

12

**DEMANDE DE BREVET D'INVENTION**

**A1**

22 Date de dépôt : 04.06.02.

30 Priorité :

43 Date de mise à la disposition du public de la demande : 05.12.03 Bulletin 03/49.

56 Liste des documents cités dans le rapport de recherche préliminaire : *Se reporter à la fin du présent fascicule*

60 Références à d'autres documents nationaux apparentés :

71 Demandeur(s) : *STMICROELECTRONICS SA  
 Société anonyme* — FR.

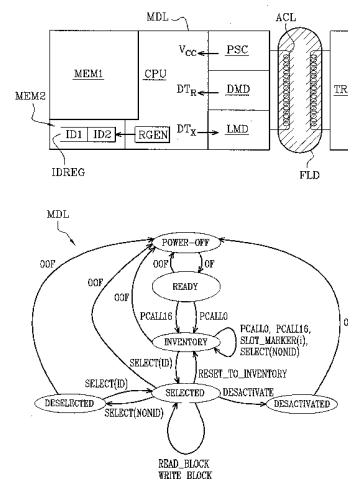
72 Inventeur(s) : MANI CHRISTOPHE.

73 Titulaire(s) :

74 Mandataire(s) : MARCHAND ANDRE.

54 **PROCEDE ANTICOLLISION POUR MODULE ELECTRONIQUE SANS CONTACT.**

57 La présente invention concerne un procédé anticollision pour l'identification et la sélection par un terminal de modules électroniques sans contact (MDL), dans lequel un module génère un numéro d'identification aléatoire (ID) avant une communication, et répond à une demande d'identification générale ou complémentaire (PCALL16) sur une position de réponse qui est fonction de son numéro d'identification (ID), tandis qu'un module non sélectionné génère un nouveau numéro d'identification aléatoire lorsqu'il reçoit une demande d'identification complémentaire (PCALL16). Ainsi, la position de réponse d'un module non sélectionné en réponse à une demande d'identification complémentaire n'est statistiquement pas la même que sa position de réponse en réponse à une précédente demande d'identification, tout en étant fonction de son numéro d'identification (ID). Application notamment aux étiquettes électroniques sans contact et aux cartes à puce sans contact.



FR 2 840 472 - A1



PROCEDE ANTICOLLISION POUR MODULE ELECTRONIQUE SANS  
CONTACT

La présente invention concerne les techniques de transmission de données sans contact, notamment par induction électromagnétique, appliquées aux étiquettes électroniques sans contact, aux cartes à puce sans contact... et autres objets électroniques portatifs sans contact de type équivalent.

La présente invention concerne plus particulièrement un procédé anticollision pour l'identification par un terminal de modules électroniques sans contact.

Dans les applications sans contact du type susmentionné, il est fréquent que plusieurs modules se trouvent au même instant dans le champ d'interrogation d'un terminal. Une telle situation est représentée en figure 1, où l'on voit un terminal TRM en présence de plusieurs modules M1, M2, M3... Mn agencés ici pour communiquer avec le terminal par couplage inductif. Le terminal TRM est équipé d'une interface de communication IT comportant une bobine d'antenne lui permettant d'émettre un champ magnétique FLD, d'envoyer des données aux modules, généralement par modulation de l'amplitude du champ magnétique, et de recevoir des données émises par les modules. Les modules M1 à Mn sont équipés d'interfaces de communication IM comportant chacune une bobine d'antenne, leur permettant d'envoyer des données au terminal, généralement par modulation de charge (affaiblissement du champ magnétique par court-circuit des bobines d'antenne) et de recevoir les données émises par le terminal.

Dans la situation illustrée en figure 1, le terminal doit établir une communication avec chacun des modules afin de réaliser une opération prévue par l'application, par exemple une transaction monétaire, une  
5 opération d'identification et d'authentification en vue de délivrer une autorisation de passage (contrôle d'accès), une opération de péage ou tout simplement une identification sans authentification des modules présents (inventaire). La communication avec un module ne peut  
10 qu'être individuelle et ne doit pas être perturbée par des messages émis par d'autres modules. Dans le cas contraire, il se produit une collision de messages qui se traduit au niveau du terminal par la réception de messages entachés d'erreur.

15 En règle générale, le terminal ne connaît pas l'identité des modules présents dans son champ d'interrogation. Cela signifie qu'il n'est pas envisageable que le terminal envoie à un module déterminé un message particulier qui permettrait de rendre actif ce  
20 module tout en maintenant les autres silencieux.

Ce problème est résolu grâce aux procédés anticollision, qui nécessitent tout d'abord que les modules possèdent chacun un numéro d'identification qui leur est propre afin de pouvoir être différenciés les uns  
25 des autres par le terminal.

Parmi les procédés anticollision connus, on distingue les procédés déterministes et les procédés probabilistes.

Selon les procédés déterministes, le terminal envoie  
30 des messages de marquage délimitant des positions de réponse ("time slots") sur une échelle temporelle de réponse comprenant  $2^N$  positions de réponse. Chaque module envoie un message d'identification comprenant son numéro d'identification, lorsqu'une position de réponse  
35 correspondant aux  $N$  premiers bits de son numéro

d'identification est atteinte. Lorsqu'un module est le seul à répondre sur une position de réponse, le terminal l'identifie et peut le sélectionner. Si deux modules répondent sur la même position de réponse, cela signifie  
5 que les N premiers bits de leur numéro d'identification sont identiques. Le terminal détecte la collision et envoie une demande d'identification complémentaire de type nominative, comprenant les N bits des numéros d'identification en collision. Cette demande  
10 d'identification complémentaire nominative ne concerne que les modules en collision sur la position de réponse définie par les N bits spécifiés. En réponse à la demande d'identification complémentaire nominative, les modules concernés élaborent une nouvelle position de réponse en  
15 utilisant cette fois les N bits suivants de leur numéro d'identification, et renvoient un nouveau message d'identification. Cette séquence peut se répéter plusieurs fois jusqu'à ce que tous les modules soient identifiés et/ou sélectionnés, et peut se répéter pour  
20 chaque position de réponse autant de fois que le nombre de sous-groupes de N bits restant à parcourir dans les numéros d'identification.

Le processus d'identification est arborescent car chaque étape d'identification complémentaire nominative  
25 ne concerne que les modules en collision sur une position de réponse déterminée. Ainsi, par exemple, si deux premiers modules présentent N premiers bits identiques et deux autres modules présentent N premiers bits identiques mais différents des N premiers bits des deux premiers  
30 modules, l'identification des quatre modules nécessitera au moins deux étapes d'identification complémentaires, l'une pour départager les deux premiers modules et l'autre pour départager les deux autres modules.

En pratique, l'arborescence des procédés  
35 déterministes peut devenir très complexe lorsque le

nombre de modules augmente et l'identification de chaque module nécessite un temps de traitement assez long.

Les procédés anticollision impliquent également que les modules exécutent des algorithmes relativement complexes leur permettant de savoir s'ils doivent ou non répondre à une demande d'identification complémentaire nominative.

D'autre part, lorsque le nombre de modules augmente, le risque augmente que deux modules aient le même numéro d'identification, ce qui correspond à une collision totale et à l'impossibilité de départager deux modules. On doit alors augmenter la longueur des numéros d'identification pour diminuer ce risque. Outre la prévision de numéros d'identification d'une grande longueur, une autre mesure permettant de diminuer le risque d'une collision totale est que chaque module génère un numéro d'identification aléatoire avant une communication. Le numéro d'identification est ensuite utilisé par le module pour toutes les interrogations successives jusqu'à ce que le module soit mis hors tension ou soit désactivé. Toutefois, la génération de numéros d'identification aléatoires de grande longueur nécessite la prévision de générateurs de nombres aléatoires puissants et performants, d'une complexité et d'un prix de revient élevé et d'un encombrement non négligeable en termes de surface de silicium.

Un procédé anticollision déterministe est par exemple préconisé par la norme ISO 14443-3 type B, qui prévoit que les modules ont des numéros d'identification de 32 bits pouvant être aléatoires.

En résumé, les procédés déterministes prévoient une identification de modules de type arborescent, qui est complexe à mettre en œuvre, peu rapide, et nécessite des générateurs de nombres aléatoires performants.

L'avantage de ces procédés est toutefois que, l'identification des modules étant de type arborescent, le choix d'un nombre important de positions de réponse n'est pas impératif. En choisissant par exemple un nombre  
5 N égal à 4 (soit quatre bits), on obtient 16 positions de réponse seulement.

Les procédés probabilistes se distinguent des procédés déterministes en ce que la position de réponse d'un module est indépendante de son numéro  
10 d'identification. Le terminal envoie tout d'abord une demande d'identification générale qui déclenche dans le module la génération d'un nombre aléatoire. Ce nombre aléatoire détermine la position de réponse du module pour l'envoi d'un message d'identification comprenant son  
15 numéro d'identification. L'efficacité des procédés probabilistes, lorsque le nombre de modules augmente, ne dépend pas de la longueur des numéros d'identification (bien que subsiste toujours la nécessité de ne pas avoir deux modules ayant le même numéro d'identification) mais  
20 de l'étalement de l'échelle de réponse. Ainsi, plus le nombre de positions de réponse est grand, plus le procédé probabiliste est efficace. En cas de collision, les modules non identifiés reçoivent une demande d'identification complémentaire non nominative et  
25 déterminent une nouvelle position de réponse aléatoire. L'étape d'identification complémentaire n'étant pas nominative, elle concerne des modules ayant été en collision sur des positions de réponse différentes, contrairement aux étapes d'identification nominatives des  
30 procédés déterministes.

En résumé, les procédés anticollision probabilistes sont plus simples à mettre en œuvre en ce qui concerne les algorithmes embarqués dans les modules, mais nécessitent une échelle de réponse étendue pour être  
35 applicables à un nombre important de modules. En

contrepartie, si l'échelle de réponse est étendue, le processus d'identification peut s'avérer inutilement long lorsqu'un nombre réduit de modules se trouve simultanément dans le champ d'interrogation du terminal.

5 Considérons à titre d'exemple que 255 positions de réponse soient prévues et que deux modules seulement se trouvent dans le champ d'interrogation du terminal. Il est possible qu'un module choisisse aléatoirement la position N°1 tandis que l'autre module choisit la  
10 position de réponse N°255. Dans ce cas, après avoir identifié le premier module, il faut parcourir toute l'échelle de réponse et atteindre la 255e position de réponse pour identifier le deuxième module.

Encore un autre inconvénient des procédés  
15 probabilistes classiques est qu'une collision sur les numéros d'identification peut intervenir entre deux modules n'ayant pas répondu sur la même position de réponse. Le terminal doit alors demander aux modules de générer de nouveaux numéros d'identification, et  
20 recommencer le processus d'identification.

Ainsi, la présente invention vise à prévoir un procédé anticollision qui ne présente pas les inconvénients des procédés connus.

A cet effet, l'idée générale de la présente  
25 invention est de prévoir une combinaison de moyens relevant à la fois des procédés déterministes et des procédés probabilistes. Selon l'invention, les demandes d'identification émises par le terminal ne sont pas nominatives, comme cela est le cas avec un procédé  
30 probabiliste, mais la position de réponse d'un module est fonction du numéro d'identification du module, comme dans un procédé déterministe. D'autre part, un module génère un nouveau numéro d'identification de façon aléatoire lorsqu'il doit répondre à une demande d'identification  
35 complémentaire, de sorte que sa position de réponse est

aléatoire et n'est pas constante, comme dans un procédé probabiliste. Le processus d'identification n'est donc pas de type arborescent et est ainsi plus simple à mettre en œuvre. De préférence, la position de réponse ne dépend  
5 que d'une partie du numéro d'identification, comme dans les procédés déterministes, de sorte que l'échelle de réponse peut comprendre un nombre réduit de positions de réponse si la partie du numéro d'identification servant à définir la position de réponse comprend un nombre de bits  
10 réduit.

Plus particulièrement, la présente invention prévoit un procédé anticollision pour l'identification et la sélection par un terminal de modules électroniques sans contact, dans lequel : avant une communication, un  
15 module génère un numéro d'identification aléatoire ; le terminal envoie une demande d'identification générale pour déclencher l'envoi de messages d'identification par les modules, et envoie une ou plusieurs demandes d'identification complémentaire non nominatives lorsque  
20 des messages d'identification se trouvent en collision ; un module est sélectionné lorsqu'il reçoit un message de sélection comprenant son numéro d'identification ; en réponse à une demande d'identification générale ou complémentaire, un module non sélectionné envoie un  
25 message d'identification comprenant son numéro d'identification, lorsqu'il atteint, sur une échelle de réponse, une position de réponse déterminée, procédé dans lequel : un module répond à une demande d'identification générale ou complémentaire sur une position de réponse  
30 qui est fonction de son numéro d'identification, et un module non sélectionné génère un nouveau numéro d'identification aléatoire lorsqu'il reçoit une demande d'identification complémentaire, de sorte que la position de réponse d'un module non sélectionné en réponse à une  
35 demande d'identification complémentaire n'est



statistiquement pas la même que sa position de réponse en réponse à une précédente demande d'identification, tout en étant fonction de son numéro d'identification.

Selon un mode de réalisation, une demande  
5 d'identification générale est identique en format et en code à une demande d'identification complémentaire.

Selon un mode de réalisation, un module détermine une position de réponse qui est égale à une partie déterminée de son numéro d'identification, la partie  
10 déterminée comprenant des bits de rang déterminé dont le rang ne varie pas lorsque le module répond à des demandes d'identification successives.

Selon un mode de réalisation, la partie déterminée du numéro d'identification d'un module comprend des bits  
15 de poids faible du numéro d'identification.

Selon un mode de réalisation, pour générer un nouveau numéro d'identification aléatoire, un module conserve une première partie de son numéro d'identification actuel, et génère de façon aléatoire une  
20 deuxième partie du numéro d'identification.

Selon un mode de réalisation, un module détermine une position de réponse qui est égale à la deuxième partie du numéro d'identification.

Selon un mode de réalisation, la première partie du  
25 numéro d'identification d'un module est générée de façon aléatoire par le module avant une communication.

Selon un mode de réalisation, la première partie du numéro d'identification d'un module est une donnée fixe enregistrée dans le module.

Selon un mode de réalisation, le terminal envoie  
30 également une demande d'identification complémentaire lorsqu'il reçoit, en réponse à une précédente demande d'identification complémentaire, un message d'identification d'un module qui n'est pas en collision  
35 avec un autre message d'identification mais qui comprend

un numéro d'identification déjà utilisé par un autre module.

Selon un mode de réalisation, un module est automatiquement activé lorsqu'il se trouve en présence  
5 d'un champ magnétique émis par le terminal, et génère un numéro d'identification aléatoire lorsqu'il est activé.

Selon un mode de réalisation, les positions de réponse sur l'échelle de réponse sont repérées par des messages de marquage envoyés par le terminal.

10 Selon un mode de réalisation, une demande d'identification générale forme un message de marquage de la première position de réponse sur l'échelle de réponse.

Selon un mode de réalisation, un message de marquage comprend un numéro de position de réponse, au  
15 moins pour des positions de réponse supérieures à une première position de réponse.

Selon un mode de réalisation, le terminal envoie des messages de marquage dans un ordre ne correspondant pas à un rang croissant des positions de réponse.

20 Selon un mode de réalisation, le terminal n'envoie une demande d'identification complémentaire qu'après que toutes les positions de réponse de l'échelle de réponse ont été parcourues.

Selon un mode de réalisation, pour initier une  
25 communication, le terminal envoie tout d'abord une demande d'identification immédiate auquel les modules répondent immédiatement, sans échelle de réponse.

Selon un mode de réalisation, un module comprend un état de sélection ainsi qu'un état de désélection ou  
30 désactivation dans lequel le terminal peut le faire basculer après l'avoir sélectionné, et dans lequel un module dans l'état de sélection ou dans l'état de désélection ou désactivation ne répond pas à une demande d'identification générale ou complémentaire.

La présente invention concerne également un module électronique sans contact, comprenant des moyens pour émettre et recevoir des messages sans contact, des moyens de décodage de messages reçus agencés pour décoder au moins une demande d'identification générale, une demande d'identification complémentaire non nominative et un message de sélection, des moyens pour basculer dans un état sélectionné sur réception d'un message de sélection comprenant un numéro d'identification du module, le module étant agencé pour : générer un numéro d'identification aléatoire avant une communication ; sur réception d'une demande d'identification générale ou complémentaire, déterminer, si le module n'est pas dans l'état sélectionné, une position de réponse sur une échelle de réponse puis envoyer, lorsque la position de réponse est atteinte, un message d'identification comprenant le numéro d'identification du module, et en réponse à une demande d'identification générale ou complémentaire, déterminer une position de réponse qui est fonction de son numéro d'identification, et générer un nouveau numéro d'identification aléatoire lorsqu'il reçoit une demande d'identification complémentaire, de sorte que la position de réponse du module en réponse à une demande d'identification complémentaire n'est statistiquement pas la même que sa position de réponse en réponse à une précédente demande d'identification, tout en étant fonction de son numéro d'identification.

Selon un mode de réalisation, le module est agencé pour décoder une demande d'identification complémentaire identique en format et en code à une demande d'identification générale.

Selon un mode de réalisation, le module est agencé pour déterminer une position de réponse qui est égale à une partie déterminée de son numéro d'identification, la partie déterminée comprenant des bits de rang déterminé

dont le rang ne varie pas lorsque le module répond à des demandes d'identification successives.

Selon un mode de réalisation, la partie déterminée comprend des bits de poids faible du numéro d'identification du module.

Selon un mode de réalisation, le module est agencé pour générer un nouveau numéro d'identification en conservant une première partie de son numéro d'identification actuel, et en générant de façon aléatoire une deuxième partie du numéro d'identification.

Selon un mode de réalisation, le module est agencé pour déterminer une position de réponse égale à la deuxième partie du numéro d'identification, en réponse à une demande d'identification générale ou complémentaire.

Selon un mode de réalisation, la première partie du numéro d'identification d'un module est une donnée fixe enregistrée dans le module.

Selon un mode de réalisation, la première partie du numéro d'identification d'un module est générée de façon aléatoire par le module avant une communication.

Selon un mode de réalisation, le module est agencé pour déterminer une position de réponse qui est égale à la deuxième partie du numéro d'identification.

Selon un mode de réalisation, le module est agencé pour basculer automatiquement dans un état activé lorsqu'il se trouve en présence d'un champ magnétique d'une fréquence déterminée, et générer un numéro d'identification lorsqu'il bascule dans l'état activé.

Selon un mode de réalisation, le module est agencé pour décoder des messages de marquage envoyés par le terminal, permettant au module de délimiter des positions de réponse sur l'échelle de réponse.

Selon un mode de réalisation, le module est agencé pour décoder des messages de marquage comprenant un numéro de position de réponse, au moins pour des

positions de réponse supérieures à une première position de réponse.

Selon un mode de réalisation, le module est agencé pour décoder une demande d'identification générale en tant que message de marquage d'une première position de réponse sur l'échelle de réponse.

Selon un mode de réalisation, le module est agencé pour décoder une demande d'identification immédiate et pour envoyer immédiatement, sans échelle de réponse, en réponse à une demande d'identification immédiate, un message d'identification.

Selon un mode de réalisation, le module comprend un état de désélection ou désactivation dans lequel il peut basculer après avoir été dans l'état de sélection, le module étant agencé pour ne pas répondre à une demande d'identification générale ou complémentaire lorsqu'il se trouve dans l'état de désélection ou de désactivation.

Ces objets, caractéristiques et avantages ainsi que d'autres de la présente invention seront exposés plus en détail dans la description suivante d'un exemple non limitatif de mise en œuvre du procédé anticollision selon l'invention, et d'un exemple non limitatif de réalisation d'un module agencé pour mettre en œuvre le procédé selon l'invention, faite à titre non limitatif en relation avec les figures jointes parmi lesquelles :

- la figure 1 précédemment décrite représente plusieurs modules sans contact présents simultanément dans le champ d'interrogation d'un terminal,
- la figure 2 est un schéma bloc d'un module agencé pour mettre en œuvre le procédé anticollision selon l'invention,
- la figure 3 est un diagramme d'états illustrant le fonctionnement du module de la figure 2,
- la figure 4 est un organigramme illustrant le fonctionnement du module de la figure 2 lorsque celui-ci

se trouve dans un état déterminé parmi les divers états représenté en figure 3,

- les figures 5A à 5D illustrent une séquence anticollision selon l'invention, comprenant plusieurs sous-séquences.

La figure 2 est le schéma bloc d'un module sans contact MDL prévu pour mettre en œuvre le procédé anticollision selon l'invention. Le module présente une architecture générale en soi classique et comprend ainsi une unité centrale CPU, une zone mémoire MEM1 de type non volatile (par exemple EPROM ou EEPROM), une mémoire MEM2 volatile (RAM) et une interface de communication équipée d'une bobine d'antenne ACL. Lorsque le module est plongé dans un champ magnétique alternatif FLD émis par un terminal TRM, une tension alternative Vi apparaît aux bornes de la bobine d'antenne, la tension Vi oscillant à la fréquence porteuse du champ magnétique.

La zone mémoire MEM1 comprend par exemple 128 blocs de 32 bits chacun. La zone mémoire MEM2 comprend divers registres classiques utilisés par le CPU, non représentés en figure 2 à l'exception d'un registre IDREG. Ce registre IDREG contient un numéro d'identification du module, ou numéro ID, comprenant ici deux parties ID1, ID2. La partie ID1 comprend par exemple les bits de poids fort du numéro ID et la partie ID2 les bits de poids faible. Dans ce qui suit, on considèrera à titre d'exemple non limitatif que les parties ID1, ID2 comprennent chacune quatre bits. Les exemples de numéros ID décrits dans ce qui suit seront exprimés en notation hexadécimale, les numéros ID pouvant aller de 00 (8 bits à 0) à FF (8 bits à 1, soit 256 en base 10).

Selon l'invention, le registre IDREG est relié à la sortie d'un générateur RGEN de nombre aléatoire, ici un générateur de 4 bits simple à implémenter et occupant une faible surface de silicium. Le générateur RGEN est

utilisé par l'unité centrale pour générer, avant une communication, les parties ID1, ID2 d'un numéro ID aléatoire. Le numéro ID est généré ici lorsque le module est activé par le champ magnétique FLD et bascule dans un état READY décrit plus loin. D'autre part l'unité CPU utilise le générateur RGEN pour générer une nouvelle partie ID2 quand elle doit répondre à une demande d'identification générale ou complémentaire correspondant à une commande PCALL16 décrite plus loin. Selon l'invention toujours, l'unité centrale, en réponse à une telle demande d'identification, choisit une position de réponse qui est égale à la partie ID2 du numéro ID.

L'interface de communication du module comprend un circuit d'alimentation PSC, un démodulateur DMD, et un modulateur LMD. Le circuit PSC délivre une tension Vcc d'alimentation du module, obtenue par redressement de la tension induite Vi. Le démodulateur DMD extrait de la tension induite Vi des données DTx envoyées par le terminal. Ces données sont généralement envoyées par modulation de l'amplitude du champ magnétique, et sont par exemple codées ASK (modulation par saut d'amplitude, ou "Amplitude Shift Keying Modulation"). Le modulateur LMD est prévu pour émettre des données DTx selon le principe de la modulation de charge de la bobine d'antenne. Les données DTx sont par exemple codées BPSK (Modulation par saut de phase, ou Phase Shift Keying Modulation) au moyen d'une sous-porteuse extraite de la porteuse du champ magnétique. La porteuse est par exemple de 13,56 MHz et la sous-porteuse de 847 KHz. Le débit de données obtenu est dans ce cas de l'ordre de 100 Kbits/s. L'unité CPU peut être du type à microprocesseur ou à logique câblée (séquenceur), et assure le décodage et le traitement des messages (commandes) envoyés par le terminal, et l'envoi de messages (réponses) au terminal,

ainsi que l'exécution d'étapes ou d'instructions spécifiques au procédé anticollision selon l'invention.

La figure 3 est un diagramme d'état illustrant le fonctionnement du module MDL. On distingue six états désignés respectivement POWER-OFF, READY, INVENTORY, 5 SELECTED, DESACTIVATED et DESELECTED, et des commandes envoyées par le terminal, PCALL0, PCALL16, SLOT-MARKER(i), SELECT(ID), SELECT(NONID), DESACTIVATE, READ-BLOCK, WRITE-BLOCK.

10 Ces commandes, ainsi que des réponses correspondantes R(ID) et R-READ-BLOCK émises par le module, sont décrites plus en détail dans les dernières pages de la présente description, auxquelles on pourra se reporter avant de lire ce qui suit.

15 Des transitions entre les divers états du module sont également déterminés par des événements désignés OF et OOF.

L'évènement OF ("on field") correspond au cas où le module entre dans le périmètre d'induction du champ 20 magnétique FLD. Le module passe alors de l'état POWER-OFF à l'état READY car le circuit PSC délivre la tension Vcc lorsque la tension alternative Vi apparaît aux bornes de sa bobine d'antenne.

L'évènement OOF ("out of field") correspond au cas 25 où le module sort du périmètre d'induction du champ magnétique émis par le terminal. Le module, s'il est actif et se trouve dans l'un des états READY, INVENTORY, SELECTED, DESELECTED ou DESACTIVATED, n'est plus alimenté électriquement par le circuit PSC en raison de la 30 disparition (ou de la valeur insuffisante) de la tension induite Vi. Le module retourne alors dans l'état POWER-OFF.

#### Etat READY

Lorsque le module bascule dans l'état READY, il 35 génère un numéro ID aléatoire et attend une commande



PCALL0 ou PCALL16 pour basculer dans l'état INVENTORY. Il sera noté ici que la commande PCALL0 est optionnelle dans le cadre de la présente invention car elle n'est pas une commande anticollision, tous les modules étant tenus de s'identifier simultanément. Cette commande est prévue pour éviter le déclenchement du processus anticollision quand un seul module est présent. Ainsi, dans une variante, le passage de l'état READY à l'état INVENTORY peut être lié à la réception de la commande PCALL16 uniquement. Inversement, il peut être prévu que le passage de l'état READY à l'état INVENTORY soit exclusivement lié à la réception de la commande PCALL0. Dans ce cas, en cas de collision, le processus anticollision n'est déclenché qu'à partir de l'état INVENTORY.

#### ETAT INVENTORY

Lorsque le module bascule dans l'état INVENTORY, il exécute un certain nombre d'actions représentées sur l'organigramme de la figure 4. Si le passage à l'état INVENTORY est déclenché par la réception d'une commande PCALL0, le module va directement à une étape S3 où il envoie un message d'identification prenant ici la forme d'une réponse R(ID) comprenant son numéro ID. Après l'étape S3, le module va à une étape S2. Si le passage à l'état INVENTORY est déclenché par la réception d'une commande PCALL16, le module va directement à une étape S1 où il génère une nouvelle partie ID2 de son numéro ID au moyen du générateur aléatoire RGEN. Le module va ensuite à l'étape S2.

A l'étape S2, le module attend une commande de marquage SLOT-MARKER(i) avec  $i=ID2$ , soit la commande SLOT-MARKER(ID2) qui marque la position de réponse correspondant à la partie ID2 du numéro ID du module. Lorsque la commande SLOT-MARKER(ID2) est reçue, le module

va à l'étape S3 où il envoie une réponse R(ID) et retourne ensuite à l'étape S2.

Dans le présent mode de réalisation où la commande PCALL16 forme un message de marquage de la première position de réponse, le module passe directement de l'étape S2 à l'étape S3 si la partie ID2 de son numéro ID est égale 0, car la partie ID2 correspond alors à la première position de réponse.

L'étape S2 est également une étape d'attente de sélection dans laquelle le module reste (ou revient) jusqu'à ce qu'une commande SELECT(ID) comprenant son numéro ID soit reçue. Lorsque cette commande est reçue, le module bascule dans l'état SELECTED.

Le module peut également recevoir une nouvelle commande PCALL16 lorsqu'il se trouve à l'étape S2. Dans ce cas le module retourne à l'étape S1 pour générer une nouvelle partie ID2 de son numéro ID puis revient à l'étape S2.

#### Etat SELECTED

Une fois dans l'état SELECTED, le module ne répond plus à des demandes d'identification immédiate PCALL0 ou à des demandes d'identification générale ou complémentaire PCALL16. La zone mémoire MEM1 du module peut être lue ou écrite par le terminal au moyen des commandes READ-BLOCK ou WRITE-BLOCK. En réponse à la commande READ-BLOCK, le terminal reçoit une réponse R-READ-BLOCK. Si le module reçoit une commande SELECT(NONID), c'est-à-dire une commande SELECT(ID) comprenant un numéro ID qui n'est pas le sien, le module bascule dans l'état DESELECTED. Si le module reçoit une commande DESACTIVATE, il bascule dans l'état DESACTIVATED.

#### Etat DESELECTED

Le module dans l'état DESELECTED ne répond à aucune commande hormis une commande SELECT(ID) comprenant son

numéro ID, qui le fait revenir à l'état SELECTED. La disparition du champ magnétique (événement OOF) le fait revenir à l'état POWER-OFF.

Etat DESACTIVATED

5 Le module dans l'état DESACTIVATED ne répond à aucune commande. Seule la disparition du champ magnétique (événement OOF) le fait revenir à l'état POWER-OFF.

Exemple de séquence anticollision

10 Un exemple de séquence anticollision selon l'invention est représenté sur les figures 5A à 5D. On suppose ici que huit modules selon l'invention, désignés respectivement TAG1 à TAG8, entrent simultanément dans le champ d'interrogation du terminal. La colonne C0 décrit les commandes envoyées par le terminal. Les colonnes C1 à  
15 C8 décrit les réponses envoyées par les modules, les numéros ID générés par les modules et les états dans lesquels se trouvent les modules. L'état POWER-OFF est désigné "P-O", l'état READY est désigné "RDY", l'état INVENTORY est désigné "INV", l'état SELECTED est désigné  
20 "SEL". L'état DESELECTED est désigné "DESEL".

Le début de la séquence est décrit par la figure 5A. Tous les modules sont dans l'état POWER-OFF et comprennent dans leurs registres IDREG respectifs des numéros d'identification non spécifiés ("xx"). Lorsque le  
25 champ magnétique apparaît (événement OF) les modules basculent dans l'état READY et génèrent chacun un numéro d'identification, respectivement 40, 13, 3F, 4A, 50, 48, 52, 7C dans cet exemple. Le terminal envoie une demande d'identification immédiate PCALL0 dans l'hypothèse où un  
30 seul module se trouverait dans son champs d'interrogation. Les modules basculent dans l'état INVENTORY et répondent simultanément par des messages R(ID) comprenant leurs numéros ID, soit R(40), R(13), R(3F), R(4A), R(50), R(48), R(52), R(7C) respectivement.  
35 Les réponses sont toutes en collision et ne peuvent être

lues par le terminal. Ce dernier envoie alors une demande d'identification générale PCALL16. Chaque module calcule une nouvelle partie ID2 de son numéro ID en conservant la partie ID1. Les numéros ID des modules deviennent  
5 respectivement, dans cet exemple : 45, 12, 30, 43, 55, 43, 53, 73.

La commande PCALL16 formant par convention un message de marquage de la première position de réponse, le module TAG3 répond à la commande PCALL16 en envoyant  
10 un message R(30) car la partie ID2 de son numéro d'identification est égale à 0 et correspond à la première position de réponse. La réponse du module n'étant pas en collision avec une autre réponse, le terminal peut la lire et renvoie une commande de  
15 sélection SELECT(30) à laquelle le module répond en envoyant à nouveau le message R(ID). Puis le module bascule dans l'état SELECTED. Le terminal envoie ensuite la commande SLOT-MARKER(1) à laquelle aucun module ne répond, puis la commande SLOT-MARKER(2) à laquelle le  
20 module TAG2 répond en s'identifiant puisque la partie ID2 de son numéro ID est égal à 2. La sélection du module TAG2 se fait selon la procédure déjà décrite, et a pour effet que le module précédemment sélectionné TAG3 bascule automatiquement dans l'état DESELECTED (Cf. figure 3, commande SELECT(NONID)). Le terminal continue ensuite d'envoyer des commandes SLOT-MARKER(i) pour i allant de 3 à F. Lorsque la commande SLOT-MARKER(3) est émise, les modules TAG4, TAG6, TAG7, TAG8 répondent simultanément et ne peuvent donc pas être sélectionnés. Les modules TAG1  
30 et TAG5 répondent simultanément à la commande SLOT-MARKER(4) et ne sont également pas sélectionnés. Aucun module ne répond sur les positions de réponse suivantes allant de 6 à F.

Ainsi, au terme de cette première sous-séquence  
35 anticollision, seuls les modules TAG2 et TAG3 ont été

sélectionnés, le module TAG3 étant dans l'état DESELCETD et le module TAG2 dans l'état SELECTED.

Une nouvelle sous-séquence anticollision, illustrée en figure 5B, est déclenchée par l'envoi de la commande  
5 PCALL16, qui constitue une demande d'identification complémentaire pour les modules encore en attente de sélection. Ceux-ci génèrent alors une nouvelle partie ID2 de leurs numéros ID qui deviennent respectivement, dans cet exemple : 40 (TAG1), 41 (TAG4), 53 (TAG5), 42 (TAG6),  
10 50 (TAG7) et 74 (TAG8). Les modules TAG1 et TAG7 répondent simultanément sur la première position de réponse et ne peuvent pas être sélectionnés. Le module TAG4 est sélectionné sur la deuxième position de réponse repérée par la commande SLOT-MARKER(1) et le module TAG6 est  
15 sélectionné sur la troisième position de réponse repérée par la commande SLOT-MARKER(2). Le module TAG5 est sélectionné sur la quatrième position de réponse repérée par la commande SLOT-MARKER(3). Le module TAG8 est sélectionné sur la cinquième position de réponse repérée  
20 par la commande SLOT-MARKER(4). Aucun module ne répond sur les positions de réponse suivantes, pour i allant de 5 à F.

Ainsi, à la fin de cette seconde sous-séquence anticollision, tous les modules sauf TAG1 et TAG7 ont été  
25 sélectionnés, et tous ont basculé dans l'état DESELECTED sauf le module TAG8 qui est dans l'état SELECTED.

Une troisième sous-séquence anticollision, illustrée en figure 5C, est déclenchée par l'envoi de la commande PCALL16. Le module TAG1 génère un numéro ID égal  
30 à 41 et le module TAG7 génère un numéro ID égal à 50, les parties ID1 étant conservées, comme précédemment. Le module TAG7 est sélectionné sur la première position de réponse marquée par la commande PCALL16. Le module TAG1 étant seul à n'être pas sélectionné, il est le seul à  
35 répondre sur la deuxième position de réponse, marquée par

la commande SLOT-MARKER(1), et envoie le message R(41). Toutefois le terminal ne le sélectionne pas car son numéro ID=41 a déjà été choisi par un autre module, à savoir le module TAG4 qui a été sélectionné au cours de  
5 la seconde sous-séquence anti-collision (fig. 5B). Ainsi, le terminal n'envoie pas la commande SELECT(41) et continue à marquer les positions de réponse suivantes jusqu'à ce que i soit égal à F, au cas où d'autres modules seraient entrés dans son champ d'interrogation.

10 Une quatrième sous-séquence anticollision, illustrée en figure 5D, est déclenchée par l'envoi de la commande PCALL16. Le module TAG1 choisi cette fois un numéro ID égal à 43 qui n'est utilisé par aucun autre module. Le module TAG1 est donc sélectionné sur la  
15 quatrième position de réponse, repérée par la commande SLOT-MARKER(3). Le terminal continue à marquer les positions de réponse suivantes, au cas où d'autres modules se trouveraient dans son champ d'interrogation.

La séquence anticollision qui vient d'être décrite  
20 est bien entendu susceptible de diverses variantes, selon l'application envisagée.

Ainsi, par exemple, lorsque le module TAG3 est sélectionné au cours de la première sous-séquence (fig. 5A), le terminal peut arrêter d'envoyer des commandes de  
25 marquage des positions de réponse suivantes, et établir une communication avec le module TAG3 pour lire ou écrire des données, puis faire basculer le module dans l'état DESACTIVATED.

Le terminal peut également, après avoir dialogué  
30 avec le module TAG3, le laisser dans l'état SELECTED et continuer la séquence anticollision. Dans ce cas, lorsque le terminal envoie la commande SELECT(12) pour sélectionner le module TAG2, le module TAG3 bascule automatiquement dans l'état DESELECTED car cette commande  
35 est pour lui une commande SELECT(NONID). La différence

avec l'état DESACTIVATED est que le terminal converse la possibilité de sélectionner le module TAG3 ultérieurement en lui envoyant la commande SELECT(30).

5 Selon encore une autre variante de réalisation, le terminal peut également envoyer les commandes de marquage dans le désordre ou dans un ordre qu'il détermine. Les commandes de marquage étant sous son contrôle, rien n'oblige le terminal à marquer les positions de réponse par ordre croissant.

10 Également, bien que la demande d'identification générale ou complémentaire PCALL16 ait été utilisée dans ce qui précède comme repère de la première position de réponse, une commande spécifique SLOT-MARKER(0) pourrait également être prévue.

15 Également, la première partie ID1 du numéro ID d'un module peut n'être pas aléatoire et être enregistrée dans la zone mémoire MEM1.

La partie ID1 peut être également d'une longueur supérieure à la partie ID2 qui sert à la détermination  
20 des positions de réponse.

Inversement, la détermination des positions de réponse peut être faite à partir du numéro ID complet. Dans ce cas, la génération d'un nouveau numéro ID en réponse à la commande PCALL16 peut porter sur tout le  
25 numéro ID, ou seulement une partie du numéro ID.

Bien que la présente invention ait été décrite dans ce qui précède dans le cadre de la réalisation de modules sans contact de type inductif, il va de soi que le procédé selon l'invention est applicable à divers autres  
30 types de modules sans contact, notamment les modules sans contact prévus pour dialoguer dans la bande UHF avec un terminal émettant un champ électrique au lieu d'un champ magnétique.

Principales commandes envoyées par le terminal et  
principales réponses renvoyées par les modules

5

Définitions, abréviations :

- commande : message émis par le terminal,
- réponse : message émis par un module.
- 10 - SOF : champ de début de trame ("start of frame"),
- EOF : champ de fin de trame ("end of frame"),
- CODE(COMMANDE) : code d'une commande ou d'une réponse,  
sur 4, 8 ou 16 bits, présent dans un champ de la commande  
ou de la réponse,
- 15 - CRCl et CRCh : bits de poids fort et de poids faible  
d'une signature de type CRC accompagnant les commandes ou  
les réponses, permettant au destinataire de détecter des  
erreurs de transmission, et permettant notamment au  
terminal de détecter des collisions entre réponses R(ID).

20

Commandes :

Commande PCALL0

- 25 Description : demande d'identification immédiate non  
nominative.

Modules exécutant la commande : modules dans l'état READY  
ou INVENTORY (soit modules non sélectionnés).

30

Réponse à la commande : les modules envoient chacun  
immédiatement un message d'identification R(ID).



Effet de la commande sur l'état d'un module : un module dans l'état READY bascule dans l'état INVENTORY ; un module dans l'état INVENTORY reste dans l'état INVENTORY.

5 Exemple de format :

SOF	CODE(PCALL0)		CRCl	CRCh	EOF
	8 bits	8 bits	8 bits	8 bits	

#### Commande PCALL16

10 Description : demande d'identification générale de type non nominative ou demande d'identification complémentaire de type non nominative, formant également, optionnellement, un message de marquage de la première position de réponse (i=0).

15 Modules exécutant la commande : modules dans l'état READY ou INVENTORY (soit modules non sélectionnés).

20 Réponse à la commande : si la commande PCAL16 est utilisée comme message de marquage de la première position de réponse, les modules ayant un numéro ID=0 renvoient un message d'identification R(ID).

25 Effet de la commande sur l'état d'un module : un module dans l'état READY bascule dans l'état INVENTORY ; un module dans l'état INVENTORY reste dans l'état INVENTORY.

Exemple de format :

SOF	CODE(PCALL16)		CRCl	CRCh	EOF
	8 bits	8 bits	8 bits	8 bits	

Commande SLOT-MARKER(i)

Description : commande de marquage d'une position de  
réponse de rang i avec "i" spécifié dans la commande, et  
5 i > 0 (pour i = 0, voir commande PCALL16).

Modules exécutant la commande : modules dans l'état  
INVENTORY.

10 Réponse à la commande : un module envoie un message  
d'identification R(ID) si le rang "i" est égal à la  
partie ID2 du numéro ID du module.

Effet de la commande sur l'état d'un module : aucun, un  
15 module dans l'état INVENTORY reste dans l'état INVENTORY  
tant qu'il ne reçoit pas de commande de sélection  
SELECT(ID) comprenant son numéro ID.

Exemple de format :

SOF	i	CODE(SLOT- MARKER)	CRC1	CRCh	EOF
	N bits	4 bits	8 bits	8 bits	

20

Commande SELECT(ID)

Description : commande de sélection du module dont le  
code ID est spécifié dans la commande (commande  
25 nominative).

Modules exécutant la commande : modules dans l'état  
INVENTORY ou DESELECTED.

Réponse à la commande : le module visé par le numéro ID envoie un message d'identification R(ID) à titre de confirmation.

- 5 Effet de la commande sur l'état d'un module : un module dans l'état INVENTORY bascule dans l'état SELECTED ; un module dans l'état DESELECTED retourne dans l'état SELECTED.

- 10 Exemple de format :

SOF	CODE (SELECT)	ID	CRCl	CRCh	EOF
	8 bits	8 bits	8 bits	8 bits	

Commande SELECT (NONID)

- 15 Description : commande de sélection reçue (commande SELECT(ID)) comprenant un code ID ne correspondant pas à celui du module.

Modules réagissant à la commande : modules dans l'état SELECTED.

20

Réponse à la commande : aucune.

Effet de la commande sur l'état d'un module : un module dans l'état SELECTED bascule dans l'état DESELECTED.

25

Commande DESACTIVATE

Description : commande de désactivation d'un module se trouvant dans l'état SELECTED (commande non nominative).

30

Modules exécutant la commande : modules dans l'état SELECTED.

Réponse à la commande : aucune.

Effet de la commande sur l'état d'un module : un module dans l'état SELECTED bascule dans l'état DESACTIVATED.

5

Exemple de format :

SOF	CODE (DEACTIVATE)	CRCl	CRCh	EOF
	8 bits	8 bits	8 bits	

Commande RESET-TO-INVENTORY

10 Description : commande de retour à l'état INVENTORY destinée à un module se trouvant dans l'état SELECTED (commande non nominative).

Modules exécutant la commande : modules dans l'état  
15 SELECTED.

Réponse à la commande : aucune.

Effet de la commande sur l'état d'un module : un module  
20 dans l'état SELECTED retourne dans l'état INVENTORY.

Exemple de format :

SOF	CODE (RESET-TO-INVENTORY)	CRCl	CRCh	EOF
	8 bits	8 bits	8 bits	

Commande READ-BLOCK

25

Description : commande de lecture d'un bloc de N bits dans la zone mémoire MEM1, à une adresse spécifiée dans la commande (champ ADRESSE).

Modules exécutant la commande : modules dans l'état  
SELECTED.

Réponse à la commande : réponse R-READ-BLOCK (voir plus  
5 loin).

Effet de la commande sur l'état d'un module : aucun.

Exemple de format :

SOF	CODE (READ- BLOCK)	ADRESSE	CRC1	CRCh	EOF
	8 bits	8 bits	8 bits	8 bits	

10

Commande WRITE-BLOCK

Description : commande d'écriture d'un bloc de N bits  
(champs DATA1 à DATA4) dans la zone mémoire MEM1, à  
15 l'adresse spécifiée dans la commande (champ ADRESSE).

Modules exécutant la commande : modules dans l'état  
SELECTED uniquement.

20 Réponse à la commande : aucune (une vérification  
d'écriture peut être faite par le terminal au moyen de la  
commande READ-BLOCK).

Effet de la commande sur l'état d'un module : aucun.

25

Exemple de format :

SOF	CODE (WRITE- BLOCK)	ADRESSE	DATA1	DATA2	DATA3	DATA4	CRC1	CRCh	EOF
	8 bits	8 bits	N/4 bits	N/4 bits	N/4 bits	N/4 bits	8 bits	8 bits	

**Réponses :**Réponse R(ID)

- 5 Description : réponse d'un module à une commande PCALL0, PCALL16, SLOT-MARKER, ou SELECT(ID). La réponse contient le numéro ID du module.

Exemple de format :

SOF	ID	CRC1	CRCh	EOF
	8 bits	8 bits	8 bits	

10

Réponse R-READ-BLOCK

- Description : réponse d'un module à une commande READ-BLOCK, contenant un bloc de N bits (champs DATA1 à DATA4)
- 15 lu dans la zone mémoire MEM1 à l'adresse spécifiée dans la commande.

Exemple de format :

SOF	DATA1	DATA2	DATA3	DATA4	CRC1	CRCh	EOF
	N/4 bits	N/4 bits	N/4 bits	N/4 bits	8 bits	8 bits	

REVENDEICATIONS

1. Procédé anticollision pour l'identification et la sélection par un terminal de modules électroniques sans contact (MDL, TAG1-TAG8), dans lequel :
- avant une communication, un module génère un numéro d'identification (ID) aléatoire,
  - le terminal envoie une demande d'identification générale (PCALL16) pour déclencher l'envoi de messages d'identification par les modules, et envoie une ou plusieurs demandes d'identification complémentaire non nominatives (PCALL16) lorsque des messages d'identification se trouvent en collision,
  - un module est sélectionné lorsqu'il reçoit un message de sélection (SELECT(ID)) comprenant son numéro d'identification,
  - en réponse à une demande d'identification générale ou complémentaire, un module non sélectionné envoie un message d'identification (R(ID)) comprenant son numéro d'identification, lorsqu'il atteint, sur une échelle de réponse, une position de réponse déterminée,
- caractérisé en ce que :
- un module répond à une demande d'identification générale ou complémentaire sur une position de réponse qui est fonction de son numéro d'identification (ID), et en ce que :
  - un module non sélectionné génère un nouveau numéro d'identification aléatoire lorsqu'il reçoit une demande d'identification complémentaire (PCALL16), de sorte que la position de réponse d'un module non sélectionné en réponse à une demande d'identification complémentaire n'est statistiquement pas la même que sa position de réponse en réponse à une précédente demande d'identification, tout en étant fonction de son numéro d'identification (ID).

2. Procédé selon la revendication 1, dans lequel une demande d'identification générale (PCALL16) est identique en format et en code à une demande d'identification complémentaire (PCALL16).

3. Procédé selon l'une des revendications 1 et 2, dans lequel un module détermine une position de réponse qui est égale à une partie déterminée (ID2) de son numéro d'identification, la partie déterminée (ID2) comprenant des bits de rang déterminé dont le rang ne varie pas lorsque le module répond à des demandes d'identification successives (PCALL16).

4. Procédé selon la revendication 3, dans lequel la partie déterminée (ID2) du numéro d'identification d'un module comprend des bits de poids faible du numéro d'identification (ID).

5. Procédé selon l'une des revendications 1 à 4, dans lequel, pour générer un nouveau numéro d'identification aléatoire, un module conserve une première partie (ID1) de son numéro d'identification actuel, et génère de façon aléatoire une deuxième partie (ID2) du numéro d'identification.

6. Procédé selon la revendication 5, dans lequel un module détermine une position de réponse qui est égale à la deuxième partie (ID2) du numéro d'identification.

7. Procédé selon l'une des revendications 5 et 6, dans lequel la première partie (ID1) du numéro d'identification (ID) d'un module est générée de façon aléatoire par le module avant une communication.



8. Procédé selon l'une des revendications 5 et 6, dans lequel la première partie (ID1) du numéro d'identification d'un module est une donnée fixe enregistrée dans le module.

5

9. Procédé selon l'une des revendications 1 à 8, dans lequel le terminal envoie également une demande d'identification complémentaire (PCALL16) lorsqu'il reçoit, en réponse à une précédente demande d'identification complémentaire, un message d'identification d'un module qui n'est pas en collision avec un autre message d'identification mais qui comprend un numéro d'identification (ID) déjà utilisé par un autre module.

10

10. Procédé selon l'une des revendications 1 à 9, dans lequel un module est automatiquement activé (READY) lorsqu'il se trouve en présence d'un champ magnétique émis par le terminal, et génère un numéro d'identification aléatoire lorsqu'il est activé.

15

11. Procédé selon l'une des revendications 1 à 10, dans lequel les positions de réponse sur l'échelle de réponse sont repérées par des messages de marquage (PCALL16, SLOT-MARKER) envoyés par le terminal.

20

12. Procédé selon l'une des revendications 1 à 11, dans lequel une demande d'identification générale (PCALL16) forme un message de marquage de la première position de réponse sur l'échelle de réponse.

25

13. Procédé selon l'une des revendications 11 et 12, dans lequel un message de marquage (SLOT-MARKER(i)) comprend un numéro (i) de position de réponse, au moins

pour des positions de réponse supérieures à une première position de réponse.

14. Procédé selon la revendication 13, dans lequel  
5 le terminal envoie des messages de marquage dans un ordre ne correspondant pas à un rang croissant des positions de réponse.

15 15. Procédé selon l'une des revendications 1 à 14,  
10 dans lequel le terminal n'envoie une demande d'identification complémentaire qu'après que toutes les positions de réponse de l'échelle de réponse ont été parcourues.

16. Procédé selon l'une des revendications 1 à 15,  
15 dans lequel, pour initier une communication, le terminal envoie tout d'abord une demande d'identification immédiate (PCALL0) auquel les modules répondent immédiatement, sans échelle de réponse.

20 17. Procédé selon l'une des revendications 1 à 16,  
dans lequel un module comprend un état de sélection (SELECTED) ainsi qu'un état de désélection ou désactivation (DESELECTED, DESACTVATED) dans lequel le  
25 terminal peut le faire basculer après l'avoir sélectionné, et dans lequel un module dans l'état de sélection ou dans l'état de désélection ou désactivation ne répond pas à une demande d'identification générale ou complémentaire (PCALL16).

30 18. Module électronique sans contact (MDL),  
comprenant des moyens (DMD, LMD) pour émettre et recevoir des messages sans contact, des moyens (CPU) de décodage de messages reçus agencés pour décoder au moins une  
35 demande d'identification générale (PCALL16), une demande

d'identification complémentaire non nominative (PCALL16) et un message de sélection (SELECTED(ID)), des moyens pour basculer dans un état sélectionné (SELECTED) sur réception d'un message de sélection (SELECT(ID))  
5 comprenant un numéro d'identification (ID) du module, le module étant agencé pour :

- générer un numéro d'identification aléatoire (ID) avant une communication,
- sur réception d'une demande d'identification  
10 générale ou complémentaire, déterminer, si le module n'est pas dans l'état sélectionné, une position de réponse sur une échelle de réponse puis envoyer, lorsque la position de réponse est atteinte, un message d'identification (R(ID)) comprenant le numéro  
15 d'identification (ID) du module,

caractérisé en ce qu'il est agencé pour :

- en réponse à une demande d'identification générale (PCALL16) ou complémentaire (PCALL16), déterminer une position de réponse qui est fonction de son numéro  
20 d'identification (ID), et
- générer un nouveau numéro d'identification aléatoire (ID) lorsqu'il reçoit une demande d'identification complémentaire (PCALL16), de sorte que la position de réponse du module en réponse à une demande  
25 d'identification complémentaire n'est statistiquement pas la même que sa position de réponse en réponse à une précédente demande d'identification, tout en étant fonction de son numéro d'identification.

30 19. Module selon la revendication 18, agencé pour décoder une demande d'identification complémentaire (PCALL16) identique en format et en code à une demande d'identification générale (PCALL16).

20. Module selon l'une des revendications 18 et 19, agencé pour déterminer une position de réponse qui est égale à une partie déterminée (ID2) de son numéro d'identification (ID), la partie déterminée (ID2)  
5 comprenant des bits de rang déterminé dont le rang ne varie pas lorsque le module répond à des demandes d'identification successives.

21. Module selon la revendication 20, dans lequel  
10 la partie déterminée (ID2) comprend des bits de poids faible du numéro d'identification du module.

22. Module selon l'une des revendications 18 à 21, agencé pour générer un nouveau numéro d'identification en  
15 conservant une première partie (ID1) de son numéro d'identification actuel, et en générant de façon aléatoire une deuxième partie (ID2) du numéro d'identification.

20 23. Module selon la revendication 22, agencé pour déterminer une position de réponse égale à la deuxième partie (ID2) du numéro d'identification (ID), en réponse à une demande d'identification générale ou complémentaire (PCALL16).

25

24. Module selon l'une des revendications 22 et 23, dans lequel la première partie (ID1) du numéro d'identification (ID) d'un module est une donnée fixe enregistrée dans le module.

30

25. Module selon l'une des revendications 22 et 23, dans lequel la première partie (ID1) du numéro d'identification d'un module est générée de façon aléatoire par le module avant une communication.

35

26. Module selon l'une des revendications 22 à 25, agencé pour déterminer une position de réponse qui est égale à la deuxième partie (ID2) du numéro d'identification (ID).

5

27. Module selon l'une des revendications 18 à 26, agencé pour basculer automatiquement dans un état activé (READY) lorsqu'il se trouve en présence d'un champ magnétique d'une fréquence déterminée, et générer un  
10 numéro d'identification (ID) lorsqu'il bascule dans l'état activé.

28. Module selon l'une des revendications 18 à 27, agencé pour décoder des messages de marquage (SLOT-  
15 MARKER) envoyés par le terminal, permettant au module de délimiter des positions de réponse sur l'échelle de réponse.

29. Module selon la revendication 28, agencé pour  
20 décoder des messages de marquage (SLOT-MARKER(i)) comprenant un numéro (i) de position de réponse, au moins pour des positions de réponse supérieures à une première position de réponse.

25 30. Module selon l'une des revendications 28 et 29, agencé pour décoder une demande d'identification générale (PCALL16) en tant que message de marquage d'une première position de réponse sur l'échelle de réponse.

30 31. Module selon l'une des revendications 18 à 30, agencé pour décoder une demande d'identification immédiate (PCALL0) et pour envoyer immédiatement, sans échelle de réponse, en réponse à une demande d'identification immédiate (PCALL0), un message  
35 d'identification (R(ID)).

32. Module selon l'une des revendications 18 à 31, comprenant un état de désélection ou désactivation (DESELECTED, DESACTIVATED) dans lequel il peut basculer  
5 après avoir été dans l'état de sélection (SELECTED), le module étant agencé pour ne pas répondre à une demande d'identification générale ou complémentaire lorsqu'il se trouve dans l'état de désélection ou de désactivation.

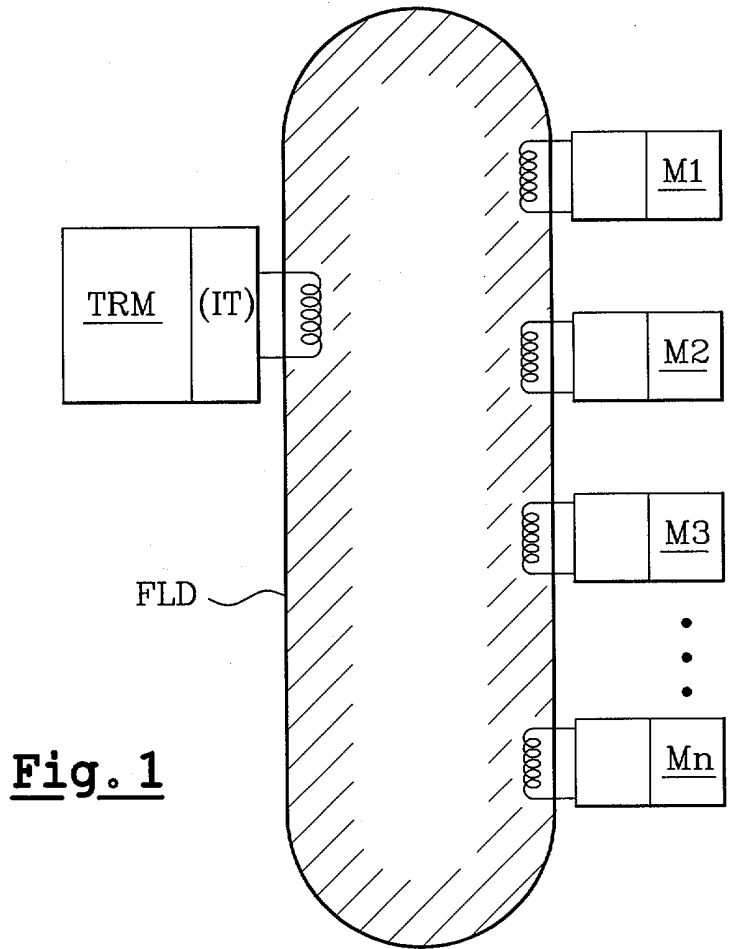
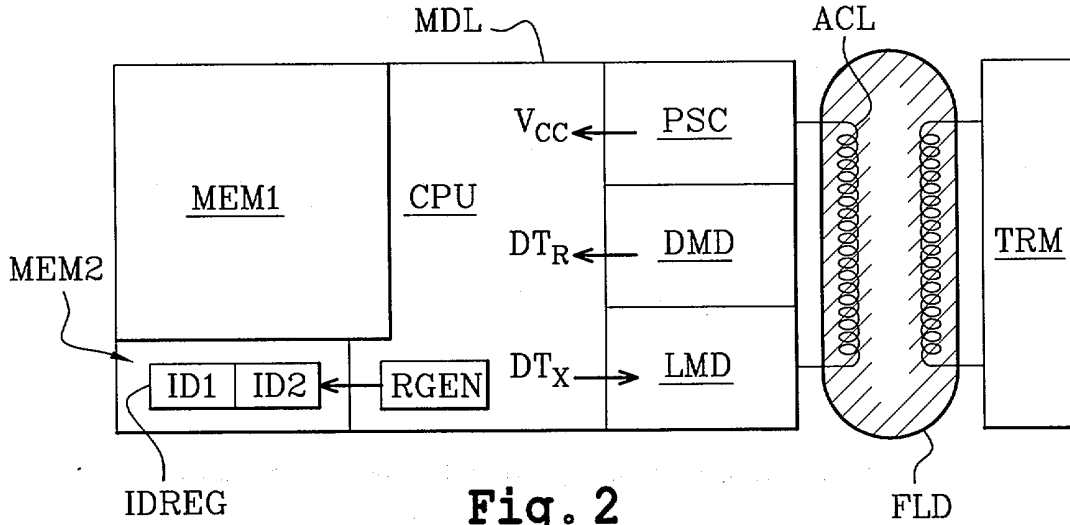
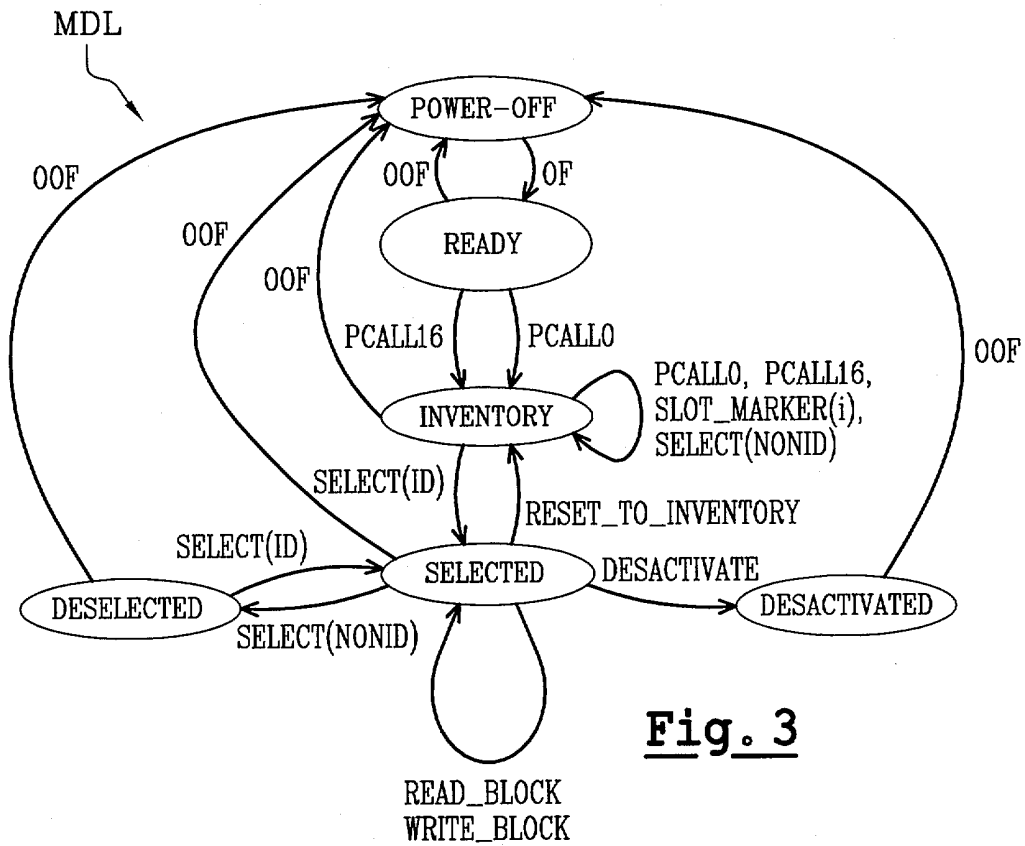


Fig. 1



**Fig. 2**



**Fig. 3**



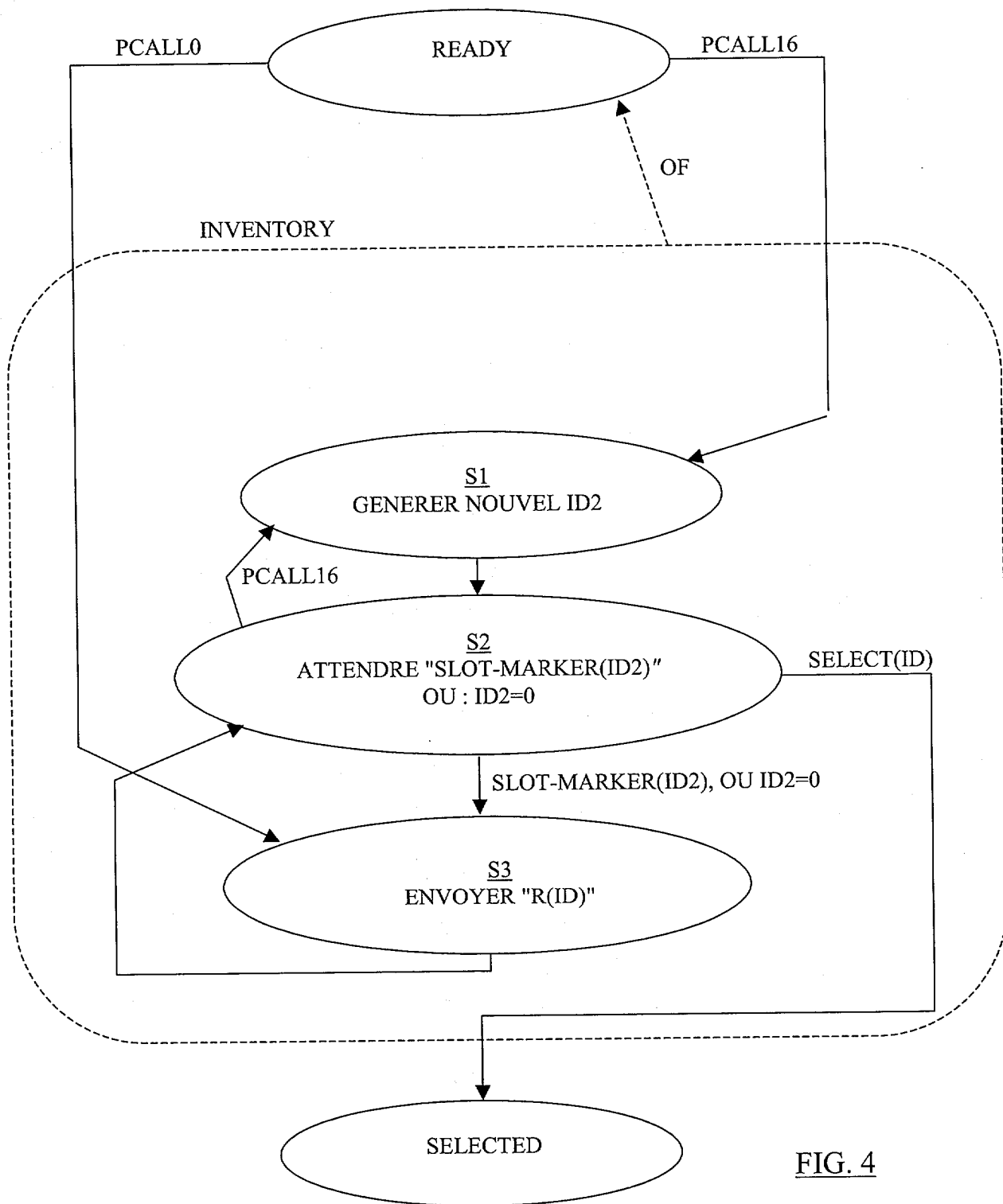


FIG. 4

FIG. 5A

C0	C1	C2	C3	C4	C5	C6	C7	C8
	<b>TAG1</b>	<b>TAG2</b>	<b>TAG3</b>	<b>TAG4</b>	<b>TAG5</b>	<b>TAG6</b>	<b>TAG7</b>	<b>TAG8</b>
	(P-O)	(P-O)	(P-O)	(P-O)	(P-O)	(P-O)	(P-O)	(P-O)
	<b>ID=xx</b>	<b>ID=xx</b>	<b>ID=xx</b>	<b>ID=xx</b>	<b>ID=xx</b>	<b>ID=xx</b>	<b>ID=xx</b>	<b>ID=xx</b>
	(RDY)	(RDY)	(RDY)	(RDY)	(RDY)	(RDY)	(RDY)	(RDY)
<b>OF</b>	<b>ID=40</b>	<b>ID=13</b>	<b>ID=3F</b>	<b>ID=4A</b>	<b>ID=50</b>	<b>ID=48</b>	<b>ID=52</b>	<b>ID=7C</b>
<b>PCALL0</b>	(INV)	(INV)	(INV)	(INV)	(INV)	(INV)	(INV)	(INV)
	<b>R(40)</b>	<b>R(13)</b>	<b>R(3F)</b>	<b>R(4A)</b>	<b>R(50)</b>	<b>R(48)</b>	<b>R(52)</b>	<b>R(7C)</b>
<b>PCALL16</b>	<b>ID=45</b>	<b>ID=12</b>	<b>ID=30</b>	<b>ID=43</b>	<b>ID=55</b>	<b>ID=43</b>	<b>ID=53</b>	<b>ID=73</b>
	(INV)	(INV)	<b>R(30)</b>	(INV)	(INV)	(INV)	(INV)	(INV)
SELECT(30)	(INV)	(INV)	(INV)	(INV)	(INV)	(INV)	(INV)	(INV)
	(INV)	(INV)	<b>R(30)</b>	(INV)	(INV)	(INV)	(INV)	(INV)
			<b>(SEL)</b>					
SLOT-MARKER(1)	(INV)	(INV)	-	(INV)	(INV)	(INV)	(INV)	(INV)
	(INV)	(INV)	-	(INV)	(INV)	(INV)	(INV)	(INV)
SLOT-MARKER(2)	(INV)	(INV)	-	(INV)	(INV)	(INV)	(INV)	(INV)
	(INV)	<b>R(12)</b>	-	(INV)	(INV)	(INV)	(INV)	(INV)
SELECT(12)	(INV)	(INV)	-	(INV)	(INV)	(INV)	(INV)	(INV)
	(INV)	<b>R(12)</b>	<b>(DESEL)</b>	(INV)	(INV)	(INV)	(INV)	(INV)
		<b>(SEL)</b>						
SLOT-MARKER(3)	(INV)	-	-	(INV)	(INV)	(INV)	(INV)	(INV)
	(INV)	-	-	<b>R(43)</b>	(INV)	<b>R(43)</b>	<b>R(53)</b>	<b>R(73)</b>
SLOT-MARKER(4)	(INV)	-	-	(INV)	(INV)	(INV)	(INV)	(INV)
	(INV)	-	-	(INV)	(INV)	(INV)	(INV)	(INV)
SLOT-MARKER(5)	(INV)	-	-	(INV)	(INV)	(INV)	(INV)	(INV)
	<b>R(45)</b>	-	-	(INV)	<b>R(55)</b>	(INV)	(INV)	(INV)
SLOT-MARKER(i = 6 ⇒ F)	(INV)	-	-	(INV)	(INV)	(INV)	(INV)	(INV)
	(INV)	-	-	(INV)	(INV)	(INV)	(INV)	(INV)

FIG. 5B

C0	C1	C2	C3	C4	C5	C6	C7	C8
	TAG1	TAG2	TAG3	TAG4	TAG5	TAG6	TAG7	TAG8
PCALL16	ID=40	-	-	ID=41	ID=53	ID=42	ID=50	ID=74
	R(40)	-	-	(INV)	(INV)	(INV)	R(50)	(INV)
SLOT-MARKER(1)	(INV)	-	-	(INV)	(INV)	(INV)	(INV)	(INV)
	(INV)	-	-	R(41)	(INV)	(INV)	(INV)	(INV)
SELECT(41)	(INV)	-	-	(INV)	(INV)	(INV)	(INV)	(INV)
	(INV)	(DESEL)	-	R(41) (SEL)	(INV)	(INV)	(INV)	(INV)
SLOT-MARKER(2)	(INV)	-	-	-	(INV)	(INV)	(INV)	(INV)
	(INV)	-	-	-	(INV)	R(42)	(INV)	(INV)
SELECT(42)	(INV)	-	-	-	(INV)	-	(INV)	(INV)
	(INV)	-	-	(DESEL)	(INV)	R(42) (SEL)	(INV)	(INV)
SLOT-MARKER(3)	(INV)	-	-	-	(INV)	-	(INV)	(INV)
	(INV)	-	-	-	R(53)	-	(INV)	(INV)
SELECT(53)	(INV)	-	-	-	(INV)	-	(INV)	(INV)
	(INV)	-	-	-	R(53) (SEL)	(DESEL)	(INV)	(INV)
SLOT-MARKER(4)	(INV)	-	-	-	-	-	(INV)	(INV)
	(INV)	-	-	-	-	-	(INV)	R(74)
SELECT(74)	(INV)	-	-	-	-	-	(INV)	-
	(INV)	-	-	-	(DESEL)	-	(INV)	R(74) (SEL)
SLOT-MARKER(i = 5 ⇒ F)	(INV)	-	-	-	-	-	(INV)	-
	(INV)	-	-	-	-	-	(INV)	-



**RAPPORT DE RECHERCHE**  
**PRÉLIMINAIRE**

établi sur la base des dernières revendications  
déposées avant le commencement de la recherche

FA 622959  
FR 0206831

DOCUMENTS CONSIDÉRÉS COMME PERTINENTS		Revendication(s) concernée(s)	Classement attribué à l'invention par l'INPI
Catégorie	Citation du document avec indication, en cas de besoin, des parties pertinentes		
X	EP 1 197 909 A (MATSUSHITA ELECTRIC IND CO LTD) 17 avril 2002 (2002-04-17)	1-4, 10-12, 15, 17-21, 27,28,32	H04B5/02 G06K19/07
A	* alinéa '0012! - alinéa '0027! *	5-9	
A	WO 99 67737 A (PINPOINT CORP) 29 décembre 1999 (1999-12-29) * page 15, ligne 19 - page 16, ligne 32 *	1,18	
A	EP 1 017 005 A (INTEGRATED SENSOR SOLUTIONS) 5 juillet 2000 (2000-07-05) * alinéa '0011! - alinéa '0019! *	1,18	
			<b>DOMAINES TECHNIQUES RECHERCHÉS (Int.CL.7)</b>
			G06K
		Date d'achèvement de la recherche	Examineur
		28 janvier 2003	Goossens, A
<b>CATÉGORIE DES DOCUMENTS CITÉS</b> X : particulièrement pertinent à lui seul Y : particulièrement pertinent en combinaison avec un autre document de la même catégorie A : arrière-plan technologique O : divulgation non-écrite P : document intercalaire		T : théorie ou principe à la base de l'invention E : document de brevet bénéficiant d'une date antérieure à la date de dépôt et qui n'a été publié qu'à cette date de dépôt ou qu'à une date postérieure. D : cité dans la demande L : cité pour d'autres raisons ..... & : membre de la même famille, document correspondant	

2

**ANNEXE AU RAPPORT DE RECHERCHE PRÉLIMINAIRE**  
**RELATIF A LA DEMANDE DE BREVET FRANÇAIS NO. FR 0206831 FA 622959**

La présente annexe indique les membres de la famille de brevets relatifs aux documents brevets cités dans le rapport de recherche préliminaire visé ci-dessus.  
 Les dits membres sont contenus au fichier informatique de l'Office européen des brevets à la date du 28-01-2003  
 Les renseignements fournis sont donnés à titre indicatif et n'engagent pas la responsabilité de l'Office européen des brevets, ni de l'Administration française

Document brevet cité au rapport de recherche		Date de publication	Membre(s) de la famille de brevet(s)	Date de publication
EP 1197909	A	17-04-2002	EP 1197909 A1	17-04-2002
			JP 2002203219 A	19-07-2002
			US 2002046226 A1	18-04-2002
WO 9967737	A	29-12-1999	AU 4715999 A	10-01-2000
			EP 1090371 A1	11-04-2001
			WO 9967737 A1	29-12-1999
			US 6353406 B1	05-03-2002
			AU 3586700 A	25-08-2000
			WO 0046771 A1	10-08-2000
EP 1017005	A	05-07-2000	EP 1017005 A2	05-07-2000
			JP 2000174658 A	23-06-2000