

RAPPORT DE PROJET DE 4 ème année



RESEAU DE CAPTEURS LONGUE DISTANCE SANS FILS

Par: **Cong CHEN**
Sonia NDUWAYO
IMA 4

Le 13/05/2016

RESUME

Avec l'émergence de l'internet des objets, les réseaux des capteurs sont en plein essor mais sont actuellement fortement limités par l'autonomie énergétique des noeuds du réseau.

Notre travail consistera donc en la conception d'une (ou plusieurs) cartes électroniques afin de pouvoir réaliser les bases d'un réseau de capteurs basse consommation et longue portée.

Les réseaux de capteurs sont constitués de systèmes électroniques appelés noeuds capables de gérer l'acquisition des données de grandeur physique de leur environnement et permettent une transmission fiable de l'information qui est une grandeur numérique. Le but sera donc de réaliser une carte électronique à base d'un microprocesseur sur laquelle sera connectée différents systèmes embarqués (comme les capteurs,..). La structure modulaire adoptée permettra de connecter à la carte d'autres extensions.

Nous avons choisi un protocole de communication intégré à la fois par le microprocesseur et la puce de transmission radio-fréquences, le protocole SPI. Nous proposons des solutions efficaces de connexions entre cartes en utilisant des aimants néodymes conducteurs.

ABSTRACT

With the emergence of the Internet of Things, networks of sensors are booming but are now severely limited by the energy independence of the network nodes.

Our job will consist of designing a (or several) electronic cards in order to achieve the basis for a network of low-power and long-range sensors.

Sensor networks consist of electronic systems called nodes capable of managing the acquisition of physical quantity data in their environment and provide reliable transmission of information that is a numerical quantity. The goal will be to achieve an electronic card containing a microprocessor which will connect various embedded devices (such as sensors, ..). The modular structure adopted will connect to other extensions card.

We chose a communication protocol integrated by both the microprocessor and radio frequency transmission chip, the protocol SPI. We also suggest effective solutions for connections between cards using magnets néodymes drivers.

SOMMAIRE

Remerciements.....	4
Introduction.....	5
Présentation du LoRa.....	6
A)Description du module LoRa.....	6
B)Fonctionnement des modules LoRa.....	6
C)Tests réalisés.....	10
Conception des cartes	13
1) Logiciel utilisé.....	13
2) Réalisation des empreintes.....	14
3) Protocole de communication SPI.....	15
4) Schémas des cartes réalisées.....	16
5) Miniaturisation des cartes.....	17
6) Connexions des cartes	18
7) Soudure.....	18
Problèmes rencontrés	19
Solutions pérennes.....	19
Conclusion.....	20
Annexes.....	21

REMERCIEMENTS

Nous remercions nos encadrants, Mr.Alexandre BOE, Mr.Xavier REDON et Mr. Thomas VANTROYS pour leur encadrement et leur disponibilité .

Nous remercions également Mr.Thierry FLAMEN pour les suggestions précieuses et son aide dans de nombreuses situations surtout en matière de soudure.

Nous tenons à remercier aussi Mr. Laurent ENGELS pour sa patience et pour la vidéo du projet.

INTRODUCTION

Dans le cadre de notre formation en Informatique Electronique Automatique en 4 è année,nous avons réalisé un projet.

Notre projet s'intitule « Réseau de capteurs longue distance sans fils ».

Un réseau est constitué de plusieurs noeuds capables de communiquer sur de très longues distances.

L'objectif du projet est de réaliser un prototype d'un noeud du réseau.Ce noeud comportera des cartes électroniques sous forme d'un puzzle.Il devra être autonome en énergie et donc intégrer des composants qui consomment moins d'énergie possible.

Dans un premier temps,nous décrirons le fonctionnement de la puce LoRa basée sur un protocole de communication LoRaWAN,conçu pour les objets connectés .Dans la deuxième partie, nous expliquerons les différentes étapes de conception des cartes.

Enfin,nous exposerons les problèmes rencontrés et proposerons des solutions d'interconnexions entre différentes cartes.

I. Présentation du LoRa

a) Description du LoRa

LoRa est une technologie qui permet de réaliser une communication sur de très longues portées à bas débit. Elle est basée sur un réseau qui s'appelle LoRaWAN.

Cette technologie est particulièrement adaptée aux applications du type Internet des objets, capteur en réseaux, environnement, bâtiments intelligents, metering, sécurité ou encore M2M.

Pour réaliser un réseau LoRa, il faut un capteur plus un module LoRa et une alimentation. Ce module s'adapte aux interférences et est capable de choisir le meilleur canal de transmission.

Comparé aux autres types de communication comme WiFi et 3G, le module LoRa présente plusieurs avantages :

- il peut émettre sur une longue distance,
- il consomme moins d'énergie,
- il coûte moins cher
- il peut choisir la bonne position.

Le seul inconvénient du LoRa est son bas débit.

Ce débit est amplement suffisant pour transmettre des données numériques de l'ordre de quelques octets.

b) Fonctionnement des modules LoRa

Afin de comprendre comment deux ou plusieurs modules communiquent et savoir sur quelle distance maximale ils peuvent communiquer, nous avons effectué des tests avec les LoRa Shields.



Fig.1: LoRa + Shield Arduino

Il existe 4 types d'utilisations d'un LoRa-Shield:

- L'émetteur seul qui envoie 8 bits,
- le récepteur seul qui reçoit 8 bits,
- l'émetteur et le récepteur simultanément,c'est à dire envoie 4 bits et reçoit 4 bits,
- enfin,une autre utilisation en mode série.

Pour effectuer notre test,nous avons sélectionné le mode série.

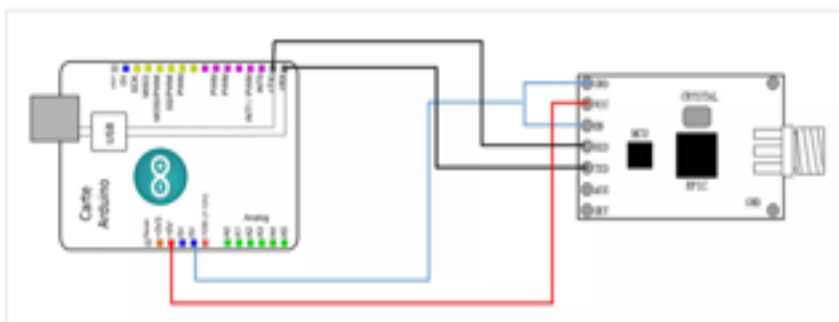


Fig.2: LoRa

Sur la figure 2,nous pouvons voir 3 pins en haut à gauche.Ces pins permettent de choisir le mode de fonctionnement d'un LoRa, les 2 pins au milieu permettent de connecter le port série de l'Arduino avec le port série du LoRa et les 8 pins en bas permettent de connecter les ports de 2 à 9 de l'Arduino sur LoRa. Il suffit de changer le mode des pins en haut pour configurer le LoRa en mode série.

Nous nous sommes servis de deux Arduino UNO afin de créer un système émission/réception.

Pour cela,il faut brancher l'Arduino avec LoRa-Shield comme montré sur la figure ci-dessous :

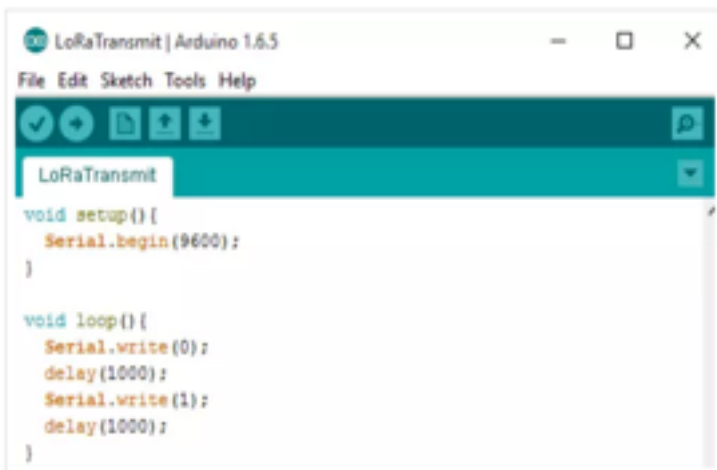


GND sur la masse de l'Arduino
Vcc sur l'alimentation 5V de l'Arduino
En permet l'activation du module s'il est branché à la masse
Rx branché sur le Tx
Tx branché sur le Rx

Pour la réalisation de ce système, nous avons programmé un Arduino UNO en mode émission et un autre en mode réception. Pour le mode émission, nous avons tout simplement envoyé des informations en cascade au module LoRa en utilisant à ce code :

```
void setup(){  
  Serial.begin(9600);  
}  
  
void loop(){  
  Serial.print(0, BIN);  
  delay(1000);  
  Serial.print(1, BIN);  
  delay(1000);  
}
```

Pour programmer l'Arduino, il suffit tout simplement de le brancher au PC, ensuite cliquer sur le bouton téléverser.



Pour le module récepteur, nous avons programmé l'Arduino comme suit:

```
#include <SoftwareSerial.h>

SoftwareSerial mySerial(2, 3); // RX, TX

void setup() {
  // Open serial communications and wait for port to open:
  Serial.begin(9600);
  while (!Serial) {
    ; // wait for serial port to connect. Needed for native USB port only
  }

  //Serial.println("Goodnight moon!");

  // set the data rate for the SoftwareSerial port
  mySerial.begin(9600);
  mySerial.println("Hello, world?");
}

long int count=0;

void loop() { // run over and over
  mySerial.println("Config :");
  //Serial.write("a?");
  Serial.write("#C\x06\x0f");
  int i=0;
  while(i<11){
    int nb=Serial.available();
    int j;
    for(j=0;j<nb;j++){
      int u = Serial.read();
      mySerial.print(i);
      mySerial.print(":");
      mySerial.println(u,HEX);
      i++;
    }
  }
  delay(1000);
  /*if (mySerial.available()) {
    Serial.write(mySerial.read());
  }
  if (Serial.available()) {
    mySerial.write(Serial.read());
  }*/
}
```

L'émetteur envoie des données au récepteur, une fois que le récepteur a reçu les données, il renvoie une réponse à l'émetteur comme le montre la figure ci-dessous:

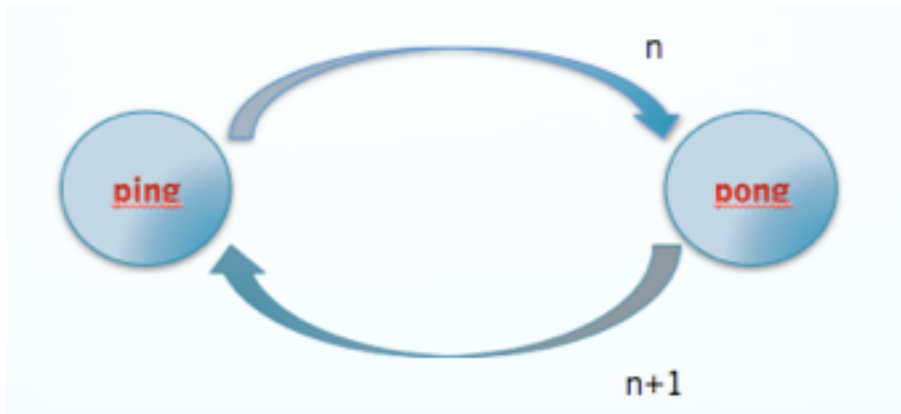


Fig.3: Envoi des données et réponse du récepteur

Cependant, il se peut que l'émetteur arrête d'envoyer des données lorsque le récepteur perd des données, ce dernier ne peut plus répondre à l'émetteur. Afin de permettre à l'émetteur d'envoyer les paquets continuellement,

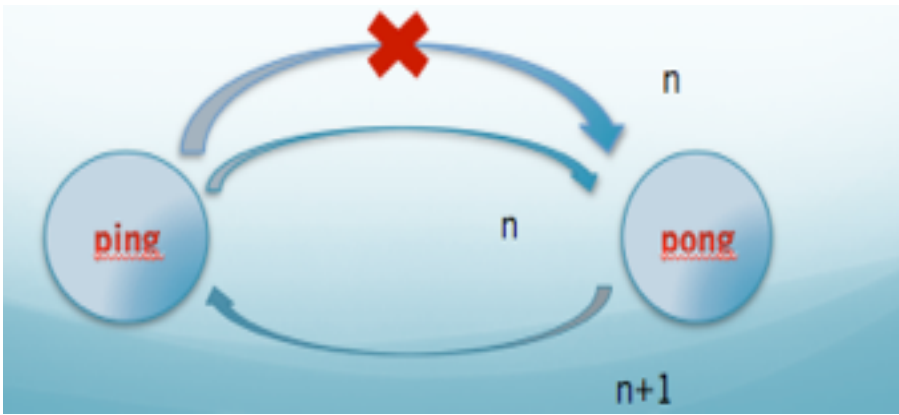


Fig.4: Envoi continu des données et réponse du récepteur
nous avons réalisé un code en langage C :

```
#include <avr/io.h>
#include <util/delay.h>
#include <stdio.h>

#include "serial.h"

#define PING_RESEND    3000

int main(void)
{
  DDRB=0x3f;
  unsigned char ball;
  init_serial(9600);
  ball=1;
  while(1){
    #ifdef PING // le code que dans le ping
      int status;
      while((status=get_serial(PING_RESEND))!=-1) send_serial(ball); // il attend
    et il relance
      ball=status&0xff;
    #else // le code que dans le pong
      ball=get_serial(-1); // il finit
    #endif
    PORTB=ball&0x3f;
    ball++; if(ball==0x40) ball=0;
    send_serial(ball);
    _delay_ms(100);
  }
  return 0;
}
```

c) Tests réalisés

Par défaut, le LoRa fonctionne en mode 5. Ce n'est pas le mode qui permet une communication sur de longues distances, nous avons donc changé le mode pour qu'il puisse émettre sur de très longues portées.

Nous avons effectué les tests en milieu urbain et rural.

En milieu urbain, nous avons laissé l'émetteur à *Polytech' Lille* et nous avons déplacé le récepteur jusqu'à la résidence *Hélène Boucher*. Les Leds du récepteur se sont allumées; preuve que le récepteur reçoit bien les données envoyées par l'émetteur. Ensuite, nous avons fait d'autres tests en milieu rural. Nous nous sommes déplacés jusqu'à *Cysoing* et *Camphin*.

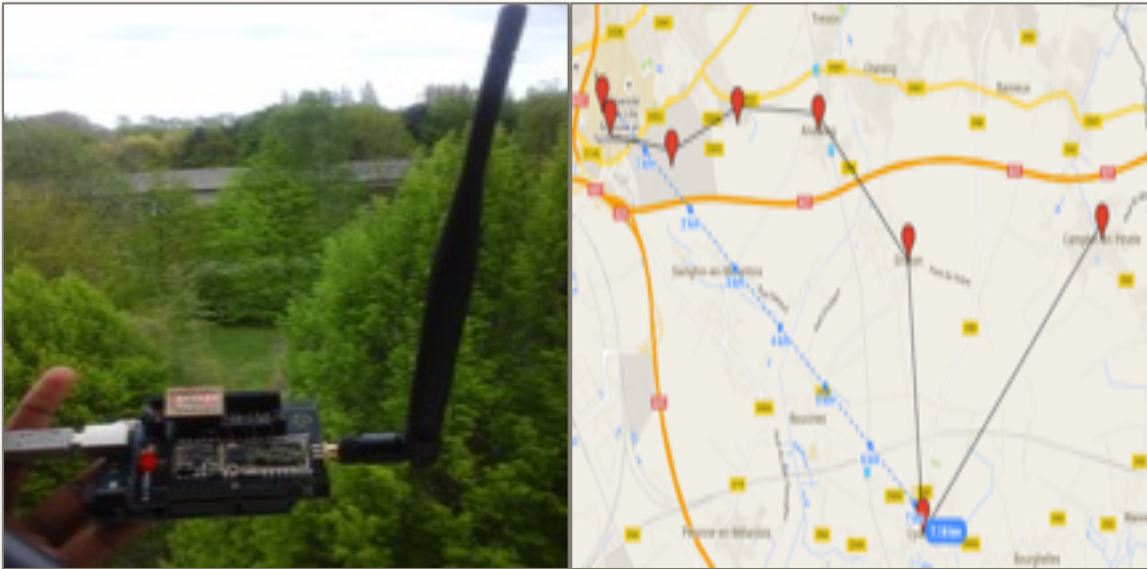


Fig.5: Module émetteur à Polytech'Lille

Voici une photo à *Camphin* qui montre que le récepteur a bien reçu les données de l'émetteur avec les Leds rouges comme témoins.

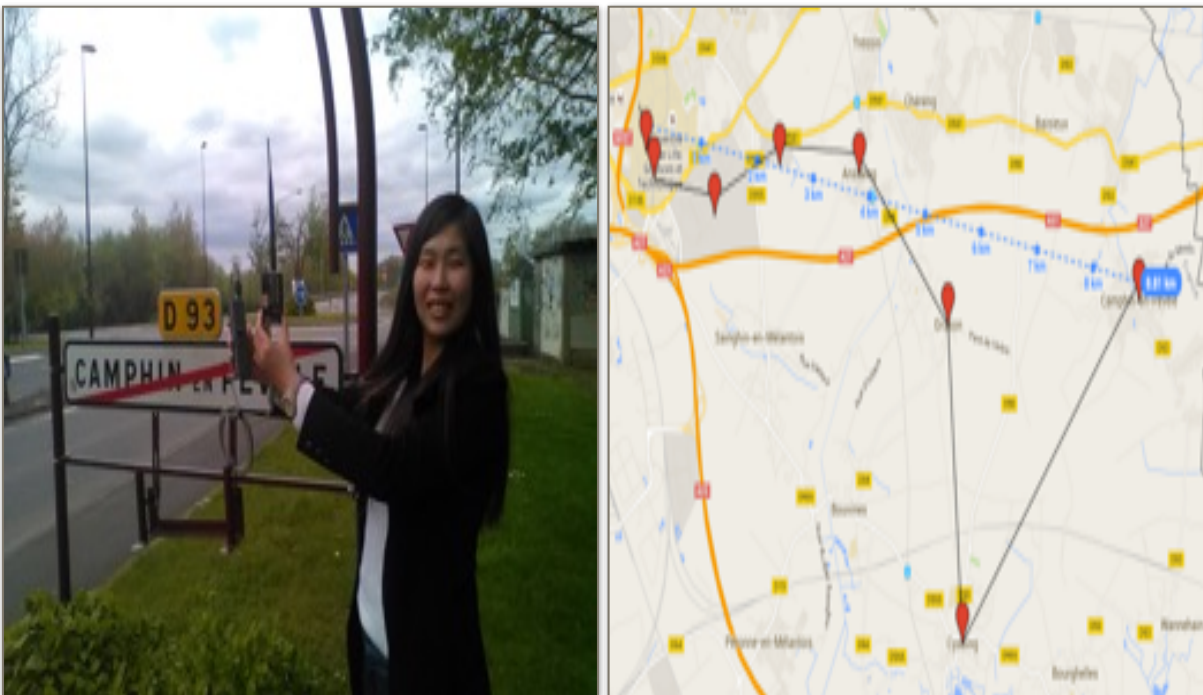


Fig.6: Module récepteur à Camphin-En-Pévèle

D'après tous les tests que nous avons effectués, nous pouvons conclure que les modules LoRa émetteur et récepteur sont capables de communiquer sur une très longue distance allant jusqu'à 9 km quelque soit les obstacles rencontrés(bâtiments, arbres,..).

Il convient, néanmoins de noter, qu'à *Cypsoing* pendant que nous effectuions le test dans un endroit un peu en relief, nous avons observé une perte des données émises.

En plus des tests réalisés avec l'Arduino UNO + LoRa Shield, nous avons essayé aussi de réaliser un autre test avec l'Arduino NANO + module LoRa 868. Le montage a été fait comme le montre la figure ci dessous :

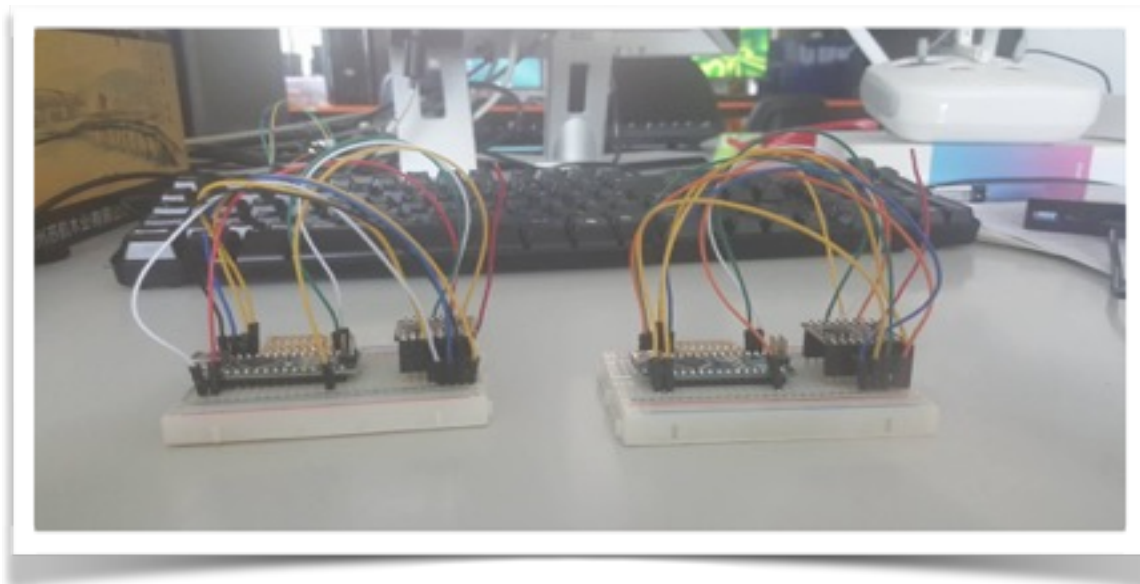


Fig.7: Montage Arduino Nano+ LoRa Shield

Nous avons trouvé un programme pour l'IDE Arduino mais ce code est prévu pour être utilisé avec un montage particulier qui n'est pas documenté.

Il faudrait bien analyser le code pour récupérer le montage, malheureusement le temps nous a fait défaut. C'est l'une des difficultés que nous avons rencontrées.

II. CONCEPTION DES CARTES

Le but du projet étant de reproduire la base d'un noeud qui consomme moins d'énergie, nous nous sommes juste limitées à la conception de 3 cartes principales à savoir :

- La carte d'alimentation,
- La carte de transmission RF LoRa ,
- La carte micro-contrôleur MC

Pour assurer l'autonomie énergétique, 3 facteurs entrent en jeu:

- les composants utilisés
- la dimension de la carte
- le moyen de communication entre cartes

Les composants choisis sont des composants CMS (composants montés en surface). Ils sont très petits et très légers et consomment moins d'énergie mais sont très difficiles à souder.

Les cartes doivent être de petite taille pour minimiser les coûts de celles-ci. Comme moyen d'échanges des données, nous avons choisi le protocole de communication SPI car il assure un bon débit de communication et est intégré dans les composants choisis.

1. Logiciel utilisé

Pour designer les cartes, nous avons utilisé le logiciel de conception électronique ALTIUM Designer disponible sur les machines de la salle C201 à Polytech'Lille.

ALTIUM Designer est un outil complet de développement de produits électroniques et inclut un outil de routage, un outil de simulation, ..

Ainsi donc, pour créer un circuit imprimé, il faut créer une schématique qui décrit le schéma électrique du circuit et un document PCB décrivant la position des composants sur la carte et le routage des interconnexions: pistes.

2. Réalisation des empreintes

Afin de réaliser nos propres cartes: Microcontrôleur,LoRa et d'alimentation, nous avons designé les librairies de certains composants tels que :

le LoRa 868

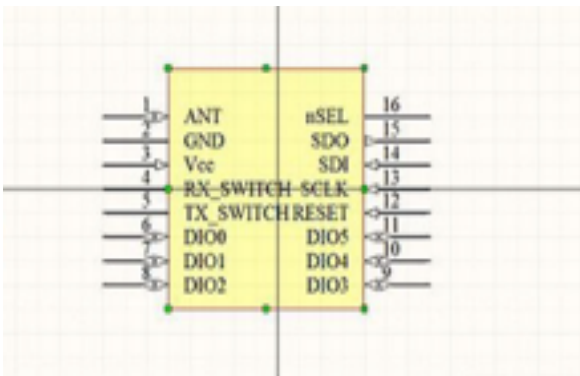
les régulateur 3.3V,régulateur 5V,

le port USB,

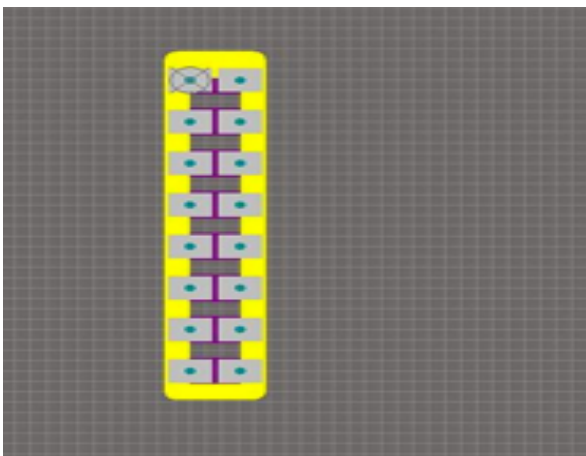
les diode et Leds,...

Pour créer une librairie, il faut d'abord créer une schématique(sur laquelle on place tous les composants dont on a besoin) et un PCB Foot Print Design(qui permet de placer les composants et de réaliser les connexions automatiquement).

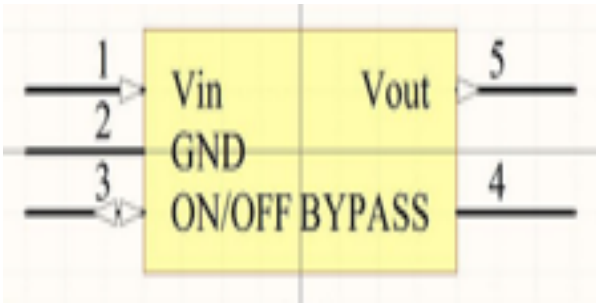
A titre d'exemple,pour le LoRa 868, nous avons créé une schématique comme suit:



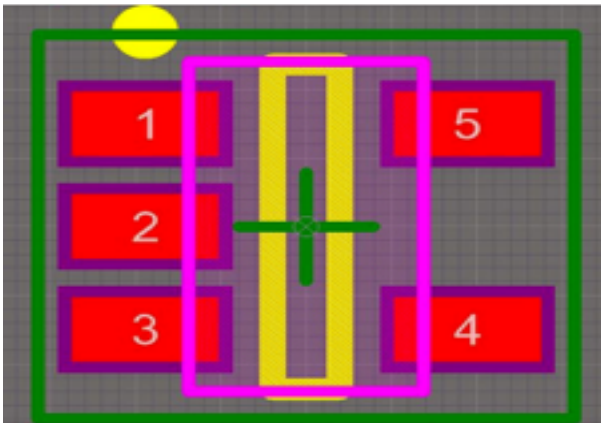
le PCB obtenu est le suivant :



Pour le régulateur 3.3V, la schématique obtenue est la suivante:



le PCB:



Ensuite, nous avons lié le PCB à la schématique.

L'empreinte du module LoRa n'étant pas correcte au départ, nous l'avons éditée à partir du packaging SOT-143 sur IPC Compliant Wizard. A cause d'une erreur de paramétrage, l'écartement des pastilles était inférieur à l'écartement réel des broches. Le problème a été corrigé temporairement par l'utilisation de barrettes femelles traversantes soudées en biais. Une autre solution consisterait à modifier l'empreinte. Avec le recul, nous réaliserons cette fois une empreinte à partir de zéro comme nous l'avons fait pour le connecteur mini USB.

3. Protocole de communication SPI

Pour que toutes les cartes puissent communiquer, il a fallu déterminer un protocole de communication supporté à la fois par le micro-contrôleur AtMega328P et la puce LoRa.

Parmi les protocoles de communication existants (SPI, I2C, ...), nous avons choisi le protocole SPI.

Le protocole SPI a été créé par Motorola ,ensuite adopté par d'autres fabricants tels que Atmel ,Microchip,TexasInstruments,...

C'est un protocole simple qui permet la transmission des données avec un très grand débit et ne nécessite pas l'adressage des données.

La transmission s'effectue sur 2 fils monodirectionnels (MISO,MOSI),une horloge indépendante est fixée par le maître, et un fil qui permet de sélectionner l'esclave.

Le noeud sans fil étant constitué de plusieurs cartes qui communiquent entre elles dont la carte LoRa en charge de la réception des données provenant d'un autre noeud.Ces données numériques sont ensuite envoyées vers le microcontrôleur(carte MC) qui à son tour ,effectue un traitement sur ces données.

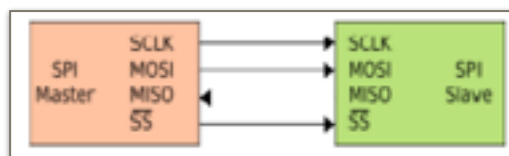


Fig.:Liaison SPI maître-esclave

4. Schémas des cartes réalisés

Avant de réaliser les cartes, nous avons d'abord essayé de comprendre le rôle de chaque composant.Tous les composants CMS ont été soudés sur la face Top.

-LA CARTE D'ALIMENTATION

En premier lieu,nous avons d'abord réalisé la carte d'alimentation. La carte d'alimentation joue le rôle d'une batterie. Elle alimente les autres cartes.

Pour plus de flexibilité, nous avons conçu cette carte de telle sorte qu'elle puisse être alimentée soit via un port USB ou une alimentation externe(pile de 9V).

Devant alimenter 2 cartes en 5V et 3,3V nous avons donc rajouté des régulateurs qui convertissent les 2 sources de tension en 5V(carte microcontrôleur) et en 3,3V(carte LoRa).



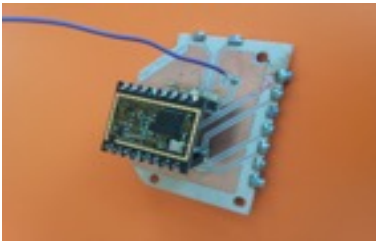
Annexe 1

-LA CARTE LoRa

La carte comporte une puce LoRa qui est basée sur le protocole LoRaWAN décrit en introduction.

La puce permet une transmission sur de très longues portées de communication.

Sur la carte, on retrouve une capacité de découplage, la puce LoRa ainsi qu'un fil de cuivre qui sert d'antenne.



Annexe 2

-LA CARTE MC

La carte a été conçue à l'image d'une carte Arduino.

Elle comporte un microprocesseur AtMega 328P avec une mémoire de 2kb de RAM et 1 kb de mémoire flash.

Cependant, le micro-contrôleur a besoin de plusieurs composants pour son bon fonctionnement.

Le micro-contrôleur a été relié à un connecteur ICSP (connecteur à 6 broches) qui permet de programmer ce dernier.

On retrouve aussi un quartz 'Crystal' 16MHz, 2 condensateurs qui permettent de fixer la fréquence de fonctionnement du micro-contrôleur ainsi que d'autres capacités et résistances pour assurer son bon fonctionnement.

Annexe 3

5. Miniaturisation des cartes

Après avoir placé les composants , il faut procéder au routage des cartes, en essayant de respecter les contraintes d'encombrement, contraintes de réalisation:

- La taille des cartes doit être minimale pour limiter le coût.

- La taille des pistes conductrices de courant (en cuivre):

Les pistes doivent être à une certaine distance des composants pour les isoler. En effet, à haute fréquence si les pistes sont proches, il peut y avoir des perturbations magnétiques.

- Les pistes doivent être de longueur minimale pour éviter la résistance qui engendre l'atténuation de la puissance transmise.

En routant, il faut aussi éviter les angles droits.

Une fois routées , les cartes ont été envoyées au service EEI de Polytech'Lille pour le tirage.

6. Connexion des cartes

En réalisant les cartes, il fallait déterminer un moyen d'interconnexions. Partant sur l'idée d'un puzzle, nous avons décidé d'utiliser des aimants néodymes conducteurs.

Avant de valider notre choix, nous avons effectué un petit test (Annexe 4) qui consistait à allumer une Led placée sur une plaquette et une pile placée sur une autre plaquette reliées entre elles avec des aimants. Le test s'est avéré concluant.

Lors de la réalisation des cartes sur Altium, nous avons dimensionné les encoches de 3,1 mm de diamètre sur 1 mm de profondeur dans lesquelles les aimants seront soudés.

7. Soudure

La soudure a été la dernière étape de réalisation des cartes. Lorsque nous avons reçu les cartes, nous avons effectué des tests de court-circuit: Vérifier s'il n'y a pas de court-circuit entre 2 pistes. Pour cela, nous avons utilisé un multimètre.

Les composants CMS sont très délicats à souder à cause de leur taille (surtout AtMega328P). Pour les souder, nous avons utilisé de la pâte à souder que nous

avons déposé sur les empreintes et ensuite posé les composants par dessus et enfin nous avons fait passer les circuits imprimés au four à refusion à 340°C environ 7 min.

Les aimants et les vias ont été soudés à l'aide du fer à souder et de l'étain.

III. PROBLEMES RENCONTRES

Lors de la conception des cartes , nous avons rencontré plusieurs difficultés.

La première difficulté a été celle de la soudure. La soudure a été une tâche particulièrement compliquée, principalement due à la taille des composants CMS et au fait que les aimants sont difficilement soudables.

A force de souder les aimants, ils perdaient leur aimantation.

Armées de patience, nous sommes arrivées à souder tous les aimants. Pour pallier au problème de désaimantation, nous avons ajouté une rangée d'aimants intermédiaires pour augmenter la force d'aimantation.

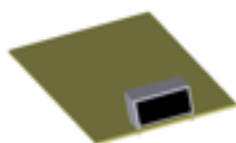
La deuxième difficulté a été celle des vias non métallisés. En soudant les vias, des bulles d'étain se formaient sur les extrémités. Nous avons donc utilisé une technique qui consiste à mettre un bout de fil dans le via ensuite le souder sur les pistes qui sont connectés à ce via.

IV. SOLUTIONS PERENNES

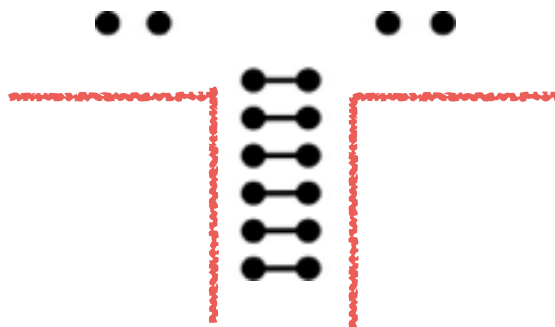
D'autres solutions d'interconnexions sont envisageables:

-La première solution consiste à souder une cage métallique sur le circuit et ensuite rentrer de force l'aimant à l'intérieur de la cage afin de bien faire contact. Le métal étant conducteur, l'aimant assurerait une bonne conductivité car n'ayant pas été soumis à la forte chaleur de la soudure classique.

En vert, le circuit imprimé, en gris, la cage métallique et en noir l'aimant



-La deuxième solution serait de réaliser une carte d'alimentation avec une partie descendante portant des aimants d'interconnexion entre les cartes MC et carte LoRa. Les aimants seraient soudés à plat sur des pastilles en cuivre. Les cartes MC et LoRa seraient connectées à la carte d'alimentation en les posant sur cette dernière.



Les aimants peuvent être fixés à plat par soudure classique (plus facile que dans l'autre sens mais aussi destructeur si on chauffe trop - 340°C), par le four à refusion (facile et seulement à 250°C) ou même avec de la colle conductrice.

CONCLUSION

L'objectif du projet était de réaliser les bases d'un réseau de capteurs sans fils. Nous avons pu réaliser les 3 cartes.

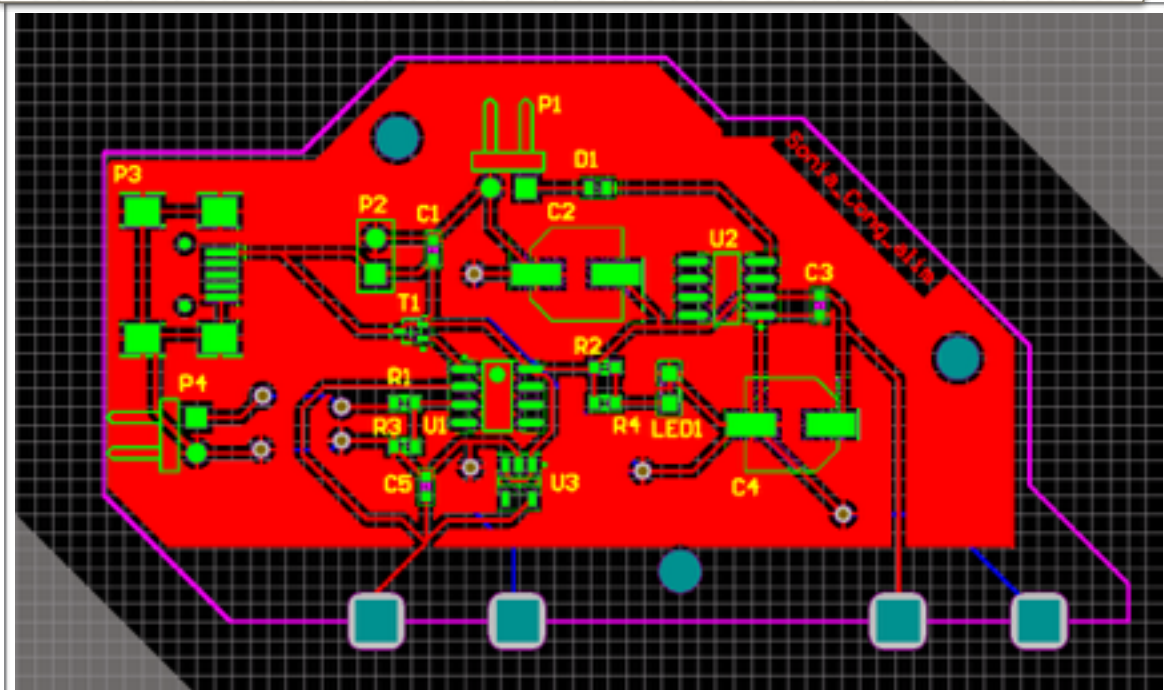
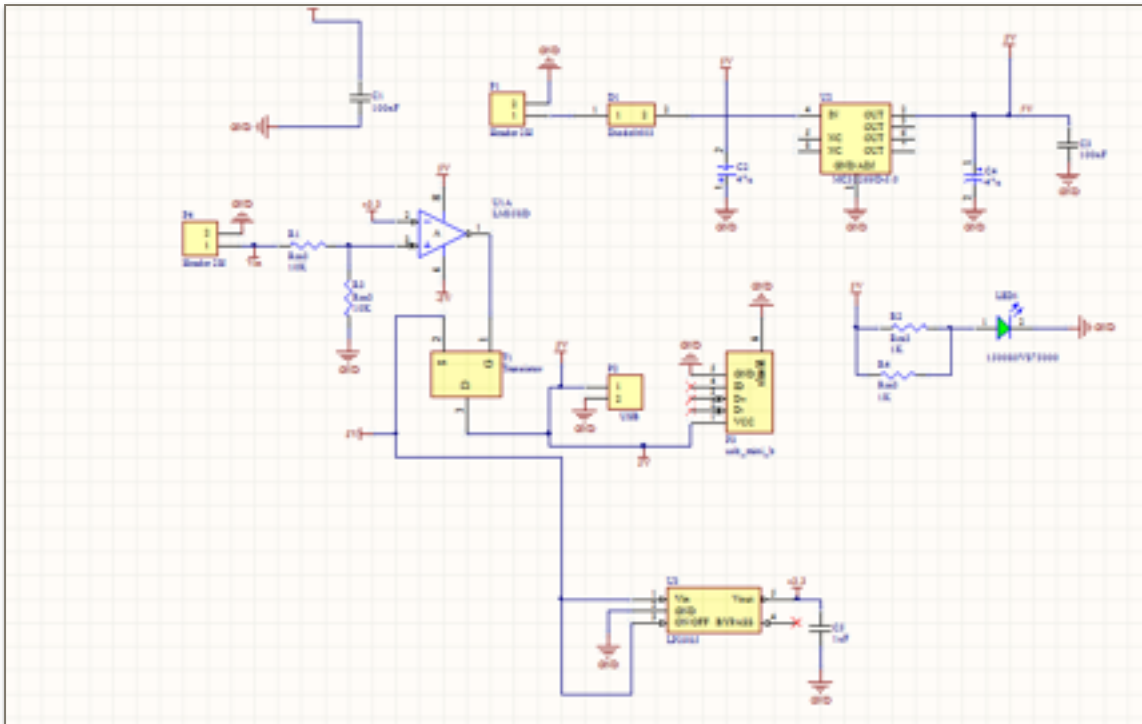
La carte d'alimentation fonctionne correctement ainsi que la carte LoRa, malheureusement, la carte MC est court-circuitée; le test de fonctionnement du micro-contrôleur AtMega328P n'a donc pas été effectué.

Les erreurs de dimensionnement sur les PCB dans la réalisation des empreintes ont énormément freiné notre travail.

Il aurait été intéressant de réaliser le test de la force du signal. Selon le datasheet du LoRa-Shield, pour effectuer ce test, il faudrait changer le mode en auto-test. Ce module incorpore un auto-test en appuyant sur un bouton. Pour tester le RSSI du récepteur, il suffit d'observer les Leds. Lorsque la puissance est maximale, les 8 Leds sont allumés.

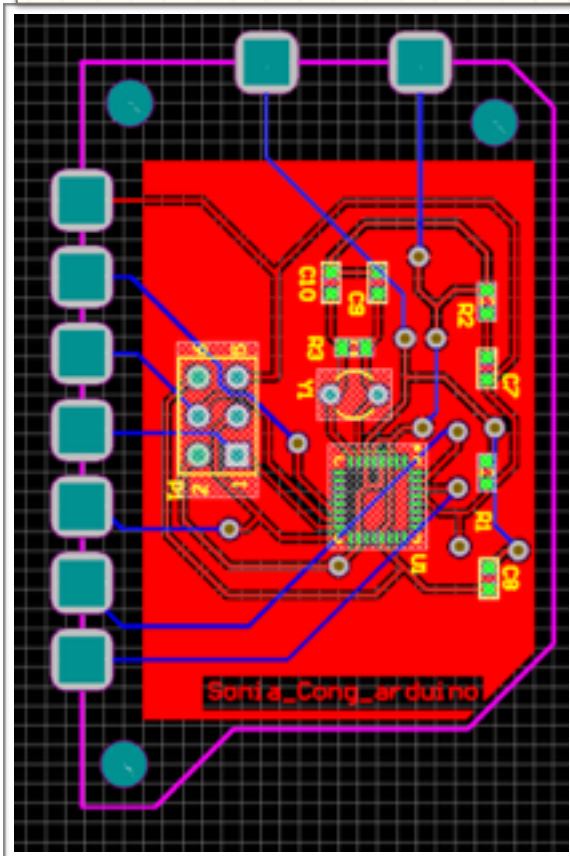
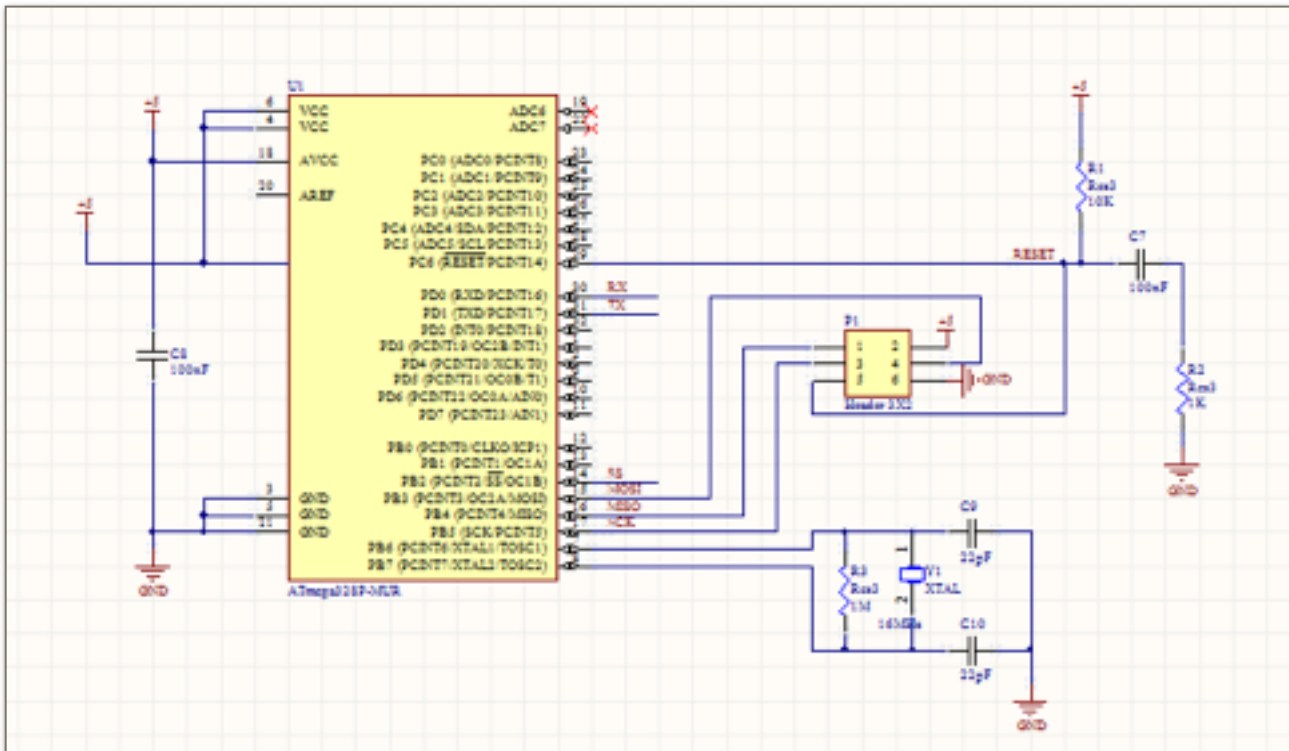
ANNEXES

Annexe 1: schématique et PCB de la carte d'alimentation



A partir de la gauche; Broche 1: 3,3V
Broche 2: 0V(masse)
Broche 3: 5V
Broche 4: 0V(masse)

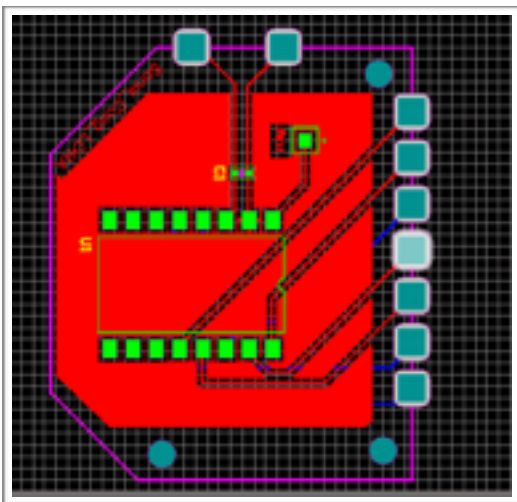
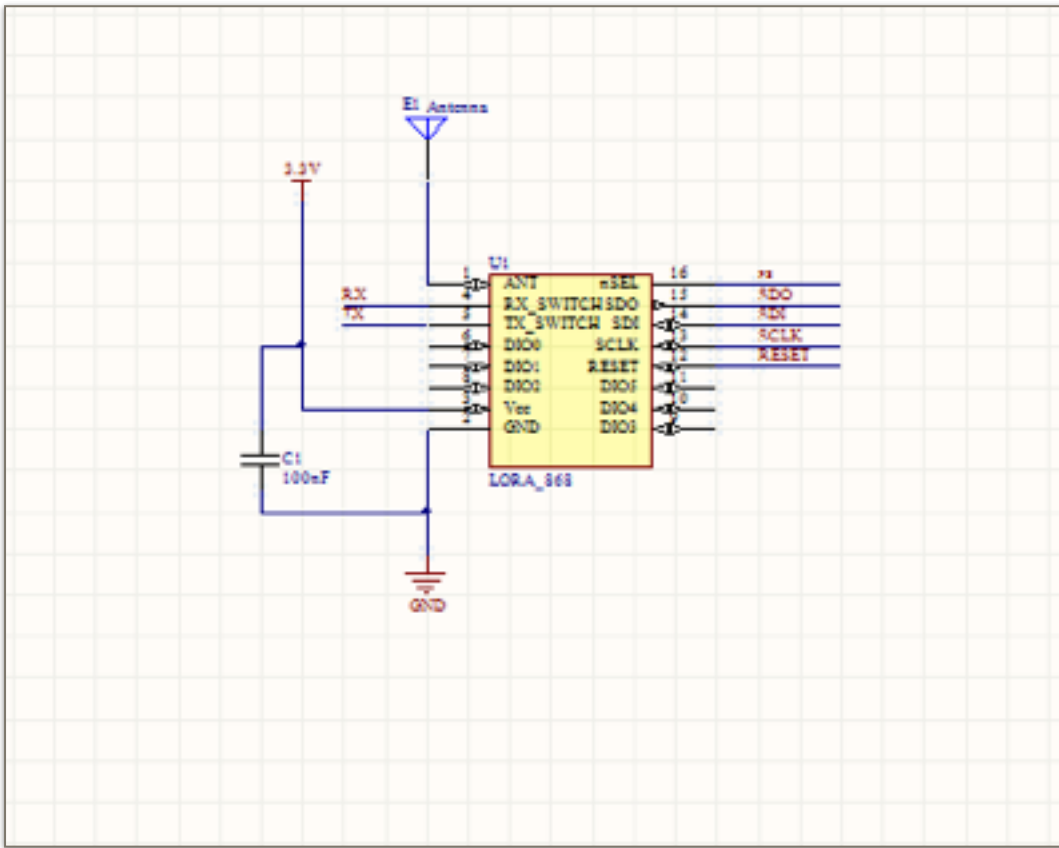
Annexe 2: schématique et PCB de la carte microcontrôleur



Broches d'en haut : Broche 1: 5V
Broche 2: 0V(masse)

Broches sur le côté de haut en bas:
Broche 1:Reset
Broche 2:SS
Broche 3: MOSI
Broche 4:MISO
Broche 5:SCK
Broche 6:Rx
Broche 7:Tx

Annexe 3: schématique et PCB de la carte LoRa



Broches d'en haut : Broche 1: 3,3V

Broche 2: masse

Broches sur le côté de haut en bas:

Broche 1:Reset

Broche 2:SS

Broche 3: MISO

Broche 4:MOSI

Broche 5:SCK

Broche 6:Tx

Broche 7:Rx

NB: Sur les captures d'écran, les noms des broches n'apparaissent. Cependant, on peut les visualiser en zoomant les fichiers PCB sur ALTIUM

Annexe 4: Test d'allumage de la Led

