



STATION DE REGROUPEMENT D'INFORMATIONS CONTENUES DANS UNE RUCHE



KHILIFI Yanis IMBERT Swann
SABATIER Lucas KIPFER Valentin
Rapport de Projet

A l'aube de l'ère des objets connectés, nous sommes en constante recherche de l'optimisation des objets du quotidien.

Cela se traduit par l'amélioration d'un objet à usage unique à un nouvel objet multitâche.

Les objets connectés arrivent petit à petit dans le milieu agricole et plus précisément dans l'apiculture. Dans une optique d'amélioration des conditions de travail des apiculteurs, les ruches connectées commencent à se répandre dans tout le continent.

Notre projet implique de "connecter" une ruche. Celle-ci aura pour objectif de réaliser différentes tâches utiles à l'apiculteur et à son exploitation. Le projet est divisé en trois parties :

- La réalisation d'un module de réception des données météorologiques
- Le développement d'un laser de protection des abeilles contre leurs prédateurs.
- La conception d'une carte électronique recueillant toutes les données internes à la ruche.

Notre groupe a donc pour mission la création de la carte électronique récoltant les données de la ruche.

Sommaire :

Table des matières

I.	Le présentation Projet	3
1.1	Description de notre problématique	3
1.2	Le phénomène d’essaimage	3
II.	La Ruche connectée.....	5
2.1	Récupération des informations	5
2.1.1	Les capteurs de vibration	5
2.1.2	Capteur de CO2, humidité et Température	7
2.1.3	Le module de communication	8
2.2	Réalisation du soft : la communication avec les modules.....	9
2.2.1	Capteur qualité de l’air	11
2.2.2	Lora	12
2.2.3	Micro et Buzzer.....	13
2.2.4	Paramètres modifiable	14
2.3	Réalisation de la carte électronique	15
2.4	Problème survenue	17
	Conclusion :	19

I. Le présentation Projet

1.1 Description de notre problématique

L'objectif de notre projet est de créer une ruche connectée permettant d'éviter un phénomène que nous reparlerons dans une partie ultérieurement.

Pourquoi donc une ruche connectée :

Car elle permet en toute autonomie de récupérer différentes informations tels que la température, le pourcentage d'humidité, le niveau de la qualité de l'air, le son et les vibrations présents dans celle-ci. Dans notre cas, ces données seront envoyées via une connexion sans fil et les modules seront alimentés par batterie. De ce fait, il n'y a aucun câble pour entacher le visuel de la ruche.

Cette ruche connectée permettra de faciliter la vie de l'apiculteur qui pourra garder une vision à distance de sa ruche.

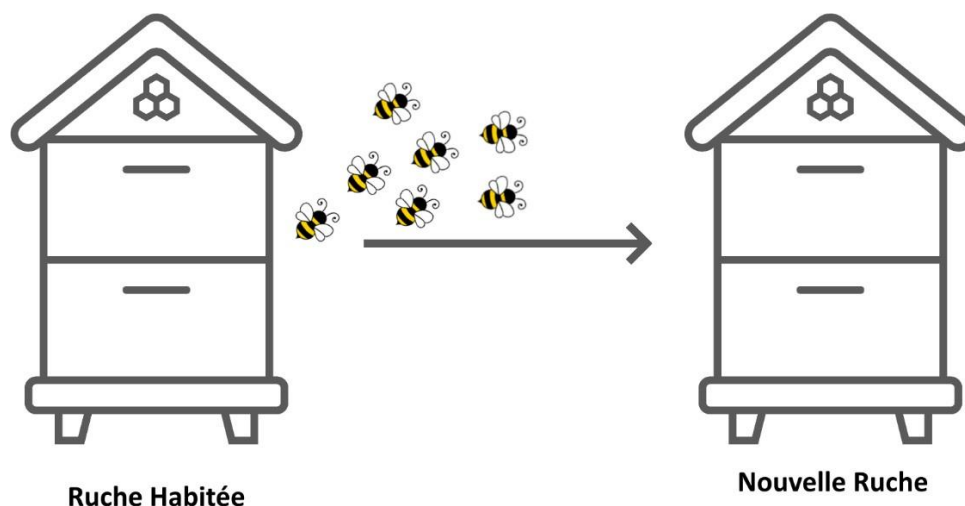
Toutes les modules seront regroupés sur une carte électronique qui sera intégrée à l'intérieur de la ruche.

Repartons maintenant sur un des phénomènes qui favorise l'utilisation d'une ruche connectée :

- "le phénomène d'essaimage"

1.2 Le phénomène d'essaimage

L'essaimage est un phénomène observé dans les ruches, quand la reine et une partie des abeilles quittent la ruche pour former une nouvelle colonie.



L'essaimage se produit généralement au milieu du printemps ou au début de l'été quand le climat est favorable et que les ressources sont disponibles. Souvent avant les miellées, étape qui permet aux abeilles de se développer, de construire une nouvelle ruche et de constituer des provisions.

Il y a plusieurs signes précurseurs mais le principal signe est un changement de fréquence dans la ruche. En temps normal (hors essaimage) la ruche émet des vibrations de l'ordre de 500 Hz alors que dès que l'essaimage approche, la fréquence émise va augmenter et pourra monter jusqu'à 2 000 Hz.

Ce signe précurseur est très facilement récupérable et analysable, pratique pour une ruche connectée. De ce fait le module interne expliqué, dans les prochaines pages, comportera d'un capteur qui aura pour fonction de récupérer cette fréquence.

II. La Ruche connectée

2.1 Récupération des informations

2.1.1 Les capteurs de vibration

Beaucoup de capteurs sont capable de récupérer une vibration : microphone, accéléromètre, capteur piézoélectrique).

La spécificité, ici, est que ce capteur doit avoir quelques spécificités propres à la ruche. Voici les conditions qui doivent être respectées par le capteur :

- Récupérer le signal auditif dont la bande passante se trouve entre 400 Hz et 2KHz
- Récupérer un signal d'amplitude de l'ordre de 10^{-6} G
- Être isolé et/ou insensible à l'eau
- Résistant à la propolis
- Dimensions ($h < 2$ cm)
- Prix

Nous en avons déduit le tableau comparatif suivant :

	Microphone	Accéléromètre (Low Cost)	Accéléromètre (Industriel)	Capteurs Cardiaques	Capteur Piézoélectrique (Guitare)
					
	ADMP401	ADXL345	CMCP1300	VMA340	AD-35
Technologie	Transducteur électromagnétique	Accéléromètre	Accéléromètre	Optique	Transducteur électromagnétique
Bande passante de récupération du signal	20 Hz - 20 KHz	3200 Hz sur un seul axe	0 Hz - 10 KHz	XX	20 Hz - 20 KHz
Sensibilité	XX	XX	100mV/g	XX	XX
Isolation	non	non	oui	non	oui
Résistant à la Propolis	non	oui	oui	non	oui
Dimensions	$H < 2$ cm	$H < 2$ cm	$H > 2$ cm	$H < 2$ cm	$H < 2$ cm
Prix	10,00 €	10,00 €	250,00 €	20,00 €	15,00 €

Il existe diverses technologies de capteurs dont chacune comporte des spécificités. La technologie optique (capteur cardiaque) utilise de la lumière infrarouge pour récupérer le pouls d'une personne. Dans notre cas il est impossible d'utiliser la lumière pour récupérer notre signal.

La bande passante et le signal sont aussi un facteur d'élimination, nous devons avoir un capteur qui réceptionne le signal et qui ne soit pas trop bruité. Nous devons récupérer un signal de fréquence de 2 000 Hz, il est donc préférable d'avoir une fréquence d'échantillonnage de $N * 2\,000$ Hz avec $N > 2$. Le capteur ADXL345 ne convient donc pas à notre application. D'un autre côté le CMCP1300 verra son signal beaucoup trop bruité.

Il ne nous reste enfin deux capteurs : le Microphone et le capteur piézoélectrique. Ces deux capteurs sont les plus adaptés pour récupérer notre information malgré le fait que le microphone soit plus sensible à son environnement extérieur (humidité et propolis).

Pour une meilleure qualité d'acquisition de données, ces deux capteurs seront utilisés ensemble dans le module.

Voici donc les deux modules retenus :

- Le microphone



Nom : ADMP401

Fonctionnalité : récupérer les sons qu'il y a dans la ruche

Alimentation : 3,3 V

Type de signal : Analogique

Bande passante : 20Hz – 20kHz

Rappelons que la ruche émet des vibrations allant de 500Hz jusqu'à 2kHz lors de l'essaimage. La bande passante de notre module nous permet de couvrir ces fréquences. De plus ce module va atténuer les fréquences de 20Hz à 200Hz et les fréquences supérieures à 4kHz. C'est une caractéristique intéressante car rappelons qu'en milieu naturel des bruits d'origine naturelle peuvent apparaître comme les orages ou encore des bruits de véhicules...

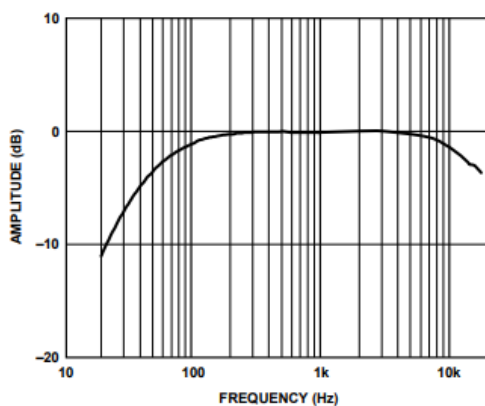


Figure 4. Typical Frequency Response (Measured)

- Le capteur de vibration

Nom : AD-35

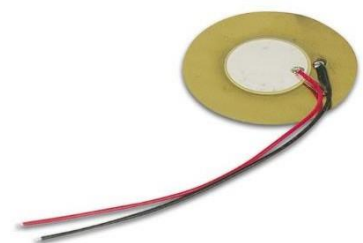
Fonctionnalité : récupérer aussi les sons qu'il y a dans la ruche

Technologie : Piézoélectrique

Alimentation : 3,3 V

Type de signal : Analogique

Bande passante : 20Hz – 20Khz



2.1.2 Capteur de CO2, humidité et Température



Nom : CCS811

Fonctionnalité : Ce capteur récupère la qualité de l'air ambiant et la température

Alimentation : 3,3 V

Type de signal : I2C

Plage d'humidité de fonctionnement : 10% à 95%

TVOC : de 0ppb à 1187ppb

eCO2 : de 400ppm à 8192ppm

Un module qui nous permet de récupérer la concentration en CO2 ambiant, le taux d'humidité et la température ambiant. D'après la datasheet ce module se doit de fonctionner au moins 30 minutes pour obtenir des mesures fiables

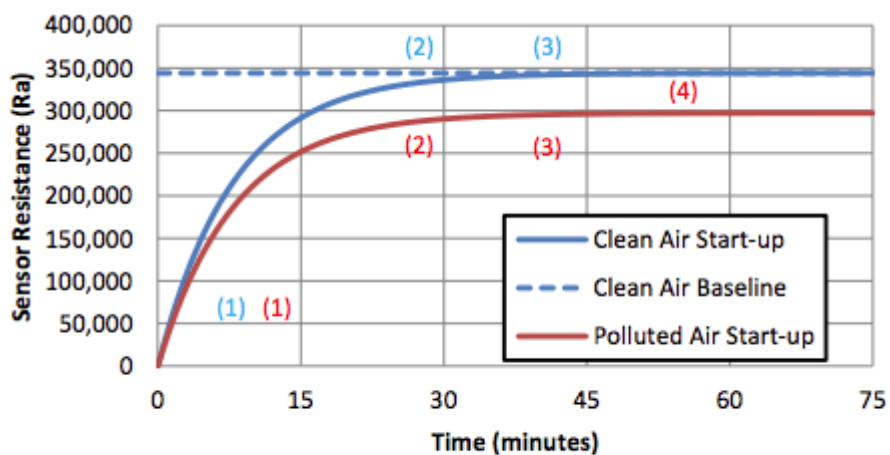


Figure 2: MOX sensor typical warm-up profile

2.1.3 Le module de communication



Nom : LORA RFM95/96/97/98

Fonctionnalité : Permet de communiquer en Lora avec les autres modules

Alimentation : 3,3 V

Type de signal : SPI

Plage de fréquences RF : bande ISM 433 / 868 / 915 MHz

Distance: 5 km(urbain)/15 km(rural)

Le LoRa est un réseau longue porte, à faible consommation énergétique, à bas débit et qui elle offre une très bonne immunité aux interférences. Longue portée car on peut atteindre une distance d'envoi de plusieurs kilomètres (jusqu'à 15km en milieu rural), à faible consommation car en dehors de l'envoi et de la réception des données notre module ne consommera qu'environ 1.5 mA et à bas débit car par trame nous envoyons que quelques dizaines de kilooctet ce qui est largement suffisant pour ce que nous souhaitons envoyer.

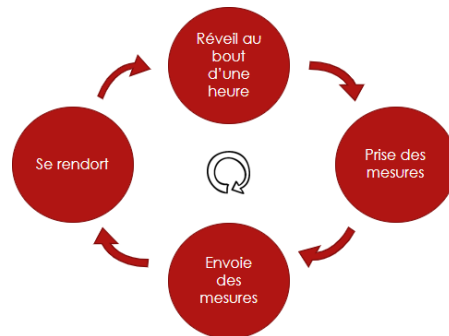
Le déploiement de cette technologie est complètement gratuit car elle utilise des bandes de fréquence sans licence.

La technologie LoRa est devenue au cours des dernières années une très bonne solution de communication pour les projets IOT. Pour notre ruche elle répond tout à fait à nos contraintes à savoir faible consommation et longue distance (la taille de la trame étant très petite, elle ne deviendra pas une contrainte pour notre projet).

2.2 Réalisation du soft : la communication avec les modules

Dès la première séance il a fallu bien mettre en place le cheminement de nos informations, à savoir des capteurs au cloud.

Nous sommes parties d'une représentation simplifiée de notre soft :



On remarque tout de suite la présence d'un thread principal qui va être amené à s'endormir et à se réveiller toutes les heures afin de transmettre une trame de donnée.

Endormir notre microcontrôleur nous permet de limiter énormément la consommation énergétique de ce dernier. Voici un comparatif de la consommation de notre microcontrôleur en fonction des différents modes de fonctionnement :

	Peripherique	Bluetooth	WiFi	Radio	Noyau et mémoire	Processeur ULP	RTC	Consommation
Mode Actif								160-260mA
Mode Modem								3-20mA
Mode Sleep								3-20mA
Deep Sleep								10uA

Allumé :



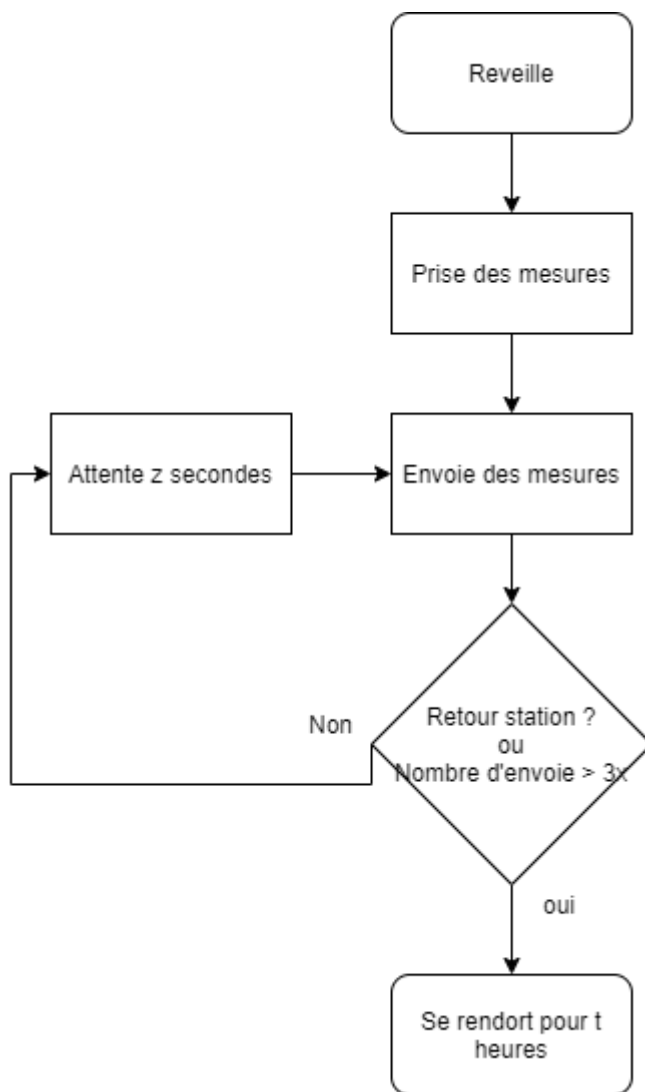
En pause :



Eteint :

Le plus intéressant est donc d'utiliser au minimum le mode Sleep voir deep Sleep pour limiter au maximum notre consommation énergétique. Nous avons choisi pour notre soft de partir sur le mode deep Sleep.

Plus tard nous avons réalisé un model un peu plus complet :



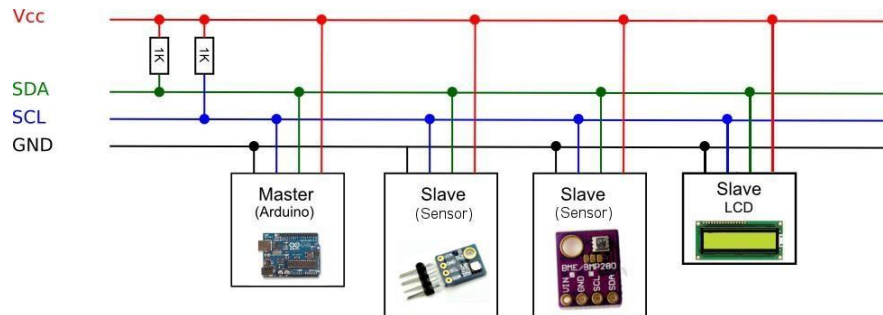
Viens ensuite la récupération et l'envoi de nos données via les différents modules mise à notre disposition à savoir :

- Récupération du son -> Micro
- Récupération vibration -> Buzzer
- Récupération CO2, humidité et température -> capteur qualité de l'air
- Envoie d'une trame de donnée via les ondes radios -> Module Lora

On rappelle que la programmation se fait avec l'IDE d'Arduino.

2.2.1 Capteur qualité de l'air

Le capteur de qualité de l'air est un capteur fonctionnant sur le bus de données I2C. On rappelle que le bus I2C permet de communiquer notre microcontrôleur avec nos modules grâce à trois fils (SDA, SCL, Masse). L'avantage de ce bus est que tous nos modules peuvent être câblés en parallèle. En effet chaque module aura une adresse qui lui sera propre. Si le microcontrôleur souhaite communiquer avec un module il utilisera cette adresse afin de le retrouver.



Le but est donc fixé l'adresse de notre capteur grâce au pin ADDR puis de communiquer celui-ci.

L'utilisation de la librairie « Adafruit_CCS811.h » nous fournit les fonctions nécessaires afin de communiquer avec le module.

Dans le soft on crée un objet « Adafruit_CS811 » et on l'initialise

```
Adafruit_CCS811 ccs;

void setup_CO2() {

    if(!ccs.begin(0x5B)) {
        Serial.println("Failed to start sensor! Please check your wiring.");
        while(1);
    }
    ..
}
```

On vérifie si le module est apte à nous renvoyer des valeurs :

```
while(!ccs.available());
```

On récupère les données (temp, co2, humidité) dans les paramètres de notre objet « Adafruit_CS811 » puis nous les lisons.

```
if(!ccs.readData()){
    ChaineCO2 = "CO2: "+String(ccs.getCO2())+"ppm, TVOC: "+String(ccs.getTVOC())+"ppb   Temp:"+String(temp);
    //Serial.println("CO2: "+String(ccs.getCO2())+"ppm, TVOC: "+String(ccs.getTVOC())+"ppb   Temp:"+String(temp));
}
```

2.2.2 Lora

Lora est un module de communication sans fil à bas débit par radio. Il communique avec notre microcontrôleur via une liaison SPI. Cette fois aussi une librairie Lora est disponible contenant toutes les fonctions nécessaires à la mise en place de ce module.

Dans le soft on commence par inclure les librairies nécessaires :

```
//Libraries for LoRa
#include <SPI.h>
#include <LoRa.h>
```

Puis nous définissons les pins utilisés pour communiquer avec notre module :

```
//define the pins used by the LoRa transceiver module
#define SCK 18
#define MISO 19
#define MOSI 23
#define SS 5
#define RST 14
#define DIO0 26
```

Dans la fonction setup_Lora() nous mettons en place les pins de notre module :

```
//SPI LoRa pins
SPI.begin(SCK, MISO, MOSI, SS);
//setup LoRa transceiver module
LoRa.setPins(SS, RST, DIO0);
|
```

Puis nous initialisons le module en passant bande de fréquence utilisé par notre Lora :

```
if (!LoRa.begin(BAND)) {
    Serial.println("Starting LoRa failed!");
    while (1);
}
Serial.println("LoRa Initializing OK!");
|
```

Pour transmettre un paquet il suffit d'initialiser un paquet pour d'écrire notre chaine dans notre paquet via la méthode print().

```
LoRa.beginPacket();
LoRa.print(Chaine);
LoRa.endPacket();
```

2.2.3 Micro et Buzzer

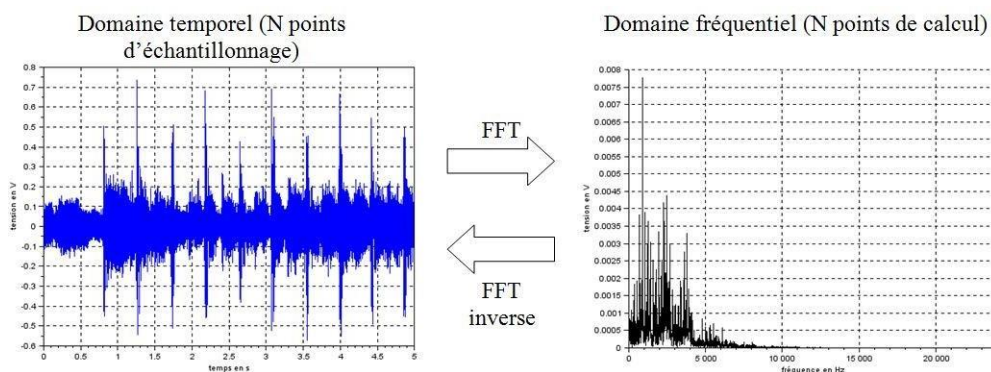
Ces deux modules ont le même objectif à savoir récupérer la fréquence d'un son.

Afin de parvenir à notre objectif nous avons dû utiliser un outil mathématique à savoir la transformée de Fourier.

Tout d'abord, il faut savoir que le son peut être décomposé en signal numérique. En effet, l'extrait reçu par notre microphone peut être représenté dans un premier temps de la manière suivante :



Nous avons ici une représentation sous le domaine temporel, notre signal n'est pas exploitable dans cet état-là. En effet le signal est bien trop complexe (trop grand nombre de points sujet à des modifications dues à des perturbations), une simplification est alors requise. La Transformée de Fourier nous permet d'exploiter le signal acquis, de le simplifier. Comme dit précédemment, le signal sonore reçu à travers notre microphone est acquis sous le domaine temporel, il représente la variation d'amplitude en fonction du temps. Or le domaine temporel peut être décomposé sous forme de somme de sinusoïde. Chaque sinusoïde présente une fréquence ainsi qu'une amplitude. On peut donc réaliser une nouvelle représentation composée de l'ensemble des fréquences perçues grâce à l'analyse des sinusoïdes. C'est le domaine fréquentiel.



La TF n'est cependant applicable que sur des signaux continus. Dans notre cas nous avons une multitude de point. Afin de traiter notre signal nous n'utiliserons pas la TF mais la Transformée de Fourier Discrète. Notre fréquence fondamentale sera la fréquence du son émis par les abeilles.

Dans notre soft nous récupérons tous les t microsecondes, pour chaque module, la valeur émise par notre module et constituons un tableau de valeurs représentant notre signal dans le temps.

```
microseconds = micros();
for(int i=0; i<samples; i++)
{
    vReal[i] = analogRead(CHANNEL_SON);
    vImag[i] = 0;
    while(micros() - microseconds < sampling_period_us){
        //empty loop
    }
    microseconds += sampling_period_us;
}
```

Une fois notre tableau de valeurs constitués, nous la passons dans la fonction FFT de l'IDE arduino afin d'en ressortir la fondamentale de notre signal.

```
double x = FFT.MajorPeak(vReal, samples, samplingFrequency);
return x;
```

2.2.4 Paramètres modifiable

Il est possible de modifier certains paramètres tels que le temps de sommeil dans le fichier Definition.h.

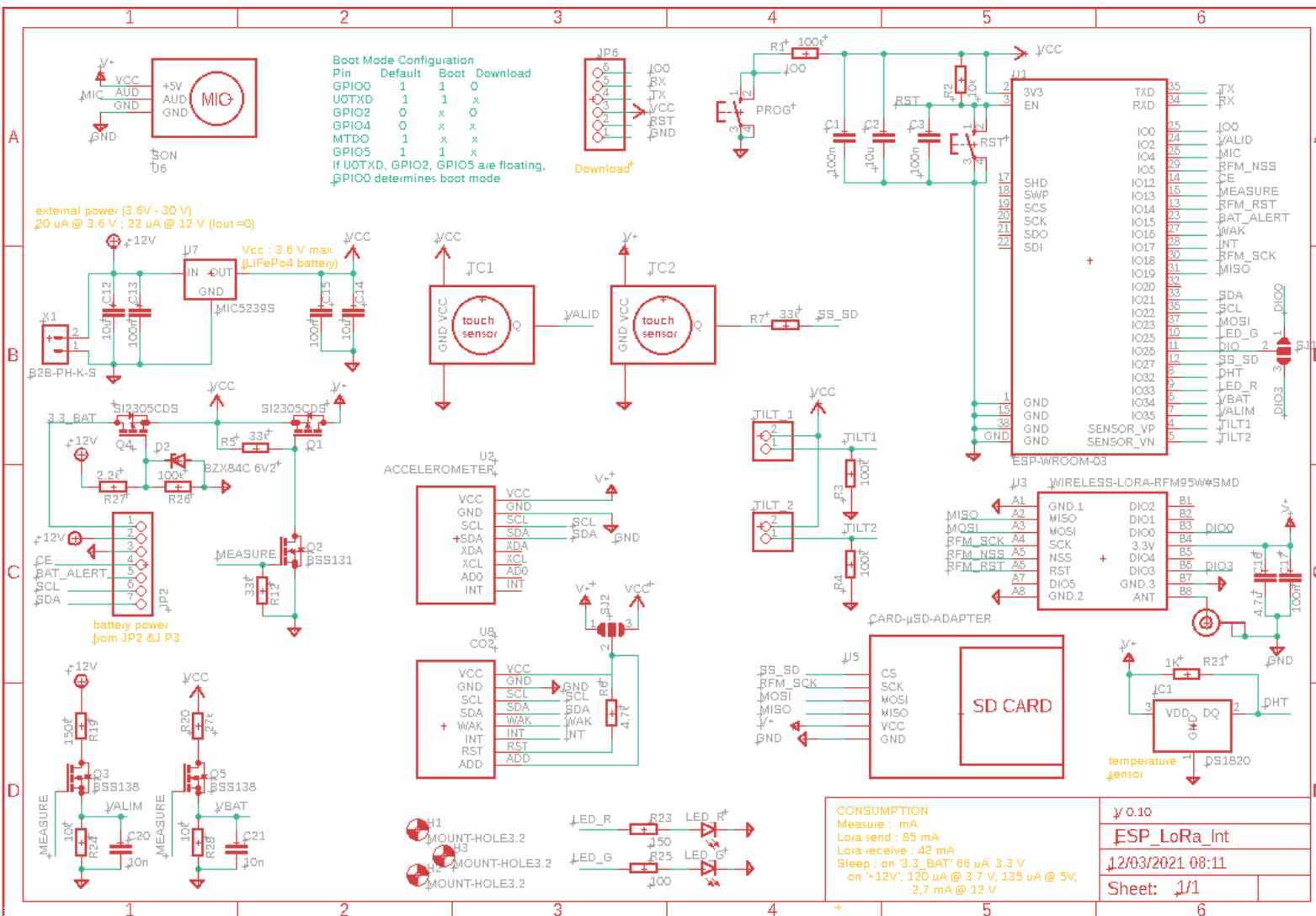
```
#define uS_TO_S_FACTOR 1000000 /* Conversion seconde to micro seconde */
#define TIME_TO_SLEEP 3600 /* Temps Sleep (en seconde)*/
```

Il est aussi possible de modifier, pour le module de son et le piezo, le nombre de valeurs à acquérir par notre soft et la fréquence d'acquisition dans le fichier VarGlobales.h.

```
const uint16_t samples = 2048; //This value MUST ALWAYS be a power of 2
const double samplingFrequency = 4000; //Hz, must be less than 10000 due to ADC
```

2.3 Réalisation de la carte électronique

Pour la réalisation de cette carte, nous avons eu plusieurs problématiques nous ont sauté aux yeux. Dans un premier temps, il fallait qu'elle soit compacte. Pour cela nous avons optimisé l'espace en regroupant les différents modules comme présenté sur le schématic ci-dessous. Ensuite, la présence d'une antenne nous aussi contraint à refaire la carte pour éviter toute interférence de celle-ci avec les autres pistes de la carte.



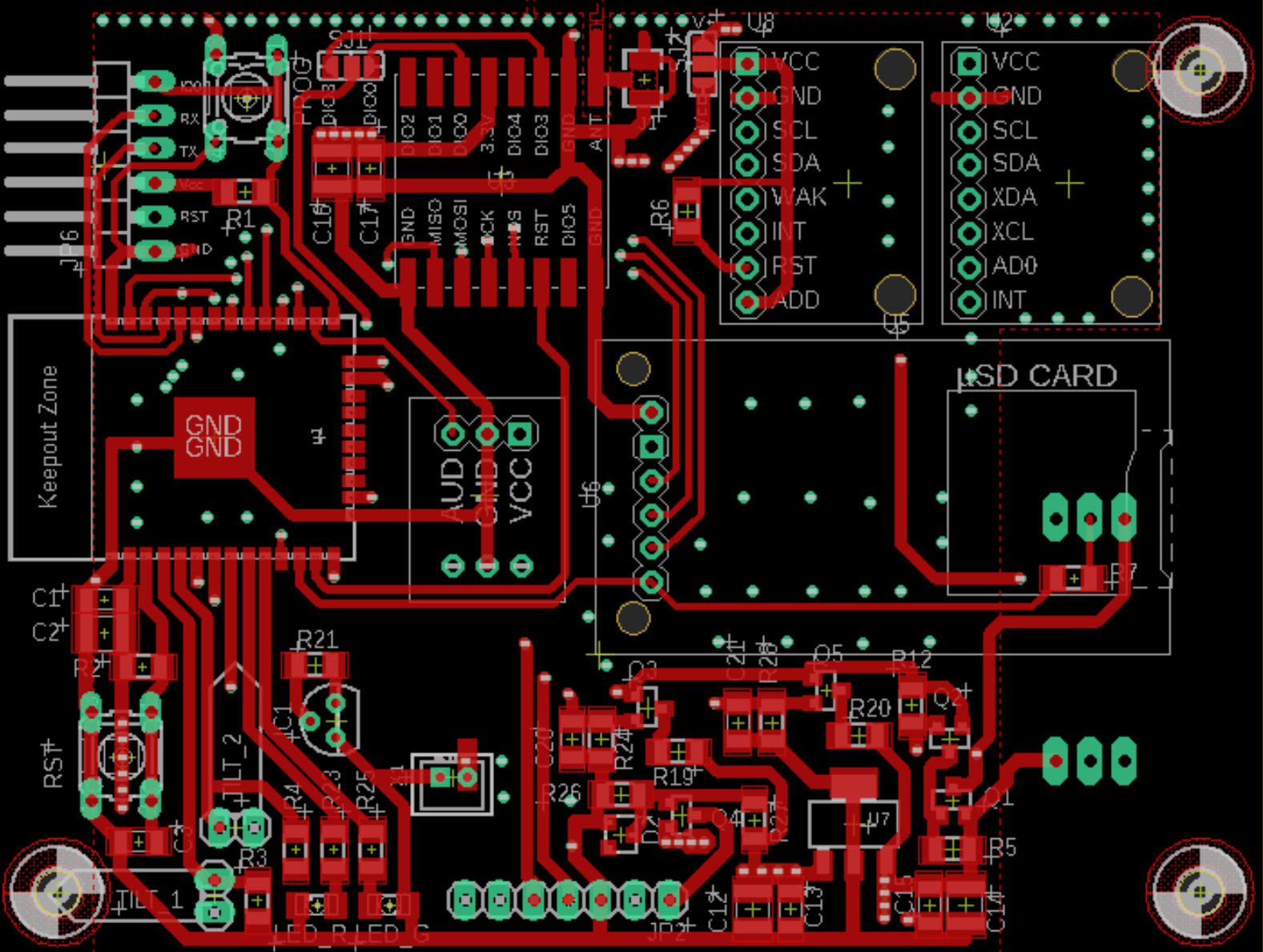
On y trouve sur notre schématic nos modules, notre microcontrôleur, une partie alimentation qui alimentera en permanence notre microcontrôleur et alimentera nos modules uniquement si le sortie MESURE de notre μC est à 1. Cela permet d'éviter que nos modules consomment de l'énergie alors que nous avons décidé de faire une acquisition de donnée 1 fois par heure. Nous avons aussi la possibilité de mettre une carte SD qui peut être une solution de sauvegarde de donnée si le LoRa est amené à avoir un default.

UCA antenna designed by
Fabien Ferrero

(users.polytech.unice.fr/~ferrero/recherche.html)

Polytech Nice Sophia⁺
projet apicole EII-ITI4
2020/2021

Swann IMBERT
Yanis KHILIFI
Valentin KIPFER
Lucas SABATIER



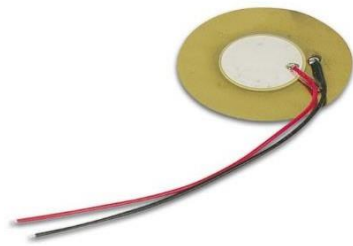
2.4 Problème survenue

A l'arrivée de la carte nous avons décidé de la tester mais nous nous sommes rendu compte que le module accéléromètre ne convenait pas car la fréquence d'échantillonnage n'était pas assez élevée (cf I.2).

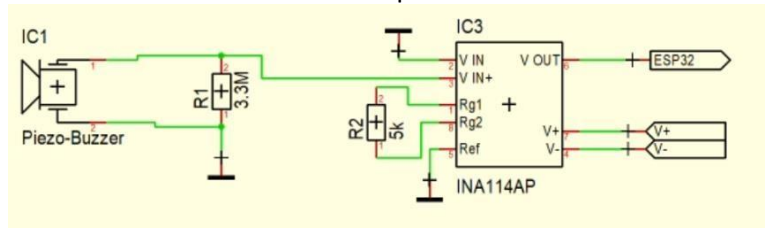
Pour justifier notre choix on utilise le théorème de Shannon qui dit que pour récupérer en intégralité notre signal on doit avoir le double de notre fréquence d'échantillonnage.

Nous avons cherché et trouvé un nouveau module correspondant à notre demande.

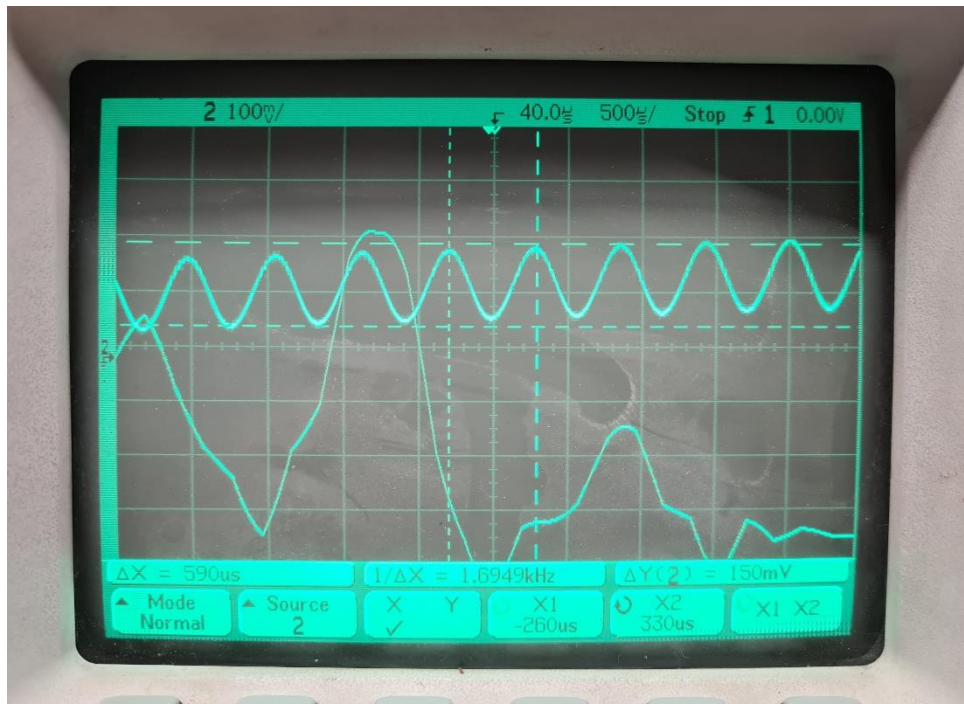
Il s'agit d'un capteur piezo.



Ce capteur piezo pour fonctionner nécessite un amplificateur dont le schéma est :



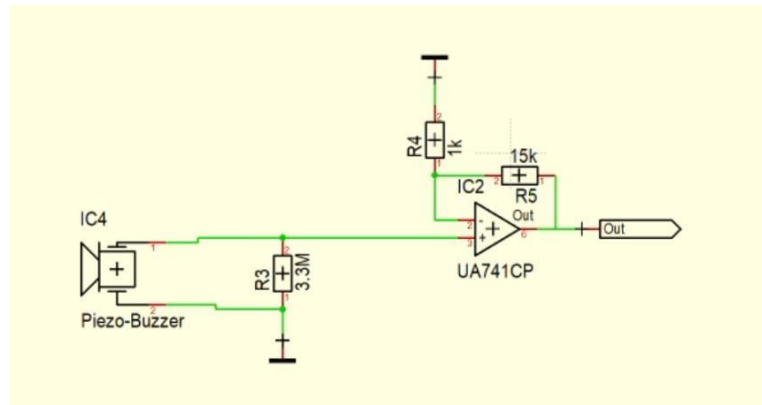
Test du montage avec un bruit de 1.6KHz :



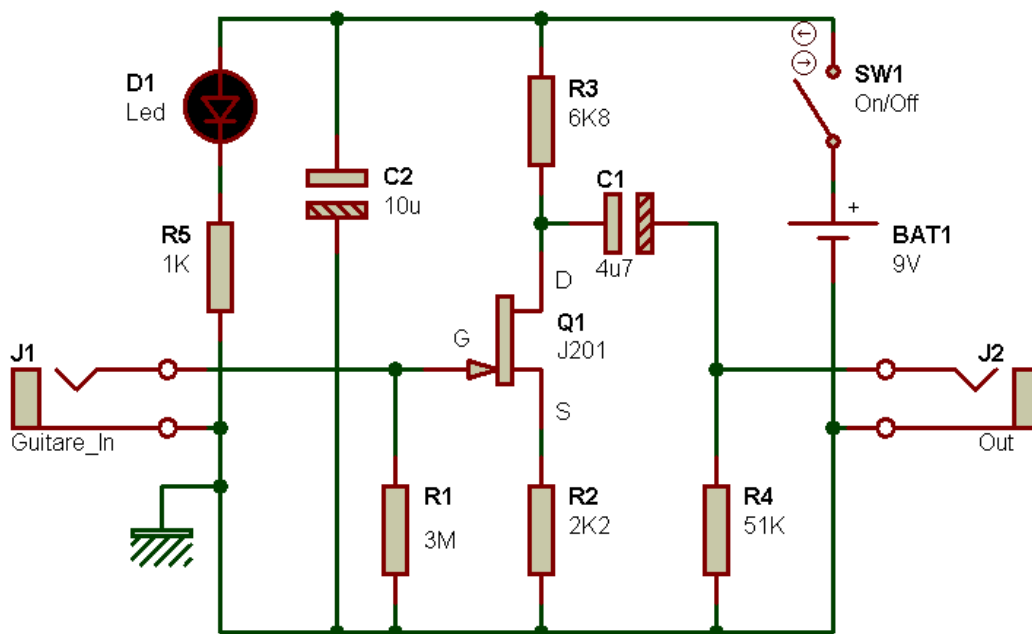
T

Le but de ce premier montage est d'amplifier le signal et non le bruit mais l'inconvénient est qu'il alimente celui-ci en +5 –5V. Pour cela, nous avons trouvé deux autres montages équivalents que nous n'avons pas pu mettre en place par manque de temps.

Le premier montage utilise un amplificateur opérationnel pour amplifier le signal.



Il en est de même pour le second montage sauf que celui-ci utilise un transistor :



Pour information ces deux schémas ne nécessitent pas d'alimentation négative.

Conclusion :

Pour en revenir à notre projet dans sa globalité :

Plusieurs fonctions et test sont encore à mettre en place :

- Ajouter le montage du piezo sur la carte électronique
- Test en situation réelle