

PFE Cycle ingénieur

Rapport Final

Département Informatique, Microélectronique, Automatique
Polytech'Lille, Villeneuve d'Ascq

Capteurs Communicants Intelligents

Projet Artistique Wax Tailor

Elèves	Enseignants	Année Scolaire
Bastien CHALAUX Nicolas HUSSE	Alexandre BOE Thomas VANTROYS Nicolas DEFRANCE	2013 / 2014

Sommaire

Introduction	3
I Le Sujet.....	4
II Cahier des charges	5
Partie 1 : Interaction Bracelet / Porteur du Bracelet	5
Contraintes.....	5
Partie 2 : Interaction Wax Tailor / L'ensemble des bracelets de la salle	5
Contraintes.....	6
III Travail réalisé	7
III.1) Résumé du rapport précédents	7
III.1.a) Mis en place d'une solution technique et premiers résultats	7
III.1.b) Réalisation d'un prototype à base de MRF49XA	7
III.1.c) Conception du protocole.....	8
III.2) Le prototype.....	9
III.2.a) Le débogage du MRF49XA	9
III.2.b) Conception d'un nouveau prototype.....	10
III.3) Partie Logicielle	11
III.3.a) Le Protocole de Communication.....	11
III.3.b) Interprétation des commandes	14
III.3.c) Structure du Code	15
III.3.d) Résultats obtenus	16
IV Conclusion.....	19
Annexe	20
Annexe 1a : Schéma du prototype actuel.....	21
Annexe 1b : Schéma du nouveau prototype.....	22
Annexe 2 : Structure du Code (Fonctions)	23

Introduction

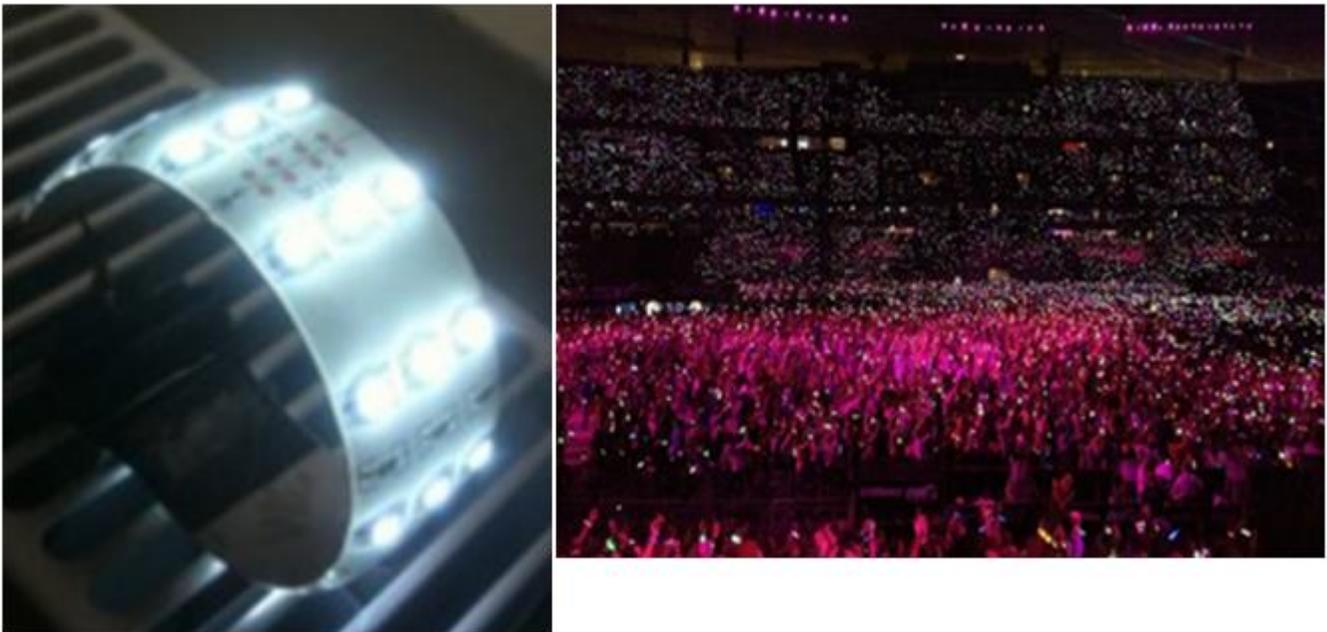
Dans le cadre de notre Projet de Fin d'Etudes (PFE) au sein de la formation Informatique Microélectronique Automatique (IMA), nous avons été amenés à travailler au sein du laboratoire de l'IRCICA sur un projet de capteurs communicants intelligents en collaboration avec l'artiste Wax Tailor et la région Nord Pas-de-Calais. Au-delà des enjeux scientifiques et technologiques lié à l'innovation, cette réalisation s'inscrit dans une collaboration artistique de grande envergure visant à améliorer l'expérience du publique lors d'un concert en mettant en œuvre des outils à la fois visuels mais aussi d'interaction et de participation à la représentation.

Nous allons voir à travers ce dernier rapport :

- L'avancement final du projet,
- La description des problèmes et les solutions proposées à ceux-ci,
- L'étude des derniers résultats obtenus,
- La conclusion sur l'ensemble du projet.

I Le Sujet

Ce projet consiste en la réalisation de bracelets lumineux localisables en intérieur dans le but de réaliser un matrice de lumière à l'échelle d'un publique de théâtre (ex : Théâtre Sébastopol de Lille). Au-delà de former des images ou un ensemble de symboles les bracelets devront pouvoir interagir avec leur porteur au travers de lumières de vibrations et d'un accéléromètre (ex : vibration synchronisée avec les applaudissements). La technique de localisation est à la base du projet totalement libre mais il parait évident qu'il est nécessaire d'intégrer un module radio.



Images tirées du dossier de financement du projet

II Cahier des charges

Le document fourni pour le projet nous donne un scénario d'interactivité déjà imaginé :

1. Le public est invité à taper dans les mains pour appeler les artistes à monter sur scène. Chaque hand-clap provoque une réaction des bracelets.

2. Wax Tailor prend la main sur les bracelets, invite le public à placer ses mains en fonction du besoin, et mappe la salle de spectacle pour faire du public un espace d'interactivité et de projection d'image pixélisée. Chaque spectateur devient donc, par son bracelet, un PIXEL.

Voici un extrait de la demande formulée par notre client qu'en aux fonctionnalités que devra comporter le bracelet. Dans une optique d'ingénierie cela peut se traduire par la prise en compte des éléments présentés dans les parties suivantes.

Partie 1 : Interaction Bracelet / Porteur du Bracelet

Dans sa partie la plus simplifiée le bracelet est considéré comme un système complètement isolé (des autres bracelets) ne relevant que des informations liées à son porteur. Pour détecter un « hand-clap » il faut pouvoir acquérir au moins un paramètre dynamique des mains du porteur. Une combinaison position/accélération donnant une meilleure précision. Ensuite acquérir cette information ne sert à rien si elle n'est pas réutilisée pour améliorer l'expérience artistique du porteur, celui-ci devra donc réémettre cette information à son utilisateur d'une façon ou d'une autre. Le support pouvant être lumineux, vibratoire ou sonore. Cette dernière option étant mise de côté car cela représente peu d'intérêt de rajouter un son par-dessus celui produit par les mains de façon naturelle lors d'un concert.

Contraintes

Le système doit être sous la forme d'un bracelet, cela induit un certain nombre de considérations qu'en à la taille de ses composants, leur flexibilité ainsi que leur consommation énergétique. En effet le spectateur ne doit pas être gêné par le système et doit pouvoir l'utiliser tout au long du spectacle sans avoir à renouveler sa source d'alimentation où à le configurer. Toutes ces contraintes sont principalement liées à la forme et à la structure du système.

Partie 2 : Interaction Wax Tailor / L'ensemble des bracelets de la salle

« Wax Tailor [...] mappe la salle de spectacle pour faire du public un espace d'interactivité et de projection d'image pixellisée »

Dans ces propos il apparaît comme indispensable que nous devrions localiser chaque bracelet de façon unique afin de pouvoir lui attribuer l'adresse d'un pixel dans le motif final. Cette localisation est donc forcément active, c'est-à-dire que la scène doit pouvoir communiquer avec chaque bracelet de façon indépendante afin à la fois de relever et de communiquer des informations.

Contraintes

Bien que l'artiste soit prêt à mettre à contribution le public pour aider à la localisation, celle-ci doit être le plus rapide possible et ne nécessiter qu'une vingtaine de seconde pour cibler l'ensemble de la salle. Ce temps peut être considérablement augmenté à l'échelle de plusieurs minutes si la localisation ne nécessite pas de position particulière du bracelet.

La conception du bracelet doit prendre en compte des aspects d'industrialisation et de répétabilité dans sa réalisation de part le grand nombre (plus de 2000) de personnes constituant le public.

Le bracelet doit pouvoir acquérir des données, les stocker et les restituer à un système principal situé sur scène.

La localisation et la communication se faisant par radio, il faut pouvoir gérer un grand nombre d'émetteurs avec lequel il peut se produire des collisions de données.

Les bandes de fréquences utilisées doivent être libre d'utilisation en Europe et ne doivent pas perturber les autres équipements de la scène ou de la régie.

Le système doit pouvoir s'installer et être utilisé très facilement pour s'adapter aux utilisateurs finaux.

III Travail réalisé

III.1) Résumé du rapport précédent

Lors de notre précédent rapport nous avons déjà obtenus plusieurs résultats que nous résumerons ici.

III.1.a) Mise en place d'une solution technique et premiers résultats

Pour répondre au cahier des charges présenté dans la partie précédente, nous avons décidé d'utiliser la puissance de réception (RSSI) pour réaliser une trilatération à partir d'un ensemble tourelles émettrices (minimum trois).

Le bracelet sera donc composé d'un microcontrôleur permettant de stocker les images, réaliser les calculs,... ainsi que d'une puce émettrice/réceptrice permettant de récupérer le RSSI.

Pour pouvoir mettre en place les premiers tests, nous nous sommes dirigés vers l'utilisation d'Arduino UNO associé à un shield Xbee. Cette solution a pour avantage d'être rapide à mettre en place et de permettre de récupérer facilement le RSSI.

Pour ces tests, nous avons conçu un programme Arduino pour réaliser la fonction des tourelles soit, dans notre cas, d'émettre simplement leur ID cinq fois de suite et chacun leur tour. Le comportement du bracelet est qu'en à lui « simulé » par un ordinateur afin de facilement récupérer le RSSI de chaque tourelle.

Les résultats reçus lors de ces tests nous ont permis de mettre en évidence que la localisation basée sur le RSSI est possible mais qu'à 2,4GHz, le corps humain interfère énormément sur celui-ci.

III.1.b) Réalisation d'un prototype à base de MRF49XA

Après l'obtention de ces premiers résultats, nous en avons conclut que le Xbee était sensible à l'être humain du à sa forte masse en eau, élément qui créé des perturbations à la fréquence de 2,4GHz.

Pour résoudre ce problème, nous nous sommes orientés vers une fréquence inférieure qui est de 868 MHz. Pour cela, nous avons choisi d'utiliser le transmetteur/récepteur MRF49XA de chez Microchip.

Nous avons alors conçu le prototype suivant :

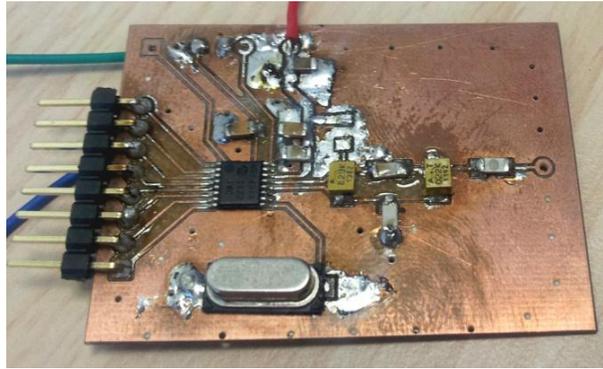


Photo 1 : Prototype actuel

Nous pouvons voir sur le schéma, présenté en annexe 1a, la présence d'un ballast afin d'adapter une antenne d'impédance de 50 ohm à l'entrée d'impédance 250 ohm de la puce. L'ensemble des entrées et sorties de la puce restent accessibles.

Lors du développement de la librairie, nous avons alors rencontré des problèmes pour l'envoi de paquets. Ayant conçu le code pour l'utilisation du port SPI, nous nous étions d'abord orienté vers un problème avec cette librairie. Un test avec une carte SD nous a alors permis de la valider. Un problème de reset avait aussi été mise-à-jour et réglé. Cependant l'envoi de paquets ne s'effectuait toujours pas.

III.1.c) Conception du protocole

Pour pouvoir faciliter le développement, nous avons ajouté une base qui permettra de synchroniser les tourelles, d'envoyer les images et récupérer les informations nécessaires à la localisation des bracelets.

Pour faire communiquer l'ensemble, un protocole a été conçu.

Nom	MRF Réveil	MRF Synchro	Mrf Taille Paquet	Type Emetteur	ID	CMD
Nb bits	8	16	8	2	16	4
Nom	ID	CMD	TX/RX CMD	DATA	Stuffing	CRC
Nb bits	16	4	1	32	1	8

Tableau 1 : Tableau du protocole

Une partie des octets échangés sont à destination du MRF et sont données à titre indicatif. Elles ne sont donc pas prises en compte durant le parsing des données mais avant, lors du traitement à la réception par la librairie MRF.

L'ensemble de ce protocole a été implémenté dans la base et les bracelets / tourelles. Dans ce dernier cas, nous avons abandonné la programmation en passant par les bibliothèques Arduino au profit des bibliothèques AVR seul. Ce choix a été fait afin d'économiser la mémoire disponible dans l'ATMega328p afin

d'y stocker les "images". L'intégration du protocole et le développement du programme seront développés plus en détail dans la partie logiciel de ce rapport (III.3).

III.2) Le prototype

III.2.a) Le débogage du MRF49XA

Comme expliqué précédemment, le programme de l'AVR ne permettait pas encore d'envoyer de paquet ou d'en recevoir.

Suite aux précédents tests effectués auparavant, nous nous sommes alors orientés vers le fonctionnement de notre librairie MRF49XA.

Un peu de technique : comme le montre le schéma ci-dessous, quand l'on souhaite envoyer un paquet, nous modifions les registres nécessaires pour activer la transmission (TXDEN et TXCEN à l'état haut) et nous envoyons le premier octet qui permet de réveiller le MRF (0xAA) précédé de l'adresse du registre de transmission. On attend ensuite une interruption de la part du MRF pour envoyer le second octet de réveil. Puis à chaque interruption suivante, un octet de donnée est envoyé. Une fois tous ces octets envoyés, on modifie de nouveau les registres nécessaires du transmetteur afin de réactiver la réception (TXCEN et TXDEN à l'état bas).

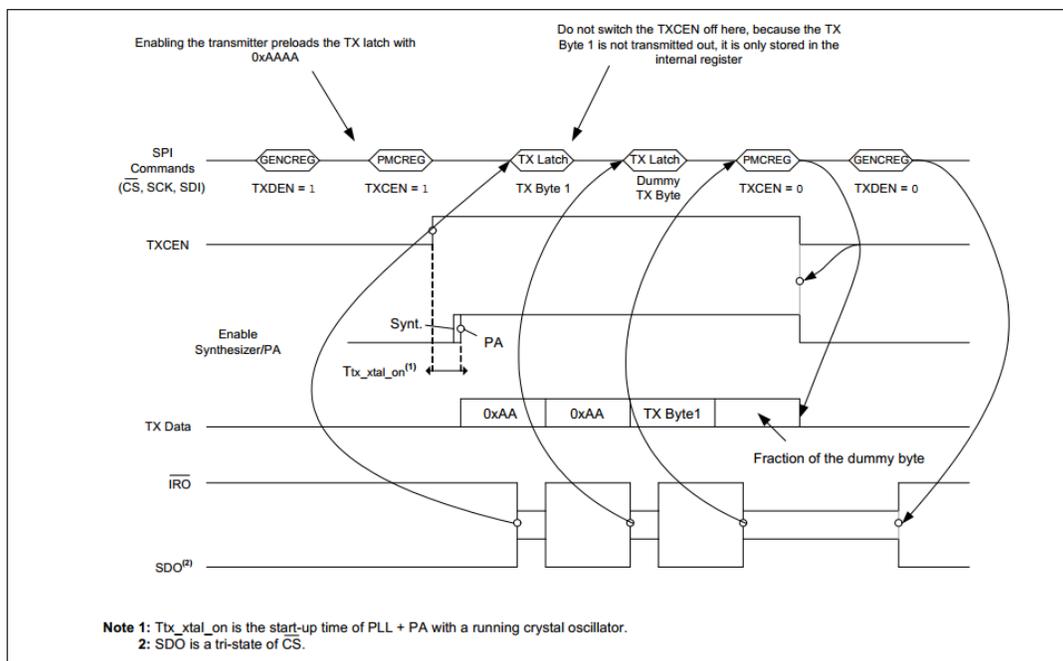


Figure 1 : Utilisation du registre de transmission¹

Dans notre cas, nous ne recevons jamais la seconde interruption, alors que nous avons bien reçu la

¹ Datasheet du MRF49XA - Page 69, Figure 3-16

première.

A l'analyseur de spectre, aucune information n'a été transmise. Ce qui est normal car les deux premiers octets sont à destination du MRF49XA afin de le réveiller.

Plusieurs défauts ont alors surgit dans notre programme :

- à chaque interruption, nous indiquions l'adresse du registre de transmission avant l'octet de donné. Hors, cette adresse ne doit être envoyé qu'à la première interruption.
- le programme de test d'envoi de paquet envoyé régulièrement une demande d'envoi, ce qui créait un conflit entre les paquets.
- la lecture régulière du registre de statut peut rentrée elle aussi en conflit avec l'envoi du paquet. (transmission de l'adresse de registre de statut en plein milieu des données). Cependant, n'étant pas envoyé pendant la période d'interruption, le MRF peut sûrement gérer ces demandes.

Après cette étude approfondit du software qui n'a, malgré le respect du protocole de la puce MRF49XA indiqué dans la datasheet, pas permis de résoudre le problème, nous soupçonnons le prototype d'être à l'origine du problème (court-circuit au niveau du ballast de l'antenne d'où le problème uniquement lors de l'envoi).

III.2.b) Conception d'un nouveau prototype

Afin de simplifier les tests, nous avons réalisé un nouveau prototype réunissant un ATMEga328p et un MRF49XA ainsi que son ballast. Ce nouveau schéma, visible en Annexe 1b, contient également une led RGB permettant de réaliser nos tests d'affichage de pixels de couleurs.

L'utilisation de la led RGB impose des modifications dans la librairie : la broche d'interruption du MRF, qui était utilisée avec la pin PD3 de l'ATMEga (Int1), est déplacée sur la pin PD2 (Int0). Les broches de la leds sont qu'en à elles branchées sur les pins permettant d'utiliser la PWM soit PD3 (OC2B), PD5 (OC0B) et PD6 (OC0A).

Afin de pouvoir ajouter de nouvelles fonctionnalités, l'ensemble des entrées et sorties du microcontrôleur reste disponible.

Il est à noter que ce schéma ainsi que le prototype actuel sont utilisés aussi bien pour le bracelet que pour la tourelle.

III.3) Partie Logicielle

III.3.a) Le Protocole de Communication

Comme précisé précédemment, le système pour bien fonctionner dans son ensemble a besoin d'un protocole de communication qui définit la structure des messages échangés ainsi que le comportement et la hiérarchie des différents éléments (BASE/TOURELLE/BRACELET). Celui-ci a évolué au fur et à mesure du projet jusqu'à se fixer définitivement lors de l'implémentation du comportement des différents organes. La structure d'une trame est toujours la même, elle est composée de 64 bits (soit 8 octets) dont la répartition et la signification est la suivante :

Signification	Base/Tourelle /Bracelet	ID Emetteur	Code Commande	Emission/Reponse Commande	Data	Stuffing	CRC
Nb Bits	2	16	4	1	32	1	8

001 = Base / 011 = Tourelle / 111 = Bracelet

1 = Emission / 0 = Réponse

CRC = Somme des 8 octets modulo 255

Le champ DATA est de taille fixe (32 bits) cependant son contenu varie en fonction de la commande et tout les bits ne sont pas toujours utilisés. Le tableau suivant renseigne sur ces différentes possibilités et fait état de ce qui a été réellement implémenté dans le code de la Base des Tourelles et des Bracelets :

Ci-dessous les commandes nécessaires pour répondre au cahier des charges.

CODE	CMD	Emetteur	Description
0001	TX TOUR	BASE	Autorise une Tourelle à Emettre
0010	RSSI TOUR	BASE	Interroge un ou des bracelets pour connaitre le RSSI des Tourelles
0011	RSSI MAX	BASE	Interroge un ou des bracelets pour connaitre les N RSSI plus fort recus
0100	ENTENDU	BASE	Demande à tout les bracelets qui ont entendu un ID particulier de répondre
0101	PING	BASE	Ping un bracelet en particulier
0110	RESET	BASE	Demande à un bracelet en particulier de vider sa mémoire
0111	SET LED	BASE	Allume la LED d'un bracelet en particulier
1000	SET POS TAB	BASE	Définit le PIXEL que représente le bracelet dans l'image en mémoire
1001	SET VIBE	BASE	Fait vibrer le vibreur d'un bracelet en particulier
0001	TX TOUR	TOUR	Trame émise par une Tourelle
1111	ACK	TOUR	Accusé de Reception
0010	RSSI TOUR	BRAC	Le Bracelet renvoi le RSSI Moyen reçu des tourelles qu'il a entendu
0011	RSSI MAX	BRAC	Le Bracelet renvoi les N plus forts RSSI reçu des autres bracelets entendus
0100	ENTENDU	BRAC	Le Bracelet renvoi le RSSI reçu du bracelet recherché
1111	ACK	BRAC	Accusé de Reception

Suivi de la description du champ DATA pour chacun des sous systèmes et de leur comportement.

CODE	CMD	EMETTEUR	DATA [nb de bits]	Comportement Recepteur		
				BASE	TOUR	BRAC
0001	TX TOUR	BASE	IDTOUR [16] / NBTRAME [3] / INTERVAL (x10ms) [8]	-	Emettre N Trames	Incremente NbTour
0010	RSSI TOUR	BASE	IDBRAC [16] / MASK [16]	-	En Option	Renvoi le RSSI des TOURS
0011	RSSI MAX	BASE	IDBRAC [16] / MASK [13] / N [3]	-	En Option	Renvoi les N plus forts
0100	ENTENDU	BASE	IDBRAC [16]	-	-	Renvoi RSSI si ID dans la Liste
0101	PING	BASE	IDBRAC [16]	-	Repond au PING	Repond au PING
0110	RESET	BASE	IDBRAC [16]	-	-	RAZ DATA enregistrées
0111	SET LED	BASE	IDBRAC [16] / RED [3] / GREEN[3] / BLUE [3]	-	En Option	Allume la LED (R,G,B)
1000	SET POS TAB	BASE	IDBRAC [16] / POSITION [16]	-	En Option	Associe le Bracelet au PIXEL d'une image enregistrée en mémoire
1001	SET VIBE	BASE	IDBRAC [16] / DUREE (x10ms) [8]	-	En Option	Vibre pendant duree (x10ms)

0001	TX TOUR	TOUR	IDTOUR [16] / NUMTRAME [3] / NBTRAME [3] / INTERVAL [8]	Save Trame	En Option	Save Trame
1111	ACK	TOUR	0x55550000	Save Trame	-	-

0010	RSSI TOUR	BRAC	IDTOUR [16] / NBTRAME [3] / RSSIMOY [10]	Save Trame	En Option	Save Trame
0011	RSSI MAX	BRAC	IDBRAC [16] / N [3] / RSSI [10]	Save Trame	En Option	-
0100	ENTENDU	BRAC	IDBRAC [16]/ Stuff [3] / RSSI [10]	Save Trame	En Option	-
1111	ACK	BRAC	0x55550000	Save Trame	-	-

Save Trame	Enregistre la Trame sans la modifier
En Option	Cas où les tourelles sont assimilées à des bracelets
XXXX	Comportements des recepteurs implémentés
XXXX	Comportements des recepteurs prévus

Remarque : Afin de récupérer facilement le RSSI non avons configuré les Xbee en API mode cela signifie que l’Xbee implémente lui même un protocole de communication qui englobe notre propre protocole. Voici un extrait de la documentation des Xbee qui explique ce qu’est “L’API Mode” et la structure des octets rajoutés lors de la transmission et qu’il faudra filtrer avant toute interprétation des commandes

API Operation (AP parameter = 1)

When this API mode is enabled (AP = 1), the UART data frame structure is defined as follows:

Figure 3-01. UART Data Frame Structure:



MSB = Most Significant Byte, LSB = Least Significant Byte

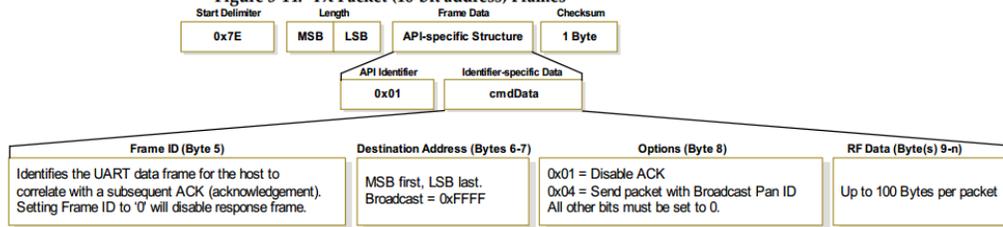
Ce qui implique les structure suivantes à ajouter/filtrer lors de la transmission/réception.

TX (Transmit) Request: 16-bit address

API Identifier Value: 0x01

A TX Request message will cause the module to send RF Data as an RF Packet.

Figure 3-14. TX Packet (16-bit address) Frames

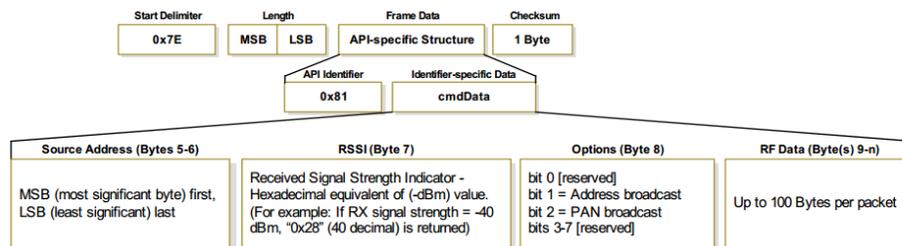


RX (Receive) Packet: 16-bit Address

API Identifier Value: 0x81

When the module receives an RF packet, it is sent out the UART using this message type.

Figure 3-17. RX Packet (16-bit address) Frames



Source : http://ftp1.digi.com/support/documentation/90000982_B.pdf

Ce qui donne pour une commande de PING par exemple :

La Base essaye de lancer un Ping sur le bracelet 4242 on voit ici le résultat correspondant à notre protocole

```
TX :TYPE=1 | ID=777 | CMD=5 | TRX=1 | DATA=10920000
RX :TYPE=3 | ID=4242 | CMD=5 | TRX=0 | DATA=170000
```

traduit en HEXA cela donne :

TX : 40 c2 56 21 24 00 01 9e

RX : c4 24 94 00 2e 00 01 ab

hors si l'on enregistre ce que reçoit réellement l'Xbee configuré en "API MODE" on obtient :

```
7e 00 0d 81 64 59 24 02 40 c2 56 21 24 00 01 9e
5f
7e 00 0d 81 64 59 3d 02 c4 24 94 00 2e 00 01 ab
2c
```

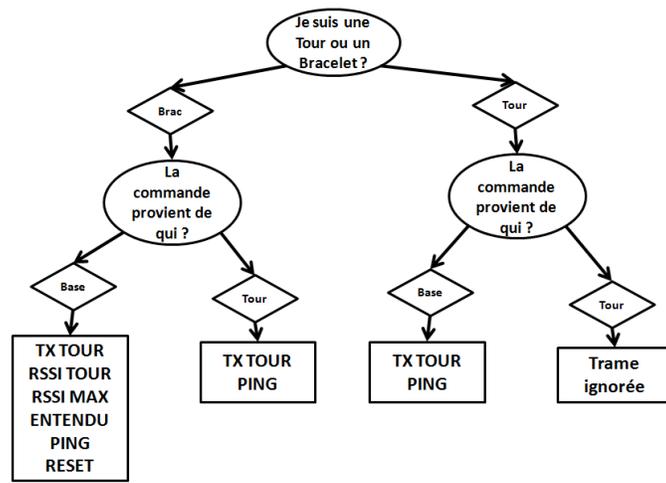
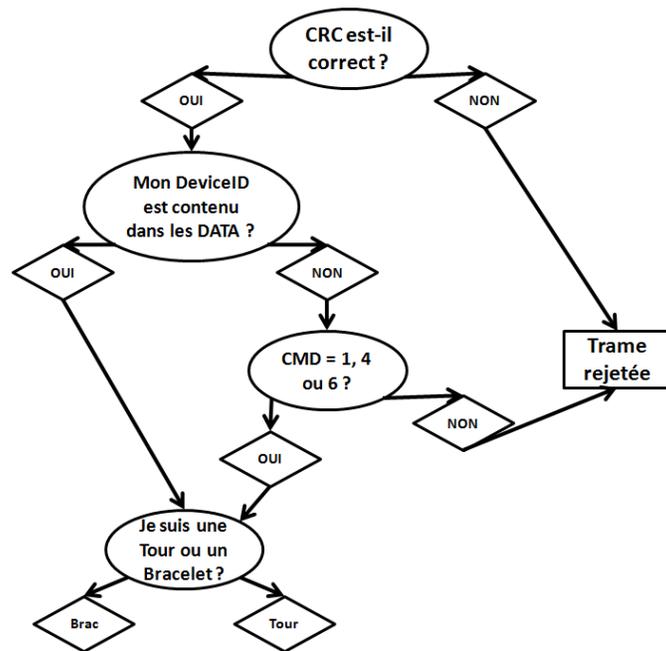
ce qui correspond à :

0x7E | API Identifier[1] + Source Address[2] + RSSI[1] + Option[1] + RF Data[8] = 13 = 0x0d | API Identifier(0x81) | Source Address(0x6459) | RSSI(0x24) | Option(0x02) | RF Data(...) | Checksum = 0xFF - ((0x81 + 0x6459 + 0x24 + 0x02 + Somme(RF DATA)) & 0xFF)

Donc quand le microcontrôleur enregistre les trames et les analyse, il tient compte de cette règle.

III.3.b) Interprétation des commandes

Grâce à ce protocole chaque sous système est en mesure d'identifier rapidement si il est concerné par la trame ou non avant de lancer la fonction de « parsing » complète très gourmande en temps de calcul (quelques secondes) pour le microcontrôleur embarqué et de choisir de l'ignorer ou non (en quelques millisecondes). De plus chacun des tests effectués sur la trame permet de structurer le code et de le rendre plus modulaire, ainsi il est facile d'ajouter ou de supprimer des commandes en fonction de leur place dans le diagramme suivant :



III.3.c) Structure du Code

Comme expliqué dans le résumé de notre précédent rapport (III.1.c), pour les bracelets et tourelles, les bibliothèques Arduino ont été abandonnées au profit des bibliothèques AVR.

Nous allons analyser ici deux situations, l'utilisation de l'ATMega328p avec le MRF49XA (pour le prototype) et l'utilisation d'un XBee. Dans ce deuxième cas, nous avons développé la bibliothèque Serial afin de communiquer avec le XBee. Dès qu'un paquet est présente sur le bus série, une interruption sur l'ATMega328p permet de la récupérer pour recomposer une trame. Une fois la trame recomposée, un "parsing" des données est lancé sur celle-ci pour récupérer les données si la vérification du CRC est bonne.

Dans le premier cas, une bibliothèque SPI a du être conçu avant de pouvoir communiquer avec le MRF49XA. La bibliothèque Serial est ici utilisée à des fins de débogage avec l'ordinateur.

Le programme de la base est lancé sur un ordinateur et communique avec les bracelets et tourelles à l'aide d'un XBee connecté à une interface série de l'ordinateur. Pour pouvoir communiquer avec le XBee, nous utilisons la librairie Teuniz RS-232². N'ayant pas accès aux interruptions comme sur l'ATMega328p, c'est un thread qui scrute régulièrement la présence de données sur le port.

A l'image du bracelet et de la tourelle, dès qu'une trame est recomposé, un "parsing" est lancé sur celle-ci afin de récupérer les données si la vérification du CRC est valide. Un ensemble de commandes permettent de réagir avec les tourelles et les bracelets.

Vous pouvez retrouver une partie de la répartition des différentes fonctions en Annexe 2

² Disponible sur <http://www.teuniz.net/RS-232/>, multiplateforme et sous licence GPL2

III.3.d) Résultats obtenus

Nous allons analyser les différents résultats obtenus lors de nos tests.

Visualisation des communications via la console de la Base

Voici ce que l'on obtient sur la console de la Base quand l'on lance quelques commandes avec le bon protocole :

Ping du Bracelet 4242 et des Tourelles 111, 222 et 333 (L'ID de la Base est 777) :

```
TX :TYPE=1 ! ID=777 ! CMD=5 ! TRX=1 ! DATA=10920000
RX :TYPE=3 ! ID=4242 ! CMD=5 ! TRX=0 ! DATA=170000

TX :TYPE=1 ! ID=777 ! CMD=5 ! TRX=1 ! DATA=14d0000
RX :TYPE=2 ! ID=333 ! CMD=5 ! TRX=0 ! DATA=300000

TX :TYPE=1 ! ID=777 ! CMD=5 ! TRX=1 ! DATA=de0000
RX :TYPE=2 ! ID=222 ! CMD=5 ! TRX=0 ! DATA=2c0000

TX :TYPE=1 ! ID=777 ! CMD=5 ! TRX=1 ! DATA=6f0000
RX :TYPE=2 ! ID=111 ! CMD=5 ! TRX=0 ! DATA=260000

TX :TYPE=1 ! ID=777 ! CMD=5 ! TRX=1 ! DATA=10920000
RX :TYPE=3 ! ID=4242 ! CMD=5 ! TRX=0 ! DATA=170000

TX :TYPE=1 ! ID=777 ! CMD=5 ! TRX=1 ! DATA=14d0000
RX :TYPE=2 ! ID=333 ! CMD=5 ! TRX=0 ! DATA=320000

TX :TYPE=1 ! ID=777 ! CMD=5 ! TRX=1 ! DATA=de0000
RX :TYPE=2 ! ID=222 ! CMD=5 ! TRX=0 ! DATA=2b0000

TX :TYPE=1 ! ID=777 ! CMD=5 ! TRX=1 ! DATA=6f0000
RX :TYPE=2 ! ID=111 ! CMD=5 ! TRX=0 ! DATA=290000
Appuyez sur une touche pour continuer...
```

Chaque sous système répond en donnant le niveau de RSSI avec lequel il a reçu le PING (17,30,2c,26,...)

On demande au bracelet 4242 d'effacer sa mémoire et on demande à la Tourelle 333 d'émettre 7 trames

```
TX :TYPE=1 ! ID=777 ! CMD=6 ! TRX=1 ! DATA=10920000
RX :TYPE=3 ! ID=4242 ! CMD=15 ! TRX=0 ! DATA=55550000

TX :TYPE=1 ! ID=777 ! CMD=1 ! TRX=1 ! DATA=14dec80
RX :TYPE=2 ! ID=333 ! CMD=1 ! TRX=0 ! DATA=14d3d90

> IDTOUR[16] : 333
> NUMTRAME[3] : 1
> MBTRAME[3] : 7
> INTERVAL[8] : 100

RX :TYPE=2 ! ID=333 ! CMD=1 ! TRX=0 ! DATA=14d5d90

> IDTOUR[16] : 333
> NUMTRAME[3] : 2
> MBTRAME[3] : 7
> INTERVAL[8] : 100
```

On demande au bracelet 4242 de nous redonner les données qu'il a enregistré

```

TX :TYPE=1 ! ID=777 ! CMD=2 ! TRX=1 ! DATA=10920000
RX :TYPE=3 ! ID=4242 ! CMD=2 ! TRX=0 ! DATA=278
  > IDTOUR[16] : 0
  > NBTRAME[3] : 0
  > RSSIMOY[10] : 79

RX :TYPE=3 ! ID=4242 ! CMD=2 ! TRX=0 ! DATA=14de278
  > IDTOUR[16] : 333
  > NBTRAME[3] : 7
  > RSSIMOY[10] : 79

RX :TYPE=3 ! ID=4242 ! CMD=2 ! TRX=0 ! DATA=14de280
  > IDTOUR[16] : 333
  > NBTRAME[3] : 7
  > RSSIMOY[10] : 80

```

Récupération des données

Grâce à un thread, la base enregistre sans filtrer toutes les trames qu'elle reçoit au format CSV après l'application de quelques règles de filtrage on obtient les différents niveaux de RSSI pour les tourelles.

A	B	C	D	E	F	G	H	I	J
Num	TYPE	ID	CMD	TRX	DATA	Stuff	CRC	RSSI	TOUR
4	3	4242	2	0	6fe2406fe240	1	148	72	111
4	3	4242	2	0	dee238dee238	1	99	71	222
4	3	4242	2	0	14de25814de258	1	130	75	333

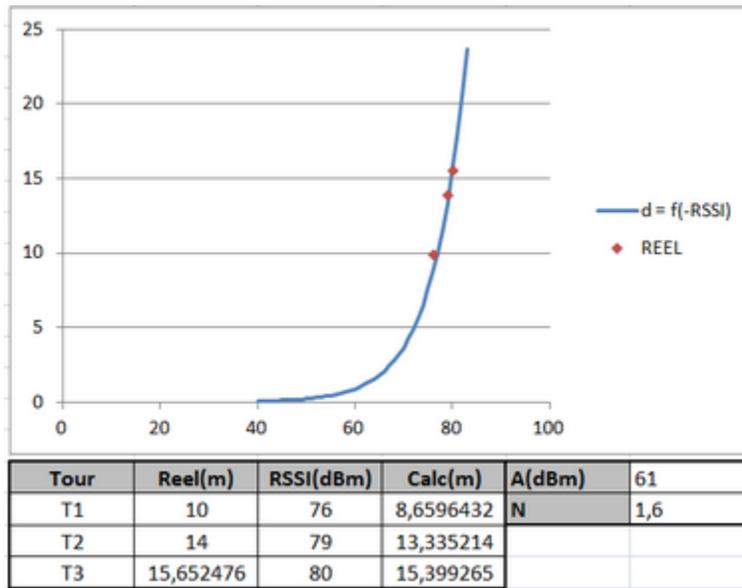
Représentation graphique des données

Afin de mieux visualiser les résultats obtenus et permettre une localisation dans le plan du bracelet on représente le RSSI sous forme de cercles centrés sur les tourelles. Pour rendre cela possible il faut convertir la valeur du RSSI reçu en dBm en mètre selon un modèle mathématique lié à la propagation des signaux dans le lieu de l'expérience. Si l'on prend tout les paramètres en compte cela peut devenir assez complexe car toutes les grandeurs ne sont pas directement et facilement accessibles (atténuation, rebond, bruitage, perturbations...)

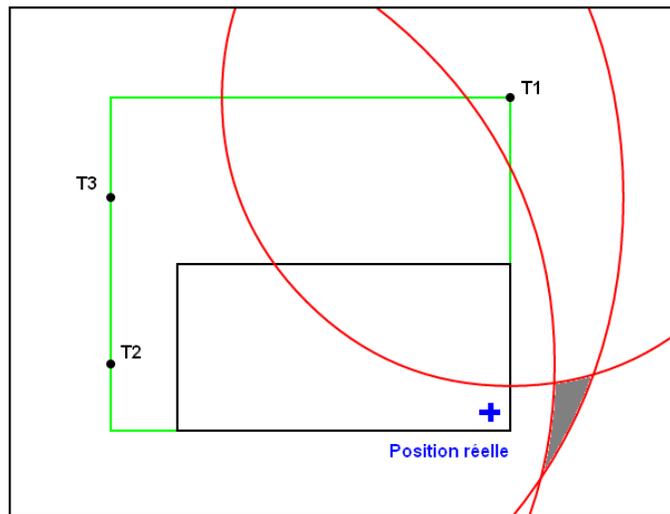
On utilise la version simplifiée suivante :

$$RSSI \text{ (dBm)} = -10 * n * \log_{10}(d) + A$$

où A représente le RSSI à une distance Emetteur <-> Récepteur de 1m et n représente un facteur d'atténuation dans le milieu (air = 2 environs).



Par rapport à une distance connue des Tourelles, on déduit les paramètres A et N approchant le mieux. On l'utilise pour représenter la position du bracelet dans l'amphithéâtre de l'IRCICA. Le carré vert représente la surface de l'amphi, le noir représente l'ensemble des sièges. Toutes les distances sont à l'échelle.



L'erreur constaté provient de différentes sources :

- La loi mathématique de conversion dBm <-> Distance et les paramètres associés
- Le fait que le RSSI est calculé seulement sur 8bits
- L'environnement lui-même (parasites, rebond, armatures en métal,...)
- etc...

Cependant même si on ne sera sûrement pas capable de distinguer 2 personnes assises côte à côte à l'échelle d'un petit amphithéâtre comme celui là on sera déjà capable de distinguer plusieurs zones.

IV Conclusion

Ces derniers mois à travailler sur le projet ont permis de mettre en évidence la complexité de la de conception d'un système de localisation INDOOR. Cependant, nos différents tests ont montré que dans une certaine mesure il était possible de distinguer un petit groupe de personnes en utilisant la puissance reçue de tourelles de référence.

La partie la plus aboutie du projet est le protocole de communication. Celui-ci est très léger (8 octets par trames) tout en permettant de réaliser l'ensemble des fonctionnalités du cahier des charges et d'inclure des données permettant d'éviter les erreurs de transmission (CRC, Checksum, Bit Start,...). Le comportement des différents sous systèmes en fonctions des différentes commandes est lui aussi assez abouti. Concernant la conception de notre propre système de transmission la conception théorique est avancée mais nous n'avons pas eu le temps de le fabriquer et de le tester de même que les modèles mathématiques complexes de localisation. L'ensemble très modulaire permettra d'aboutir de meilleurs résultats lors de futurs recherches.

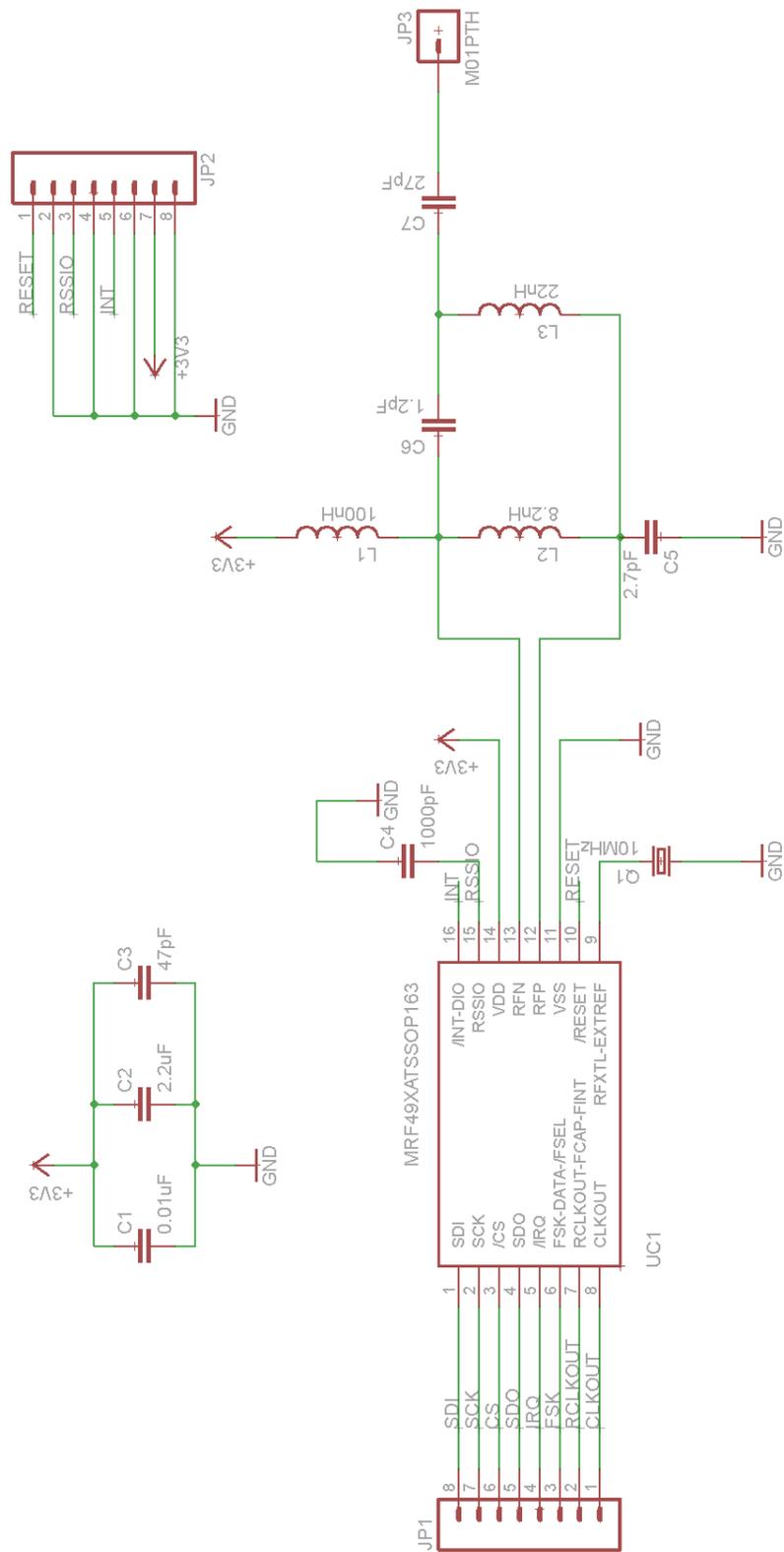
Une plus grande expérience ou une plus longue durée de travail aurait pu nous permettre d'effectuer nos dernières taches tels que

- La réalisation du nouveau prototype et les tests plus poussés de celui-ci,
- Un test grandeur nature dans une plus grande salle et en présence de plus de personnes,
- Une optimisation du prototype avec, notamment la suppression du ballast, ...

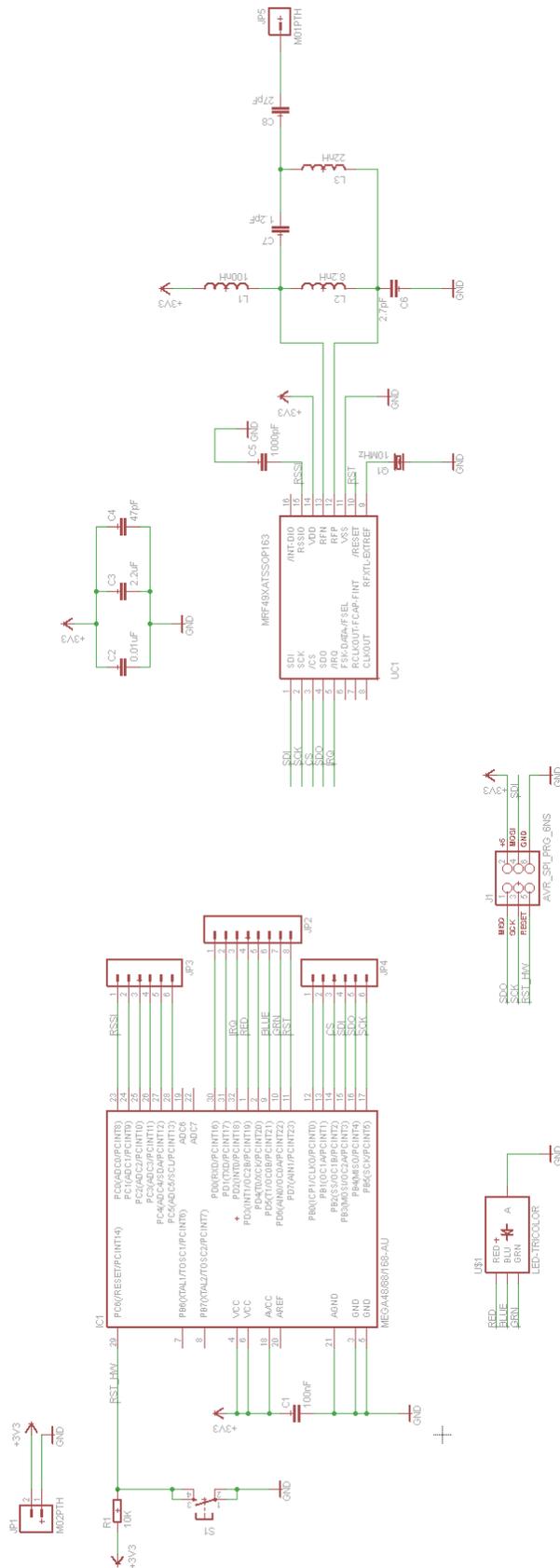
Annexe

- Annexe 1a : Schéma du prototype actuel
- Annexe 1b : Schéma du nouveau prototype
- Annexe 2 : Structure du Code (Fonctions)

Annexe 1a : Schéma du prototype actuel



Annexe 1b : Schéma du nouveau prototype



Annexe 2 : Structure du Code (Fonctions)

