

**PROJET TUTEURÉ :**  
**Dispositif RFID de mesures  
d'un temps de parcours**  
**Association Tours'N Aventure**



LELERRE Sébastien  
Licence EAM initiale  
2010-2011

Enseignant : Christine MÉRY  
Responsable : Thierry LEQUEU  
Demandeur : Denis COURATIN

**PROJET TUTEURÉ :**  
**Dispositif RFID de mesures  
d'un temps de parcours**  
**Association Tours'N Aventure**

## **REMERCIEMENTS**

Je tiens à remercier Monsieur Thierry Lequeu pour m'avoir fourni son aide et ses connaissances, ainsi que de m'avoir permis de travailler sur ce sujet enrichissant et intéressant.

Je remercie aussi Monsieur Yves Raingeaud de m'avoir apporté un supplément de connaissances à propos de la technologie RFID.

Je voudrais remercier pour finir le groupe de 2e année ayant mis au point les programmes des bornes.

# SOMMAIRE

Remerciements.....	3
Introduction.....	5
1.Présentation du système.....	6
1.1.Client : Tours'N Aventure[1].....	6
1.2.Cahier des charges[2].....	6
1.2.1.Présentation .....	6
1.2.2.Schéma structurel de niveau 1.....	7
1.2.3.Contraintes.....	7
2.Qu'est-ce que la RFID?[3].....	8
2.1.Le marqueur : TAG.....	8
2.2.Système RFID : module et antenne[4][5].....	10
3.Planning.....	11
4.La borne RFID : Départ, Intermédiaire, Arrivée[6].....	12
4.1.Schéma fonctionnel de niveau 2.....	12
4.2.Fonctionnement : FP1.....	13
4.2.1.FS1.1 : ALIMENTATION.....	13
4.2.2.FS1.2 : GESTION DE LA CARTE ET AFFICHAGE.....	13
4.2.3.FS1.3 : RESET.....	14
4.2.4.FS1.4 : MODULE RFID.....	15
4.3.Fonctionnement : FP2.....	16
4.3.1.Recherche de solutions.....	16
4.3.2.FS2.1 : AIGUILLAGE DES INFORMATIONS[7].....	18
4.3.3.FS2.2 : CHOIX DU TYPE DE SORTIE.....	19
4.3.4.FS2.3 : ADAPTATION DE TENSION.....	20
4.3.5.FS2.4 : CONNECTEUR RS232 ET USB[8][9].....	21
5.Nomenclature et coût de la carte d'arrivée.....	23
6.Création du typon et de la carte.....	24
6.1.ORCAD LAYOUT.....	24
7.Création de la carte et soudage des composants.....	26
7.1.L'insolation.....	26
7.2.Révélation.....	26
7.3.Gravure.....	26
7.4.Nettoyage.....	27
7.5.Soudage des composants.....	27
8.Programmation et test.....	28
8.1.Test.....	28
8.2.Programmation.....	29
Conclusion.....	31
Résumé.....	32
Abstract.....	33
Annexe.....	36

# INTRODUCTION

Durant mon année de licence professionnelle à l'IUT GEII<sup>1</sup> de Tours, il m'a été demandé d'effectuer un projet tuteuré. J'ai choisi le sujet de la borne RFID<sup>2</sup> qui permet la mesure d'un temps de parcours, ce sujet me paraissant enrichissant et intéressant, car la RFID de nos jours connaît un essor important dans différents domaines. Ainsi il m'est apparu que ce sujet m'apporterait de nouvelles connaissances sur l'électronique et ces applications.

Le projet ayant déjà été précédemment traité par d'autres élèves de l'IUT: Carl GIROUX année 2009, Vivien MARTINEZ année 2010, Thomas Hoguet et Rémi MARCHAND année 2010, il m'a fallu prendre connaissance des différentes remarques de chacun, les différentes solutions envisagées et gardées, ainsi une grande partie de mon projet fut la prise en main du projet antérieur.

Il m'a été demandé par la suite de créer une interface entre le système de borne et un ordinateur afin d'obtenir un traitement des informations de la borne. J'ai effectué différentes modifications sur les travaux précédents et mis au point l'interface.

Je présenterai le projet en commençant par son environnement, puis j'expliquerai quelques points sur la RFID, le système dans sa globalité, et enfin je détaillerai la partie sur laquelle j'ai le plus travaillé, l'interface avec l'ordinateur.

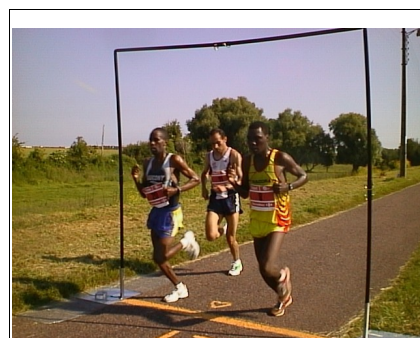
---

1 IUT GEII : Institut Universitaire de Technologie Génie Électrique et Informatique Industriel

2 RFID : Radio Frequency Identification (IDentification par Radio Fréquence)

# 1. PRÉSENTATION DU SYSTÈME

Le système de mesure d'un temps de parcours existe déjà sous différentes formes, par exemple un système de portique, ou le classique chronomètre. Les défauts de ces systèmes sont l'obligation d'une gestion en temps réel d'un opérateur comme pour le chronomètre, mais aussi le prix qui pour le portique a un coût important même à la location.



*Illustration 1: exemple de système (portique)*

## 1.1. CLIENT : TOURS'N AVENTURE[1]

Tours'N Aventure, créée en 2001, est une association à but non lucratif, composée d'une quarantaine de sportifs de tous âges, de tous sexes, principalement de la région de tours, adepte de différents styles de courses :

- ◆ trails et ultra-trails<sup>3</sup> (toute distance, de 10 à + de 200 km),
- ◆ raid multisport (à base de Trail, course d'orientation, VTT et canoë),
- ◆ course d'orientation (seul ou par équipe, de jour ou de nuit),
- ◆ randonnée VTT.

L'association a pour but d'effectuer des épreuves n'importe où en France, ainsi ces différentes données conditionneront le cahier des charges.

## 1.2. CAHIER DES CHARGES[2]

### 1.2.1. PRÉSENTATION

Lors d'une épreuve de course à pied, de plusieurs kilomètres, le coureur se verra remettre un badge RFID personnel. Il viendra alors pointer pendant la course, sur une borne prévue à cet effet. L'opérateur, une personne de l'organisation de l'événement gèrera la borne de départ et la borne d'arrivée en lui attribuant un numéro et en les synchronisant (possibilité de mettre des bornes intermédiaires).

Pour permettre une intervention rapide et sans difficultés de l'opérateur, les cartes seront placées dans un boîtier s'ouvrant facilement. Les commandes permettant le réglage de l'horloge et du numéro de borne seront placées en façade.

De plus la météo est à prendre en compte. Elle peut varier sur le lieu de l'épreuve (soleil, pluie...). Il faut que les cartes soient implantées dans un boîtier étanche. Les bornes doivent être transportables et stables, leurs poids doit donc être raisonnable.

---

<sup>3</sup> trails et ultra-trails : Le trail est le nom que l'on donne aux courses à pied sur terrain accidenté . Se court sur une distance de quelques dizaines de kilomètres (de 20 à 50 km). Au-delà, on parle d'ultra-trail.

Il est décidé que le budget doit être inférieur à 400 € pour les bornes et environ 200 € pour les badges pour l'association.

### 1.2.2. SCHEMA STRUCTUREL DE NIVEAU 1

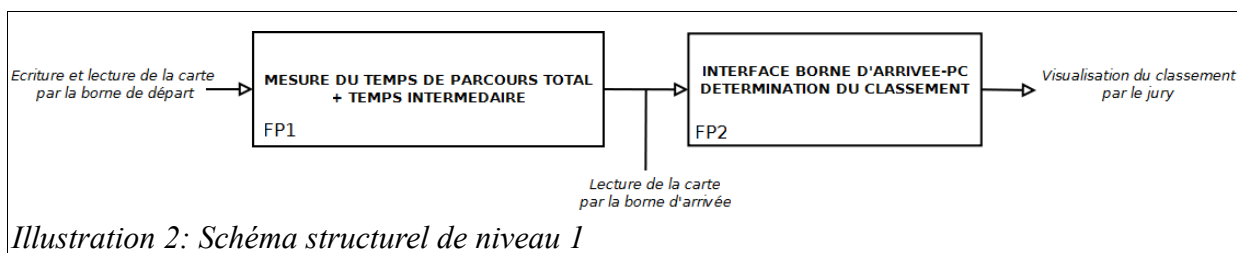


Illustration 2: Schéma structurel de niveau 1

Au départ, le participant à l'épreuve se positionne sur la ligne de départ après avoir reçu son badge, préalablement préparé par l'opérateur chargé de la course. Au lancement de la course, le sportif doit passer son badge devant la borne, qui lira les données du badge, et écrira l'heure de départ.

Durant la course, il est possible de disposer différentes bornes à différents endroits, afin d'éviter la tricherie par exemple, ainsi la borne écrira sur le badge l'heure de passage.

À l'arrivée est installée une dernière borne, qui lira les différents temps, et enverra, grâce à une liaison filaire de type RS232 ou USB, les informations à un ordinateur, qui disposera d'un programme permettant la création d'un classement. Il restera alors au jury à annoncer le classement.

### 1.2.3. CONTRAINTES

Lors de la prise de connaissance du sujet, j'ai dû définir les contraintes :

- ◆ numéro de chaque borne sélectionnable : interrupteurs (switchs)
- ◆ transmission d'informations : technologie RFID
- ◆ le microcontrôleur : ATMega 8535
- ◆ accumulateur 1800 mA/h 9 V
- ◆ autonomie des bornes : 2 heures
- ◆ interface avec un ordinateur : USB<sup>4</sup> et RS232<sup>5</sup>
- ◆ dispositif portable configuré par ordinateur
- ◆ enregistrement des temps sur le badge
- ◆ signal sonore lors de la bonne lecture du badge
- ◆ traitement des temps par ordinateur : réalisation du classement
- ◆ dimension du boîtier (fourni par l'enseignant) : 160x96x68 mm
- ◆ prix : 400 € par borne et 200€ pour 100 badges
- ◆ dispositif étanche
- ◆ poids raisonnable
- ◆ accessibilité simple aux configurations :
  - ◆ système de vis
  - ◆ commande en façade de la carte

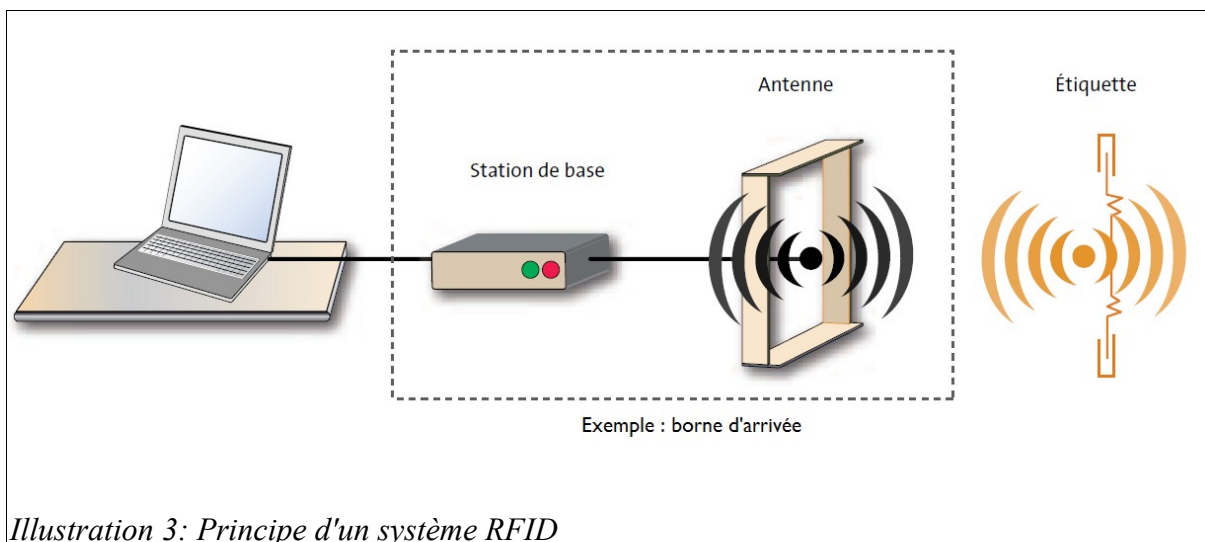
4 USB : Universal Serial Bus (Bus Série Universel)

5 RS232 : est une norme standardisant un bus de communication de type série sur trois fils minimum

Les différentes contraintes ont été décidées tout au long de la conception du système, ainsi les différents intervenants ont ajouté des contraintes qu'il m'a fallu comprendre et mettre au point. L'une des grandes exigences est la technologie RFID.

## 2. QU'EST-CE QUE LA RFID?[3]

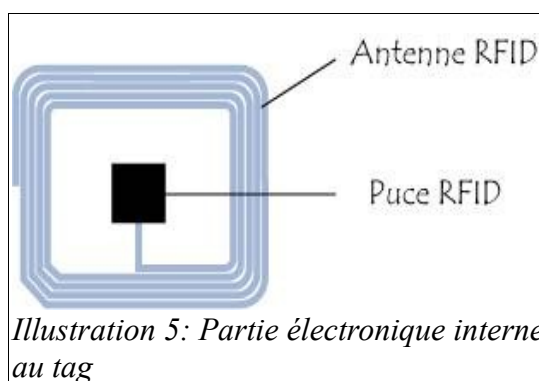
La radio-identification, plus souvent désignée par le sigle RFID (radio frequency identification) est une méthode développée pour mémoriser et récupérer des données à distance en utilisant des marqueurs appelés « radio-étiquettes », ou tag, ici cet objet correspond au badge porté par le coureur.



### 2.1. LE MARQUEUR : TAG

Les tags (marqueurs) RFID correspondent à un couple puce / antenne, apposé sur un produit ou un ensemble de produits. Ils permettent l'identification à distance, grâce à un lecteur qui capte les informations contenues dans la puce (un numéro de série, une description sommaire ou un numéro de lot par exemple). A la différence des codes barres, les tags RFID ne nécessitent pas que le lecteur soit approché du produit pour que l'identification s'opère.

Les tags peuvent être de deux types : actif ou passif. Les tags actifs contiennent une batterie interne qui permet à la puce d'être alimentée et de diffuser un signal à destination du lecteur. A l'inverse, les tags passifs n'ont pas de batterie, c'est le signal électromagnétique du lecteur qui active le tag et lui permet de fonctionner en y induisant un courant. Dans notre cas





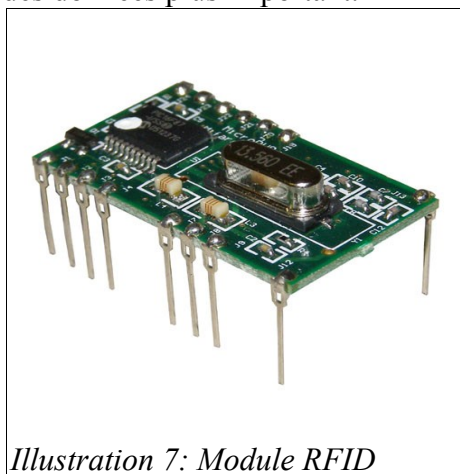
le badge est passif, cette caractéristique comporte l'avantage d'être moins chère que sa concurrente.

Une autre distinction peut être faite entre les tags inscriptibles et les tags uniquement lisibles. Les premiers peuvent recevoir de nouvelles informations en cours de route alors que les seconds ne peuvent que délivrer celles qu'ils renferment. Dans notre situation, les tags doivent être inscriptibles afin de mémoriser les temps de passages.

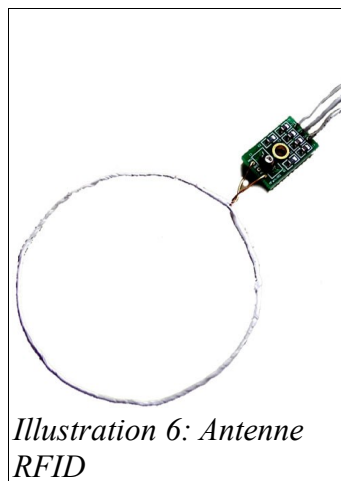
Pour communiquer, tag et lecteur doivent être sur la même fréquence mais, comme pour un poste de radio, celle-ci s'étale sur un spectre allant des basses aux très hautes fréquences. On trouve pour la technologie RFID les fréquences à 125 kHz (LF<sup>6</sup>), 13,56 MHz (HF<sup>7</sup>) et 850 MHz, dites UHF<sup>8</sup>. La fréquence choisit pour le projet est 13,56 MHz, j'en expliquerais la raison dans le point suivant.

## **2.2. SYSTÈME RFID : MODULE ET ANTENNE[4][5]**

Les différentes valeurs de fréquences énumérées précédemment correspondent à des fréquences libres, c'est à dire qu'elles sont du domaine public, ce qui correspond à notre besoin. Le choix d'un système fonctionnant à 13,56 MHz, est due au faite que les systèmes employant les basses et hautes fréquences coûtent moins chers que les ultra-hautes fréquences. La haute fréquence a été retenue car celle ci apporte l'avantage d'avoir une distance de lecture du tag plus importante que la basse, environ 10 cm pour la HF et moins de 10 cm pour la LF, cela ayant été testé par mes prédécesseurs. Mais elle permet aussi un débit de transfert des données plus important.



*Illustration 7: Module RFID*



*Illustration 6: Antenne RFID*

Pour ces raisons le module et l'antenne RFID choisi ont une fréquence de fonctionnement à 13,56 MHz.

---

6 LF : Low Frequency (BF : Basses Fréquences)

7 HF : High Frequency (Hautes Fréquences)

8 UHF : Ultra High Frequency (Ultra Hautes Fréquences)

### 3. PLANNING

Semaines	41	42	43	44	45	46	47	48	49	50	51	52	1	2	3	4	5	6	7
Etude du projet + cahier des charges	Prévisionnel	Prévisionnel																	
Étude des cartes réalisées				Réel	Réel														
Initiation à Code vision AVR					Prévisionnel														
Recherche de solutions techniques					Prévisionnel														
Mise en œuvre + réalisation						Prévisionnel	Prévisionnel	Prévisionnel	Prévisionnel	Prévisionnel	Prévisionnel	Prévisionnel	Prévisionnel	Prévisionnel	Prévisionnel	Prévisionnel	Prévisionnel	Prévisionnel	Prévisionnel
Tests + mesures														Prévisionnel	Prévisionnel	Prévisionnel	Prévisionnel	Prévisionnel	Prévisionnel
Présentation projet														Prévisionnel	Prévisionnel	Prévisionnel	Prévisionnel	Prévisionnel	Prévisionnel
Rédaction du dossier	Prévisionnel	Prévisionnel	Prévisionnel	Prévisionnel	Prévisionnel	Prévisionnel	Prévisionnel	Prévisionnel	Prévisionnel	Prévisionnel	Prévisionnel	Prévisionnel	Prévisionnel	Prévisionnel	Prévisionnel	Prévisionnel	Prévisionnel	Prévisionnel	Prévisionnel

■ Prévisionnel    
 ■ Réel    
 ■ Oral    
 ■ Vacances    
 ■ Formation Grenoble

J'expose ici mon planning prévisionnel et réel. On peut observer un écart entre le réel et le prévisionnel, cela est dû à une mauvaise évaluation du sujet, la recherche et la documentation au sujet de la RFID et la prise de connaissance de la partie déjà créée dans son intégralité m'ayant pris beaucoup de temps, les semaines non prévues m'ont permis de compenser ce retard. La partie routage m'a aussi pris plus de temps que prévu, ayant rencontré plusieurs difficultés.

## 4. LA BORNE RFID : DÉPART, INTERMÉDIAIRE, ARRIVÉE[6]

Je présenterai dans cette partie les différentes bornes permettant la mise en œuvre du système dans son intégralité. Les bornes de départ et intermédiaires se trouvent être identiques du point de vue matériel, tandis que la dernière borne, celle d'arrivée, correspond aux autres bornes, plus une partie permettant le transfert des informations vers un ordinateur.

### 4.1. SCHÉMA FONCTIONNEL DE NIVEAU 2

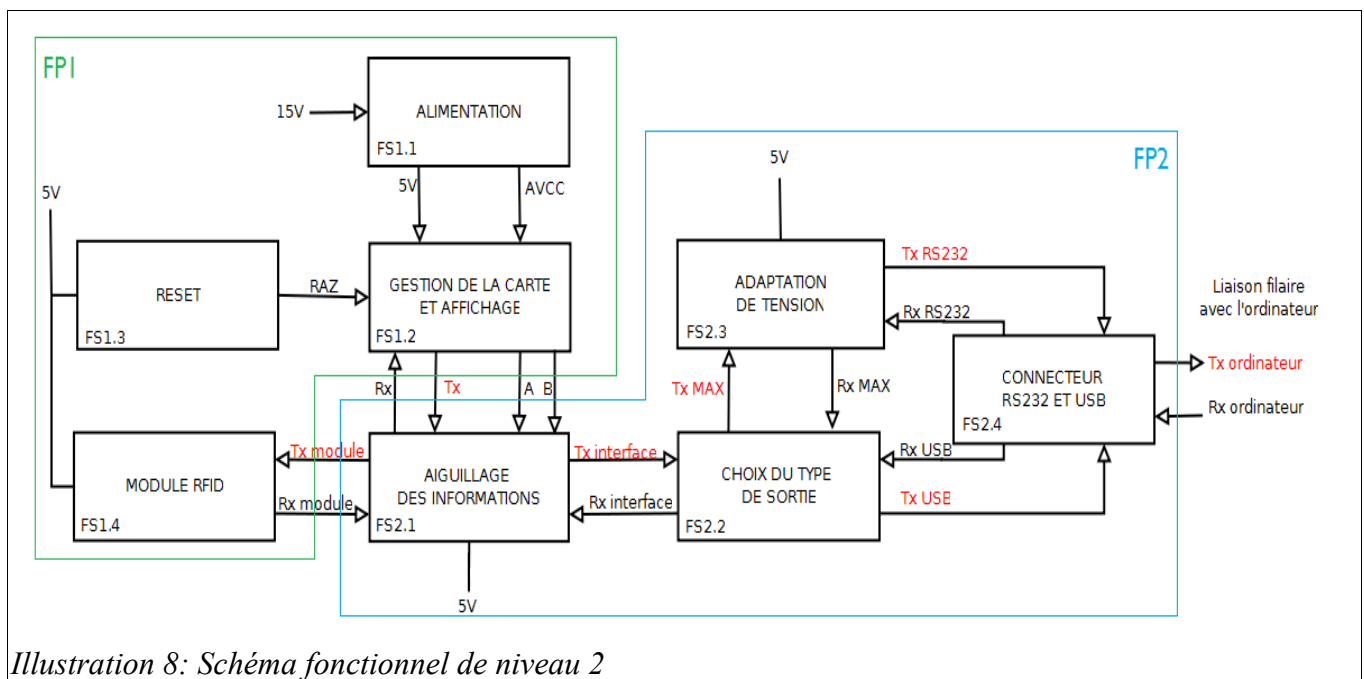
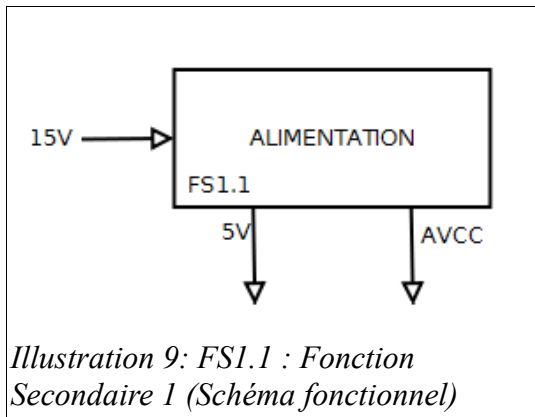


Illustration 8: Schéma fonctionnel de niveau 2

Nous pouvons visualiser sur l'illustration 8 les deux fonctions principales composées chacune de quatre fonctions secondaires. J'expliquerai par la suite succinctement les différents fonctionnements de chaque partie, puis je détaillerai la fonction principale 2 qui est la partie que j'ai mise au point.

## 4.2. FONCTIONNEMENT : FPI

### 4.2.1. FS1.1 : ALIMENTATION

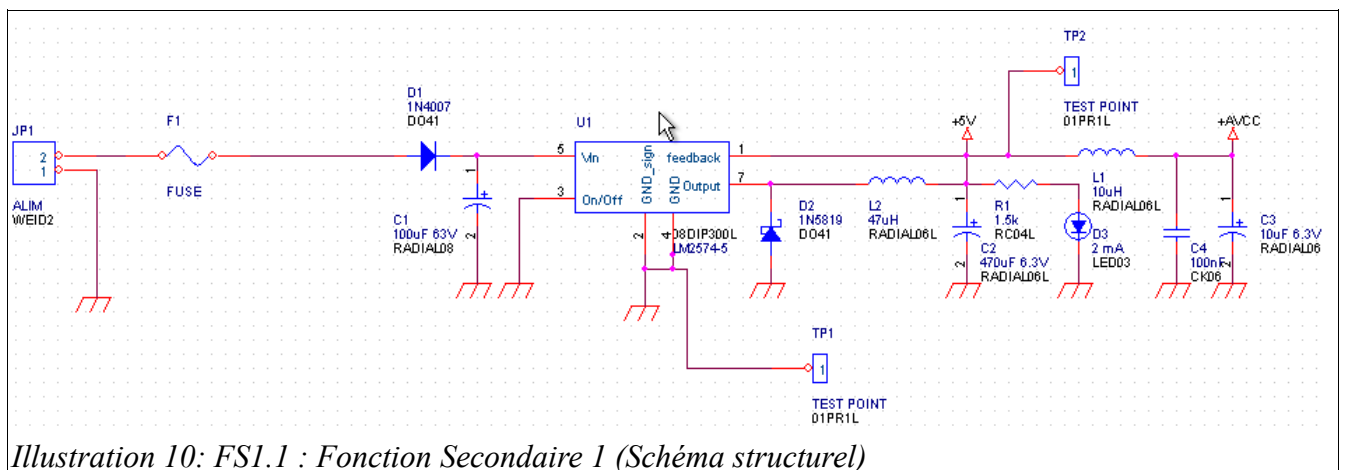


L'alimentation est composée d'un LM2574-5, qui permet de transformer une tension maximum de 40 V en 5 V, cette tension alimentant un grand nombre de composants.

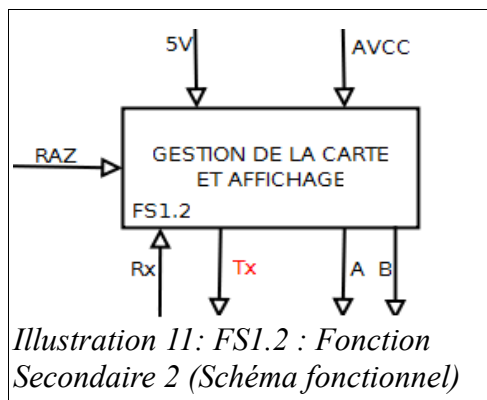
Un fusible est là pour éviter de possibles surtensions sur la plaquette, dans ce cas le fusible a son filament qui fond de façon à couper la liaison entre la source et la carte. De cette manière cela évite la destruction de composants.

Une LED<sup>9</sup> est présente, permettant de visualiser si le système est sous tension.

Une autre partie du montage permet d'obtenir une tension AVCC, qui correspond à une tension de 5 V Cette tension permet le fonctionnement du microcontrôleur.



### 4.2.2. FS1.2 : GESTION DE LA CARTE ET AFFICHAGE



Cette fonction est composée du microcontrôleur permettant la gestion de la carte, c'est à dire que c'est ce composant qui permet le traitement des données et le commandement de certains composants. Pour imager, les broches A et B permettent de piloter la fonction aiguillage, que j'expliquerai dans une prochaine partie.

9 LED (DEL) : Light-Emitting Diode ( Diode électroluminescente)

Il est alimenté en 5 V, ce qui correspond à la tension lui permettant de fonctionner. Le fonctionnement du composant est cadencé grâce à un quartz à 16 MHz.

Une notion importante d'un microcontrôleur est sa programmation, qui se fait grâce au « connecteur ISP ». Lors d'une programmation en cours la diode reliée au connecteur est allumée. Cette programmation permet entre autre, de faire interagir le système avec l'utilisateur grâce à l'écran LCD<sup>10</sup> qui affiche différentes informations.

Ainsi l'écran peut afficher le temps de parcours d'un concurrent dans notre cas. Le microcontrôleur permet aussi l'activation du buzzer d'une bonne lecture du badge, et une dernière partie, les switches (interrupteurs) permet d'attribuer le numéro à la borne.

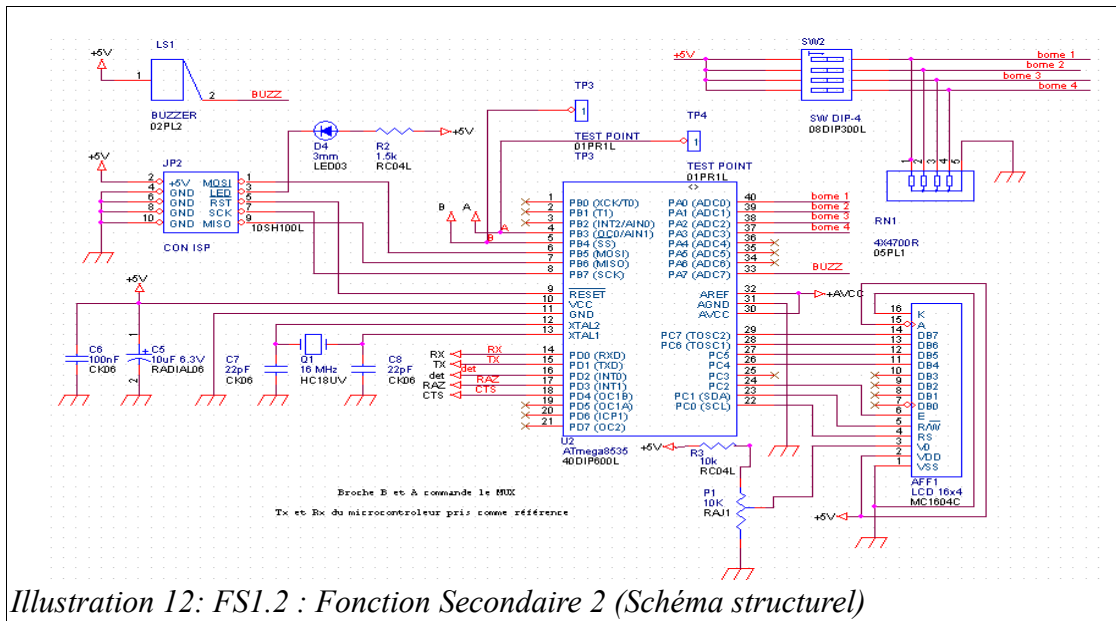


Illustration 12: FS1.2 : Fonction Secondaire 2 (Schéma structurel)

### 4.2.3. FS1.3 : RESET

Cette fonction permet la remise à zéro des bornes, cela afin de les synchroniser de façon à ce qu'elles soient synchronisées pour obtenir une mesure de temps valide.

Ainsi avant une épreuve, pour préparer les bornes, un opérateur devra relier toutes les bornes les unes aux autres, afin d'effectuer la remise à zéro en même temps, cela grâce à l'application d'un niveau bas sur la broche RAZ du microcontrôleur.

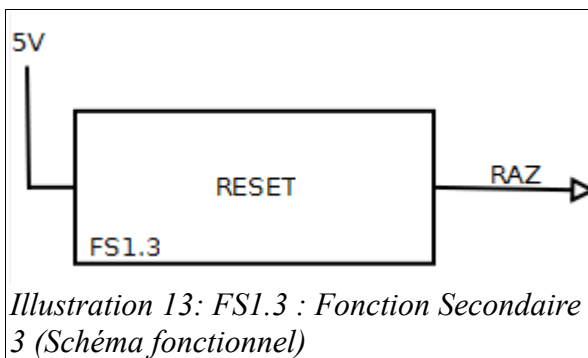


Illustration 13: FS1.3 : Fonction Secondaire 3 (Schéma fonctionnel)

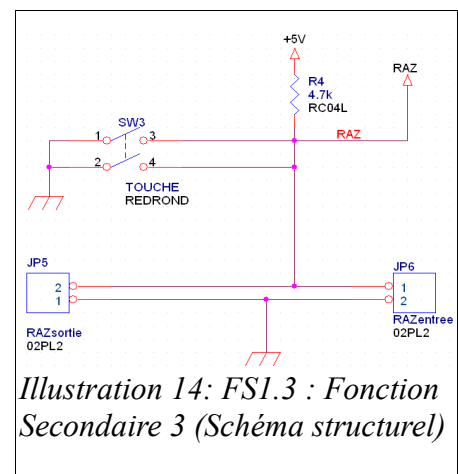


Illustration 14: FS1.3 : Fonction Secondaire 3 (Schéma structurel)

<sup>10</sup> LCD : Liquid Crystal Display (écran à cristaux liquide)

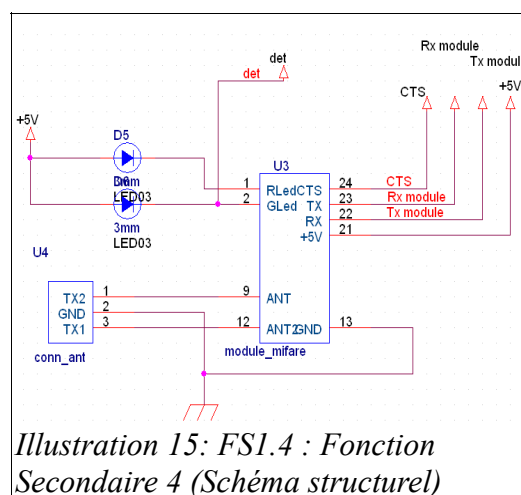
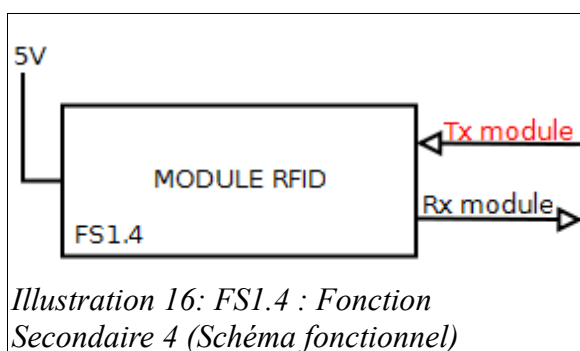
#### 4.2.4. FS1.4 : MODULE RFID

Le module RFID est le composant permettant la lecture et l'écriture du badge, ceci grâce à l'antenne correspondante à la fréquence de fonctionnement choisi, ici 13,56 MHz.

Lorsque les badges sont "éveillés" par le lecteur, ici le module, un dialogue s'établit selon un protocole de communications prédéfinies, et les données sont échangées. Cette partie correspond à de la programmation effectuée par un groupe de deuxième année.

Le badge fonctionne grâce à un champ électromagnétique créé par l'antenne associé au module, et l'antenne interne au badge pour communiquer. Le champ électromagnétique alimente le badge et active la puce. Cette dernière va exécuter les programmes pour lesquels elle a été conçue. Le lecteur reçoit ces informations et les transforme en code binaire. Les informations alors codées, sont transmises sur les ports Rx et Tx<sup>11</sup> permettant une liaison série avec le microcontrôleur.

Les deux LEDs permettent de visualiser la lecture de la carte. Au repos, la LED D3 est allumée, tandis que l'autre est éteinte, lors de la détection de la carte, nous avons la situation inverse, qui est conditionnée par le microcontrôleur, qui lors d'une détection applique un niveau bas sur D5. Le connecteur conn\_ant lui, permet de connecter l'antenne.



11 Tx et Rx : broche pour la liaison série (Tx : Transmit ; Rx : Receive)

### 4.3. FONCTIONNEMENT : FP2

La Fonction Principale 2 est la partie que j'ai dû mettre au point après avoir pris connaissance des travaux précédents, ceci afin de pouvoir faire la mise en œuvre de la fonction aiguillage, qui permet d'envoyer les informations du badge sur un port série, permettant le traitement des informations par un ordinateur.

Cette fonction est seulement présente sur la dernière borne : l'arrivée. Ainsi toutes les autres bornes, intermédiaires, départ, sont identiques et correspondent à la fonction FP1.

Pour expliquer ma démarche je commencerai par exposer mes recherches, puis j'expliquerai la mise en œuvre de la carte, en exposant chaque fonction d'un point de vue fonctionnel.

#### 4.3.1. RECHERCHE DE SOLUTIONS

La fonction a pour rôle d'envoyer les informations du badge sur un ordinateur, soit par port USB ou connecteur RS232. Afin de la mettre en œuvre, j'ai effectué différentes recherches, et ai reçu de l'aide de la part de mon tuteur Mr LEQUEU.

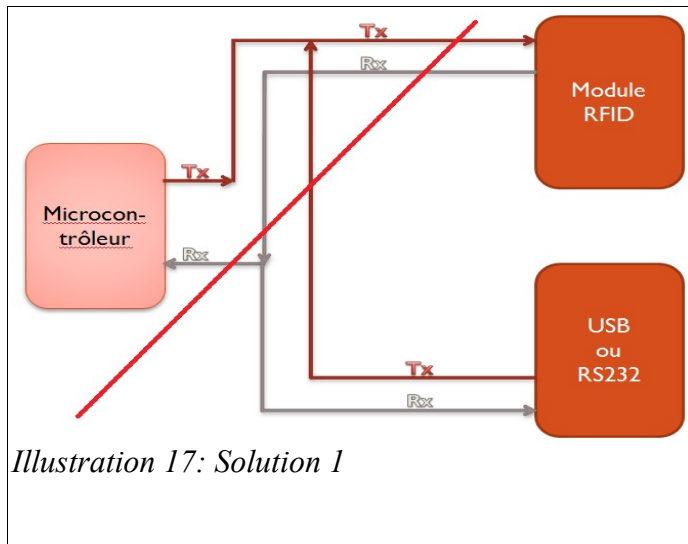
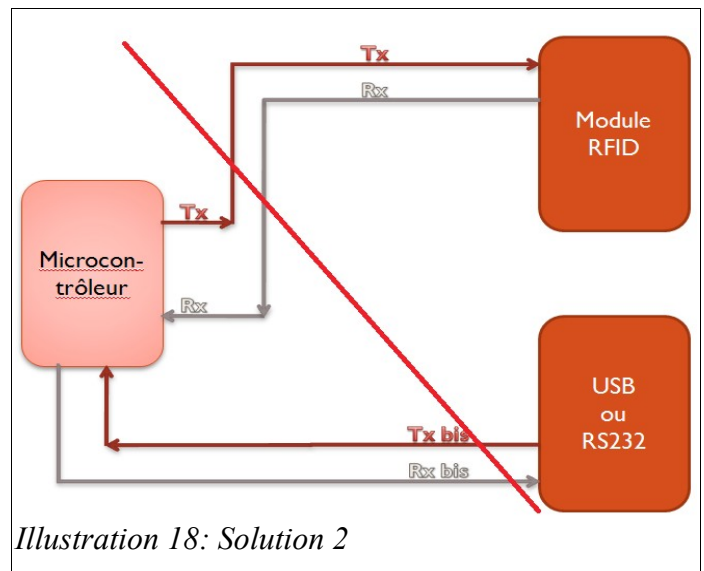


Illustration 17: Solution 1

Pour commencer je me suis donc tout d'abord demandé si il était possible de ce brancher sur les broches Rx et Tx, et ainsi transmettre les informations aux moments même de la lecture du badge. Cette solution c'est révélée impossible car cela provoquerai des problèmes sur la liaison série, il m'a donc fallu me diriger vers une autre solution.

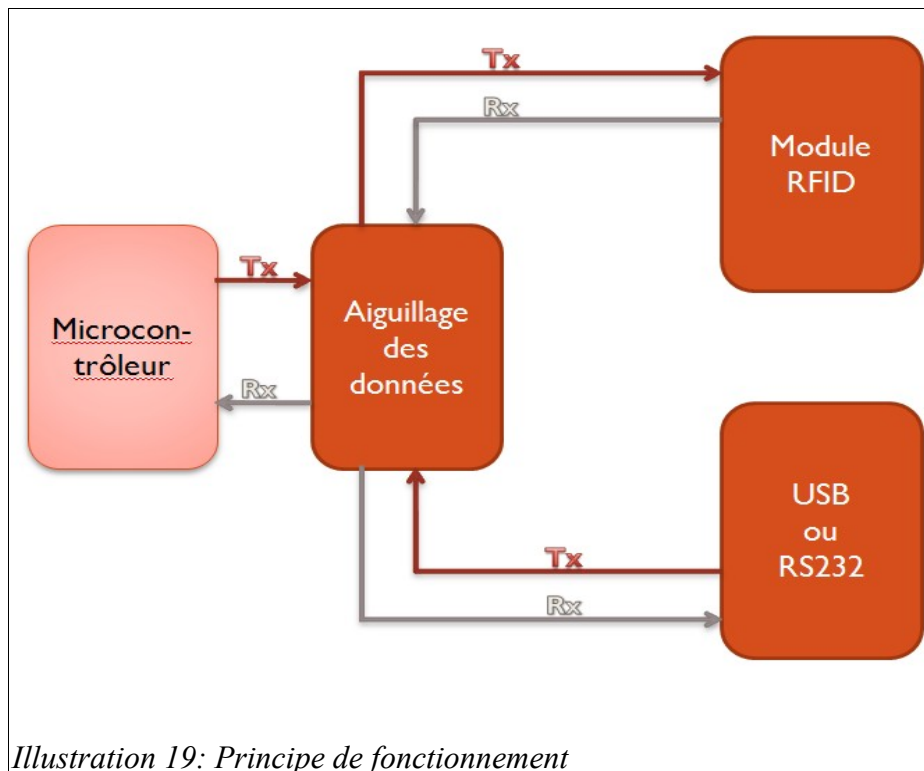
J'ai alors cherché une solution afin de ne pas interféré avec les broches Rx et Tx, tout en faisant circuler les informations sur ces broches. L'idée met venu alors d'un principe d'aiguillage. J'ai pensé effectuer cette fonction grâce à des transistors, mais mes simulations ne correspondant pas à mes attentes, j'ai alors continué mes recherches sur un composant me permettant de mettre en œuvre la fonction.

Sans toucher aux broches Rx et Tx, j'ai pensé à transmettre les informations sur d'autres broches inutilisées du microcontrôleur, en effectuant tout d'abord une sauvegarde des informations dans le composant, puis d'effectuer l'envoi de l'information sur un port. Cette solution c'est elle aussi trouver être impossible à mettre en œuvre car les informations ne peuvent être véhiculées que sur les broches Rx et Tx.



La solution retenue fut alors un mixe des précédentes recherches. Après consultation avec mon tuteur, j'ai découvert un composant qui m'était alors inconnu. Ce composant est de type multiplexeur/démultiplexeur, mais qui a une particularité qui est la possibilité de faire circuler les informations dans le sens que l'on veut, ces ports n'étant pas prédéfinis d'un statut d'entrée ou de sortie.

Ce composant a alors complètement correspondu à mes attentes, pouvant permettre d'aiguiller les broches Rx et Tx soit sur le module, soit sur les ports. Et afin de transmettre les informations, ces dernières seront enregistrées à l'intérieur du microcontrôleur lors de la lecture, puis transférées sur les ports, le microcontrôleur pouvant piloter le multiplexeur/démultiplexeur CD4052.

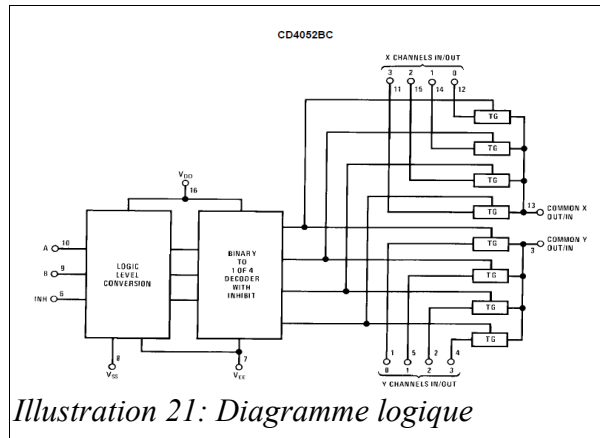
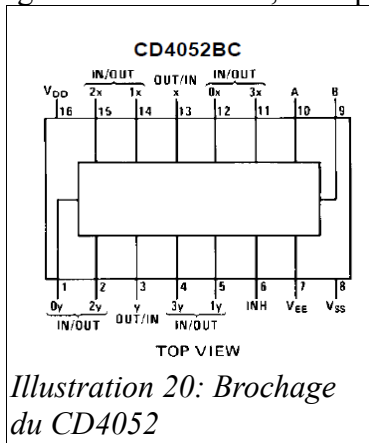


Par la suite j'expliquerai clairement le fonctionnement du composant dans l'étude fonctionnelle.



### 4.3.2. FS2.1 : AIGUILLAGE DES INFORMATIONS[7]

Je présenterai dans cette partie le fonctionnement du composant CD4052 permettant l'aiguillage des informations, en expliquant son fonctionnement et les branchements choisis.



En me référant à la documentation constructeur, j'ai ainsi branché les broches VDD, VEE, INH conformément au fonctionnement que je désirai pour ce composant.

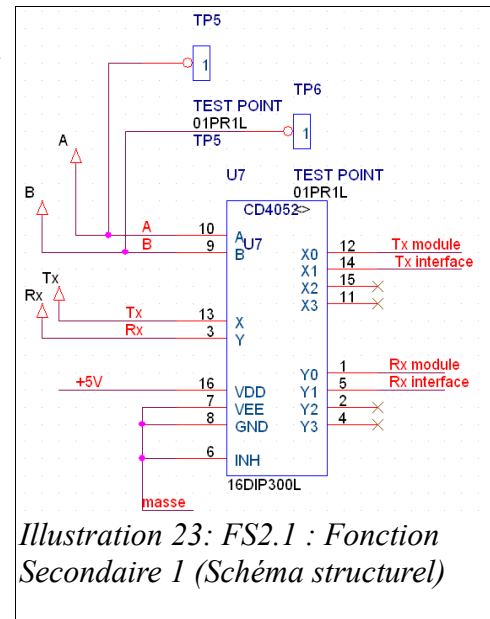
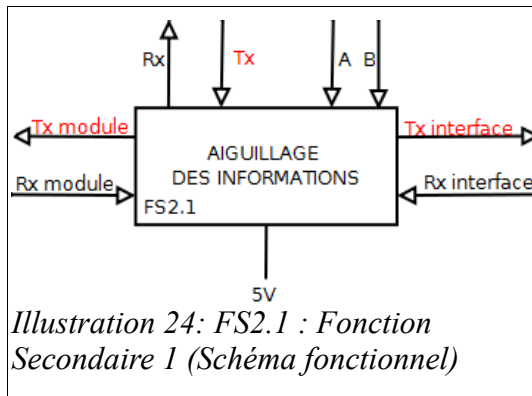
INPUT STATES				"ON" CHANNELS		
INHIBIT	C	B	A	CD4051B	CD4052B	CD4053B
0	0	0	0	0	0X, 0Y	cx, bx, ax
0	0	0	1	1	1X, 1Y	cx, bx, ay
0	0	1	0	2	2X, 2Y	cx, by, ax
0	0	1	1	3	3X, 3Y	cx, by, ay
0	1	0	0	4		cy, bx, ax
0	1	0	1	5		cy, bx, ay
0	1	1	0	6		cy, by, ax
0	1	1	1	7		cy, by, ay
1	*	*	*	NONE	NONE	NONE

Illustration 22: Table de vérité du CD4052

Nous pouvons observer que pour faire fonctionner le multiplexeur il faut imposer un niveau bas sur la broche INH<sup>12</sup> (INHIBIT). On peut observer le fonctionnement selon le niveau appliqué sur les broches A et B par le microcontrôleur, c'est grâce à ces deux broches que l'on peut aiguiller les informations en positionnant soit en 0X,0Y, qui correspond à la liaison avec le module, ou en 1X,1Y qui est la position pour envoyer les informations sur l'ordinateur. De cette façon nous pouvons aiguiller les informations.

12 INH : broche d'inhibition, valide le fonctionnement du composant ou non

Ce composant à donc répondu à mes attentes car il n'a pas d'entrée-sortie définie, qu'il supporte la tension de 5 V. Il est aussi simple d'utilisation car simple à commander.



### 4.3.3. FS2.2 : CHOIX DU TYPE DE SORTIE

Grâce à un système de cavalier, j'ai mis au point un moyen manuel de sélectionner le type de sortie désirée : RS232 ou USB.

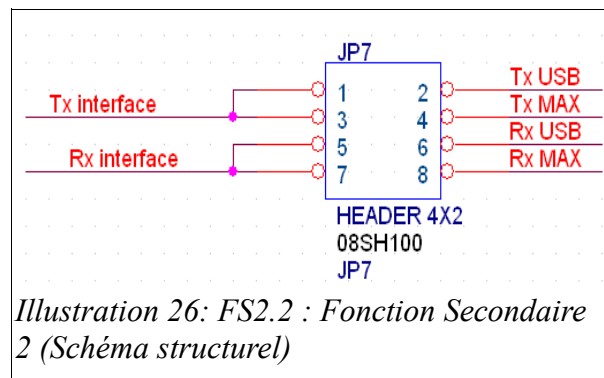
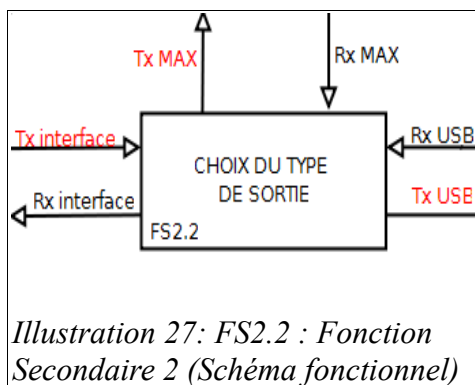
Broches reliées	Sélection du port	
	USB	RS232
1 et 2		
3 et 4		
5 et 6		
7 et 8		

zone rouge : correspond au broches à relier afin de sélectionner le port désiré

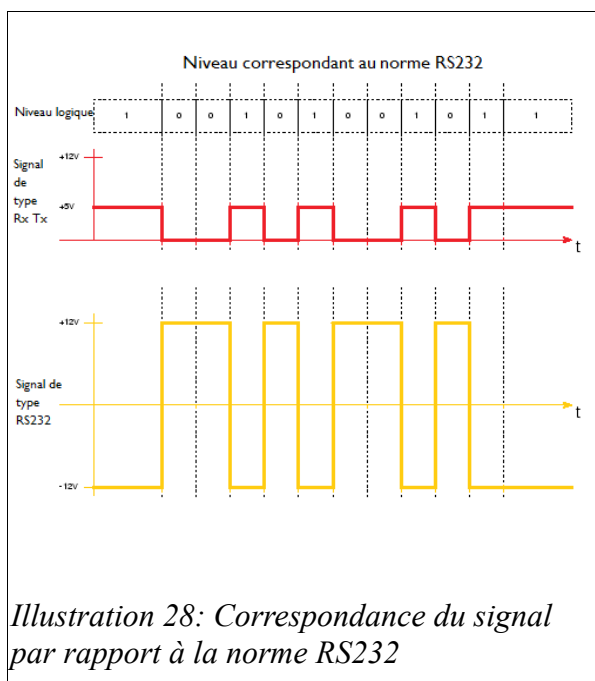
*Illustration 25: Sélection d'un port*

De cette manière, grâce à deux cavaliers positionnés à deux endroits différents, il est possible de faire passer le signal soit par la liaison série allant à l'USB ou le RS232.

Cette fonction sera prédéterminée par l'opérateur qui s'occupera de l'événement. Pour cette raison, il est facile de faire son choix, et la fonction se trouve en façade de la carte.



### 4.3.4. FS2.3 : ADAPTATION DE TENSION



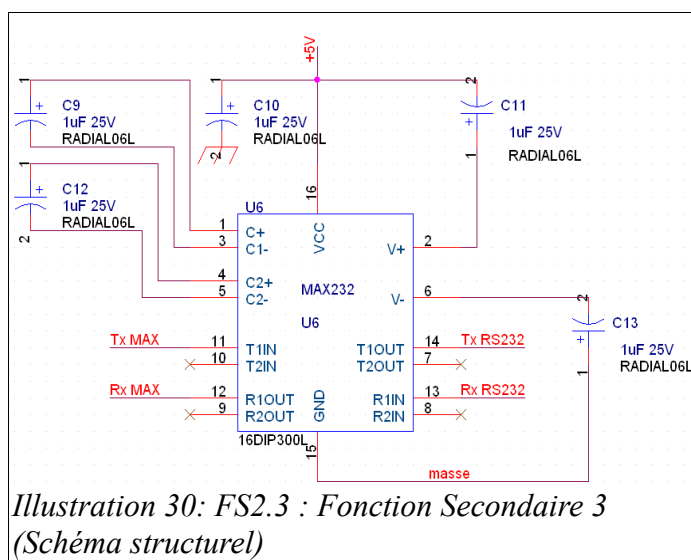
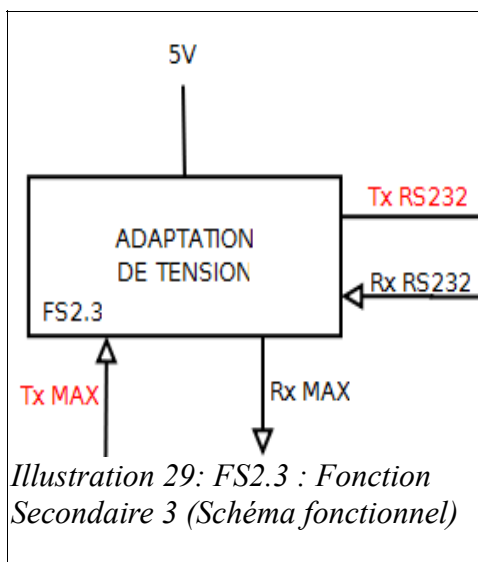
Cette fonction permet l'adaptation du signal transmis au niveau de tension compatible avec le protocole RS232 pour permettre la communication avec l'ordinateur.

La liaison série RS232, fonctionne avec des niveaux de tensions 12 V et -12 V. notre signal étant composé de tension de 0 V et 5 V, il faut donc adapter les tensions du signal au norme de la liaison RS232.

Les niveaux de tensions correspondant sont :

- ◆ niveau logique 1 : -12 V
- ◆ niveau logique 0 : 12 V

Le composant permettant de mettre cette fonction en œuvre est un MAX232, le fonctionnement du MAX232 est tel qu'il crée des niveaux de tension de 12 V sur V+, et un niveau de -12 V sur V-, cela grâce à un système de pompe de charge, d'où l'intérêt des condensateurs branchés sur le composant, permettant d'obtenir les niveaux de tension typique du protocole RS232. La liaison comportant un Rx et un Tx, le composant est capable de passer du 12 V au 5 V, et inversement.



### 4.3.5. FS2.4 : CONNECTEUR RS232 ET USB[8][9]

#### Connecteur USB

Pour connecter l'ordinateur à la carte, il a été choisi d'avoir le choix entre un port RS232 ou USB, ce choix est du au faite que certaines machines n'ont plus de port RS232, mais il est parfois présent, et le matériel à la disposition du technicien peut aussi être limité, bien que le port USB est de plus en plus présent.

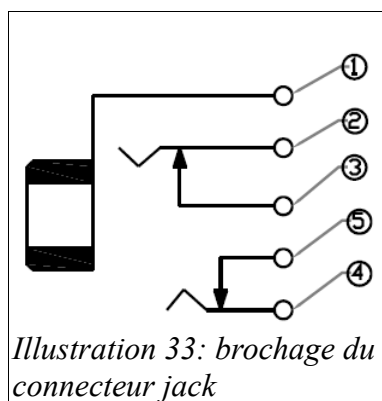
Afin d'avoir un coup plus faible pour la mise au point du système, mon tuteur monsieur Lequeu m'a demandé de mettre en place un connecteur permettant une sortie USB grâce à un câble qu'il m'a fourni, qui est composé d'un coté d'une prise jack, et de l'autre un embout de type USB.



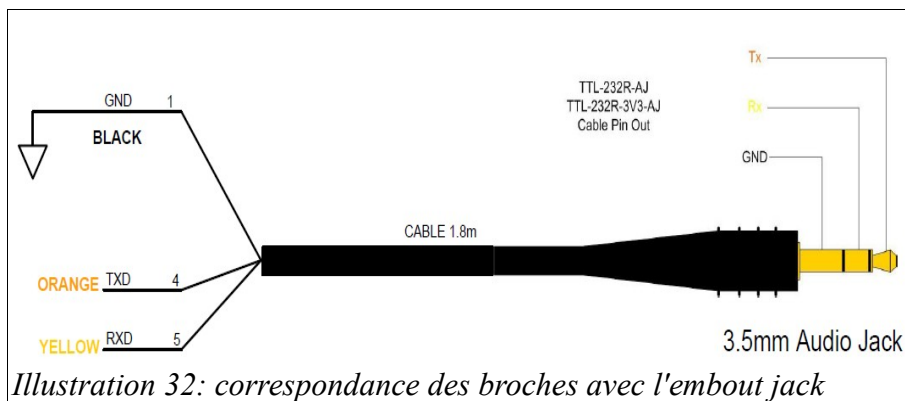
*Illustration 31: Extrémité du câble permettant la liaison USB*

Un autre problème c'est alors présenté, la mise en œuvre d'un connecteur jack. J'ai recherché donc les composants pouvant correspondre à ma recherche, et il est apparu que le critère qui fut décisif est le prix du composant, car n'ayant pas besoin d'une grande qualité, ce composant permet la mise en œuvre de la liaison, ce composant ce nomme Embase stéréo jack femelle 3.5 mm, 3.5 mm parce que c'est la taille de la prise jack du câble.

Après avoir déterminé le composant, il m'a fallu déterminer quel était le brochage approprié, je me suis donc référé aux documentations constructeur du connecteur et du câble.



*Illustration 33: brochage du connecteur jack*



*Illustration 32: correspondance des broches avec l'embout jack*

A partir de ces informations j'ai donc cherché des correspondances entre les broches, et il est apparu que la broche 1 du connecteur jack correspondait à la masse. J'ai à partir de la déduit que la broche 2 correspondais au Rx, puis la broche 3 au Tx.

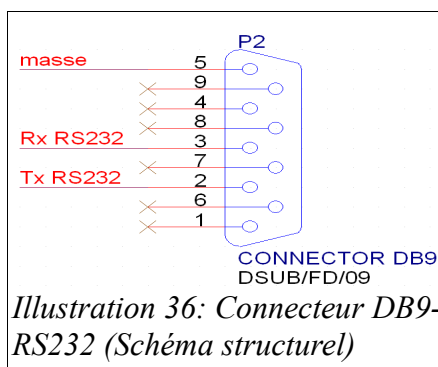
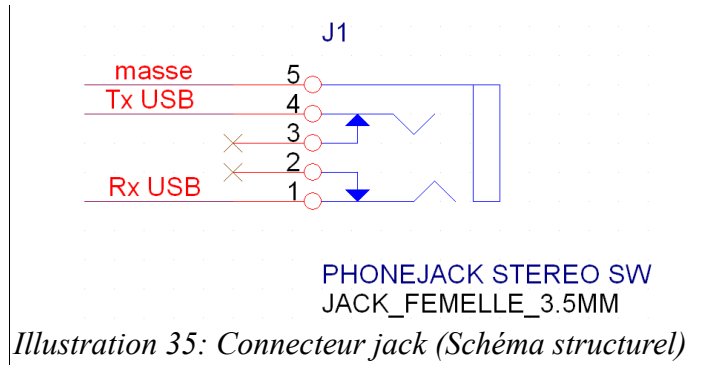
Une fois cela éclairci, un autre problème est apparu, la représentation schématique ne correspondant pas au numéro de broche de la documentation constructeur, j'ai donc du chercher des correspondances.

## Connecteur RS232

La première liaison mise en place, je vais expliquer la seconde possibilité, la liaison RS232.

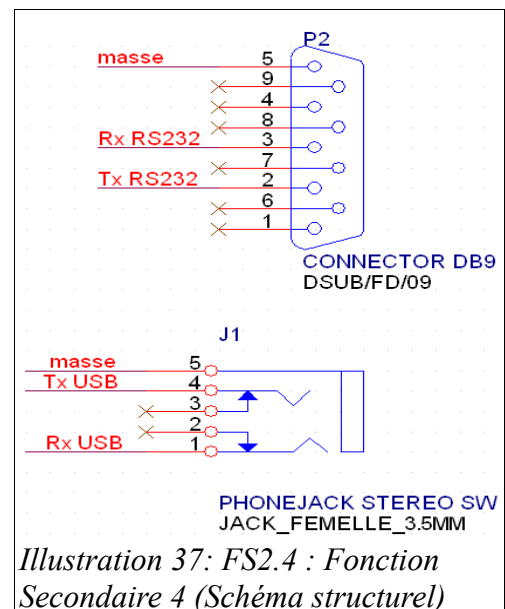
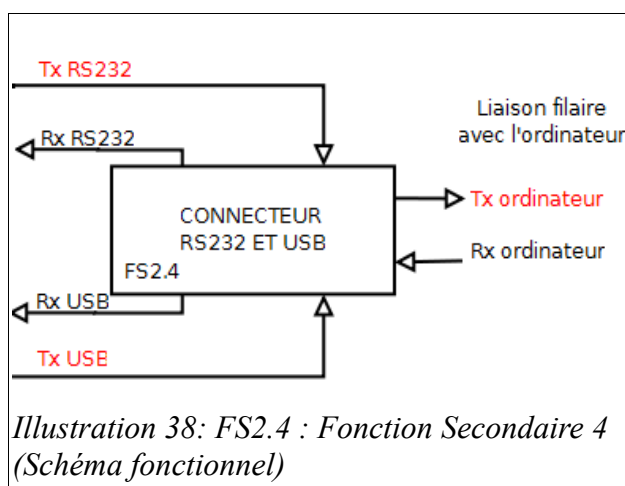
Numérotation des broches		
Doc cons.	Schématique	Rôle
1	5	masse
2	1	Rx
3	2	non reliée
4	4	Tx
5	3	non reliée

*Illustration 34: Correspondance des broches*



Afin de mettre en œuvre le connecteur DB9 femelle, je me suis inspiré des informations disponibles sur le site de Mr Lequeu. Cela m'a donc permis d'économiser du temps.

### FS2.4 : Les 2 connecteurs



Les 2 connecteurs ensemble permettent donc le choix d'une liaison série. Après avoir réalisée tout la partie schématique du système, je passe à la création du typon, c'est à dire le calque permettant la création du circuit imprimé grâce à un logiciel de dessin, ORCAD LAYOUT.

## 5. NOMENCLATURE ET COÛT DE LA CARTE D'ARRIVÉE

Mais avant cette étape il faut générer une netlist<sup>13</sup> (net list : liste des liaisons), et afin de déterminer le coup de la carte, créer une liste du matériel (Bill of materiel) afin de savoir les types de composants choisis, et donc déterminer le prix de l'ensemble des composants du système ainsi que la nomenclature de la carte.

NOMENCLATURE CARTE					
Nbr	Nomination orcad	Désignation	Empreinte	Valeur/ Tension	Coût (€)
1	AFF1	Afficheur	MC1604C	LCD 16x4	7,5 (à l'unité)
1	C1	Condensateur	Radial 08	100µF/63V	0,28 (par 5)
1	C2	Condensateur	Radial06L	470µF/6,3V	0,272 (par 5)
2	C5, C3	Condensateur	Radial06	10µF/6,3V	0,908 x 2 = 1,816 (par 5)
2	C6, C4	Condensateur	CK06	100nF	0,102 x 2 = 0,204 (par 5)
2	C8, C7	Condensateur	CK06	22pF	0,0244 x 2 = 0,0488 (par 25)
5	C9, C10, C11, C12, C13	Condensateur	Radial06L	1µF/25V	0,582 x 5 = 2,91 (par 5)
1	D1	Diode	DO41	1N4007	0,18 (par 5)
1	D2	Diode	DO41	1N5819	0,24 (par 5)
4	D3, D4, D5, D6	Diode	LED03	2mA 3mm (3 couleurs)	0,19 x 2 + 0,15 + 0,19 = 0,56 (par 50)
1	F1	Porte fusible	FUSE	X	0,93 (par 5)
1	J1	Connecteur jack	JACK_FEMELLE_3,5MM	jack femelle 3,5mm	5,58 (par 5)
4	JP1, JP4, JP5, JP6	Connecteur	WEID2_02PL1_2x02PL2	Bornier 2 fils	0,65 x 4 = 2,60 (par 5)
1	JP2	Connecteur de programmation	10SH100L	Embase RJ11	XX
1	JP7	Connecteur	08SH100	HEADER 4x2 (2 cavaliers)	2,254 ( par 5 x 20 tulipes) + 2 x 0,085 (par 100) = 2,424
1	L1	Bobine	RADIAL06L	10µH	0,976 (par 5)
1	L2	Bobine	RADIAL06L	47µH	0,975 (par 5)
1	LS1	Buzzer	02PL2	KPEG220A	2,81 (l'unité)
1	P1	Potentiomètre	RAJ1	10kΩ	0,0432 (par 5)
1	P2	Connecteur DB9	DCSUB/FD/09	connecteur db9	3,20 (à l'unité)
1	Q1	Quartz	HS18UV	16MHz	2,4 (par 10)
2	R2, R1	Résistance	RC04L	1,5kΩ	0,0616 x 2 = 0,1232 (par 5)
1	R3	Résistance	RC04L	10kΩ	0,25 (par 5)
1	R4	Résistance	RC04L	4,7kΩ	0,0636 (par 5)
1	RN1	Réseau de résistances	05PL1	4x4700R(Ω)	1,054 (par 5)
1	SW2	Interrupteur	08DIP300L	switch 4	0,51 (à l'unité)
1	SW3	Interrupteur	RDEROND	bouton poussoir 4 broches	0,75 (par 5)
2	TP1, TP2	Point test	01PR1L	tulipe	XX
1	U1	Régulateur	08DIP300L	LM2574N-5	4,06 (à l'unité)
1	U2	Microcontrôleur	40DIP600L	ATMEGA 8535	6,88 (à l'unité)
1	U3	Module MIFARE	module_mifare	RWD MIFARE	30,59 (à l'unité)
1	U4	Connecteur antenne	conn_ant	connecteur antenne (3 tulipes)	XX
1	U6	Max232	16DIP300L	MAX232	3,50 (à l'unité)
1	U7	Multiplexeur démultiplexeur	16DIP300L	CD4052	0,321 (par 10)
1	X	Antenne	X	ANT 13,56MHz	10,90 (à l'unité)
1	X	carte imprimé	X	160x100	4,96 (à l'unité)
1	X	connecteur afficheur	X	X	XX
1	X	Fusible	X	X	0,66 (par 5)
1	X	support ATMEGA	X	X	0,407 (par 50)
1	X	support MAX232	X	X	0,176 (par 50)
1	X	support multiplexeur	X	X	0,176 (par 50)
1	X	support régulateur	X	X	0,51 (à l'unité)
1	X	support switch	X	X	0,51 (à l'unité)
1	X	support Module	X	support 7 broches+3 tulipes	0,156 (par 50) + 3 tulipes
59					<b>coût total = 102,44€</b>

Tout les composants de cette nomenclature ont été sélectionné sur le site de « Radiospare », les connecteurs n'ayant pas de prix défini peuvent être des tulipes, ainsi celles restantes de certaines utilisations, comme pour le header, seront réutilisées. Le connecteur de programmation n'a pas de prix car celui ci peut être exclu de la version définitive. L'estimation du coût de la carte est donc de 102,44 €, ce qui respecte l'exigence de prix du cahier des charges.

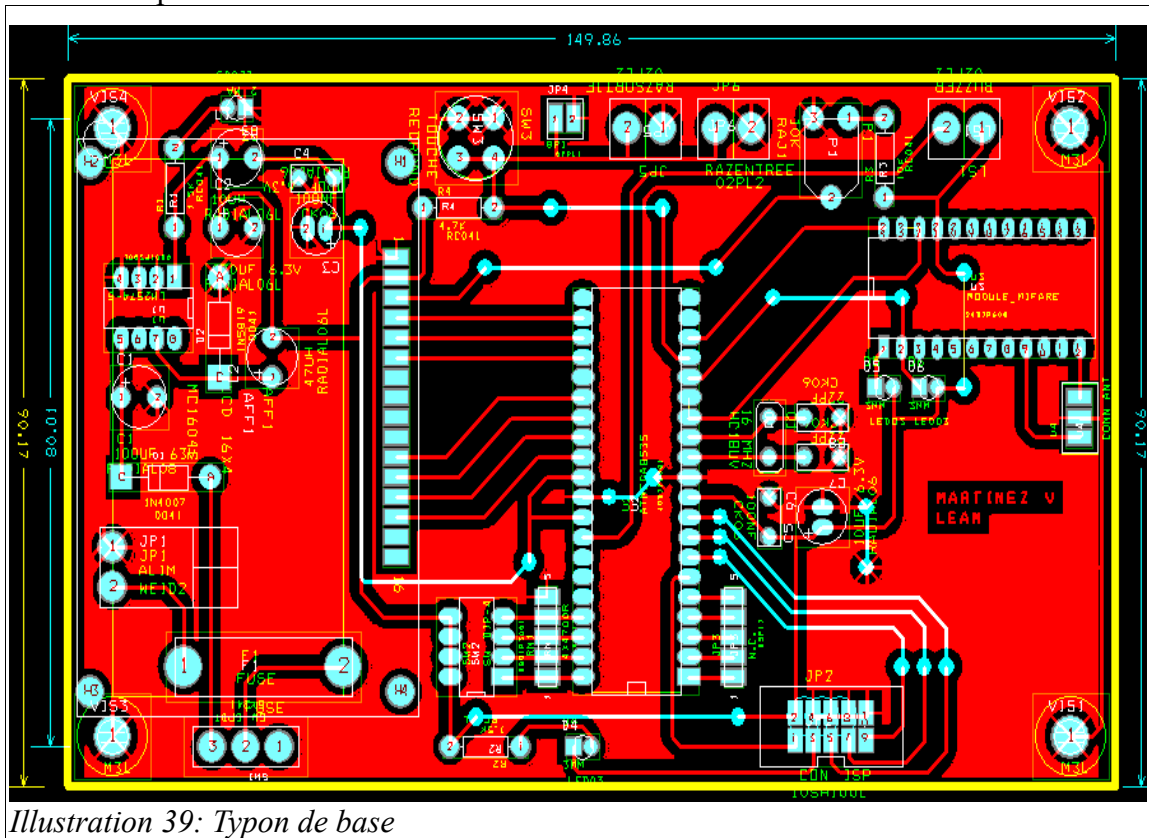
<sup>13</sup> Netlist : génération des différents points du schéma structurel en document texte, lisible par le logiciel ORCAD LAYOUT, permettant de déterminer les liaisons entre composants et leurs empreintes.

## 6. CRÉATION DU TYPON ET DE LA CARTE

La création du typon permet de créer la plaque où sera représenté le circuit électrique du système, cela en dessinant les liaisons électriques entre composants, qui correspondront à la fin de la création de la carte à des pistes cuivrées.

### 6.1. ORCAD LAYOUT

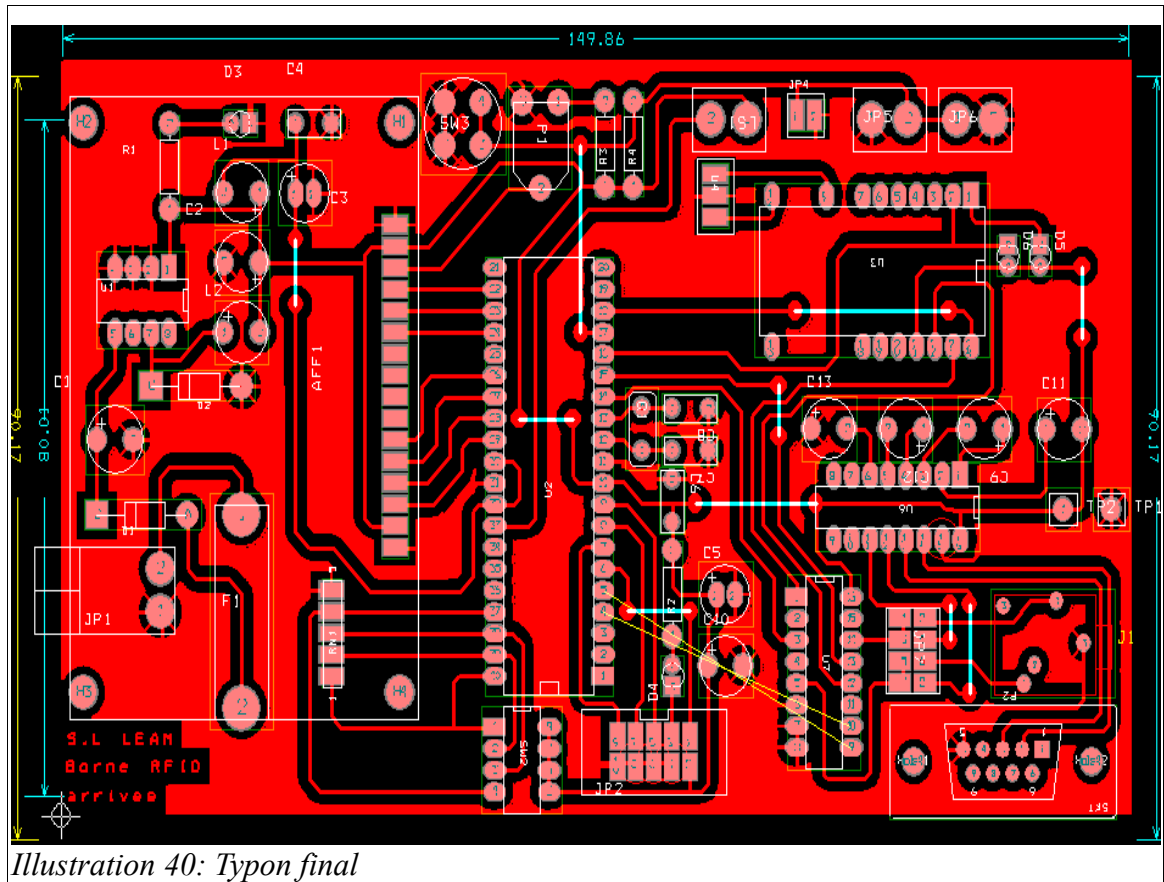
Après la création du schéma électrique du système, il faut associer les empreintes à chaque composant. Une empreinte correspond à la place qu'occupe un composant, ainsi que l'emplacement de ses connexions. De cette manière lors de l'ouverture de la netlist avec le logiciel ORCAD LAYOUT, il est généré les empreintes des composants, ainsi que leurs liaisons correspondantes.



Pour ma part, une partie de ce travail ayant déjà été effectué par mes prédécesseurs, la taille de la carte ayant été déterminé, les composants de la 1<sup>er</sup> fonction principal était déjà disposés. Il m'a alors fallu réorganiser les composants afin de pouvoir intégrer la 2<sup>nd</sup> fonction principale, et cela en conservant la taille de la carte. Cette partie du travail m'a alors pris beaucoup de temps, car certaines parties ne pouvant être totalement changées au niveau de l'organisation, d'autre on du totalement être déplacé afin de gagner de la place, et aussi réduire le nombre de via déjà présent, cela réduisant les risques électriques. Le choix des vias est du au prix de la plaque, ceux ci permettant de substituer une plaque simple face à une double face, la double face étant plus chère que la simple face, et le système n'étant pas influencé par ce choix, le critère prix a été privilégié.

Lors de la réalisation de mon typon je me suis efforcé de diminuer le nombre de vias, ou de diminuer leurs tailles. D'obtenir un maximum de place entre les pistes, afin d'éviter des

problèmes de CEM<sup>14</sup>, et obtenir un routage correcte en évitant les angles droits, les pistes trop longues, les départs de pistes à 45°, et avoir un plan de masse faisant le tour de la carte, et qui en recouvre le plus possible. Ces règles étant celle que j'ai apprise au cours de mes années d'études.



Après une nouvelle répartition et l'intégration de la nouvelle fonction, afin de réaliser toute les liaisons, mon typon est composé de 20 vias et de 2 fils volants, c'est à dire deux liaisons non reliées par une piste de cuivre, mais grâce à un fil pour chaque liaison, qui devront être branchés aux pattes des composants, de cette manière le fil représentatif d'une piste de cuivre passera au dessus des pistes. Les vias eux, sont des pastilles permettant de faire passer un fil d'un point à un autre afin de relier deux pistes.

Durant la mise au point du typon, je me suis rendu compte que certains composants étaient inutiles, et que certaines pistes étaient absentes du schéma d'origine. J'ai donc du corriger cela, ce qui m'a pris encore pas mal de temps, ayant pensé que les données que l'on m'avait transmis été bonne. Cette étape m'a donc fait prendre conscience de la difficulté que représente la prise en main d'un sujet déjà en partie traité.

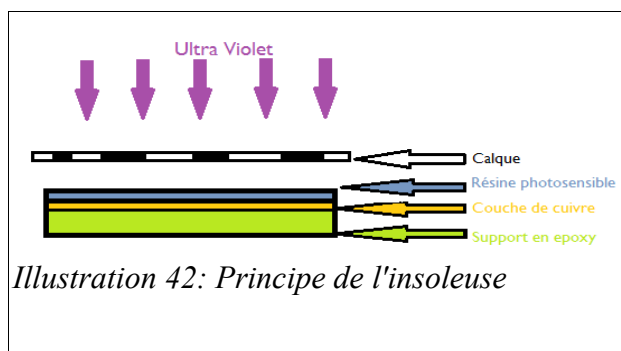
14 CEM : compatibilité électromagnétique



## 7. CRÉATION DE LA CARTE ET SOUDAGE DES COMPOSANTS

La création de la carte (l'époxy), ce fait après avoir imprimé le typon représentant le circuit cuivré sur un papier calque, visible en annexe. Le typon une fois imprimé, est aspergé d'un produit chimique, permettant d'augmenter le contraste entre le noir de l'impression et le transparent du calque. Un calque est utilisé car l'étape suivante est l'insolation.

### 7.1. L'INSOLATION



Cette étape consiste à insérer la plaque d'époxy avec le calque positionné coté encre contre la plaque, dans une machine appelée insolation, qui envoi des rayons Ultra Violet sur la plaque, qui au dessus du cuivre est recouverte d'une résine photosensible. La plaque est dite positive, c'est à dire que c'est la résine insolée sera fragilisée.

## 7.2. RÉVÉLATION

Une fois la résine insolée, il faut transférer la plaque dans une solution appelée révélateur, qui permet d'enlever la résine insolée. De cette façon, la résine restante protégera le cuivre correspondant au typon.



*Illustration 43: Bassine contenant le révélateur*

## 7.3. GRAVURE

Cette étape permet d'attaquer le cuivre situé là où la résine fut insolée, cela grâce à un produit chimique, du perchlore. Lors de cette étape on voit alors le circuit électrique apparaître.



*Illustration 44: Machine permettant la gravure*

## 7.4. NETTOYAGE

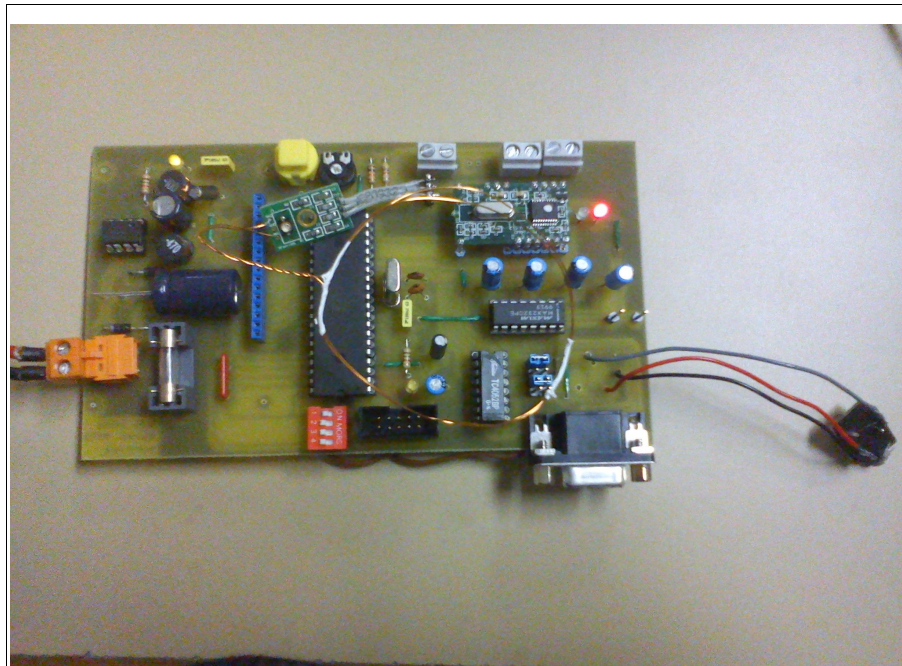
Après la gravure la plaque il reste alors à passer dans l'éliminateur afin d'enlever le cuivre attaqué, puis nettoyer afin d'avoir une plaque non recouverte de produit.

## 7.5. SOUDAGE DES COMPOSANTS

La plaque une fois prête, il faut effectuer un test de continuité des pistes, celui fut sans problème, puis il faut percer les pastilles afin de souder les composants, pour cela il faut mesurer la largeur des pattes et choisir le foret correspondant.

Le perçage fini il faut alors souder les composants, pour éviter d'être gêné par la tailles de certains composants, j'ai commencé par souder les composants les plus proches de la cartes, en 1er les connexions correspondant aux vias, puis les résistances...

Lors du soudage du connecteur jack, j'ai observé une erreur de ma part, qui lors de la création de l'empreinte du composant, je n'ai pas du faire attention au niveau où j'avais créé mes pastilles, pour cette raison le composant c'est retrouvé à l'envers.



*Illustration 45: Carte final après pose des composants*

## **8. PROGRAMMATION ET TEST**

Les composants soudés, j'ai alors vérifié le bon fonctionnement du système en alimentant la carte. Aucun problème ne m'est apparu. Puis pour vérifier le fonctionnement du module RFID, j'ai chargé le programme qu'un groupe de 2<sup>e</sup> année a mis au point en langage C. Le programme ne fonctionna pas du 1<sup>er</sup> coup, il a donc fallu que je recherche le problème, il c'est avéré que les niveaux et statu des broches du microcontrôleur ne correspondais pas à celle dont j'avais besoin avec mes fonctionnalités, le programme modifié est visible en annexe. Une fois cela réglé il reste donc à créer le programme permettant le transfert des informations vers l'ordinateur.

### **8.1. TEST**

Mais avant cela, j'ai effectué quelque test afin de savoir si le CD4052 et les deux ports permettaient bien d'envoyer un signal.

J'ai donc imposé des niveaux sur le microcontrôleur afin de commander le CD4052 pour avoir accès au port. Ce composant fonctionnant parfaitement je me suis intéresser au port jack et RS232.

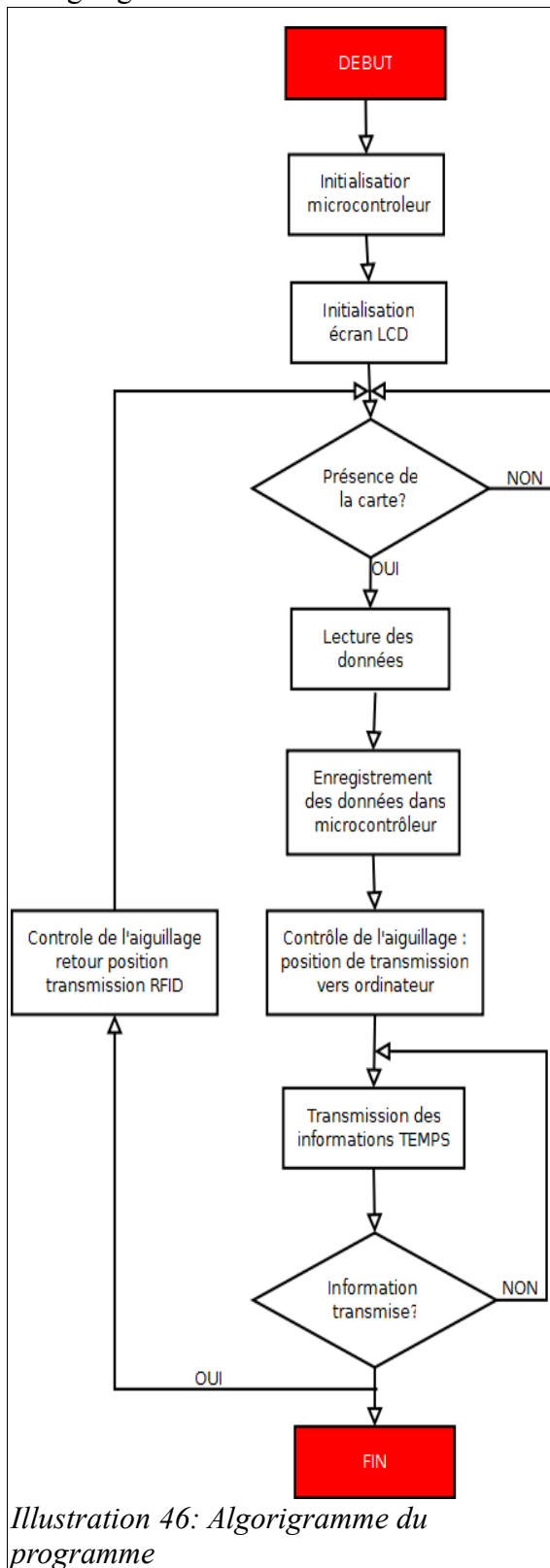
Pour cela j'ai imposé un niveau haut sur la broche Tx du microcontrôleur, de cette manière il me restait à mesurer la tension au Tx des connecteurs pour savoir si le signal pouvait parvenir jusqu'à eux.

En positionnant correctement les cavaliers j'ai donc pu mesurer une tension de 5 V sur le Tx du port jack. Cette valeur correspondant à mes attentes j'ai validé le fonctionnement du port.

J'ai effectué le même procédé pour le port RS232, en plaçant convenablement les cavaliers, et j'ai mesuré une tension de 12 V, convenant à ma plage de tension désiré, j'ai donc pu valider la liaison RS232 et le bon fonctionnement du MAX232.

## 8.2. PROGRAMMATION

Une fois la fonction principale 1 validée, il m'a fallu mettre au point un programme permettant le transfert des données. Pour commencer j'ai d'abord mis au point un algorithme.



Cette partie correspond à l'initialisation du programme, cela afin de remettre chaque variables et les sous programmes à zéro.

Cette condition et déterminer grâce à la broches CTS qui indique selon le niveau la détection ou non d'un badge au microcontrôleur.

La lecture des données permet de lire le temps du coureur, il faut donc enregistrer la valeur dans le microcontrôleur.

Ces deux fonctions permettent l'aiguillage des information, en ayant comme référence le microcontrôleur, elles partent soit vers l'ordinateur, ou le module, le niveau appliqué sur les broches A et B du CD4052 définisse ce choix.

La transmission permet l'envoi de l'information enregistrée, et si celle ci comporte une erreur ou que le transfère c'est mal passé alors l'information est de nouveau transmise jusqu'au bon transfert effectué

Illustration 46: Algorithme du programme

La seconde partie consiste donc à mettre en œuvre l'algorithme, mais mes connaissances limitées avec le logiciel Code Vision AVR et l'ATMega8535, je me suis retrouvé ralenti, ce qui ne m'a pas laissé le temps de mettre au point ce programme. Ainsi donc le projet n'est pas fini mais proche de la fin, j'espère à ma dernière séance de projet tuteuré réussir à mettre au point un programme validant la totalité du système, si cela n'est pas possible il restera ainsi donc le programme à créer.

## CONCLUSION

Mon projet « Borne RFID : Mesure d'un temps de parcours », fut un projet enrichissant et riche en expérience.

J'ai de cette manière pu découvrir une nouvelle technologie : la RFID

Durant ce projet j'ai eu à rechercher des solutions à un problème, comme la fonction d'aiguillage, ou finalement j'ai fini par employer un composant qui m'était inconnu, cela m'a donc appris à ouvrir mes connaissances et ne pas se fixer sur le déjà connu.

J'ai aussi appris durant ce projet, la difficulté que représente la prise en main d'un sujet déjà traité par d'autres personnes. Pour cause, ce critère m'a pris beaucoup de mon temps afin de me documenter et de prendre connaissance du sujet dans sa globalité.

Et puis j'ai aussi été confronté à des problèmes logiciels, qui soit m'étaient inconnus, soit qui me créaient des problèmes me faisant perdre beaucoup de temps.

Ainsi, cette expérience me paraît positive, un système presque fini, mais malgré tout, une mauvaise estimation du temps de travail pour certaines parties, ce qui a pour cause un programme commencé, mais qu'il faudra finir afin d'avoir un système opérationnel.

## RÉSUMÉ

Il m'a été demandé de mettre en œuvre un système de borne RFID permettant de mesurer le temps de parcours d'un coureur. Pour débiter, j'ai présenté l'environnement du projet, puis une introduction à la technologie RFID, cela afin de mieux comprendre le système.

Le sujet ayant déjà été traité précédemment par d'autres étudiants il m'a fallu m'informer de leur avancement, et ainsi m'approprier leurs travaux, que j'ai concentrés en une seule fonction, la fonction principale 1, dont j'ai expliqué chaque fonction secondaire.

Après cela il m'a fallu mettre en œuvre une interface USB et RS232, j'ai donc présenté ma démarche et les solutions envisagées, et enfin la solution retenue. Cette interface est là pour transmettre les informations du badge, porté par le coureur, qui contient ces temps, à un ordinateur, j'ai appelé cette fonction la fonction principale 2. J'ai par la suite présenté chaque fonction secondaire.

Pour cela j'ai effectué différentes recherches et ai retenu le composant essentiel de cette fonction qu'est le CD4052. Après avoir mis au point le système sur ORCAD CAPTURE, j'ai créé la carte électronique en employant le logiciel ORCAD LAYOUT pour créer le typon. Puis j'ai moi même préparé ma carte, j'ai alors expliqué chaque étape effectuée afin de créer la carte.

Et pour finir j'ai effectué des tests de fonctionnement et ai commencé la programmation, mais celle-ci n'est actuellement pas finie.

228 mots

## ABSTRACT

I was asked to implement an RFID terminal system for measuring the route time of a runner. To begin, I presented the project environment and an introduction to RFID technology, to better understand this system.

The subject on which I worked had already begun, I have should my inform the level, that way I became aware of the progress of the project, and have grouped the work already carried out in the main function 1.

After, I had to implement a USB interface and RS232, so I made my approach and alternatives, and finally the solution chosen. This interface is to transmit information of the badge worn by the rider, which contains a time, to a computer, I called this function the main function 2. I then presented each secondary function.

For this, i have effectuated search different, which my allow of find the composant to of do the fonction. Having developed the system on ORCAD CAPTURE, I created the PCB using OrCAD Layout software to create the artwork. Then I prepared myself my card, I then explained each step performed to create the map.

And finally, I have effectuated tests of function and programming, but this one is not currently finished.



## INDEX DES ILLUSTRATIONS

Illustration 1: exemple de système (portique).....	6
Illustration 2: Schéma structurel de niveau 1.....	7
Illustration 3: Principe d'un système RFID.....	8
Illustration 4: badge (tag).....	9
Illustration 5: Partie électronique interne au tag.....	9
Illustration 6: Antenne RFID .....	10
Illustration 7: Module RFID.....	10
Illustration 8: Schéma fonctionnel de niveau 2.....	12
Illustration 9: FS1.1 : Fonction Secondaire 1 (Schéma fonctionnel).....	13
Illustration 10: FS1.1 : Fonction Secondaire 1 (Schéma structurel).....	13
Illustration 11: FS1.2 : Fonction Secondaire 2 (Schéma fonctionnel).....	13
Illustration 12: FS1.2 : Fonction Secondaire 2 (Schéma structurel).....	14
Illustration 13: FS1.3 : Fonction Secondaire 3 (Schéma fonctionnel).....	14
Illustration 14: FS1.3 : Fonction Secondaire 3 (Schéma structurel).....	14
Illustration 15: FS1.4 : Fonction Secondaire 4 (Schéma structurel).....	15
Illustration 16: FS1.4 : Fonction Secondaire 4 (Schéma fonctionnel).....	15
Illustration 17: Solution 1.....	16
Illustration 18: Solution 2.....	17
Illustration 19: Principe de fonctionnement.....	17
Illustration 20: Brochage du CD4052.....	18
Illustration 21: Diagramme logique.....	18
Illustration 22: Table de vérité du CD4052.....	18
Illustration 23: FS2.1 : Fonction Secondaire 1 (Schéma structurel).....	19
Illustration 24: FS2.1 : Fonction Secondaire 1 (Schéma fonctionnel).....	19
Illustration 25: Sélection d'un port.....	19
Illustration 26: FS2.2 : Fonction Secondaire 2 (Schéma structurel).....	19
Illustration 27: FS2.2 : Fonction Secondaire 2 (Schéma fonctionnel).....	19
Illustration 28: Correspondance du signal par rapport à la norme RS232.....	20
Illustration 29: FS2.3 : Fonction Secondaire 3 (Schéma fonctionnel).....	20
Illustration 30: FS2.3 : Fonction Secondaire 3 (Schéma structurel).....	20
Illustration 31: Extrémité du câble permettant la liaison USB.....	21
Illustration 32: correspondance des broches avec l'embout jack.....	21
Illustration 33: brochage du connecteur jack.....	21
Illustration 34: Correspondance des broches.....	22
Illustration 35: Connecteur jack (Schéma structurel).....	22
Illustration 36: Connecteur DB9-RS232 (Schéma structurel).....	22
Illustration 37: FS2.4 : Fonction Secondaire 4 (Schéma structurel).....	22
Illustration 38: FS2.4 : Fonction Secondaire 4 (Schéma fonctionnel).....	22
Illustration 39: Typon de base.....	24
Illustration 40: Typon final.....	25
Illustration 41: L'insoleuse.....	26
Illustration 42: Principe de l'insoleuse.....	26
Illustration 43: Bassine contenant le révélateur.....	26
Illustration 44: Machine permettant la gravure.....	26
Illustration 45: Carte final après pose des composants.....	27
Illustration 46: Algorithme du programme.....	29

## BIBLIOGRAPHIE

- [1] OneToBe, "L'association", dans "Tours'N Aventure" [en ligne] - [http://www.toursnaventure.com/index.php?option=com\\_content&view=article&id=49&Itemid=57](http://www.toursnaventure.com/index.php?option=com_content&view=article&id=49&Itemid=57) - [consulté le 29/10/2010].
- [2] Thierry LEQUEU, "Projet borne RFID 2010", dans "La documentation de Thierry LEQUEU sur OVH" [en ligne] - <http://www.thierry-lequeu.fr/data/Projet-bornes-RFID-2010.pdf> - [consulté le 29/10/2010].
- [3] inrs, "ed4217.pdf (Objet application/pdf)", dans "<http://www.inrs.fr>" [en ligne] - [http://www.inrs.fr/inrs-pub/inrs01.nsf/IntranetObject-accesParReference/ED%204217/\\$File/ed4217.pdf](http://www.inrs.fr/inrs-pub/inrs01.nsf/IntranetObject-accesParReference/ED%204217/$File/ed4217.pdf) - [consulté le 05/11/2010].
- [4] ib technology, "ib technology - RWD modules - RFID Single Chip Reader Solutions - UK", dans "ib technology" [en ligne] - <http://www.ibtechnology.co.uk/rwdmodules.htm> - [consulté le 05/11/2010].
- [5] ib techonlogy, "Antennas", dans "ib technology" [en ligne] - <http://www.ibtechnology.co.uk/antennas.htm> - [consulté le 05/11/2010].
- [6] Thierry LEQUEU, "S. LELERRE, Mesures du temps de parcours par RFID pour une course à pieds ou en vélo, projet Licence EE-EAM, IUT GEII Tours, 11 février 2011", dans "La documentation de Thierry LEQUEU sur OVH" [en ligne] - <http://www.thierry-lequeu.fr/data/DATA437.HTM> - [consulté le 26/11/2010].
- [7] Fairchild, "CD4052 datasheet", dans "Datasheetcatalog.com" [en ligne] - [http://www.datasheetcatalog.org/datasheets/150/109160\\_DS.pdf](http://www.datasheetcatalog.org/datasheets/150/109160_DS.pdf) - [consulté le 26/11/2010].
- [8] FTDI Chip, "DS\_TTL-232R\_CABLES.pdf", dans "FTDI Chip" [en ligne] - [http://www.ftdichip.com/Support/Documents/DataSheets/Cables/DS\\_TTL-232R\\_CABLES.pdf](http://www.ftdichip.com/Support/Documents/DataSheets/Cables/DS_TTL-232R_CABLES.pdf) - [consulté le 16/12/2010].
- [9] DECELECT-FORGOS, "EMBAS STEREO CI DIAMETRE 3,5 MM", dans "Radiospare" [en ligne] - <http://docs-europe.electrocomponents.com/webdocs/00f8/0900766b800f82a2.pdf> - [consulté le 16/12/2010].

# ANNEXE

## Programme de réception :

/\*\*\*\*\*\*

This program was produced by the

CodeWizardAVR V1.24.2c Professional

Automatic Program Generator

© Copyright 1998-2004 Pavel Haiduc, HP InfoTech s.r.l.

<http://www.hpinfotech.ro>

e-mail:office@hpinfotech.ro

Project :

Version :

Date : 12/11/2010

Author : F4CG

Company : F4CG

Comments:

Chip type : ATmega8535

Program type : Application

Clock frequency : 16,000000 MHz

Memory model : Small

External SRAM size : 0

Data Stack size : 128

\*\*\*\*\*/

```
#include <mega8535.h>
```

```
// Alphanumeric LCD Module functions
```

```
#asm
```

```
.equ __lcd_port=0x15 ;PORTC
```

```
#endasm
```

```
#include <lcd.h>
```

```
#include <delay.h> // pour la fonction delay_ms();
```

```
#include <stdlib.h>
```

```
// Standard Input/Output functions
```

```

#include <stdio.h>

#define CTS    PIND.4
#define BUZZER PINA.7

// Declare your global variables here

unsigned char trame, Statut, Information;
unsigned char Temps[20];
unsigned char tampon[20], Identifiant[10];

void USART_Transmit( unsigned char data )
{
    /* Wait for empty transmit buffer */
    while ( !(UCSRA & (0x20)) ); // Test de UDRE bit 5
    /* Put data into buffer, sends the data */
    UDR = data;
}

unsigned char USART_Receive( void )
{
    /* Wait for data to be received */
    while ( !(UCSRA & 0x80) ); // Test de RXC bit7

    /* Get and return received data from buffer */
    return UDR;
}

int VerifStatut (void)
{
    int Presence;

    while(CTS==1); //Attente que l'entrée CTS soit active (à 0)
    trame='S'; //ou 0x53 : permet de vérifier le statut
    USART_Transmit(trame);
    Statut = USART_Receive();

    if(Statut==0x96) //Bit d'identification de la carte
    {
        Presence=1;
    }
}

```

```

    sprintf(tampon,"Lecture ");
    lcd_gotoxy(0,0);
    lcd_puts(tampon);
}
else
{
    Presence=0;
    sprintf(tampon,"Pas Carte");
    lcd_gotoxy(0,0);
    lcd_puts(tampon);
}
return Presence;
}

```

```

void Recoit_UID (void)

```

```

{
    int i, z;

    unsigned char car;
    i=0;
    for(z=0;z<7;z++)
    {
        Identifiant[z] = 0;
    }
    while(CTS==1);
    trame='U', //ou 0x55 : permet de lire l'UID
    USART_Transmit(trame);
    Statut = USART_Receive();

    if(Statut == 150)
    {
        while(i<=6) // 6 pour les MIFARE 1k/4k card types
        {
            car = USART_Receive();
            Identifiant[i]=car;
            i++;
        }
    }
}

```

```

}

void Recoit_Info (int Borne)
{
    int i;

    unsigned char car;
    i=0;
    while(CTS==1);

    trame='R'; //permet de lire la trame
    USART_Transmit(trame); //Transmission de 'R' ou 0x52
    USART_Transmit(Borne*18); //Transmission de l'adresse de location
    USART_Transmit(0x00); //Transmission du code de lecture
    Information = USART_Receive();

    Information = (Information & 0xCF); //Mise en place d'un masque pour lecture nécessaire du Acknowledge

    if(Information == 0x86)
    {
        //Temps[1] = USART_Receive();

        while(i<=15) //16 bits à récupérer
        {
            car = USART_Receive();
            Temps[i] = car;
            i++;
        }
    }

    lcd_clear();
}

void main(void)
{
    // Declare your local variables here
    // Input/Output Ports initialization
    // Port A initialization

```

```

// Func7=Out Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=1 State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTA=0x80;
DDRA=0x80;

// Port B initialization
// Func7=In Func6=In Func5=In Func4=Out Func3=Out Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=0 State3=0 State2=T State1=T State0=T
PORTB=0x00;
DDRB=0x18;

// Port C initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTC=0x00;
DDRC=0xFF;

// Port D initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=Out Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=1 State0=T
PORTD=0x02;
DDRD=0x02;

// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: Timer 0 Stopped
// Mode: Normal top=FFh
// OC0 output: Disconnected
TCCR0=0x00;
TCNT0=0x00;
OCR0=0x00;

// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: Timer 1 Stopped
// Mode: Normal top=FFFFh
// OC1A output: Discon.
// OC1B output: Discon.
// Noise Canceler: Off

```

```
// Input Capture on Falling Edge
TCCR1A=0x00;
TCCR1B=0x00;
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;

// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: Timer 2 Stopped
// Mode: Normal top=FFh
// OC2 output: Disconnected
ASSR=0x00;
TCCR2=0x00;
TCNT2=0x00;
OCR2=0x00;

// External Interrupt(s) initialization
// INT0: Off
// INT1: Off
// INT2: Off
MCUCR=0x00;
MCUCSR=0x00;

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x00;

// USART initialization
// Communication Parameters: 8 Data, 1 Stop, No Parity
// USART Receiver: On
// USART Transmitter: On
// USART Mode: Asynchronous
// USART Baud rate: 9600
UCSRA=0x00;
```



```

UCSRB=0x18;
UCSRC=0x86;
UBRRH=0x00;
UBRRL=0x67;

// Analog Comparator initialization
// Analog Comparator: Off
// Analog Comparator Input Capture by Timer/Counter 1: Off
// Analog Comparator Output: Off
ACSR=0x80;
SFIOR=0x00;

// LCD module initialization
lcd_init(16);

while (1)
{
    unsigned char NumeroBorneHexa;
    int z, PresenceCarte;
    PresenceCarte = VerifStatut (); // Appelle de la fonction permettant de vérifier le statut

    NumeroBorneHexa = (PINA&0x0F);

    sprintf(tampon, "%x", NumeroBorneHexa);
    lcd_gotoxy(12, 0);
    lcd_puts(tampon);
    if(PresenceCarte==1)
    {
        // Possibilité de lire l'identifiant unique de la carte

        Recoit_UID();

        for(z=0;z<4;z++)
        {
            sprintf(tampon,"%3d",Identifiant[z]);
            lcd_gotoxy(z*4,1);
            lcd_puts(tampon);
        }
        Recoit_Info(NumeroBorneHexa);
    }
}

```

masque)

```
sprintf(tampon, "0x%x", Information); //Affichage du Acknowledge de reception des 16 bits (avec
```

```
lcd_gotoxy(0,3);
```

```
lcd_puts(tampon);
```

```
for(z=0;z<4;z++) //Lecture complète des 16 bits affiché sur 4 lignes
```

```
{
```

```
/*sprintf(tampon, "%3d", Temps[z]); //4 premiers bits = UID
```

```
lcd_gotoxy(z*4, 1);
```

```
lcd_puts(tampon); */
```

```
sprintf(tampon, "%3d", Temps[z+4]); //bits qui nous interesse
```

```
lcd_gotoxy((z)*4, 2);
```

```
lcd_puts(tampon);
```

```
sprintf(tampon, "%3d", Temps[z+8]);
```

```
lcd_gotoxy((z)*4, 3);
```

```
lcd_puts(tampon);
```

```
/*
```

```
sprintf(tampon, "%3d", Temps[z+12]);
```

```
lcd_gotoxy((z)*4, 3);
```

```
lcd_puts(tampon);*/
```

```
}
```

```
}
```

```
}
```

```
}
```

Calque :

