Banque PT
PHYSIQUE & TECHNOLOGIE

Epreuve d'Informatique et Modélisation

Durée 4 h

Si, au cours de l'épreuve, un candidat repère ce qui lui semble être une erreur d'énoncé, d'une part il le signale au chef de salle, d'autre part il le signale sur sa copie et poursuit sa composition en indiquant les raisons des initiatives qu'il est amené à prendre.

L'usage de calculatrices est interdit.

AVERTISSEMENT

La **présentation**, la lisibilité, l'orthographe, la qualité de la **rédaction, la clarté et la précision** des raisonnements entreront pour une **part importante** dans **l'appréciation des copies**. En particulier, les résultats non justifiés ne seront pas pris en compte. Les candidats sont invités à encadrer les résultats de leurs calculs.

L'épreuve comporte deux parties qui peuvent être traitées indépendamment l'une de l'autre. Une première partie modélisation (durée conseillée 1H30) et une seconde informatique (durée conseillée 2 H30)

Remarques préliminaires importantes : il est rappelé aux candidat(e)s que

Les explications des phénomènes étudiés interviennent dans la notation au même titre que les développements analytiques et les applications numériques ; les résultats exprimés sans unité ne seront pas comptabilisés.

Tout au long de l'énoncé, les paragraphes en italiques ont pour objet d'aider à la compréhension du problème.

Tout résultat fourni dans l'énoncé peut être admis et utilisé par la suite, même s'il n'a pas été démontré par le (la) candidat(e).

Les applications numériques, effectuées sans calculatrice, pourront supporter des arrondis ou simplifications judicieux.

Ce problème traite de systèmes d'identification par radio fréquence (RFID). Il est constitué de deux parties totalement indépendantes : ***l'étude d'un système de type navigo*** *pour la 1ère partie. Aucune connaissance particulière sur les antennes n'est demandée.*

En 1948, Harry Stockman décrivit dans un article un moyen de communiquer grâce à la réflexion des ondes, annonçant ainsi les futurs systèmes autonomes communiquant par ondes électromagnétiques. Leur but est, par exemple, d'identifier des objets à distance, d'où le nom d'identification radio-fréquence (RFID : Radio Frequency Identification). Quelques années plus tard, les premiers systèmes d'identification utilisant les ondes réfléchies étaient développés.
Les progrès réalisés dans le domaine de l'électronique ont permis son développement et son intégration dans de nombreux domaines. Aujourd'hui, la RFID (du fait de son prix et de la taille des étiquettes) prend une place de plus en plus importante dans la vie courante, et d'un simple fonctionnement en mode tout-ou-rien, au stockage et au traitement d'informations, les applications couvrent des domaines allant de la télédétection (identification d'animaux, antivols, localisation...) aux transactions de la vie courante: par exemple dans les systèmes de contrôle d'accès aux transports en commun, type passe Navigo de la RATP pour le métro parisien. Les puces RFID tentent aujourd'hui de supplanter les codes à barres en jouant de leurs avantages, à savoir qu'il est possible d'écrire, d'effacer et de réécrire les données stockées dans une puce un grand nombre de fois, que leur portée peut être supérieure aux lecteurs optiques utilisés pour les codes à barres, et que la communication peut se faire à travers certains obstacles contrairement aux systèmes à lecture optique.

Un système RFID passif est composé de deux entités qui communiquent entre elles (Fig.1) :

- *Un TAG passif (dénommé TAG par la suite) ou étiquette intelligente (aussi appelé transpondeur), associé à l'élément à identifier. Il est capable de répondre à une demande venant d'un lecteur. Le TAG n'a pas d'alimentation de type batterie ou pile mais est autoalimenté par l'onde électromagnétique reçue.*
- *Une station de base ou lecteur RFID qui a pour mission d'identifier le TAG. Le lecteur envoie une onde électromagnétique en direction de l'élément à identifier, cette onde alimente le TAG qui peut alors communiquer avec le lecteur grâce à sa puce électronique interne. En retour, le lecteur reçoit l'information renvoyée par le TAG.*

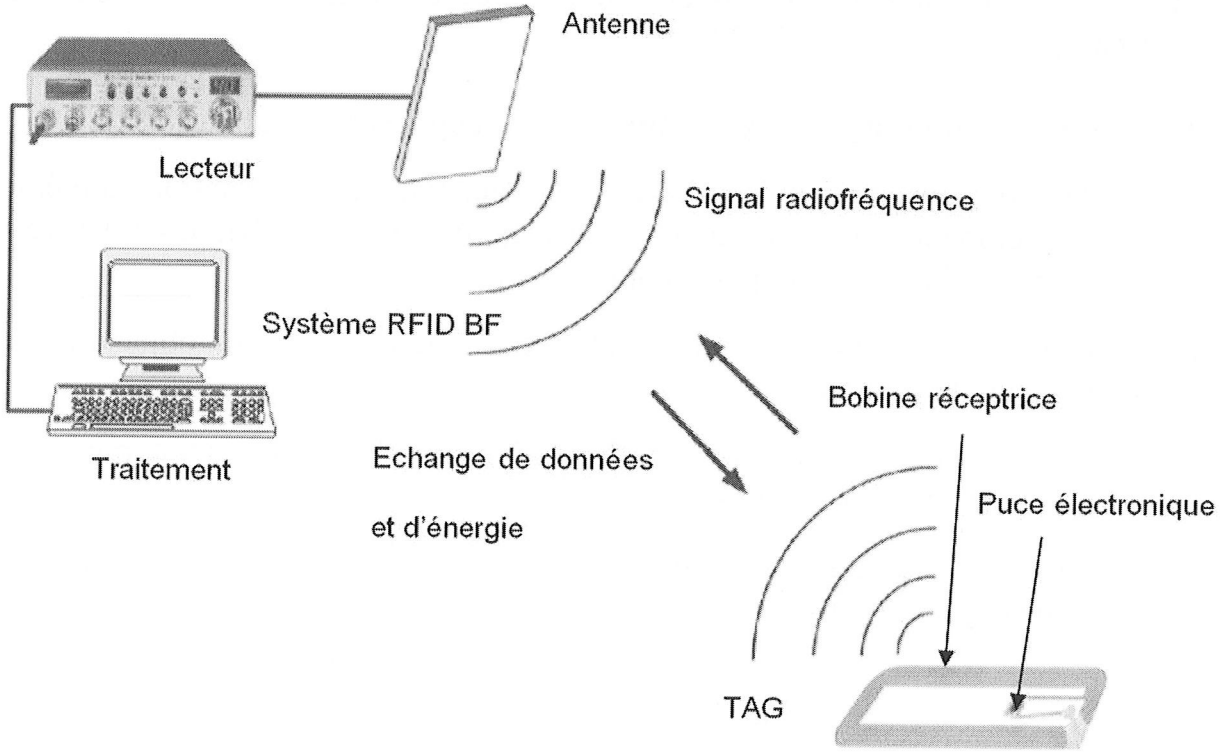


Figure 1

La figure 1 présente le fonctionnement général d'un système RFID. Le lecteur relié à une antenne émettrice agit en maître par rapport au TAG : si le TAG est dans la zone de lecture du lecteur, ce dernier l'active en lui envoyant une onde électromagnétique et entame la communication. Le TAG est quant à lui, constitué d'une antenne et d'une puce électronique qui module l'onde réémise vers le lecteur. En démodulant le signal reçu, le lecteur relié à une application interne récupère l'information pour la traiter, il est chargé de l'interface et de la gestion de l'identification des TAGs qui se présentent à lui.
Il existe plusieurs familles de systèmes RFID dont le principal critère de différentiation est la fréquence de fonctionnement. Les système RFID utilisent des bandes de fréquence à 125 kHz (bande BF), 13,56 MHz (bande HF), 860-960 MHz (bande UHF) et 2,45 GHz. On précise que la puissance rayonnée par l'antenne d'émission est une fonction croissante de la fréquence est varie en $\frac{1}{\lambda^4}$.

Bande	*Fréquence*	*Portée*	*Pouvoir de Pénétration dans un conducteur*
BF	*125 kHz*	+	++++
HF	*13.56 MHz*	++	+++
UHF	*860-960 MHz*	+++	++
UHF	*2,45 GHz*	++	+

+ : faible , ++ : Assez bon , +++ : Bon , ++++ : Excellent

En fonction des différentes fréquences, les principes physiques mis en œuvre ne sont pas les mêmes et le problème aborde certains aspects de la communication.

PREMIERE PARTIE

Système RFID à 13.56 MHz en couplage magnétique

I.A Les fréquences utilisées en RFID

I.A1. Pour quelle expérience majeure le physicien Hertz (1857-1894) est-il resté célèbre ?

I.A2. Que signifient bande HF et bande UHF ? Pourquoi dit-on que la fréquence 2,45 GHz appartient aux microondes?

I.A3. Pourquoi les ondes UHF portent plus loin que les ondes HF et les ondes BF ?

I.A4. Pourquoi le pouvoir de pénétration des ondes dans un conducteur augmente quand la fréquence baisse ?

I.A5. Pourquoi tous les systèmes RFID HF utilisent-ils la même fréquence de 13,56 MHz ?

I.B Modélisation d'un système RFID à 13.56 MHz : la carte à puce sans contact.

On considère une bobine b_1 circulaire d'axe (Oz) située dans le plan (Oxy), de rayon a, comprenant N_1 spires identiques et supposées confondues (toutes de même diamètre) réalisées en fil conducteur de diamètre négligeable et parcourue par le courant i_1 sinusoïdal de fréquence f : $i_1 = I_1\sqrt{2}\sin(\omega t)$. *Les dimensions caractéristiques (a, OM,...) sont de l'ordre de 10 cm. On rappelle que :* $\mu_0 = 4\pi \cdot 10^{-7}\,H \cdot m^{-1}$.

Tournez la page S.V.P.

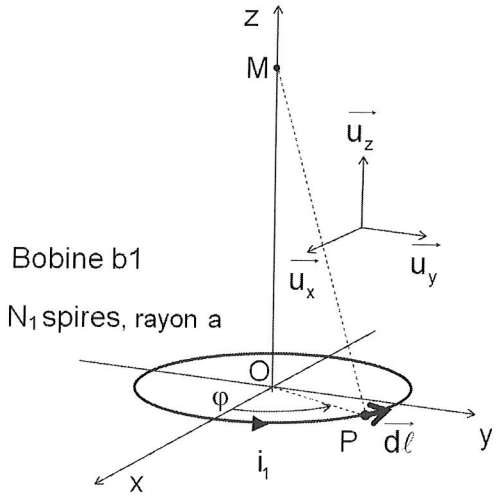


Figure 2

I.B1. Rappeler les conditions d'application de l'approximation des régimes quasi stationnaires (ARQS). Cette approximation s'applique-t-elle ici ?

Soit M un point de l'axe (Oz) tel que $\overrightarrow{OM} = z \cdot \vec{u_z}$ *et* $z \geq 0$.

I.B2. Quelle est la direction du champ magnétique $\vec{B_1}(M,t)$ en M? On note $B_{1eff}(a,z)$ la valeur efficace de la norme de $\vec{B_1}(M,t)$. Pour $I_1 = 20$ mA, on a représenté sur la figure 3 les courbes de niveau de $B_{1eff}(a,z)$ pour $0 \leq a \leq 0.2$ m et pour $0 \leq z \leq 0.2$ m. A partir de ces courbes de niveau et sans calcul supplémentaire représenter B_{1eff} en fonction de a pour $z = z_0$. Préciser les valeurs remarquables.

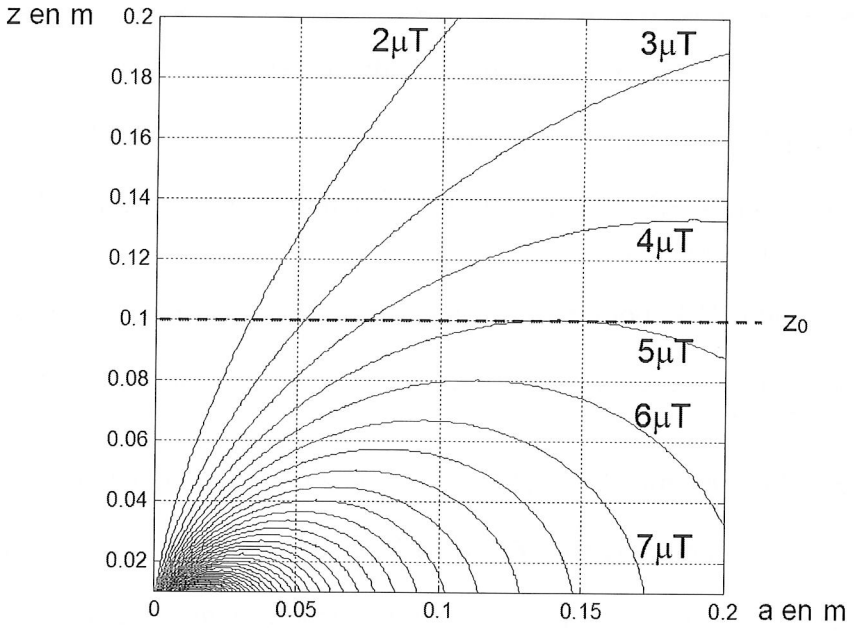


Figure 3

Application numérique : pour $N_1 = 100$ spires, $I_1 = 20$ mA, $z_0 = 10$ cm déterminer la valeur maximale de B_{1eff}.

I.B3. Donner une expression du flux propre Φ_1 de la bobine b_1 faisant intervenir une intégrale simple. On ne cherchera pas à calculer Φ_1.

I.B4. Rappeler la définition de l'inductance propre L_1 de b_1.

On considère maintenant une bobine b_2 circulaire d'axe (Oz) située dans un plan parallèle au plan (Oxy) à $z = d$, de rayon b et de surface $S_2 = \pi \cdot b^2$, comprenant N_2 spires identiques et confondues (toutes de même diamètre) réalisées en fil conducteur de diamètre négligeable. Les deux bobines b_1 et b_2 sont couplées magnétiquement et forment un transformateur à air. On note L_2 l'inductance propre de b_2 et M l'inductance mutuelle entre b_1 et b_2 ($M = k\sqrt{L_1 L_2} > 0$, où k est le coefficient de couplage entre les deux bobines). La bobine b_2 est connectée à un circuit électrique comportant une résistance R_L en série : c'est le modèle du transpondeur (TAG). La bobine b_1 est alimentée par une tension u_1 et constitue l'antenne associée au lecteur RFID, R_1 et R_2 modélisent les résistances des bobines b_1 et b_2 et R_L modélise la résistance d'entrée de la puce électronique du TAG. En modulant R_L, le TAG communique avec le lecteur. (Le TAG est donc modélisé par L_2, R_2 et R_L qui est variable).

Pour les applications numériques prendre : d = 10 cm, N_1 = 100 spires, N_2 = 4 spires, S_2 = 30 cm^2, L_1 = 40 µH, L_2 = 6.3 µH, R_2 = 10 Ω, R_L = 30 kΩ ; k = 0,63% (soit M = 0,1 µH); f_0 = 13,56 MHz, I_1 = 20 mA.

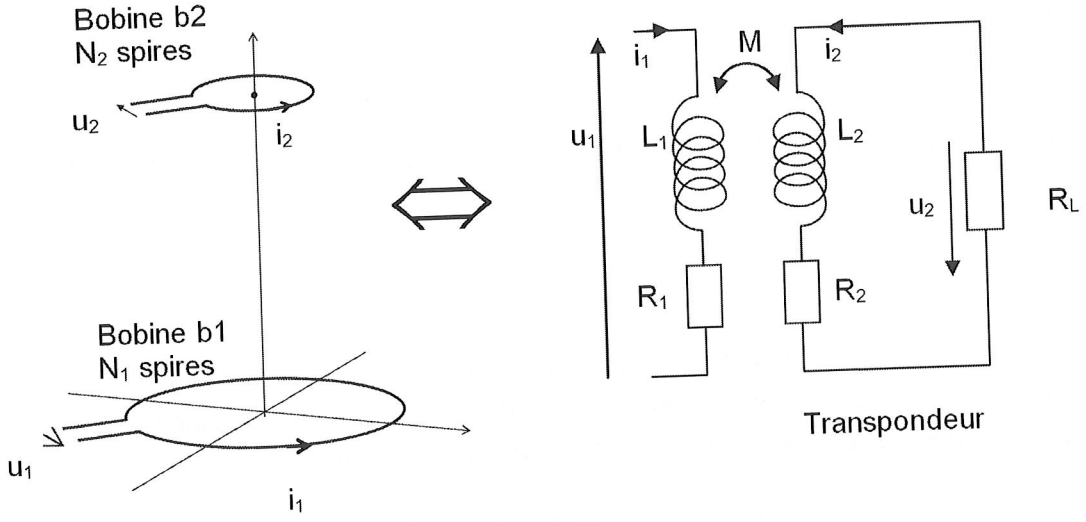


Figure 4

Tournez la page S.V.P.

I.B5. Donner une approximation de M en considérant que $b \ll a$ et que $b \ll d$. En exploitant le résultat de la question I.B2, et pour des nombres de spires N_1 et N_2 et une distance d entre les deux bobines fixés, il existe une valeur optimale du rayon a de la bobine b_1 maximisant la mutuelle M. Application numérique : déterminer la valeur optimale de a.

I.B6. Exprimer les tensions u_1 et u_2 en fonction de R_1, R_2, L_1, L_2, M, i_1, i_2, $\frac{di_1}{dt}$ et $\frac{di_2}{dt}$.

Aux grandeurs temporelles, u_1, u_2, i_1 *et* i_2 *on associe les grandeurs complexes* $\underline{U_1}$, $\underline{U_2}$, $\underline{I_1}$ *et* $\underline{I_2}$. *La seule entrée du montage est la tension* u_1 (*ou* $\underline{U_1}$) ; u_2, i_1 *et* i_2 *sont donc des inconnues* (ou $\underline{U_2}$, $\underline{I_1}$ *et* $\underline{I_2}$).

I.B7. Ecrire trois équations liant les tensions complexes $\underline{U_1}$ et $\underline{U_2}$ et les courants $\underline{I_1}$ et $\underline{I_2}$.

I.B8. Mettre $\underline{U_1}$ et $\underline{U_2}$ sous la forme : $\underline{U_1} = \underline{Z_{11}} \cdot \underline{I_1}$ et $\underline{U_2} = \underline{Z_{21}} \cdot \underline{I_1}$. Montrer en particulier que $\underline{Z_{21}} = \dfrac{-jMR_L\omega}{R_L + R_2 + jL_2\omega}$ et que $\underline{Z_{11}} = a + jb + \dfrac{(M\omega)^2}{R_L + R_2 + jL_2\omega}$. Préciser les expressions des coefficients réels a et b.
Application numérique : donner l'ordre de grandeur de la tension efficace de u_2.

On insère un condensateur C_2 *en parallèle à* R_L *tel que* $L_2C_2\omega_0^{\,2} = 1$ *et* $\omega_0 = 2\pi f_0$. *On note* $Q_2 = 1/(R_2C_2\omega_0)$ *et* $Q_L = R_L/(L_2\omega_0)$; *les valeurs numériques correspondantes sont* $C_2 = 22$ pF, $Q_2 = 53{,}3$ *et* $Q_L = 55{,}9$.

I.B9. En négligeant R_2 devant R_L, déterminer la nouvelle expression de l'impédance $\underline{Z'_{21}}$. Mettre sous la forme :

$$\underline{Z'_{21}} \approx \frac{-R_{21}}{1 + jQ_T \cdot \left(\dfrac{\omega}{\omega_0} - \dfrac{\omega_0}{\omega}\right)}$$

Exprimer la résistance R_{21} ainsi que le facteur de qualité Q_T équivalent du transpondeur en fonction de Q_L et de Q_2. Conclure sur l'intérêt du condensateur C_2. Application numérique : déterminer les ordres de grandeur des valeurs efficaces de i_2 et de u_2.

Si le champ électromagnétique a une amplitude trop faible, l'énergie reçue par le transpondeur n'est pas suffisante pour un fonctionnement normal. Ainsi, l'alimentation de la puce ne se fait correctement que si la tension vérifie $U_{2eff} > 4{,}6\,V$.

I.B10. En appliquant la loi de Faraday et avec les approximations adéquates, indiquer quel est l'ordre de grandeur de la valeur numérique du champ efficace B_{min} nécessaire au niveau de la bobine b_2.

I.B11. Faire un bilan de puissance sur le schéma de la figure 4 en modifiant R_L par une impédance $\underline{Z_L} = \dfrac{R_L}{1 + jQ_L\dfrac{\omega}{\omega_0}}$.

La bobine b_1 est en fait reliée à un condensateur, C_1 tel que $L_1C_1\omega_0^2 = 1$. L'ensemble est alimenté par une source de tension de fem e_G et de résistance interne R_G (Fig. 5). Pour les applications numériques prendre $R_G = 50\,\Omega$.

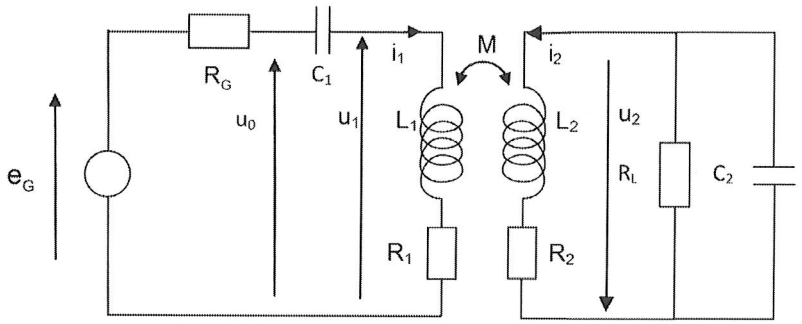


Figure 5

En plus de la valeur 30 kΩ, R_L *peut prendre les deux autres valeurs particulières suivantes :* 0 Ω *et l'infini.*

I.B12. Montrer que pour $\omega = \omega_0$ la tension $\underline{U_0}$ peut s'écrire $\underline{U_0} = \left(R_1 + \underline{Z}(\omega_0)\right)\cdot \underline{I_1}$ où l'impédance $\underline{Z}(\omega_0)$ s'écrit :

$$\underline{Z}(\omega_0) = \frac{(M\omega_0)^2}{\dfrac{R_L}{1 + j\omega_0 R_L C_2} + R_2 + jL_2\omega_0}$$

Application numérique : déterminer l'ordre de grandeur de $\underline{Z}$ (module et argument) lorsque $R_L = 0$ puis lorsque $R_L = \infty$. Montrer comment réaliser en pratique ces deux conditions avec des interrupteurs commandables.

Le transpondeur communique avec la base émettrice en modulant la résistance R_L entre les deux valeurs $R_L = 0$ et $R_L = \infty$.

I.B13. Montrer alors que l'amplitude de la tension u_0 contient l'information délivrée par la puce. Conclusion.

Tournez la page S.V.P.

Deuxième partie : Informatique. Modélisation d'un système de titres de transport sans contact

La première partie a permis de montrer comment la puce RFID peut être alimentée à distance par le lecteur, et comment un signal peut être transmis de l'une à l'autre. L'objectif des parties suivantes est de mettre en place quelques algorithmes visant à :

- simuler la récupération d'un message binaire à partir de la tension captée par le lecteur (Partie 1) ;
- contrôler l'intégrité du message récupéré et corriger les erreurs éventuelles (Partie 2),
- déterminer si un voyageur est autorisé ou non à franchir un point de contrôle (Partie 3),
- traiter les informations recueillies par le système afin d'améliorer le service (Partie 4).

Ces quatre parties sont indépendantes.

Les algorithmes demandés seront réalisés, au choix, dans le langage Python ou le langage Scilab. **Le candidat doit précsier en tête de partie le langage choisi et s'y tenir.**On supposera que tout module nécessaire à l'utilisation des fonctions usuelles (pi, sin...) a été importé.

Simulation numérique de la démodulation d'amplitude par le lecteur

La première partie a montré que la puce et le lecteur peuvent tous deux recevoir des messages contenus dans des tensions alternatives d'amplitudes variables. Nous allons maintenant modéliser une telle tension puis simuler numériquement le comportement d'un dispositif permettant d'en extraire une information binaire.

La tension $e(t)$ reçue par le lecteur peut être modélisée par une sinusoïde de fréquence $f = 13{,}56$ MHz dont l'amplitude E_0 est modulée par le signal binaire $b(t)$ transmis par la puce :

$$e(t) = E_0(t)\sin(2\pi f t) \text{ avec } \begin{cases} E_0(t) = E_{\text{min}} = 1{,}3 \text{ V lorsque } b(t) = 0 \\ E_0(t) = E_{\text{max}} = 1{,}5 \text{ V lorsque } b(t) = 1 \end{cases}$$

La puce transmet un nouveau bit toutes les 16 périodes de la porteuse. A titre d'illustration, la **Figure 1** représente la tension $e(t)$ correspondant à la séquence de bits (0, 1, 0). La tension est en volts et le temps en secondes.

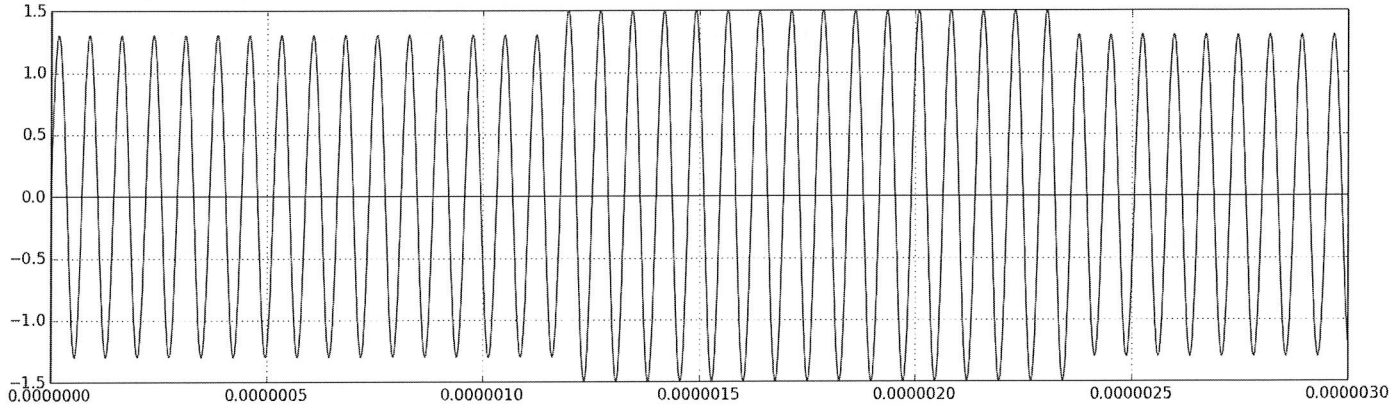


Figure 1 : tension d'entrée du démodulateur (sinusoïde modulée en amplitude par un signal binaire)

La simulation porte sur un intervalle de temps $[0, T_{\text{max}}]$. Le temps sera représenté numériquement par une liste de N instants régulièrement espacés que l'on écrira, compte tenu des conventions de numérotation propres à chaque langage, $T = [t_0=0, t_1, ..., t_{N-1}=T_{\text{max}}]$ sous Python ou $T = [t_1=0, ...t_N=T_{\text{max}}]$ sous Scilab.

Q1. Écrire une fonction `init_T(Tmax,dt)` prenant pour arguments la durée `Tmax` de la simulation et le pas de temps `dt` et retournant la liste `T`. Si `Tmax` n'est pas multiple de `dt`, on arrondira la durée de la simulation au multiple entier de dt immédiatement supérieur.

La tension d'entrée $e(t)$ sera de même représentée par une liste `E` telle que `E[i]`$=e(t_i)$.

Q2. Écrire une fonction `init_E(T,f)` prenant pour arguments la liste `T` des instants de la simulation et la fréquence `f` de la porteuse et retournant la liste des valeurs $e(t_i)$ de la tension e aux instants `T[i]`, d'après la définition de $e(t)$ et pour le message binaire (0,1,0) (on considèrera que la simulation ne se poursuit pas au-delà).

Pour récupérer l'information binaire contenue dans une telle tension, il faut en extraire l'amplitude. On utilise pour cela le dispositif de la **Figure 2**, appelé **détecteur d'enveloppe**.

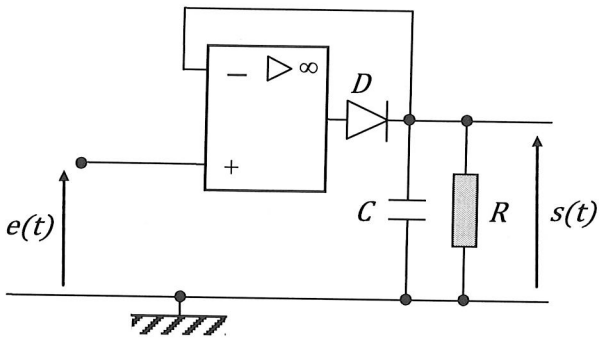


Figure 2 : détecteur d'enveloppe

La modélisation du comportement de ce montage permet d'exprimer la tension de sortie $s(t)$ en fonction de la tension d'entrée $e(t)$ de la façon suivante (τ est la constante de temps du circuit RC) :

1. Si la diode est *bloquée*, alors $\frac{ds}{dt}(t) + \frac{1}{\tau}s(t) = 0$ et il faut vérifier que $s(t) > e(t)$
2. Si la diode est *passante*, alors $s(t) = e(t)$ et il faut vérifier que $\frac{ds}{dt}(t) + \frac{1}{\tau}s(t) > 0$

Autrement dit, la diode peut prendre deux états (bloquée ou passante), chacun soumis à une condition (inégalité) et muni d'une équation d'évolution (égalité) ; lorsque la condition cesse d'être vérifiée, la diode change d'état. Nous allons utiliser ces équations pour simuler numériquement l'évolution de la tension $s(t)$. Pour cela, on propose d'utiliser une variable (par exemple booléenne) indiquant l'état de la diode et, à chaque pas de temps t_i :

- de calculer $s(t_{i+1})$ en utilisant l'équation correspondant à la valeur de la variable à l'instant t_i,
- *puis* de tester la condition correspondante à partir de la valeur de $s(t_{i+1})$ et, si elle n'est plus vérifiée, de mettre à jour la variable.

Le résultat est stocké dans une liste `S` avec `S[i]` $= s(t_i)$.

Q3. Donner une approximation de $\frac{ds}{dt}(t_i)$ en fonction de $s(t_i)$, $s(t_{i+1})$ et $\Delta t = t_{i+1} - t_i$ en utilisant la formule d'Euler explicite. En déduire, dans le cas où la diode est *bloquée* à l'instant t_i, la relation de récurrence donnant $s(t_{i+1})$ en fonction de $s(t_i)$, τ et Δt.

Dans le cas où la diode est *passante* à l'instant t_i, on ne peut pas utiliser cette formule pour tester l'état de la diode à t_{i+1}. On utilise donc l'approximation *d'Euler arrière* ou *Euler implicite*, qui consiste à utiliser la même formule mais pour approcher $\frac{ds}{dt}(t_{i+1})$ au lieu de $\frac{ds}{dt}(t_i)$.

Q4. Pourquoi ne peut-on pas utiliser la formule d'Euler explicite pour effectuer le test ici ? Donner, en utilisant la démarche proposée, une condition portant sur $s(t_{i+1})$, $s(t_i)$, τ et Δt permettant de déterminer si la diode se bloque ou non à l'instant t_{i+1}.

Q5. Écrire alors une fonction `solve(T,E,tau)` prenant pour arguments la liste `T` des instants de la simulation, la liste `E` des tensions d'entrée et la constante de temps `tau` et retournant la liste `S` des tensions de sortie. Les conditions initiales seront prises nulles et, si nécessaire, l'état initial de la diode sera supposé passant.

Les **Figures 3, 4 et 5** donnent les résultats (entrées et sorties numériques) obtenus pour trois pas de temps différents choisis parmi 1 ns, 10 ns et 100 ns. Sur ces trois graphes, la tension d'entrée *avant discrétisation* est celle de la **Figure 1** et la constante de temps du circuit est $\tau = 1$ µs. Seul le pas de temps change d'une simulation à l'autre.

Tournez la page S.V.P.

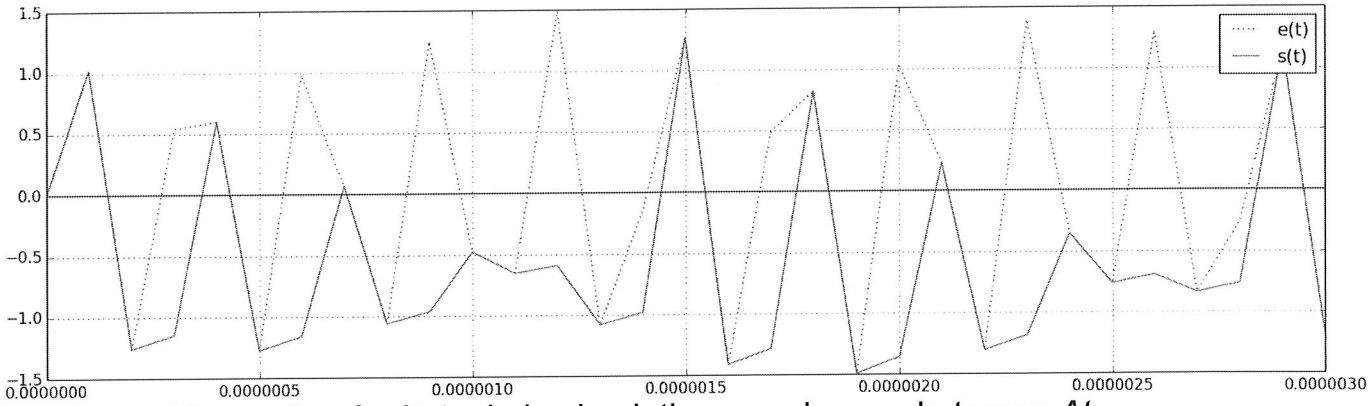


Figure 3 : résultats de la simulation pour le pas de temps Δt_1

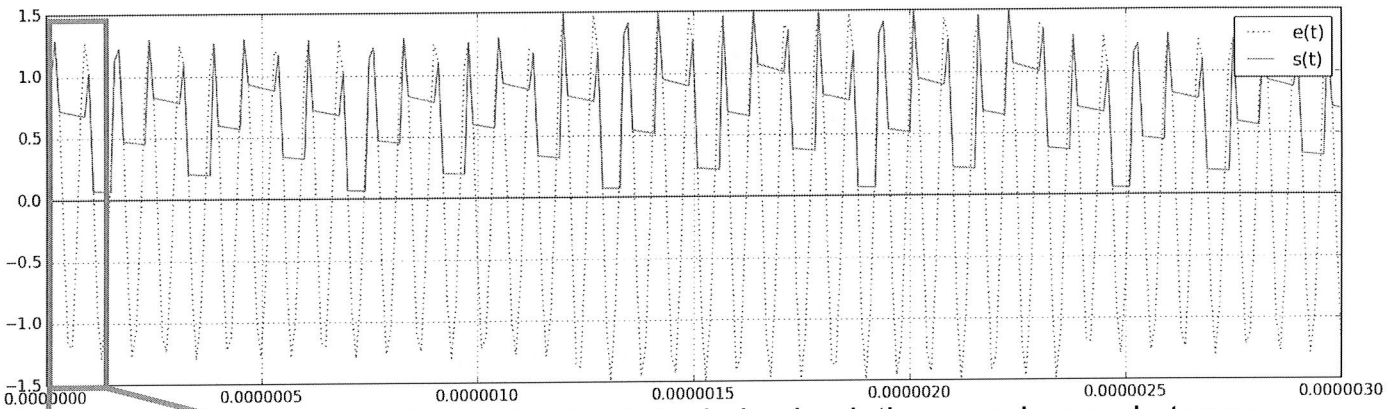
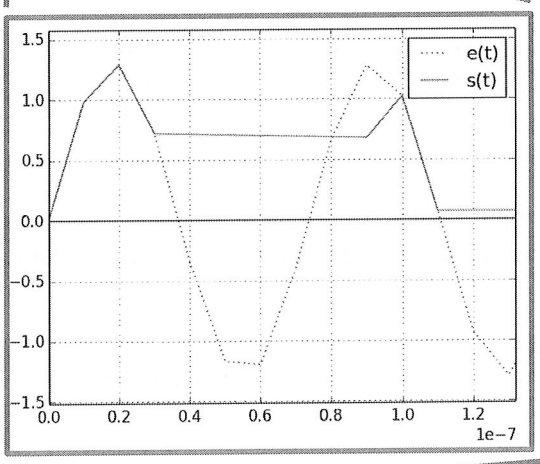
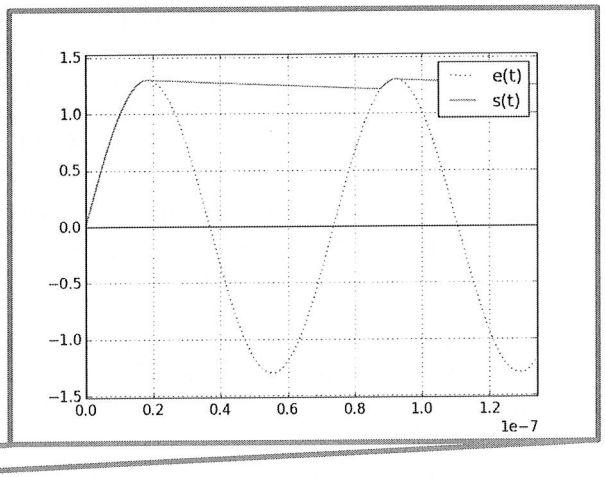


Figure 4 : résultats de la simulation pour le pas de temps Δt_2 (avec zoom à l'origine)





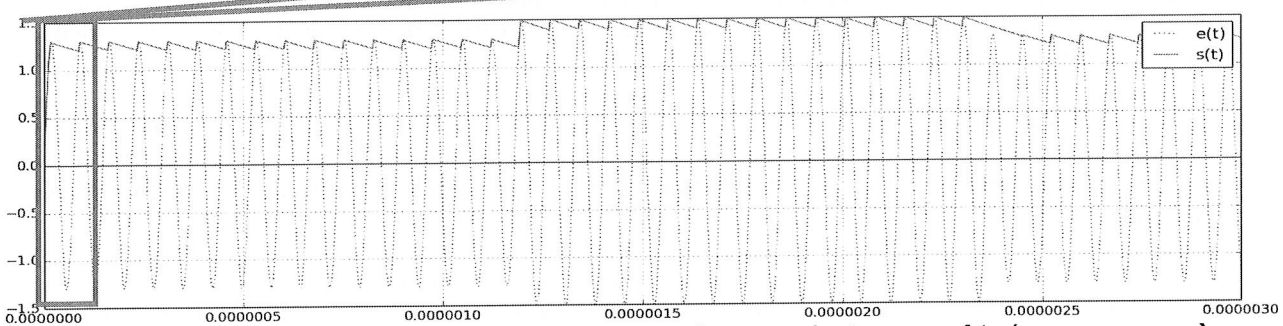


Figure 5 : résultats de la simulation pour le pas de temps Δt_3 (avec zoom à l'origine)

Q6. **Indiquer la valeur du pas de temps (1, 10 ou 100 ns) correspondant à chacune de ces trois simulations, en justifiant vos réponses.**

Q7. Expliquer en quelques phrases les causes des différences obtenues entre les trois résultats $s(t)$. Repérer en particulier les instants auxquels la diode change d'état. Que constatez-vous ?

Pour distinguer l'état "haut" de l'état "bas" de la tension $s(t)$ et ainsi extraire les 0 et les 1 du message binaire transmis par modulation, il est nécessaire de déterminer un seuil séparant les deux niveaux.

Q8. Indiquer, pour chacun des trois résultats, s'il est possible d'identifier un tel seuil, et donc si la récupération du message binaire semble réalisable *si l'on se base uniquement sur ce résultat*. Conclure sur le critère que doit respecter le pas de temps d'une simulation temporelle pour que les résultats de celle-ci aient une chance d'être pertinents.

Vérification de l'intégrité des données et correction des erreurs

Le signal transmis par une liaison RFID 13,56 MHz peut être perturbé par toutes sortes de facteurs pouvant provoquer des erreurs dans les données : autres signaux électromagnétiques, masses métalliques, imperfections du matériel électronique... En pratique, il est donc indispensable de pouvoir détecter ces erreurs et, dans la mesure du possible, les corriger sans que cela ne nécessite une nouvelle transmission ; l'objet de cette partie est de mettre en place quelques algorithmes dans ce but.

2.1. Bit de parité

Une technique simple et très répandue pour s'assurer qu'une donnée binaire sera lue correctement par son récepteur est de lui adjoindre un **bit de parité**, égal par définition à :

- 0 si la donnée contient un nombre pair de 1 (et, donc, si ses bits sont de somme paire),
- 1 si la donnée contient un nombre impair de 1 (et, donc, si ses bits sont de somme impaire).

Après réception de la donnée, le récepteur recalcule le bit de parité et le compare à celui que l'émetteur lui a adressé. Si la donnée n'a pas été altérée lors de la transmission, alors les deux bits de parité sont forcément identiques.

Q9. Donner les bits de parité associés aux représentations binaires des entiers 5, 16 et 37.

Q10. Écrire une fonction `parite(bits)` prenant pour argument une liste `bits` constituée d'entiers valant 0 ou 1 et retournant l'entier 0 ou 1 correspondant à son bit de parité.

Les techniques de vérification les plus simples consistent à découper la donnée en blocs et à joindre un bit de parité à chaque bloc. Par exemple, certains protocoles transmettent sept bits de données pour un bit de parité.

Q11. Donner un exemple d'erreur n'étant pas détectable par cette technique. Si une erreur a été détectée, est-il possible de la corriger sans retransmettre la donnée ?

2.2. Code de Hamming

Le code de Hamming est un exemple d'utilisation des bits de parité pour détecter et corriger des erreurs. Nous nous intéressons ici au code dit (7,4), ainsi appelé car il consiste à joindre trois bits de parité à quatre bits de données, ce qui donne un message d'une longueur totale de sept bits. Ces trois bits de parité sont définis ainsi : si la donnée s'écrit (d_1, d_2, d_3, d_4) avec $d_i = 0$ ou 1, alors :

- p_1 est le bit de parité du triplet (d_1, d_2, d_4),
- p_2 est le bit de parité du triplet (d_1, d_3, d_4),
- p_3 est le bit de parité du triplet (d_2, d_3, d_4).

Le message encodé, que l'on transmet, s'écrit alors comme suit : (p_1, p_2, d_1, p_3, d_2, d_3, d_4).

Q12. Écrire une fonction `encode_hamming(donnee)` prenant pour argument une liste `donnee` de quatre bits (représentés par des entiers valant 0 ou 1) et retournant une liste de bits contenant le message encodé. On pourra appeler la fonction `parite(bits)` précédemment définie.

Le contrôle après réception d'un message ainsi encodé est relativement simple. On pourrait naturellement recalculer les trois bits de parité de la donnée et les comparer aux valeurs transmises, mais la technique proposée par Hamming est de calculer les trois *bits de contrôle* suivants, notés (c_1, c_2, c_3), à partir du *message complet* (données et bits supplémentaires), noté (m_1, ..., m_7) :

- c_1 est le bit de parité de l'ensemble (m_4, m_5, m_6, m_7),
- c_2 est le bit de parité de l'ensemble (m_2, m_3, m_6, m_7),
- c_3 est le bit de parité de l'ensemble (m_1, m_3, m_5, m_7).

On montre que si le message a bien été encodé selon les règles précédentes et n'a pas été altéré, alors les trois bits de contrôle doivent être à 0. Si ce n'est pas le cas, alors il y a eu une erreur ; l'intérêt de la technique de Hamming est que

Tournez la page S.V.P.

dans le cas particulier où l'erreur est unique, le mot de contrôle donne la représentation binaire de la position de cette erreur en numérotant à partir de 1. Par exemple, si $(c_1, c_2, c_3) = (0,1,1)$, alors l'erreur porte sur le troisième bit du message. Il suffit ainsi d'inverser ce bit (le mettre à 1 s'il est à 0, et inversement) pour corriger l'erreur.

La donnée décodée est alors constituée des quatre bits (d_1, d_2, d_3, d_4) qui se trouvent respectivement en positions 3, 5, 6 et 7 (toujours en numérotant à partir de 1) conformément à la description de l'encodage donnée ci-dessus.

Q13. Écrire une fonction `decode_hamming(message)` prenant pour argument une liste de sept bits et retournant une liste de quatre bits contenant la donnée décodée. En cas d'erreur, on affichera à l'écran un avertissement indiquant la position du bit affecté et on effectuera la correction. On supposera dans cette question que s'il y a une erreur, alors elle est unique.

Q14. Déterminer le codage de Hamming de la donnée `1011`, puis la donnée décodée par l'algorithme dans l'hypothèse où les deux premiers bits du message codé ont été incorrectement transmis. Quel a été l'effet de la "correction" sur la donnée dans ce cas ?

Q15. Sans coder, proposer un moyen simple de différencier une double erreur d'une erreur unique au moyen d'un bit de parité supplémentaire et expliquer comment cela permet d'éviter le problème mis en évidence à la question précédente. On s'appuiera sur les techniques introduites dans cette partie. On ne demande pas d'essayer de corriger la double erreur.

Utilisation des données de la puce pour autoriser ou non le passage

Lorsqu'un utilisateur présente son titre de transport face au lecteur d'un point de contrôle, la puce et le lecteur s'identifient mutuellement, puis le lecteur récupère les données de la puce permettant de déterminer si le passage est autorisé ou non. A l'issue de cette récupération, l'ordinateur auquel est relié le lecteur dispose d'un fichier texte `0001.txt` semblable à l'exemple ci-dessous :

```
49987654
1, 3, 2015-08-31
2014-10-29, 08:34:15, 4568
2014-10-28, 20:21:48, 365
2014-10-28, 18:47:54, 987
```

(identifiant du titre de transport)
(première et dernière zone de validité, date de fin de validité)
(dates, heures et identifiants des lieux des trois derniers passages)

En d'autres termes, ce fichier possède toujours *exactement* la structure suivante :

- la première ligne contient un entier servant à identifier le titre de transport,
- le réseau de transport est divisé en plusieurs zones numérotées, et le titre n'est valide que dans un ensemble de zones contigües ; la seconde ligne contient les bornes de l'intervalle dans lequel le titre est valide (*ici, il s'agit des zones 1 à 3 incluses*) ainsi que la date de fin de validité du titre au format *aaaa-mm-jj* (*ici, il s'agit du 31 août 2015*),
- les trois lignes suivantes contiennent des données relatives aux trois derniers passages effectués à l'aide du titre : date, heure (au format *hh:mm:ss* sur 24 heures) et identifiant entier du point de passage (gare, arrêt...).

On donne le bloc d'instructions utilisé pour lire les deux premières lignes de ce fichier. On précise qu'il n'est pas nécessaire de connaître toutes les syntaxes relatives à la manipulation des chaînes pour traiter les questions suivantes.

sous Python

```
fichier = open('0001.txt')
lignes = fichier.readlines()
fichier.close()

# Ligne 1: recuperation de l'identifiant du titre
id_titre = int(lignes[0])

# Ligne 2 : recuperation des donnees du titre de transport
donnees_titre = lignes[1].rstrip('\n').split(',')
zones = [ int(donnees_titre[0]), int(donnees_titre[1]) ]
ch_date_fin = donnees_titre[2].split('-')
```

```
date_fin = [ int(ch_date_fin[0]), int(ch_date_fin[1]), int(ch_date_fin[2]) ]
```

Sous Scilab

```
fichier = mopen('0001.txt')
lignes = mgetl(fichier)
mclose(fichier)

// Ligne 1: recuperation de l'identifiant du titre
id_titre = strtod(lignes(1))

// Ligne 2 : recuperation des donnees du titre de transport
donnees_titre = strsplit(lignes(2),',')
zones = [ strtod(donnees_titre(1)), strtod(donnees_titre(2)) ]
ch_date_fin = strsplit(donnees_titre(3),'-')
date_fin = [ strtod(ch_date_fin(1)), strtod(ch_date_fin(2)), strtod(ch_date_fin(3)) ]
```

Q16. Donner les types et les valeurs des variables `id_titre`, `zones` et `date_fin` à l'issue de ces instructions pour le fichier `0001.txt` donné ci-dessus.

On souhaite placer le contenu des trois dernières lignes dans un tableau d'entiers nommé **`passages`**, dont chaque ligne corresponde à un passage (dans l'ordre dans lequel ils apparaissent dans le fichier) et dont les colonnes soient définies comme suit (le premier chiffre donné est l'indice sous Python, le deuxième est l'indice sous Scilab) :

Indice	0/1	1/2	2/3	3/4	4/5	5/6	6/7
Contenu	Année	Mois	Jour	Heures	Minutes	Secondes	Point de passage

Q17. Écrire le bloc d'instructions à exécuter à la suite des opérations précédentes pour construire le tableau `passages` à partir des lignes 3, 4 et 5 contenues dans la liste `lignes`.

Les données relatives au lecteur sont décrites par les variables suivantes (*on utilise une initiale majuscule pour bien différencier les données du lecteur de celles du titre ; le type est indiqué entre parenthèses*) :

- **`Zone`** (*entier*) : indique la zone dans laquelle se trouve le lecteur,
- **`Id_Point`** (*entier*) : indique l'identifiant du point de passage où se trouve le lecteur,
- **`Liste_noire`** (*liste d'entiers*) : contient les identifiants des titres ayant été déclarés perdus, volés ou détériorés par leurs propriétaires, et devant donc être refusés,
- **`Maintenant`** (*liste de six entiers*) : contient la date et l'heure au format ci-dessus [année, mois, jour, heures, minutes, secondes].

Le passage doit être autorisé si les conditions suivantes sont *toutes* vérifiées :

- l'identifiant du titre n'est pas dans la liste noire du lecteur,
- la zone du lecteur appartient à l'intervalle de validité du titre,
- la date du jour est antérieure à la date de fin de validité du titre,
- si l'une des trois dernières validations a été effectuée au même point de passage que celui où est installé le lecteur, elle doit avoir été effectuée il y a plus de 450 secondes (ceci afin de décourager l'utilisation frauduleuse d'un même titre par plusieurs voyageurs).

Enfin, lorsqu'un passage est refusé, un message apparaît sur un afficheur LCD pour donner la raison du refus. Cet afficheur possède une seule ligne et il faut donc définir des priorités au cas où plusieurs des conditions ci-dessus ne seraient pas remplies. L'ordre de priorité et les messages correspondants sont donnés ci-dessous :

1. `"Titre refusé"` si l'identifiant est dans la liste noire,
2. `"Non valide dans cette zone"` si le lecteur est hors des zones de validité du titre,
3. `"Titre expiré"` si la date de fin de validité du titre est dépassée,
4. `"Titre déjà validé"` si le titre a déjà été validé dans le même lieu il y a moins de 450 secondes.

Q18. Écrire une fonction `estAvant(date1,date2)` prenant pour arguments deux dates au format `[année, mois, jour]` (donc sous forme de listes de trois entiers chacune) et retournant `True` si `date1` est antérieure ou égale à `date2`, et `False` sinon.

Q19. Écrire une fonction `nbSecondesEntre(heure1,heure2)` prenant pour arguments deux horaires au format `[heures, minutes, secondes]` (donc sous forme de listes de trois entiers chacun) et retournant le nombre de secondes séparant les deux instants. Le résultat devra être positif si heure1 est *postérieure* à heure2.

Q20. Écrire alors une fonction `testPassage`, dont les arguments sont à préciser, retournant la valeur `True` si le passage est autorisé et la valeur `False` sinon et, dans ce dernier cas, affichant à l'écran le message correspondant aux règles de priorité ci-dessus. On pourra utiliser toutes les variables définies dans cette partie et appeler les fonctions définies dans les deux questions précédentes.

Exploitation des données enregistrées par le système

A chaque fois qu'un point de contrôle est franchi, le système collecte des données relatives au passage : lieu, date, heure... Ces données sont ensuite enregistrées dans la base de données du transporteur, qui comporte trois tables :

- la table **`passages`** dédiée aux passages des voyageurs, constituée des champs :
  - **`date`** qui contient la date du passage au format *aaaa-mm-jj*,
  - **`heure`** qui contient l'heure du passage au format *hh:mm:ss*,
  - **`id_point`** qui est l'identifiant (entier) du point de passage,
  - **`id_titre`** qui est l'identifiant (entier) du titre de transport utilisé ;

- la table **`points`** dédiée aux points de passage, constituée des champs (entiers) :
  - **`id`** qui est l'identifiant du point de passage (clé primaire),
  - **`zone`** qui est le numéro de la zone où se trouve le point de passage,
  - **`ligne`** qui est le numéro de la ligne sur laquelle se trouve le point de passage ;

- et la table **`titres`** dédiée aux titres de transport, constituée des champs (entiers) :
  - **`id`** qui est l'identifiant du titre de transport (clé primaire),
  - **`zone_min`** qui est la plus petite zone couverte par le titre,
  - **`zone_max`** qui est la plus grande zone couverte par le titre.

Ces informations sont utilisées par le transporteur pour effectuer des études statistiques sur la fréquentation de ses lignes, en vue d'améliorer le service.
Par exemple, certaines lignes desservant majoritairement des zones d'activités ou des établissements d'enseignement connaissent une forte baisse de leur fréquentation en été, ce qui permet d'alléger le service ; pour choisir la période concernée, il faut connaître précisément l'évolution de la fréquentation au cours de l'été, ainsi que sa répartition au cours de la journée.

Q21. **Donner la requête SQL permettant de récupérer les dates et les heures de tous les passages ayant eu lieu sur la ligne numérotée 1 entre le 1er juillet et le 31 août 2014 (inclus).**

Un autre exemple de problématique est celui de l'efficacité des dézonages : à certaines périodes de l'année, les voyageurs sont autorisés à emprunter l'ensemble du réseau de transport quelles que soient les zones de validité de leur abonnement. Pour évaluer l'efficacité d'une telle mesure, il faut connaître le nombre de trajets en ayant bénéficié.

Q22. **Donner la requête SQL permettant de compter le nombre de passages dézonés, c'est-à-dire ayant eu lieu hors de l'intervalle de validité du titre utilisé, effectués le 31 décembre 2014.**

IN CHOISY – 15 1043 – D'après documents fournis