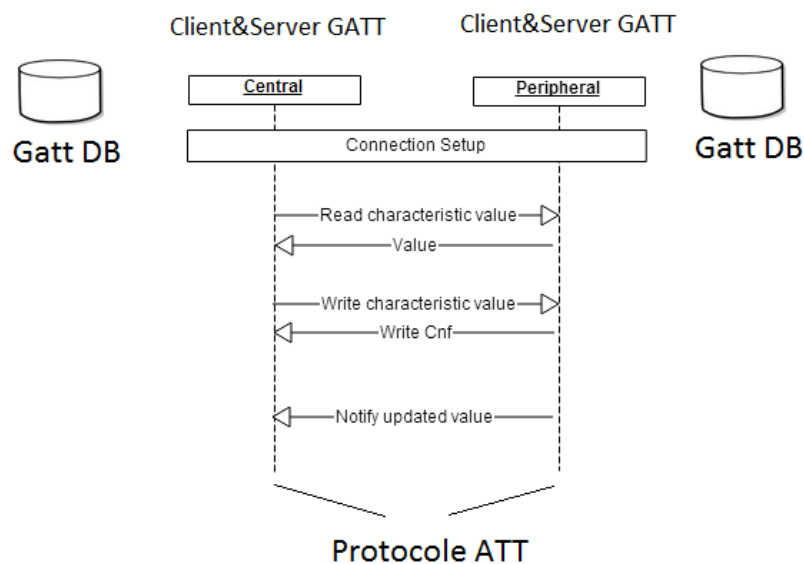


Figure 14 Fonctionnement du transfert de données dans BLE

Les actions suivantes peuvent être utilisées pour transférer des données

Read/Write: Lire/écrire une valeur dans la base de données distantes

Notifiy : Permet au server GATT de « pusher » des données dans le client GATT. Les notifications ne sont pas accusé.

Indicate : Même chose que notifiy, mais il y a un accusé.

La Gatt est stocké dans la mémoire flash du MCU. La valeur max d'un attribut est de 255 Byte. Le nombre d'attribut est limité par la mémoire flash.

Les actions définies ci-dessous sont régis par les droit qu'on met sur les attribut. Ses droits porte le même nom que les actions

5.1.2 Pile de protocoles BLE

On a ci-dessous la pile de protocole du Bluetooth Low Energy.

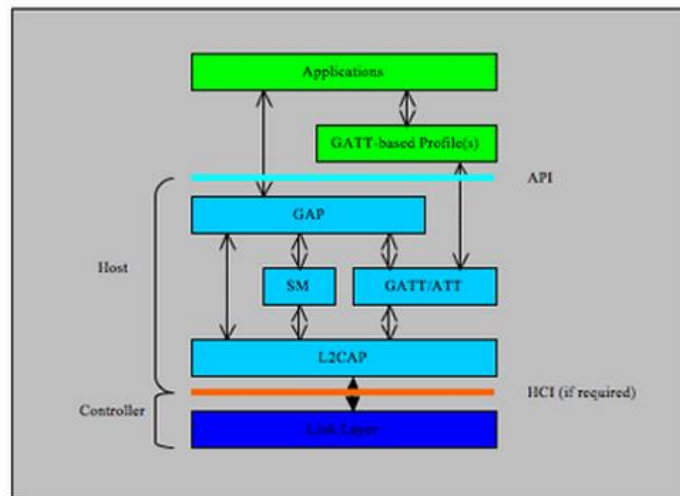


Figure 15 Pile de protocole BLE⁵

Pour bien comprendre le fonctionnement du BLE il faut s'intéresser principalement au **GAP** et **GATT/ATT**

GAP : Generic Access Profile : Définie principalement les procédures pour la découvertes d'appareils (voisins) et pour l'**établissement de communication**. Il définit les 4 rôles possibles pour les dispositifs BLE: Central et peripheral (maitre-esclave), Observer et broadcaster

GATT : Generic Attribut : Définie l'organisation des données et consiste en une base de donnée accessible en local ou à distance, **appelée GATT**. La **communication** entre 2 dispositifs BLE se fait en écrivant ou en lisant **les attributs** de la base de données GATT distante. Un dispositif est donc à la fois un serveur et client GATT car il va lire et écrire dans la base donnée GATT de l'autre dispositif mais lui donne également accès à la sienne.

ATT : Attribut protocole : Protocole de communication utilisé pour écrire et lire dans les bases de données du dispositif distant.

Pour résumé GAP permet aux dispositifs de faire connaissance et de se connecter. Une fois la communication initialisée le transfert de données s'effectue en écrivant et lisant les bases de données GATT distantes au moyen du protocole ATT.

⁵ Galeev, Mikhail. *Bluetooth 4.0: An introduction to Bluetooth Low Energy—Part II*. 28 Juillet 2011. http://www.eetimes.com/document.asp?doc_id=1278966.

5.1.3 GAP

Advertisissing et scanning

Un paquet d'advertissing contient les informations suivantes :

- Les modes de découvrabilité et de connectivité
- L'adresse de l'advertisser
- Le niveau de puissance de tranmission à la source
- Les services supportés
- Des données d'applications

Il y a deux modes de scanning :

1) Passive Scanning :



Figure 16 Passive scanning

2) Active :

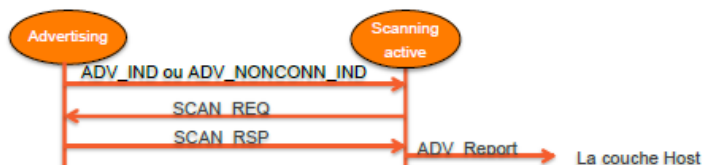


Figure 17 Active scanning

La réponse « SCAN-RSP » contient :

- Le nom du dispositif qui est advertiser
- Les services supportées

Mode d'Advertisissing

Il y a différents mode d'advertissing qui permette d'implémemter les rôles respectivant d'advertiser et broadcaster. Pour plus d'informations voir le rapport du groupe Cornaz&Elgueloui

5.1.4 Présentation du BLE112

On va présenter rapidement le BLE112 pour pouvoir ensuite décider de comment on va réaliser.

Le BLE112 utilise le Chipset TI CC2540 qui contient le module BLE et le MCU 8051.

On peut coder dessus

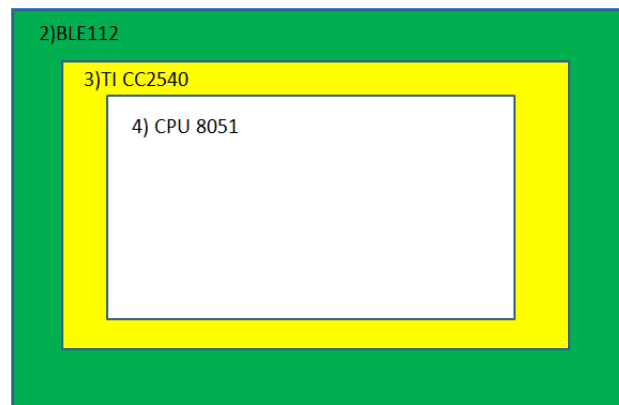


Figure 18: Ce que contient un BLE 112

Il y a plusieurs façons de réaliser une application sur ce BLE112 :

- En utilisant le langage propriétaire BGSCRIPT

- En utilisant l'environnement de travail de TI on peut coder en C⁶ directement sur le BLE en utilisant IAR

- En pilotant depuis un autre MCU le BLE112. La communication entre les deux nécessite UART. Cette approche sous-entend de coder le MCU extérieur mais également le BLE112 pour gérer le contrôle.

MCUext [Envoie commande] → [Réception commande] BLE112 [interprétation et exécution de la commande]

Puis

[Résultat commande] BLE [Envoie du résultat] → [Réception résultat] MCUext

⁶ blog.bluetooth-smart. *Programming the BLE112 with C code using IAR*. Septembre 2012.

<http://blog.bluetooth-smart.com/2012/09/11/programming-the-ble112-with-c-code-using-iar/>.

5.2 Besoins

On va dans ce chapitre étudié quelles seraient les fonctionnalités intéressantes pour l'apiculteur. Puis on va s'intéresser aux valeurs à surveiller et les dispositifs d'interaction qui serait pertinent pour lui.

5.2.1 Cas d'utilisation

Utilisations

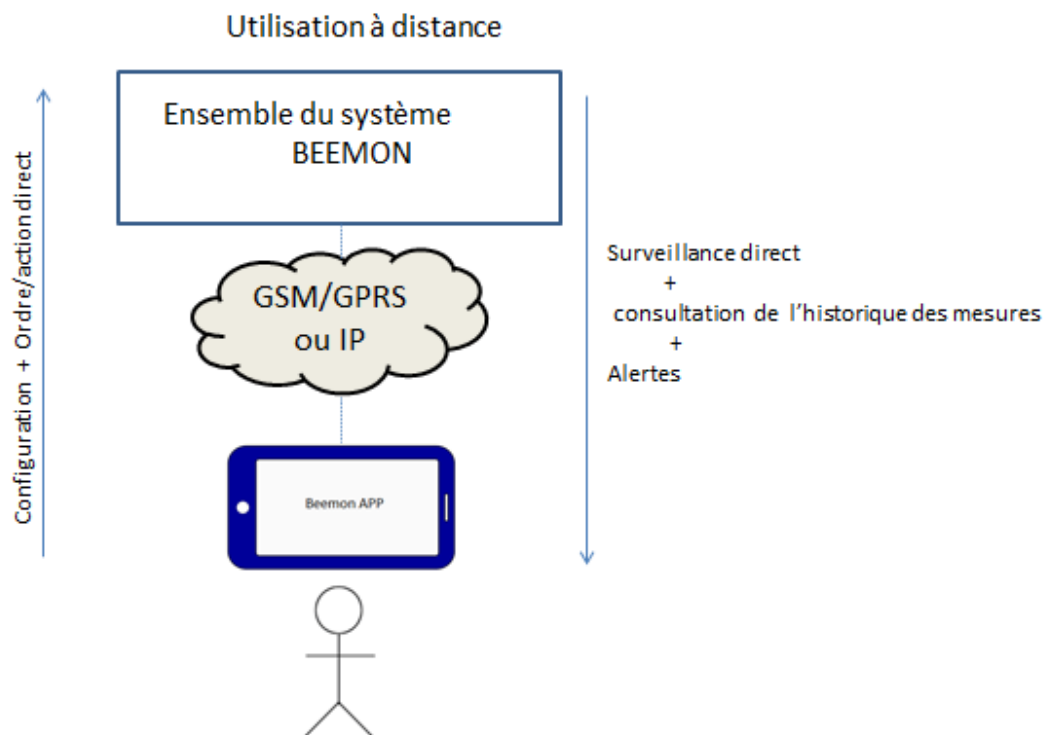


Figure 19: Utilisations du système BEEMON

Configuration : Permet de configurer les paramètres de fonctionnement, les valeurs critiques du SSI et / ou du SSC.

Actions/Ordres directs : Permet de contrôler en direct un dispositif d'interaction sur une ruche.

Surveillance directe : Permet de surveiller en direct l'état des ruches.

Consultation historique : Permet de consulter toutes les valeurs collectées à ce jour.

Alertes : Avertissement en cas de problèmes ou sur l'état de

Mise en service

Mise en service

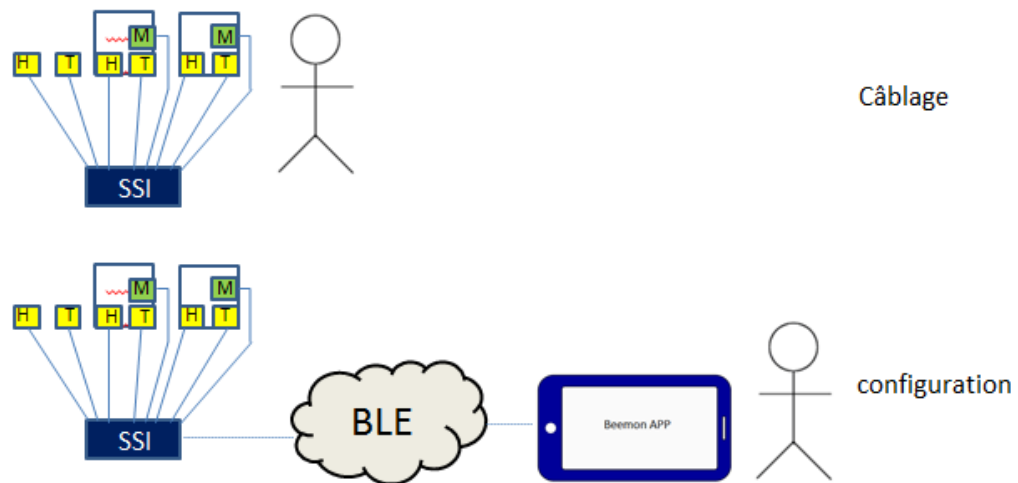


Figure 20 : Mise en service

- 1) L'utilisateur équipe sa/ses ruche(s) avec les capteurs et équipements qu'il souhaite et les connecte au SSI.
- 2) L'utilisateur configure le SSI à l'aide d'un terminal standard du type Smartphone ou Notebook. Il peut par exemple :
 - Donner un nom au SSI et une description
 - Configurer son comportement temporel
 - Donner des noms aux capteurs ainsi que des valeurs de test

5.2.2 Moyens de supervision

La première étape consiste à déterminer qu'elles seront les entrées et sorties de notre SSI. Les entrées donnent l'**information** et les sorties permettent d'**interagir** avec la ruche. Cette partie d'analyse est un **complément** à celle du premier projet.

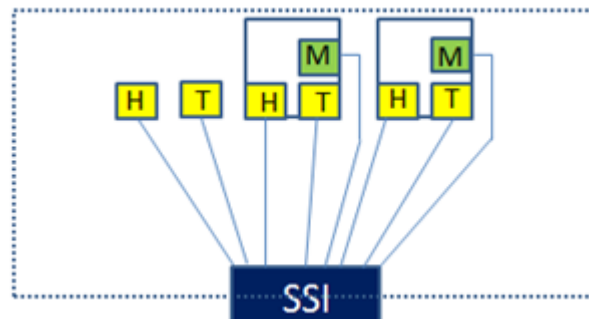


Figure 21: Interfaces SSI

Voici comment va se dérouler cette partie d'analyse.

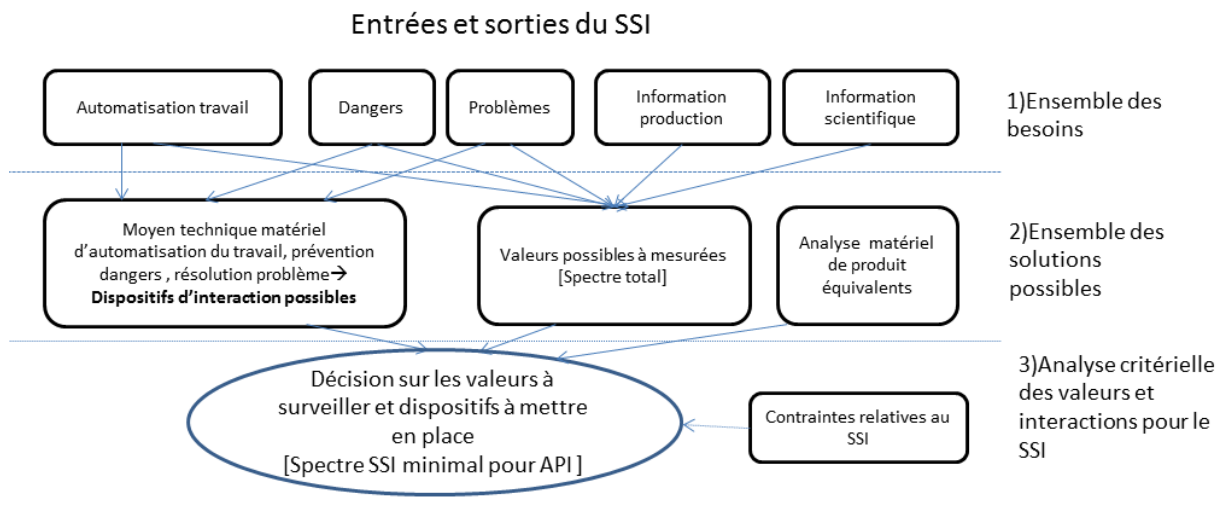


Figure 22: Plan analyse interfaces

5.2.2.1 Ensemble des besoins

Dangers et problèmes relatifs aux abeilles et aux ruches

Problèmes/dangers



Problèmes/dangers: Causes possibles [Type : Anciens problèmes ou Nouveaux problèmes]

- 1) **Gel des abeilles** (en grappe ou pas) [Vp]: Mauvaise gestion de la température et de l'humidité dans la ruche
 - 2) **Provisions inexploitable** [Vp]: Mauvaise gestion de la température et de l'humidité dans la ruche
 - 3) **Epuisement abeilles** [Vp]:
 - A) Mauvaise gestion de la température et de l'humidité dans la ruche
 - B) Pesticide dans le nectar récolté - affaiblissement de la ruche
 - C) No sema: Il mange ce qui se trouve dans l'intestin de l'abeille
 - 4) **Perte des abeilles** [Np]: Ondes électromagnétiques perturbent les abeilles
 - 5) **Prolifération de maladies** [Vp] : Mauvaise gestion de la température et de l'humidité dans la ruche
 - 6) **Parasites** [Np] : - Varroa
 - 7) **Météo non favorable pour sortie** [Vp]: - Température, humidité, pluie, vent, soleil
- http://www.abeille-et-nature.com/index.php?cat=traitement&page=nourrissement_des_abeilles
- 8) **Déplacement/chute de la ruche** [Vp] : Vent, le vol


Automatisation des tâches

Le système doit pouvoir effectuer automatiquement des tâches techniquement faciles à mettre en place et relativement économiques pour équiper chaque ruche et qui répondent à un besoin réel quant à la fréquence d'exécution par l'apiculteur.

Information de production

L'apiculteur doit pouvoir consulter les informations qui indiquent l'état de production d'une ruche :

Informations scientifiques

 EIA-FR	Projet de semestre 6 Monitoring de ruches 2	Cahier des charges Version 3
---	--	---------------------------------

Le système doit si possible permettre la surveillance de valeurs qui ont un intérêt dans l'étude de l'effondrement des colonies. Ce thème a déjà été traité dans « dangers et problèmes »

5.2.2.2 Ensemble des solutions possibles

Valeurs possibles à surveiller

En se basant sur le tableau « Valeurs possibles à surveiller », qui se trouve en annexe, on peut se rendre compte du genre d'information nécessaire pour détecter tous les problèmes.

On s'aperçoit qu'avec les moyens matériels suivants Il est possible de détecter **tous** les problèmes **sans pouvoir pour autant en identifier formellement les causes. Il permet également de surveiller certains facteurs à risque (température peut être une cause et une conséquence, il est donc à la fois un paramètre et une constante vitale) (spectre minimal : pour les apiculteurs)**

- Enregistrement sonore de la ruche
- Capture de températures
- Hygromètre (humidité relative)
- Compteur Trafic d'entrées/sorties
- Pluviomètre et anémomètre, et mesure d'ensoleillement

Pour mieux cerner les problèmes et surveiller d'autres facteurs à risque (pour les scientifiques)


- Système individuel de tracking pour chaque abeille
- Capteur de polluants biologique de type acariens
- Capteur de polluants chimique pour les pesticides, insecticides et fongicides
- Caméra
- Microphone
- Capteur de CO2
- Capteur de pression atmosphérique

Automatisation des tâches

Le système permettant d'effectuer les tâches principales quotidiennes de l'apiculteur, qui sont la surveillance de l'état de santé des abeilles et le bilan de production de la ruche, il ne se déplacera qu'en cas de besoin pour l'entretien ou la résolution d'un problème.

- 1) **Ouverture/fermeture porte d'entrée** : Cette tâche étant nécessaire le printemps et l'été, et étant l'unique tâche quotidienne nécessitant l'action physique sur la ruche par l'apiculteur, même sans autre événement majeur, est la fermeture de la porte d'entrée de la ruche la nuit et l'ouverture de cette dernière le matin, une fois les conditions météo favorables pour une sortie (température, luminosité, humidité, vent, pluie)

Information de production

 EIA-FR	Projet de semestre 6 Monitoring de ruches 2	Cahier des charges Version 3
---	--	---------------------------------

- 1) **Poids d'une ruche** : La **pesée de la ruche** est l'unique indicateur du niveau de production (mettre lien)

Information scientifique

Cette partie a déjà été traitée dans « dangers et problèmes »

Moyen d'automatisation, prévention et résolution possibles

En se basant sur le tableau « Dispositifs possibles pour la prévention de dangers et la résolution de problèmes »

- 1) **Chauffage**⁷ : Le chauffage que l'on pensait idéal pour contrôler la température ambiante de la ruche est à éviter, car tout en soulageant les abeilles, notamment en période d'hivernage, cette fonction peut dérégler leur cycle biologique. Par contre, le chauffage peut être utile en cas **de problème**. Imaginons qu'une difficulté survienne et que la température chute. Le système alerte l'apiculteur et maintient par cette fonction les abeilles en vie jusqu'à l'arrivée de l'apiculteur.

Contrôle actif des conditions (automatisation)	NON
Prévention d'un problème	OUI (jusqu'à l'arrivée de l'api)
Résolution d'un problème	NON


Tableau 3 : Chauffage ruche

- 2) **Humidificateur** : Il permet de régler le taux d'humidité au sein de la ruche. Bien qu'il n'y ait aucune mise en garde quant à son utilisation, la logique veut que cette fonction « **dérègle** » également le cycle biologique des abeilles. Il pourrait, **en cas de problème** (dilution provisions, aération naturelle provenant de la différence de pression atmosphérique) comme le chauffage, permettre de maintenir de bonnes conditions dans la ruche jusqu'à l'arrivée de l'apiculteur. Cependant en hiver, l'action d'humidifier l'air en cas de détection d'un air trop sec, aggraverai la situation.

Contrôle actif des conditions (automatisation)	NON
Prévention d'un problème	NON (risque d'aggraver la situation)
Résolution d'un problème	NON

Tableau 4 : Humidificateur ruche

⁷ ActuAPI. Isoler, attention à l'humidité. Février 2009. <http://www.cari.be/medias/actuapi/actuapi46.pdf>

 EIA-FR	Projet de semestre 6 Monitoring de ruches 2	Cahier des charges Version 3
---	--	---------------------------------

3) Goutte à goutte : A pour but de diluer les provisions lorsque les conditions intérieures ne sont pas réunies.

Contrôle actif des conditions (automatisation)	NON
Prévention d'un problème	OUI (jusqu'à l'arrivée de l'api)
Résolution d'un problème	NON

Tableau 5 : Dilution provisions

4) Ouverture/fermeture de la porte : Permet par aération et hermétisme de contrôler la température et l'humidité intérieures de la ruche.

Contrôle actif des conditions (automatisation)	OUI
Prévention d'un problème	OUI (jusqu'à l'arrivée de l'api)
Résolution d'un problème	NON

Tableau 6 : Ouverture et fermeture de la porte de la ruche

Analyse matériel de produit équivalent

Il faut se référé ici au tableau « les offres de systèmes équivalents »

La plupart propose les mesures suivants :

- Mesure du poids de la ruche
- Mesure de la température et humidité ambiante

Certains

- Pluviomètre
- Capteur de choc pour l'antivol
- Intensité lumineuse
- Volume du couvain et de la grappe
- Niveau de bruit
- Monoxyde de carbone
- Activité sur le pont d'envol

5.2.2.3 Solutions choisies pour le SSI idéal

Cette section a pour but de déterminer quel sera le spectre **minimum** de valeurs à surveiller et d'interactions possibles du SSI. Les sections précédentes avaient pour but de définir le spectre idéal qu'un système doit avoir pour faire face à toutes les situations (détecter, prévenir et résoudre tous les dangers/problèmes). Ce spectre ne tenait pas compte des limites et contraintes inhérentes au SSI (consommation électrique, puissance de calcul nécessaire) et la réalité pratique (complexité d'installation, coût des dispositifs et capteurs).

Avant de continuer voici une représentation de l'intérêt respectif des solutions pour l'apiculteur et le scientifique (le tout est le **spectre total**).

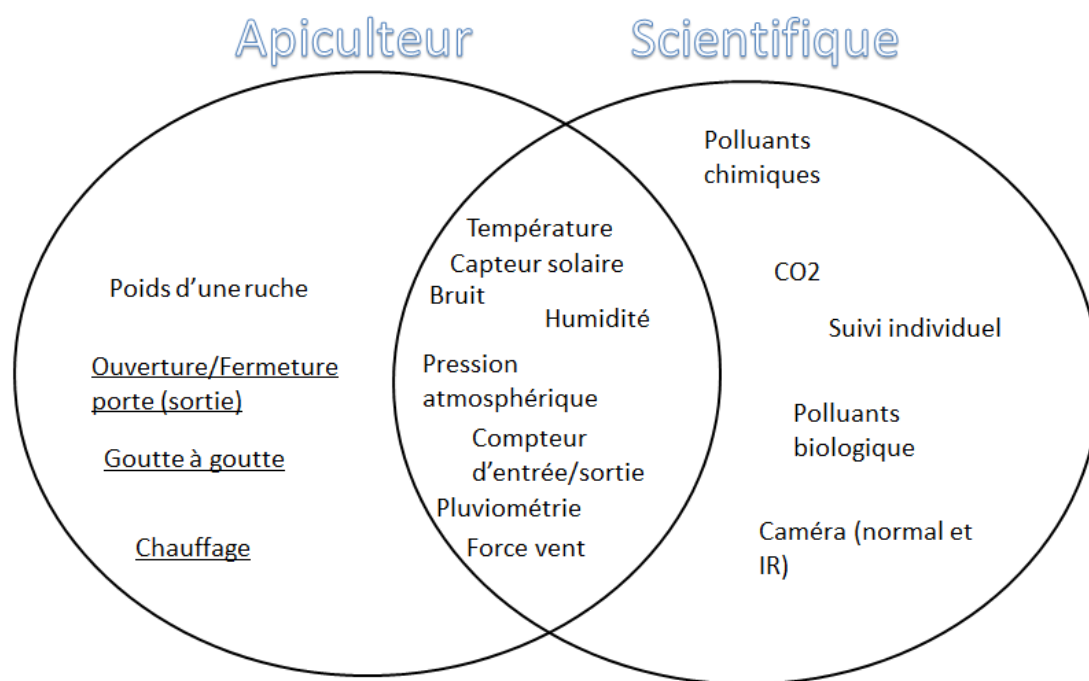



Figure 23: Spectre total des valeurs à surveiller et des dispositifs d'interactions

Température->T	Caméra-> Cam	Gaz carbonique-> CO2
Humidité-> H	Microphone-> Mic	Polluants chimiques->PC
Ensoleillement-> S	Suivi d'abeilles-> SI	Polluants biologiques-> PB
Pluviométrie -> P	Poids d'une ruche- > PR	Compteur trafic ->CT
Anémométrie-> Vt	Pression atmosphère- >PA	<u>Porte</u> -> <u>D</u>
<u>Goutte à goutte</u> -> <u>GG</u>	<u>Chauffage</u> -> <u>Ch</u>	

Tableau 7 : Abréviations des valeurs

 EIA-FR	Projet de semestre 6 Monitoring de ruches 2	Cahier des charges Version 3
---	--	---------------------------------

Lors du premier projet nous avons défini les contraintes suivantes pour le SSI :

- 1) Consommation énergétique minimale, donc autonomie étendue,. La consommation est directement liée au nombre et aux types de composants que l'on branche sur le SSI

L'énergie consommée dépend :

- De la puissance nécessaire au fonctionnement du composant
- Du temps d'utilisation : Durée d'une action ou d'une mesure, fréquence de d'action ou de la mesure
- Des ressources en calcul pour la mesure (MCU)

- 2) Economique pour équiper un maximum de ruches (coût direct et frais d'utilisation) :

A) Coût direct : Les composants constituant le SSI doivent être calculés en fonction des besoins. Une grande puissance de calcul pour l'acquisition de valeurs et des traitements simples sur ceux-ci n'est pas nécessaire.

B) Frais d'utilisation : Pour communiquer avec l'extérieur on a choisi le GSM/GPRS, dont le prix est basé sur le volume de données transmis. Ainsi la quantité d'informations résultant du choix du matériel utilisé n'est pas à négliger.

- 3) Il faut aussi tenir compte de l'utilité et de la pertinence des solutions pour l'apiculteur.

-Valeurs à surveiller: Si une valeur physique est commune à plusieurs problème ou dangers elle sera plus utile qu'une autre valeur spécifique à un problème uniquement. Il y a aussi la notion de rapidité d'aggravation d'un problème et la vitesse du passage entre la détection d'un danger et l'apparition du problème résultant.

-Dispositifs d'interactions : Ici les notions principales relatives à une situation problématique sont la fréquence de survenue du problème et la vitesse d'aggravation du problème si rien n'est entrepris dans les instants qui suivent sa détection. En ce qui concerne l'automatisation, l'utilité se mesure à la fréquence de la tâche et la pertinence vis-à-vis de la complexité et des coûts de l'installation.

Haut(e) -> H

Moyen(ne)-> M

Bas(se)-> B

*Pour les dispositifs, voir les situations énumérées à la section « Moyen d'automatisation, prévention et résolution possibles.

Spectre élargit scientifique (connu)																	
Spectre minimal api																	
Contr	T	H	Vt	CT	PR	<u>D*</u>	S	P	PA	PC	PB	CO 2	<u>GG</u> *	<u>Ch*</u>	SI	Ca m	Mic
1	B	B	B	B	B	M	B	B	B	B	B	B	B	H	H	H	M
2	B	B	B	B	B	B	B	B	B	B	B	B	B	H	H	H	H
3	H	H	H	H	H	H	M	M	B	B	B	B	B	B	B	H	B

Tableau 8 : Analyse des solutions possibles

On a donc pour ces valeurs à surveiller (encadré bleu) des périphériques qui ne requièrent pas une vitesse de transfert rapide (comme une caméra par exemple), ni de grandes ressources en eau niveau MCU ni une grande capacité de mémoire (comme la caméra et le mic). Excepté le système d'ouverture et fermeture porte, le périphérique ne devrait pas consommer beaucoup.

Compteur d'entrées/sorties et ouverture/fermeture porte

Un compteur d'entrées/ sorties utilise des capteurs infrarouges. Avec 2 capteurs il est possible de savoir si l'abeille rentre ou sort. Un compteur d'abeilles prêt à l'emploi n'a pas été trouvé (en raison de contraintes mécaniques). Cependant un prototype a été réalisé.

Pour l'ouverture/fermeture de la porte la même réflexion peut être faite, aucun dispositif prêt à l'emploi.

On donc décidé de concevoir une esquisse sur papier de ce dispositif, qui, si mécaniquement n'est probablement pas la solution parfaite, il permet de définir le type de composants nécessaires à sa réalisation.

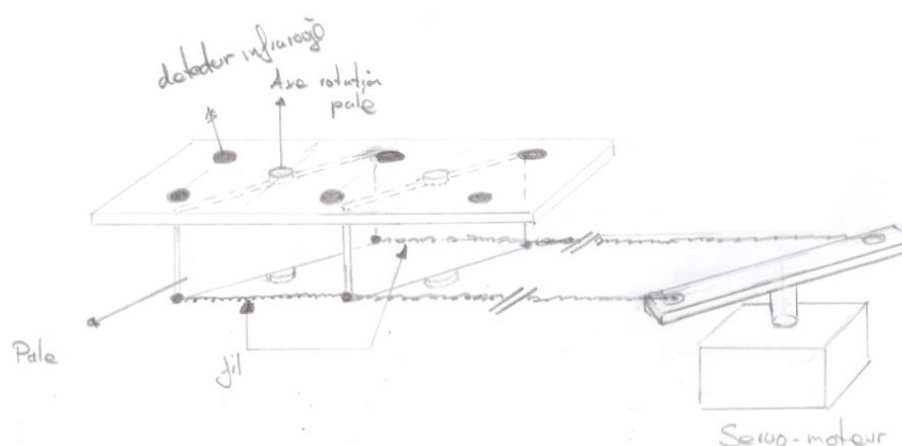



Figure 24 : Portes de détection

Ouverture /porte	En utilisant un servomoteur
Compteur d'entrées/sorties	Basé sur capteur infrarouge

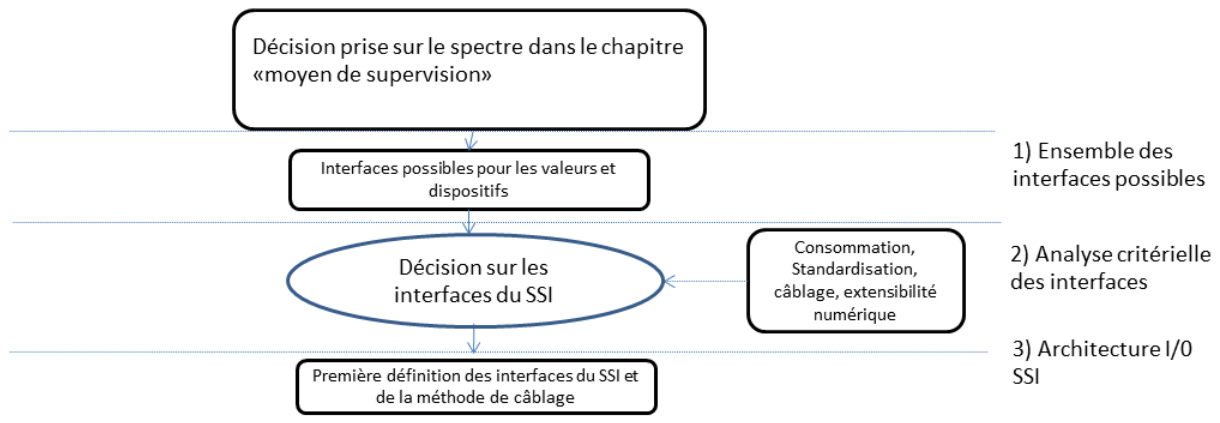
 EIA-FR	Projet de semestre 6 Monitoring de ruches 2	Cahier des charges Version 3
---	--	---------------------------------

Nombre et types de capteurs pour l'apiculteur

Pour l'apiculteur il faut calculer au bas mot le matériel suivant pour **une ruche**

- 1 capteur de température interne
- 1 capteur d'humidité interne
- 1 capteur de température externe
- 1 capteur d'humidité externe
- 1 anémomètre
- 1 ou plusieurs détecteurs infrarouges
- 1 servomoteur
- 1 pèse ruche

5.3 Interfaces



5.3.1 Interfaces possibles

Entrées/sorties	Valeurs spécifiques
Capteurs température et humidité (ruche et météo)	Analogique, I2C, SPI, Propriétaire
Anémomètre simple	Analogique, Variation fréquentielle d'impulsion*** (à chaque tour), propriétaire
Station météo	Propriétaire
Pèse ruche (d'après celle fournit par M.Clément)	Propriétaire
Servomoteur	I2C, PWN
Détecteur infrarouge	Bi-state


Tableau 9 : Interfaces possibles

5.3.2 Analyse critique des possibilités

Analogique versus numérique

- Les capteurs numériques sont un peu plus chers mais ont comme principal avantage de faciliter l'interconnexion capteur-entrée du microcontrôleur, comme déjà constaté dans le premier projet. On va donc privilégier ce type de capteurs.
- On va, pour assurer la plus grande compatibilité possible, évaluer les interfaces les plus standardisées pour ce type de composants. Ce sont le SPI et l'I2C

Critères	I2C	SPI
Consommation électrique	Relativement élevée dû à l'architecture open-drain	Faible à cause de l'architecture CMOS
Standardisation	La majorité des capteurs simples numériques utilisent I2C ou des dérivées.	Très utilisé mais beaucoup moins que l'I2C. SPI est surtout employé pour la communication avec un périphérique demandant une vitesse de transfert élevée

 EIA-FR	Projet de semestre 6 Monitoring de ruches 2	Cahier des charges Version 3
---	--	---------------------------------

Câblage	Permet un câblage en BUS et ne nécessite que deux fils. L'alimentation peut également être en bus	Nécessite une ligne de sélection par composant, donc le macro câblage est en étoile
Extensibilité numérique	Permet de communiquer avec 128 composants seulement avec un bus à 2 fils en utilisant un adressage. *Cependant comme tous les périphériques I2C ne proposent pas de configurer l'adresse, il faudra certainement multiplexer les lignes du bus	Permet de communiquer avec plusieurs composants mais nécessite une ligne de sélection par composant

Tableau 10 : I2C vs SPI

- Choix des interfaces : Dans le cadre de notre projet, I2C a l'avantage d'offrir plus de **composants compatibles** pour étendre le spectre mais **consomme plus** que le SPI. L'avantage qu'offrirait l'I2C avec un adressage software est relatif car certains composants I2C permettent de configurer cette adresse quand les autres en ont une fixe.

5.3.3 Interfaces nécessaires


- Pour l'apiculteur il faut prendre en compte **au minimum** le matériel suivant **pour une ruche**1
- capteur de température externe [I2C]
- 1 capteur d'humidité externe [I2C]
- 1 anémomètre [1 ligne d'interruption ou I2C-Propriétaire]

3 Ligne I2C + lignes avec interruption

Pour chaque ruche cela nécessite

- 1 capteur de température interne [I2C]
- 1 capteur d'humidité interne [I2C]
- plusieurs détecteurs infrarouges (2,4,6,8) [2 lignes d'interruption min]
- 1 servomoteur [Ligne simple pour génération PWN]
- 1 pèse ruche [I2C-Propriétaire]

3 ligne I2C + 2 Lignes d'interruption + une ligne simple

 EIA-FR	Projet de semestre 6 Monitoring de ruches 2	Cahier des charges Version 3
---	--	---------------------------------

5.3.4 Offre en interfaces du BLE112

On va dans cette voir ce qu'offre le BLE au niveau des interfaces. Pour ceci il faut se baser sur le tableau « offre du BLE112 ⁸ » se trouvant en annexe. Il faut tenir compte des 2 limitations suivantes

- 1) Si I2C est utilisé l'UART ou le SPI du canal 1 de l'alternative 2 ne peut pas être utilisé
- 2) Pas d'interruption sur le port 2

On s'aperçoit que le BLE112 offre toutes les interfaces nécessaires, voire plus. A savoir :

- I2C
- Sortie PWM (sortie normale)
- Lignes d'interruption

Il reste en plus des interfaces disponibles qui sont :

- SPI : pour d'autres capteurs ou dispositifs. Permet notamment de piloter un affichage
- Analogique
- UART : Pour permettre une communication entre 2 MCU

Etant donné que **l'I2C a été privilégié** pour les capteurs, il faut **multiplexer** cette sortie. Ce qui inclut d'utiliser d'autres ports pour les lignes de sélection. Pour le moment on a pu vérifier que l'offre répond à la demande

⁸ Bluegiga. *BLE112 Datasheet*. Mars 2014. CD : ..\Documentation technique\BLE\BLE112_Datasheet.pdf P.9

5.4 Architectures matérielles

On a observé dans le chapitre « moyen de supervision » les valeurs intéressantes à mesurer et les dispositifs d'interactions pour l'apiculteur et les scientifiques. Le premier est appelé « spectre minimal » et le second « spectre élargit ». Dans le chapitre « interfaces possibles » on a pu vérifier que le BLE112 offrait les interfaces nécessaires. On sait de plus que les dimensions de ce dernier sont suffisamment petites pour envisager de le placer à l'intérieur.

Ce chapitre a pour but d'étudier les différentes configurations matérielles que pourrait prendre le SSI en tenant compte des contraintes d'utilisation pour l'apiculteur et de l'extensibilité souhaitée pour les scientifiques.

L'apiculteur doit utiliser un système :

- Sans ou peu de câblage au vu du nombre accru de ruches à équiper
- Aux interfaces et capteurs clairement définis (avec code de couleur si nécessaire)
- Avec un découpage éventuel en plusieurs SSI.

Le scientifique :

- Câblage acceptable au vu du nombre restreint de ruches à équiper
- Permettant de brancher davantage de capteurs et moyens d'interactions
- Permettant de manier les interfaces à sa convenance, sans être limité par un nombre restreint de capteurs ou de dispositifs d'interactions acceptés

Ci-après une découpe d'une ruche Dadant

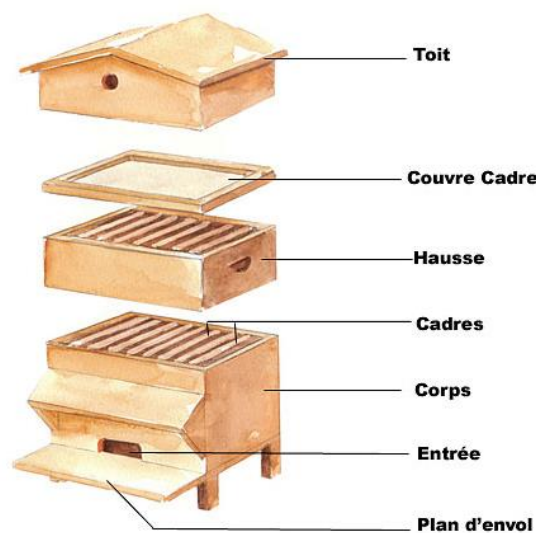


Figure 25 Découpage ruche Dadant

Localisation des capteurs et dispositifs pour le « spectre minimal »

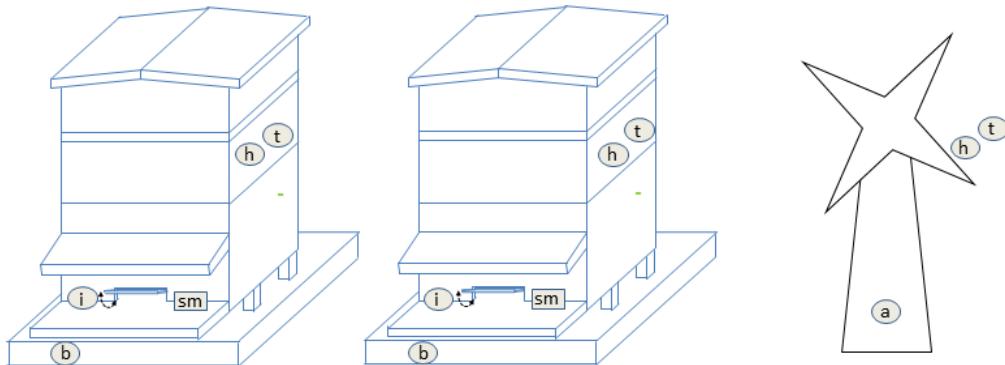


Figure 26 Localisation des mesures pour l'apiculteur

Les capteurs de température et d'humidité internes doivent être placés le plus haut possible. L'idéal serait de les positionner sur le couvre cadre.

La balance, le système d'ouverture et le compteur se situent en bas. Le système de porte peut être simplement placé à l'extérieur, devant la porte et entre le plateau d'envol et le toit de sortie.

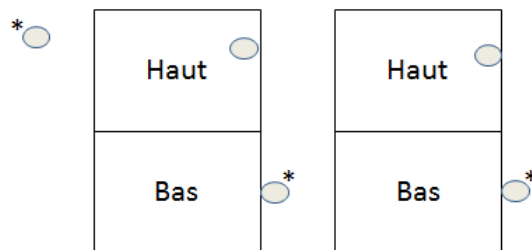


Figure 27 Localisation mesures simplifié

L'analyse qui suit se base sur les contraintes d'utilisation de l'apiculteur et l'extensibilité nécessaire pour les scientifiques.

On peut envisager les dispositions du SSI suivantes :

E1 : 1 SSI pour plusieurs ruches et les capteurs extérieurs (communs à plusieurs ruches)

E2 : 1 SSI par ruche et 1 SSI pour les capteurs extérieurs (communs à plusieurs ruches)

E3 : SSI pour la partie supérieure (balance, porte, compteur), 1 SSI pour la partie inférieure (température et humidité) et 1 SSI pour les capteurs extérieurs

Avec un SSI pouvant prendre les formes suivantes

F1 : Une carte avec toutes les interfaces nécessaires au spectre minimum et des supplémentaires pour le spectre élargit. Les deux étant confondus sur un seul port.

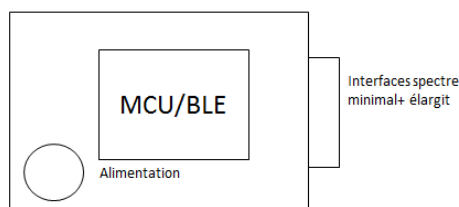


Figure 28 SSI forme 1

F2 : Une carte avec toutes les interfaces nécessaires mais dont le spectre minimal est séparé du spectre élargit.

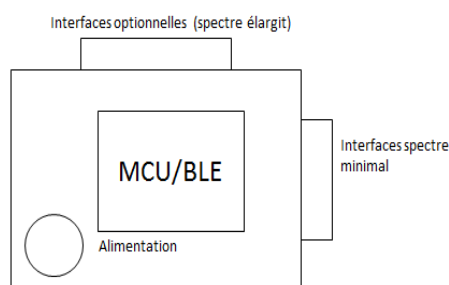


Figure 29 SSI forme 2

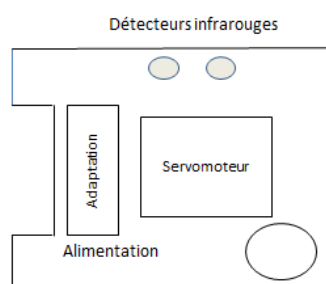


Figure 30 Extension pour la porte

Il s'agit ici d'une variante de F1 mais qui propose un standard que nous pouvons reprendre par la suite pour notre système de porte par exemple. L'idée est donc de fournir à l'avenir un dispositif de « porte » prêt à l'emploi qui serait une extension de la carte principale. Comme le dispositif d'ouverture qui pourrait être éventuellement placé sur l'extérieur de la ruche à l'extérieur (comme imaginé plus), on peut directement relever la température et l'humidité extérieures. Ici le nombre de capteurs peut varier entre ce qui nécessaire a une ou plusieurs ruches

F3 : Idem F2 mais avec des capteurs embarqués.

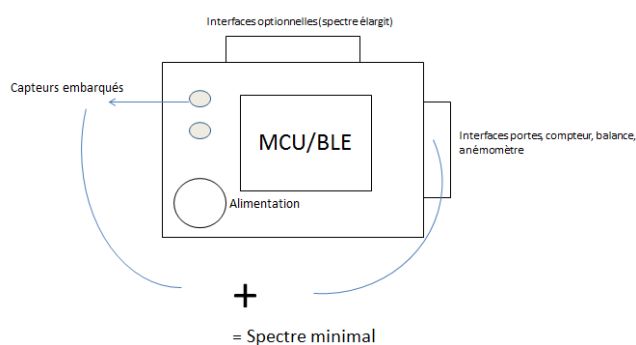



Figure 31 SSI forme 3

 EIA-FR	Projet de semestre 6 Monitoring de ruches 2	Cahier des charges Version 3
---	--	---------------------------------

Installation	E1	E2	E3
F1	Câblage fastidieux	Câblage fastidieux	Acceptable
F2	Câblage fastidieux	Acceptable	Acceptable
F3	Câblage fastidieux	Acceptable	Aisée

Tableau 11 Formes de SSI vis-à-vis de ses dispositions

On s'aperçoit que la solution F1 convient le mieux à toutes autres configurations envisageables. Les trois répondent au besoin d'extensibilité scientifique, mais les 2 dernières séparent clairement les 2 spectres. L'avantage de F3 par rapport aux autres est qu'il ne nécessite aucun câblage pour les capteurs de température et d'humidité. On a donc opté pour la **forme numéro 3** car elle offre la meilleure solution vis-à-vis de l'installation. Un BLE112 coûte actuellement environ CHF 40, le reste de la carte peut être acheté pour moins de CHF10. Si l'on tient compte que le BLE est nouveau, on peut envisager une baisse de son coût dans l'avenir.

5.5 Architectures communications

Pour analyser les différentes architectures possibles et choisir la bonne il faut se baser sur

- Les solutions de supervision proposées
- Connaître le fonctionnement et les limites du Bluetooth low energy.
- En partant d'une idée contraire au premier projet, on va préférer une solution plus réactive en utilisant le module GSM plus souvent, voire constamment, car l'émetteur / récepteur est éloigné des ruches grâce au BLE (100m)
- Le nombre de ruches dans un groupe d'une grande exploitation peut aller jusqu'à **30** et la **disposition n'est pas forcément ordonnée**

Bluetooth Low energy

-Le nombre de dispositifs esclaves est illimité grâce au système d'advertising.

-Le nombre d'esclaves avec lesquels un maître peut communiquer simultanément est de **8**


-Le fait de rester connecté ne consomme pas beaucoup d'énergie grâce au système d'évènement de connexion

MCU

L'unique contrainte est que le CC2540 ne dispose pas d'une horloge interne permettant de connaître la date. (RTC)

Topologie d'un groupe

Nous avons vu que dans un groupe les ruches étaient organisés en sous-groupe mais qu'il existait aussi des ruches isolées. Il y a en tout 30 ruches.

	Projet de semestre 6 Monitoring de ruches 2	Cahier des charges Version 3
---	--	---------------------------------

Architectures BEEMON

Architectures BLE SSI-SSC

- A) Le nombre de SSI dépasse le nombre de connexions simultanée maximal on privilégie donc des connexions BLE courtes pour la notification d'un problème ou le dépôt sur le SSC. Une fois déposée, la communication est rompue jusqu'au prochain dépôt ou problème.
- B) Le nombre de SSI est égal au nombre maximum de connexions parallèles possibles qui peuvent rester constamment connecté au SSC.
- C) Idem A, mais propose que le SSI essaie de temps à autre de se connecter au SSC uniquement pour voir s'il y a une information le concernant ou si le **SSC souhaite se connecter avec lui**.

Architecture GSM SSC-Serveur

A) GSM connecté en cas d'évènement [alerte, dépôt]

B) GSM connecté constamment ou périodiquement

Archi.	Réactivité ensemble du système	Consom. SSI	Consom. SSC	Extensibilité nombre SSI
BLE(A)-GSM(A)	Basse Les dispositifs peuvent réagir uniquement après avoir déposé des données ou notifié un problème	Basse	Basse Le module GSM n'est activé qu'en cas de besoin ou ponctuellement pour voir s'il y a du nouveau de l'apiculteur	Illimité
BLE(B)-GSM(A)	Basse Même si les SSIs et le SSC sont toujours connectés cela n'apporte rien car le SSC est coupé du monde la plupart du temps (1)	Basse	Basse	Nbre max de connexions simultanées limité
BLE(C)-GSM(A)	Basse (1)	Basse	Basse	Illimité
BLE(A)-GSM(B)	Basse ou moyenne Dépend de la fréquence de dépôt du SSI au SSC	Basse	Haute	Illimité
BLE(B)-GSM(B)	Haute Du fait que les SSIs sont constamment connectés au SSC	Basse	Haute	Nbre max de connexions simultanées limité
BLE(C)-GSM(B)	Moyenne - Haute Dépend de la fréquence de pooling du SSI au SSC.	Basse	Haute	Illimité

Tableau 12 Architectures de communications

Si nous voulons offrir la possibilité à l'apiculteur de surveiller et d'interagir en direct avec son exploitation il faut privilégier les architecture **BLE(B) – GSM(B)** et **BLE(C) – GSM(B)**

Reste encore à choisir entre ces 2 options. La question est de savoir quelle quantité de SSI nécessite une petite et une grande exploitation pour couvrir ses besoins.

Il peut y avoir, dans le cas du CC2540, un maximum de 8 Esclaves ⁹connectés simultanément

On a choisi lors de l'architecture matérielle de privilégier un SSI aux dimensions très réduite embarquant deux capteurs pour la température et l'humidité, qui a un port formatée pour répondre au besoin d'une seule ruche. Il propose également un port supplémentaire pour élargir le spectre minimal s'il le faut.

En sachant que le nombre de ruche dans un groupe peut avoisiner la trentaine, il semble que l'architecture BLE(C)-GSM(B) est la mieux adaptée. Elle permet d'avoir autant de SSI que nécessaire tout en pouvant assez rapidement communication avec eux en cas de besoin

⁹ *BLE112 Bluetooth Smart Module* . s.d. <https://www.bluegiga.com/en-US/products/bluetooth-4.0-modules/ble112-bluetooth-smart-module/> (accès le Mai 5, 2014).

5.6 Composants du SSI

On va dans ce chapitre synthétiser ce qui a été décidé pour les points fondamentaux vu précédemment. Puis dans un second temps se pencher sur les aspects secondaires comme la prise en main du SSI

Dans le chapitre « Interfaces possibles » on a défini le type d'entrées et de sortie que l'on va aller utiliser pour répondre au besoin minimal d'une exploitation d'au minimum une ruche. On a par la suite vérifié que le BLE112 répondait à ces besoins en se reportant au tableau des interfaces de celui-ci.

On a donc :

Capteur d'humidité et température-> entrée I2C

Balance -> entrée I2C ou propriétaire

Portier -> sortie I2C ou PWN

Détecteur-> Ligne d'interruption

Anémomètre -> Ligne d'interruption, analogique, propriétaire

On a également décidé pour le cas où un élément utilisant une communication propriétaire ou une autre technologie de communication d'utiliser des circuits d'adaptation I2C. On va donc fixer les besoins suivants

Capteur-> I2C *4

Balance-> I2C *1

Portier -> I2C ou PWN *1

Détecteur-> lignes d'interruption*2

Anémomètre-> ligne d'interruption ou I2C * 1

7 *I2C + 3 lignes interruption + 1 PWN Si 1 SSI utilise tout ce périphérique


Après vérification on sait qu'il faudra de toute façon multiplexer les entrées/sorties I2C.

Après recherche, le nombre max de lignes i2c pour un multiplexeur est de 8.

On a également vu dans les cas d'utilisation qu'il faudrait, pour la mise en service, disposer de quoi produire et connaître une identité provisoire. On va utiliser un mini affichage LCD avec le moins de segments possible mais avec au minimum 4. On va également pouvoir vérifier si le SSI réagit. Il pourrait également servir à changer d'info sur l'affichage.

Pour voir l'utilisation des ports choisis sur le BLE112 il faut regarder le tableau situé dans l'annexe appelé « **choix sur les interfaces proposées par le BLE112¹¹** »

Il nous reste 3 ports analogiques que bien entendu utiliser, même si on a privilégié l'I2C.

	Projet de semestre 6 Monitoring de ruches 2	Cahier des charges Version 3
---	--	---------------------------------

5.7 Commande de matériel

Description	Nom	Quantité
Capteur de température et d'humidité I2C	HIH6121	1
Switch I2C 8 Channel 0 à 400 KHz avec une ligne de reset (voir port libre dans tableau annexe)	PCA9548APW	2
Module BLE	BLE112	0 (déjà à disposition)

Tableau 13 Commande matériel

6 Spécification du SSI

On va dans ce cl
enfin définir sor

6.1 Utilisati

BEEMON et

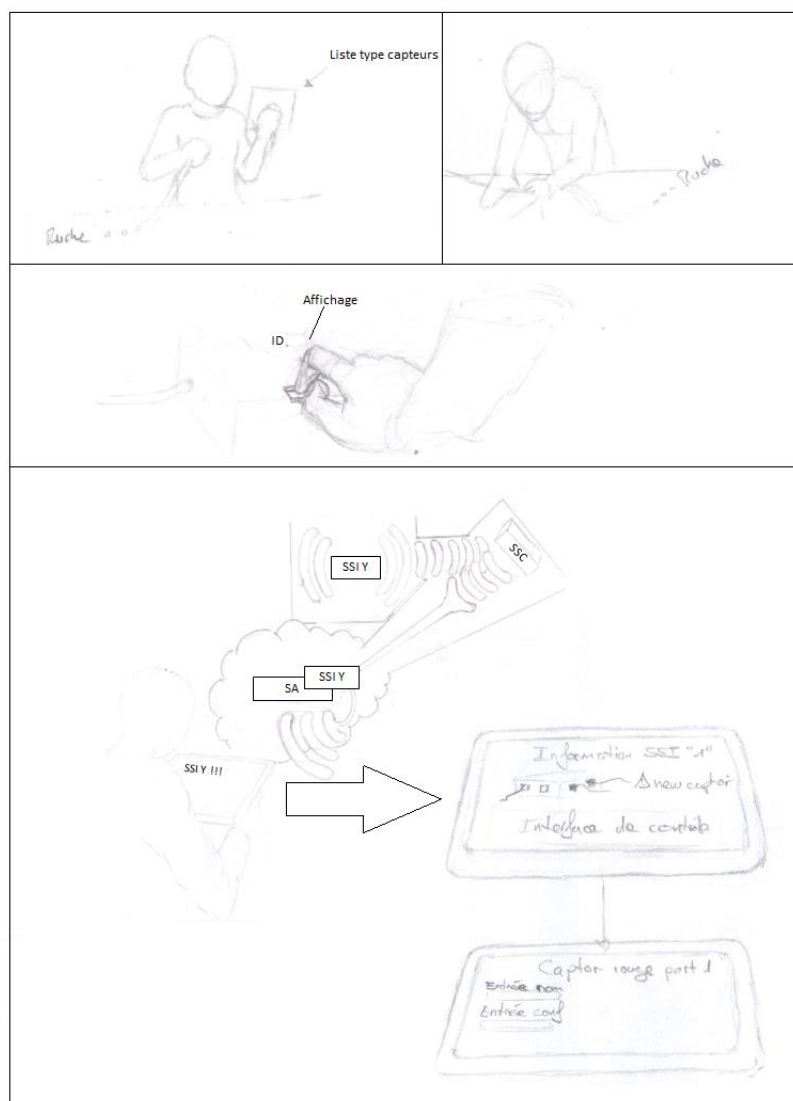



Figure 32 BD mise en service

	Projet de semestre 6 Monitoring de ruches 2	Cahier des charges Version 3
---	--	---------------------------------

2) L'utilisateur met pour la première fois en service un SSI. Il commence par un capteur de température, qui selon le mode d'emploi est à brancher sur un port rouge. Il le(s) branche.

3) Il appuie le bouton démarrer. Le SSI va dans un premier temps va détecter tous les capteurs branché. Il génère également un identifiant (ou donne son adresse mac) et le présente sur un petit affichage. L'utilisateur retient cet identifiant. Remarque : La génération peut être remplacé par un code que l'utilisateur saisi sur des switch ou par encore par l'adresse MAC si elle est connue.

3) L'utilisateur lui se connecte à son compte « Beemon » et introduit l'ID. Le nouveau SSI apparaît dans la boîte « mes SSI ». Le SSI est maintenant lié à son compte. En cliquant dessus, il ouvre le panneau de contrôle du SSI ou apparaît sa configuration actuelle (donc aucune). Il configure le SSI. Il lui donne pour description un nom, l'appartenance au groupe ou sous-groupe, des coordonnées géographiques. Il peut également définir son comportement.

Il peut s'il le souhaite définir des valeurs de test pour les capteurs qu'il vient d'installer. Une fois que cela est fait il valide la configuration. Le SSI à maintenant configurer

Configuration à distance, supervision directe

Le SSI à été mis en service par l'utilisateur. Quelques temps après il décide de changer certains paramètre

- 1) Il se connecte à son compte et choisie le SSI qui veut configurer/surveiller/contrôler
- 2) Il attend que le SSI en question soit disponible (voir fonctionnement)
- 3) Il le configure/surveille/contrôle

Alertes

- 1) En cas de problème l'utilisateur reçoit un SMS , car ce mode reste le plus apte pour être certain d'informer l'utilisateur

Consultation historique

- 1) L'utilisateur se connecte à son compte et choisie simplement le SSI dont il aimerait connaître l'historique

6.2 Systèmes BEEMONS

Scope des systèmes

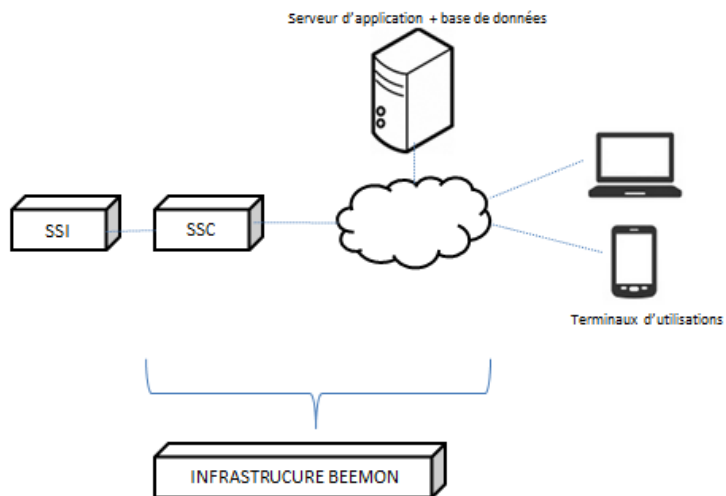


Figure 33 Architecture logicielle Global

	SSI	SSC	SA + BD	Terminals
Rôle	1) Collecte et dépose 2) Teste les valeurs et alerte si besoin 3) Pilote des interfaces	Stock Sert de relais entre l'apiculteur et les SSI (Exécute un planning)	Stockage permanent Permet la communication directe utilisateur SSI Autres services	Présenter et offrir une interface d'utilisation
Update & Fix	Non ou rare	Non ou rare	oui	Oui ou non (application native ou web)
Valeur	Brut	Brut ou transformé	Transformé	Transformé et présenté

Figure 34 Scopes des systèmes beemon

6.3 Fonctionnement

Comportement temporel

Avant d'expliquer le fonctionnement général on va définir le comportement temporel du système

Fréquence de collecte : Tout les X temps le SSI mesure les valeurs d'entrées et les stockent

Fréquence de proposition de connexion directe : Tout les Y temps le SSI propose une connexion au SSC pour voir si l'utilisateur souhaite communiquer avec lui. Cette proposition sert à implémenter la configuration, la surveillance et interaction directes.

Moment de dépôt : Quand le SSI a collecté suffisamment de mesure il doit les déposer sur le SSC. Pour être plus précis le SSI essaiera de déposer dès que le premiers tampon est plein

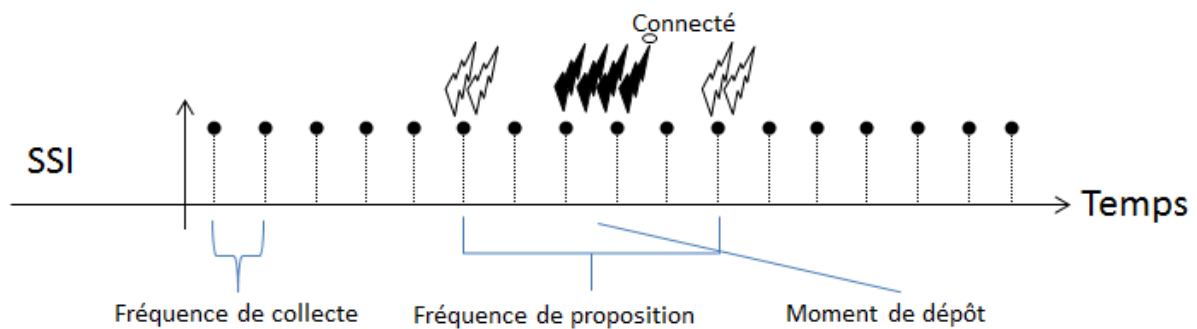


Figure 35 Comportement temporel du SSI

Les bâtonnets : Point de mesure.

Les éclairs blancs : Propositions émit par la SSI au SSC tous les Y temps. La proposition ne dure pas longtemps.

Les éclairs noirs : Ordre du SSI au SSC d'établir une connexion pour pouvoir déposer les valeurs collectées. Ici le SSI insiste jusqu'à obtention de la connexion.

Proposition de connexion directe

Le SSI demande de temps au en temps au «système» si l'utilisateur souhaite interagir avec lui:

Configuration, Supervision en direct

1^{ère} cas

- 1) Le SSI envoie des propositions au système. Ce dernier gère les demandes de l'utilisateur

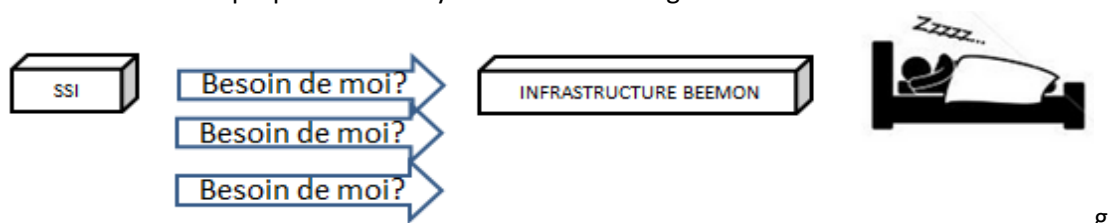


Figure 36 Connexion directe 1

- 2) L'utilisateur n'a déposé aucune demande au système. Il ignore donc les propositions du SSI. Après un certain le SSI considère que l'on n'a pas besoin de lui et stoppe ces annonces.

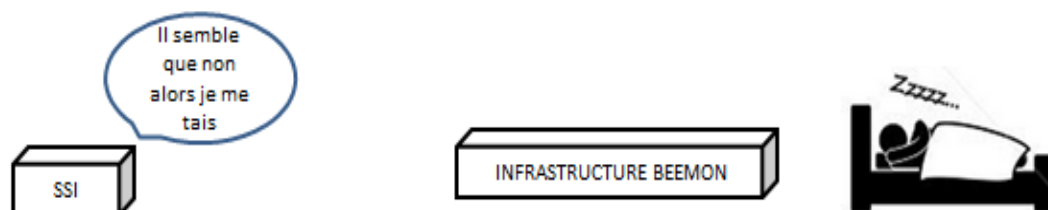


Figure 37 Connexion directe 2

2^{ème} cas

- 1) L'utilisateur souhaite communiquer avec le SSI, il le fait savoir au système et tout particulièrement au SSC. Après, à un moment donné, le SSI se manifeste auprès du SSC et celui établit la connexion

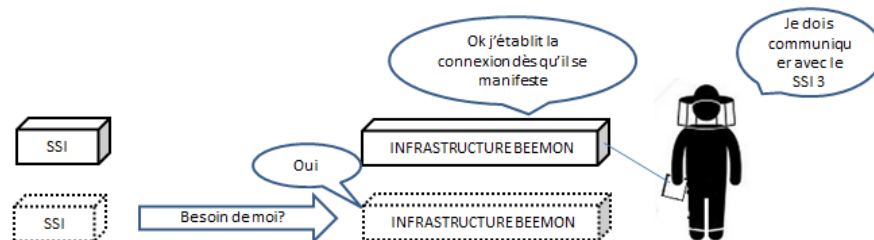


Figure 38 Connexion directe 3

- 2) La connexion directe est établie. L'utilisateur se renseigne et ensuite ordonne une tâche au SSI

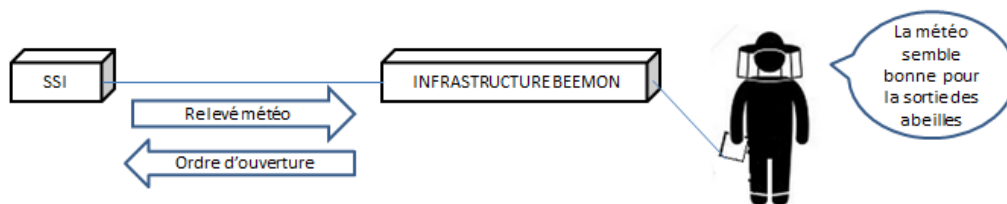


Figure 39 Connexion directe 4

- 2b)

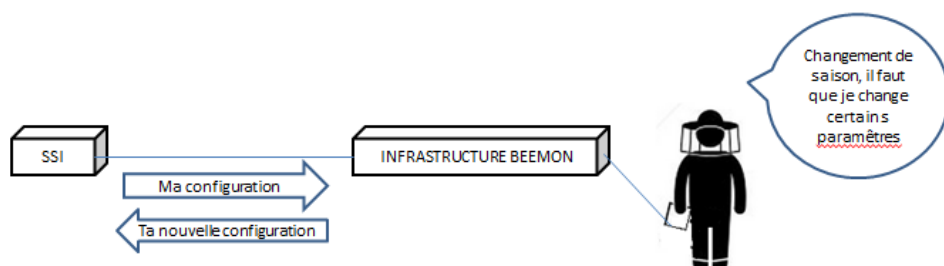


Figure 40 Connexion direct 5 variante

7 Conception

On va dans ce chapitre d'abord présenter un premier design de la carte SSI. Pour avoir une idée des composants qu'il faudra utiliser. Puis on abordera rapidement la communication au sein de l'infrastructure BEEMON

Et enfin on traitera la partie software du SSI, ou l'on va présenter les différents concept qui sont à la base de l'application. A partir de là on montrera comment on a construit l'application

7.1 Architecture matérielle

Voici a quoi pourrait ressembler la carte selon la formule F3. L'idée est d'avoir une carte aux dimensions très réduites embarquant les capteurs nécessaire à la mesure des valeurs internes de la ruche, à savoir la température et l'humidité, pour pouvoir la placée directement dans la ruche. Elle doit également mettre à disposition des ports pour y brancher le reste de l'équipement qui constitue le spectre minimal. Des ports supplémentaires sont encore ajoutés pour d'éventuels autres capteurs

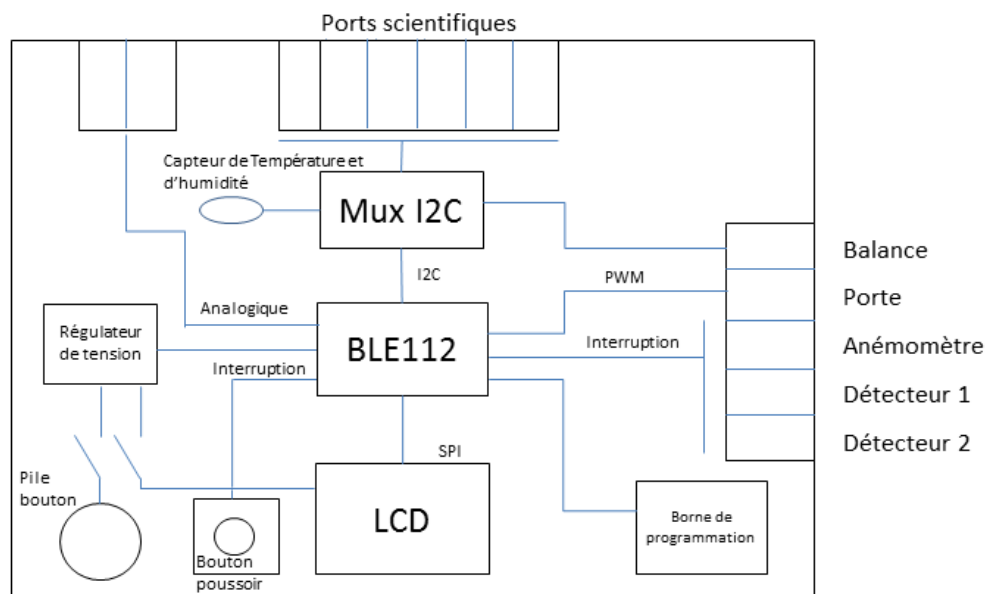


Figure 41 Architecture matérielle du SSI

Les dimensions finales devront être réduites. Seul l'affichage peut poser un problème.


BLE 112 : Composant faisant office de MCU et de module Bluetooth. Il va piloter le reste de la carte

Borne de programmation : Permet de programmer la carte. Elle sera notamment utilisée pour la mise à jour du logiciel

Borne standardisé « beemon » : Il s'agit d'une borne qui contient tous les ports nécessaire au spectre minimal.

Borne extension : Il s'agit d'une borne qui offre des ports I2C supplémentaire pour éventuellement étendre le spectre

Affichage LCD : Permet de notamment de récupérer l'ID temporaire.

 EIA-FR	Projet de semestre 6 Monitoring de ruches 2	Cahier des charges Version 3
---	--	---------------------------------

Capteur(s) embarquée : Permet de relevé la température et l'humidité relative. Le SSI étant une sorte de capteur individuel extensible

Mux I2C : Le BLE 112 n'a qu'une seule interface I2C. Dans le cas où il faudrait brancher des capteurs ayant la même adresse il faut prévoir plusieurs ports.

Un interrupteur d'alimentation : Permet d'enclencher/déclencher le SSI

Un interrupteur pour l'affichage : Permet de l'alimenter qu'en cas de besoin.

Un bouton poussoir : Pour permettre certaine action sur le SSI, comme changer l'information affichée.

Il s'agit ici des premières lignes de conception. Certains composants sont déjà fixés comme le BLE, le capteur, ainsi que le multiplexeur.

7.2 Communication

On va ici rapidement présenter le chemin emprunté pour nos différents.

7.2.1 Chemins dans le système d'informations Beemon

Les différents services que l'on va proposer utiliserons les chemins suivants.

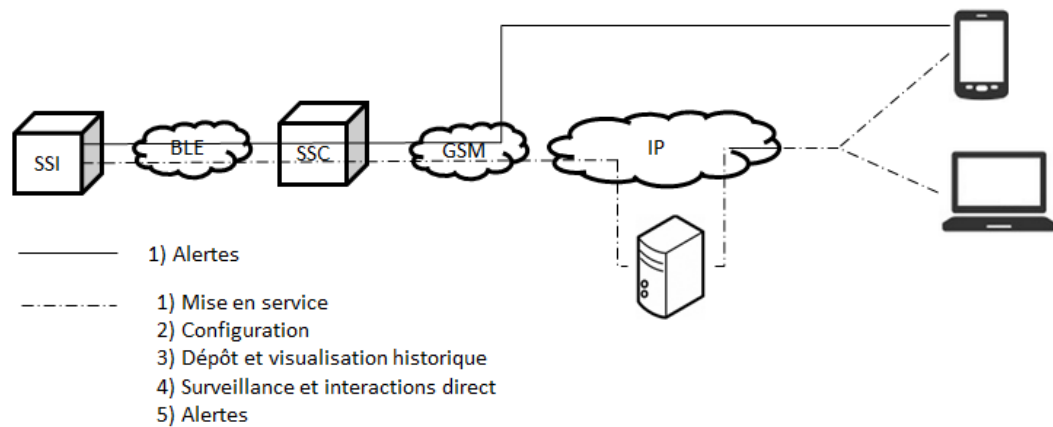


Figure 42 Résumé des chemins empruntés

Le système Beemon imaginé nécessite un serveur d'application, car il doit pouvoir permettre une communication en direct. Ce qui faut comprendre de ce graphique que l'utilisateur passera tout le temps par le serveur d'application, même quand il est proche des SSI., Tout les services, excepté l'alerte utiliserons le SA.

7.3 Software

On va dans un premier temps exposer les concepts qui sont à la base de l'application et ensuite montrer comment ils sont utilisés pour construire l'application.

7.3.1 Bases de l'application

Cette section présente les concepts sur lesquels le SSI est construit.

7.3.1.1 Machine d'état

La machine d'état ci-dessous, qui est simplifiée au maximum pour ce chapitre d'introduction,

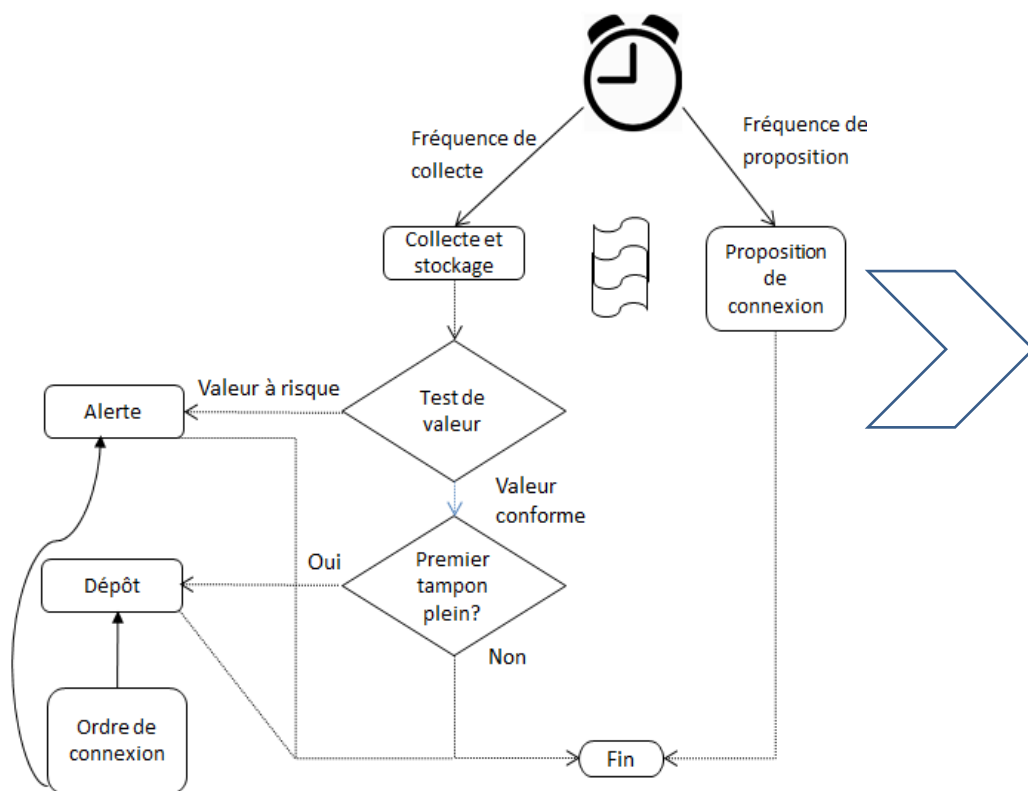


Figure 43 Squelette de l'application

Elle utilise une horloge qui exécute périodiquement une ou l'autre des tâches principales du SSI.

La première est de collecter, stocker temporairement des valeurs avant de les déposer sur le SSC. Si une ou plusieurs valeurs posent problèmes, le SSI va dans un premier temps « ordonner » au SSC de se connecter afin de lui transmettre l'alerte. L'ordre de dépôt, ou plutôt de retrait, est lancé lorsque la première mémoire tampon est pleine.

La seconde tâche du SSI est de proposer de temps en temps au SSC d'établir une connexion. La proposition ne dure que quelque instant alors que l'ordre est maintenu jusqu'à obtention de la connexion. Il faut dans le premier cas implémenter un « time out ».

Les fréquences d'appels, comme d'autres paramètres, sont configurables par l'utilisateur. Cette aspect est abordé dans la prochaine section

7.3.1.2 Système de configuration

Le SSI doit pouvoir être configuré à l'aide d'un terminal distant. La configuration pour pouvoir être interprété par les 2 systèmes doit être un ensemble d'informations formatées. Cette structure est appelée « étiquettes de configuration ».

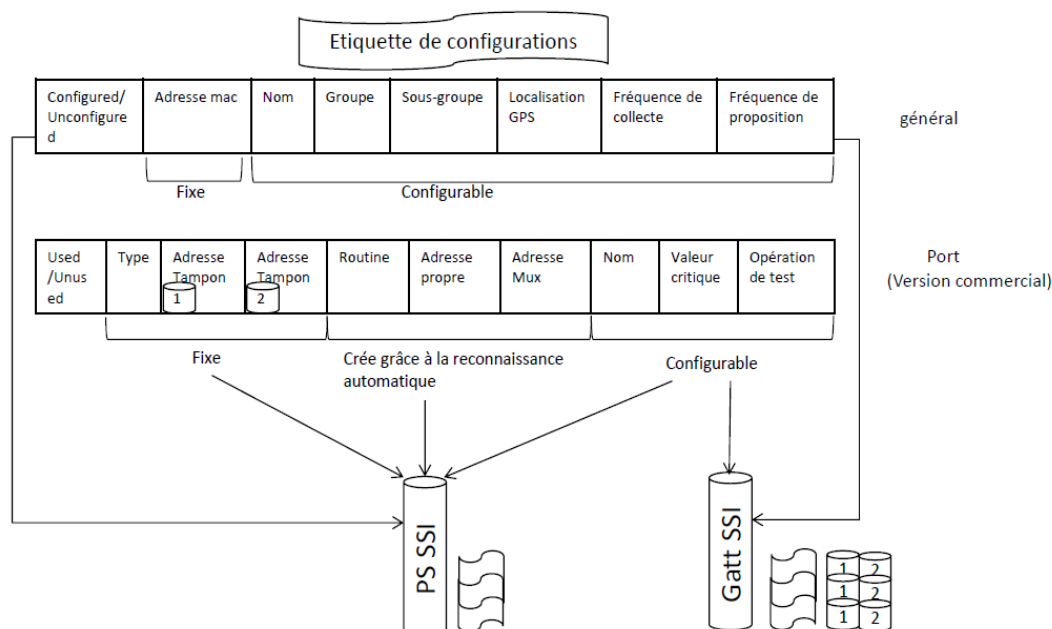


Figure 44 Etiquettes de configurations

Il y a 2 types d'étiquettes de configurations et chacune d'elle a une partie configurable par l'utilisateur, une partie fixe et variable

Générale : Définie le comportement temporel du SSI est contient également un descriptif de celui-ci

Capteur : Il y une étiquettes par capteurs (port A étiquette A). Définie principalement l'accès à celui-ci ainsi que les routines d'extraction et de transformation de l'information. Cette information est créée lors de la détection automatique du capteur lors de son insertion. La partie configurable contient un descriptif du capteur ainsi que le test de mesure accompagnée de valeur de référence.

Le SSI utilise ces étiquettes pour sa configuration. Le terminal les utilise pour faire une représentation de cette configuration.

Les étiquettes sont stockées dans la mémoire Flash pour sauvegarder la configuration et sont mises à disposition dans la Gatt pour que le terminal y ai accès.

La mémoire Flash contient des 127 emplacements mémoire de 32 Byte chacun. Cela va suffire pour les capteurs en ayant le découpage suivant :

On va donc segmenter ces étiquettes sur différentes clés PS.

 EIA-FR	Projet de semestre 6 Monitoring de ruches 2	Cahier des charges Version 3
---	--	---------------------------------

Pour la description et la configuration du SSI on va séparer en 3 clés. Les 2 premières pour la description et la dernière pour la configuration

Pour les ports on va utiliser une clé PS pour la configuration du port et une clé pour la description

Used/Unused	I2C , analogique	Adr Mux	Adr I2C	Routine	Unité physique	Valeur de test	Operateur
				Pour l'extraction des bits et le calcul	C , %	20,5	+
US/UN	I2 AN	0X0X	OXOY	OxO7	C %	2050	+
2	2	2	1	1	1	4	1
ascii	ascii	numérique	numéri que	ascii	ascii	ascii	ascii

Tableau 14 Détail étiquettes de configurations

La troisième ligne correspond à ce qui serait écrit dans l'étiquette. La quatrième au nombre de byte employé. Et enfin le cinquième la façon dont on interprète l'étiquette.

La Gatt elle permet d'avoir des mots de 255 Byte, cependant il est préférable de les séparé pour faciliter l'accès à l'information pour le master

7.3.1.3 *Persistence et mise à disposition de la configuration*

Le SSI qui utilise une pile doit pouvoir préserver la configuration lors du changement de cette dernière. Il faut donc la stocker dans une mémoire non volatile. On utilise ici la mémoire Flash. Lors du redémarrage la configuration doit être rétabli et remise à disposition sur la Gatt.

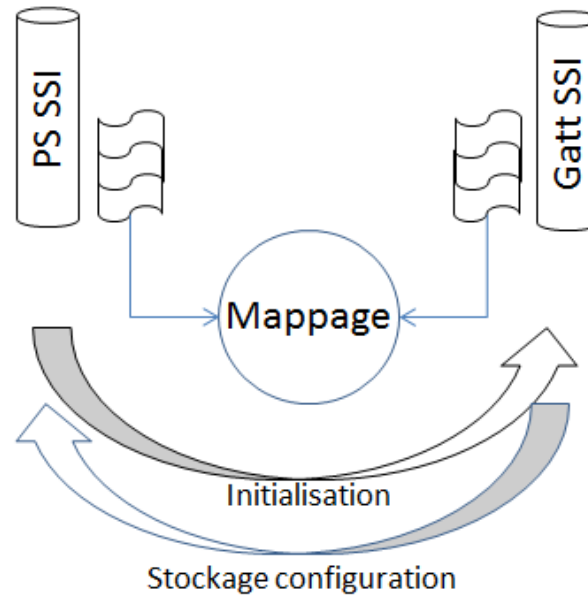


Figure 45 Persistence de la configuration

On donc faire en sorte de faciliter le transfert d'informations de l'une à l'autre. Pour cela on va associer une clé Gatt et une PS à une configuration

Le but étant d'avoir ensuite jouer avec un pointeur commun pour faciliter par exemple le travail de collecte.

7.3.1.4 Les raisons de se connecter

Les ordres et propositions utilisent les messages d'advertising du Bluetooth Low Energy. Ces messages ont une partie utilisable pour transmettre des informations d'application. Elle va servir à indiquer au SSC la raison pour laquelle le SSI lui propose ou lui intime l'ordre de se connecter.

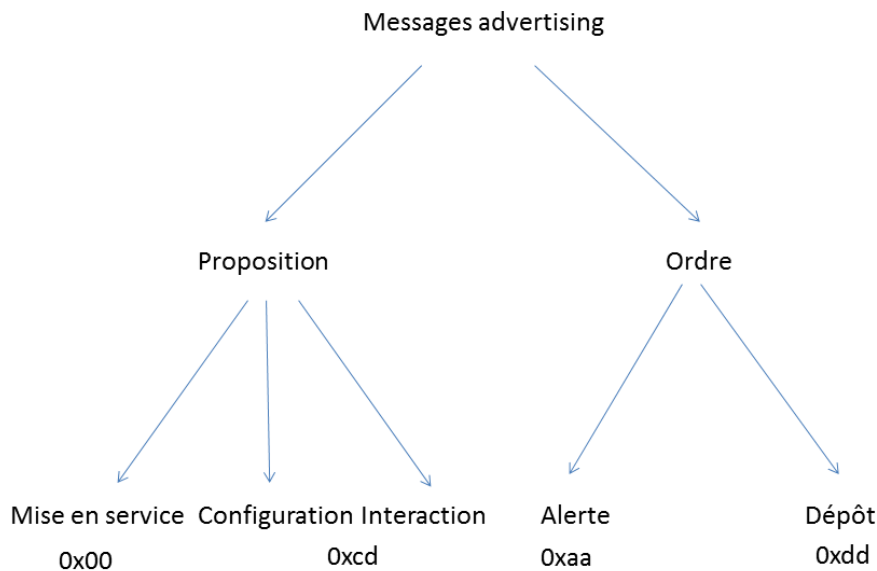


Figure 46 Message d'advertising

La proposition permet d'implémenter la configuration et l'interaction. Elle est émise périodiquement et peut être refusée.

Un ordre de connexion peut être émis pour lorsqu'un le SSI rencontre un problème, lorsqu'il doit déposer les mesures ou lorsque l'utilisateur souhaite le mettre en service. La première raison est prioritaire sur les autres.

Les ordres ont bien entendu la priorité sur les propositions. Une connexion peut donc être rompue pour en privilégier une autre

7.3.2 Mise en service et configuration

On va maintenant définir comment va se passer la procédure de mise en service au niveau des échanges entre le SSI, l'infrastructure BEEMON et finalement le terminal de l'utilisateur.

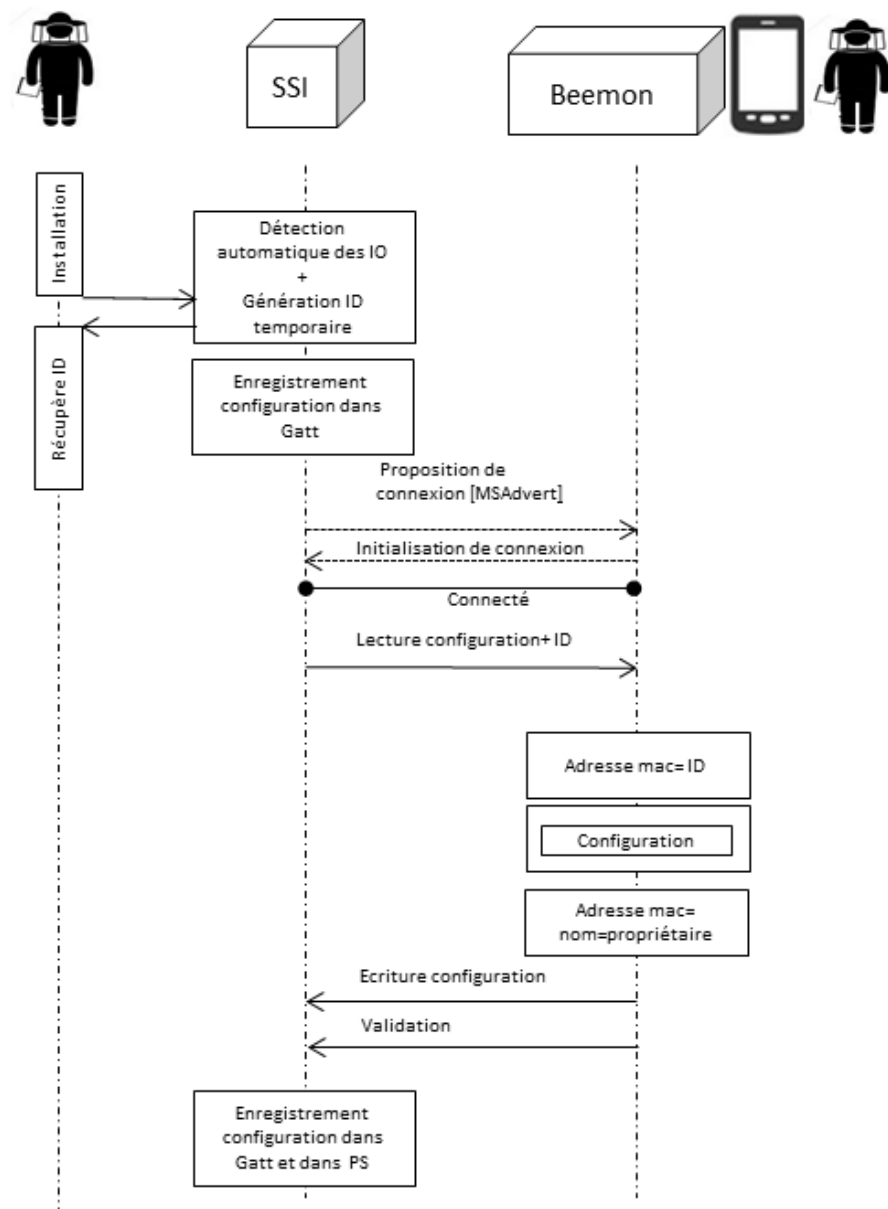



Figure 47 Mise en service et configuration

 EIA-FR	Projet de semestre 6 Monitoring de ruches 2	Cahier des charges Version 3
---	--	---------------------------------

- 1) L'utilisateur branche un ou plusieurs dispositifs, une fois l'installation matériel effectué il presse le bouton de lancement du SSI (ou simplement l'alimente)
- 2) Le SSI va sonder tous les ports en essayant pour chacun d'eux toutes les adresses I2C jusqu'à obtention d'une réponse. Après avoir établis qu'elle port est utilisée et part qu'elle capteur, il enregistre cette configuration dans la Gatt.
- 3) le SSI génère une ID temporaire qui transmet à l'utilisateur via un affichage LCD. L'ID (code) pourrait prendre la forme suivante ZB5Y.
- 4) Le SSI se connecte au serveur d'application Beemon via le SSC et s'enregistre avec l'ID généré accompagné de son adresse MAC. Chaque CC2540 a une adresse mac unique¹²
- 5) L'apiculteur se connecte à son compte Beemon et saisit l'ID dans une barre de recherche
- 6) Le serveur d'application récupère le SSI, le lie au compte.
- 7) Il récupère la configuration du SSI en lisant dans la Gatt et l'affiche à l'utilisateur sous une forme visuelle.
- 8) L'utilisateur entre les paramètres de configuration et les valide
- 9) Le serveur les écrit dans la Gatt. Puis envoie le signal de validation (qui correspond à écrire dans la Gatt)

¹² K.Gautham, G.Raghav, V.Krishnamurthy, N.R.Raajan. *Personnel Security System using Bluetooth*. Mai 2013.
CD : ..\Documentation technique\BLE\CD : ..\Documentation
technique\BLE\BLE112_Bluetooth_Smart_Configuration_Guide_v33.pdf P.2

Voici comment pourrait être découpée la procédure, elle est sujette à caution pour le moment et est mise à titre indicative.

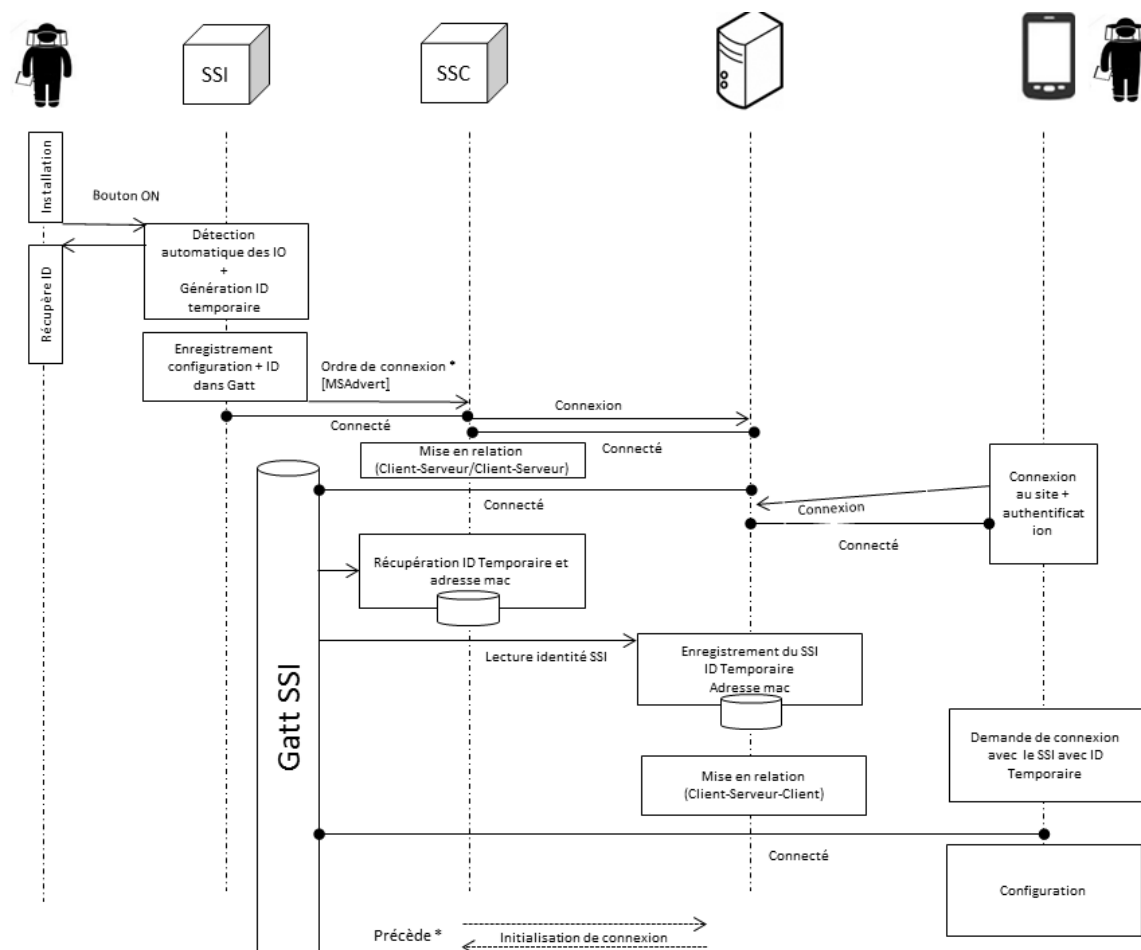


Figure 48 Mise en service et configuration vue complète 1

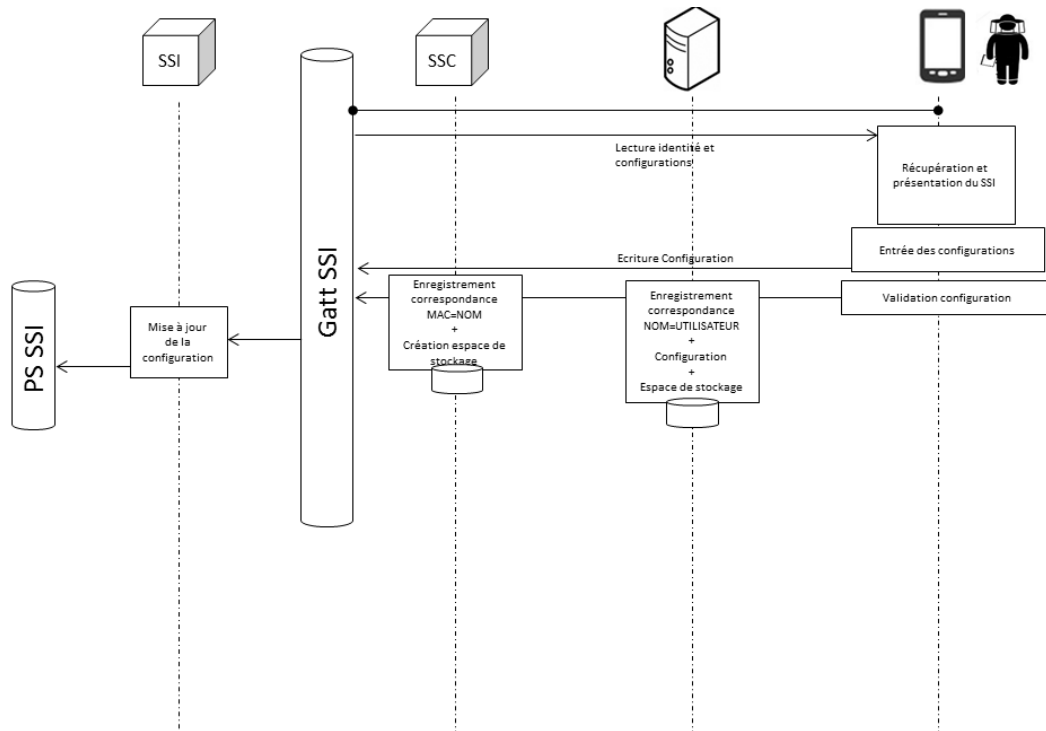


Figure 49 Mise en service et configuration vue complète 2

7.3.3 Collecte et dépôt/alerte

On va utiliser pour la collecte, le traitement, et dépôt/alerte les étiquettes de configurations présenter plus haut.

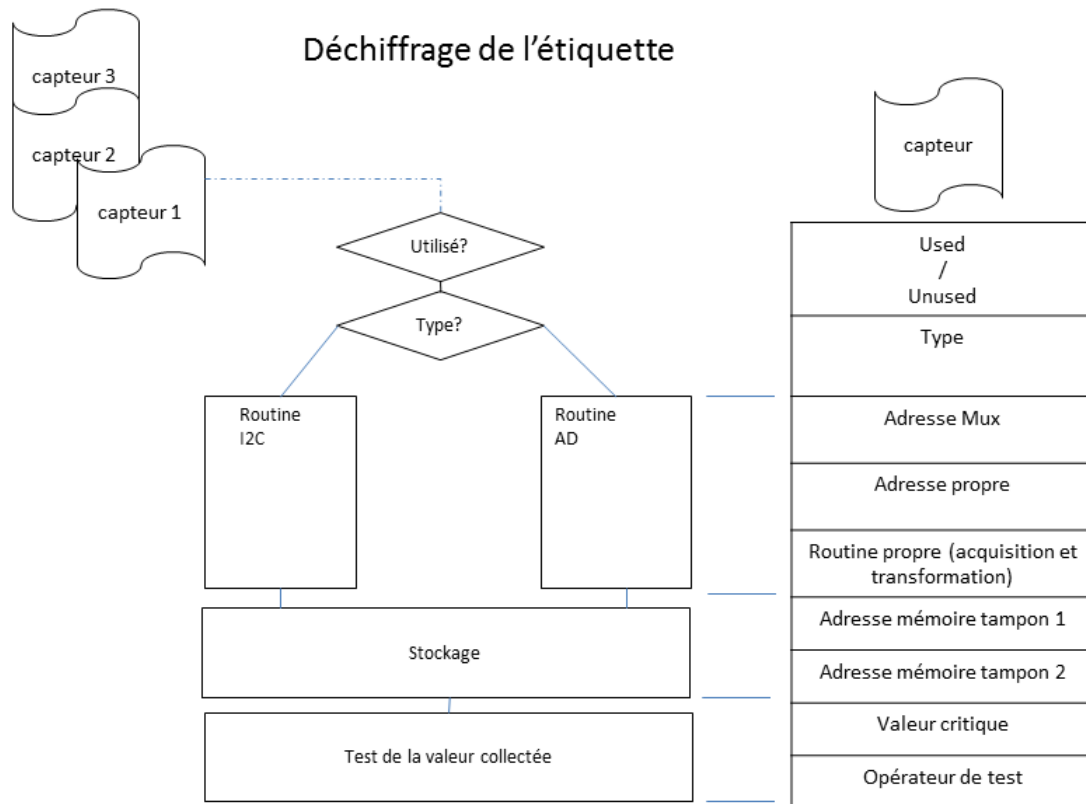


Figure 50 Processus de collecte

Le principe est que pour chaque port il y a une étiquette et lors de la collecte on les évalue toutes.

- 1) Le SSI déchiffre chacun d'entre elles pour savoir comment acquérir la valeur et à quoi elle doit être comparée pour vérifier si elle conforme.
- 2) Le SSI stocke la valeur dans Gatt l'attribut « mémoire tampon » de ce port.
- 3) Le SSI teste la valeur mesurée à celle stockée dans l'étiquette

On va maintenant comment se passe le dépôt

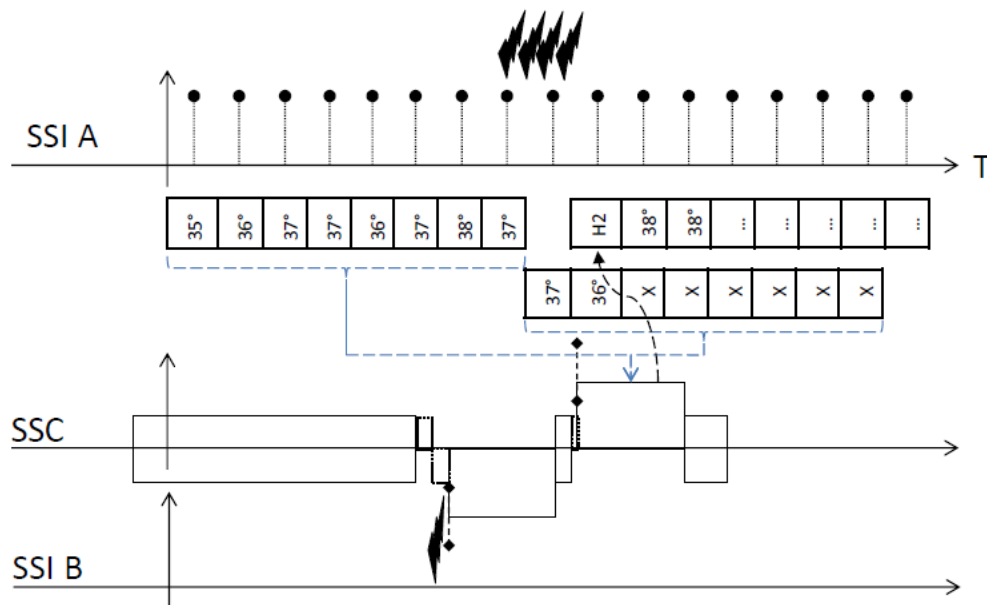


Figure 51 Comportement temporel du dépôt

Le moment de déposer survient lorsque le premier buffer est plein. Le SSI envoie des ordres de connexion au SSC, qui y répondra dès qu'il le peut. En attendant le SSI continue son travail de collecte en stockant les valeurs dans une seconde mémoire tampon.

Lorsque le SSC établit la connexion, il va lire les 2 Buffers. Il récupère dans le premier l'heure du dernier dépôt, et comme la configuration est mise à disposition, il va y lire la fréquence de collecte. Avec ces informations et l'heure actuelle on peut un graphique temporelle.

Remarque : Comme pour le moment le dépôt est fait en une seule fois pour tous les ports, l'heure du dernier dépôt pourrait, comme pour la fréquence e stocké dans la Gatt

La partie basse est juste là pour expliquer que le SSC ne peut pas répondre tout de suite au SSI A.

Cela correspond au cas où le SSC ne pourrait être connecté qu'à 2 SSI à la fois. Ainsi les états milieu correspondant à l'état scanning. Les milieux pointillés correspondent au moment où le SSC évalue la requête. L'état haut correspond à l'état connecté.

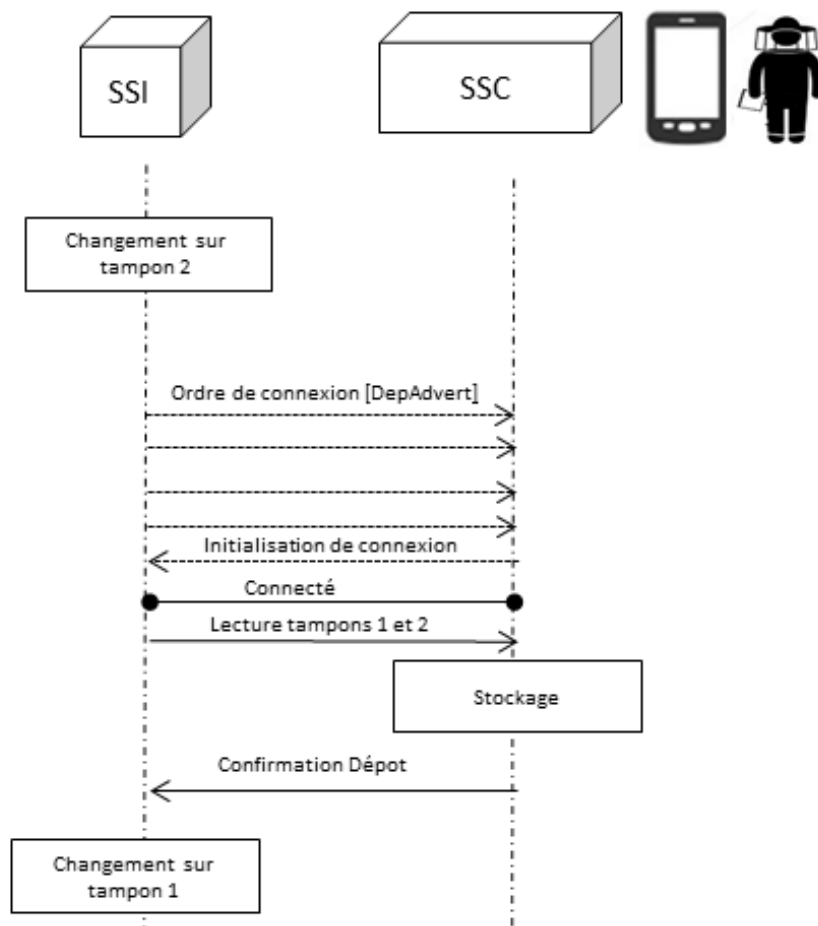


Figure 52 Dépôt variante advertising

Au moment de déposer le SSI envoie les advertising avec la raison « dépôt ». Le SSC établit la connexion et va lire les tampons dans la Gatt. Il confirme cette lecture et le SSI repasse sur le tampon 1

On va dans la prochaine exposé une variante qui utilise les « indicate ». I

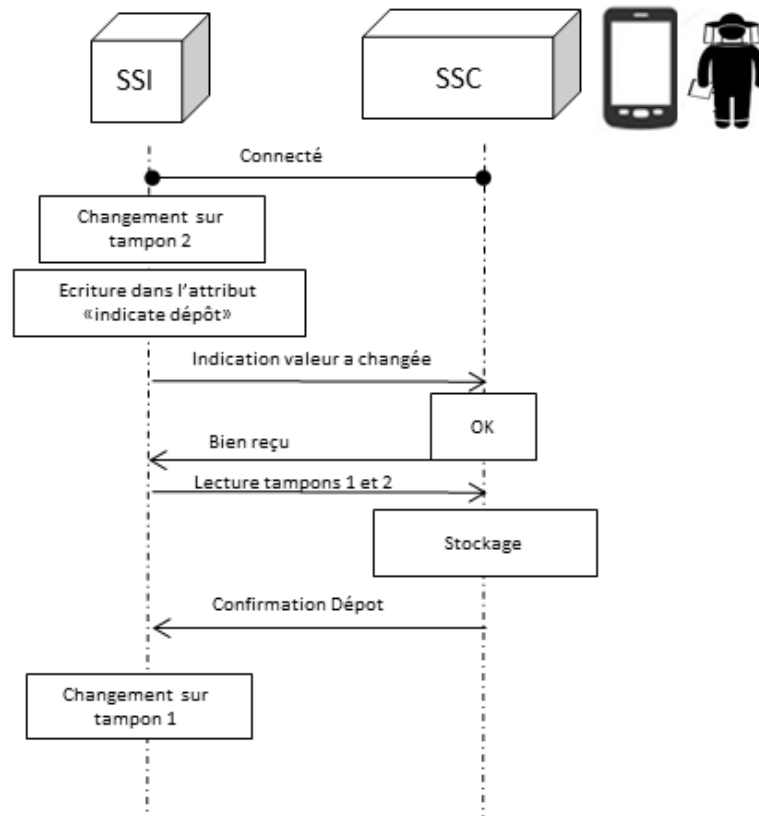


Figure 53 Dépôt variante indicate

Il s'agit ici d'une variante qui correspond au cas où le dépôt doit avoir lieu alors que le SSI et le SSC sont déjà connectés. Pour indiquer au SSC qu'il faut aller lire les tampons on va utiliser une propriété des attributs (characteristic) qui s'appelle « indicate ». Elle permet au serveur Gatt, le SSI d'indiquer qu'une valeur a changé. Le reste se déroule comme avant

Pour l'alerte les mêmes principes que pour les dépôts sont appliqués pour l'alerte, que le SSI soit connecté ou non au SSC. La différence c'est qu'on va indiquer quel port. Le SSC ira ainsi lire uniquement les tampons de ce port.

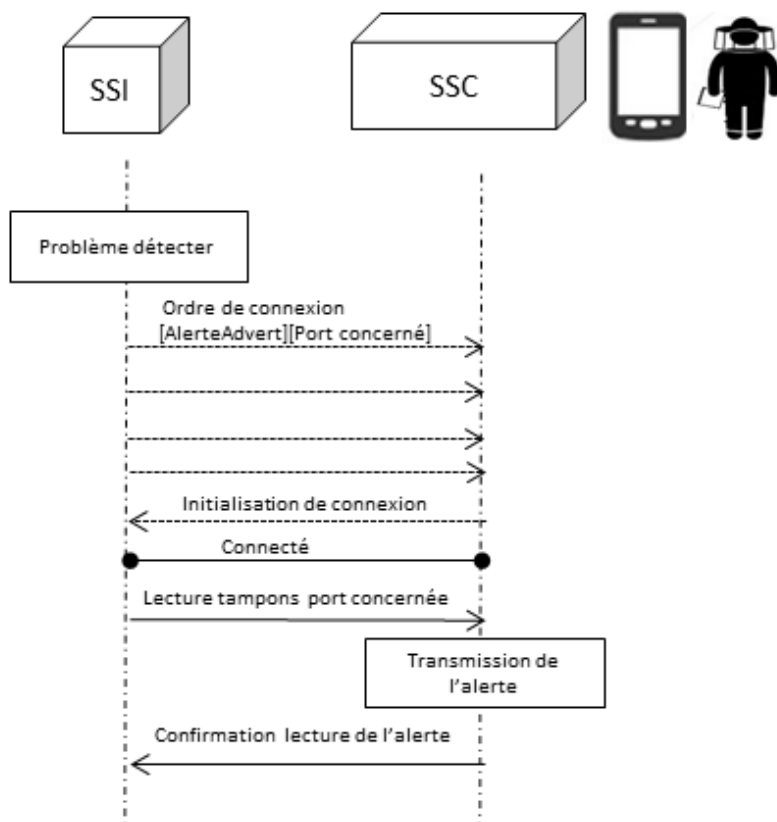


Figure 54 Alerte problème variante Advertising

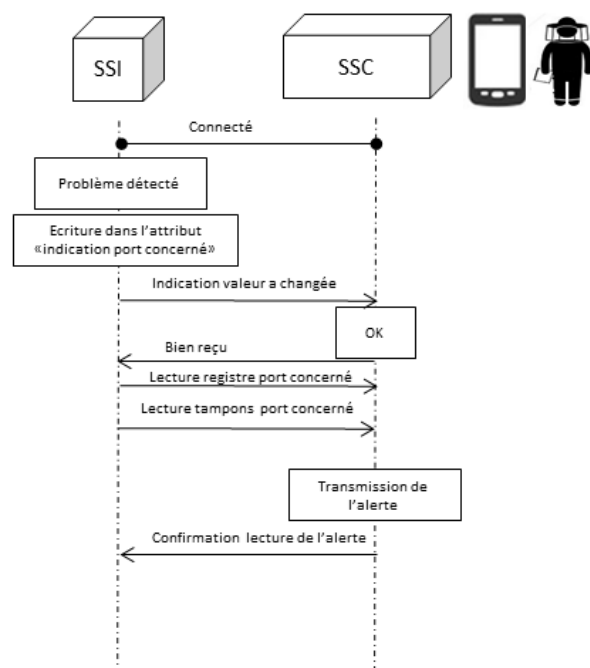


Figure 55 Alerte variante indicate



Projet de semestre 6
Monitoring de ruches 2

Cahier des charges
Version 3

8 Réalisation prototype fonctionnel

8.1 Matériel

On va d'abord réaliser le prototype fonctionnel sur la carte de développement DKBLE112 de BlueGiga utilisant le BLE112. Ce dernier ou son successeur sera à la base du circuit industrialisable. Le langage de programmation utilisé est BGScript.

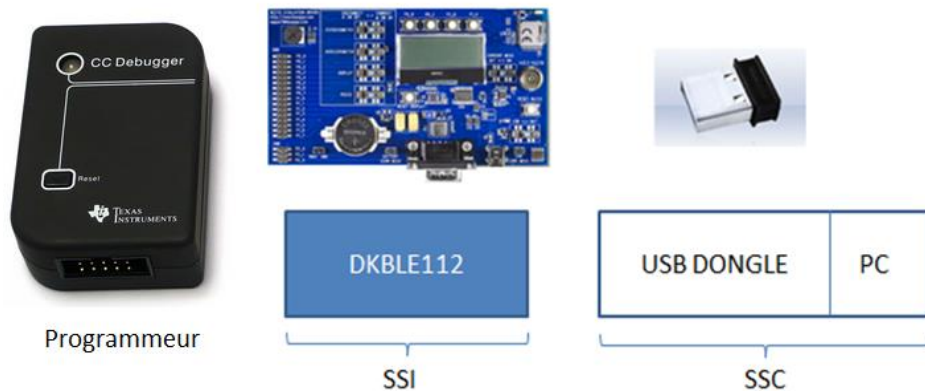


Figure 56 Matériel de base du prototype fonctionnel

8.2 Environnement de développement

Voici l'essentiel des outils utilisés lors de la réalisation. Le premier tableau contient ceux utilisés pour produire et le second pour vérifier.

BLE Update	Permet de programmer le DKBLE112
BGBuild	Permet de compiler le code V 1.2.2
Générateur UUID	Permet de générer des IDs pour les services et caractéristiques Gatt ¹³

Tableau 15 Outils de productions

Blegui	Permet de communiquer avec le DKBLE112 grâce au Dongle V1.2.2
Convertisseur Hexa-ascii	Utilisé pour décoder l'information reçue dans BLEGUI ¹⁴
Affichage LCD	Permet de déboguer temporellement l'application

Tableau 16 Outils de vérification

¹³ guidgenerator. *Online GUID Generator*. s.d. <http://guidgenerator.com/> (accès le Avril 15, 2014).

¹⁴ rapidtables.com. *Hex to ASCII converter*. s.d. <http://www.rapidtables.com/convert/number/hex-to-ascii.htm> (accès le Avril 15, 2014).

8.3 Programmation du BLE112

Une application BGSCRIPT nécessite les fichiers suivants pour pouvoir compiler

Project.xml : Fichiers qui lie tous les fichiers ci-dessous pour la compilation. Veiller à reprendre les mêmes noms de fichiers

CodeApplication.bgs : C'est dans ce fichier qu'écrit le code l'application

Gatt.xml : C'est dans ce fichier qu'on définit la structure de données de l'application. On y définit les attributs que l'on a besoins ainsi que les droits sur ceux-ci-

Hardware.xml : C'est dans ce fichier qu'on définit le matériel nécessaire à l'application ainsi que leurs paramètres (UART, USB, BLE, Pull-up).

Sous Windows pour compiler il faut :

- 1) Ouvrir une boîte de commande (cmd.exe)
- 2) Aller dans le dossier du projet : **cd chemin projet**
- 3) Compiler avec BGscript en choisissant le fichier « project.xml »: **../bin/bgbuild.exe project.xml**

Après compilation on obtient :

out.hex : Il s'agit du code compilé. La compilation prend en compte tous les fichiers présenter avant

Attributes.txt : Fichier d'information indiquant le numéro d'handle pour chaque attributs que l'on à définie dans la Gatt

VariableMemoryUsage : Donne un bilan de l'usage mémoire Flash de l'application. On y retrouve toutes les commandes, les attributs, ainsi que les variables.

Il faut maintenant flasher le CC2540. Pour ceci on a besoin de brancher la carte au PC grâce au debugger USB « Texas Instruments ».

Ensuite ouvrir le programme « **BLE SW Updates Tools** », choisir le fichier **out.hex** et presser sur écrire.

Le BLE112 est maintenant programmé.

8.4 Débogage du BLE112

Pour déboguer l'application, donc savoir ce qu'il se passe on a utilisé conjointement l'affichage LCD et l'écriture/lecture dans la Gatt. On pouvait ainsi voir les valeurs évalué depuis Blegui.

8.5 Software

On va présenter dans ce chapitre comment on s'y ai pris pour réaliser le prototype fonctionnel.

8.5.1 Base de l'application

On a d'abord se présenter comment on a construit les éléments principaux de l'application

8.5.1.1 Machine d'état

```

event hardware_soft_timer(handle)

#Horloge principale
if handle=1 then
    Compteur_Acquisition=Compteur_Acquisition+1
    Compteur_Interaction=Compteur_Interaction+1

    # On regarde si c'est le moment de collecter
    if Compteur_Acquisition>=F_Acquisition then
        ContextSSI(0:5)="Acqui"
        Compteur_Acquisition=0
        call Collecte() #Le dépôt est géré dans la collecte
    end if
    # On regarde si c'est le moment de proposer une connexion
    if Compteur_Interaction>=F_Inter_Conf then
        Compteur_Interaction=0
        # On vérifie que l'on est pas déjà connecté
        # ou que le SSI n'envoie pas de proposition en ce moment
        if Connected=0 & Adv_Enable=0 then
            ContextBLE(0:4)="Prop"
            # On envoie des advertissing du type "proposition de
            connexion direct"
            NatureProp=$cc
            call CreateAdvert()
            Adv_Enable=1
            #On stoppe l'horloge de la machine d'état pour lancer le
            Time
            call hardware_set_soft_timer(0,1,0) # Horloge Time_out
            call hardware_set_soft_timer(32768/10,2,0) # Horloge
            Time_out
        end if
    end if
    call LCD_Conf()
end if

# Horloge de Timeout
if handle=2 then
    Time_Out=Time_Out-1
    # Timeout expiré?
    if Time_Out=0 then
        Time_Out=10
        # On arrete les advertising en stoppant le BLE
        # et on entre a nouveau en power mode 3
    
```

```

call gap_set_mode(0,0)
ContextBLE(0:4)="  "
Adv_Enable=0
call LCD_Conf()
# On stoppe l'horloge timeout et on remet la principale
call hardware_set_soft_timer(0,2,0) # Horloge Time_out
call hardware_set_soft_timer(32768*5,1,0) # Horloge
Time_out
# Remarque: Il faudra soustraire le temps passé a
advertiser
# pour ne pas induire de décalage dans le temps
end if
end if
end

```

Code 1 : Machine état

On a donc d'abord réalisé la machine d'état de l'application qui on le rappelle a pour tâche principal de d'exécuter périodiquement une ou plusieurs tâches. Dans notre cas il s'agit de la collecte et de l'envoi de propositions. La proposition ne dure pas longtemps, il a fallu donc également implémenter un Timeout

L'idéal aurait été d'avoir des horloges séparés tournant en même tant pour la machine d'état et le Timeout de la proposition. Mais ce n'est pas possible car le BLE112 ne permet que de faire tourner une horloge à la fois.¹⁵

Mais on peut configurer plusieurs horloges et leur attribuées un identifiant appelé « handle ». Ainsi lorsque la routine d'événement d'horloge est réveillée elle sait quelle horloge l'a fait.

Dans notre cas on a défini l'horloge ayant l'handle 1 comme étant la principale, l'horloge avec handle 2 est utilisée pour le Timeout.

Le code correspondant au lancement de la proposition est appelé par l'horloge ayant l'handle 1, à la fin de ce code on stoppe cette horloge actuelle et on démarre l'autre. Maintenant le code correspondant à l'handle 2 est exécuté. A la fin du timeout on stoppe l'horloge actuelle (2) et on redémarre celle de la machine d'état (1).

¹⁵ Bluegiga. BGSCRIPT SCRIPTING LANGUAGE DEVELOPER GUIDE. Mars 2014. CD : ..\Documentation technique\BLE\BLE112_BGScript_developer_guide_v3.4.pdf P.31

8.5.1.2 Persistance et mise à disposition de la configuration

```

*****Mappage PS et
GATT*****
procedure MappageConfiguration()

PS_KEYS_CONFIGURED=$8000      # Status de configuration :
Configuré/non configuré
GATT_KEYS_CONFIGURED=$70

PS_KEYS_SSI_DESC(0:2)=$8001    #Nom SSI, groupe et sous-groupe
GATT_KEYS_SSI_DESC(0:1)=3
. . . . .
PS_KEYS_SSI_CONF(0:2)= $8006   #Fréquences de proposition et de
dépôt
GATT_KEYS_SSI_CONF(0:1)=16
. . . . .
PS_KEYS_PORT_CONF(0:2)= $8008  #Configuration port 1
GATT_KEYS_PORT_CONF(0:1)=23
. . . . .
GATT_DEPT_KEYS(0:1)=29         #Port 1 -> Tampon 1
GATT_DEPT_KEYS(1:1)=32        #Port 1 -> Tampon 2
. . . . .

end

```

Code 2 : Mappage de la configuration

Pour pouvoir passer facilement la configuration de la Gatt à la PS et vice-versa il a fallu lier leurs adresses. Pour cela on a donné à chaque description, configuration, mémoire tampon une clé Gatt et PS. Elles sont rangées dans leurs buffer respectives dans le même ordre de sorte qu'on puisse les lier facilement avec un pointeur.

La mémoire flash contient 126 emplacement de 32 Byte chacun commençant à l'adresse 0x8000 et allant jusqu'à 0x807f ¹⁶

```

*****Stockage et mise à jour de la
configuration*****
procedure Update_Configuration()
Stockage de la description
call attributes_read(GATT_KEYS_SSI_DESC(0:1), 0) (Handle, Offset,
Result, Value_len, Value_data(0:Value_len))
call flash_ps_save(PS_KEYS_SSI_DESC(0:2), Value_len,
Value_data(0:Value_len))

#Stockage des configuratons générales, et mis à jour des variables

```

¹⁶ Bluegiga. BGSCRIPT SCRIPTING LANGUAGE DEVELOPER GUIDE. Mars 2014. CD : ..\Documentation technique\BLE\BLE112_BGScript_developer_guide_v3.4.pdf P.38

```

de fonctionnement
call attributes_read(GATT_KEYS_SSI_CONF(1:1), 0) (Handle, Offset,
Result, Value_len, Value_data(0:Value_len)) #KO
call flash_ps_save(PS_KEYS_SSI_CONF(2:2), Value_len,
Value_data(0:Value_len))
F_Inter_Conf=Value_data(0:Value_len)

. . . . .
# Stockage des configurations de port
call attributes_read(GATT_KEYS_PORT_CONF(0:1), 0) (Handle, Offset,
Result, Value_len, Value_data(0:Value_len)) #KO
call flash_ps_save(PS_KEYS_PORT_CONF(0:2), Value_len,
Value_data(0:Value_len))
. . . . .

```

Code 3: Stockage de la nouvelle configuration et mise à jour des variables d'environnement

Le code ci-dessous récupère la configuration qui est sur la GATT, qui vient peut-être d'être modifié par l'utilisateur, et la stocke dans la Flash. Il faudrait utiliser un pointeur comme pour la afin d'éviter de faire ces entrées manuellement.

```

*****Initialisation de la
configuration*****
procedure InitiationSSI()
#Mise à disposition de la description dans la Gatt
call flash_ps_load(PS_KEYS_SSI_DESC(0:2)) (Result, Value_len,
Value_data(0:Value_len))
call attributes_write(GATT_KEYS_SSI_DESC(0:1), 0, Value_len,
Value_data(0:Value_len))
. . . . .
#MàD dans la Gatt et intialisation des variables de fonctionnnement
call flash_ps_load(PS_KEYS_SSI_CONF(2:2)) (Result, Value_len,
Value_data(0:Value_len))
call attributes_write(GATT_KEYS_SSI_CONF(1:1), 0, Value_len,
Value_data(0:Value_len))
F_Inter_Conf=Value_data(0:Value_len)

. . . . .

#MàD dans la Gatt
call flash_ps_load(PS_KEYS_PORT_CONF(0:2)) (Result, Value_len,
Value_data(0:Value_len))
call attributes_write(GATT_KEYS_PORT_CONF(0:1), 0, Value_len,
Value_data(0:Value_len))
. . . . .

end

```

Code 4: Récupération de la configuration dans la Flash pour la mettre dans le Gatt

Ce code fait l'inverse du précédent. Il récupère la configuration stockés dans la PS pour dans un premier initialiser le SSI puis dans un second mettre cette configuration à disposition sur la Gatt.

Pour attribuer les bons numéro dans la Gatt il faut compiler le programme. A cemoment il génère une fichier qui contient une liste de correspondance

On a ci-dessous le fichier de configuration de la Gatt. On a défini un service par capteur. On y

```
<service uuid="624e957f-cb42-4cd6-bacc-84aeb898f69b" advertise="true">
  <description>PORT 1</description>

  <!-- custom write-only characteristic for setting mode -->
  <characteristic uuid="e4c937b3-7f6d-41f9-b997-40c561f4453b" id="C1CO">
    <description>Configuration</description>
    <properties read="true" write="true" />
    <value variable_length="true" length="10" ></value>
  </characteristic>

  <!-- custom read/write characteristic for getting/setting I/O port
status -->
  <characteristic uuid="df342b03-53f9-43b4-acb6-62a63ca0615a" id="C1VC">
    <description>ValeurCritique</description>
    <properties read="true" write="true" indicate="true" />
    <value variable_length="true" length="2" ></value>
  </characteristic>

  <!-- custom write-only characteristic for setting mode -->
  <characteristic uuid="e4c937b3-7f6d-41f9-b997-40c561f4453b" id="C1T1">
    <description>Port1_Tampon1</description>
    <properties read="true" write="true" />
    <value variable_length="true" length="255" ></value>
  </characteristic>

  <!-- custom read/write characteristic for getting/setting I/O port
status -->
  <characteristic uuid="df342b03-53f9-43b4-acb6-62a63ca0615a" id="C1T2">
    <description>Port1_Tampon2</description>
    <properties read="true" write="true" />
    <value variable_length="true" length="255" ></value>
  </characteristic>
</service>
```

Code 5 : Services et caractéristique Gatt pour les ports

Configuration : C'est là que sera stockée l'étiquette du port pour la mettre à disposition du SSC. On lui a définie pour le moment une longueur variable de façon a facilité le développement. Dans l'absolu elle sera fixe

Valeur critique : Il s'agit en fait d'un attribut qui va permettre, grâce, à l'indicate de savoir quand retire. Cette fonction n'as pas était vérifié car on ne savait pas vraiment comment le faire avec Bleguil.

On trouve ensuite les mémoires tampons 1 et 2 , qui ont chacune une longueur de 255 Bytes

8.5.1.3 Envoie d'advertissing personnalisés

```

Procedure CreateAdvert ()
if Connected=0 then

    . . . . .

    # Pas utilisé
    Adv_data(10:1) = $00
    #
    Adv_data(11:1) = NatureProp
    # Numero du capteur posant problème
    Adv_data(12:1) = $00
    call gap_set_adv_data(0, 13, Adv_data(0:13))
end if
end

                                Programme appelant

    # On envoie des advertissing du type "proposition de connexion
direct"
    NatureProp=$cd
    call CreateAdvert ()
    Adv_Enable=1

```

Code 6 : Envoie d'advert personnalisés

Ce code représente comment on peut utiliser les trois derniers Byte du message d'advertissing. On l'utilise pour indiquer au SSC la raison de la proposition ou de l'ordre. Cela lui permet de considérer la requête à base d'une information. Il peut dans ce cas accepter ou non d'établir une connexion.

Le SSC lorsqu'il reçoit un message d'advertissing réveil un évènement «appelé GAP_SCAN_RESPONSE » ou il a accès notamment à l'adresse MAC de l'advertiser et aux données contenant dans le message d'advertissing.

8.5.2 Mise en service et configuration

La mise en service et configuration se repose essentiellement sur les bases présentées plus haut. Une fois utilisées le code restant n'est pas si compliqué.

Lorsque l'utilisateur à entrer ses configurations, il envoie un signal au SSI en écrivant dans un des attributs de ce dernier. L'attribut qui est à ici le numéro 70 pourrait s'appeler « Configuration effectuée ». Le SSI une fois qu'il a reçu ce signal sauvegarde la configuration dans la PS et met à jour les variables de fonctionnement (voir code « UpdateConfiguration »).

```
event attributes_value(connection, reason, handle, offset, value_len,
value)

#Signal que la configuration a été effectuée
#On met à jour la configuration
if handle=70 & value(0:4)="DONE" then    #Signal configuration
effectuée
    call flash_ps_save(PS_KEYS_CONFIGURED, value_len,
value(offset:value_len))
    call Update_Configuration()
    ContextSSI(0:5)="Conf"
end if

. . . . .

end
```

Code 7 : Validation configuration effectuée

8.5.3 Collecte et dépôt

```

procedure Collecte()
p=0
i=0
Nombre_port=2
#On regarde s'il faut enregistrer dans le premier ou second tampon
if Tampon1_full=1 then
    i=1
end if
# On traite tout les ports
while Nombre_port>0
    # On extrait récupère l'étiquette du port courant
    call flash_ps_load(PS_KEYS_PORT_CONF(p:2)) (Result, Value_len,
Value_data(0:Value_len))
    # On vérifie que le port est utilisé
    if Value_data(0:2)="US" then
        # On regarde quel type de port il s'agit
        if Value_data(2:2)="I2" then
            # Capteur I2C
            # Pas fait, on regarde de quel capteur il s'agit
            # Puis on appelle la routine correspondante
            call HIH6121_TemperatureHumidity ()
        end if
        if Value_data(2:2)="AN" then
            # Capteur analogique
            # Rien pour le moment
        end if
        #Stockage de la valeur mesurée dans le tampon
        correspondant a ce port
        call attributes_write( GATT_DEPT_KEYS(i:1)
,Offset_gatt,6,Valeur_Imprimable(0:6))
    end if
    i=i+2
    p=p+2
    Nombre_port=Nombre_port-1
end while

Offset_gatt=Offset_gatt+6
#On test la charge du premier tampon
#S'il est plein on ordonne au SSC de se connecter
#Et on passe du tampon 1 au tampon 2
if Offset_gatt>=255 then
    Offset_gatt=0
    if Tampon1_full=0 then
        Tampon1_full=1
        Adv_Enable=1
        NatureProp=$dd
        call CreateAdvert()
        ContextSSI(0:4)="Depo"
        ContextBLE(0:4)="Prop"
    end if
end if

```

Code 8 : Collecte et dépôt

On va utiliser dans la collecte les étiquettes de configurations de ports. Pour cela on fait évaluer l'étiquette dans l'ordre pour savoir si le port est utilisé, à qu'elle type il appartient. Puis on appelle la routine correspondante.

Une fois la valeur mesurée on la stocke dans la Gatt. Si le premier buffer est plein on envoie un ordre de dépôt au SSC et on passe au tampon numéro 2.

```
event attributes_value(connection, reason, handle, offset,  
value_len, value)  
  
. . . . .  
  
#Signal que le dépôt à été effectué  
#On repasse au tampon 1  
if handle=73 & value(0:4)="DEPD" then  
    Offset_gatt=0  
    Tampon1_full=0  
end if  
end
```

Code 9 : Validation dépôt effectué

Une fois que le SSC a lu tous les tampons il le notifie au SSI, grâce au même système que pour la configuration, et ce dernier passe du tampon 2 au tampon 1 avec un offset de 0.

8.5.3.1 Lecture des valeurs par I2C du HIH6121

On a branché la patte SDA du capteur à la borne p1_6¹⁷ et le CLK sur la patte p1_7. Ensuite on a branché un condensateur de 220nF¹⁸ entre le VCC et le Ground.

On a d'abord configuré résistance pull-up sur le port 1 avec cette ligne dans le fichier hardware.xml.

```
<port index="1" tristate="0" pull="up" />
```

Les pull-up¹⁹ ne peuvent pas être configuré pour chaque pin individuellement. Lorsqu'on active la pull-up sur le port 1, tous les pins du port 1 ont la pull-u

```
procedure HIH6121_TemperatureHumidity (selector)
#(Adresse I2C* pour passer de 7 a 8 bit , envoie d'un stop)
call hardware_i2c_write(78,1,0,0) (written)
call hardware_i2c_read(78,1,4) (resulte,i2c_dl,i2c_data)

#Si l'etiquette contient TE, on mesure la température
if Value_data(4:2) ="TE"then
    #On extrait les bits correspondant a la température
    Valeur = (((i2c_data >> 16) & $ff) << 8) | ((i2c_data >> 24) &
$ff)
    #On calcul la valeur réel
    Valeur= ((Valeur*165*10)/(16382*4))-(40*10)
    #On ajoute le bon signe %
    Valeur_Imprimable(4:1) = 67      #"Flottant"
end if

#Si l'etiquette contient HU, on mesure l'humidité
if Value_data(4:2) ="HU"then
    Valeur= ((i2c_data & $ff) << 8)|((i2c_data >> 8) & $ff)
    Valeur= ((Valeur*10*100)/(16382*4))
    Valeur_Imprimable(4:1) = 37      #"%"
end if

#On extrait les dizaine,unité et flottant puis on crée une valeur
#Affichable
Dizaine = (Valeur/ 100)              #2X.X
```

¹⁷ Bluegiga. BGSCRIPT SCRIPTING LANGUAGE DEVELOPER GUIDE. Mars 2014. CD : ..\Documentation technique\BLE\BLE112_BGScript_developer_guide_v3.4.pdf P.25.

¹⁸ Honeywell. Honeywell HumidCon Digital Temperature Humidity Sensors. Juillet 2012. CD : ..\Documentation technique\capteur\H6121_Datasheets.pdf P.8

¹⁹ Bluegiga. BLUETOOTH SMART MODULE CONFIGURATION GUIDE. Février 2014. CD : ..\Documentation technique\BLE\BLE112_Bluetooth_Smart_Configuration_Guide_v33.pdf P.20.

```

Unite = (Valeur/ 10) + (Dizaine*-10)           #X4.X
Flottant = (Valeur)      + (Dizaine*-100) + (Unite*-10) #XX.8
Valeur_Imprimable(0:1) = Dizaine + 48 #"2"
Valeur_Imprimable(1:1) = Unite + 48 #"4"
Valeur_Imprimable(2:1) = 46      #"."
Valeur_Imprimable(3:1) = Flottant + 48 #"8"
Valeur_Imprimable(5:1) = 32      #espace
end

```

Code 9 : Acquisition d'une valeur I2C

On a d'abord configuré la fonction I2C de s'adresser au capteur. Pour cela on a pris l'adresse I2C fourni dans la datasheet et on l'a multiplié par deux pour passer d'une adresse 7 à 8bits. Comme cela est demandée dans la figure ci-dessous. Ensuite on a également configuré la fonction pour qu'elle envoie un stop à la fin de la réception des 4 Bytes.

Ensuite il a fallu extraire les parties correspondantes à la température et l'humidité. Pour comprendre le code il faut regarder comment sont organisé les bits dans les réponses I2C et quels sont ceux qui représentent la température et l'humidité.

Figure 4. Humidity and Temperature Data Fetch, Four Byte Data Read

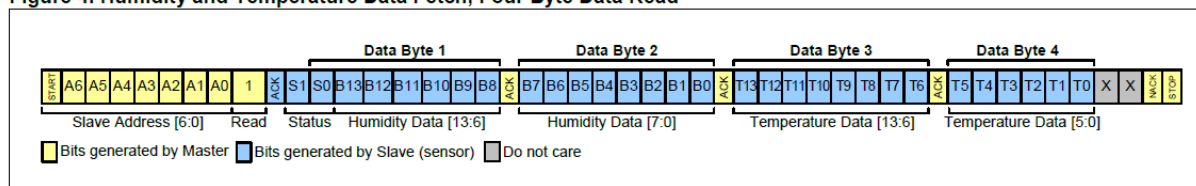


Figure 57 : Bytes transmis par H6121²⁰

On voit sur la figure que les MSB sont à gauche. Les bytes reçus sont imprimé dans ce sens dans la variable de réception (i2c_data). Or Le bit de poids fort, lors de l'évaluation de la variable est considéré comme étant à droite. S'il on imprime la variable on obtient

1e5e63a8

La partie gauche correspond à l'humidité et la partie gauche à la température. Si on prend la partie droite, qu'ont la colle dans la calculette scientifique et qu'on fait le calcul on trouve la bonne température. Mais le BLE112 interprète ces mots hexadécimal dans l'autre sens, si bien que pour lui le 63a8, après l'avoir masqué est décalé à gauche est interprété en oxa863.

²⁰ Honeywell. *I2C Communication with the Honeywell HumidCon* juin 2012. CD : ..\Documentation technique\capteur\H6121_CommunicationI2C P.2 ;3

Il a donc fallu isoler chacune des 4 parties de 8 bits et, et les réassembler pour former 2 mots de 16 bit.

Si on prend le cas de la température l'opération est la suivante

1e5e63a8

Décalage de 16 vers le LSB de bit

63a80000

Masquage avec \$ff

63000000

Puis on revient vers le MSB avec un décalage de 8 bits

006300000

On reprend le mot origine et on lui applique le même principe mais en décalant cette fois de 24

A8000000

A8000000 | 006300000 = a863

Maintenant que les bits ont leurs bon poids, le BLE peut calculer la température avec la formule donnée dans la datasheet.

$$\text{Temperature (}^{\circ}\text{C)} = \frac{\text{Temperature Output Count}}{(2^{14} - 2)} \times 165 - 40$$


Et pour l'humidité

$$\text{Humidity (\%RH)} = \frac{\text{Humidity Output Count}}{(2^{14} - 2)} \times 100\%$$

On a encore multiplié les deux membres de la soustraction car le BLE ne gère pas à proprement parler des floats mais peut les stocker à l'aide d'une mantisse²¹ et d'un exposant. Donc si la valeur de départ est trop petite, le résultat de la division peut donner 0.

$$\text{Valeur} = ((\text{Valeur} * 165 * 10) / (16382 * 4)) - (40 * 10)$$

²¹ Bluegiga. *BGSCRIPT SCRIPTING LANGUAGE DEVELOPER GUIDE*. Mars 2014. CD : ..\Documentation technique\BLE\BLE112_BGScript_developer_guide_v3.4.pdf P.13

 EIA-FR	Projet de semestre 6 Monitoring de ruches 2	Cahier des charges Version 3
---	--	---------------------------------

9 Test et validation

On maintenant vérifier si notre application fonctionne correctement et si on a rempli les objectifs


OK	Fonctionne parfaitement
OK~	Fonctionne parfait mais est incomplet ou doit être associé à une autre partie qui n'est pas encore terminée
KO~	Ne fonctionne pas correctement
KO	Ne fonctionne pas du tout

Tableau 17 Légendes pour les test

1	Machine d'état	OK
2	Proposition et Time out	OK
3	Persistence de la configuration	OK~
4	Mise en service et configuration à distance	OK
5	Interaction direct (au niveau com)	OK~
6	Collecte	OK~
7	Dépôt	OK~
	Alerte	PAS FAIT
8	Acquisition I2C	OK
9	Contrôle PWM	PAS FAIT
10	Compteur d'entrées sorties	KO~ (vous trouverez le code en annexe)

Tableau 18 Test des fonctionnalités

	Améliorations / travail encore a effectué	Description du problème
1	Lorsque le timeout est échu on appelle arrête l'horloge courante et on réveille celle de la machine d'état. Il faudra soustraire le temps passée à attendre le Timeout a la nouvelle horloge pour sa première exécution. Ainsi on n'indura pas de décalage.	
2	-	
3	Il faudrait utiliser une boucle et un pointeur pour mettre à jour la configuration dans la Gatt et la reprendre lors de l'initialisation. Cela au lieu de le faire manuellement.	
4	-	
5	On a pu piloter l'affichage du LCD en simulant la porte d'entrée. Il faut maintenant faire le PWM pour pouvoir lier les deux.	
6	Tous les champs de l'étiquette de configuration ne sont pas encore évalués, comme celui qui devrait contenir la valeur critique. Ce explique pourquoi l'alerte n'as pas été réalisé.	
7	L'indication du dépôt par advert est réalisé, il reste cependant à faire la variante en mode connecté. Cela n'as pas était fait car on ne savait pas comment le simuler.	
8	-	

	Projet de semestre 6 Monitoring de ruches 2	Cahier des charges Version 3
---	--	---------------------------------

9	BGSscript propose une commande pour générer facilement un signal PWM	
10		

Si on prend les objectifs fixés au début du semestre :

Architecture hardware et software pour le système SSI	réussi
Spécification de l'ensemble des systèmes et interne du SSI	réussi
Réaliser un prototype fonctionnel	réussi
Réalisation d'une carte industrialisable (opt)	Pas réussi

Tableau 19 Validation des objectifs

10 Conclusion

10.1 Problèmes rencontrés

1) Trop de temps consacré à l'analyse au détriment de la réalisation

C'est la plus grande erreur commise durant ce projet, et qui m'a mis dans une position délicate. Il y a un peu plus d'un mois je n'avais encore rien conçu ni réalisé alors que ce semestre était principalement axé sur la réalisation. Alors qu'au premier semestre le projet était orienté sur la partie théorique, c'est en réalité le contraire qui s'est produit.

Réaliser dans de telles conditions est très difficile car on est stressé et on ne se donne pas la peine de réfléchir.

1) Résultat de masquage qui varie d'une exécution à l'autre

Pour pouvoir lire la température ou l'humidité, il faut pouvoir extraire la portion de bits qui leur correspond puis leur donner le bon point en les décalant.

Le code ci-dessous avait pour but de comprendre s'il était possible de ne préserver qu'une partie d'un mot de 32 bits avec le masquage, mais surtout de comprendre comment était ordonnée l'information I2C dans ce mot et comment le masque était appliqué. A cet effet on a imprimé le résultat dans la Gatt.


```
call
hardware_i2c_read(78,1,4) (resulte,i2c_dl,i
2c_data)
call
attributes_write(OP1,offset_gatt,4,i2c_dat
a) #25

v=i2c_data & (4294967295-65535) # Masque
applique, partie droite restante(63a8)
call attributes_write(OP2,offset_gatt,4,v)
#28
```

La première constatation est qu'il aurait été plus judicieux de travailler avec des valeurs hexadécimales. Mais si nous prenons des valeurs décimales le résultat donne ce qui suit :

MSB 1111 1111 1111 1111 0000 0000 0000 0000 0000 0000 LSB

Si l'on prend maintenant le résultat net de l'acquisition I2C la partie rouge correspond à la température et donc à la partie préservée. Pour plus d'informations il faut se référer au chapitre de réalisation sur l'acquisition I2C.

 EIA-FR	Projet de semestre 6 Monitoring de ruches 2	Cahier des charges Version 3
---	--	---------------------------------

25 (0x0019) type: 0 (0x00) value:1e5e63a800001e4a63a000001e3e63a000001e3163a0

Quand tout se déroule parfaitement le masque donne ceci :

28 (0x001c) type: 0 (0x00) value:000063a80000000063a00000000063a00000000063a0

On voit déjà que le MSB se trouve à droite.

Maintenant regardons ce qui se passe si on exécute le même code plusieurs fois, recompilé ou non.

Résultat code compilé une première fois


1er Lancement

25 (0x0019) type: 0 (0x00) value:1e5e63a800001e4a63a000001e3e63a000001e3163a0 28 (0x001c) type: 0 (0x00) value:000063a80000000063a00000000063a00000000063a0	OK
25 (0x0019) type: 0 (0x00) value:1e3563a800001e3163a000001e3563a000001e3563a0 28 (0x001c) type: 0 (0x00) value:000063a80000000063a00000000063a00000000063a0	OK
Recompiler une première fois	
25 (0x0019) type: 0 (0x00) value:1de463d000001de463d000001de463d000001de463d0 28 (0x001c) type: 0 (0x00) value:010063d00000010063d00000010063d00000010063d0	KO
25 (0x0019) type: 0 (0x00) value:1e5263ac00001e4a63b000001e4663a800001e4263a8 28 (0x001c) type: 0 (0x00) value:000063ac0000000063b00000000063a80000000063a8	OK
25 (0x0019) type: 0 (0x00) value:1e5263ac00001e4a63b000001e4663a800001e4263a8 28 (0x001c) type: 0 (0x00) value:000063ac0000000063b00000000063a80000000063a8	OK
25 (0x0019) type: 0 (0x00) value:1c3770a00000170a7af4000011787fec00000dcb82bc 28 (0x001c) type: 0 (0x00) value:000070a0000001007af4000001007fec0000010082bc	OK

Tableau 20 Résultat masquage

On s'aperçoit que le mot n'est pas toujours correctement masqué. Un 1 vient se placer sur le 2^{ème} bit. On ne le voit pas ici mais on a constaté que cela semblait se produire lorsque les bits 5 à 8 correspondent en hexa à D ou en binaire

LSB 1011 ... MSB.

 EIA-FR	Projet de semestre 6 Monitoring de ruches 2	Cahier des charges Version 3
---	--	---------------------------------

Malgré cette probable observation on ne comprend toujours pas pourquoi le résultat varie au fil des exécutions. Après un laps de temps à tenter de comprendre, on a décidé de demander à l'un de nos experts son point de vue (*email). Il a suggéré d'utiliser des mots Hexadécimal comme masque, car plus pratique à manipuler. **Il est allé plus loin en fournissant le code complet d'extraction pour l'humidité et la température.**

2) Portion de code non exécutée lors de l'utilisation du capteur I2C

La machine d'état de l'application et la routine d'acquisition ont été écrites séparément et fonctionnaient parfaitement.

La machine d'état d'origine était réveillée toute les secondes. Elle devait, sur la base de compteurs, soit exécuter la collecte, soit envoyer une proposition.

Lorsque l'on a inséré le code d'acquisition du capteur I2C, l'une des 2 possibilités n'était jamais exécutée.

Si on reprend le code de la machine d'état, on remarque que la seconde possibilité est évaluée après l'évaluation et/ou l'exécution de la première.

Le problème était le suivant : Si on mettait la collecte (et donc la routine I2C) avant la proposition, cette dernière n'était plus exécutée. Idem dans le cas contraire. Il semblerait que le uP n'a pas eu le temps d'exécuter le second après l'avoir fait avec le premier.


C'est étonnant, car le temps d'acquisition d'une valeur pour ce capteur est de maximum 30.6 ms, ce qui est largement en dessous de la seconde.

Pour corriger ce problème on a modifié 2 paramètres

- Augmenter la période de l'horloge de la machine d'état. On est passé à 5 secondes (pour une démo).

Ceci n'est pas un problème, car le SSI désiré aura une fréquence de collecte basse. L'intervalle minimum définit étant de 1 min.

- Changer les évaluations sur les compteurs en utilisant le « plus grand ou égal » au lieu du simple « égal ». Ceci apporte l'assurance qu'un compteur ne débordera pas, même si la possibilité qui y est associée (collecte/proposition) n'a pas pu être exécutée.

 EIA-FR	Projet de semestre 6 Monitoring de ruches 2	Cahier des charges Version 3
---	--	---------------------------------

10.2 Perspectives

Dans le cadre du projet il y a des choses qui doivent être fait en premier lieu pour valider l'ensemble du concept et donc du système « beemon ».

Ce qui doit être fait en premier lieu :

1) Implémenter la communication SSI-SSC :

Durant ce semestre on a simulé le SSC avec BLEGUI. Il implémente les fonctionnalités principales d'un Scanner/Master et a donc été suffisant pour réaliser le code du SSI. Cependant certaines choses n'ont pas pu être simulées comme l'« indicate ». Il faut donc, pour valider, encore réaliser la partie Bluetooth du SSC.

2) Passer d'un serveur FTP à un serveur d'application avec base de données

On a introduit le concept d'interaction entre l'apiculteur et les SSI. Cela nécessite une connexion en direct et bidirectionnelle. Pour y arriver il faut implémenter un serveur d'application qui liera les 2 parties. Avec le FTP on a un moyen simple de stocker de l'information et qui est rapide à installer. Mais l'interaction change les besoins et nécessite obligatoirement un SA, on en profitera donc pour qu'il serve également d'intermédiaire entre le SSC et une BD

Remarque : La SIM900 ne donne aucune fonction prête à l'emploi pour communiquer avec une BD mais permet en fourni pour l'HTTP.

On peut donc imaginer ce qui suit :


SSC----http post----- SA [PHP ou Servlet/JSP] -----BD

Si les points 1) et 2) sont réalisés, on pourrait valider l'ensemble des fonctionnalités actuellement proposées. La seule chose que l'on peut actuellement et éventuellement concevoir sur le plan matériel est le SSI. Il y a encore des doutes à émettre avec le SSC et sa SIM 900.

Ce qui doit être fait par la suite mais qui n'est pas urgent :

- Conception et réalisation de la carte :

Il s'agit ici de l'objectif optionnel de ce semestre. On a pu poser définitivement ses interfaces et concevoir un début de design. On sait également qu'il faudra multiplexer l'unique entrée/sorties I2C du BLE112. Mais un travail supplémentaire devra être consacré pour améliorer son design matériel.

 EIA-FR	Projet de semestre 6 Monitoring de ruches 2	Cahier des charges Version 3
---	--	---------------------------------

Cependant rien n'empêche de commencer à réaliser son prototypage pour les éléments essentiels.

10.3 Conclusion

Ce projet m'intéresse toujours autant car il touche aux systèmes embarqués mais surtout parce que je le réalise dès le début. Cette forme de liberté et de créativité m'a motivée principalement durant tout le processus.

Pour ce qui est du déroulement du projet et du résultat produit je vous propose de lire les problèmes rencontrés.

Personnellement je ne suis pas satisfait du travail de réalisation car il est en dessous de mes espérances, même s'il répond partiellement aux objectifs exigés.

11 Référence

Tableaux

Tableau 1 Intervalles d'advertissig et fenêtres de scan possibles	14
Tableau 2 : Paramètres de maintien d'une connexion	15
Tableau 3 : Chauffage ruche	25
Tableau 4 : Humidificateur ruche	25
Tableau 5 : Dilution provisions	26
Tableau 6 : Ouverture et fermeture de la porte de la ruche	26
Tableau 7 : Abréviations des valeurs	27
Tableau 8 : Analyse des solutions possibles	29
Tableau 9 : Interfaces possibles	31
Tableau 10 : I2C vs SPI	32
Tableau 11 Formes de SSI vis-à-vis de ses dispositions	37
Tableau 12 Architectures de communications	39
Tableau Commande matériel	42
Tableau 14 Détail étiquettes de configurations	54
Tableau 15 Outils de productions.....	67
Tableau 16 Outils de vérification.....	67
Tableau 17 Légendes pour les test	81
Tableau 18 Test des fonctionnalités	81
Tableau 19 Validation des objectifs.....	82
Tableau 20 Résultat masquage	84

Tableaux

Figure 1: Contexte système	5
Figure 2:Architecture des systèmes matériels de supervision.....	7
Figure 3: Vue d'artiste de l'installation	8
Figure 4 Etude Bluetooth LE	10
Figure 5 Topologie BLE 1	11
Figure 6 Topologie BLE 2	11
Figure 7 Topologie BLE 3	11
Figure 8 Broadcaster	12
Figure 9 et Observer	12
Figure 10 Machine d'état du module BLE.....	12
Figure 11 :Etats possibles du BLE	13
Figure 12 Scanning et advertissig	14
Figure 13 Maintien de la connexion	15
Figure 14 Fonctionnement du transfert de données dans BLE.....	16
Figure 15 Pile de protocole BLE	17



	Projet de semestre 6 Monitoring de ruches 2	Cahier des charges Version 3
---	--	---------------------------------

Figure 16 Passive scanning	18
Figure 17 Active scanning	18
Figure 18: Ce que contient un BLE 112	19
Figure 19: Utilisations du système BEEMON	20
Figure 20 : Mise en service	21
Figure 21: Interfaces SSI	22
Figure 22: Plan analyse interfaces	22
Figure 23: Spectre total des valeurs à surveiller et des dispositifs d'interactions	27
Figure 24 : Portes de détection	29
Figure 25 Découpage ruche Dadant	34
Figure 26 Localisation des mesures pour l'apiculteur	35
Figure 27 Localisation mesures simplifié	35
Figure 28 SSI forme 1	36
Figure 29 SSI forme 2 Figure 30 Extension pour la porte	36
Figure 31 SSI forme 3	36
Figure 32 BD mise en service	43
Figure 33 Architecture logicielle Global	45
Figure 34 Scopes des systèmes beemon	45
Figure 35 Comportement temporel du SSI	46
Figure 36 Connexion directe 1	47
Figure 37 Connexion directe 2	47
Figure 38 Connexion directe 3	48
Figure 39 Connexion directe 4	48
Figure 40 Connexion direct 5 variante	48
Figure 41 Architecture matérielle du SSI	49
Figure 42 Résumé des chemins empruntés	51
Figure 43 Squelette de l'application	52
Figure 44 Etiquettes de configurations	53
Figure 45 Persistance de la configuration	55
Figure 46 Message d'advertising	56
Figure 47 Mise en service et configuration	57
Figure 48 Mise en service et configuration vue complète 1	59
Figure 49 Mise en service et configuration vue complète 2	60
Figure 50 Processus de collecte	61
Figure 51 Comportement temporel du dépôt	62
Figure 52 Dépôt variante advertising	63
Figure 53 Dépôt variante indicate	64
Figure 54 Alerte problème variante Advertising	65
Figure 55 Alerte variante indicate	65
Figure 56 Matériel de base du prototype fonctionnel	67
Figure 57 : Bytes transmis par H6121	79

 EIA-FR	Projet de semestre 6 Monitoring de ruches 2	Cahier des charges Version 3
---	--	---------------------------------

12 Glossaire

CBI : Capteur Bluetooth individuel. L'idée est d'avoir un ou 2 capteurs pour un module BLE afin d'avoir un système très petit capable de s'insérer directement dans la ruche

SSI : Système de supervision intermédiaire

SSC : Système de supervision central

Infrastructure BEEMON : Ensemble des systèmes utilisés pour proposer des services Beemon

Bluetooth Low Energy : Technologie de communication basée sur le Bluetooth Classic qui est moins gourmand en consommation et qui apporte de nouvelles fonctionnalités

Gatt : Base de données utilisée dans la communication Bluetooth LE


PS : Mémoire non volatile, ici de type Flash

Etiquettes de configuration : Format d'information permettant au SSI d'interpréter des valeurs de configuration et pour le SSC de connaître la configuration du SSI

GSM : Technologie de communication utilisée dans les réseaux mobiles et qui donne accès à internet avec des protocoles supplémentaires utilisant ce dernier comme le GPRS.

Spectre minimal : Ensemble restreint de valeurs utiles à l'apiculteur

Spectre élargi : Ensemble s'ajoutant au premier et qui pourrait être utile à la recherche scientifique

 EIA-FR	Projet de semestre 6 Monitoring de ruches 2	Cahier des charges Version 3
---	--	---------------------------------

13 Bibliographie

ActuAPI. *Isoler, attention à l'humidité*. Février 2009.

<http://www.cari.be/medias/actuapi/actuapi46.pdf>.

blackberry.com. *Bluetooth LE primer for developers*. 4 Mai 2013.

<https://supportforums.blackberry.com/t5/Native-Development-Knowledge/BlackBerry-10-Bluetooth-LE-primer-for-developers/ta-p/2287377>.

blog.bluetooth-smart. *Programming the BLE112 with C code using IAR*. Septembre 2012.

<http://blog.bluetooth-smart.com/2012/09/11/programming-the-ble112-with-c-code-using-iar/>.

Bluegiga. *BGSCRIPT SCRIPTING LANGUAGE DEVELOPER GUIDE*. Mars 2014. CD : ..\Documentation technique\BLE\BLE112_BGScript_developer_guide_v3.4.pdf P.25.

—. *BLE112 Bluetooth Smart Module*. s.d. <https://www.bluegiga.com/en-US/products/bluetooth-4.0-modules/ble112-bluetooth-smart-module/> (accès le Mai 5, 2014).

—. *BLE112 Datasheet*. Mars 2014. CD : ..\Documentation technique\BLE\BLE112_Datasheet.pdf P.29.

—. *BLUETOOTH SMART MODULE CONFIGURATION GUIDE*. Février 2014. CD : ..\Documentation technique\BLE\BLE112_Bluetooth_Smart_Configuration_Guide_v33.pdf P.20.

csr. *Bluetooth 4.0: Low Energy*. s.d. <http://chapters.comsoc.org/vancouver/BTLER3.pdf> P.18 (accès le Avril 2014).

Galeev, Mikhail. *Bluetooth 4.0: An introduction to Bluetooth Low Energy—Part II*. 28 Juillet 2011.

http://www.eetimes.com/document.asp?doc_id=1278966.

guidgenerator. *Online GUID Generator*. s.d. <http://guidgenerator.com/> (accès le Avril 15, 2014).


Honeywell. *Honeywell HumidCon Digital Temperature Humidity Sensors*. Juillet 2012. CD :

..\Documentation technique\capteur\H6121_Datasheets.pdf P.8.

—. *I2C Communication with the Honeywell HumidCon*. juin 2012. CD : ..\Documentation technique\capteur\H6121_CommunicationI2C P.2.

<http://support.connectblue.com/>. *Bluetooth Low Energy Serial Port Adapter - Getting Started*. 15 Novembre 2013.

<http://support.connectblue.com/display/PRODBTSPA/Bluetooth+Low+Energy+Serial+Port+Adapter+-+Getting+Started>.

 EIA-FR	Projet de semestre 6 Monitoring de ruches 2	Cahier des charges Version 3
---	--	---------------------------------

K.Gautham, G.Raghav, V.Krishnamurthy, N.R.Raajan. *Personnel Security System using Bluetooth*. Mai 2013. CD : ..\Documentation technique\BLE\CD : ..\Documentation technique\BLE\BLE112_Bluetooth_Smart_Configuration_Guide_v33.pdf P.2.

M., Cornaz G. Elgueloui. *Projet de semestre 5 Bluetooth low energy*. 26 Janvier 2014. CD : ..\Documentation technique\CornazElgueloui\rapport_final.pdf P.2.

Moreaux, André. *Les chiffres et les Abeilles !* 09 août 27.
<http://home.nordnet.fr/~amoreaux/abeille/chfabeil.htm>.

rapidtables.com. *Hex to ASCII converter*. s.d. <http://www.rapidtables.com/convert/number/hex-to-ascii.htm> (accès le Avril 15, 2014).

Riondet, Jean. *La ruche au fil des saisons*. s.d. <http://www.abeillesentinelles.net/abeille/la-ruche-abeille-au-fil-des-saisons.html> (accès le mars 15, 2014).

14 Contenu du CD-ROM

Documents ruche	Contient une étude sur la déperdition de la température dans une ruche
Documents technique	Contient les datasheets pour le capteurs , le module et le multiplexeur. Pour plus d'informations voir le read me
Documents BEEMON	-Toutes les versions de cahiers des charges Toutes les versions du rapport -Images rapports
Environnement de travail	-Contient les outils de développement de bluegiga
Réalisation	-Contient les différentes versions du code de l'application
Readme.txt	-Contient une description plus détaillée des fichiers