

Projet Tutoré – Étude et Réalisation

Mesure de temps de parcours par RFID





Projet Tutoré – Étude et Réalisation

Mesure de temps de parcours par RFID

Sommaire

Introduction.....	4
1. Présentation du sujet.....	5
1.1. Cahier des charges	5
1.2. Qu'est-ce que la RFID?.....	6
1.3. Planning	7
2. Études du système.....	9
2.1. Solutions envisagées	9
2.2. Choix des composants.....	12
2.3. Les principaux composants	14
3. Réalisation.....	17
3.1. Programmation	17
3.2. Routage sur Orcad	22
3.3. Problèmes rencontrés	29
3.4. Ce qu'il reste à faire.....	30
Conclusion.....	31
Résumé.....	32
Index des illustrations.....	33
Bibliographie.....	35
Annexes.....	36

Introduction

Au cours du 3ème semestre, nous sommes conviés à effectuer un projet en relation avec les matières du GEII. Nous avons choisi de poursuivre le travail débuté par Vivien MARTINEZ[1], un étudiant de l'IUT GEII, actuellement en licence.

Celui-ci consiste à calculer des temps de parcours, lors d'une course à pied, grâce à des bornes sans fil et autonomes. Nous avons trouvé l'idée intéressante à mettre en place car elle regroupe à la fois l'électronique et l'informatique. Vivien était au stade de la réalisation d'un prototype, mais il n'avait pas terminé sa carte électronique. Notre but a été dans un premier temps, de finir ce qui a été commencé. Puis de modifier le prototype pour pouvoir le programmer et créer plusieurs versions finales.

Comment avons-nous procédé pour mettre en place ce dispositif ? C'est ce que nous verrons en commençant par une présentation détaillée du projet. Nous étudierons ensuite les solutions technologiques mises en place pour créer le dispositif. Enfin, la dernière partie sera consacrée à la réalisation et à la programmation des bornes.

1. Présentation du sujet

Préalablement conçu pour des raids ou des cross, le chronométrage sans fil est généralement loué pour des manifestations sportives regroupant de nombreux concurrents. Lors d'une course à pied par exemple, les participants se verront remettre avant le départ, un badge personnel. En passant à travers des portiques disséminés sur le parcours, le temps de passage sera automatiquement inscrit dans les badges. Il suffira de lire les données des badges à la fin de la course pour déterminer le temps de parcours de chacun.



Illustration 1: Portique disponible à la location[2]

1.1. Cahier des charges

Ces dispositifs coûtent très cher, c'est pourquoi l'association Tours'n Aventure avait demandé à Vivien MARTINEZ[1] de refaire de toute pièce, les boîtiers de mesure. Notre rôle est de terminer ce projet. Le principe sera quelque peu simplifié. Les portiques sont remplacés par des bornes.

Chaque concurrent devra présenter son badge devant chacune d'entre elle. Avant la manifestation, une personne de l'organisation doit affecter une fonction à chaque boîtier (borne de départ, d'arrivée, intermédiaire) en attribuant un numéro et en synchronisant l'horloge interne.

Pour permettre une intervention rapide et sans difficulté de l'opérateur, les boîtiers devront s'ouvrir facilement et l'accès aux boutons de réglage doit être aisé. L'affichage du numéro de borne et de l'horloge sera placé en façade. Chaque

boîtier devra être facilement transportable et stable. De plus, pour éviter les court-circuits, les bornes devront être étanches et solides.

L'autonomie électrique de chaque borne se fera grâce à une batterie qui doit assurer le fonctionnement pendant une centaine d'heures. Le boîtier a été choisi et possède les dimensions suivante : 160x98x68 mm. Chaque transmission se fera par technologie RFID¹.



Illustration 2: Le boîtier[3]

L'association à qui bénéficie le projet, a décidé que le budget total doit être inférieur à 400€ pour l'ensemble des bornes et de 200€ pour les badges individuels.

1.2. Qu'est-ce que la RFID?

La RFID est une technologie analogue au code à barre. Contrairement à celui-ci, l'étiquette RFID, aussi appelée TAG, n'a pas besoin de lecteur optique pour être lu. Elle contient un système électronique noyé dans du plastique sous forme de carte de crédit, de pièce ou de badge. Elle n'a pas besoin d'une source d'alimentation externe. La lecture se fait donc à distance pouvant aller d'une dizaine de centimètres à quelques mètres. Un module distant, composé d'un circuit imprimé et d'une antenne, interroge le TAG qui envoie ensuite ses données. Un TAG peut être incorporé à un produit ou bien porté sur soi. Grâce à son numéro d'identification unique, il est nominatif. Cette technologie est

¹ Radio Frequency IDentification (Identification par Radio-Fréquence)

principalement utilisée pour stocker et récupérer des données à distance. De plus, il existe une procédure de sécurité pour certains modèles.

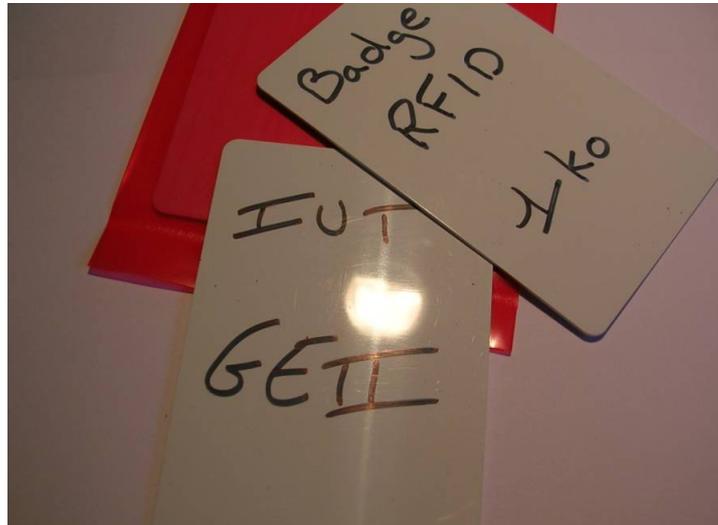


Illustration 3: Des TAGs (ou badge RFID)[3]

La technologie RFID trouve son utilisation dans de nombreux domaines, dans lesquels il est nécessaire de suivre un produit à distance ou d'authentifier une personne : industries, portes d'immeubles, protections antivols dans les grands magasins...

1.3. Planning

Pour répondre au cahier des charges, nous avons pour l'instant bénéficié de onze séances de projet et d'une séance libre. Un cours, a été consacré à la formation au logiciel Orcad qui permet, de réaliser schéma électrique et typon. De plus, le travail à la maison fut primordial.

Lors de la conception du planning prévisionnel, nous ne nous sommes pas rendu compte que le projet possédait déjà un prototype et qu'il existait une partie programmation. C'est pourquoi le planning final n'est pas en adéquation avec le prévisionnel. De plus, nous nous sommes heurtés à plusieurs problèmes techniques qui nous ont retardés.

La partie programmation aura été la plus longue et la plus complexe : elle a occupé sept séances. Le rapport des semaines figure en annexe 4.

	8	9	10	11	12	13	14	15	16	17	18	19	20	20	21	23	23	24
Prise de connaissance du sujet	Green	Green	Green	Green			Grey	Grey										
Études des différentes solutions techniques			Green	Green	Green		Grey	Grey										
Choix des solutions techniques					Green		Grey	Grey										
Étude du système					Green	Green	Grey	Grey	Green									
Mise en œuvre			Black				Grey	Grey	Green	Green	Green							
Réalisation du système				Yellow	Yellow		Grey	Grey					Green				Yellow	Yellow
Programmation						Yellow			Yellow									
Mesures et tests							Grey	Grey					Green	Green	Green			Yellow
Rédaction du rapport							Grey	Grey						Yellow	Yellow		Green	Green
Présentation Orale							Grey	Grey										Green

Green	Planning prévisionnel
Yellow	Planning réel
Blue	Séance libre
Black	Apprentissage Orcad

Tableau 1: Planning prévisionnel et final[4]

2. Études du système

2.1. Solutions envisagées

Vivien MARTINEZ avait envisagé de faire une carte, qui pouvait recevoir deux types de modules RFID. Le premier (à gauche sur l'image) ayant une fréquence de 125KHz et le second de 13,56 MHz (à droite). En effet, plus on augmente la fréquence, plus le TAG peut être lu à une distance éloignée. Or, nous nous sommes aperçus que la distance après plusieurs tests, avec le module longue portée, n'était que d'une dizaine de centimètres. Il n'a donc pas été question de prendre le module de 125 KHz. C'est pourquoi notre carte a été faite pour recevoir seulement le module haute fréquence (13,56 MHz).

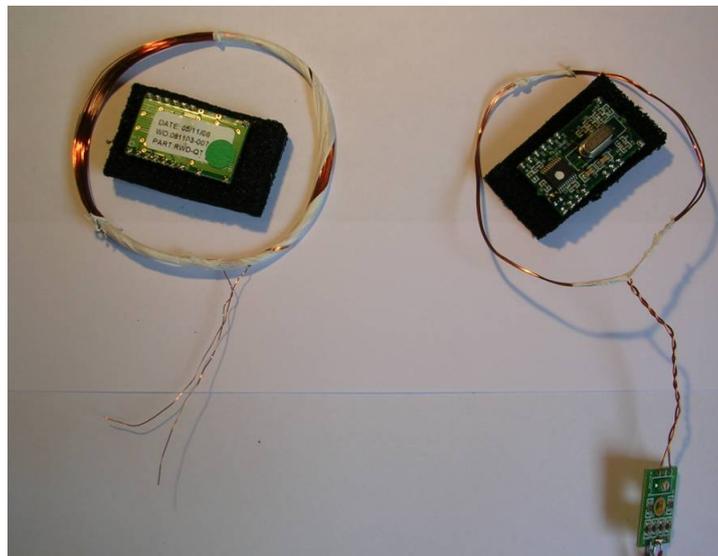


Illustration 4: Les Modules RFID et leurs antennes[3]

Par contre, le choix du micro-contrôleur ATMEGA 8535 et du régulateur de tension LM2574 a été conservé.

Voici les schémas fonctionnels du projet :

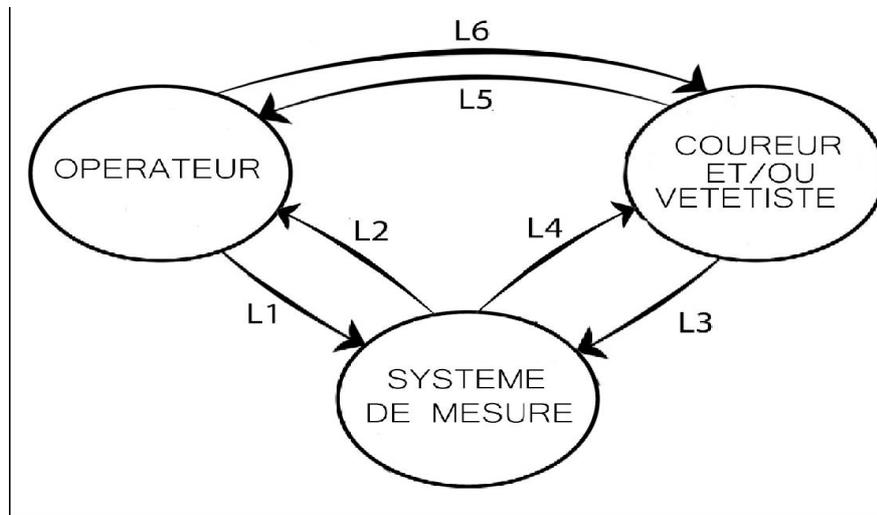


Illustration 5: Schéma global du projet[1]

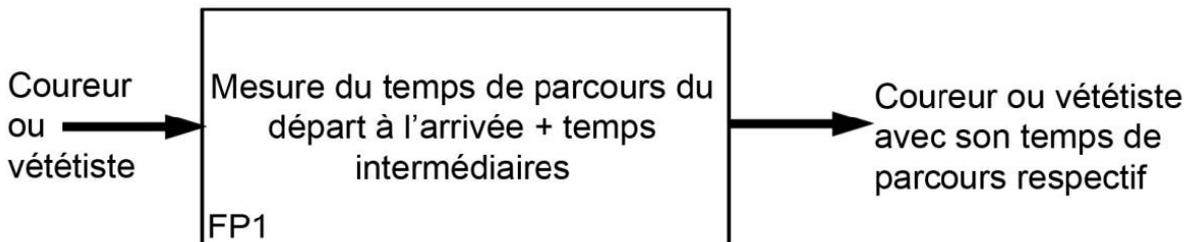


Illustration 6: Schéma fonctionnel de niveau 1[1]

On mesure donc le temps de parcours du coureur. Chaque badge va être analysé par les bornes qui vont ensuite écrire le temps de passage.

Fp1 :

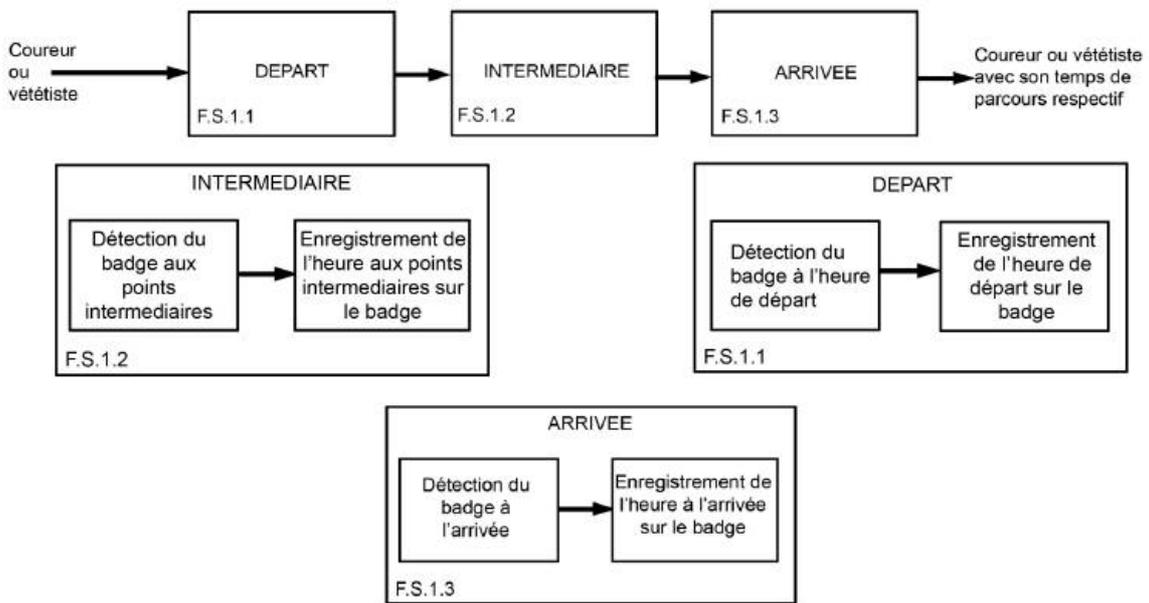


Illustration 7: Schéma fonctionnel de niveau 2[1]

Dans un premier temps, un opérateur va affecter à chaque borne de chronométrage un numéro différent (grâce au switch, en rouge sur l'image), pour ensuite les relier grâce à un fil et synchroniser leurs horloges.

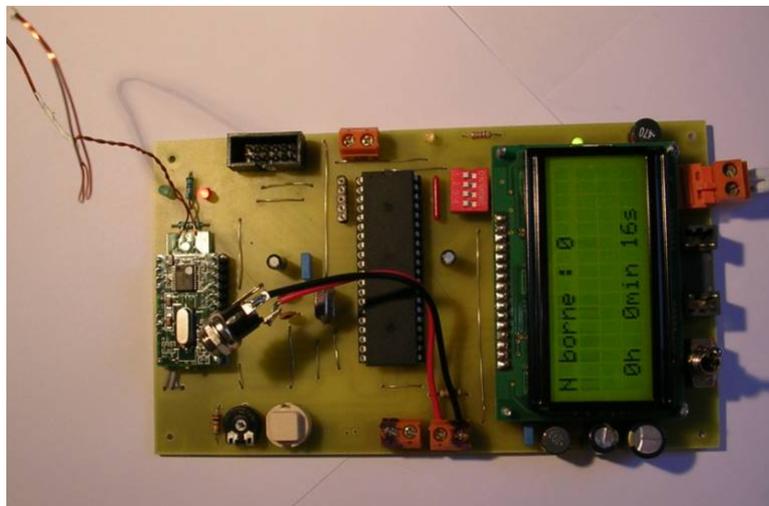


Illustration 8: La carte finale[3]

Ensuite, les responsables de la manifestation vont affecter à chaque participant leur badge (ou TAG). Grâce à son identifiant unique, il va être nominatif et permettra de faire en sorte que chaque personne ait son numéro comme avec un dossard.

La course commence! Le coureur passe par la borne de départ. Son badge est détecté et le temps de passage y est inscrit. S'ajoutent ensuite les temps de passage de chaque borne intermédiaire. Puis l'heure d'arrivée lorsque le sportif passe par le dernier point de passage.

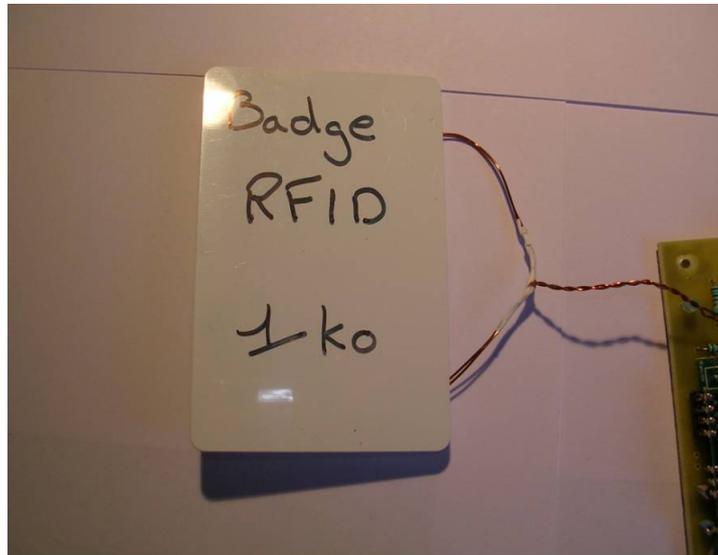


Illustration 9: Présentation du TAG devant l'antenne[3]

Enfin chaque participant remet aux responsables, son badge qui va être décodé par une borne spécifique à la lecture. Les informations sont alors stockées dans une base de données et le classement est établi.

2.2. Choix des composants

Pour respecter le budget, nous avons pris de préférence les composants les moins chers. Voici leur nomenclature et le prix de chacun.

Fournisseur	Nature du composant	Référence	Prix unitaire (HT)	Quantité	Prix total (HT)	Notes
Radiospares	Micro-contrôleur	ATMEGA 8535	6,17	1	6,17	
Radiospares	Régulateur de tension	LM2574N-5	2,83	1	2,83	
Radiospares	Module RFID	RWD MIFARE	38,75	1	38,75	
Radiospares	Antenne RFID	ANT 1356M	14,23	1	14,23	
Radiospares	Afficheur LCD	16x4 caractères	7,5	1	7,5	
Radiospares	Condensateur	100 uF 63V	0,28	1	0,28	Vendu par sachet de 5
Radiospares	Condensateur	470 uF 6,3V	0,272	1	0,272	Vendu par sachet de 5
Radiospares	Condensateur	10 uF 6,3V	0,908	2	1,816	Vendu par sachet de 5
Radiospares	Condensateur	100 nF	0,102	2	0,204	Vendu par sachet de 5
Radiospares	Condensateur	22 pF	0,0244	2	0,0488	Vendu par sachet de 25
Radiospares	Diode	1N4007	0,18	1	0,18	Vendu par sachet de 5
Radiospares	Diode	1N5819	0,24	1	0,24	Vendu par sachet de 5
Radiospares	DEL	2ma 3mm Rouge	0,156	1	0,156	Vendu par sachet de 50
Radiospares	DEL	2ma 3mm Orange	0,075	1	0,075	Vendu par sachet de 50
Radiospares	DEL	2ma 3mm Verte	0,166	2	0,332	Vendu par sachet de 50
	Batterie		15	1	15	Pas encore commandé
Radiospares	Buzzer	KPEG220A	2,81	1	2,81	
Radiospares	Fusible		0,66	1	0,66	Vendu par sachet de 5
Radiospares	Porte-Fusible		0,93	1	0,93	Vendu par sachet de 5
Radiospares	Switch	Glissière 4 voies	1,37	1	1,37	
Radiospares	Quartz	16Mhz	2,4	1	2,4	Vendu par 10
Radiospares	Inductance	10 uH	0,976	1	0,976	Vendu par sachet de 5
Radiospares	Inductance	47 uH	0,975	1	0,975	Vendu par sachet de 5
Radiospares	Bornier	2 fils	0,65	3	1,95	Vendu par sachet de 5
Radiospares	Bouton Poussoir	4 broches	0,75	1	0,75	Vendu par sachet de 5
Radiospares	Potentiomètre	10k	0,0432	1	0,0432	Vendu par sachet de 5

Radiospares	Résistance	1,5k	0,0616	1	0,0616	Vendu par sachet de 5
Radiospares	Résistance	4,7k	0,0636	2	0,1272	Vendu par sachet de 5
Radiospares	Résistance	10k	0,25	2	0,5	Vendu par sachet de 5
Radiospares	Réseau résistance	4x4,7k	1,054	1	1,054	Vendu par sachet de 5
Radiospares	Interrupteur	2 positions	1,21	1	1,21	
Radiospares	Support ATMEGA	Tulipe	4,12	1	4,12	
Radiospares	Support LM et Switch	Tulipe	0,91	2	1,82	
Radiospares	Plaque cuivrée	160x100	3,24	1	3,24	
	Boitier	160x98x68 mm				
			Total (HT)	44	113,08 €	

Tableau 2: Nomenclature et coût de fabrication d'une borne[4]

Le prix d'une borne est donc d'environ 135€ TTC. Il faut savoir que plus les composants sont commandés en quantité, plus le coût diminue. C'est pourquoi, construire les bornes à l'IUT a été moins onéreux.

2.3. Les principaux composants

2.3.1. Le micro-contrôleur ATMEGA 8535

L'ATMEGA 8535 est un micro-contrôleur très utilisé à l'IUT. Notre choix s'est tourné vers cette référence, car elle est disponible en quantité au magasin. Il n'y a donc pas eu besoin de créer une commande.

Ce composant permet après programmation de gérer des entrées/sorties en autonomie. C'est le cerveau de la borne. Il fait non seulement le lien entre le module RFID et sa mémoire, mais il écrit aussi sur l'afficheur. Pour fonctionner, il a besoin d'un montage annexe composé d'un quartz et de condensateurs. Celui-ci va permettre de donner une vitesse de calcul comme dans le cas d'un ordinateur.

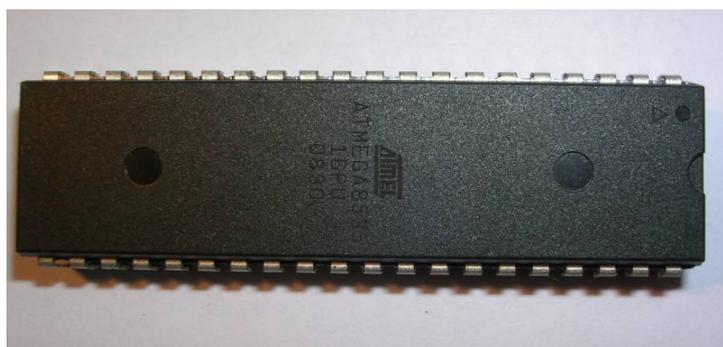


Illustration 10: Le micro-contrôleur ATMEGA 8535[3]

Pour programmer l'ATMEGA 8535, on utilise le logiciel CodeVision AVR développé par la société Hewlett Packard. La programmation se fait en C, en utilisant les fonctions basiques de ce langage.

2.3.2. Le régulateur de tension LM2574N-5

Le LM2574N-5 est un régulateur de tension, capable de réguler une tension d'entrée comprise entre 7 et 40 volts continu. Il va faire en sorte d'obtenir 5 volts en sortie en toutes circonstances. C'est pourquoi lors des essais, le montage a été alimenté avec une alimentation continue de 12 volts et la version autonome sera alimentée avec une batterie de 9 volts. Le bon fonctionnement du régulateur est indiqué par la led verte positionnée à côté de l'afficheur.



Illustration 11: Le régulateur de tension LM2574N-5[3]

2.3.3. Le module RFID RWD² MIFARE

Comme vu précédemment, le module RFID assure la liaison entre le TAG et le micro-contrôleur par le biais de son antenne. Sa fréquence est de 13,56 Mhz. Il dispose d'un programme interne qui gère des trames de données. Ainsi l'ATMEGA

2 Read / Write Device : Dispositif de lecture et d'écriture

peut par exemple envoyer une demande de lecture du code d'identification unique par liaison série RS232.

L'information arrive au module qui avec un autre langage, va demander des informations sur l'UID³ au TAG pour les réceptionner et les renvoyer au micro-contrôleur.

Il faut faire attention de bien gérer la sortie CTS « Ready to Receive ». Celle-ci envoie une impulsion qui montre que le module est prêt à réceptionner les informations. Dans le cas contraire, les données sont perdues et le module ne fonctionne pas.

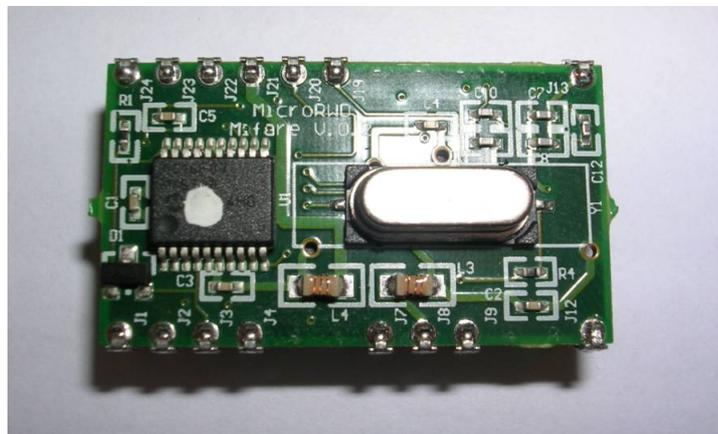


Illustration 12: Le module CMS RFID 13,56 MHz[3]

La détection de la carte est gérée par deux voyants : une led verte et une rouge. Si le TAG est bien reconnu et qu'aucune erreur n'est détectée lors de la transmission de la trame, la led verte s'allume. Sinon, la led rouge persiste.

La datasheet de ce module, et son implantation, est disponible en annexe 1 et 2.

2.3.4. L'afficheur 16x4 caractères

L'afficheur choisi dispose de 16 caractères sur 4 lignes. Il affiche grâce aux données transmises par l'ATMEGA des informations pratiques pour l'utilisateur.

On va alors faire figurer, dans un premier temps le numéro de la borne, puis le statut de la carte et enfin l'horloge interne du dispositif. La largeur de l'écran permet d'afficher sur une même ligne, le temps avec ses heures, ses

³ UID : Unique Identifier : Identifiant Unique

minutes, ses secondes. L'écran est rétro-éclairé ce qui permet un meilleur confort lorsqu'il fait sombre ou même nuit.



Illustration 13: Afficheur large écran[3]

3. Réalisation

3.1. Programmation

La programmation de l'ATMEGA 8535 est faite à partir de CodeVision AVR, un logiciel de programmation en C payant distribué par Hewlett Packard.

L'intégralité du programme est disponible en annexe 3. La partie écriture dans les TAGs n'a pas encore été testée car des erreurs sur le module RFID de la carte finale subsiste.

3.1.1. L'affichage du numéro de borne

On déclare dans un premier temps les variables permettant de gérer l'affichage du numéro de borne. Ces variables sont définies en « variables locales » c'est à dire qu'elles ne peuvent être utilisées seulement dans la fonction où elles sont utilisées.

```
unsigned char NumeroBorneHexa;
```

```
int Dizaine, Unite, NumeroBorne;
```

On récupère ensuite les informations envoyées par l'interrupteur à 4 positions. Il a fallu créer un masque pour ne lire que les entrées intéressante.

```
NumeroBorneHexa = (PINA&0x0f);
```

On procède ensuite à la conversion du numéro de borne en hexadécimal.

```
Dizaine=NumeroBorneHexa/10;  
NumeroBorne=Dizaine<<4;  
Unite=((int)NumeroBorneHexa)%10;  
NumeroBorne=NumeroBorne+Unite;
```

Enfin, on écrit ce numéro de borne en hexadécimal sur la première ligne de l'afficheur.

```
sprintf(tampon,"N borne : %2d",NumeroBorne);  
lcd_gotoxy(0,0);  
lcd_puts(tampon);
```

On peut alors écrire sur l'afficheur une valeur de 1 à 9 permettant donc de différencier 9 bornes.

3.1.2. La mise en place de l'horloge interne

Pour mettre en place l'horloge, il faut configurer une interruption interne. Pour la déclarer, on écrit les lignes suivantes dans la fonction principale (fonction main):

```
// Timer/Counter 1 initialization  
// Clock source: System Clock  
// Clock value: 2000,000 kHz  
// Mode: CTC top=OCR1A  
// OC1A output: Toggle  
// OC1B output: Discon.  
// Noise Canceler: Off  
// Input Capture on Falling Edge  
// Timer 1 Overflow Interrupt: Off  
// Input Capture Interrupt: Off  
// Compare A Match Interrupt: On  
// Compare B Match Interrupt: Off  
TCCR1A=0x40; // Configuration de la valeur de comparaison  
TCCR1B=0x0A;  
TCNT1H=0x00;
```

```
TCNT1L=0x00;
```

```
ICR1H=0x00;
```

```
ICR1L=0x00;
```

```
OCR1AH=0x4E; // Sélectionne la base de temps sur une fréquence de 2 MHz, soit un top d'horloge toutes les 0,5 us.
```

```
OCR1AL=0x20; // Une interruption quand on arrive à 20 000 top d'horloge (0x4E20 soit 10 ms)
```

```
OCR1BH=0x00;
```

```
OCR1BL=0x00;
```

Attention de ne pas oublier ces déclarations :

```
// Timer(s)/Counter(s) Interrupt(s) initialization
```

```
TIMSK=0x10;
```

et

```
// Global enable interrupts
```

```
#asm("sei")
```

On déclare enfin la fonction interruption permettant de compter les centièmes de secondes (variable « Temps »), les secondes, les minutes et les heures.

```
interrupt [TIM1_COMPA] void timer1_compa_isr(void)
```

```
{
```

```
    Temps++;
```

```
    if (Temps >= 100)
```

```
    {
```

```
        Temps=0;
```

```
        Seconde++;
```

```
        if(Seconde >= 60)
```

```
        {
```

```
            Seconde=0;
```

```
            Minute++;
```

```
            if(Minute >= 60)
```

```
            {
```

```
                Minute=0;
```

```
                Heure++;
```

```
        if(Heure>=99)
        {
            Heure=0;
        };
    };
};
};
};
```

Attention, pour pouvoir être récupérées facilement, les variables Temps, Seconde, Minute et Heure sont déclarées en tout début de programme en tant variables globales.

```
// Declare your global variables here
unsigned char Temps, Seconde, Minute, Heure;
```

3.1.3. L'écriture du temps de passage dans le TAG

(Cette partie du programme n'a lors de la rédaction, toujours pas été testée.)

Il faut faire attention à ce que le module RFID soit prêt à recevoir les données envoyées. C'est pourquoi avant chaque envoi, on attend que l'entrée « CTS » soit à 1. On récupère au préalable le numéro de borne dans la fonction principale.

```
void EcritureTemps(int Borne)
{
    while(CTS==1);
    trame='W', //ou 0x57 : permet d'écrire dans la carte
    USART_Transmit(trame);

    while(CTS==1);
    trame=Borne, //Écriture à l'adresse du numéro de borne
    USART_Transmit(trame);

    while(CTS==1);
```

```
    trame=0x01, //Clé de sécurité à choisir entre 0 et 31 sur les 5 premiers bits
de l'octet
    USART_Transmit(trame);

// Écriture du temps de passage

while(CTS==1);
trame=Temps;
USART_Transmit(trame);

while(CTS==1);
trame=Seconde;
USART_Transmit(trame);

while(CTS==1);
trame=Minute;
USART_Transmit(trame);

while(CTS==1);
trame=Heure;
USART_Transmit(trame);

}
```

Une donnée sera stockée sur chaque octet. Le numéro de borne va sélectionner la partie de la carte où écrire le temps de passage. Ainsi, la borne d'arrivée n'écrasera pas par exemple, les informations fournies par la borne de départ.

Après la programmation, on transfère le programme au micro-contrôleur par liaison série. On a besoin de la carte convertisseur parallèle-série, disponible à l'IUT.

3.2. Routage sur Orcad

Le prototype réalisé par Vivien MARTINEZ nous a permis d'effectuer une série de tests.

Vu que le typon montrait quelques imperfections et erreurs, nous avons conçu un autre routage sur Orcad pour créer des versions finales. Les différents schémas électriques qui composent la carte ont été effectués à partir du logiciel Orcad Capture. Les bibliothèques et les empreintes des composants utilisés ont été fabriqués par M. LEQUEU et M. BRAULT.

3.2.1. Schéma d'alimentation

Voici donc le schéma électrique de l'alimentation qui permet entre autre d'obtenir la tension nécessaire à la carte de 5 volts. Pour protéger le circuit, un fusible a été ajouté.

La diode verte (notée D3) montre le bon fonctionnement de celui-là.

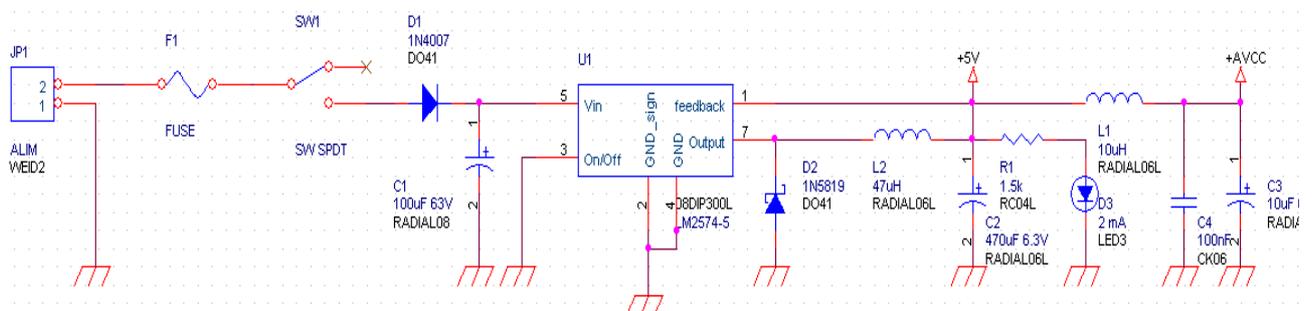


Illustration 14: Schéma électrique du régulateur de tension[5]

3.2.2. La partie ATMEGA 8535 et sélection de borne

Le micro-contrôleur possède le schéma électrique le plus complexe. En effet, il est le centre de la carte et gère tous les composants.

En bas à droite du schéma, le switch (sélection de la borne) permet de sélectionner le numéro de la borne. Il est relié au 4 premiers bits du port A⁴ de l'ATMEGA. Entre les deux, le réseau de résistances permet de définir le niveau de détection des bits. Ici l'état 1.

4 Broches 37 à 40 de l'ATMEGA 8535

Juste au dessus, un buzzer a pris place et est connecté au port A⁵. Grâce à son signal sonore, il indique la fin de la détection.

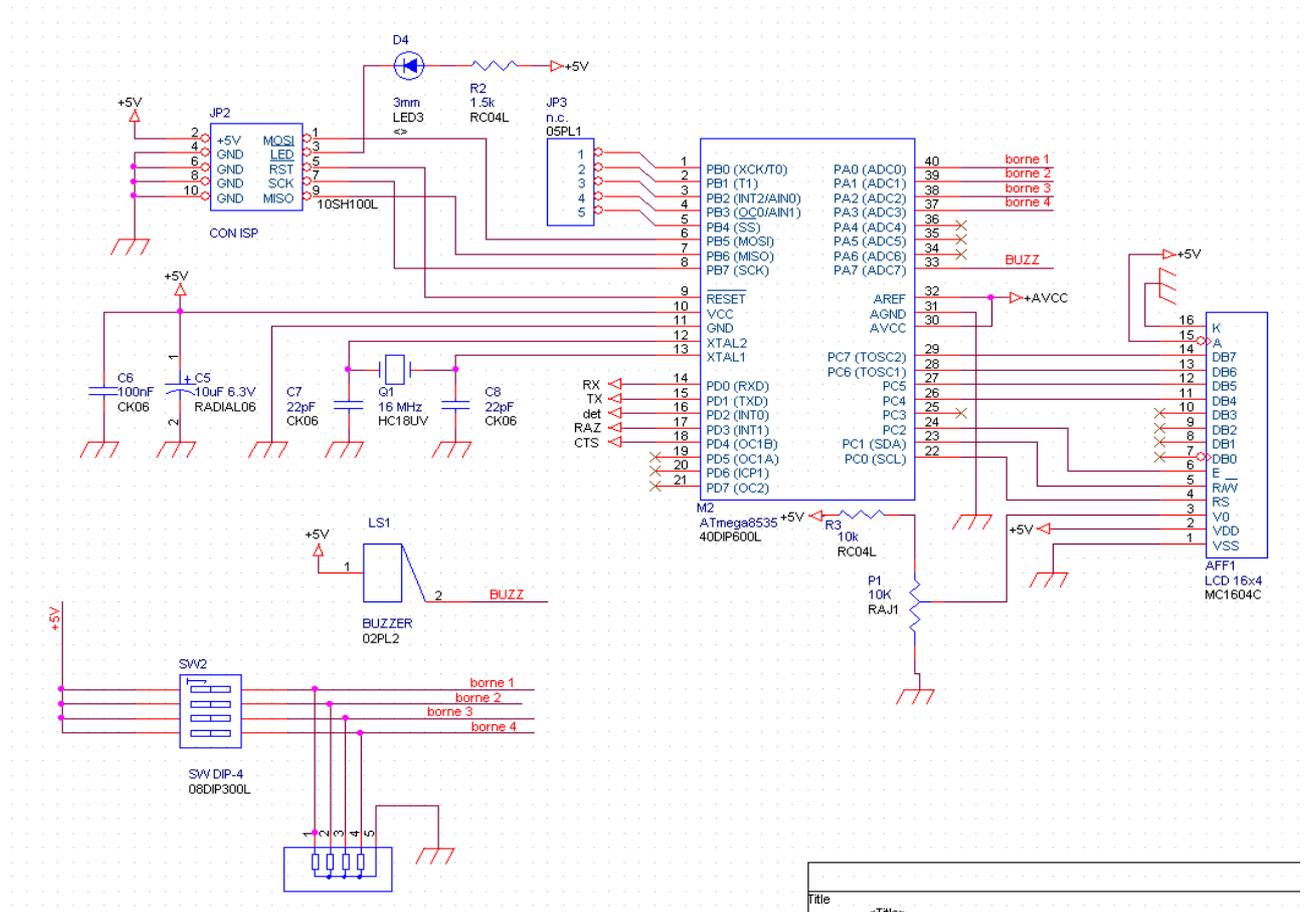


Illustration 15: Schéma électrique du micro-contrôleur[5]

3.2.3. Le schéma du RFID

Les broches 22 à 24 (Rx : Réception, Tx : Transmission et CTS : prêt à recevoir) du module permettent l'envoi des trames par liaison série. Des résistances ont été ajoutées aux diodes pour la détection par contrôle visuel. Enfin l'antenne 13,56Mhz est reliée au broches 9 et 12.

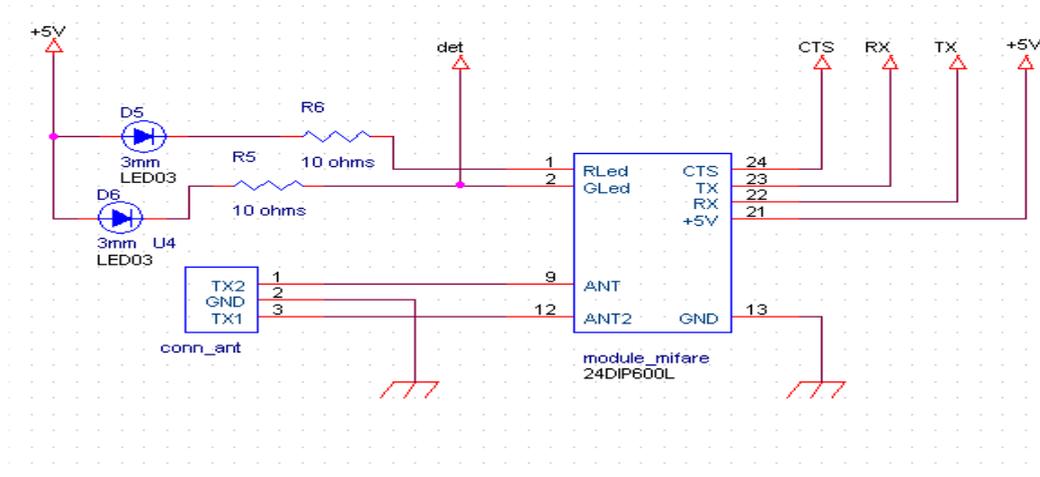


Illustration 16: Schéma électrique du module RFID[5]

3.2.4. La remise à zéro synchronisée

Pour permettre de synchroniser les horloges des boîtiers, des connecteurs ont été ajoutés. Ils permettent de les relier grâce à un fil, pour réinitialiser en même temps leurs horloges en appuyant sur le bouton poussoir (beige sur nos bornes). Le signal noté RAZ sur le schéma est connecté à l'entrée d'interruption INT1 de l'ATMEGA (voir programmation).

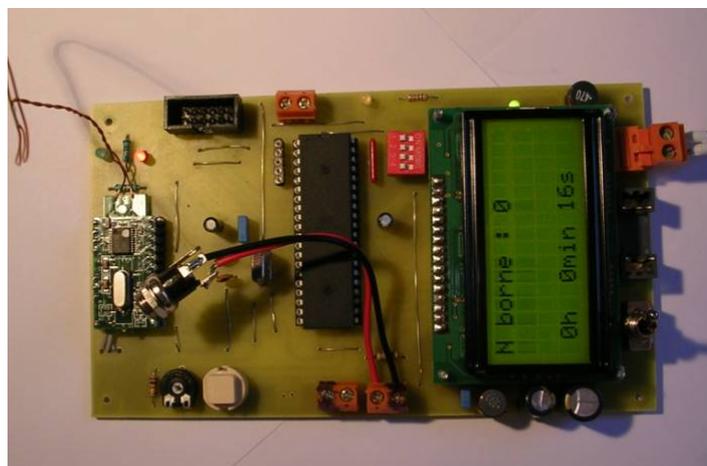


Illustration 17: La carte finale[3]

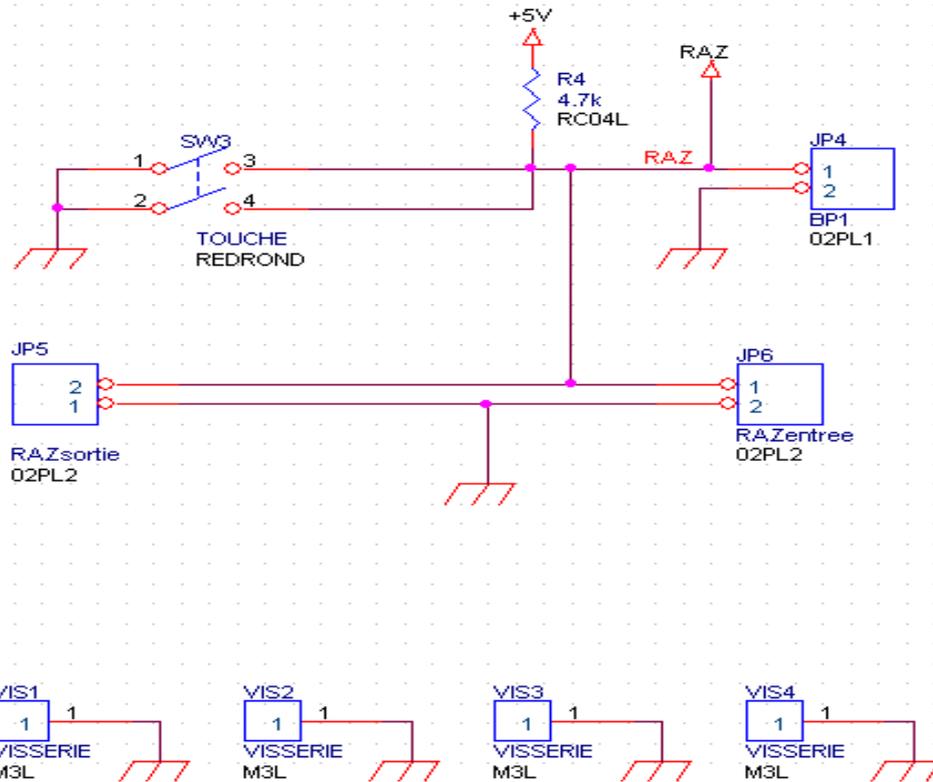


Illustration 18: Schéma électrique de la remise à zéro synchronisée[5]

3.2.5. Le typon

Après avoir effectué différents réglages, on obtient le typon final. Pour l'obtenir, il a fallu créer une nouvelle empreinte pour le module RFID car ses caractéristiques sont spécifiques.

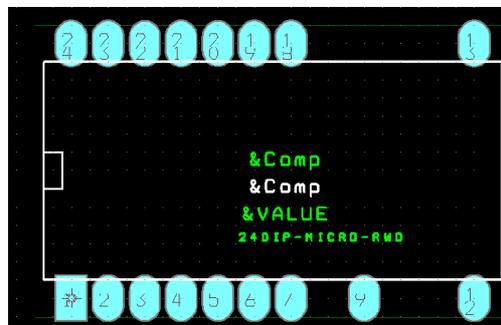


Illustration 19: L'empreinte du module RFID[6]

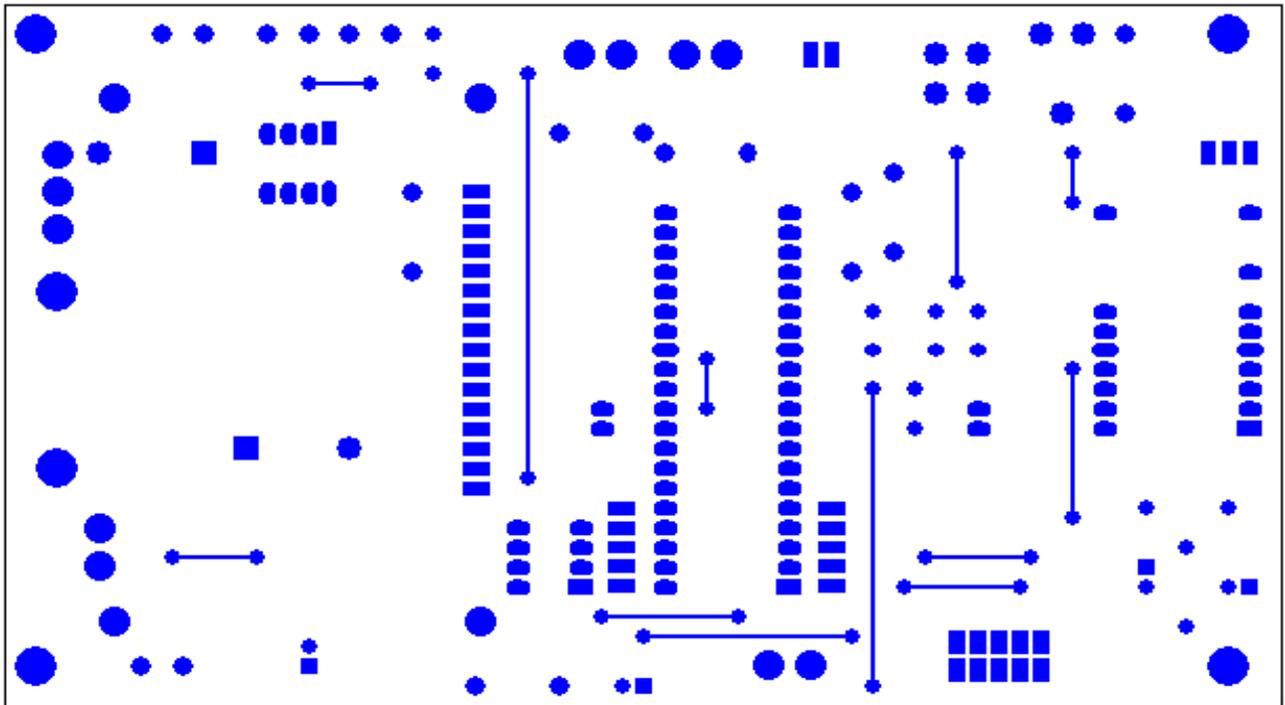


Illustration 20: Routage face composant[5]

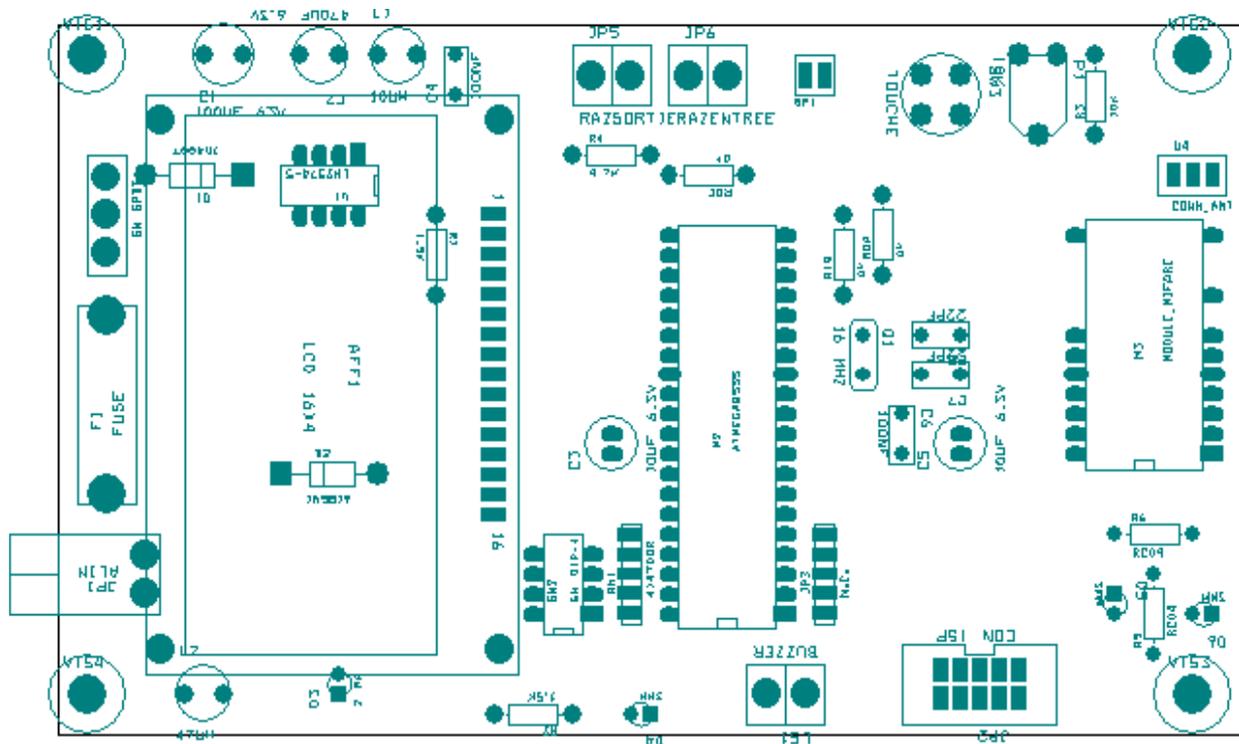


Illustration 21: Routage coté composant[5]

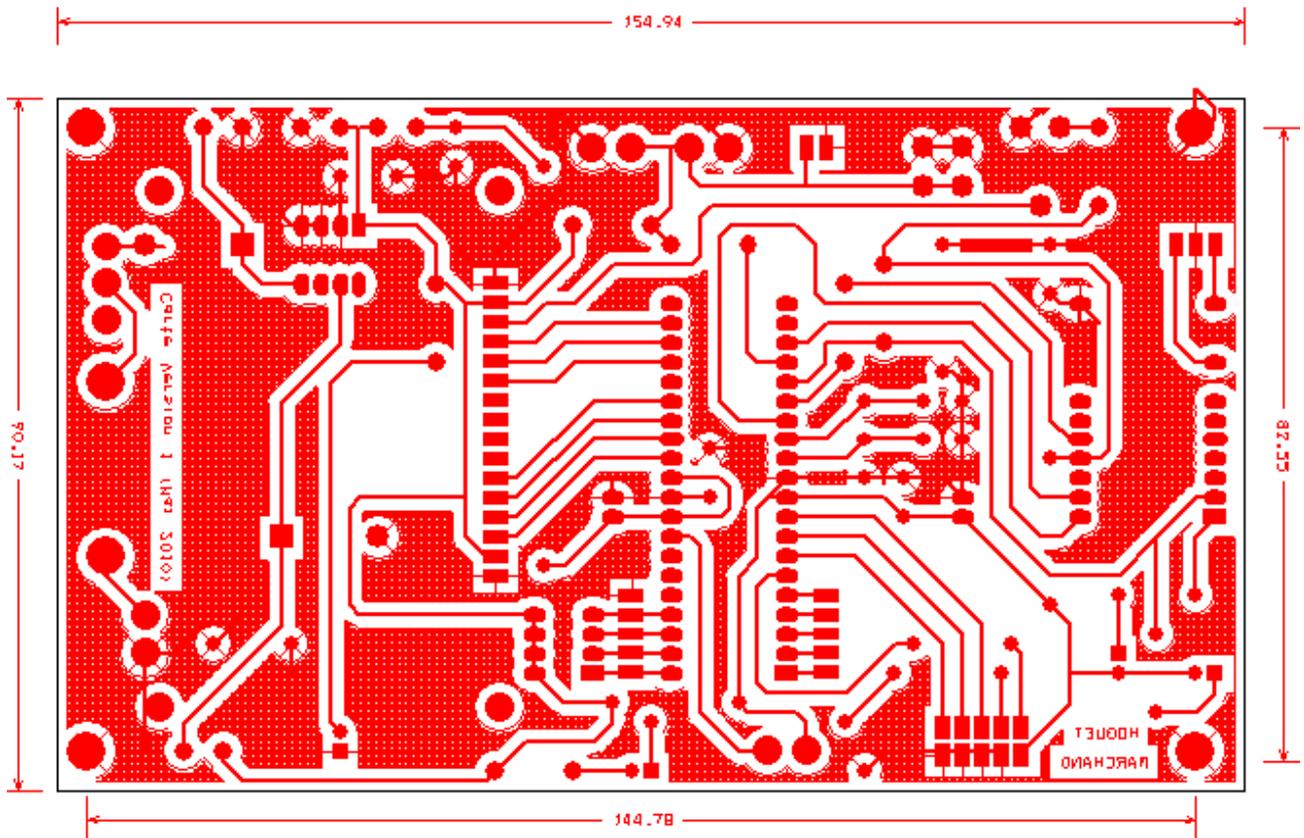


Illustration 22: Routage coté cuivre[5]

Le routage a été réalisé pour plus de facilité en simple face. C'est pourquoi, des fils (appelés aussi straps) sont visibles sur la face composant. Au moment de la rédaction du rapport, l'ensemble des composants a été soudé et une partie de la carte a été testée. Par contre il reste une erreur au niveau du module RFID qui empêche la détection des TAGs.

3.3. Problèmes rencontrés

Lors de la création, nous avons rencontré quelques problèmes qui nous ont pris plus ou moins de temps à résoudre.

3.3.1. Le module RFID

C'est la partie, qui a été la plus problématique. Premièrement, il a fallut ajouter des composants et des connexions sur la carte prototype pour la faire fonctionner.

Ensuite, nous devons comprendre le processus de reconnaissance et de transmission des trames du circuit imprimé. On a ainsi vu que pour envoyer une trame, il faut auparavant vérifier que la patte CTS du module RFID est prête à recevoir : c'est à dire qu'elle soit à l'état 1. Dans le cas contraire, elle reste à 0 pour montrer que le module n'est pas prêt.

Enfin, à l'heure de la rédaction il subsiste toujours un problème au niveau de la détection des TAGs, alors que celle-ci fonctionnait avec le prototype. Ce problème sera résolu dans les prochains jours.

3.3.2. La programmation

Avant le projet, le logiciel CodeVision AVR nous était totalement inconnu. Après quelques essais, nous nous sommes lancés dans la programmation. Nous avons eu deux problèmes : gérer l'interruption interne et lire les trames en réception. Pour l'interruption, il suffisait seulement d'ajouter deux lignes d'initialisation. Et pour les trames de bien lire et comprendre la datasheet de l'annexe 3.

3.3.3. Le routage sur Orcad

Le plus dur dans le routage a été de respecter le cahier des charges. Il fallait non seulement faire attention à la disposition et l'accessibilité des composants mais aussi ne pas dépasser les dimensions de la carte de : 150*90mm.

Enfin, le plan de masse et le placement du maximum de pistes sur la face cuivre a été laborieux, mais cependant faisable.

3.4. Ce qu'il reste à faire

Il reste pour l'instant une erreur au niveau du module RFID. Celui-ci ne veut reconnaître aucun TAG sur la borne version finale.



Illustration 23: La carte finale dans son boîtier[3]

Il faudra penser à faire un trou dans le boîtier pour que l'écran soit visible sans démonter le couvercle.

Nous n'aurons pas le temps de créer la borne de lecture après course. Ce boîtier doit lire les informations de tous les badges pour constituer une base de données sur un tableur. Le micro-contrôleur de cette borne aura besoin d'une double liaison série, pour à la fois lire les badges et effectuer une transmission à l'ordinateur, qui effectuera les calculs et le classement.

Conclusion

Durant les heures consacrées à la réalisation du projet, nous avons mis en application nos connaissances théoriques et pratiques acquises lors des trois semestres. Il nous a fallût comprendre et maîtriser la technologie RFID, afin de mettre en place des solutions pratiques et fonctionnelles, pour fabriquer le chronomètre sans fil.

Dans un premier temps, des études ont été faites sur la réalisation de Vivien. Puis des changements ont été apportés aux schémas électriques et typons pour concevoir la version finale.

Lors du projet, nous nous sommes heurtés à de nombreux problèmes . Pour les résoudre, nous avons accomplis une démarche scientifique a été mise en place, afin de contourner chacun d'eux. Malgré cela une erreur subsiste toujours, et les prochains jours permettront de présenter une version finale fonctionnelle lors de l'oral.

Le projet a donc été pour nous un reflet du monde du travail. Avec notre démarche, notre rigueur et nos efforts, on s'est approché d'une véritable expérience professionnelle.

Résumé

La technologie RFID est une technologie analogue au code à barre. Elle permet dans notre cas, d'établir un chronométrage et un classement de chaque participant lors d'une manifestation sportive. Un badge nominatif sera présenté par le coureur à des points stratégiques (départ, arrivé...), devant une borne d'écriture.

L'objectif a donc été de créer ces boîtiers, capable de communiquer les temps de passage à distance par liaison RFID. Dans un premier temps, le prototype de Vivien MARTINEZ a été récupéré pour analyse et modification. Ensuite, une longue période a été passée à l'élaboration du programme interne permettant entre autre de faire fonctionner une horloge, d'écrire dans les cartes personnels et de différencier les bornes de départ, intermédiaire et d'arrivée.

Les cartes électroniques de chaque borne sont conçu de la même façon. Chacune possède un régulateur de tension, un module RFID et un micro-contrôleur. Pour pouvoir les différencier, un bouton spécifique permet de sélectionner un numéro entre 1 et 9. Afin que les bornes est la même horloge interne, une interface de synchronisation est disponible.

Après avoir résolu les problèmes auxquels nous nous sommes confrontés, nous avons établi schémas électriques et typons afin de concevoir notre propre version finale. Cependant, nous n'aurons pas le temps de fabriquer la borne de lecture après course. De plus, du travail reste à produire dans les prochains jours pour présenter une version correct lors de l'oral.

231 mots

Index des illustrations

Illustration 1: Portique disponible à la location[2].....	5
Illustration 2: Le boîtier[3].....	6
Illustration 3: Des TAGs (ou badge RFID)[3].....	7
Illustration 4: Les Modules RFID et leurs antennes[3].....	9
Illustration 5: Schéma global du projet[1].....	10
Illustration 6: Schéma fonctionnel de niveau 1[1].....	10
Illustration 7: Schéma fonctionnel de niveau 2[1].....	11
Illustration 8: La carte finale[3].....	11
Illustration 9: Présentation du TAG devant l'antenne[3].....	12
Illustration 10: Le micro-contrôleur ATMEGA 8535[3].....	15
Illustration 11: Le régulateur de tension LM2574N-5[3].....	15
Illustration 12: Le module CMS RFID 13,56 MHz[3].....	16
Illustration 13: Afficheur large écran[3].....	17
Illustration 14: Schéma électrique du régulateur de tension[5].....	22
Illustration 15: Schéma électrique du micro-contrôleur[5].....	23
Illustration 16: Schéma électrique du module RFID[5].....	24
Illustration 17: La carte finale[3].....	25
Illustration 18: Schéma électrique de la remise à zéro synchronisée[5].....	25
Illustration 19: L'empreinte du module RFID[6].....	26
Illustration 20: Routage face composant[5].....	26
Illustration 21: Routage coté composant[5].....	27
Illustration 22: Routage coté cuivre[5].....	27
Illustration 23: La carte finale dans son boîtier[3].....	30

Index des tables

Tableau 1: Planning prévisionnel et final[4].....	8
Tableau 2: Nomenclature et coût de fabrication d'une borne[4].....	14

Bibliographie

- [1] **MARTINEZ Vivien**, "*Document technique : Le Chronométrage RFID*", 2010.
- [2] **DAG Europe**, "*Chronométrages RFID*", [En ligne]. <<http://eu.dag-system.com/Product.aspx?Id=12>> (Page consultée le 2 juin 2010).
- [3] **HOGUET/MARCHAND**, "*Photographies personnels*", 2010.
- [4] **HOGUET/MARCHAND**, "*Tableau personnel*", 2010.
- [5] **HOGUET/MARCHAND**, "*Schéma personnel*", 2010.
- [6] **LEQUEU Thierry**, "*Empreinte spécifique sous ORCAD*", 2010.

Annexes

Annexe 1 : L'implantation du module RFID (voir fichier joint)

Annexe 2 : La datasheet du module RWD MIFARE (voir datasheet du module)

Annexe 3 : L'ensemble du programme de gestion des bornes d'écriture

Annexe 4 : Fiche de suivi de projet

Annexe 3 : Programmation

```
/*  
  
This program was produced by the  
CodeWizardAVR V1.25.3 Evaluation  
Automatic Program Generator  
© Copyright 1998–2007 Pavel Haiduc, HP InfoTech s.r.l.  
http://www.hpinfotech.com
```

```
Project : Test-LCD  
Version : 6  
Date : 13/05/2010  
Author : Hoguet Marchand  
Company : Thierry  
Comments: Programmation Borne d'écriture
```

```
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>  
/* the LCD module is connected to PORTC */  
#asm  
    .equ __lcd_port=0x15  
#endasm  
  
/* now you can include the LCD Functions */  
#include <lcd.h>
```

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

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

```
// Declare your global variables here
```

```
unsigned char tampon[20], Temps1[20], Identifiant[8];  
unsigned char trame, Statut;  
unsigned char Temps, Seconde, Minute, Heure;
```

```
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;  
}
```

```
// Permet de vérifier le Statut de la carte
```

```
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(0,2);  
            lcd_puts(tampon);  
        }  
        else  
        {  
            Presence=0;  
            sprintf(tampon,"Pas Carte");  
            lcd_gotoxy(0,2);  
            lcd_puts(tampon);  
        }  
    return Presence;  
}
```

```
void Recoit_UID (void)  
{  
    int i;  
    unsigned char car;;  
    i=0;  
    while(CTS==1);  
    trame='U',    //ou 0x55 : permet de lire l'UID  
    USART_Transmit(trame);  
    Statut = USART_Receive();
```

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

// Mise en place de l'interruption interne 2000 KHz permettant d'établir le temps

```
interrupt [TIM1_COMPA] void timer1_compa_isr(void)
{
    Temps++;
    if (Temps>=100)
    {
        Temps=0;
        Seconde++;
        if(Seconde>=60)
        {
            Seconde=0;
            Minute++;
            if(Minute>=60)
            {
                Minute=0;
                Heure++;
                if(Heure>=99)
                {
                    Heure=0;
                };
            };
        };
    };
};
}
```

```
void Horloge(void)
```

```
{  
    sprintf(Temps1, "%2dh %2dmin %2ds", Heure, Minute, Seconde);  
    lcd_gotoxy(0,3);  
    lcd_puts(Temps1);  
}
```

```
//Remise à zéro synchronisée
```

```
interrupt [EXT_INT1] void ext_int1_isr(void)
```

```
{  
    Temps=0;  
    Seconde=0;  
    Minute=0;  
    Heure=0;  
}
```

```
//Déclenchement à la detection d'une carte
```

```
interrupt [EXT_INT0] void ext_int0_isr(void)
```

```
{  
  
}
```

```
void EcrireTemps(int Borne)
```

```
{  
    while(CTS==1);  
    trame='W', //ou 0x57 : permet de lire l'UID  
    USART_Transmit(trame);  
  
    while(CTS==1);  
    trame=Borne, //Ecriture à l'adresse du numéro de borne  
    USART_Transmit(trame);
```

```
    while(CTS==1);
    trame=0x01; //Clé de sécurité à choisir entre 0 et 31 sur les 5 premiers bits
de l'octet
    USART_Transmit(trame);

    while(CTS==1);
    trame=Temps;
    USART_Transmit(trame);

    while(CTS==1);
    trame=Seconde;
    USART_Transmit(trame);

    while(CTS==1);
    trame=Minute;
    USART_Transmit(trame);

    while(CTS==1);
    trame=Heure;
    USART_Transmit(trame);

}

void main(void)
{
// Declare your local variables here

// Input/Output Ports initialization
// Port A initialization
// Func7=Out Func6=Out Func5=Out Func4=Out Func3=In Func2=In Func1=In
Func0=In
// State7=0 State6=0 State5=0 State4=0 State3=T State2=T State1=T State0=T
PORTA=0x00;
```

DDRA=0xF0;

// Port B 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

PORTB=0x00;

DDRB=0x00;

// Port C initialization

*// Func7=Out Func6=Out Func5=Out Func4=Out Func3=Out Func2=Out
Func1=Out Func0=Out*

// State7=0 State6=0 State5=0 State4=0 State3=0 State2=0 State1=0 State0=0

PORTC=0x00;

DDRC=0xFF;

// Port D 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

PORTD=0x00;

DDRD=0x00;

// Timer/Counter 0 initializatio_n

// Clock source: System Clock

// Clock value: Timer 0 Stopped

// Mode: Normal top=FFh

// OCO output: Disconnected

TCCR0=0x00;

TCNT0=0x00;

OCR0=0x00;

// Timer/Counter 1 initialization

// Clock source: System Clock

// Clock value: 2000,000 kHz

```
// Mode: CTC top=OCR1A
// OC1A output: Toggle
// OC1B output: Discon.
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer 1 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: On
// Compare B Match Interrupt: Off
TCCR1A=0x40; // Configuration de la valeur de comparaison
TCCR1B=0x0A;
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x4E; // Sélectionne la base de temps sur une fréquence de 2 MHz, soit
un top d'horloge toutes les 0,5 us.
OCR1AL=0x20; // Une interruption quand on arrive à 20 000 top d'horloge
(0x4E20 soit 10 ms)
OCR1BH=0x00;
OCR1BL=0x00;

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

// External Interrupt(s) initialization
// INT0: On
```

```
// INT0 Mode: Falling Edge
// INT1: On
// INT1 Mode: Falling Edge
// INT2: Off
GICR|=0xC0;
MCUCR=0x0A;
MCUCSR=0x00;
GIFR=0xC0;

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x10;

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

/* initialize the LCD for 2 lines & 16 columns */
lcd_init(16);

/* switch to writing in Display RAM */
```

```
//lcd_gotoxy(0,0);
//lcd_putsf("Projet Borne RFID");

// initialisation des variables globales
Temps=0;
Seconde=0;
Minute=0;
Heure=0;
BUZZER=1;

// Global enable interrupts
#asm("sei")

while (1)
{
    //Déclaration des variables locales

    unsigned char NumeroBorneHexa;
    int Dizaine, Unite, NumeroBorne;
    int PresenceCarte;

    // Appelle de L'horloge
    Horloge();

    //Changement du numéro de borne de 0 à 9

    NumeroBorneHexa = (PINA&0x0f);
    Dizaine=NumeroBorneHexa/10;
    NumeroBorne=NumeroBorneHexa/10;
    Unite=((int)NumeroBorneHexa)%10;
    NumeroBorne=NumeroBorne+Unite;

    sprintf(tampon,"N borne : %2d",NumeroBorne);
```

```
lcd_gotoxy(0,0);  
lcd_puts(tampon);
```

```
//Vérification du statut de la carte
```

```
PresenceCarte = VerifStatut (); // Appelle de la fonction permettant de vérifier  
le statut
```

```
if(PresenceCarte==1)
```

```
{
```

```
/* Possibilité de lire l'identifiant unique de la carte
```

```
Recoit_UID();
```

```
sprintf(tampon,"%s",Identifiant);
```

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

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

```
// Ecriture du temps de passage sur la carte
```

```
EcritureTemps(NumeroBorneHexa);
```

```
//on attend 500 ms et on déclenche le buzzer pour marquer la fin de la  
lecture il faut faire attention de
```

```
//ne pas figer l'affichage de l'horloge
```

```
BUZZER=0; //logique inversée
```

```
delay_ms(100);
```

```
Horloge();
```

```
delay_ms(100);
```

```
Horloge();
```

```
delay_ms(100);
```

Horloge();

delay_ms(100);

Horloge();

delay_ms(100);

BUZZER=1;

}

}

}

Annexe 4 : Fiche de suivi de projet

Semaine 8 :

- Explication par M. LEQUEU du déroulement des séances de projet,
- Choix des sujets par binôme. Nous avons convenu de continuer le projet « mesure du temps de parcours », débuté par Vivien MARTINEZ, un étudiant de l'IUT actuellement en licence.

Semaine 9 :

- Prise de connaissance du sujet et lecture des différentes datasheets.

Semaine 10 : Formation sous Orcad

- Présentation et formation Orcad. Ce logiciel est composé de 2 parties. « Capture » permet de saisir les schémas électriques, tandis que « Layout » fait le routage et le placement des composants,
- Le logiciel va nous permettre de router notre carte rapidement.

Semaine 11 :

- Étude des solutions à apporter,
- Rassemblement et soudure des composants sur le prototype de Vivien MARTINEZ,
- Recherche de datasheet de l'ATMEGA 8535 et du LM2574N-5,
- Tests du prototype → Dysfonctionnement

Semaine 12 :

- Recherche de la panne de détection avec la datasheet du composant. Résultat : ajout de 2 résistances de 1KOhms et mise à la masse de la broche 13,
- Test du module RFID longue portée : 13,56 Mhz.

Semaine 13 :

- Familiarisation avec le logiciel CodeVision AVR.
- Début de la programmation de la borne : sélection du numéro de borne. Pour une bonne lecture il a fallut créer un masque pour ne lire que les 4 premiers bits du port A. Masque : $a=(PINA\&0x0f)$ avec « a », variable « unsigned char »..

Semaine 16 : (Rédaction du rapport écrit à la maison)

- Suite de la programmation,
- Recherche d'une démarche pour effectuer la lecture des trames du module RFID.

Semaine 17 : (Programmation à la maison)

- Programmation des trames de statut et de réception avec les commandes « U » et « R »,
- Problème, nous n'arrivons pas à lire les trames,
- Lecture de la datasheet et remplacement du bouton poussoir de synchronisation pour créer une configuration « pas à pas ».

Semaine 18 :

- Lecture de la datasheet du module RWD Mifare 13,56 Mhz.
- Programmation : nous avons compris que pouvoir lire les trames, il faut auparavant l'entrée « CTS » du module RFID. Cette broche permet de voir si le module est prêt à recevoir des informations. Pour cela on créer une boucle « while » (tant que) qui attend que cette entrée soit à l'état 0 : voir datasheet.

Semaine 19 :

- Nous avons entrepris de modifier les schémas électriques du prototype sous capture,
- Nombreuses erreurs à la compilation qui ont pris beaucoup de temps à être résolu. Entre autre des erreurs d'empreintes, et d'implémentation,
- Nous avons établi le typon de la carte finale à la maison. Beaucoup de temps passé sur cette étape pendant le week-end. En effet, la disposition et la surface de la carte était contraignante.

Semaine 20 : Mercredi 19 mai 2010

- Le typon étant fini sur le logiciel Orcad, nous avons procédé à la construction des deux cartes jouant le rôle de la borne de départ et d'arrivée. Pour cela, nous avons insolé, gravé, percer celles-ci. Puis, nous avons soudé les composants, pris chez le magasinier.
- Nous avons fini de souder chez nous.

Vendredi 21 mai 2010 :

- Test du programme fait à la maison permettant d'écrire dans les badges.
- Test des interruptions internes et externes permettant de créer l'horloge interne et de la synchroniser. Problème : nous avons perdu du temps car il manquait des lignes de code d'initialisation de l'interruption interne. De plus, une erreur était présente dans l'incrémentation du temps de parcours. Celle-ci fut vite résolue,
- Commande des buzzers chez radiospars,
- Rédaction du rapport écrit.

Semaines 21 : Séance libre

- Suite de la rédaction du rapport,
- Vérification et tests sur les bornes finales. Problème : la détection des badges ne se fait pas.