Analyse des étiquettes d'identification par radiofréquence (RFID)

J.-M. Friedt, 30 août 2008

Les étiquettes interrogeables par radiofréquence (RFID, *RadioFrequency IDentifier*) sont utilisées, souvent à notre insu, dans de nombreuses activités quotidiennes : antivol, suivi et identification de marchandises, contrôle d'accès, bientôt les passeports ... autant d'applications qui justifient de comprendre la technologie sous-jacente, ses domaines d'applications et les éventuels risques associés. Nous allons mettre en pratique les connaissances acquises sur les puces servant au tatouage électronique des animaux de compagnie, en nous fixant pour objectif de réaliser l'ensemble de la chaîne de lecture du code identifiant un chien.

1 Introduction

Nous proposons d'étudier l'ensemble de la chaîne d'interrogation d'étiquettes d'identification [1, 2] interrogeables par radiofréquences (RFID), du circuit électronique d'acquisition des données au traitement du signal pour finalement interpréter les données résultantes. Notre objectif est de lire le code (tatouage) contenu dans les étiquettes d'identification d'animaux de compagnie. Ces puces ont pour propriété de ne pas embarquer de source d'énergie mais d'"absorber" une partie de l'énergie fournie par l'interrogateur pour alimenter leur circuit électronique. Il s'agit donc de dispositifs totalement passifs, de durée de vie *a priori* illimitée, fournissant des fonctionnalités simples telles que le stockage et la restitution d'un code d'identification. La tendance actuelle vise à compléter l'identification avec la mesure de quantités physiques [3].

Ce projet est relativement simple pusqu'il concerne les dispositifs les plus accessibles – ne contenant aucune séurité pour restituer les informations contenues dans l'étiquette – et la principale difficulté consiste à trouver les acronymes pertinents pour focaliser les recherches sur le web sur les aspects techniques noyés dans les publicités, légendes urbaines et autre débats stériles, sans intérêt pour le domaine technique qui nous intéresse.

Parmi les trois grandes gammes de dispositifs que nous allons brièvement répertorier ici, nous ne nous intéresserons qu'à ceux fonctionnant aux fréquences les plus basses : ces dispositifs sont les plus simples à mettre en œuvre (les fréquences mises en jeu ne nécessitent aucune précaution de fabrication du circuit électronique d'interrogation) et celles utilisées dans les tatouages électroniques des animaux de compagnie. Les trois grandes classes de RFID – définies par leur gamme de fréquence de fonctionnement et donc leur domaine d'application (différentes portées, efficacité de l'antenne et pénétration de l'onde électromagnétique dans l'objet porteur du RFID) sont :

- les dispositifs fonctionnant en basse fréquence (en dessous de 150 kHz, et en particulier 125 et 134,2 kHz). Compte tenu des longueurs d'ondes (plusieurs kilomètres), le couplage entre la puce et l'interrogateur ne peut être que par induction magnétique entre bobines. Ce couplage décroît rapidement (d^{-3}) avec la distance d, et la portée de ces dispositifs est nécessairement réduite.
- les dispositifs fonctionnant aux fréquences dans la gamme 10-1000 MHz, avec en particulier les plages normalisées de 13,56 MHz [4, 5] et 868 MHz. L'efficacité de couplage des ondes électromagnétiques dans l'antenne est accrue, mais l'atténuation par des objets diélectriques augmente.
- les dispositifs fonctionnant aux très hautes fréquences (au-delà du GHz, en particulier 2,45 GHz, soit des longueurs d'onde de l'ordre de la dizaine de centimètres). L'antenne de l'interrogateur peut être très efficace – voir directive – mais la pénétration de l'onde dans les objets est pratiquement nulle : la puce interrogée doit se trouver en surface, mais le débit est d'autant plus élevé que la fréquence de porteuse et la bande passante associée sont élevées.

2 Circuit électronique d'interrogation

La réalisation du circuit va se résumer en l'exploitation d'un composant dédié à l'interrogation des RFID basses fréquences : l'Atmel U2270B [6, 7]. Ce composant, disponible pour 3,30 euros chez Farnell (référence 1095806), se charge de la partie analogique de l'interrogation : oscillateur générant la porteuse d'interrogation du RFID, amplificateur pour alimenter la bobine faisant office d'antenne, circuit de mesure de l'intensité du couplage de la bobine d'interrogation avec le RFID, signal de sortie exploitable par un microcontrôleur (Fig. 1). Nous ne nous servirons pas de la fonctionnalité de modulation de la porteuse afin d'envoyer des commandes au RFID. Dans le cas qui va nous intéresser, la porteuse est émise en continue et la modulation d'impédance vue par l'antenne est induite par le RFID lorsque celui-ci a accumulé assez d'énergie pour entrer en fonctionnement.

Les deux subtilités notables pour exploiter ce circuit sont :

- prévoir une résistance variable pour régler la fréquence de l'oscillateur et l'ajuster à la fréquence à laquelle répond le RFID (125 ou 134,2 kHz, à ajuster en fonction de l'inductance de la bobine faisant office d'antenne)
- la sortie est en *collecteur ouvert*, ce qui signifie qu'en l'absence de résistance de tirage vers l'alimentation du microcontrôleur, aucun signal n'apparaît sur la broche 2 de l'U2270B. Ce point crucial n'est pas détaillé dans la datasheet mais a été identifié en consulting `http://kudelsko.free.fr/transpondeur/presentation2.htm`.

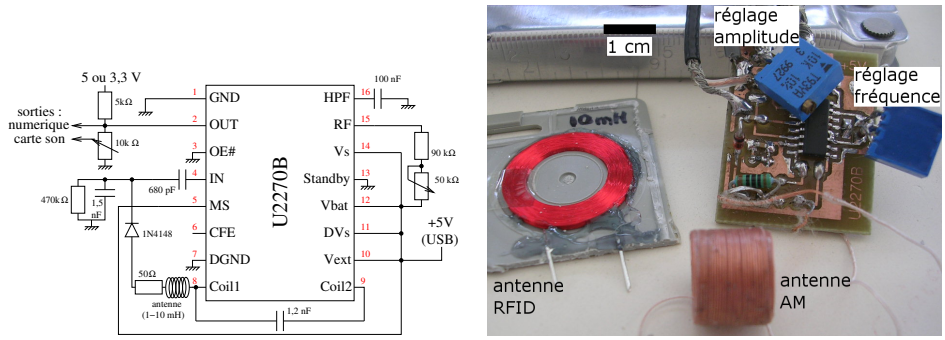

FIG. 1 – La réalisation du circuit ne pose pas de problème majeur puisque la fréquence de travail est de l'ordre de la centaine de kilohertz. Il s'agit d'un circuit simple face de petites dimensions dont la face inférieure, non-visible sur cette photo, est totalement couverte de cuivre pour faire office de plan de masse. Le circuit est alimenté en 5 V, obtenu depuis un port USB pour rendre le montage mobile lorsqu'il est utilisé avec un ordinateur portable. Diverses antennes pourront être utilisées en fonction des rebuts disponibles : nous avons expérimenté avec une bobine faisant à l'origine office d'antenne de récepteur radio AM (bobine soudée au circuit, en bas), et avec une bobine extraite d'une carte d'accès RFID (bobine rouge à gauche). Le bobinage, de quelques centaines de tours, doit présenter une inductance de quelques millihenry [8].

Le bon fonctionnement du circuit est validé en observant à l'oscilloscope ou au compteur de fréquence les signaux attaquant la bobine (broches 8 et 9) : ajuster la résistance sur la broche 15 jusqu'à atteindre la fréquence voulue de 134,2 kHz.

3 Acquisition et traitement du signal

En l'absence d'informations détaillées sur le protocole de communication du RFID avec l'interrogateur, nous désirons utiliser un outil aussi souple que possible pour acquérir et traiter les signaux. Nous allons donc acquérir les signaux en sortie de l'U2270B sur la carte son d'un PC, pour ensuite traiter le fichier résultant et en extraire les valeurs des bits transmis. En effet, nous

avons un *a priori* sur la forme des signaux attendus, information acquise grâce à quelques mots clés qu'il nous faudra utiliser dans la majorité des recherches sur le web

- les protocoles de communication utilisés par les RFID implantés dans les animaux sont décrites dans deux standards que sont les ISO 11784 et 11785. L'obtention de ces documents est payante, mais les grandes lignes sont résumées à ref. [9].
- ce même site web fournit le nom du protocole de communication – FDX-B – et la fréquence qui va nous intéresser pour une utilisation en Europe : 134,2 kHz. Il s'agit d'un protocole full duplex dans lequel l'émetteur fournit constamment de l'énergie à l'étiquette, qui transmet son information par modulation de l'impédance vue par le bobinage de l'émetteur (circuit ouvert ou fermé du côté de l'étiquette pour moduler le signal radiofréquence).
- deux documents, refs. [10] et [11], vont nous fournir quelques informations sur la partie radiofréquence mais surtout sur la partie numérique d'interprétation des données qui va nous intéresser par la suite (section 4).

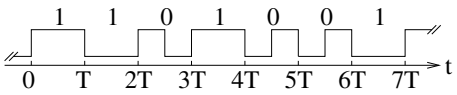

FIG. 2 – Codage de Manchester

Nous apprenons dans ce document que le codage sera de type Manchester (Fig. 2) : une transition doit toujours survenir après une période d'horloge (période que nous ne connaissons pas encore), et un zéro est indiqué par *deux* transitions dans cette période d'horloge. Le décodage du signal issu de l'interrogateur consistera donc en une mesure de durée, nécessitant donc une base de temps stable du côté de l'interrogateur : cette fonctionnalité est fournie par la carte son.

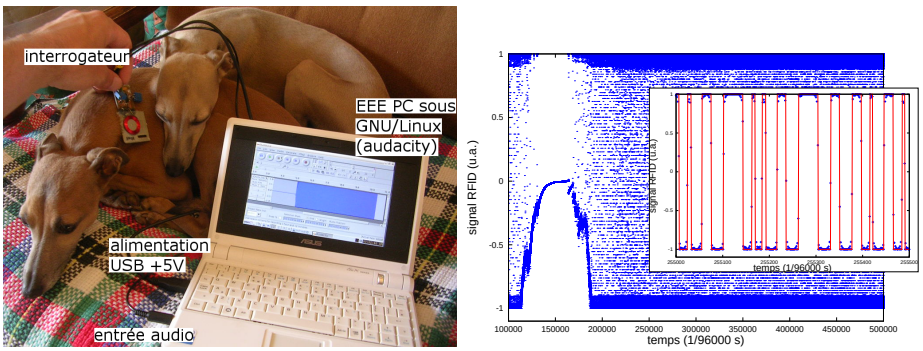

FIG. 3 – Gauche : acquisition des données. La principale difficulté de la mesure tient en l'immobilité du chien lors de la recherche de la position de l'étiquette. La portée réduite de notre circuit nécessite en effet de se placer parfaitement au-dessus de l'étiquette pour obtenir un signal stable et exploitable. Droite : exemple de signal acquis à 96 kHz, et gros plan sur une zone des points acquis présentant clairement les créneaux encodant les signaux sur les données brutes (points bleus) et après saturation des données brutes pour les convertir en valeurs binaires (traits rouges).

L'enregistrement du signal se fait par la carte son d'un Asus EEE PC 701 : afin de faciliter le traitement ultérieur du signal, nous échantillonnons à la fréquence maximale de 96 kHz, sans nous inquiéter de l'espace disque occupé (Fig. 3). Tout logiciel d'acquisition capable de contrôler les options de la carte son conviendra : nous avons pour notre part utilisé `audacity` afin de rapidement couper les segments de l'enregistrement sans intérêt et ne traiter que les parties pertinentes (Fig. 4). Les réglages sont un enregistrement à 96 kHz, mono, 16 bits/échantillons et sauvegarde au format PCM (*i.e.* fichier wav, sans compression, qui sera le plus simple à exploiter par la suite).

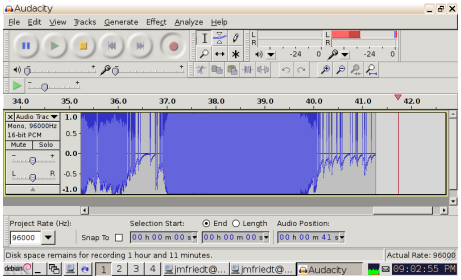

FIG. 4 – Capture d'écran de `audacity` lors de l'acquisition d'un signal de RFID : le signal valable est disponible de 37 à 40 secondes, tandis que les recherches de la position de l'étiquette sur le cou du chien avant et après ces dates induisent des fluctuations d'amplitude du signal qui le rendent inexploitable.

Toute la suite des traitements se fait au moyen de GNU/Octave (version opensource de Matlab), dont l'installation sous Debian est complétée par l'équivalent libre du Signal Processing Toolbox sous forme du paquet `octave-signal`.

Sous `octave`, les données du fichier audio sont mises en mémoire par la fonction `wavread` qui renvoie dans `fs` la fréquence d'échantillonnage et surtout dans `a` les données ajustées pour tenir entre les bornes $[-1;+1]$:

```
[a,fs,siz]=wavread('adelie96k.wav');
x=find(a>=0.);a(x)=1;
x=find(a< 0.);a(x)=-1;
```

Le signal analogique est converti en signal binaire en saturant les valeurs négatives à -1 et les valeurs positives à +1, puis nous recherchons les transitions positives (variable `pos`) et négatives (variable `neg`) des créneaux en étudiant la valeur de la dérivée du signal. Ces deux ensembles de signaux, les dates des transitions positives et négatives des créneaux acquis, vont être les bases sur lesquelles nous allons effectuer la suite de nos traitements.

```
pos=find(diff(a)> 0.5); % date montant
neg=find(diff(a)<-0.5); % date descendant
```

Sans savoir à quoi correspondent ces transitions, nous pouvons chercher à savoir si ces transitions sont représentatives d'un signal. Compte tenu de la longueur des mesures (plusieurs secondes), le signal transmis par le RFID doit nécessairement se répéter. Une transformée de Fourier doit donc présenter des pics correspondant au taux de répétition si le signal est périodique. [1].

La transformée de Fourier est calculée sur un intervalle de temps qui semble "visuellement" intéressant, *i.e.* dans lequel le signal acquis par la carte son est saturé par la réponse du RFID. La transformée de Fourier (Fig. 6) est calculée sur les points 2000 à 8000 des données contenant les intervalles de temps (`pos` ou `neg`, Fig. 5), ce qui correspond aux points 240750 à 424477 dans le fichier audio original. 60 répétitions dans (424477-240750)/96000=1,9 s signifient qu'une réponse du RFID met 32 ms. Si nous faisons l'hypothèse d'un message de 128 bits, le débit est de 4013 bits/seconde ou 0,25 ms/bit, très proche de la valeur proposée sur la page [9] : nous sommes sur la bonne voie, nous avons été capables d'obtenir les bits individuels du message, qu'il nous faut interpréter comme un message complet.

[1] Un signal contient N répétitions d'un motif de p points. Les $N \times p$ points au total représentent donc un signal périodique, de période p, *i.e.* de fréquence $1/p$. Cette fréquence caractéristique apparaît sur une transformée de Fourier sur $N \times p$ points comme un pic d'abscisse N. À titre d'illustration, prenons un signal de durée 1 s contenant un signal à 100 Hz échantillonné à la fréquence f_{ech}=1000 Hz. Alors $p = 10$ (le signal à 100 Hz est échantillonné 10 fois par période), $N \times p$=1000 et la transformée de Fourier sur les $N \times p$ points s'étale sur un intervalle $\pm f_{ech}/2$, soit -500 Hz à +500 Hz. Le pic des 100 Hz se trouve à 1/5ème de l'abscisse maximum, soit le point d'indice 500/5=100 qui est bien égal à N.

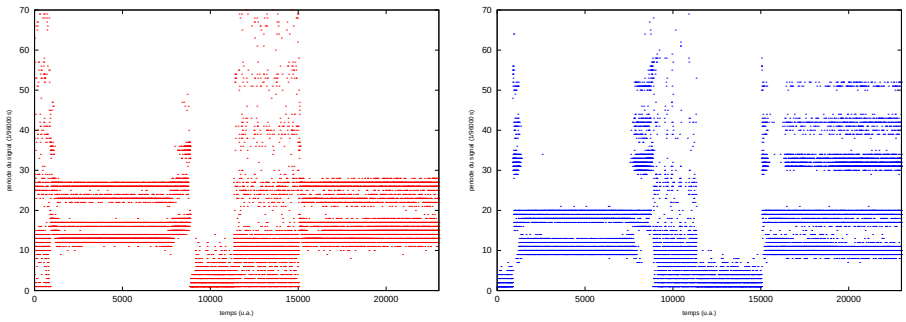

FIG. 5 – Intervalles de temps entre deux transitions positives successives (droite, en bleu), et entre deux transitions négatives successives (gauche, en rouge). L'interrogateur était correctement placé entre les dates 1000 et 9000 puisque les deux distributions sont bimodales, tandis que les données de 15000 à la fin ne sont pas exploitables puisque la distribution des transitions positives (droite) n'est pas bimodale et ne peut donc pas représenter un signal en codage Manchester.

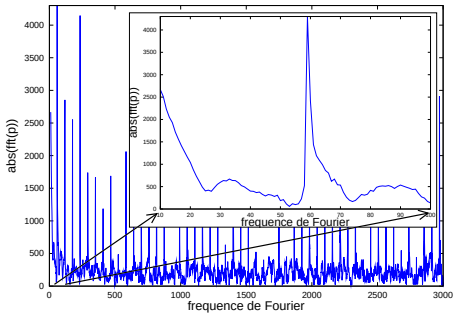

FIG. 6 – Transformée de Fourier du signal `pos` calculé plus haut (Fig. 5), pour les points d'abscisses 2000 à 8000. Un pic d'abscisse 60 indique que l'ensemble des points sur lesquels cette opération a été effectuée contient 60 répétitions du signal. Nous supposerons que ce signal est la réponse du RFID que nous cherchons à décoder.

4 Interprétation des données

Ayant constaté que les données `pos` et `neg` calculés plus haut contiennent probablement une information pertinente, il nous reste à extraire cette information et l'interpréter.

Un signal périodique est formé d'alternances de fronts montant et de fronts descendant. Deux cas se présentent, selon que le premier front soit montant ou descendant :

```
pp=length(pos);pn=length(neg);  % autant de fronts montant que descendant
if (pn<pp) p=pos(1:pn); clear pos;pos=p;
   else    p=neg(1:pp); clear neg;neg=p;
   end

n=(neg-pos);            % intervalle de temps entre descendant et montant
if (n(1)<0)             % si negatif, c'est le mauvais sens
  n=neg(2:length(pos))-pos(1:length(neg)-1);
  p=pos-neg;
  premier=0;
else
  p=pos(2:length(pos))-neg(1:length(neg)-1);
```

```
  premier=1;
end
```

Lorsque nous calculons `neg-pos`, si la première transition est un front montant, alors la date de première transition de `pos` est inférieure à la date de première transition négative et la soustraction des tableaux ne contient que des intervalles de temps négatifs. Dans ce cas, il nous faut décaler le tableau `pos` afin de soustraire l'élément d'indice `n+1` de `pos` à l'élément `n` de `neg`.

Les intervalles de temps sont alors seuillés afin de devenir des données binaires : l'histogramme de `pos` et `neg` nous indique où placer le seuil (au creux de la fonction bimodale) séparant le 1 du 0. Les deux ensembles de valeurs centrées sur 13 et 26 (alternance négative, Fig. 5 gauche) ou 10 et 20 (alternance positive, Fig. 5 droite) correspondent aux 0 et 1 respectivement, et placent le seuil vers 19 et 15 respectivement. Le fait que ces deux seuils soient différents indique que le signal en créneau initial n'est pas symétrique et que notre première étape de saturation du signal audio à ± 1 a induit cette asymétrie en prenant pour seuil 0.

```
SEUILp=19.5;SEUILn=14.5;  % parametre a regler pour chaque montage

x=find(n<SEUILn);n(x)=0;
x=find((n>0)&(n<3*SEUILn));n(x)=1;

x=find(p<SEUILp);p(x)=0;
x=find((p>0)&(p<3*SEUILp));p(x)=1;
```

Dans cette opération, les seuils `SEUILn` et `SEUILp` sont des paramètres fondamentaux, *a priori* constants puisque fixés par la période T (Fig. 2), mais que nous avons constaté devoir adapter en pratique lors de chaque nouvelle acquisition.

Nous intercalons dans la variable `total` les intervalles de temps des créneaux positifs et des créneaux négatifs pour ainsi former la séquence du code de Manchester.

```
total=zeros(length(p)+length(n),1);

if (premier==1)
 total(1:2:2*length(n))=n;
 total(2:2:2*length(p))=p;
else
 total(1:2:2*length(p))=p;
 total(2:2:2*length(n))=n;
end
```

Chaque zéro est indiqué par deux transitions courtes successives. Il nous faut donc remplacer toute paire de 0 par un zéro unique. Par ailleurs, la présence dans les données initiales d'un zéro unique est indicateur d'une erreur de décodage : ce cas ne peut théoriquement jamais survenir.

```
sortie=zeros(length(total),1);

sta=5000;sto=7000;  % plage de travail : debut et fin
pp=1;
k=sta*2+1;

 while (k < sto*2)  % elimine les "0" en double
  if (total(k)==1)  % (un zero = deux transitions courtes)
     sortie(pp)=1;pp=pp+1;k=k+1;
  else
     pp=pp+1;k=k+1;
     if (total(k)~=0) disp('erreur');k
```

```
      else k=k+1;
      end
  end
end

s=sortie(1:pp)'    % afficher le resultat
save -text s s     % attention, PAS compatible Matlab
```

En exécutant ces routines sur deux séries de points acquis sur deux chiens différents, nous obtenons les deux séries suivantes qui ont été validées par la présence de doublets de 0 dans `total`.

```
... 0000100000001110000110111111100010000000010000000001000000001
0000000000100100010110011111111010011011000101110111111011011111001000000010000000
1110000110111111000100000000100000000100000000 1
0000000000100100010110011111111010011011000101110111111011011111001000000010000000
1110000110111111000100000000100000000100000000 1
0000000000100100010110011111111010011011000101110111111011011111001000000010000000
1110000110111111000100000000100000000100000000 1
0000000000100100010110011111111010011011000101110111111011011111001000000010000000
1110000110111111000100000000100000000100000000 1 ...
```

et

```
... 1111111010111010011101111101101111100100000000010000000
1100011010110000101100000000100000000100000000 1
0000000000111001011101111011111111010111010011101111101101111100100000000010000000
1100011010110000101100000000100000000100000000 1
0000000000111001011101111011111111010111010011101111101101111100100000000010000000
1100011010110000101100000000100000000100000000 1
0000000000111001011101111011111111010111010011101111101101111100100000000010000000
1100011010110000101100000000100000000100000000 1
0000000000111001011101111011111111010111010011101111101101111100100000000010000000
1100011010110000101100000000100000000100000000 1
0000000000111001011101111011111111010111010011101111101101111100100000000010000000
1100011010110000101100000000100000000100000000 1
0000000000111001011101111011111111010111010011101111101101111100100000000010000000
1100011010110000101100000000100000000100000000 1 ...
```

Ces données ont déjà été découpées pour les faire commencer par l'entête 00000000001 qu'on ne peut trouver *que* en début de trame. Nous constatons que les séquences se répètent et font bien 128 bits de longueur.

5 Calcul du code résultant

La série de bits obtenue précédemment s'interprète au moyen des informations fournies dans [11] :

- un entête formé de 10 zéros suivis de 1 un, suite unique de chiffres indiquant de façon univoque le début d'une trame.
- 72 bits de données contenant l'identifiant qui va nous intéresser
- 18 bits de CRC
- 27 bits de données étendues

```
00000000001
 header v        v        v        v         v         v          v          v
0010001011001111111010011011000101110111111 01101111100 1 0 0000000100000000 1 1
  64-27 = Code d'identification national   pays (ISO)    16             animal
        v        v         v        v        v
```

```
100001101111110001 00000000100000001000000001
      CRC          data stream : que des 0 separes de 1
```

Les symboles `v` identifient les 1 qui sont insérés dans la trame pour garantir que l'entête (incluant ses 10 zéros consécutifs) ne se retrouve nulle part dans le corps du message. Ces "1" ne sont pas inclus dans le message et doivent être éliminés. Il nous reste donc un code de pays `0101111100` et un identifiant national `0010001010011111101001101000101101111`. En prenant le bit de poids le plus faible à gauche et l'octet de poids le plus faible à gauche, ces valeurs binaires sont égales à 0xFA et 0x3ED165F944.

Le code national de la France est 250=0xFA tel que défini dans le norme ISO 3166-1 [12].

L'identifiant national est égal en décimal à 269801093444.

Le code d'identification fourni sur le passeport de ce chien est 250269801093444 : il y a bien concordance entre cette valeur et la concaténation du code de pays suivi de l'identifiant national que nous venons de calculer.

À titre d'exercice, le lecteur pourra confirmer que le second code fournit bien le même code de pays, tandis que l'identifiant national du second chien est 269700030163.

Ces informations sont cohérentes avec la structure des tatouages électroniques, exprimée en décimal :

- les trois premiers chiffres sont le code du pays selon ISO 3166-1, 250 pour la France
- deux chiffres pour identifier la race de l'animal, avec 26 pour les chiens
- deux chiffres pour identifier le fabriquant, en général pour les chiens un nombre entre 96 et 98
- finalement, les 8 derniers chiffres identifient l'animal

6 Conclusion

Nous avons présenté pas à pas une démarche pour identifier la réponse d'une étiquette passive communiquant par radiofréquence (*RFID tag*). Nous avons identifié la fréquence de travail (134,2 kHz) et le codage de type Manchester, en accord avec les informations accumulées sur le protocole de communication avec les étiquettes implantées dans les animaux pour leur identification (FDX-B). Nous avons acquis avec du matériel communément disponible (PC équipé d'une carte son) des signaux caractéristiques de la réponse de l'étiquette, et avons décodé le contenu de cette information pour finalement trouver la valeur attendue pour l'identification de deux chiens.

Cette démarche mérite d'être étendue à d'autres types d'étiquettes [14], notamment celles équipant les badges d'accès ou les futurs passeports dits biométriques. Nous n'avons fait qu'effleurer un domaine riche en interrogeant une puce sans protection répondant systématiquement à toute sollicitation. Nous avons expérimenté avec une carte d'accès qui répond à la même fréquence, sans avoir cherché à en décoder le signal numérique. Afin d'étendre la gamme d'environnements dans laquelle le circuit d'interrogation fonctionne, les schémas plus complexes séparant l'alimentation de la bobine (jusqu'à 12 V) de la partie numérique (5 V) et effectuant une rétroaction de la fréquence reçue de l'étiquette sur la fréquence d'excitation sont fournis dans la datasheet de l'U2270B ([7], applications 2 et 3).

Finalement, nous n'avons jamais communiqué d'ordre à l'étiquette – option disponible en commandant la broche 6 (CFE) de l'U2270B sous contrôle d'un microcontrôleur afin de moduler la porteuse. Le circuit présenté ici mériterait d'être connecté à un microcontrôleur afin d'automatiser la phase de lecture de l'étiquette sans avoir à développer toute la procédure sous Octave présentée ici.

Adélie et Cézane sont deux Petits Lévriers Italiens, matricules 250269700030163 et 250269801093444 respectivement. Munies de puces d'identification radiofréquence, elles désiraient appréhender les technologies de transfert des informations afin d'en comprendre les implications pour leur vie privée. Elles ont été assistées dans ce projet par J.-M. Friedt, membre de l'association Projet Aurore de Besançon et ingénieur dans la société SENSeOR.

Références

[1] La traduction de *RFID tag* est inspirée de http://fr.wikipedia.org/wiki/Radio-identification

[2] RFID, instrument de sécurité ou de surveillance, misc **33**, septembre/octobre 2007

[3] J.R. Smith, A.P. Sample, P.S. Powledge, S. Roy & A. Mamishev, *A Wirelessly-Powered Platform for Sensing and Computation*, Springer Lecture Notes in Computer Science **4206** (2006), pp.495-506, ou K. Opasjumruskit, T. Thanthipwan, O. Sathusen, P. Sirinamarattana, P. Gadmanee, E. Pootarapan, N. Wongkomet, A. Thanachayanont & M. Thamsirianunt, *Self-Powered Wireless Temperature Sensors Exploit RFID Technology*, IEEE Pervasive Computing **5** (1) (2006), pp.54-61

[4] une plateforme opensource pour l'interrogation d'étiquettes HF : http://2008.rmll.info/CRESITAG-Plateforme-opensource.html. Cette présentation contient quelques références en français qui n'ont pas été consultées au cours de notre étude.

[5] R. Ryan, Z. Anderson & A. Chiesa, *Anatomy of a Subway Hack*, DEFCON 16 (2008), disponible à http://www-tech.mit.edu/V128/N30/subway/Defcon_Presentation.pdf

[6] K. Finkenzeller, *RFID Handbook – Fundamentals and Applications in Contactless Smart Cards and Identification, Second Edition*, John Wiley & Sons (2003)

[7] www.atmel.com/dyn/resources/prod_documents/doc4684.pdf

[8] www.datasheetcatalog.org/datasheet/Temic/mXyzutqu.pdf

[9] http://en.wikipedia.org/wiki/ISO_11784_%26_11785

[10] une présentation très complète sur le projet www.rfidiot.org, contenant notamment des informations sur les codages de l'information transmise par un RFID : http://www.blackhat.com/presentations/bh-europe-07/Laurie/Presentation/bh-eu-07-laurie.pdf

[11] http://www.emmicroelectronic.com/webfiles/ref/h4005_ds.pdf décrit en détail les "1" ajoutés dans le signal à décoder pour ne pas retrouver l'entête dans es données, ainsi que l'organisation des 128 bits du message.

[12] en.wikipedia.org/wiki/ISO_3166-1

[13] V. D. Hunt, A. Puglia & M. Puglia, *RFID – A guide to radio frequency identification*, John Wiley & Sons (2007) est un ouvrage ne contenant aucune information technique, que je ne mentionne ici que pour éviter au lecteur de gaspiller son argent.

[14] A. Graafstra, *RFID Toys : 11 Cool Projects for Home, Office and Entertainment*, Wiley ExtremeTech (2006), se contente d'exploiter des lecteurs et étiquettes commerciaux, sans en expliquer le fonctionnement. Le seul chapitre intéressant est le premier, disponible gratuitement sur amazon.com. À éviter donc.