PROJET RUCHER

Par

Christopher Bergont

Léa Cruiziat

Alexandre La Torre

Benoit Moderiano

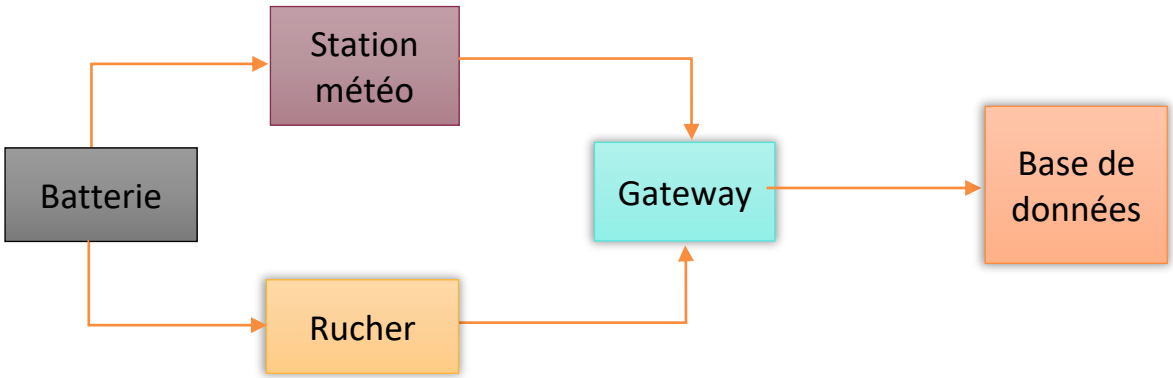
Christian Peter

TABLE DES MATIERES

I. PRESENTATION GENERALE DU PROJET

1. Généralités

Le projet Ruche est un projet qui consiste à monitorer une colonie d'abeilles. En effet, la vie d'une colonie d'abeilles est composée d'étapes. La surveillance de ces étapes par l'apiculteur permet une meilleure appréhension d'éventuels problèmes ou de gestion de sa colonie. Ce projet regroupe différents sous-groupes : Gateway, station météo, rucher, batterie et base de données (voir figure 1).



2. Module ruche interne

Le module Rucher est le module qui va permettre de monitorer la ruche. L'apiculteur peut se baser sur certaines grandeurs physiques pour appréhender la vie de la colonie.

Si la masse de la ruche varie signifie qu'il y a eu des œufs ou une production de miel. S'il y a trop de naissances, une colonie peut se diviser en deux ce qui peut amener à une mort de la colonie. La connaissance des variations de températures peut permettre d'éviter de perdre une colonie suite à de température inadaptées.

Le module Rucher est divisé en deux cartes. Une carte fille qui sera placé dans la ruche et une carte mère qui sera placé à l'extérieur de la ruche. Les deux cartes communiquent entre elles par Bluetooth.

a. Carte fille

La carte fille présente dans la ruche est équipée des capteurs internes de la ruche : Capteur de température, capteur d'hygrométrie, capteur de vibration, capteur de bruit interne et un capteur de CO2. Au monitoring de la ruche s'ajoute la communication Bluetooth avec les informations envoyées à la carte mère.

b. Carte mère

La carte mère se charge de la gestion la balance ainsi que de la communication à la GateWay. La technologie utilisée pour la communication est le LoRa, détaillée dans le IV.

3. Contraintes de l'électronique dans une ruche

La carte fille est placée dans un environnement hostile pour une carte électronique. En effet, les abeilles bouchent tous les trous existants dans leur ruche. De plus, nous ne pouvons pas imaginer une carte une grande taille de carte. Cette carte est un élément non habituel pour les abeilles et il s'agit de réduire au maximum l'impact de cet ajout.

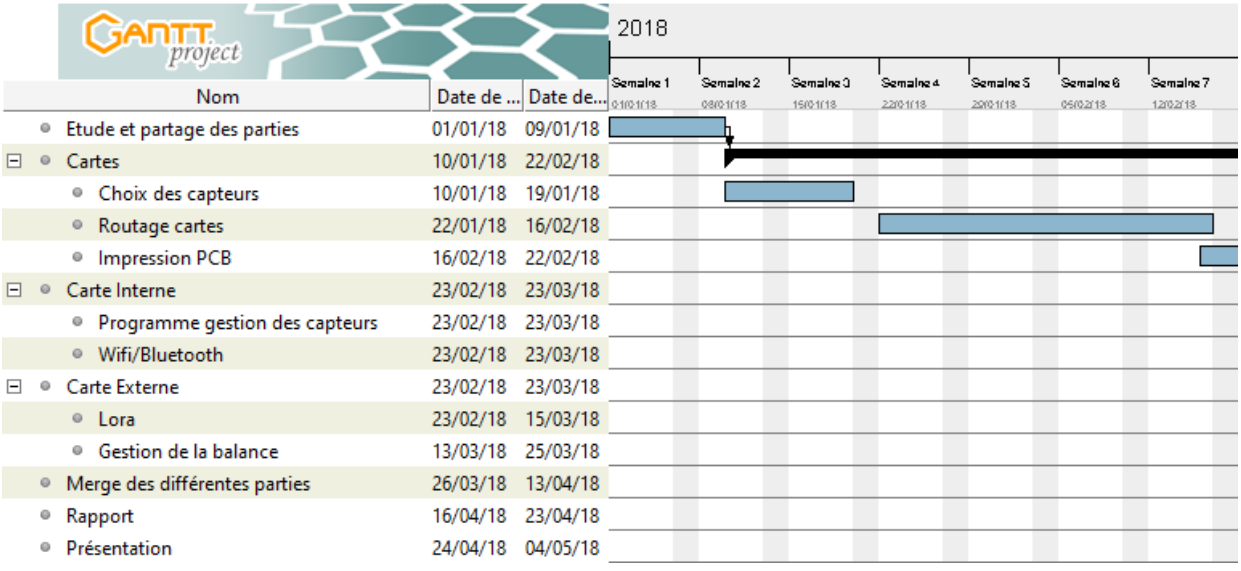
4. Contraintes de l'électronique dans une ruche

La carte fille est placée dans un environnement hostile pour une carte électronique. En effet, les abeilles bouchent tous les trous existants dans leur ruche. De plus, nous ne pouvons pas imaginer une carte une grande taille de carte. Cette carte est un élément non habituel pour les abeilles et il s'agit de réduire au maximum l'impact de cet ajout.

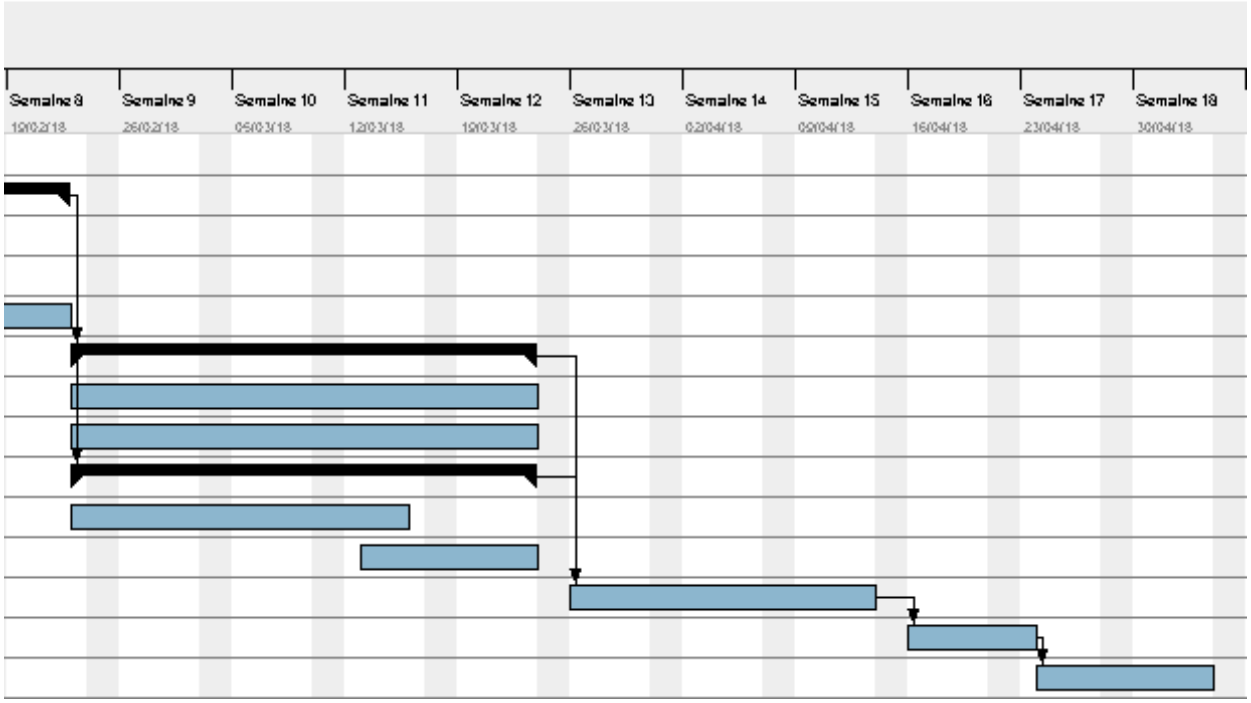
II. PLANIFICATIONS ET OUTILS DE L'EQUIPE

1. Gantt

Le diagramme de Gantt, couramment utilisé en gestion de projet, est un outil qui permet de représenter visuellement l'état d'avancement des différentes activités qui constituent un projet.

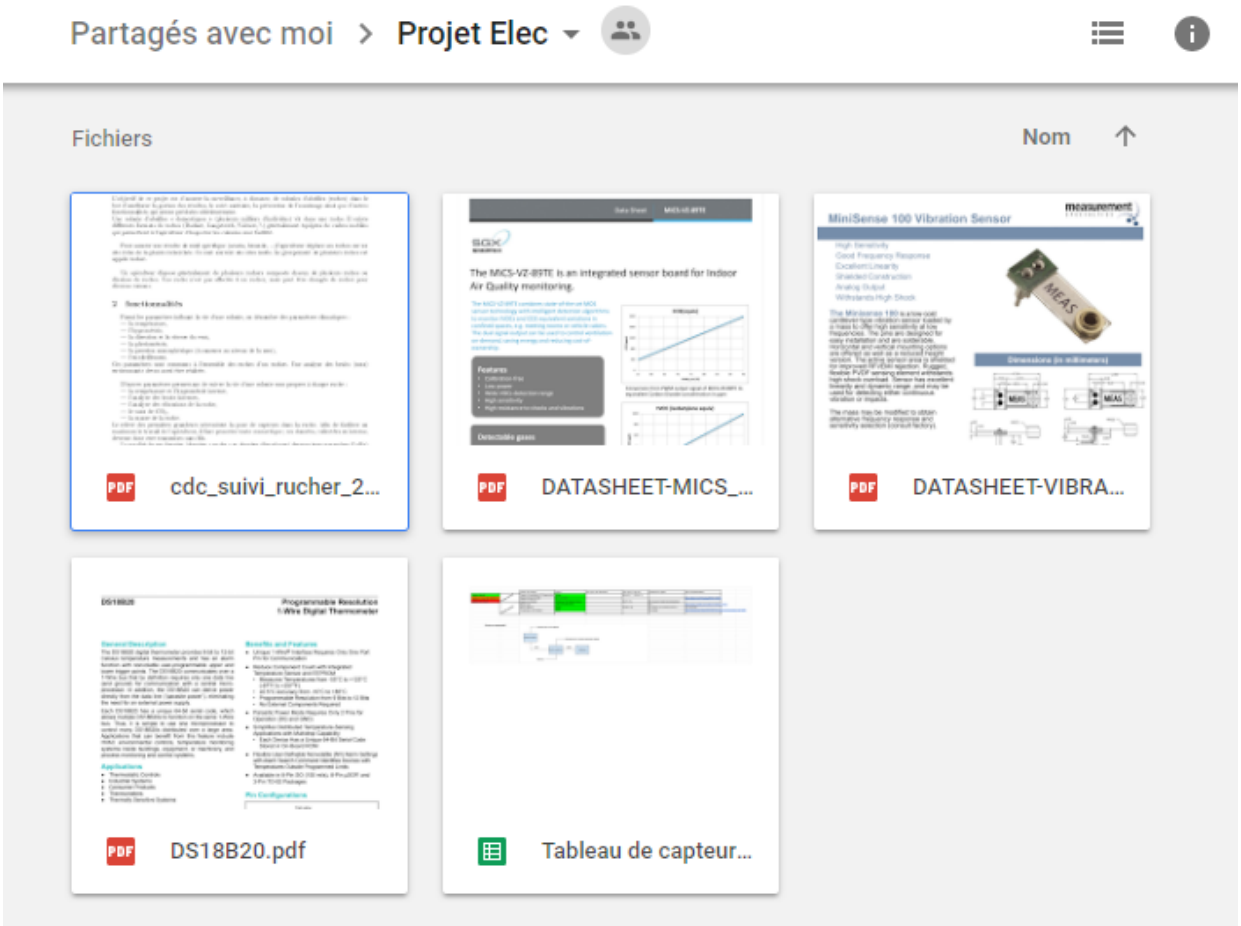


Nom	Date de ...	Date de...
Etude et partage des parties	01/01/18	09/01/18
Cartes	10/01/18	22/02/18
Choix des capteurs	10/01/18	19/01/18
Routage cartes	22/01/18	16/02/18
Impression PCB	16/02/18	22/02/18
Carte Interne	23/02/18	23/03/18
Programme gestion des capteurs	23/02/18	23/03/18
Wifi/Bluetooth	23/02/18	23/03/18
Carte Externe	23/02/18	23/03/18
Lora	23/02/18	15/03/18
Gestion de la balance	13/03/18	25/03/18
Merge des différentes parties	26/03/18	13/04/18
Rapport	16/04/18	23/04/18
Présentation	24/04/18	04/05/18



2. Google Drive

Afin de centraliser les différents documents que ce soit texte, tableau ou cahier des charges nous avons décidé de crée un Google Drive. Le principal avantage est de pouvoir éditer des documents à plusieurs et en même temps tout en ayant une sauvegarde automatique.



3. Bitbucket et GitHub

Google Drive est très utile mais sa limitation dans notre projet résidait autour du code.

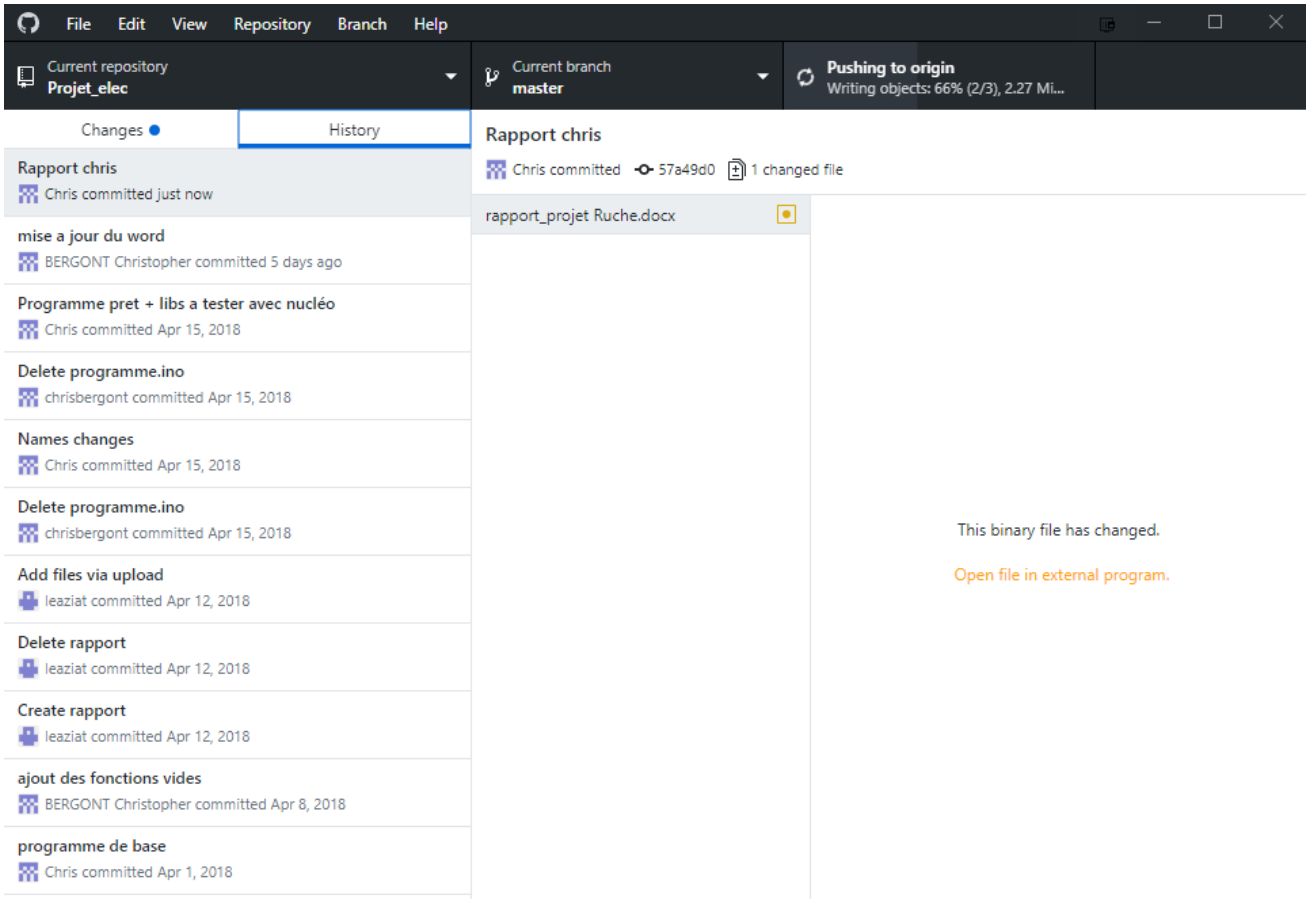
En effet pour se partager des ressources de programmation il y a des outils qui permettent de merger facilement 2 codes ou d'avoir un historique de modifications.

Dans cet esprit nous avons décidé de créer dans un premier temps un Bitbucket mais après quelques problèmes rencontres nous avons sommes passés sur GitHub.

Pour réaliser des opérations sur celui-ci il convient de s'inscrire sur le site et de créer un repository.

Ensuite on peut ajouter tous les membres de l'équipe pour qu'ils récupèrent ou poussent du code.

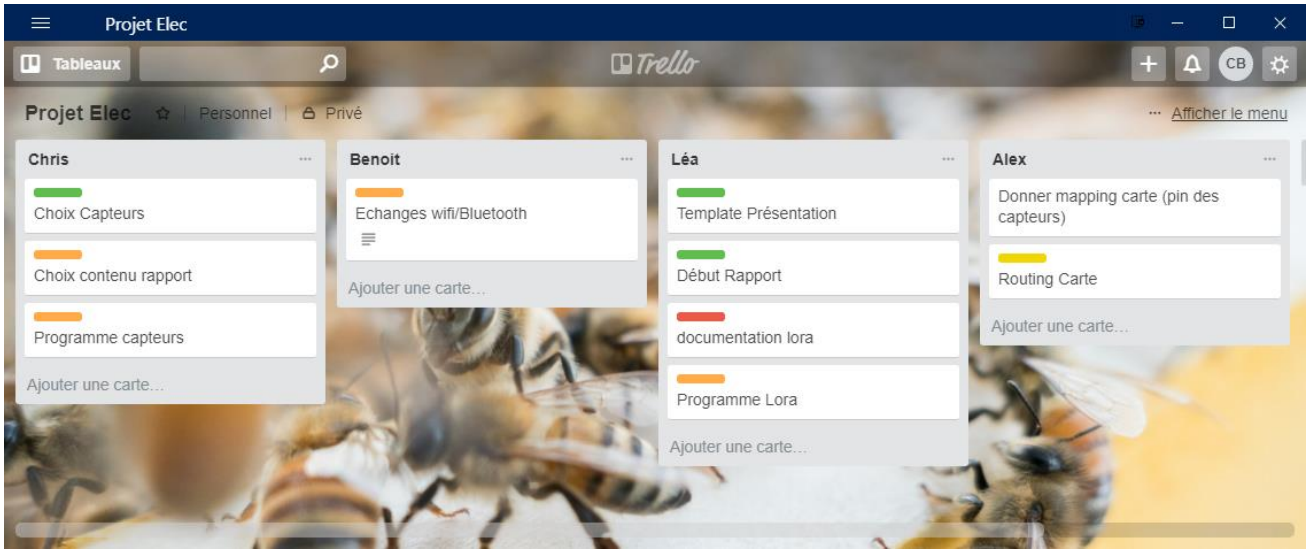
Précisons que les actions sur ce repository pourront se faire par lignes de commande ou grâce à une interface graphique nommée GitHub Desktop, qui nous permettra de ne pas retenir toutes les commandes !



4. Trello

Trello est un outil de gestion de projet en ligne et inspiré par la méthode Kanban de Toyota. Il est basé sur une organisation des projets en planches listant des cartes, chacune représentant des tâches. Les cartes sont assignables à des utilisateurs et sont mobiles d'une planche à l'autre, traduisant leur avancement.

Nous avons décidé de l'utiliser pour nous séparer les taches et connait l'avancement de chacun.



5. Slack

Slack est une plate-forme de communication collaborative propriétaire.

Elle permet une synchronisation avec le Trello et elle nous a permis de communiquer durant tout le projet.

III. LES CAPTEURS DE LA CARTE FILLE

1. Cahier des charges

Le cahier des charges nous a imposé un nombre de capteurs

-CO2

-Vibration

-Bruit interne

-Température

-Hygrométrie

Comme expliqué précédemment, nous avons des contraintes environnementales liées au rucher. Nous n'utiliserons pas de capteur de bruit interne car le microphone est inadapté dans une ruche.

Température	De -40° à 80° %	0,1° de précision
CO2	Rien d'imposé car grandeur nouvelle	
Vibration	De 50 Hz à 1Khz.	1Hz de précision
Hygrométrie	De 0 à 100 %	1 % de précision

2. Capteurs choisis

Le choix des capteurs a été fait en coordination entre les différentes parties du projet et le cahier des charges, il était important de pouvoir adapter les capteurs sur la carte en fonctions des différentes contraintes (tension, résistance interne, capacité interne, etc …).

Un autre point important à prendre en compte était la valeur retournée par le capteur et le moyen de communication, mais nous reviendrons sur ce point plus tard.

Comme décrit quand la partie précédente nous avions un cahier des charges à respecter et je vais vous préciser ici la démarche qui nous a permis de choisir les capteurs.

Le but était pour nous dans un premier temps d'avoir des capteurs avec des sorties linéaires et qui nous permettent donc de traiter leur valeur de sortie facilement avec notre programme. Vous comprendrez qu'un capteur avec un courbe de sortie pour laquelle nous ne pouvons définir mathématiquement la courbe de sortie est plus difficile à traiter.

A) CAPTEUR DE TEMPERATURE ET HYGROMETRIE

Pour cette partie nous avons choisi le Si7006-A20.

Le capteur d'humidité et de température Si7006 I2C est un circuit intégré CMOS intégrant des capteurs sensibles à l'humidité et à la température et un capteur analogique-numérique.

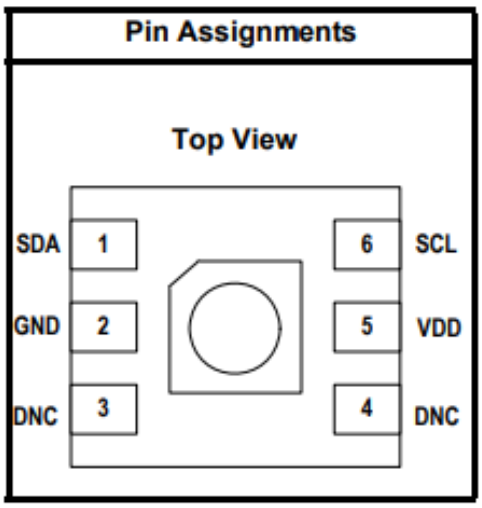
Les sondes d'humidité et de température sont étalonnées en usine et l'étalonnage est stockée dans la mémoire non volatile de la puce. Ceci permet de s'assurer que les capteurs sont entièrement interchangeables et ne nécessitent aucun recalibrage ou changement de logiciel.

Le Si7006 est disponible en boîtier DFN 3x3 mm.

Ordering Information:
See page 28.

Voici ses caractéristiques :

- Capteur d'humidité (RH ± 5 % (max), 0-90% HR)
- Capteur de température (±1 °C (max.), -10 à 85 °C)
- 0 à 100 % : plage de fonctionnement RH
- 40 à +125 °C : plage de fonctionnement
- Tension de fonctionnement (1,9 à 3,6 V)
- Faible consommation d'énergie
    - 150 µA courant actif
    - 60 nA courant de veille
- Calibré d'usine
- Interface I2C
- 3x3 mm DFN Package DFN
- Excellente stabilité à long terme

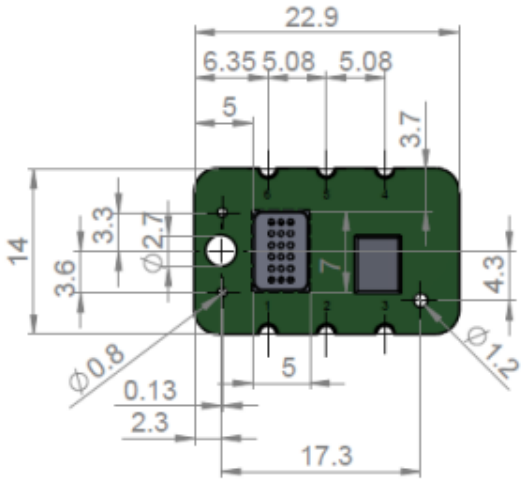


B) CAPTEUR DE TAUX DE CO2

Pour cette partie nous avons choisi le MiCS-VZ-89TE.

Voici ses caractéristiques :

- Sans calibrage
- Faible puissance
- Large plage de détection des VOCs
- Sensibilité élevée
- Haute résistance aux chocs et aux vibrations
- Tension d'alimentation (3,3V DC régulée +/- 5%)
- Puissance de fonctionnement (125 mW)
- Température de fonctionnement (0°C à 50°C)
- Humidité en fonctionnement (0% HR à 95% HR (sans condensation))
- Température de veille (-40°C à 80°C)
- Humidité de veille (0%HR à 95%HR (sans condensation))

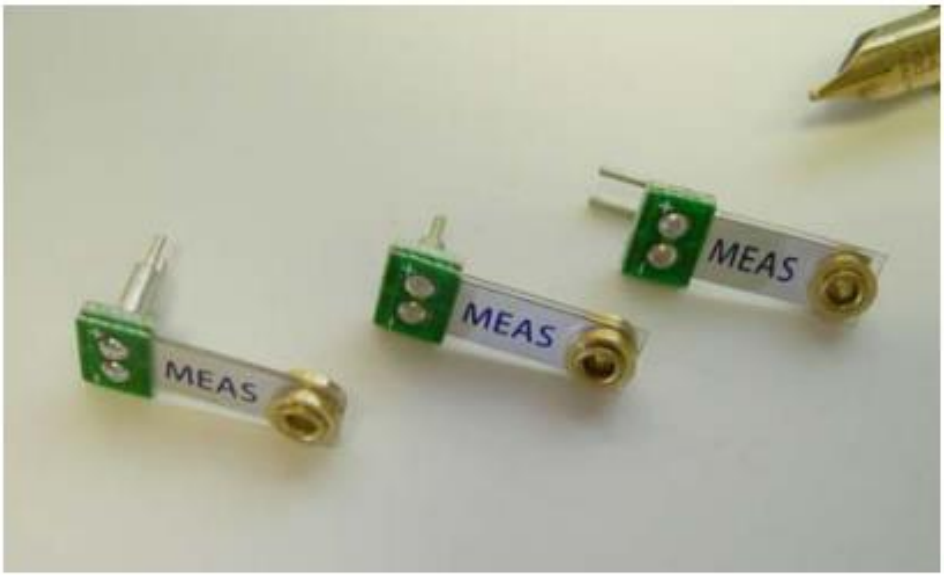


Pin Connection VZ-89TE

6:+ 3.3V	5: NC	4: SDA
1: PWM OUT	2: SCL	3: GND

C) CAPTEUR DE VIBRATION

Pour cette partie nous avons choisi le MiniSense 100 Vibration Sensor.



Le Minisense 100 est de vibrations chargé par une masse permettant d'offrir une sensibilité élevée même avec un faible niveau de fréquences. Les épingles sont conçues pour être faciles à installer et sont soudables.

Des possibilités de montage horizontal et vertical sont offerts ainsi qu'une hauteur réduite.

La zone active du capteur est blindée pour un meilleur rejet des perturbations.

La masse peut être modifiée pour obtenir une réponse en fréquence et une sensibilité différente.

Voici ses caractéristiques :

- Haute sensibilité de tension (1 V/g)
- Montage horizontal ou vertical
- Broches soudables, montage sur circuit imprimé
- Faible coût
- < 1% Linéarité
- Fonctionnement jusqu'à 40 Hz (2 400 tr/min)
- Sensibilité à la tension (1,1 V/g)
- Sensibilité à la charge (260 pC/g)
- Fréquence de résonnance (75 Hz)
- Sensibilité à la tension (6 V/g)
- Fréquence limite supérieure (42 Hz)
- Capacité (244 pF)
- Température de veille (-40 à +80 °C)
- Température de fonctionnement (-20 à +60 °C)

D) UN IMPREVU …

Une des données que j'ai citées précédemment n'avait pas été prise en compte lors de notre choix des capteurs, nous avions pour objectif de choisir un capteur pour lequel on récupérait tout simplement une valeur de tension en fonction de sa résistance de sortie.

Car par manque d'expérience nous pensions que la grande partie des capteurs fonctionnaient comme ceci.

En réalité excepté le capteur de vibration, les composants choisit fonctionnent en I2C. Ce qui nous a demandé une étude sur le fonctionnement et la mise en place d'une ligne I2C.

Nous allons donc détailler maintenant le fonctionnement de cette technologie.

3. Protocole des capteurs : l'I2C (Inter-integrated circuit)

Tout d'abord qu'est-ce que l'I2C, c'est un bus de données qui fait partie des standards les plus connus dans la microélectronique il permet généralement de connecter et de faire communiquer un microprocesseur et de nombreux composants.

Elle est créée en 1992 par Philips et est aujourd'hui maintenu par NXP.

C'est un bus série synchrone bidirectionnel half-duplex, où plusieurs équipements, maitres ou esclaves, peuvent être connectés au bus.

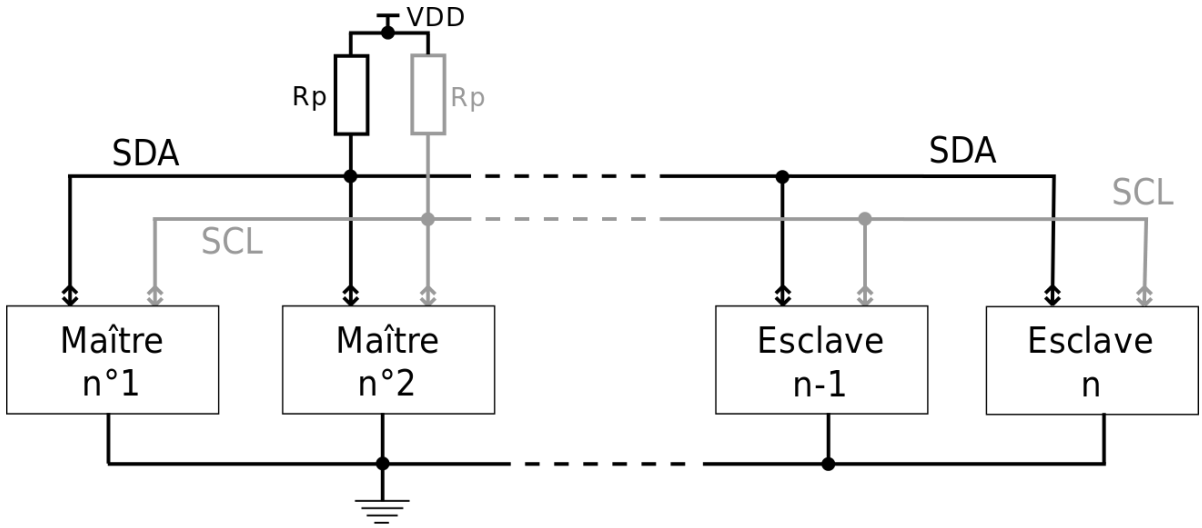
Il y a toujours un seul maitre qui parle à la fois (à un ou plusieurs esclaves).

La connexion est réalisée par l'intermédiaire de 3 fils :

- SDA (Serial Data Line) : ligne de données bidirectionnelle,
- SCL (Serial Clock Line) : ligne d'horloge de synchronisation bidirectionnelle.
- Masse

Les 2 premières lignes sont tirées au niveau de tension V_{DD} à travers des résistances de pull-up (R_P).

Le nombre maximal d'équipements est limité par le nombre d'adresses disponibles, 7 bits d'adressage et un bit R/W (lecture ou écriture), soit 128 périphériques. Il y a une adresse par périphérique.



Pour qu'un maitre prenne le contrôle du Bus il faut que SDA et SCL soit au repos (donc à '1').

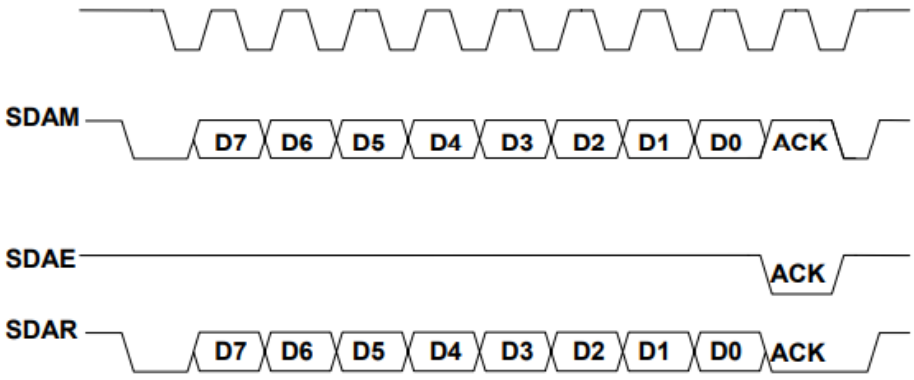
Pour transmettre des données, il faut surveiller :

- La condition de départ : SDA passe à 0, SCL reste à 1
- La condition d'arrêt : SDA passe à 1, SCL reste à 1

C'est alors le nouveau maitre qui génère son signal d'horloge.

Pour la transmission d'un octet le maitre envoi le bit de poids fort sur SDA et valide l'envoi en passant à '1' le SCL. Puis il repasse SCL à '0'. Il fait ainsi pour tous les bits jusqu'à la fin de l'octet. Puis envoi un bit d'acknowledgement.

L'esclave met ensuite le SDA à '0' pour confirmer que la transmission s'est bien passé.



SCL : Horloge imposée par le maître
SDAM : Niveaux de SDA imposés par le maître
SDAE : Niveaux de SDA imposés par l'esclave
SDAR : Niveaux de SDA réels résultants

Après avoir étudié toute la technologie I2C nous avons découvert que des librairies existaient pour une grande partie des capteurs I2C que l'on peut trouver sur le marché.

4. Le programme

Au départ nous comptions programmer directement le microcontrôleur sur le PCB final. Mais comme nous avons pris du retard nous avons préféré commencer par programmer sur des cartes de test.

A. ARDUINO UNO

Etant donné que nous n'avions que 2 ESP32 que Léa et Benoit ont dû se partager pour travailler en parallèle sur le Wifi/Bluetooth et sur le Lora, Christopher a décidé de coder sur une carte Arduino Uno personnelle afin de n'avoir qu'à adapter le programme en fonction du mapping final du PCB et d'éventuelles caractéristiques différentes de l'ESP32 par rapport à cette carte.

Après la création d'un programme « prototype », il nous fallait commander les capteurs choisis mais il était déjà trop tard. Travaillant dans le domaine de l'électronique nous avons décidé de rechercher dans nos entreprises si certains d'entre nous avaient des capteur I2C pour tester l'implémentation de ce protocole.

Christopher à trouver un « shield » développé par STMicroelectronics. Le X-NUCLEO-IKS01A1.

Il a adapté le programme en fonction du prototype qu'il avait fait pour les autres capteurs et des nouvelles librairies. Malheureusement à la compilation nous avions un problème concernant la compatibilité de la librairie qui nous demandait un ST32 pour fonctionner idéalement. Mais c'était uniquement des warnings.

Après nos tests, nous en avons conclus que le problème était la carte. Nous sommes donc passés sur une nucléo de chez STMicroelectronics.

B. MBED ET NUCLEO

Après le passage sur Nucléo nous avions toujours des problèmes sur le téléversement du programme, Léa ayant l'habitude de travailler sur Nucléo à proposer de passer sur l'interface Mbed.

Sur cette dernière nous y avons trouver une librairie pour le « shield » entier et non capteurs par capteurs.

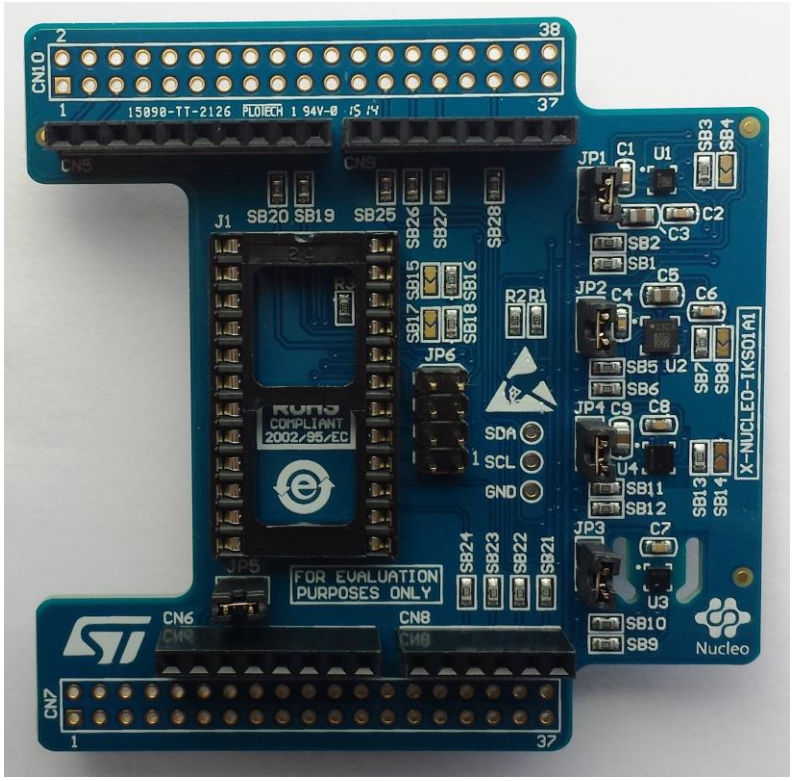
Après utilisation et adaptation du programme pour celle-ci nous avons réussi à le faire fonctionner.

C. PRENTATION DU SHIELD ET DE LA LIBRAIRIE

Le X-NUCLEO-IKS01A1 est un système de cartes d'évaluation de capteurs environnementaux.

Il est compatible avec la disposition des connecteurs Arduino UNO R3 et est conçu autour de l'accéléromètre 3 axes LSM6DS0 de STMicroelectronics + gyroscope 3 axes, du magnétomètre 3 axes LIS3MDL, du capteur d'humidité et de température HTS221 et du capteur de pression LPS25HB*.

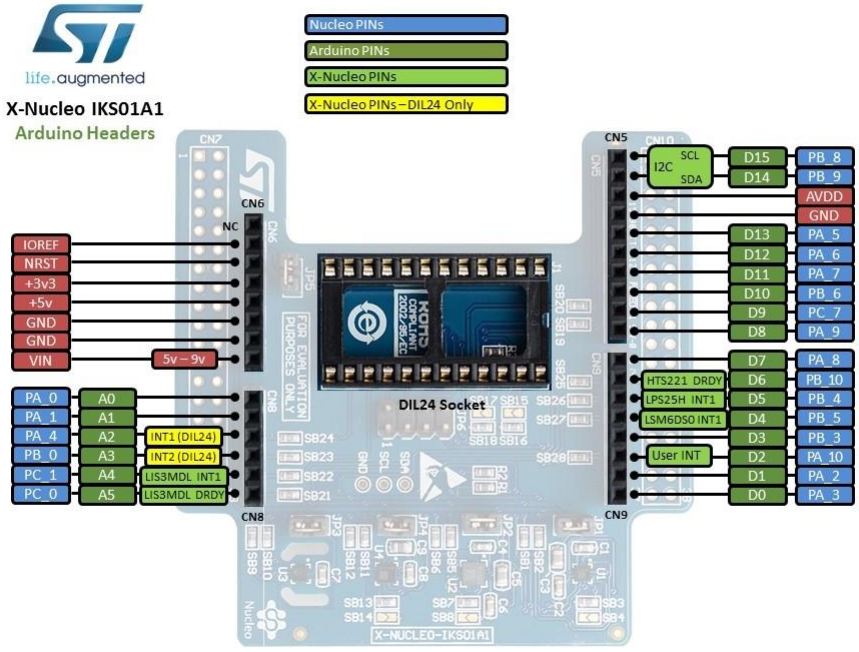
Le X-NUCLEO-IKS01A1 s'interface avec le microcontrôleur STM32 via la broche I²C, et il est possible de changer le port I²C par défaut.



CARACTERISTIQUES :

- LSM6DS0 : accéléromètre 3D (±2/±4/±8 g) + gyroscope 3D (±245/±500/±2000 dps)
- LIS3MDL : magnétomètre 3D (±4/ ±8/ ±12/ ±12/ 16 gauss)
- LPS25HB* : capteur de pression MEMS, 260-1260 hPa : baromètre de sortie numérique absolu.
- HTS221 : humidité relative et température numérique capacitive numérique
- Prise DIL 24 broches disponible pour d'autres adaptateurs MEMS et autres capteurs
- Librairie complète et gratuite de firmware de développement et exemple pour tous les capteurs compatibles avec le firmware STM32Cube.
- Compatible avec les cartes STM32 Nucleo.
- Equipé d'un connecteur Arduino UNO R3.

- Compatible RoHS

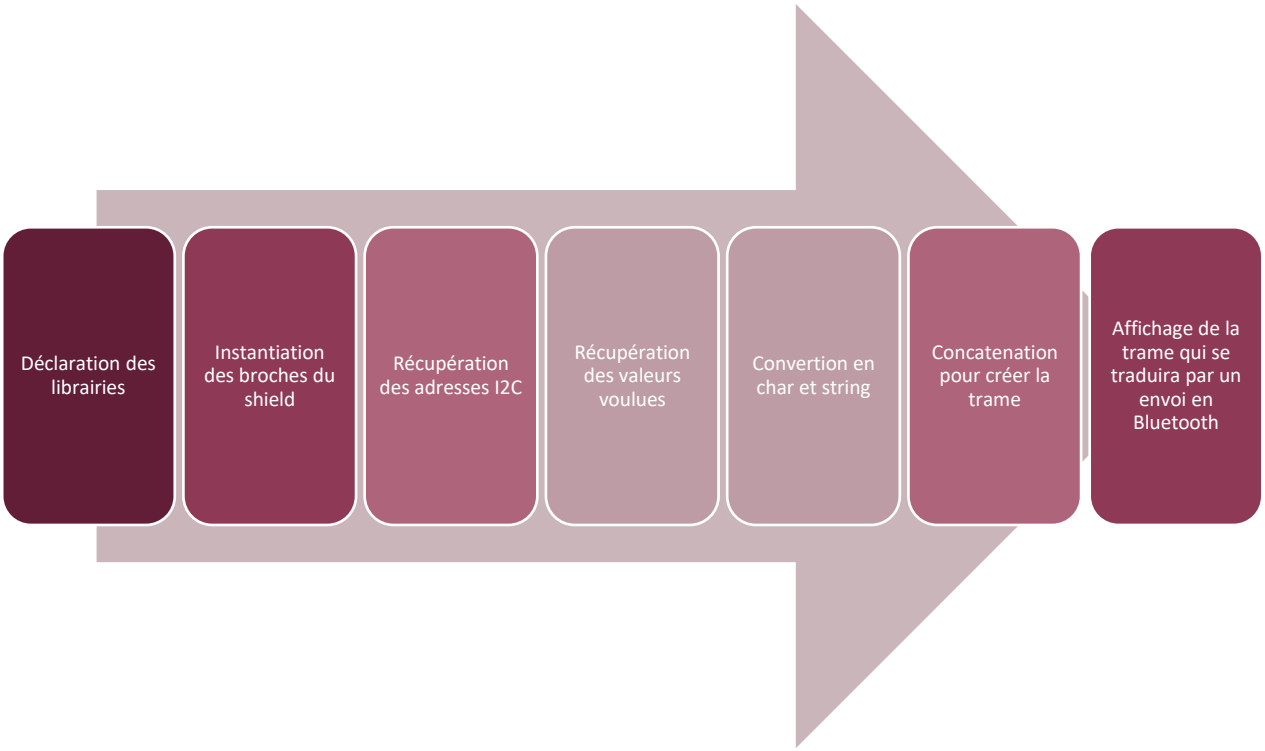


Les fonctions de la librairie que nous utilisons dans notre programme sont :

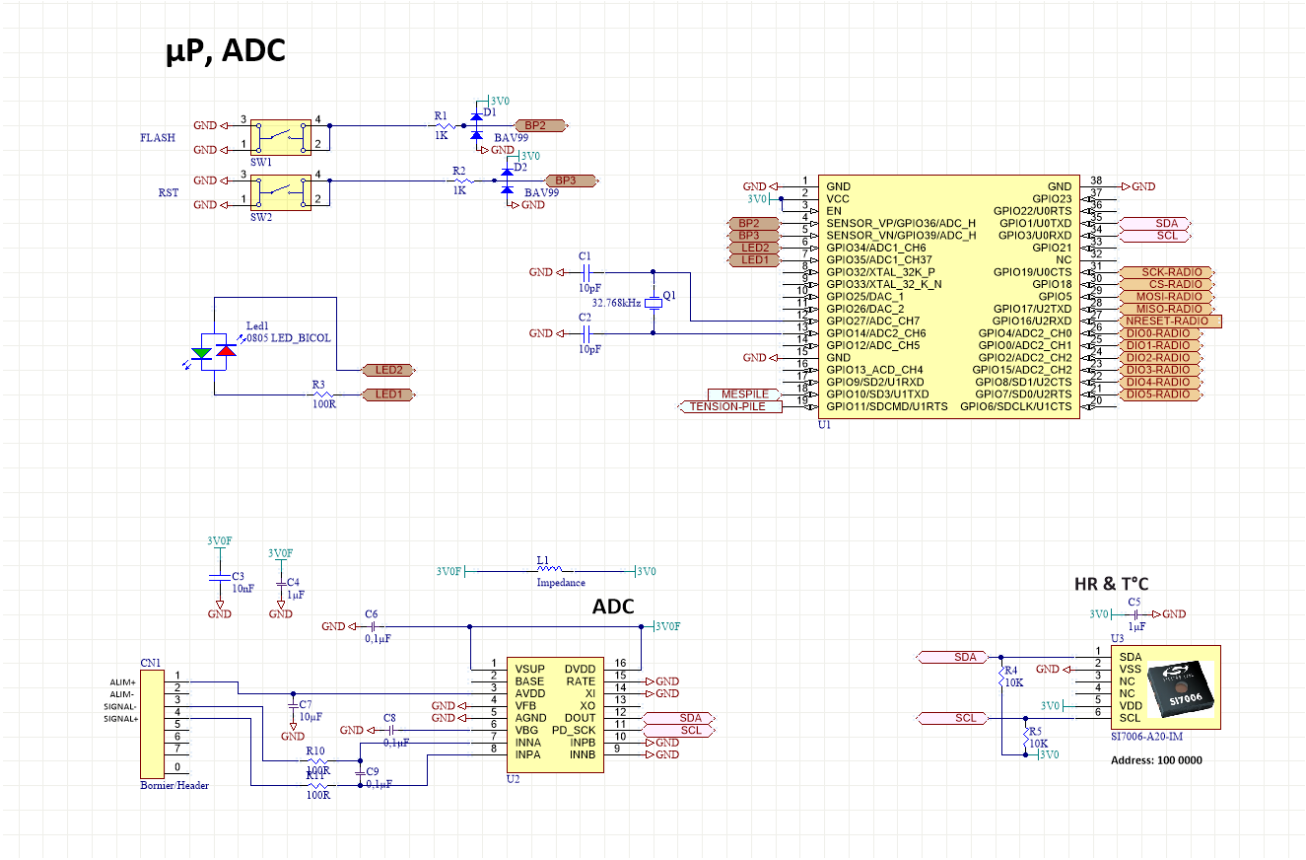
- *Vitual int read_id (unint8_t *ht_id)* : permet de récupérer l'adresse I2C des capteurs.
- *Virtual int get_temperature (float *pfData)* : permet de récupérer la temperature en °C
- *Virtual int get_humidity (float *pfData)* : permet de récupérer le taux d'humidité
- *Virtual int get_fahreinheit (float *pfData)* : permet de récupérer la temperature en °F
- *Virtual int get_pressure (float *pfData)* : permet de récupérer la pression

D. PROGRAMME FINAL

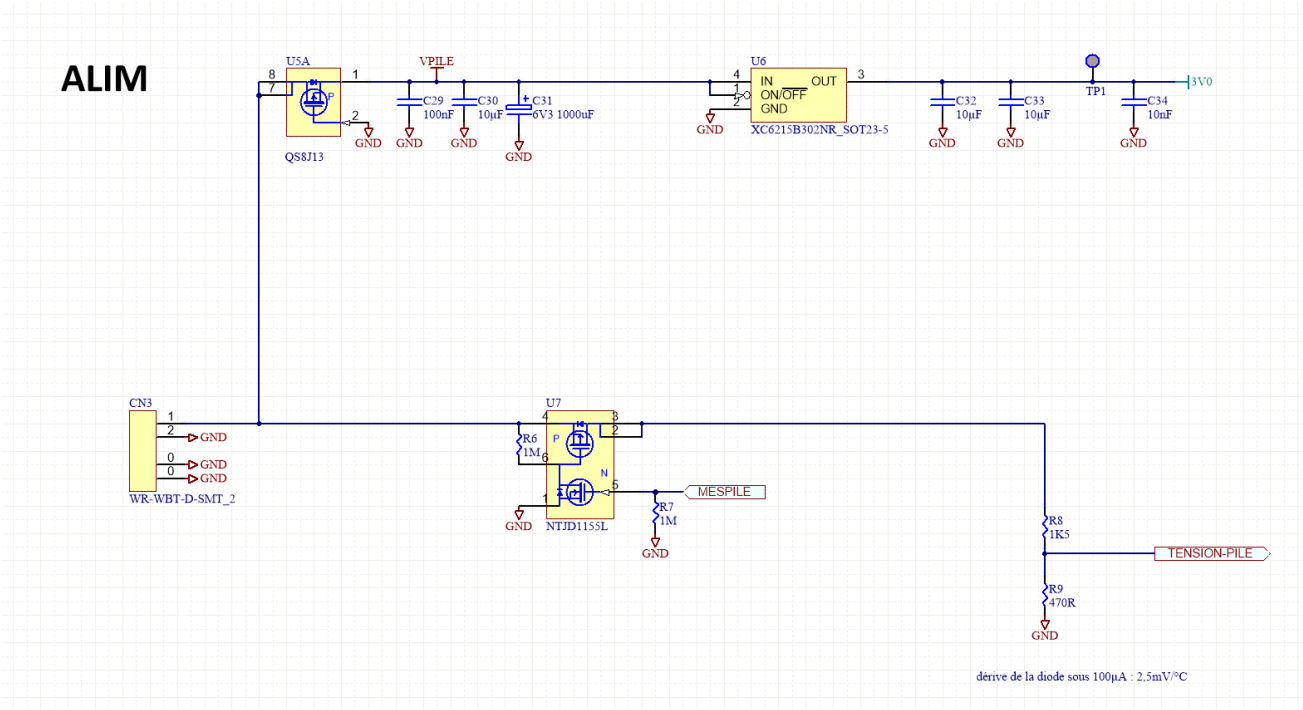
Dans le programme final, les différentes étapes du processus se décomposent comme suit :



E. SCHEMA



IV. ALIMENTATION



V. LIAISON ENTRE CARTE FILLE ET CARTE MERE

1. Technologie utilisée

BLE : Bluetooth Low Energy : Bluetooth à basse énergie anciennement connu sous le nom de *Wibree* puis devenu la marque déposée *Bluetooth Smart*, est une technique de transmission sans fil créée par Nokia en 2006 sous la forme d'un standard ouvert basé sur Bluetooth, qu'il complète mais sans le remplacer.

Le Bluetooth a basse consommation envoi autant de donnée que le Bluetooth classique (1MB/s) mais avec une consommation 10 fois moins importante.

Les modes *BLE* (bande passante plus limitée et très faible consommation) et *Bluetooth* standard (niveau d'émission plus élevé et portée plus grande) sont donc des technologies complémentaires.

2. Avantages - Contraintes

Le Bluetooth basse consommation est plus adapté pour envoyer de petites quantités de données à 1MB/s. Il est parfaitement adapté pour envoyer des données de capteurs de température, d'accéléromètre ou de capteur de géolocalisation etc. Cependant il n'est pas adapté pour envoyer des données en temps réel à un serveur.

Lorsqu'il est placé à l'intérieur, le Bluetooth basse consommation utilise des paquets de publicité pour fournir des informations supplémentaires qui adhèrent à la norme iBeacon, Eddystone. Des balises utilisent le BLE pour détecter la proximité de l'utilisateur et diffuser des informations sur son appareil. Ils disposent d'un UID unique qui active certains emplacements ou fonctionnalités sur le récepteur (envoi d'informations bidirectionnelles inutiles). En BLE les informations sont transmises grâce à des ondes radios qui peuvent pénétrer des barrières physiques. La géolocalisation dans des lieux confinés est donc plus précise par l'utilisations du BLE.

La connexion Bluetooth basse consommation fonctionne parfaitement jusqu'à 30m mais peut difficilement aller plus loin contrairement au WIFI qui peut aller plus loin sans compter l'utilisation d'antenne externe.

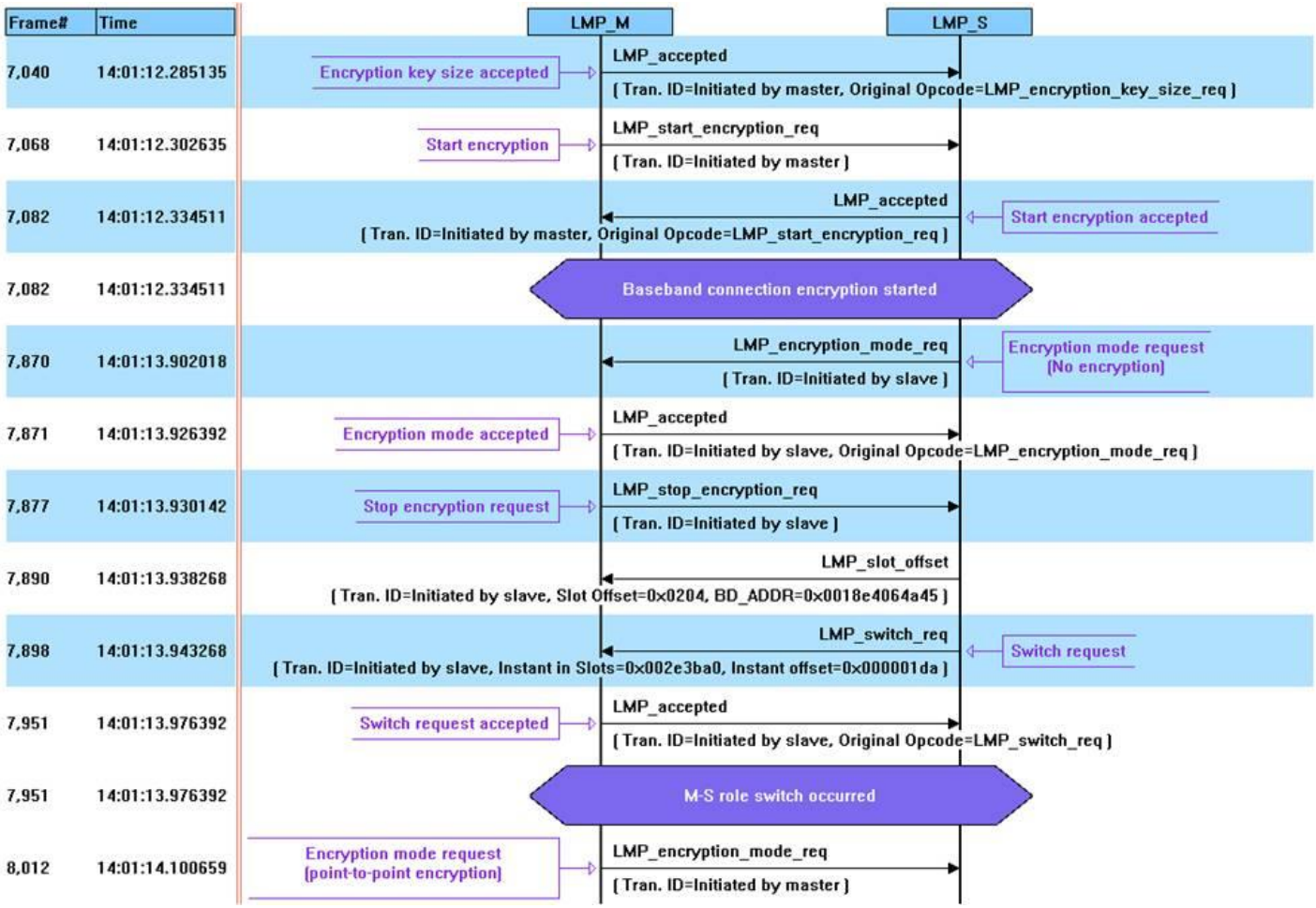
La compatibilité des appareils est également un avantage pour le BLE, la principale fonctionnalité des balises BLE est la capacité à détecter les appareils compatibles.

La sécurité du protocole BLE est légèrement moins efficace que celle du Wifi, encryptions à 128 bits contre 256, cela n'a pas de réel impact dans le cas du rucher.

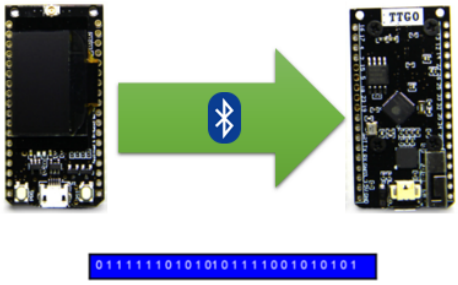
Les appareils Bluetooth consomment moins d'énergie que les appareils Wi-Fi, probablement parce que la portée de diffusion des deux technologies est variée. Les deux technologies émettent un signal avec une

fréquence de 2,4 gigahertz. Les signaux édités par iBeacon se déplacent sur environ 30 pieds, tandis que les signaux Wi-Fi peuvent voyager 10 fois plus loin. C'est pourquoi le Wi-Fi nécessite 10 fois plus de puissance que BLE, même s'il effectue les mêmes tâches. Les appareils Wi-Fi ont besoin d'une puissance considérable, environ 500µW pour dix messages par jour, tandis que BLE ne consomme que 50µW.

3. Message Sequence Chart du BLE



- Création de la tram
- Connexion du Slave au Maitre
- Choix du mode d'encryption
- Envoi des données



Vous avez ci-dessus le fonctionnement du Bluetooth Low Energy. Dans un premier temps, on remarque que temps le master (hôte de gauche) accepte la clé déjà donnée par l'esclave (hôte de droite). Ensuite l'encryption à la demande du maitre démarre et l'esclave lui envoit les infos. On note que le maitre et l'esclave discute dans un mode d'encryption à la demande de l'un ou de l'autre.

VI. LIAISON CARTE MERE VERS GATEWAY

1. Technologie utilisée

Nous avons utilisé la technologie LoRa pour envoyer les informations à la Gateway. La technologie LoRa est un protocole de télécommunications pour l'IoT. Il est bas sur la modulation de fréquence. Il permet de paramétrer le cycle émission/réception en mode continu. LoRa est un réseau longue portée à débit compris entre 0,3 et 50Kbps soit un débit plus faible que le 2G.

La particularité du LoRa est que ce protocole est opérationnel en bas débit, il est économe en énergie et de longue portée. Ce qui est intéressant c'est que notre circuit est branché sur batterie, l'économie d'énergie est donc un élément majeur dans le développement de notre carte.

2. Trame envoyée

Nous avons collaboré avec le groupe Gateway. En effet, pour optimiser les calculs, il fallait que l'on soit coordonné entre l'émission de la trame et la réception de la trame.

Le groupe gateway nous a fourni la forme de la trame avec le nombre d'octets attendu ainsi que l'ordre des grandeurs physiques

1 octet	6 octets	1 octet	6 octets	1 octet	1 octet	2 octets	1 octet	1 octet	2 octets	2 octets	1 octet	1 octet
Type d'émetteur	Adresse MAC	Type d'émetteur	Adresse MAC	Gestion d'erreur	Valeur de la tension de la batterie	Température	Hygrométrie	Température Balance	Vibration	Taux De CO2	Nourrissement (0 à 9)	Masse Balance
						Offset X10						Offset X10

Pour que la trame envoyée ne soit pas trop lourde et qu'elle remplisse les contraintes imposées par la gateway, des offsets sont configurés. Les offsets permettent aussi de gérer les nombres à virgules en le transformant en entier.

Si un offset x10 est appliqué en partant de la ruche interne, en recevant la trace la gateway devra insérer cet offset en divisant par 10.

3. Schéma

LoRa