

TISSOT Elise

TIRABY Céline



Rapport intermédiaire de Projet de Fin d'études

Thermostat connecté et intelligent

IMA5

2015-2016

Table des matières

- Introduction2
- I Cahier des charges3
 - A) Objectif.....3
 - a. Le thermostat d’ambiance et étude3
 - b. Fonctionnalités du thermostat4
 - c. Fonctionnalités “avancées” du thermostat à implémenter4
 - B) Les différentes étapes du projet.....5
 - C) Schéma général du projet6
- II Réalisation.....6
 - A) Partie Électronique6
 - a. Recherches bibliographiques afin de réaliser les fonctions souhaitées6
 - b. Choix des composants8
 - c. Réalisation des schémas Altium et routage des cartes9
 - B) Partie Informatique12
 - a. Présentation des Panstamps13
 - b. Première communication13
- III Planning post pré-soutenance et avancement du projet16
 - A) Avancement du projet.....16
 - B) Planning post-présoutenance.....16
- Conclusion.....18
- Annexe : consommation du circuit carte 119

Introduction

Les **objets connectés** prenant de plus en plus de place dans la vie de tous les jours, notre projet de fin d'études nous permet de rester dans l'ère du temps. En effet, son but est de concevoir et de réaliser un **thermostat connecté intelligent**.

Ce thermostat devra s'adapter à la tendance mondiale qui est d'**automatiser** un maximum d'objets au quotidien ainsi que de réduire les coûts énergétiques liés en partie à la **consommation** des différents composants des cartes électroniques.

Ce projet nous permettra de **mettre en pratique des connaissances** acquises lors des cours que nous avons suivis durant les trois dernières années, mais est également pour nous une bonne occasion de développer nos compétences et nos connaissances.

Il se composera en **deux parties principales** qui consistent respectivement en la conception de deux **cartes électroniques** et **la réalisation du programme** permettant la communication entre ces dernières.

De plus, nous effectuerons une interface web, ainsi que la conception d'un boîtier à l'aide d'une découpeuse laser qui contiendra l'une des deux cartes électroniques.

I Cahier des charges

A) Objectif

Comme énoncé précédemment, l'objectif général de notre projet est de réaliser un **thermostat d'ambiance connecté** possédant différentes fonctionnalités.

a. Le thermostat d'ambiance et étude

Un thermostat d'ambiance permet de **réguler la température** ambiante d'une pièce. Il peut être incorporé dans l'appareil de chauffe comme par exemple une chaudière, ou déporté dans la pièce et être installé en position murale. La plupart du temps de conception numérique, le thermostat d'ambiance peut également être équipé de **fonctions de programmation** mode occupation ou confort, mode inoccupation ou mode hors-jeu.



Cependant, nous pouvons nous demander quel est l'intérêt de concevoir un thermostat connecté. Après avoir fait quelques recherches, nous avons trouvé que 85% des personnes interrogées pour un sondage (sur un échantillon de 1148 personnes) s'intéressaient aux objets connectés. De plus, **76% les trouvent utiles**, ce qui est un bon point pour la conception d'un tel projet.

Lorsque la question de l'objet connecté qui semble le plus utile dans une maison est posée, 54,4% citent le thermostat connecté qui se place donc derrière les alarmes (59,7%) et les caméras de surveillance (56,4%). Plus **d'une personne sur deux** est donc susceptible de s'intéresser à un thermostat connecté puisqu'il est jugé utile.

De plus, lorsqu'on demande quel objet connecté est en possession des personnes interrogées pour le sondage, seulement **7,4%** déclarent avoir un thermostat connecté à la maison. Nous pouvons donc penser qu'il existe un **marché important** pour les thermostats connectés.

Enfin pour terminer cette étude, nous pouvons noter que **79%** des Français cherchent à **réduire leur consommation d'énergie**. Il est donc intéressant d'inclure dans les fonctionnalités du thermostat un moyen permettant la réduction de la consommation d'énergie. En effet, cela pourrait être un facteur qui peut pousser les clients à en acquérir un.

b. Fonctionnalités du thermostat

Le thermostat connecté devra donc posséder plusieurs fonctionnalités.

Dans un premier temps, il devra avoir les mêmes fonctionnalités qu'un **thermostat classique**. Il sera donc capable de **capter la température** de la pièce ainsi que de la modifier selon les souhaits de l'utilisateur.

De plus, le thermostat connecté sera contrôlable à partir d'une **interface web** afin de faciliter l'utilisation du thermostat pour le client.

Il devra permettre, en outre, d'effectuer des **mesures de la consommation énergétique** en Wh et en Euro. En effet, le prototype devra avoir une consommation minimale dans le but d'augmenter sa longévité et d'être plus attractif pour l'acheteur potentiel. Un thermostat où la nécessité de changer les piles est moins récurrente que la normale attirera plus facilement qu'un thermostat qui nécessite un changement de piles plus fréquent.

Ensuite, le thermostat devra être capable **de mesurer les importants écarts de température** dus par exemple à l'ouverture d'une fenêtre. De plus, il pourra également apprendre du temps de chauffe nécessaire pour atteindre une température donnée en consigne, grâce à sa mémoire. En effet, le but du projet est d'obtenir un thermostat intelligent.

Enfin, le thermostat sera installé dans un **boîtier** modélisé fabriqué au Fablab de l'école afin de, cette fois encore, améliorer la prise en main du thermostat et donc le confort de l'utilisateur.

Au début du projet, nous avons également l'ambition **d'auto-alimenter** le thermostat à l'aide d'une énergie lumineuse par exemple. En effet, cela aurait permis de réduire fortement la consommation d'énergie et aurait été une fonctionnalité intéressante du thermostat. Malheureusement, nous avons dû **abandonner** cette idée car il s'est avéré que l'auto-alimentation du thermostat aurait pu être un sujet de projet à lui seul et nous n'aurions donc pas suffisamment de temps pour réaliser cette fonctionnalité supplémentaire.

c. Fonctionnalités "avancées" du thermostat à implémenter

En plus des fonctionnalités énoncées ci-dessus que le thermostat doit être capable d'effectuer, nous allons par la suite rajouter d'autres fonctions qui valoriseront le projet.

Par exemple, une fonctionnalité essentielle serait **l'augmentation ou la diminution** d'un certain nombre de degrés de la température par rapport à celle ressentie par le capteur de température.

De plus, il pourrait être intéressant de permettre **l'affichage de la température actuelle** de la pièce sur l'interface graphique ainsi qu'un suivi de la température à l'aide de courbes qui retracent son évolution dans le temps.

Une autre fonctionnalité intéressante serait de pouvoir commander **l'arrêt ou la mise en marche** de la chaudière à partir de l'interface graphique.

Enfin, d'autres fonctionnalités pourraient être implantées si la possibilité se présente comme l'envoi d'un **bilan énergétique** permettant à l'utilisateur d'**optimiser** son chauffage le mois suivant ou une auto-adaptation du thermostat qui détermine le temps nécessaire pour atteindre la température souhaitée. Nous pourrions également inclure des modes de fonctionnement différents selon la période à laquelle l'on se situe (**vacances, période hivernale, etc**).

B) *Les différentes étapes du projet*

Pour réaliser notre thermostat, nous avons découpé le travail en plusieurs parties.

La première consiste à la **construction des cartes électroniques**, qui sont au nombre de deux. En effet, la première carte correspond au **thermostat en lui-même**, c'est-à-dire qu'il capte la température de la pièce et envoie les instructions permettant de commander la chaudière à la seconde carte. Cette dernière quant à elle, réalise la **commande la chaudière**.

La particularité de la première carte est qu'elle est portable, l'utilisateur peut donc placer le thermostat où il le souhaite dans la pièce de son choix, tandis que la seconde est directement reliée à la chaudière. Nous avons choisi de réaliser une carte au lieu d'une seule pour justement permettre à l'utilisateur de placer le thermostat où il le souhaite dans la maison plutôt que de le laisser dans la pièce où se situe la chaudière.

Afin de réaliser les cartes électroniques, il nous faut tout d'abord sélectionner les **composants adéquats** pour la construction de notre thermostat. Une autre particularité de notre projet réside dans l'utilisation de composants **CMS** (en anglais : SMD qui signifie Surface Mounted Device ou Composant Monté en Surface). Le seul composant imposé est l'utilisation du **panStamp NRG 2** qui est l'élément contrôlable de nos cartes. Il y a un panStamp sur les deux cartes électroniques, c'est ce qui permet la communication entre les deux cartes.

Après avoir déterminé les composants nécessaires pour la réalisation du thermostat, il nous faut alors les **commander** en faisant attention aux performances des éléments mais surtout aux correspondances entre chaque composant.

Les composants étant commandés, il nous faut par la suite **concevoir les schémas électriques** à l'aide des datasheets des composants, puis les **PCB** des deux cartes ainsi que le routage.

Enfin, après que les cartes soient tirées, vient l'étape de **soudage des composants** puis celle des tests afin de vérifier la **validité** de nos cartes.

La seconde étape de notre projet **consiste à l'établissement d'une communication** entre les différentes "parties" du projet. Pour ce faire, nous utilisons les panStamps en RF (radio fréquence).

Outre les deux panStamps situés sur les cartes électroniques, nous utilisons également un panStamp relié à une RaspberryPi permettant la communication avec l'interface graphique.

Il nous faut alors configurer les différents panStamps afin de permettre l'envoi et la réception des données, et donc permettre l'échange des informations.

La troisième étape réalise **l'implémentation des différentes fonctions** sur chaque panStamp. Par exemple, il peut s'agir de récupérer la température de la pièce via le capteur de température ou alors d'effectuer la commande de la chaudière.

Nous allons également effectuer des fonctions plus avancées comme des calculs de consommation (si possible) ainsi que des fonctions permettant la détection d'un changement brutal de température.

La quatrième étape consiste en **l'élaboration d'une interface web graphique** permettant à l'utilisateur de contrôler à distance le thermostat connecté.

Cette interface doit être simple d'utilisation, intuitive et réalisée avec un design attractif. Elle sera réalisée à l'aide des langages PHP, HTML, CSS et JavaScript.

Enfin, la dernière étape du projet est la fabrication **d'un boîtier pour le thermostat** (carte portable).

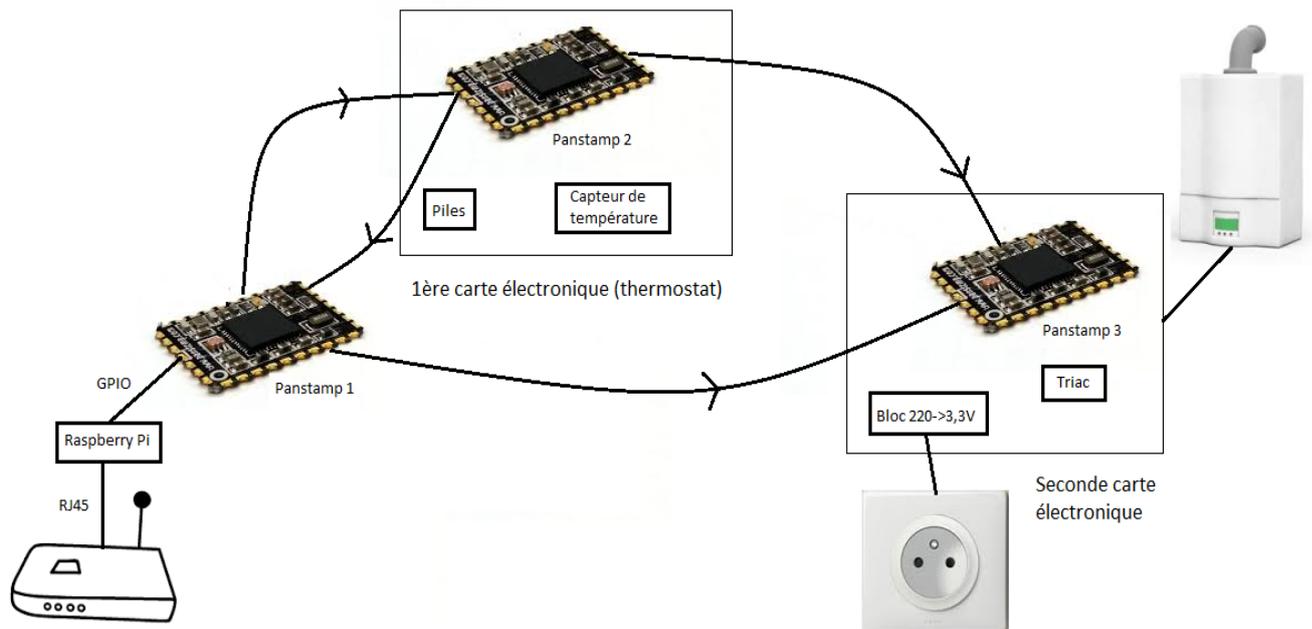
Ce boîtier sera réalisé au Fablab à l'aide d'une découpeuse laser. Le logiciel permettant de modéliser le prototype est Solid Works.

Ce dernier aura pour but de rendre le thermostat plus esthétique et devra ne pas être trop imposant afin de permettre une meilleure prise en main du thermostat pour l'utilisateur.

Une étape supplémentaire consistant à la réalisation d'un prototype permettant de commander un radiateur à la place d'une chaudière peut être ajoutée au projet par la suite.

C) Schéma général du projet

Afin d'avoir une vue plus générale de notre projet, voici un schéma récapitulant les différents éléments ainsi que les communications entre chacun d'entre eux :



II Réalisation

A) Partie Électronique

La partie électronique consiste, comme cela est présenté dans le cahier des charges, en la réalisation de **deux cartes électroniques**, ayant respectivement pour fonction de capter la température et d'actionner la chaudière.

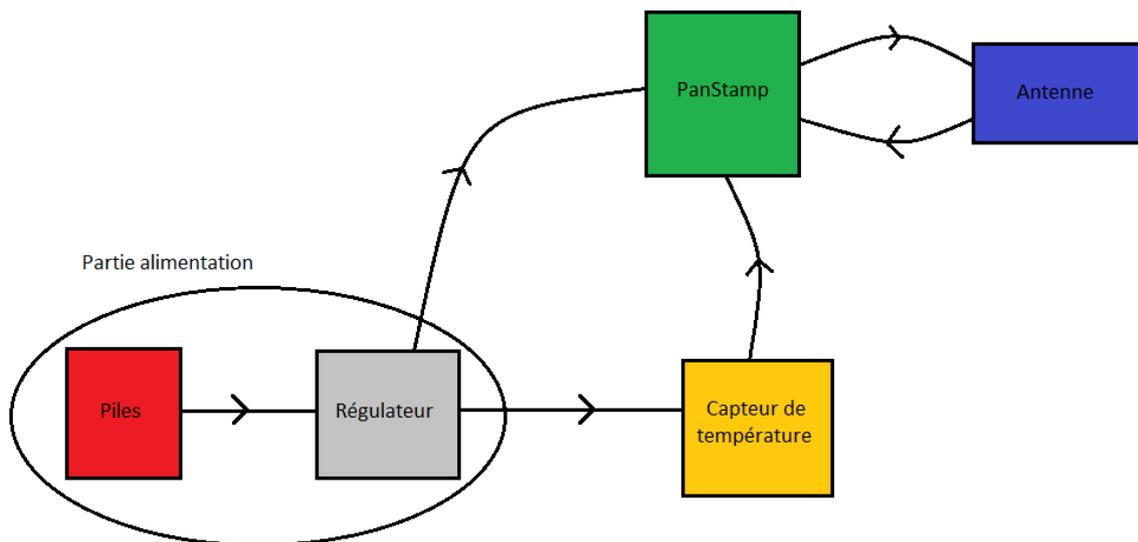
a. Recherches bibliographiques afin de réaliser les fonctions souhaitées

Afin de réaliser la partie mobile du thermostat, il était évident que nous devions nous munir d'un **capteur de température**, d'un **panStamp** (fonctionnant de 2VDC to 3.6VDC) et de piles pour l'alimenter (2 piles 1.2V). Néanmoins, afin d'optimiser la consommation du panStamp, et dans le cas où les piles ne fourniraient plus une tension suffisante (lorsqu'elles commencent à se décharger par exemple) il était nécessaire d'implanter également un

régulateur. Ce composant permet en effet de réguler la tension de sortie à la valeur souhaitée. Nous avons ici choisi 2.4V, la tension minimum pour utiliser la fonction RF du panStamp étant de 2.2V. Nous pouvons ainsi réduire la consommation du circuit en ne l'alimentant pas en 3.3V par exemple.

Pour synthétiser la carte 1 possède :

- Un panStamp NRG 2
- Une antenne
- Un boîtier de deux piles 1.2V rechargeables
- Un régulateur de tension
- Un capteur de température

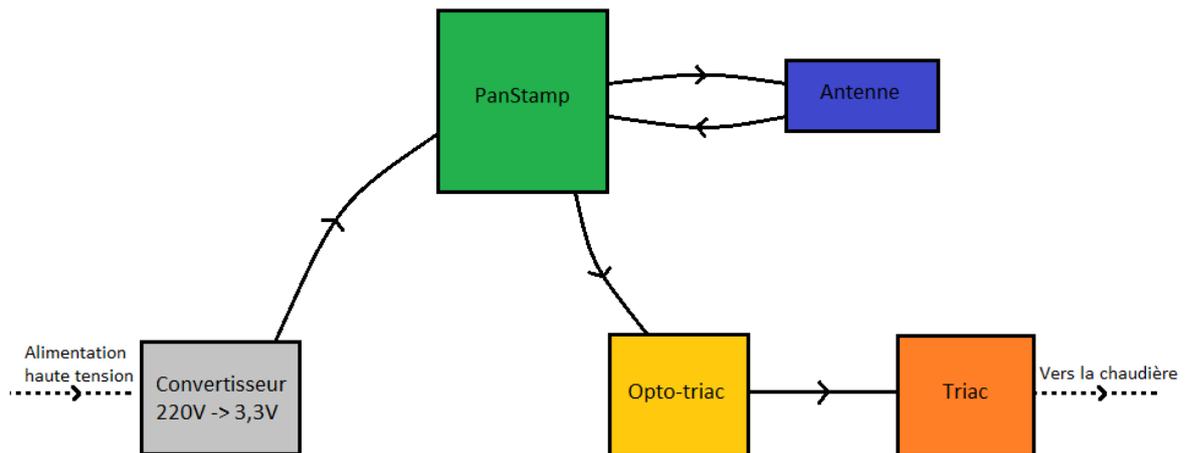


Concernant la carte directement implantée sur la chaudière, nos recherches bibliographiques nous ont tout d'abord amenées à l'utilisation d'un **relais**, afin de piloter la chaudière en ouvrant et en fermant le circuit électrique. Malgré tout, la plupart des chaudières délivrent une tension et un courant élevés (220 V AC et 16A). Nous devons donc trouver un relais 3,3V/220V-16A, mais ce type de composant est cher et prend beaucoup de place. La solution était donc de prendre un triac et un opto-coupleur, beaucoup moins contraignants. Suite à une discussion avec M.Benabou et M.Flamen, nous avons opté pour un opto-triac suivi d'un triac, configuration qui réduit fortement les parasites.

La carte ne nécessitant pas d'être déplacée, nous avons décidé de l'alimenter grâce au secteur. L'emploi d'un **bloc convertisseur 220V/3,3V** devenait alors indispensable afin de ne pas endommager le panStamp et les autres composants.

Pour synthétiser la carte 2 possède :

- Un bloc convertisseur 220V/3.3V
- Un panStamp NRG 2
- Une antenne
- Un opto-triac
- Un triac



b. Choix des composants

Nous avons décidé d'utiliser des composants **CMS** pour plusieurs raisons :

- Leurs **avantages mécaniques** (Réduction du volume des composants, réduction de la surface du circuit imprimé, gain de poids, etc.)
- Leurs **avantages électriques** (Réduction des parasites, plus grande fiabilité, etc)

Le seul **inconvenient** est la difficulté à souder les petits composants.

Nous avons utilisé des critères comme le prix, la disponibilité et bien sur la fonctionnalité pour établir notre liste de composants principaux (Régulateur, capteur de température, triac..). Nous avons ensuite étudié les datasheets pour acheter les composants annexes (résistances, capacités etc).

Pour le triac et l'opto-triac, nous avons du effectuer un travail de recherche un peu plus important étant donné que la correspondance entre ces deux composants devait être parfaitement réalisée. Nous avons donc concentré nos recherches dans un premier temps sur l'opto-triac afin de trouver un composant permettant le passage d'une tension supérieure ou égale à 220V. Ensuite, en examinant la datasheet, nous avons trouvé un schéma électrique correspondant à une application de l'opto-triac avec notamment la référence du triac utilisé. Ce dernier étant à présent obsolète, nous avons alors recherché un triac avec des caractéristiques similaires et surtout qui supporte un courant de minimum 16A.

Nous sommes ainsi parvenues à la liste de composants suivante :

Boîtier à piles de référence fournisseur (Mouser) **534-2462**
Piles AA de référence fournisseur (Mouser) **547-NH1250AA**
Capteur de T°C de référence fournisseur (Mouser) **595-TMP112AIDRLT**
Bloc 220-3.3V de référence fournisseur (Mouser) **709-IRM05-3.3**
Fusible de référence fournisseur (Mouser) **576-083506.3MXEP**
Porte fusible de référence fournisseur (Mouser) **693-0031.8201**
Antenne plane de référence fournisseur (Mouser) **609-0868AT43A0020E**
Condensateur 10u de référence fournisseur (Mouser) **80-C0805L106K8P**
Condensateur 0.01u de référence fournisseur (Mouser) **80-C0805C103M5RAUTO**
Bobine 4,7u de référence fournisseur (Mouser) **710-744031004**
Régulateur Boost TEXAS de référence fournisseur (Mouser) **595-TPS61070DDCR**
LED de référence fournisseur (Mouser) **720-LHR974-LP-1**
Résistance 348kOhm de référence fournisseur (Mouser) **667-ERJ-6ENF3483V**
Résistance 1MOhm de référence fournisseur (Mouser) **71-CRCW08051M10FKEA**
Résistance 5kOhm de référence fournisseur (Mouser) **71-CRCW0805-5K**
Opto-triac de référence fournisseur (Mouser) **512-MOC3062SR2VM**
Triac de référence fournisseur (Mouser) **576-Q6016NH4RP**
Résistance 360Ohm de référence fournisseur (Farnell) **2331668**
Résistance 39Ohm de référence fournisseur (Mouser) **279-SMW539JT**
Résistance 150Ohm de référence fournisseur (Mouser) **279-SMW5150RJT**
Condensateur 0,01uF de référence fournisseur (Mouser) **603-CC126KKX7RZBB103**
Inductance 12nH de référence fournisseur (Mouser) **609-L-07C12NJV6T**
Inductance 1,8nH de référence fournisseur (Mouser) **609-L-07C1N8SV6T**

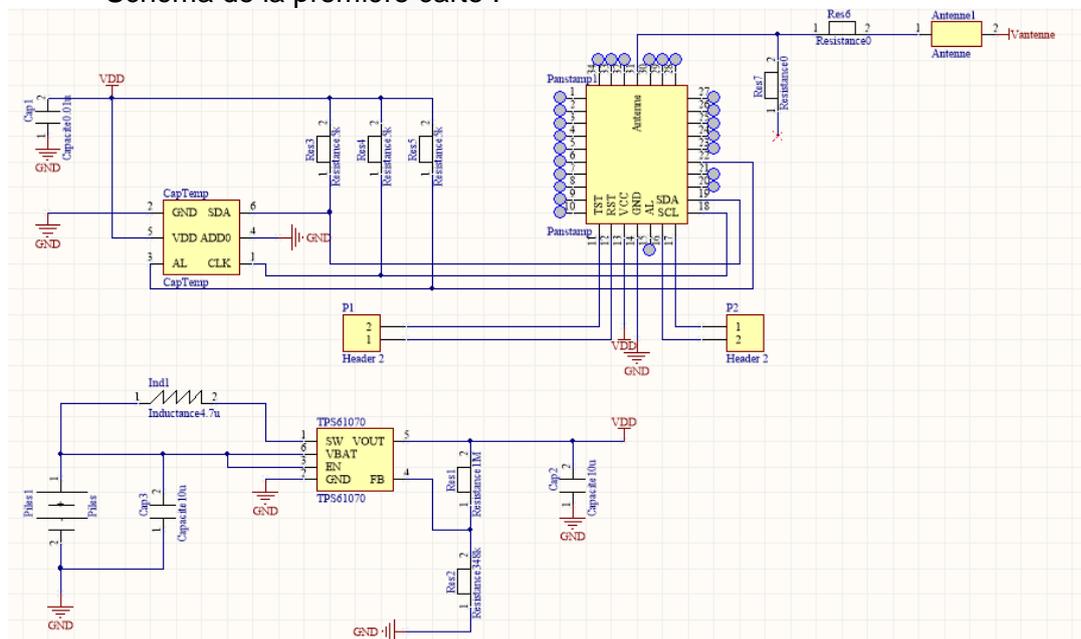
c. Réalisation des schémas Altium et routage des cartes

La 1ère étape fut la **réalisation des schématiques Altium** sur la base des schémas présentés en section en II.A.a.

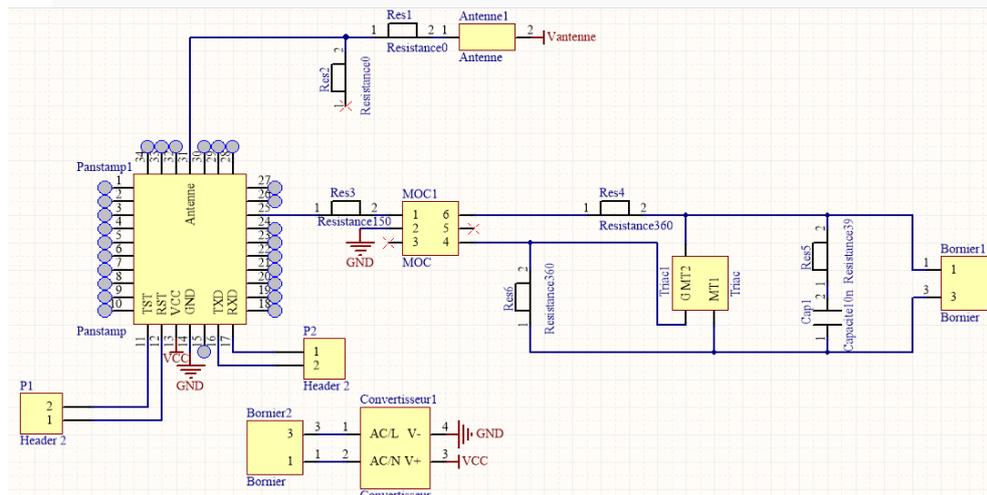
Pour cela nous avons attentivement étudié les datasheets et les circuits basiques de fonctionnement préconisés par les constructeurs.

Nous sommes ainsi parvenues aux schémas suivants :

- Schéma de la première carte :



- Schéma de la seconde carte :



Après avoir réalisé les schémas électriques, il nous restait donc à **créer les PCB** correspondant à chacune des cartes afin de “construire” ces dernières.

Dans un premier temps, nous avons créé les footprints des différents composants en utilisant notamment l’outil de création automatique d’Altium et en se référant encore une fois aux datasheets des composants.

La bibliothèque des composants de nos cartes associés à leur empreinte respective créée, nous avons ensuite généré les PCB.

Il restait alors à placer les composants de façon à regrouper les éléments associés ensemble et en essayant de réduire la taille de la carte le plus possible.

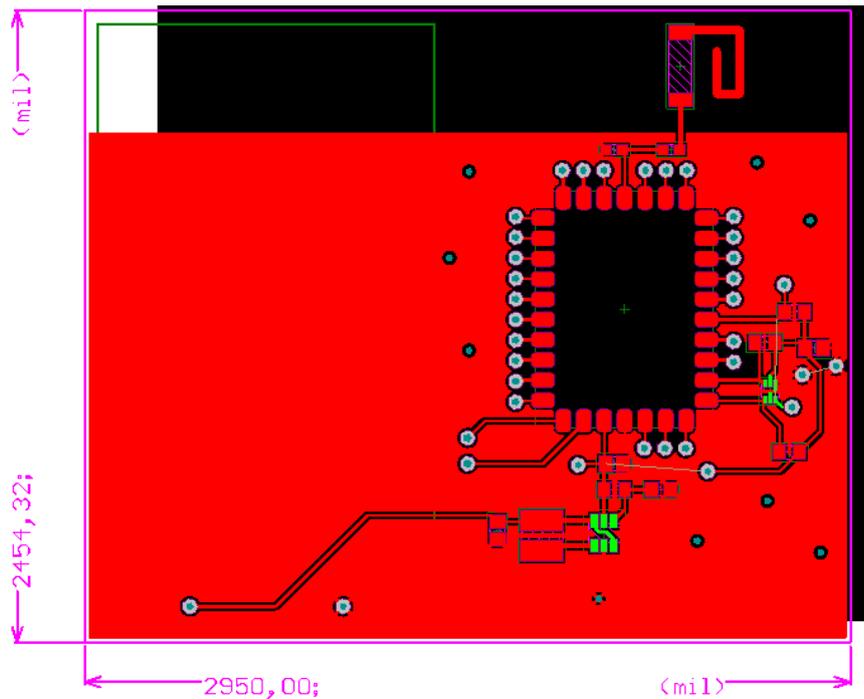
Pour la carte 1

Nous avons regroupé les composants correspondant à la partie “alimentation”, c’est-à-dire le boîtier de piles, le régulateur et les composants tels que les résistances, bobines et condensateurs associés au régulateur. Nous avons fait de même avec la partie permettant de capter la température ainsi que l’antenne.

Nous avons également ajouté un **plan de masse** ainsi que des trous situés sur l’ensemble du PCB afin de maintenir un potentiel de masse égal à chaque endroit de la carte. Cependant, le plan de masse ne doit pas être sous l’antenne sous peine d’empêcher son fonctionnement.

De plus, certaines pistes ne pouvaient être placées sur la face “top” du PCB donc pour contourner ce problème et éviter de graver une carte en double face, la solution est d’utiliser des **straps** passant à l’arrière de la carte électronique.

Voici le PCB de la carte 1 :



Pour la carte 2

La deuxième carte étant moins complexe que la première (moins de composants) et ayant déjà effectué le PCB de la carte 1 auparavant, la réalisation de ce PCB fut **plus rapide** que le précédent.

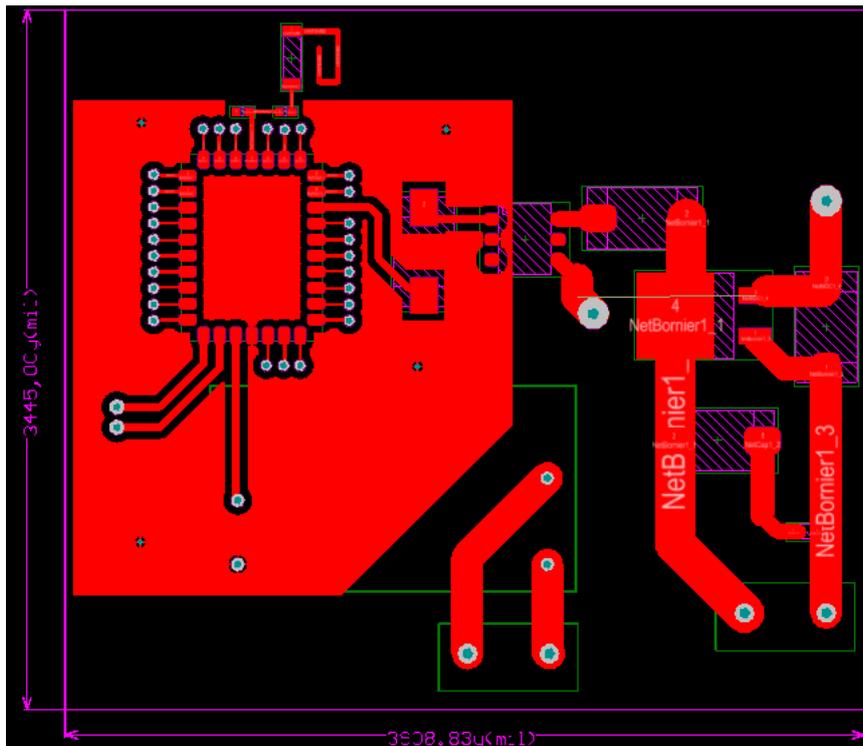
Comme pour la carte 1, nous avons séparé les différentes parties de la carte qui correspondent ici aux parties commande et puissance.

La difficulté de cette carte résidait dans la précaution quant aux **largeurs des pistes** et à l'**espacement** entre ces dernières. En effet, la tension pouvant être de 220V maximum, il nous fallait espacer les pistes de 3mm minimum. De plus, la carte doit pouvoir supporter un courant maximal de 16A. Pour cela, les pistes devaient être élargies d'au moins **4mm** afin de répondre à cette contrainte.

Nous avons également ajouté un plan de masse sur ce PCB, qui cette fois-ci ne doit ni interférer avec l'antenne, ni avec la partie puissance.

De plus, nous devons penser par la suite à un moyen de dissiper la **chaleur générée** par le triac. En effet, les premiers tests de cette carte ne seront pas effectués à haute tension donc cela n'est pas gênant pour le moment. Pour résoudre ce problème, nous avons le choix entre deux solutions : soit de mettre un **dissipateur** sur le triac, soit d'insérer un plan relié au triac qui dissiperait suffisamment la chaleur générée par ce dernier. Dans ce dernier cas de figure, le triac serait alors placé sur la face bottom du PCB.

PCB de la carte 2 :



Au moment où le rapport est rédigé, la **carte 1 a été gravée, soudée** et est en cours de phase de tests. La partie alimentation fonctionne (la sortie du régulateur alimente correctement le panStamp en 2,4V). Cependant, un court-circuit persiste, probablement dans la partie capteur de température. Des tests supplémentaires seront réalisés après la soutenance afin de déterminer l'origine du problème.

Si nous étions amenées à réaliser de nouveau cette première carte (pour cause de non fonctionnement du premier prototype), nous éviterons de refaire certaines erreurs. Un PCB a d'ores et déjà été refait en plaçant le boîtier de piles plus intelligemment afin d'avoir un gain de place, les composants ont donc été espacés les uns des autres et les pistes agrandies. En effet, nous avons constaté qu'à force de souder et de dessouder les composants, les pistes se décollaient de la carte. De plus, l'espacement entre le plan de masse et les pistes a été agrandi pour limiter au maximum les risques de court-circuit. Enfin, les composants ne doivent pas être **chauffés trop longtemps**. Dans le cas contraire, ces derniers ne sont plus fonctionnels.

Pour ce qui est de la seconde carte, elle a été lancée en production suite aux erreurs commises sur la première. Nous avons donc tenté de ne pas reproduire ces dernières en espaçant les pistes du plan de masse par exemple. Néanmoins, les composants devant résister à une tension maximale de 220V, ils sont donc plus imposants que sur la première carte et devraient alors être plus faciles à souder.

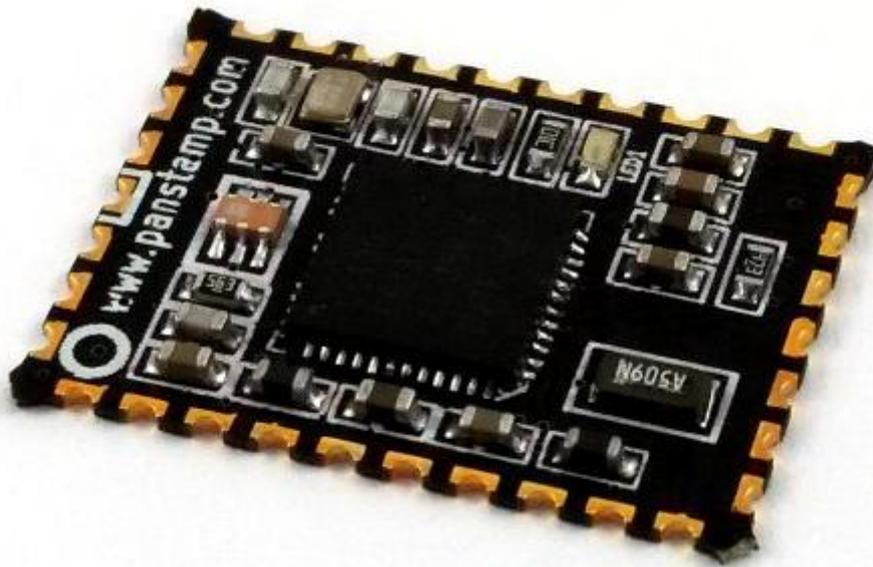
B) Partie Informatique

La partie informatique représente une partie importante de notre projet. En effet, la communication entre les différents panStamps (comme vu ci-dessus) est **indispensable** pour réaliser un thermostat fonctionnel. Nous avons donc choisi d'effectuer les échanges en **RF** (Radio Fréquence) en utilisant des panStamps.

a. Présentation des Panstamps

Les panStamps sont des **modules autonomes** à basse consommation conçus pour des applications de télémétrie et de contrôle. Ils utilisent une plage de fréquence de communication comprise entre 868 - 915 MHz. Nous avons choisi d'utiliser la fréquence **868 MHz**, la plus part des autres étant plus réglementées en France (certaines nécessitent notamment des licences). De part leur ergonomie, il est facile d'implanter un panStamp sur une carte électronique, ce qui justifie encore ici le choix de cet équipement pour notre projet. Leur **durée de vie** relativement **longue** est également un argument en faveur d'une conception grâce aux panStamps. Enfin, le fait qu'ils utilisent l'environnement **Arduino** permet de les prendre en main plus rapidement et ainsi d'optimiser le temps de réalisation du projet.

Nous utiliserons ainsi les panStamps **NRG 2**, dernière version de panStamps mise sur le marché, afin de pouvoir bénéficier des améliorations les plus récentes en termes de consommation, de temps de traitement etc.



b. Première communication

Comme précisé dans le cahier des charges, le but est de faire communiquer **trois panStamps** (le panStamp relié à la raspberry et à l'interface graphique (1), celui présent sur le thermostat déplaçable (2) et celui directement implanté sur la chaudière (3). Une communication entre les panStamps 1 et 3 n'est toutefois pas exclue).

Nous avons reçu de la part de notre tuteur industriel deux panStamps **AVR** afin de pouvoir commencer à effectuer les premiers tests de communication. Nous disposons alors d'un panStick (sous forme de clé usb) et d'un **AVR** alimenté grâce à une RaspberryPi. Nous ne pouvons pas encore utiliser les panStamp NRG pour deux raisons :

- Le délai de livraison (ils n'étaient pas encore disponibles)

- Leur implantation directe sur les cartes, qui implique le fait que ces dernières soient opérationnelles

Notre tuteur industriel nous a fourni une base de code (en langage C++/Arduino) pour la communication entre panStamps. Nous avons également eu accès aux différents sketches fournis par la marque panStamp et notamment à **“Modem”** qui permet également des échanges entre deux modules.

Tout d’abord, nous devons **implanter** les différents codes sur nos panStamps. Pour cela, nous avons utilisé une méthode de compilation ne nécessitant pas de Makefile et nous permettant de conserver notre code en langage Arduino et C++. En effet, même si l’optimisation reste moins bonne qu’en utilisant du C, nous avons choisi de conserver cette méthode dans un premier temps, quitte à l’améliorer lorsque que le projet sera plus avancé.

Le but était de simplifier le code fourni par notre tuteur, en implémentant une **lecture clavier** des différentes commandes sur le port série. Cette lecture était en partie disponible sur le code “modem” fourni par le constructeur.

Le module de communication RF nécessite l’envoi de certaines données dans les paquets dont la forme peut être caractérisée de la façon suivante :

CC11XX packet format



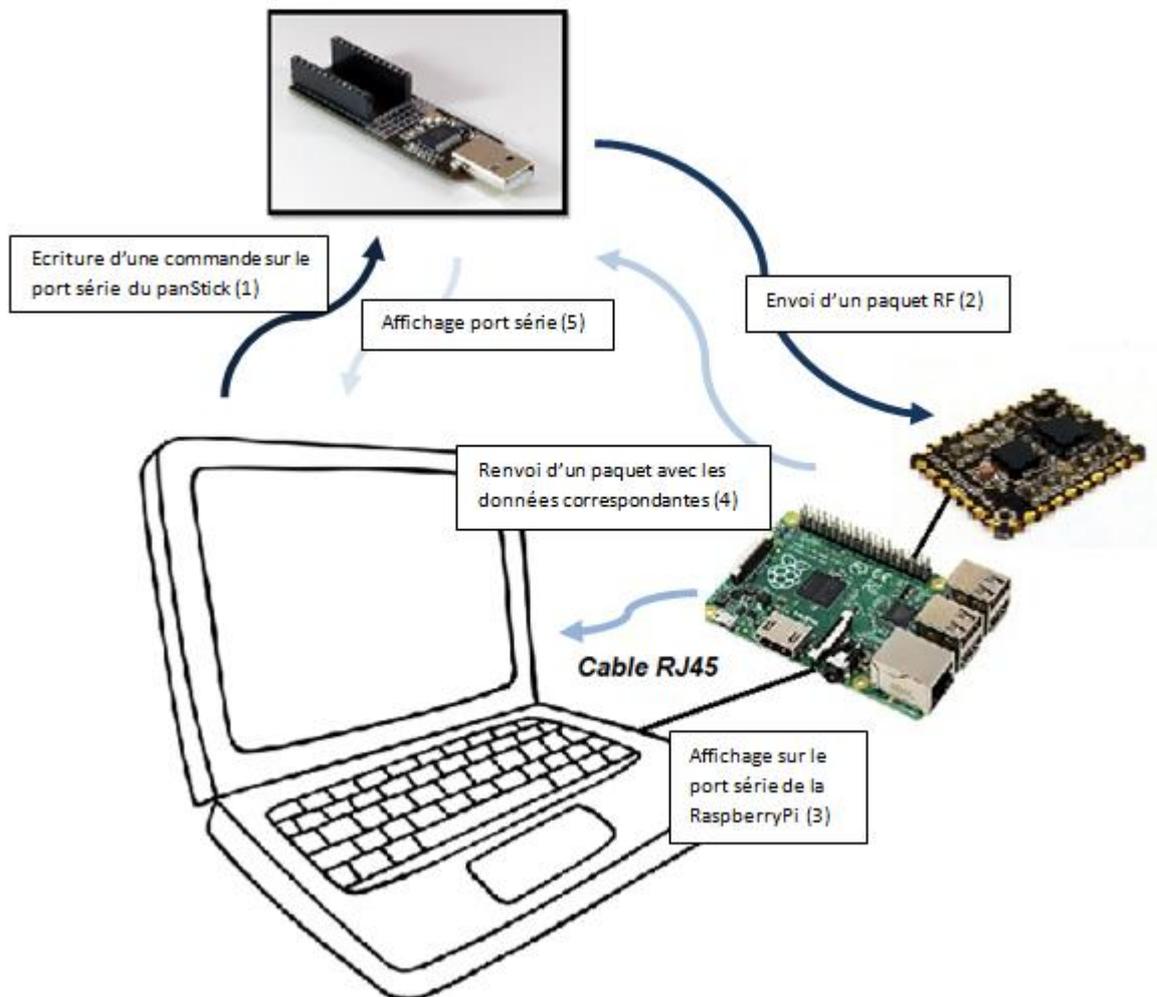
- 4 octets réservés aux **mots de synchronisation**, qui permettent de percevoir plus facilement le paquet dans le bruit
- 1 octet pour la **longueur du paquet**
- 1 octet pour le **CRC (Cyclic Redundancy Check)** calculé automatiquement

Nous avons également choisi de décomposer les données en sous parties afin de réaliser des paquets plus clairs pour une future lecture. Ainsi l’espace des données est divisé de la façon suivante :

- 1 octet pour l’**adresse de destination**
- 1 octet pour l’**adresse source**
- 1 octet pour le **numéro de commande**
- 1 octet pour le **nombre de paquets**
- 1 octet pour le **numéro du paquet**
- Les octets restants sont dédiés aux **données**

Le code utilise les **diverses fonctions** mise à disposition dans les bibliothèques de panStamp afin d’effectuer une première communication (`panstamp.attachInterrupt(rfPacketReceived)`, `panstamp.radio.setCCregs()`...).

Voici le schéma de la communication réalisée :



Une fois le programme de première communication fonctionnel, il reste à le complexifier en y ajoutant les différentes fonctions présentées dans le cahier des charges.

L'un des enjeux principaux du projet est la **réduction de la consommation** du circuit global, afin que l'utilisateur change les piles le moins souvent possible. C'est dans cette optique que nous avons choisi de mettre le panStamp (2) en mode sommeil la plupart du temps.

Nous avons à l'heure actuelle :

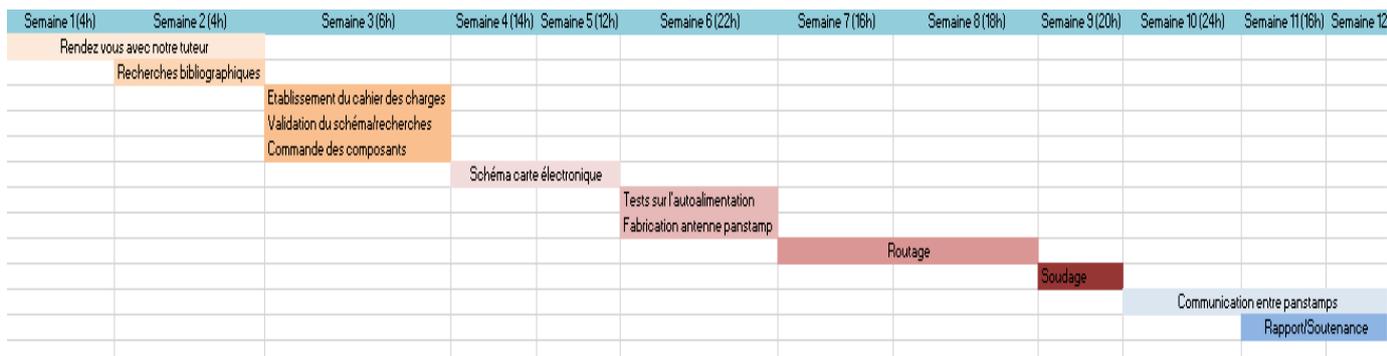
- Réussi à implanter le **TIMER** dans le programme, afin de pouvoir compter le nombre de secondes où le panStamp (2) doit dormir. A noter que les codes sont légèrement différents entre un panStamp AVR et un panStamp NRG.
- Effectué un premier calcul de **consommation énergétique** ayant pour but de déterminer le temps de sommeil de notre panStamp (2) (voir annexe)

Malgré tout, le temps en mode "sleep" du module (2) est assez flexible. En effet le panStamp stocke une température, qui sera mise à jour uniquement lors du réveil. Sachant qu'il est difficile de percevoir une variation de température dans un laps de temps court (< 5 min), l'utilisateur ne fera pas la différence entre la valeur réelle de température et la valeur stockée par le panStamp à son dernier réveil. Il en est de même pour l'activation ou non de la chaudière par exemple.

III Planning post pré-soutenance et avancement du projet

A) Avancement du projet

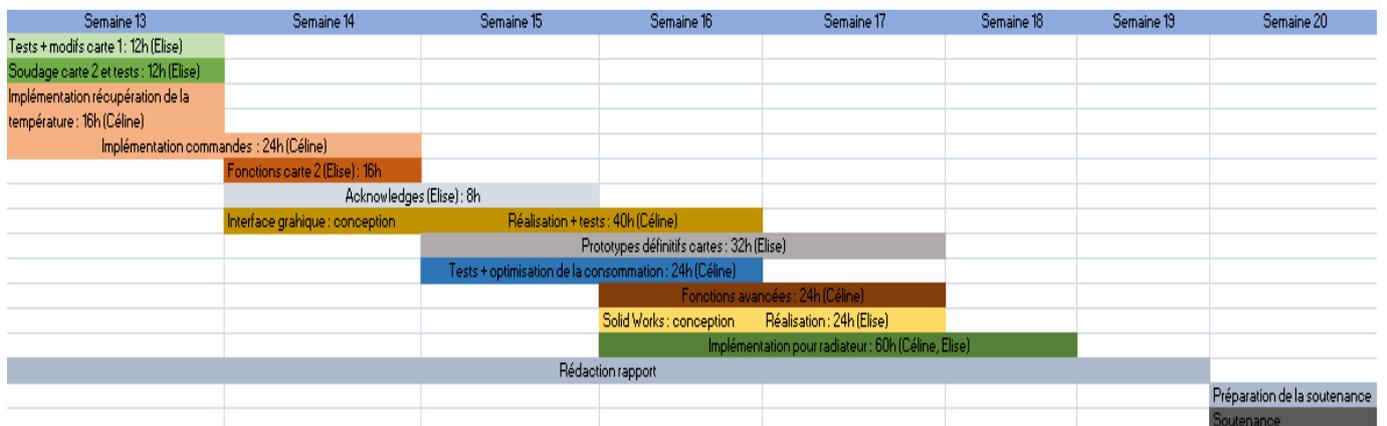
Actuellement, on peut considérer que l'état d'avancement du projet est d'**environ 40%**. On peut remarquer que le planning prévisionnel établi en début de projet (voir ci dessous) à été en partie respecté.



Nous avons malgré tout rencontré des **difficultés** au niveau de la **réalisation des cartes** et de leurs tests, notamment à cause des soudures difficiles à réaliser en **CMS** et le **manque d'expérience** pour router avec le logiciel Altium. La réalisation et les tests des prochaines cartes seront cependant plus rapides maintenant que nous avons acquis l'expérience nécessaire pour réaliser ce type de projet.

B) Planning post-présoutenance

Voici le planning post-présoutenance de notre projet :



Comme on peut le voir sur le graphique, les tâches seront **réparties équitablement** entre nous. Elise qui s'occupait jusqu'alors de la partie Électronique, participera également à certaines implémentations de la partie informatique, même si cela restera d'une manière succincte. Cette répartition nous permettra d'avancer plus efficacement sur les deux phases

du projet. Celles-ci étant liées, il sera assez gênant de prendre du retard, notamment sur la partie électronique.

On peut voir que le **projet** a une grande possibilité d'être **fonctionnel** avant la soutenance finale, si nous ne rencontrons aucun problème majeur.

Il restera, pour la **partie électronique** de :

- *Tester et modifier la carte 1*
- *Souder et tester la carte 2*
- *Réaliser les prototypes définitifs des cartes*
- *Réaliser les tests de consommation*
- *Faire si possible l'implémentation pour un radiateur*

Il restera, pour la **partie informatique** de :

- *Implanter la récupération de température*
- *Implanter les fonctions définies pour les cartes 1 & 2*
- *Réaliser l'interface graphique du thermostat*
- *Faire des fonctions avancées si le temps le permet*

En outre, il faudra réaliser le **boîtier** du thermostat et rédiger le **rapport**.

Conclusion

L'objectif de notre projet est de réaliser un **thermostat d'ambiance connecté** permettant de réaliser plusieurs fonctionnalités.

Le projet se découpe en **deux principales parties**. La première est la partie électronique comprenant essentiellement la **conception des deux cartes**, le routage et le soudage des composants. A cela s'ajoute une partie traitant la consommation des cartes afin de limiter cette dernière au maximum. La seconde est la **partie informatique** comprenant la communication entre les trois panStamps ainsi que l'implémentation d'une interface graphique et des fonctions nécessaires au bon fonctionnement du projet.

A mi-parcours, **40% du travail** a d'ores et déjà été effectué. Cependant, nous n'étions pas à temps plein sur la période allant des mois de septembre à décembre. De plus, nous avons rencontré des problèmes qui ont retardé la réalisation du projet, notamment sur la conception des cartes électroniques et surtout sur la soudure des différents composants sur ces dernières.

Néanmoins, les problèmes rencontrés nous ont permis d'apprendre de nos erreurs et nous serons plus opérationnelles après la soutenance. En effet, l'expérience acquise lors de ces mois de travail ainsi que le temps plus important alloué au projet nous permettra d'avancer plus rapidement lors des deux derniers mois de notre projet de fin d'études.

Annexe : consommation du circuit carte

1

Afin de calculer la consommation du circuit, nous devons d'abord trouver le nombre de données échangées en kBps. Après avoir cherché dans des datasheets et sur le site de panStamp, nous avons trouvé que nos panStamps utilisaient un datarate de 38383,5 bps. Le minimum de bits échangés à chaque envoi de paquet étant de 104, nous obtenons facilement les temps de transmission/réception du panStamp :

$$104 \text{ } 38383,4 = 0.002708333 \text{ secondes}$$

Nous avons supposé que le panStamp effectuait 2 transmissions et une réception.

Consommation	Nombre de mA	Temps en sec	Consommation sur le temps
Réception	18	0,002708333	0,04875
Transmission	36	0,005416667	0,195
Traitement	0	0	0
Sleep	0,002	60	0,12
		Temps total	Consommation totale sur le temps
		60,008125	0,36375
		Consommation ramenée à l'instantanée	0,006061679
		Capacité des piles en mA/h	2500
		Consommation du circuit en heures	412426,9759

Nous trouvons un résultat de 412426.9759 heures, ce qui correspond à environ 47 années lorsque le panStamp est mis en sommeil pendant 60 secondes.

Pour être plus précis, nous pouvons également rajouter les consommations du régulateur et du capteur de température, qui sont respectivement de 1.9×10^{-5} A et de 1×10^{-5} A.

Nous voyons donc que les résultats théoriques semblent très corrects. Néanmoins des tests de consommation seront effectués lorsque les cartes seront opérationnelles, ainsi qu'une étude plus détaillée.