

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE
LA RECHERCHE SCIENTIFIQUE
UNIVERSITE MOULOU D MAMMERI DE TIZI-OUZOU
FACULTE DE GENIE ELECTRIQUE ET INFORMATIQUE
DEPARTEMENT D'INFORMATIQUE

MÉMOIRE

De fin d'études.

En vue de l'obtention du diplôme de Master en informatique.

Spécialité : Réseaux, Mobilité et Systèmes embarqués.

Thème :

Systeme antivol par géolocalisation GPS/GSM.

Proposé et dirigé par :

M^r DAOUI Mehammed.

Réalisé par :

- M^{lle} BOUKELA Lynda.
- M^r AIT TAYEB Mouhand Oussedik.

Membres du jury:

Président :

Examineurs :

Promotion: 2014/2015.

Remerciements

A Monsieur DAOUI, notre encadreur,

Nous vous sommes extrêmement reconnaissants d'avoir accepté de nous encadrer. Nous avons eu la chance de bénéficier de vos conseils éclairés pour la réalisation de notre travail. Nous vous adressons ici nos sincères remerciements et nous vous prions de croire en notre plus haute considération.

Aux membres du jury,

Nous tenons à vous remercier d'avoir accepté d'être membre de notre jury. Nous sommes sensibles à l'attention que vous avez bien voulu porter à ce travail. Nous vous exprimons ici l'assurance de notre profond respect.

A toute personne ayant contribué au bon déroulement de notre travail,

Nous tenons à vous remercier d'avoir été à nos côtés. Nous sommes reconnaissants pour le temps que vous nous avez consacré. Nous vous présentons ici nos sincères remerciements.

Dédicaces

A mes parents,

Que dieu vous donne santé et longue vie, je vous remercie pour votre continuel soutien et vos sacrifices.

A mes frères,

Que dieu soit à vos côtés pour illuminer vos chemins, je vous remercie pour vos conseils et encouragements.

A tous mes amis(es),

En particulier **SOFIA, BAYA, FETTA, MOHAMED, SEDIK, THIZIRI...**

A toute ma famille.

Lynda

Dédicaces

A ma mère,

Je dédie ce modeste travail à celle qui m'a donné la vie. Le symbole de tendresse qui s'est sacrifiée pour mon bonheur et ma réussite.

A mon père,

Ecole de mon enfance, qui a été mon ombre durant toutes les années d'études, et qui a veillé tout au long de ma vie à m'encourager, à me donner l'aide et à me protéger.

Que dieu les garde et les protège.

A mes frères, à mes amis(es),

A tous ceux qui me sont chères. A tous ceux qui m'aiment. A tous ceux que j'aime.

M^d Oussedik

Sommaire

INTRODUCTION GENERALE	2
CHAPITRE I	4
Introduction	5
I. Définition	5
II. Historique des systèmes embarqués	6
III. Caractéristiques d'un système embarqué	6
IV. Les types de systèmes embarqués	9
V. Architecture matérielle d'un système embarqué	9
V.1. Les mémoires	11
V.2. Les périphériques d'entrée/sortie	12
V.3. Les processeurs	13
V.4. Les Systèmes On Chip	14
V.5. Les Systèmes On Module	15
VI. Langages de développement utilisés	15
VII. Systèmes d'exploitation pour systèmes embarqués	16
VIII. Exemples d'application	18
Conclusion	19
CHAPITRE II	20
Introduction	21
I. La géolocalisation	22
II. Le système de coordonnées géographiques	22
III. Le GPS	23
III.1. Définition	23
III.2. Historique	24

III.3. La structure du système	24
III.4. Principe de positionnement par GPS	26
III.5. Les systèmes d'amélioration de la précision.....	28
IV. Les moyens de communication des coordonnées géographiques	29
V. Le GSM	31
V.1. Le concept cellulaire du GSM.....	31
V.2. Architecture d'un réseau GSM	31
V.3. L'interface SMS du GSM	32
VI. La géolocalisation GPS/GSM	32
VI.1. Géolocalisation GPS/GSM pour les systèmes antivol de véhicules	33
VI.2. Outils de géolocalisation GPS/GSM.....	33
Conclusion.....	34
CHAPITRE III	35
Introduction	36
I. Méthodologie de conception	36
II. Analyse.....	37
II.1. Idée générale sur notre travail.....	37
II.2. Spécification des exigences fonctionnelles.....	39
II.3. Spécification des besoins matériels	40
III. Conception	41
III.1. Conception matérielle.....	41
III.1.1. Description des composants matériels utilisés	42
III.1.1.1. Description de la carte Atmel AT91SAM7S256	42
III.1.1.2. Description du module SIM908.....	45

III.1.2. L'UART.....	47
III.2. Conception logicielle.....	49
III.2.1. Comportement du système logiciel	51
III.2.1.1. Le cas d'utilisation « Démarrage du système ».....	51
III.2.1.2. Le cas d'utilisation « Demande de la position du véhicule ».....	52
III.2.1.3. Le cas d'utilisation «Demande de modification du mot de passe»	54
III.2.2. Structure du système logiciel.....	55
III.2.3. Algorithmes de quelques procédures	56
Conclusion.....	63
CHPITRE IV	64
Introduction	65
I. Outils de développement utilisés.....	65
I.1. Outils logiciels	65
I.1.1. Keil - μ vision	65
I.1.2. Putty	66
I.2. Langage de programmation (Assembleur ARM).....	67
I.3. Outils matériels	68
II. Les tests	69
II.1. Les tests unitaires.....	69
II.1.1. Test de l'AT91SAM7S256.....	69
II.1.2. Test du module SIM908	70
II.2. Test de notre système.....	73
II.2.1. Demande de la position du véhicule	74

II.2.2. Demande de modification du mot de passe	75
Conclusion.....	75
CONCLUSION GENERALE	76
ANNEXE	78
A- Les protocoles SMS-SUBMIT et SMS-DELIVER du réseau GSM.....	79
I. SMS-SUBMIT	79
II. SMS-DELIVER	81
B- Le standard NMEA	82
I. Format de la trame NMEA 0183.....	82
II. Exemple de trame (GLL)	83
III. La trame manipulée par notre système.....	84
C- La technologie Machine To Machine (M2M).....	84
I. Définition	84
II. Architecture	85
III. Opérations principales.....	85
D- Les commandes « AT ».....	86
I. Syntaxe d'une commande AT.....	87
II. La norme GSM07.07	88
III. La norme GSM07.05	89

Liste des figures

Fig.I.1. Vue simplifiée de l'architecture d'un système embarqué.

Fig.I.2. Architecture typique d'un système embarqué.

Fig.II.1. Les coordonnées géographiques.

Fig.II.2. Structure du système GPS.

Fig.II.3. Satellite GPS.

Fig.II.4. Les stations de surveillance au sol du système GPS

Fig.II.5. Principe de positionnement par GPS.

Fig.III.1. Schéma de la méthodologie de conception.

Fig.III.2. Fonctionnement du système de géolocalisation.

Fig.III.3. Diagramme des cas d'utilisation.

Fig.III.4. Vue simplifiée de l'architecture matérielle du système.

Fig.III.5. Composants matériels utilisés.

Fig.III.6. La carte AT91SAM7S256.

Fig.III.7. Le module SIM908.

Fig.III.8. Schéma de connexion entre deux équipements à l'aide d'une liaison UART.

Fig.III.9. Schéma synoptique du système.

Fig.III.10. Schéma de fonctionnement global du système.

Fig.III.11. Diagramme de séquence pour le cas « démarrage du système »

Fig.III.12. Diagramme de séquence pour le cas « Demande de la position du véhicule».

Fig.III.13. Diagramme de séquence pour le cas « Demande de modification du mot de passe ».

Fig.III.14. Diagramme de classes.

Fig.IV.1. La plateforme Keil- μ vision.

Fig.IV.2. La fenêtre d'établissement d'une connexion série du logiciel Putty.

Fig.IV.3. Test de l'AT91SAM7S256.

Fig.IV.4. Résultats du test de l'AT91SAM7S256.

Fig.IV.5. Test du module SIM908.

Fig.IV.6. Résultats de la configuration initiale du module SIM908.

Fig.IV.7. Résultats de la demande de la trame via le module SIM908.

Fig.IV.8. Envoi d'un SMS via le module SIM908.

Fig.IV.9. Résultat de l'envoi d'un SMS via le module SIM908.

Fig.IV.10. Branchement du système.

Fig.IV.11. Envoi du mot de passe par SMS et réception de la position.

Fig.IV.12. Visualisation de la position du véhicule.

Fig.IV.13. Envoi de la commande de modification du mot de passe et réception de la confirmation.

Introduction
générale

Ces 25 dernières années, et sous l'effet d'entraînement des grandes industries (aviation, spatial, défense, énergie ...), les progrès de l'électronique, de la mécanique, de l'automatique, de l'informatique et des communications ont abouti à une indéniable maîtrise technique et technologique des systèmes embarqués. Aussi, la miniaturisation, la puissance de calcul, la taille des mémoires, l'autonomie énergétique, les progrès en fiabilité et la baisse des coûts ont permis – et permettent encore aujourd'hui – de concevoir et proposer de plus en plus de systèmes, produits et services dans le domaine de l'embarquée.

Initialement, les systèmes embarqués ont été utilisés pour des applications limitées à l'aérospatiale et à l'aviation. Désormais ils se sont étendus à des domaines grand public : transport, santé, télécommunications, sécurité...etc. Ils sont tellement omniprésents qu'ils consomment à eux seuls 95% à 98% des puces électroniques fabriquées dans le monde [SE5] et leur marché mondial a été évalué en 2013, à 112.5 milliards de dollars par le cabinet BBC Research.

L'automobile est l'un des plus importants domaines d'application des systèmes embarqués et actuellement, plus de 20% de la valeur d'une voiture est composée de systèmes et électroniques embarqués [SE7]. L'outil indispensable dans tout véhicule est l'outil de géolocalisation. Le plus connu est communément appelé «GPS», celui-ci est doté d'un récepteur qui permet au conducteur de se localiser et l'aide à s'orienter. En associant un moyen de communication à distance au dispositif de géolocalisation, on peut assurer la sécurisation du véhicule. En effet, ces dispositifs vont permettre au propriétaire de connaître l'emplacement de son véhicule, en cas de vol. Il peut aussi voir les déplacements de son véhicule en temps réel ainsi que sa vitesse.

Durant notre travail nous allons concevoir et développer un localisateur de véhicule. C'est un système utilisant un récepteur GPS pour la localisation et un module GSM pour la transmission de la position à l'aide des SMS. Ce localisateur va permettre à son utilisateur, muni d'un smartphone, de repérer son véhicule volé et voir sa position sur une carte.

Pour bien présenter notre travail, nous avons structuré notre mémoire comme suit :

- Le chapitre I portera sur des généralités sur les systèmes embarqués.
- Le chapitre II expliquera le principe de la géolocalisation GPS/GSM.
- Le chapitre III présentera les phases d'analyse et de conception de notre système, que ça soit la partie logicielle ou la partie matérielle.

- Le chapitre IV permettra de présenter les outils de développement utilisés et les résultats des tests du système.
- Nous terminons par une conclusion et des perspectives.
- Nous avons ajouté à la fin du mémoire une annexe présentant les détails des protocoles utilisés par le service SMS du GSM, les trames NMEA, l'architecture M2M et les commandes AT.

CHAPITRE 1

***Généralités sur les systèmes
embarqués.***

Introduction :

L'avènement des microprocesseurs en 1970 a orienté le développement informatique vers un nouvel horizon. Ce dispositif simple a permis, pour la première fois, la construction de systèmes relativement complexes. Initialement, les systèmes embarqués étaient des systèmes matériels et logiciels intégrés dans des navettes spatiales, des avions ou des trains. Par extension, les appareils portables ont été à leurs tours désignés comme des systèmes embarqués. Actuellement, les systèmes embarqués sont présents dans des applications de plus en plus nombreuses, par exemple les cartes à puce, les systèmes mobiles communicants (tels les téléphones mobiles, les capteurs dans les réseaux ad hoc), l'automobile, la santé et l'électronique grand public. Ils représentent un secteur important pour l'informatique, et chaque jour, notre vie en devient de plus en plus dépendante.

L'objet de ce chapitre est de présenter les généralités sur les systèmes embarqués. D'abord, nous définissons ces systèmes, nous énumérons ensuite leurs caractéristiques, mais aussi leur architecture, les langages de développement utilisés...etc.

I. Définition :

En anglais « Embedded system », ce terme désigne tout système électronique et informatique autonome, utilisant généralement un microprocesseur et possédant des ressources d'ordre spatial et énergétique limitées [SE6].

Un système embarqué désigne aussi bien le matériel que le logiciel utilisé. Souvent invisible à l'utilisateur, il est intégré dans un équipement dédié à la résolution d'un problème ou d'une tâche spécifique [SE2]. Les systèmes embarqués fonctionnent généralement en temps réel : les opérations de calcul sont alors faites en réponse à un événement extérieur (interruption matérielle). Dans ce cas, ils doivent utiliser un système d'exploitation temps-réel (RTOS¹). Les systèmes embarqués couvrent aussi bien les commandes de navigation et les commandes de trafic aérien qu'un simple agenda électronique de poche [SE1].

¹ RTOS : Real Time Operating System.

II. Historique des Systèmes Embarqués : [SE3]

L'un des premiers systèmes embarqués reconnu est le « Apollo Guidance Computer », développé en 1961 au MIT² Instrumentation Laboratory. Composé d'environ un millier de circuits intégrés identiques (portes NAND).

Un autre premier système embarqué est le « Autonetics D-17 guidance computer » qui servait de système de contrôle aux missiles nucléaires américains LGM-30 Minuteman, développé à partir de 1962, il était basé sur des transistors et contenait un disque dur comme mémoire principale.

En 1971, Intel produit le premier microprocesseur au monde. Cette puce, le 4004 (4 bits, composé de 2300 transistors), a été conçue suite à une demande de la compagnie japonaise Busicom. Premier circuit générique, personnalisable par logiciel. Le microprocesseur a été un succès immédiat, et son utilisation a augmenté régulièrement au cours de la prochaine décennie.

Une multitude de systèmes embarqués a vu le jour depuis, nous citerons quelques-uns ci-après:

- 1972 : lancement de l'Intel 8008, premier microprocesseur 8 bits (48 instructions, 800kHz).
- 1974 : lancement du 8080, premier microprocesseur largement diffusé. 8 bits, (64KB d'espace adressable, 2MHz - 3MHz).
- 1978 : création du Z80, processeur 8 bits.
- 1979 : création du MC68000, processeur 16/32 bits.

III. Caractéristiques d'un système embarqué : [SE4]

Les points suivants permettent de caractériser un logiciel embarqué :

- **Ciblé** : Le domaine d'action d'un système embarqué est limité aux fonctions pour lesquelles il a été créé.
- **Autonome** : Une fois enfouis dans l'application, les systèmes embarqués ne sont (le plus souvent) plus accessibles.

² MIT: Massachusetts Institute of Technology.

- ***Maintenable dans le temps*** : Dans la majorité des domaines industriels, et en dehors du logiciel classique, la durée de vie des produits est longue de par des obligations légales ou du moins commerciales qui obligent l'industriel à maintenir le produit pendant une dizaine d'années, cas de l'industrie automobile. Dans le cas d'industries plus sensibles comme l'aéronautique, le militaire ou le spatial, cette durée peut être doublée. Il est donc indispensable que le logiciel embarqué soit maintenable durant toute la durée de vie du produit en cas de découverte de problème de fonctionnement majeur.
- ***Sécurisé et fiable*** : Le logiciel nécessite une grande fiabilité car il est destiné à un fonctionnement complètement autonome et/ou critique.
- ***Mobilité*** : Souvent, les systèmes embarqués sont intégrés dans les systèmes qui ne sont pas liés à des lieux géographiquement statiques. Des petits lecteurs MP3 aux vitres électriques des automobiles à l'unité de contrôle d'un moteur d'un navire porte-conteneurs. Le système peut être appelé à prendre en compte des exigences implicites telle que le support de coups et de chocs pendant le fonctionnement.
- ***Temps réel*** : Certains systèmes doivent avoir un fonctionnement en temps réel, ils doivent être réactif, c.à.d. exécuter des opérations de calcul en réponse à un événement extérieur (interruption matérielle), ils sont utilisés dans des domaines où la validité d'un résultat (et sa pertinence) dépend du moment où il est délivré car le fait de rater une échéance va causer une erreur de fonctionnement.
- ***Coût du produit*** : Les systèmes embarqués se trouvent dans un marché très concurrentiel, et cette donne fait que le sujet est soumis à des mesures d'économie plus que dans d'autres systèmes informatiques. Pour prendre une place dans ce marché, un producteur doit non seulement accélérer le temps de mise sur le marché mais aussi maintenir le prix par unité au minimum.
- ***Spécificité*** : L'interface de dialogue avec l'utilisateur est spécifique : dans la majorité des cas, un système embarqué n'utilise pas les interfaces classiques clavier/souris propres à la micro-informatique. S'ils existent, les périphériques

d'affichage sont souvent limités à des panneaux de petite taille de type LCD³, et les périphériques d'entrée à quelques boutons poussoirs ou autres composants inhabituels dans l'informatique traditionnelle.

- **Faible encombrement** : Le plus souvent, mais ce n'est pas obligatoire, un système embarqué est de petite taille si on le compare aux volumes importants atteints par les systèmes classiques multiusages comme en bureautique, la raison en est double:
 - La nécessité de fiabilité citée précédemment cohabite mal avec un grand volume: plus on écrit de lignes de codes, plus on a de chances que celles-ci contiennent des bugs.
 - Le logiciel est souvent embarqué dans des équipements produits à grande échelle sur lesquels le moindre écart de coût dû à un bug imprévu du logiciel peut avoir de fortes répercussions.
- **Faibles ressources** : Les systèmes embarqués sont généralement limités par leurs ressources spatiales. Etant utilisés comme composants enfouis dans des systèmes plus complexes et étant conçus pour réaliser des tâches spécifiques, des mémoires de petite taille et des unités de calcul d'une faible puissance sont souvent suffisantes.
- **Communication** : Généralement les objets pour lesquels les systèmes embarqués sont conçus, sont des objets communicants, ainsi ces systèmes sont dotés de modules (GSM⁴, Bluetooth, GPS⁵...) qui leur permet de communiquer et de recevoir des informations sur de longues ou petites distances.

Lors du développement et de la conception d'un système embarqué, ces caractéristiques sont considérées comme étant des contraintes, des facteurs à prendre en considération afin d'aboutir à un système optimal. Ces mesures concurrencent souvent l'une avec l'autre, en effet, une amélioration de l'une conduit souvent à une dégradation dans une autre. Par exemple, si nous réduisons la taille d'une implémentation, sa performance peut en souffrir.

³ LCD (Liquid Crystal Display) : écran à cristaux liquides.

⁴ GSM : Global System for Mobile communication.

⁵ GPS : Global Positioning System.

IV. Les types de systèmes embarqués :

On peut énumérer les types de systèmes embarqués, et cela en tenant compte de leur rôle, comme suit :

- **Calcul général** : Ce sont des applications similaires aux applications de bureau mais empaquetées dans un système embarqué. Exemple : jeu vidéo, set-top box.
- **Contrôle de systèmes** : Des applications dédiées au contrôle de systèmes en temps réel. Exemple : moteur d'automobile, traitement chimique, traitement nucléaire, système de navigation aérien.
- **Traitement du signal** : Calcul sur de grosses quantités de données. Exemple : GPS, radar, sonar, compression vidéo.
- **Télécommunication et réseaux** : Transmission d'information et commutation. Exemple : téléphone, internet.

V. Architecture matérielle d'un système embarqué :

Un système embarqué dispose d'une architecture semblable à celle des ordinateurs standards, en effet, on y retrouve les mêmes composants (processeur, mémoires, dispositifs d'entrée/sortie) interconnectés à l'aide des bus (Fig.I.1).

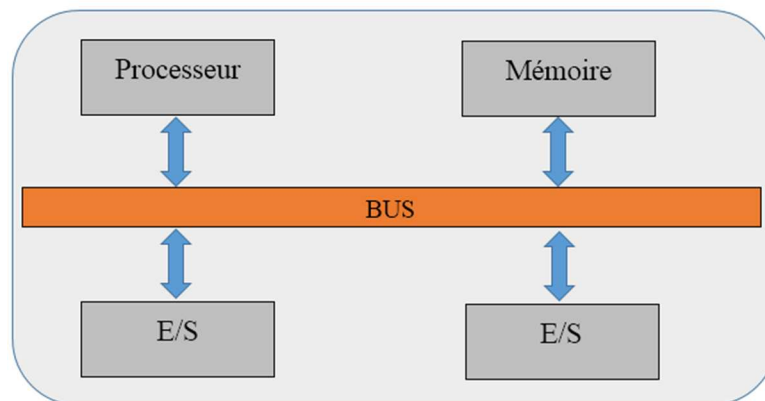


Fig.I.1 Vue simplifiée de l'architecture d'un système embarqué.

Des différences existent entre les systèmes embarqués et les ordinateurs conventionnels, ainsi, d'autres composants viennent compléter l'architecture simplifiée. Ces composants répondent aux caractéristiques spécifiques et permettent la réalisation des tâches propres à ces systèmes. L'architecture typique d'un système embarqué peut être représentée par le schéma de la figure I.2.

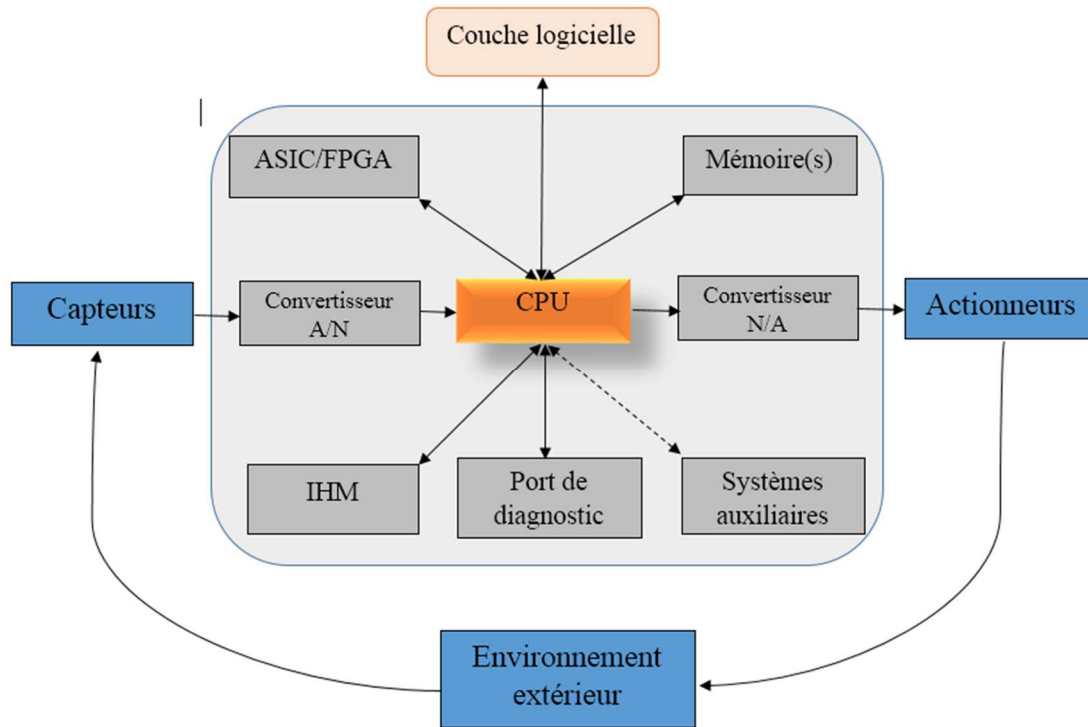


Fig.I.2 Architecture typique d'un système embarqué.

On trouve dans cette architecture:

- Capteurs (interrupteurs, antennes, etc.) couplés à des convertisseurs analogique/numérique.
- Actionneurs (LED, etc.) couplés à des convertisseurs numérique/analogique.
- CPU⁶ (processeur embarqué et ses E/S).
- Un/des FPGAs⁷ et/ou ASICs⁸ pour jouer le rôle de coprocesseurs. (accélération matérielle).

⁶ CPU : Central Processing Unit.

⁷ FPGA (Field-Programmable Gate Array) : Circuit intégré composé de nombreuses cellules logiques élémentaires reprogrammables après fabrication.

⁸ ASIC (Application-Specific Integrated Circuit) : Circuit intégré configure pour réaliser une tâche spécifique.

- Ports de diagnostic utilisés pour la configuration du système et le débogage (USART⁹, JTAG¹⁰, etc.)
- Interfaces Homme-Machine (écran LCD, etc.) permettant à l'utilisateur d'interagir avec le système.
- Des systèmes auxiliaires qui peuvent compléter le système existant afin de réaliser plus de fonctionnalités.

Cette architecture peut varier selon les systèmes : par exemple l'IHM utilisée dans un système diffère de celle utilisée dans un autre.

On présente dans ce qui suit les composants matériels essentiels d'un système embarqué :

V.1. Les mémoires :

Il existe plusieurs types de mémoires qu'on peut classer selon différents critères :

➤ **Les mémoires non volatiles** : sont des mémoires qui conservent leurs données en l'absence d'alimentation électrique, on trouve deux types :

- La *ROM (Read Only Memory)*: Caractérisées par leur petite taille notamment dans les systèmes embarqués, les ROMs contiennent le programme de démarrage du système (le bootstrap) qui charge généralement un autre programme (système opératoire) qui est mis au préalable dans la mémoire flash du système.

Par convention la ROM est à lecture seule, mais des ROMs programmables existent : la PROM (Programmable Read Only Memory) sur laquelle on peut écrire mais qu'une seule fois. La EPROM (Erasable Programmable Read Only Memory), la EEPROM (Electrically Erasable Programmable Read Only Memory) sont des mémoires non volatiles qui peuvent être effacées et réécrites en utilisant des dispositifs spécifiques, elles sont souvent utilisées durant les phases de développement des bootstrap.

⁹ USART (Universal Synchronous/Asynchronous Receiver Transmitter) : interface de communication série en mode synchrone ou asynchrone.

¹⁰ JTAG (Joint Test Action Group) : utilisé au lieu du terme générique Boundary Scan pour désigner le port de teste des cartes électroniques.

- *La mémoire FLASH* : La mémoire flash est en lecture écriture, elle fournit une grande densité de sauvegarde comparée à la ROM avec un temps d'accès équivalent aux RAMs (Random Access Memory) lentes.

Elle contient le logiciel du système embarqué qui est chargé vers la RAM pour exécution.

On trouve deux types : La NOR-FLASH (petite densité de sauvegarde mais rapide) et La NAND-FLASH (grande densité de sauvegarde mais lente).

- **Les mémoires volatiles (RAMs)** : sont des mémoires qui ont besoin d'alimentation électrique continue pour conserver les données qui y sont enregistrées, elles sont à accès aléatoire en lecture et en écriture.

On retrouve deux types de mémoires volatiles :

- *Mémoires dynamiques (DRAM : Dynamic Random Access Memory)* : grande densité de sauvegarde, relativement lentes, elles contiennent le programme en exécution et ses données de travail.
- *Mémoires statiques (SRAM : Static Random Access Memory)*: petite densité de sauvegarde, très rapides, coutent cher, utilisées surtout dans les mémoires cache et Scratchpad pour exécuter des portions de code les plus fréquentes ou les plus critiques.

V.2. Les périphériques d'entrée/sortie :

Contrairement aux ordinateurs conventionnels, les périphériques des systèmes embarqués sont plus nombreux mais leur rôle est le même c.à.d. qu'ils permettent au processeur d'être en communication avec l'environnement extérieur à l'aide des interfaces d'entrée/sortie.

Les périphériques typiques des systèmes embarqués sont :

- Interfaces de communication série : USART, I2C¹¹, SPI¹², USB¹³, etc.
- Les cartes multimédias (cartes SD, Compact Flash, etc.) ;

¹¹ I2C ou IIC : Inter Integrated Circuit.

¹² SPI: Serial Peripheral Interface.

¹³ USB : Universal Serial Bus.

- Réseaux: Ethernet, etc.
- Timers ;
- Entrée/sortie discrète: General Purpose Input / Output (GPIO) ;
- Convertisseur Analogique-numérique / convertisseur numérique-analogique (ADC / DAC) ;

V.3. Les processeurs :

Etant donnée la grande variété d'applications embarquées, une grande variété de processeurs est utilisée. Il y a des processeurs à faible puissance, lents, peu coûteux, mais aussi des processeurs à haute performance, à usage spécial...etc. Cette section donne un aperçu de certains des types de processeurs disponibles.

➤ *Les microcontrôleurs :*

Un microcontrôleur rassemble les éléments essentiels d'un ordinateur sur un même circuit intégré, il comprend une unité de traitement (CPU) à laquelle des périphériques sont associés tels que des mémoires, des dispositifs d'entrée/sortie, des timers...etc.

Les microcontrôleurs simples opèrent sur des mots de 8 bits et ils conviennent pour les applications nécessitant de petites mémoires et de simples fonctions logiques. Ils ont une très faible consommation d'énergie, et comprennent souvent un mode de veille (sleep mode) qui permet de réduire plus cette consommation.

Les processeurs 16 bits sont relativement peu répandus. La tendance actuelle est de s'éloigner de plus en plus des microcontrôleurs 8 bits pour faire place aux UC 32 bits et ce, essentiellement à cause des prix plus qu'intéressants de ces derniers (le processeur 32 bits le moins cher peut ne coûter qu'un seul euro). De plus, ils proposent des avantages de puissance et de vitesse de calcul supérieure avec une taille mémoire assez grande.

➤ ***Les processeurs DSP***

Les DSPs (Digital Signal Processors) sont des processeurs utilisés dans le traitement du signal. Initialement utilisés pour gérer la carte son d'un ordinateur, les DSPs ont vu leurs utilisations s'accroître considérablement depuis 1985, tout d'abord grâce au développement des télécommunications (téléphonie numérique, puis téléphonie sans fil GSM ...), puis grâce à leurs possibilités de traitement rapide de certaines commandes numériques faisant appel à des algorithmes complexes permettant ainsi le travail en « temps réel ».

Les DSPs ont une architecture optimisée pour traiter une grande quantité de données en parallèle à chaque cycle d'horloge.

➤ ***Les processeurs graphiques GPU***

L'unité de traitement graphique GPU (Graphics Processing Unit) est un processeur dédié au traitement des données graphiques. Ces processeurs remontent aux années 1970, d'abord utilisés à des fins de décryptage, ils ont évolué vers des modèles plus généraux de programmation, et donc ont commencé à apparaître dans d'autres applications de calcul intensif, comme l'instrumentation (accélérateur de particules). Les GPU sont généralement assez gourmands en énergie, et donc aujourd'hui ne sont pas un bon challenge pour les applications embarquées à contraintes énergétiques.

V.4. Les Systèmes On Chip :

Un SOC (Système On Chip) est un circuit intégré plus spécialisé qu'un microcontrôleur classique, il contient un ensemble d'éléments (processeurs, mémoires, entrées/sorties) interconnectés pour réaliser une fonction (ou un type de fonctions) spécifique.

Comme exemple, on peut citer : Les puces téléphoniques, les systèmes d'acquisition intégrés, les systèmes intégrés pour ordinateur de bord...etc.

V.5. Les Systèmes On Module :

Un SOM (Système On Module) est un système autonome et ne nécessite qu'un minimum d'ajout pour fonctionner, en effet, ce système est une association d'un SoC et d'un ensemble d'éléments supplémentaires comme une extension de mémoire, d'autres périphériques, des connecteurs pour les périphériques internes du SoC, ...etc. Le SoC et ses extensions sont implémentés sur un circuit imprimé (PCB¹⁴).

VI. Langages de développement utilisés : [SE4]

Pour des raisons évidentes de contraintes matérielles, le langage assembleur a longtemps été le choix de prédilection des technologies de l'embarqué car il permettait à la fois d'optimiser la taille du code généré mais aussi ses performances et il permettait aux développeurs d'avoir un contrôle total sur le processeur et le matériel.

La gestion d'un projet complexe écrit en assembleur est cependant difficile et coûteuse, et le code assembleur manque de portabilité, cela a conduit à l'utilisation de l'assembleur principalement comme un complément aux langages de haut niveau, de plus, l'évolution des performances du matériel et des compilateurs permet aujourd'hui de se tourner vers des solutions plus confortables.

Aujourd'hui, le C est le langage de programmation le plus utilisé, en effet des compilateurs sont disponibles pour presque chaque processeur en usage. On trouve également un très grand corps de programmeurs expérimentés en C. En outre, le C a l'avantage d'être indépendant du processeur ce qui permet aux programmeurs de se concentrer sur les algorithmes et les applications, plutôt que sur les détails de l'architecture d'un processeur particulier. Le langage C est bien adapté au logiciel embarqué car il permet une programmation relativement proche du matériel (a une nature « de bas niveau »), il donne aux programmeurs en systèmes embarqués un haut degré de contrôle direct sur le matériel sans sacrifier les avantages des langages de haut niveau.

Le langage C++ est aussi l'un des choix favori des développeurs, il représente la version orientée objet du C qui offre une meilleure abstraction des données.

D'autres langages orientés objet sont utilisés, et celui qui s'impose est le Java notamment Java Android qui est utilisé pour développer des applications destinées au système qui porte le

¹⁴ PCB (Printed circuit board): une plaque permettant de maintenir et de relier un ensemble de composants électroniques.

même nom (Android), et qui est l'un des systèmes les plus connus pour la téléphonie mobile et les tablettes.

Dans le cas de l'utilisation de systèmes de type Unix, on pourra également employer d'autres langages de programmation ou langages de scripts plus appropriés dans certains cas. On citera en particulier le shell-script, langage de script d'Unix (ou Bourne shell) qui associé à d'autres commandes pourra se révéler très utile dans l'écriture de procédures système.

D'autres langages de scripts célèbres dans le monde Unix comme Python, Perl ou Tcl/Tk sont aussi utilisés dans les environnements embarqués.

VII. Systèmes d'exploitation pour systèmes embarqués : [SE4]

Les systèmes d'exploitations offrent les possibilités de bénéficier des mêmes avancées technologiques que les applications classiques (TCP/IP¹⁵, HTTP¹⁶...) tout en augmentant la portabilité du système quoiqu'au détriment de l'empreinte mémoire.

Certains parlent de logiciel embarqué alors qu'il est fréquent d'entendre la terminologie de système embarqué. Cette terminologie désigne le plus souvent un système d'exploitation, version complexe et multi-usage du concept de logiciel. Un système d'exploitation inclut un ensemble de programmes permettant :

- de gérer les ressources matérielles en assurant leurs partages entre un ensemble plus ou moins grand d'utilisateurs ;
- d'assurer un ensemble de services en présentant aux utilisateurs une interface mieux adaptée à leurs besoins que celle de la machine physique.

On considère souvent qu'un système d'exploitation est beaucoup trop complexe et surdimensionné pour remplir les tâches d'un dispositif embarqué. C'est vrai dans certains cas de spécificité extrême du logiciel à embarquer ou de fortes contraintes matérielles mais il est également vrai que l'amélioration des performances du matériel permet dans un grand nombre de cas d'utiliser un système d'exploitation adapté au lieu d'un simple logiciel dédié.

D'un point de vue technique, le champ d'application peut être divisé en deux grandes familles :

- le contrôle de processus sans contrainte ou à faible contrainte temps réel ;
- le contrôle de processus avec contrainte temps réel.

¹⁵ TCP/IP : Transmission Control Protocol / Internet Protocol.

¹⁶ HTTP : HyperText Transfer Protocol.

Cette séparation est fondamentale car elle déterminera complètement le choix des composants logiciels à utiliser. Dans le premier cas, on pourra envisager l'utilisation d'un système d'exploitation dérivé d'un système classique comme Linux. L'adaptation se situera principalement au niveau du développement de pilotes de périphériques et de l'optimisation du système en taille ou en performances. Dans le second cas, les contraintes matérielles nécessiteront l'utilisation de composants spécialisés, soit un logiciel spécifique, soit un système d'exploitation dit temps réel (RTOS).

➤ Exemples de systèmes d'exploitation:

Il est difficile de recenser les nombreux systèmes d'exploitation embarqués existants d'autant que certains industriels ont parfois développé leurs propres systèmes afin de satisfaire des besoins ou des normes très particulières (cas de l'aéronautique). En voici quelques exemples :

- **VxWorks** : Noyau RT le plus utilisé dans l'industrie. Supporte TCP/IP et une API socket.
- **QNX** : Noyau RT de type UNIX. Intègre une interface graphique proche de XWindow. QNX peut être utilisé gratuitement pour des applications non commerciales. Très faible empreinte mémoire.
- **uC/OS** : Destiné à des microcontrôleurs type Motorola 68HC11. Support de TCP/IP. Gratuit pour l'enseignement.
- **Windows CE** : Victime d'une réputation de fiabilité approximative, est pour l'instant cantonné à l'équipement de nombreux assistants personnels (PDA¹⁷)
- **Lynx OS** : Système RT conforme à la norme POSIX¹⁸.
- **Nucleus** : Noyau RT avec support de TCP/IP, une interface graphique, un navigateur web et serveur HTTP.
- **eCos** : Noyau RT, faible empreinte mémoire, basé sur Linux, support de TCP/IP. Licence proche GPL¹⁹. Largement utilisé (automobile, imprimante, lecteur MP3). Support des processeurs x86, PowerPC, SH3 Hitachi ou StrongARM.

¹⁷ PDA : Personal Digital Assistant.

¹⁸ POSIX (Portable Operating System Interface X): famille de normes techniques de standardisation des interfaces de programmation des logiciels destinés à fonctionner sur les variantes du système d'exploitation UNIX.

¹⁹ GPL (General Public License): est une licence qui fixe les conditions légales de distribution des logiciels libres du projet GNU.

VIII. Exemples d'application:

Le champ d'application des systèmes embarqués est très vaste. En effet, de nombreuses fonctions autrefois réalisées par des systèmes mécaniques ou du moins analogiques sont aujourd'hui remplacées par des composants logiciels, même rudimentaires.

Au niveau applicatif, les systèmes embarqués se retrouvent sur les sujets suivants :

- Contrôle de processus industriels ;
- Commande numérique;
- Automobile ;
- Télécommunications : centraux téléphoniques, téléphones mobiles ;
- Réseaux informatiques : routeurs, systèmes ;
- Périphériques informatiques : imprimantes, photocopieurs ;
- Aéronautique et transports en général ;
- Systèmes médicaux ;
- Multimédia : audio, vidéo, TV ;
- Domotique.

Dans ce qui suit, on présente deux exemples détaillés des domaines d'application des systèmes embarqués. [SE5]

➤ Voitures & transports

C'est dans ce domaine que s'applique une très grande quantité d'innovations. Il suffit d'observer l'expression des besoins et les souhaits de l'automobiliste : il exige de sa voiture, sécurité, fiabilité, communicabilité, personnalisation, assistance, services, sobriété, respect de l'environnement et distraction. Il souhaite aussi de nombreuses fonctions automatisées : créneaux simplifiés par exemple, contrôle de vitesse et de distance entre véhicules, communication actualisée avec l'infrastructure routière... Autant d'exigences, autant de systèmes embarqués composites, inter-opérants, à fiabilité et capacité accrues.

Les spécialistes de l'électronique, de l'informatique embarquées et de la communication mobile savent d'ores et déjà réaliser tout ça, ou alors ils sont en train de le développer.

Outre la voiture, les projets innovants faisant appel aux systèmes embarqués se portent également sur le transport professionnel : fret et logistique sur route (avec le suivi temps réel de flotte et l'optimisation des tournées); trains, métro, tramway, bus (de l'informatisation des

aiguillages aux systèmes vidéo de sécurité des passagers en passant par les fonctions intelligentes d'économie d'énergie).

➤ **Aide à la personne**

Assistance et sécurité : les systèmes embarqués ont un rôle-phare à jouer avec, et autour de l'Homme, dans sa sphère professionnelle comme personnelle.

Au chapitre de la mobilité, le citoyen se veut joignable à chaque instant avec les téléphones portables. Ce terminal, fer de lance de l'embarqué grand public, a pris en quelques années une grande importance : il a su dépasser sa fonction première téléphonique et empiler les applications : photo, vidéo numériques, lecteur de musique, terminal de présence, GPS, outil de téléchargement et d'échange de fichiers, boîte de réception et d'envoi de mails, messagerie instantanée, terminal de consultation de réseaux sociaux, terminal de paiement sécurisé, de paiement du transport, d'assistance...etc.

Il y a aussi toutes les applications professionnelles, et celles des PDAs : connexion mobile et sécurisée au système d'information de l'entreprise, pour les commerciaux itinérants, les techniciens, les collaborateurs nomades et les télétravailleurs.

Conclusion :

Ce chapitre nous a permis de voir que les systèmes embarqués ont certaines particularités en ce qui concerne leur architecture et leurs systèmes d'exploitation, et cela leur permet de réaliser une tâche bien précise tout en respectant un bon nombre de contraintes. Devenus indispensables dans divers domaines, le développement des systèmes embarqués ne cesse de croître et d'attirer de nouveaux développeurs, les industriels sont en concurrence permanente, chose qui rend le marché de ces systèmes très intéressant.

CHAPITRE 2

Géolocalisation GPS/GSM.

Introduction :

L'Homme a toujours eu besoin de localiser les objets et de se situer dans l'environnement. Confronté à ce sujet, il a cherché des repères dans le ciel (étoiles et constellations), a cherché et créé des repères sur la planète (fleuves, sommets des montagnes...), a tracé des cartes, des itinéraires, a découvert et créé divers instruments (la boussole, le théodolite...).

À l'air des nouvelles technologies, ce besoin s'est accru et les exigences sont de plus en plus pointues. Aujourd'hui pour répondre à cette nécessité et pouvoir se localiser, on utilise les coordonnées géographiques, on parle alors de géolocalisation.

Les possibilités en termes de géolocalisation ont connu un développement inouï au cours des dix dernières années révolutionnant ainsi de nombreux domaines. L'alliance des nouvelles technologies telles que les terminaux mobiles (Smartphones, tablettes numériques...), les réseaux mobiles et le GPS a su optimiser l'usage de la géolocalisation. C'est à présent un outil majeur de confort personnel et une valeur ajoutée pour les professionnels, en effet, de plus en plus d'entreprises font appel à cette technologie. Par exemple, les sociétés de dépannage ou de taxis utilisent des systèmes leur permettant de localiser les véhicules de leurs employés.

La géolocalisation d'objets distants repose sur le principe d'utilisation d'équipements intelligents pouvant communiquer des informations vers un système informatique, c'est ce qu'on appelle le Machine To Machine (M2M).

Dans ce chapitre et dans le cadre de notre projet, nous allons définir la géolocalisation, nous détaillerons ensuite le système de positionnement par satellite le plus connu, le GPS. Les données GPS devant être transmises à distance, nous allons voir les réseaux utilisés notamment le GSM. Ce chapitre porte aussi sur les outils de localisation et leur utilisation pour la sécurisation des véhicules.

I. La géolocalisation :

La géolocalisation est un procédé utilisant les coordonnées géographiques pour localiser un objet (personne, véhicule...etc.), la position peut être visualisée sur une carte à travers une plateforme de géolocalisation le plus souvent accessible depuis internet.

Un objet peut être localisé à l'aide : [GEO1]

- Du système de positionnement par satellites, un terminal équipé d'une puce compatible reçoit les signaux des satellites de la constellation puis calcule sa position et la traduit en termes de latitude, longitude et parfois altitude. Le GPS (développé un peu plus loin) est le système de positionnement par satellite le plus répandu, mais d'autres systèmes existent tels que le GLONASS qui est entièrement opérationnel et accessible, Galileo et Beidou sont respectivement les alternatives européenne et chinoise en cours de déploiement.
- Des ondes radios, cela consiste à se localiser par rapport à un point fixe (ou plusieurs points fixes) dont la position est connue. Par exemple pour localiser un terminal GSM, il suffit d'identifier l'antenne relais à laquelle il est connecté, et pour plus de précision, il faut connaître la position de plusieurs cellules, ainsi pouvoir estimer que l'équipement mobile se trouve dans la zone d'intersection des cellules.

L'appareil ayant calculé sa position, celle-ci peut être stockée au sein du terminal et être extraite postérieurement, ou être transmise en temps réel vers une plateforme logicielle de géolocalisation. La transmission temps réel nécessite un terminal équipé de modules lui permettant de communiquer à distance, par exemple, via les réseaux de téléphonie mobile de type GSM, GPRS¹ ou UMTS².

II. Le système de coordonnées géographiques : [GEO2]

Un système de coordonnées géographiques (GCS³) utilise une surface sphérique à trois dimensions pour définir des emplacements sur la terre. Un point est référencé d'après ses valeurs de longitude, de latitude et d'altitude (Fig.II.1.). La longitude et la latitude correspondent aux angles mesurés depuis le centre de la terre vers un point de surface.

¹GPRS : General Packet Radio Service

² UMTS : Universal Mobile Telecommunications System

³ GCS : Geographic Coordinate System.

La latitude est une mesure angulaire ; elle varie entre la valeur 0° à l'équateur et 90° aux pôles. Tous les endroits ayant une latitude donnée sont désignés collectivement sous le nom de parallèle géographique, car tous ces lieux sont placés sur une ligne parallèle à l'équateur.

La longitude est une mesure angulaire sur 360° par rapport à un méridien de référence, avec une étendue de -180° Ouest à $+180^\circ$ Est, le méridien 0° est celui de Greenwich. Le méridien est donc une ligne reliant le pôle nord au pôle sud et à laquelle tous les points de même longitude appartiennent.

L'altitude est l'élévation verticale d'un lieu ou d'un objet par rapport à un niveau de base, par convention, sur terre ce niveau est le plus souvent le niveau de la mer (ou « niveau zéro »).

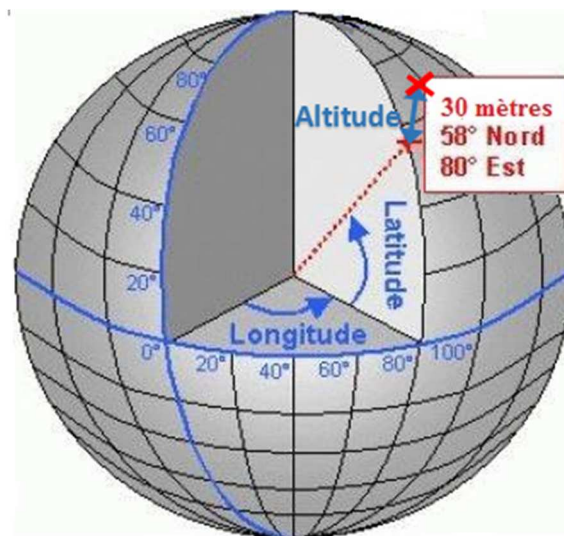


Fig.II.1. Les coordonnées géographiques.

III. Le GPS

III.1. Définition :

GPS, En anglais «Global Positioning System» que l'on peut traduire en français par «système de localisation mondial», est un système de positionnement par satellites fonctionnant au niveau mondial et accessible au grand public. Ce système a été théorisé par le physicien D.Fanelli et mis en place à l'origine par le DoD (Department of Defence) [GPS/GSM-2].

Le GPS est un système capable de donner une position n'importe où sur le globe en temps réel, avec une précision de quelques mètres, de jour comme de nuit, et quelles que soient les conditions météorologiques. [GPS/GSM-1]

III.2. Historique : [GPS1]

- Avant la deuxième guerre mondiale, le premier système qui utilise le principe d'analyse de la différence du temps entre l'émission et la réception du signal radio a vu le jour, baptisé LORAN (LONG Range Aid Navigation), aujourd'hui ce même système est utilisé par le GPS.
- EN 1959, le premier satellite de radio navigation TRANSIT a été lancé.
- En Avril 1980, lancement de 18 satellites sur les 24 initialement prévus, pour des raisons financières le DoD a dû réduire la constellation GPS.
- En 1982, mise en service du système GPS uniquement pour utilisation militaire, à partir de 1983, il a été mis au service de l'aviation civile suite au crash du vol Korean Air 007.
- En 1988, les 21+3 satellites (3 de secours) ont été tous lancés et mis en service.
- En 1990, le premier récepteur GPS a été rendu public et le DoD a mis en place un système de dégradation SA⁴ (Selective Availability) qu'il a activé en juillet 1991 et enlevé en juin 2000.

III.3. La structure du système : [GPS/GSM-1] [GPS3]

Le système GPS est constitué de trois segments : le segment espace, le segment de contrôle et le segment utilisateur (Fig.II.2).

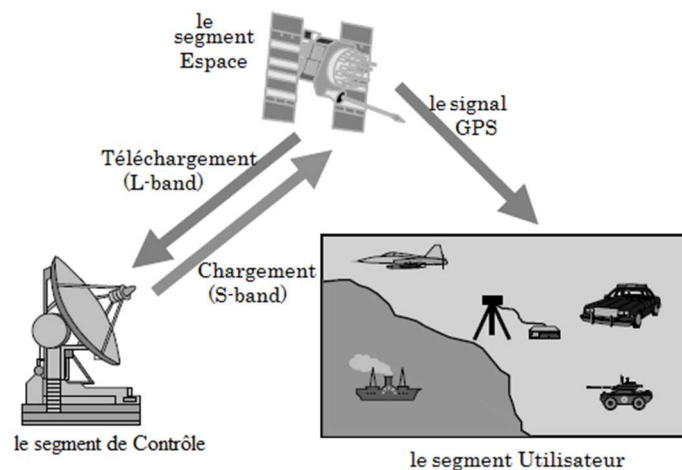


Fig.II.2. Structure du système GPS.

⁴ SA (disponibilité sélective) : certaines informations peuvent être volontairement dégradées et priver les récepteurs ne possédant pas les codes correspondants de la précision maximale

Le segment spatial consiste en la constellation des 30 satellites NAVSTAR⁵ (image de la figure II.3) de 500 à 800 kg, ils sont répartis sur 6 orbites à une altitude de 20 184 km qu'ils parcourent en 11 h 58 min 2s à la vitesse de 3 km/s. Chaque satellite possède une horloge. Les horloges de l'ensemble de la constellation sont parfaitement synchronisées (précision de 100 ns). Cette répartition spatiale garantit la visibilité en permanence d'au moins 6 satellites, en tout point du globe.

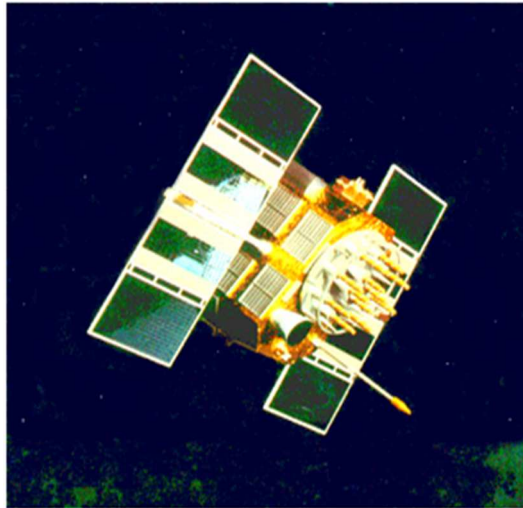


Fig.II.3. Satellite GPS

Chaque satellite envoie un signal qui permet à l'utilisateur de déterminer d'une part sa position et d'autre part la distance qui le sépare du récepteur GPS.

Le segment de contrôle du système GPS est constitué de cinq stations de surveillance (image de la figure II.4) dont la station de contrôle principale MCS (Master Control Station) située au Colorado springs (USA), la première tâche de ce segment est de surveiller les satellites afin de pouvoir déterminer leur localisation, contrôler leur bon fonctionnement mais aussi mettre à jour les informations transmises par ces derniers.

⁵ NAVSTAR Navigation Satellite Timing And Ranging.



Fig.II.4. Les stations de surveillance au sol.

Le **segment utilisateur** inclut tous les utilisateurs militaires et civils, ces derniers sont équipés de récepteurs GPS connectés à des antennes, ils reçoivent, décodent et traitent les signaux émis par les satellites afin de calculer leurs positions en coordonnées géographiques dans le système géodésique WGS 84⁶, le système ne peut être saturé et le nombre maximum d'utilisateurs GPS est illimité.

Une fois calculées, les données doivent être extraites et exploitées. Différents formats, standards ou non, permettent la récupération et l'intégration des données de localisation dans les applications les utilisant directement :

- Le format standard Rinex ;
- Le format standard NMEA 0183 (celui utilisé pour notre application et dont on donne plus d'informations dans L'annexe B) ;
- Les formats basés sur XML, NVML, POIX, GPSml... ;
- Les formats propriétaires (Garmin, Trimble...).

III.4. Principe de positionnement par GPS : [GPS/GSM-1] [GPS2]

Un récepteur GPS reçoit les signaux de tous les satellites de la constellation qui sont en visibilité, on définit ainsi des sphères centrées sur les satellites dont le rayon est égal à la distance satellite-récepteur, l'intersection de ces sphères donne la position du récepteur.

⁶ WGS 84 (World Geodesic System 1984) : système géodésique standard mondial.

Les informations envoyées par chaque satellite sont entre autre : le numéro d'identification du satellite, sa position et sa hauteur précise par rapport aux coordonnées géographiques terrestres (3 coordonnées) et l'heure exacte d'émission du signal.

Le récepteur GPS, grâce à son horloge synchronisée sur celle des satellites, détermine le temps que met une onde pour parcourir la distance qui le sépare d'un satellite et la multiplie par la vitesse de la lumière pour en déduire la distance (R_i) au satellite :

$$R_i = \text{Vitesse} \times (T - T_i)$$

Ou bien

$$R_i = ((X-X_i)^2 + (Y-Y_i)^2 + (Z-Z_i)^2)^{1/2}$$

T_i : date d'émission des ondes par le satellite

T : date de réception des ondes

X_i, Y_i, Z_i : coordonnées du satellite

X, Y, Z : coordonnées du récepteur.

Grâce à cette distance et la position d'un satellite il est possible de tracer un cercle imaginaire sur la surface de la terre.

Pour localiser un objet, théoriquement, il suffit de disposer au minimum de trois satellites visibles. En effet, deux satellites conduisent à l'estimation de deux positions dans l'espace, un troisième satellite est nécessaire pour déterminer la position exacte qu'occupe l'utilisateur (Fig.II.5). Pour connaître en plus l'altitude, un quatrième satellite est nécessaire.

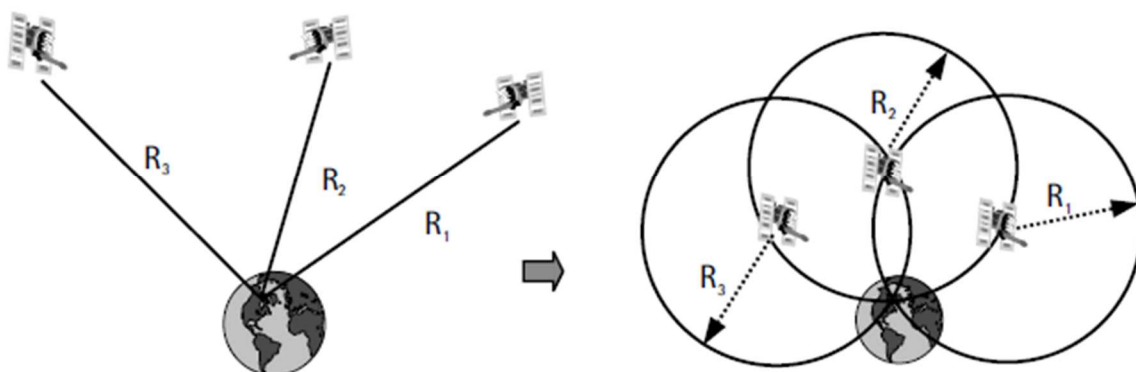


Fig.II.5. principe de positionnement par GPS

III.5. Les systèmes d'amélioration de la précision :

Le GPS a deux modes de fonctionnement : un mode précis de positionnement à priori réservé à des utilisateurs identifiés (code P pour *Precise code*; précision de l'ordre de 10m) et un mode standard de positionnement, sans restriction d'utilisation (code C/A pour *Coarse Acquisition*; précision de l'ordre de 100 m).

Le signal reçu par le récepteur GPS est quelque peu « dégradé », il diffère du signal émis, les principales sources d'erreurs sont :

- Le bruit du canal de transmission ;
- Les couches atmosphériques (troposphère et ionosphère) qui peuvent retarder ou accélérer les signaux ;
- le décalage temporel dû au temps de propagation et aux décalages des horloges de l'émetteur et du récepteur ;
- Le nombre de satellites en visibilité et leur répartition dans le ciel ont une influence sur la précision de la position.

Pour corriger en partie ces problèmes et améliorer les performances du système, on peut recourir au GPS différentiel ou DGPS (Differential Global Positioning System) et le GPS Assisté ou AGPS (Assisted Global Positioning System).

- **Le GPS différentiel (DGPS) :**

Le GPS différentiel ou DGPS (Differential Global Positioning System) utilise une référence au sol qui permet d'améliorer la connaissance de la position du récepteur. Son principe est basé sur le fait qu'en des points géographiquement proches, les effets des erreurs de mesure sont très semblables. Le fonctionnement du DGPS est comme suit :

- On place un récepteur GPS à un point connu avec grande précision.
- On lui raccorde un émetteur radio pour pouvoir envoyer la position mesurée.
- Le récepteur GPS client dispose également d'un récepteur radio et d'un calculateur.
- Le calculateur récupère les données fournies par les deux récepteurs GPS, et la position réelle de celui de référence lui étant connue, il calcule l'écart avec la position mesurée.
- Ce calcul lui permettra ensuite de corriger la position du récepteur mobile.

- **Le GPS assisté (AGPS) :**

La technologie A-GPS (Assisted GPS) ou WAG (Wireless Assisted GPS) fournit des données de localisation précises et fonctionne dans des lieux mal couverts par le GPS (bâtiments, canyons urbains, tunnels, etc.)

Un système A-GPS utilise les récepteurs GPS d'un opérateur pour aider le terminal mobile à connaître quels signaux GPS il doit suivre. Grâce à cette assistance, la recherche de signal effectuée par le terminal est grandement réduite. La durée nécessaire pour la première connexion ou TTFF (Time To First Fix) passe de plusieurs minutes à seulement quelques secondes. De plus, contrairement aux récepteurs GPS traditionnels, le récepteur A-GPS intégré dans le terminal est en mesure de détecter et démoduler des signaux de très faible magnitude. Le fonctionnement de l'AGPS est comme suit :

- Un serveur A-GPS hébergé chez l'opérateur suit en permanence et en temps réel le positionnement des satellites.
- Lors de sa mise en route, le terminal mobile envoie une requête via le réseau IP au serveur à l'aide du réseau mobile (GSM, GPRS, etc.) pour connaître quels signaux GPS il doit suivre.
- le mobile se met alors en liaison avec les satellites indiqués par le serveur pour connaître sa position. Dès lors, le terminal fonctionne de manière autonome.

Si le terminal perd tout signal GPS, il demande alors l'assistance du serveur A-GPS pour l'aider.

IV. Les moyens de communication des coordonnées géographiques :

Comme on l'a souligné précédemment, une fois la position d'un objet est connue, dépendamment de l'application, on peut soit la sauvegarder, soit la visualiser directement sur l'équipement si celui-ci dispose d'un écran ou alors la communiquer à distance via un réseau de communication tel que les réseaux mobiles GSM, GPRS et UMTS. Dans ce qui suit, nous allons voir les services offerts par ces réseaux qui permettent la transmission des informations d'un récepteur GPS.

- **Le GSM :** Le « Global System for Mobile communication » est le système cellulaire standard de la deuxième génération des réseaux mobiles. Adopté par plus de 180 pays, aujourd'hui, plus d'un milliard d'utilisateurs sont équipés d'une solution GSM. Parmi les services qu'offre ce réseau, le service SMS « Short Message Service » qui permet à un utilisateur de composer un message textuel à partir de son terminal mobile et de l'envoyer à un destinataire possédant également un téléphone mobile. Un peu plus loin, nous allons nous intéresser à l'architecture de ce réseau, mais aussi décrire le service SMS qui est utilisé pour communiquer les coordonnées GPS notamment par notre application.
- **Le GPRS :** Le « *General Packet Radio Service* » est une extension de l'architecture du standard GSM, elle s'appuie sur la technique de transmission par paquets et permet ainsi d'accéder aux services internet avec des débits théoriques maximums de l'ordre de 171,2 kbit/s (en pratique jusqu'à 114 kbit/s). Grâce au mode de transfert par paquets, les transmissions de données n'utilisent le réseau que lorsque c'est nécessaire. Le standard GPRS permet donc de facturer l'utilisateur au volume échangé plutôt qu'à la durée de connexion, ce qui signifie notamment qu'il peut rester connecté sans surcoût. [RES]
- **L'UMTS :** « Universal Mobile Telecommunications System » désigne une technologie retenue comme norme pour les systèmes de télécommunications mobiles dits de troisième génération. La 3G permet de profiter d'un très haut débit pour la transmission de données. Elle ouvre ainsi la porte à des applications et services nouveaux. Elle permet en particulier un accès internet performant et confortable et un transfert en temps réel des contenus multimédia tels que les images, le son et la vidéo. [RES].

V. Le GSM :

Ce standard pour les communications de téléphonie mobile permet de transmettre numériquement non seulement la voix, mais aussi des données informatiques (envoi de messages et connexion possible à internet).

V.1. Le concept cellulaire du GSM : [GPS/GSM-1]

Le principe du système cellulaire est de diviser le territoire en de petites zones, appelées cellules, au centre de chacune d'elles se situe une station de base (antenne d'émission) à laquelle on associe un certain nombre de fréquences.

La station de base communique par radio avec un mobile en lui allouant un canal de communication dès qu'il en a besoin, le terminal doit évidemment se trouver dans la zone couverte par la station.

Etant donné que le nombre de fréquences accordées au système GSM est restreint (il utilise les bandes de fréquences 900 MHz et 1800 MHz), l'opérateur est obligé de réutiliser les mêmes fréquences sur des cellules suffisamment éloignées de telle sorte que deux communications utilisant la même fréquence ne se brouillent pas.

V.2. Architecture d'un réseau GSM : [GPS/GSM-1]

Le réseau mobile se décompose en trois parties distinctes :

- **Le sous-système Radio (BSS : Base Station Subsystem)** qui assure les transmissions et gère la ressource radio ; il comprend essentiellement les stations de base (BTS : Base Transceiver Station), et leurs contrôleurs (BSC : Base Transceiver Station).
- **Le sous-système Réseau (NSS : Network Switching Subsystem)** qui comprend l'ensemble des fonctions nécessaires à l'établissement des appels et à la mobilité est essentiellement constitué de MSC (Mobile Service Switching Centre), VLR (Visitor Location Register), HLR (Home Location Register) ;
- **Le sous-système d'exploitation (OSS : Operation Support Subsystem)** dont l'OMC (Operation and Maintenance Center) l'équipement principal qui permet à l'opérateur d'administrer son réseau.

Le centre des messages courts SMSC (Short Message Service Center), localisé au niveau du sous-système réseau (NSS), fonctionne comme une plateforme d'enregistrement et de transfert des SMS.

V.3. L'interface SMS du GSM : [GSM1]

Pour une communication vocale, le canal de contrôle est utilisé pour initier la communication qui est ensuite transmise sur un canal de trafic. Les SMS, par contre, sont acheminés directement sur un canal de contrôle. Chaque message (limité théoriquement à 160 caractères) envoyé transite vers un centre de messagerie (SMSC) où il est temporairement stocké, c'est ce qu'on appelle le SMS-SUBMIT. Dès que le destinataire est disponible, c'est-à-dire lorsque le mobile est sous tension et présent dans une zone couverte par le réseau GSM, le message est transmis, on parle alors du SMS-DELIVER (plus de précisions sur les protocoles SMS-SUBMIT et SMS-DELIVER sont données dans l'annexe A). Les SMSC sont identifiés par un numéro d'appel spécifique à chaque opérateur.

Un message SMS peut être transmis de deux façons :

- En mode PDU (Protocol Description Unit) qui est une suite de caractères hexadécimaux qui codifient le SMS ;
- En mode TEXT qui est une représentation sous forme de texte des données qui composent le SMS.

Différents types d'alphabets sont utilisés pour passer du mode PDU au mode TEXT. Par exemple le téléphone portable détermine automatiquement quel type d'alphabet utiliser pour afficher en mode TEXT, sur son écran, les données d'un éventuel SMS reçu.

Il faut noter que les protocoles SMS-SUBMIT et SMS-DELIVER sont des PDU.

VI. La géolocalisation GPS/GSM : [GPS/GSM-1]

Les systèmes de géolocalisation GPS/GSM reposent sur le principe suivant : Une demande (un appel ou un SMS) d'information est adressée par le biais du réseau GSM à l'aide d'un terminal GSM. Le dispositif à localiser, équipé d'un récepteur GPS, reçoit la demande et calcule en temps réel sa position et la renvoie sous forme de message SMS par le réseau GSM.

L'affichage de la position du récepteur, et donc des coordonnées géographiques, se fera sur l'écran du terminal ayant émis la demande (ou sur un autre) à l'aide d'un logiciel approprié.

La géolocalisation GPS/GSM est l'une des applications de la technologie M2M, en effet, on constate qu'il y a quatre opérations principales : demande d'information à un équipement, celui-ci récolte la donnée demandée qui est ensuite traitée et communiquée. (La technologie M2M est présentée dans l'annexe C).

Il faut noter que notre système est une solution M2M qui repose sur la géolocalisation GPS/GSM, le terminal qui envoie la demande sous forme d'un SMS est un téléphone et le récepteur GPS est inclus dans le module SIM908.

VI.1. Géolocalisation GPS/GSM pour les systèmes antivol de véhicules :

Parce qu'ils coûtent de moins en moins chers et peuvent s'avérer très utiles, tant au niveau privé qu'au niveau professionnel, les dispositifs de géolocalisation sont en plein essor et les applications en plein développement et diversification, cela va de la balise pour marin au suivi des animaux dans la nature en passant par la boussole pour touriste. Pouvoir localiser et se localiser permet de répondre au besoin de sécurité, notamment des biens. Ainsi pour pouvoir lutter contre les vols de véhicules, les systèmes de géolocalisation sont de plus en plus répandus.

Un petit boîtier embarqué dans la voiture, permettra à son propriétaire de :

- Demander la position actuelle de son véhicule, le boîtier joue alors le rôle d'un localisateur (le cas de notre système).
- Définir un rayon autour de la position actuelle du boîtier, dès que celui-ci dépasse ce rayon, le propriétaire est mis au courant. Cette fonction est appelée le «geofencing» (géorepérage).
- Demander une trace du véhicule, le boîtier joue alors le rôle d'un traceur qui vérifie et envoie régulièrement sa position.

VI.2. Outils de géolocalisation GPS/GSM :

Beaucoup d'objets du quotidien sont équipés d'une antenne GPS, smartphones, PDA, tablettes, appareils photo... cette antenne est soit interne ou externe (elle est alors connectée à l'équipement via un port USB par exemple), et dans les deux cas un logiciel de cartographie est nécessaire. Des dispositifs généralement conçus pour une architecture M2M intègrent en plus

d'un module GPS, un module GSM (ou GPRS/EDGE⁷). Parmi ces dispositifs on retrouve ceux de la société SIMCOM dont on peut citer le sim908, le sim928A et le sim968, des modules sont aussi proposés par TELIT tels que le GM862-gps et le GE864-gps ...etc.

Conclusion :

Différents moyens sont utilisés pour localiser divers objets, notamment les véhicules. La position de ce dernier peut être aisément connue grâce à la géolocalisation GPS/GSM, chose qui est très utile en cas de vol. Cette technologie utilise des systèmes qui doivent répondre aux différentes exigences de leurs utilisateurs, fiabilité, facilité d'utilisation et surtout à des prix abordables. Ainsi le choix des composants de ces systèmes et leur développement sont des démarches d'une grande importance.

⁷ EDGE (Enhanced Data Rates for GSM Evolution) : evolution du GPRS et considéré comme une technologie pré-3G.

Chapitre 3

Conception du système.

Introduction :

Les statistiques mondiales des vols de voitures montrent que ce type de criminalité est en hausse constante, dans notre pays la moyenne des vols est de trois voitures par jour sur l'ensemble des 48 wilayas. De ce fait, on se retrouve face à une situation qui exige de nouvelles technologies et de nouvelles techniques pour assurer la sécurité des véhicules. Les industriels se penchent de plus en plus sur les systèmes de localisation qui sont devenus des composants indispensables dans les voitures modernes. Ainsi notre travail consiste à développer un système qui exploite les techniques de positionnement par GPS pour pouvoir localiser le véhicule en cas de vol et utilise le réseau GSM pour communiquer la position.

Dans ce chapitre, nous allons étudier le problème et analyser l'aspect matériel et logiciel du travail à réaliser afin de définir les besoins et les différentes fonctionnalités qui doivent être assurées. Nous allons ensuite procéder à la résolution du problème dans la partie conception à travers la conception matérielle et logicielle. Cela va nous permettre de modéliser le système, c.à.d. le représenter de manière simplifiée et abstraite.

I. Méthodologie de conception :

Une méthodologie de conception correspond à un ensemble de méthodes que doit suivre le concepteur afin de réaliser son système, dans notre cas, nous allons suivre un processus de conception typique [Con1], celui-ci consiste en la spécification des besoins, la conception, le codage et les tests de validation.

Dans ce contexte, un langage de modélisation est souvent utilisé. En ce qui nous concerne, pour la modélisation de la partie logicielle, nous allons nous inspirer du support graphique du langage UML afin de mieux expliquer les caractéristiques structurelles et comportementales de notre système. Notre méthodologie peut être décrite par le schéma de la figure III.1:

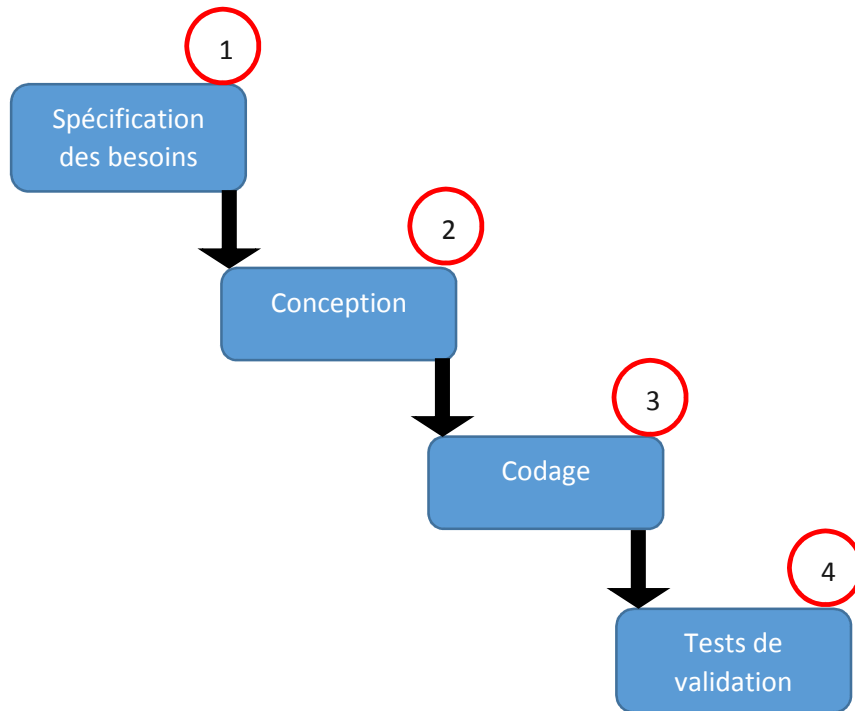


Fig. III.1. Schéma de la méthodologie de conception.

II. Analyse :

Dans cette partie, nous allons étudier le système c.à.d. recueillir et organiser les informations et les connaissances nécessaires à la définition des fonctionnalités et besoins auxquels le système doit répondre.

II.1. Idée générale sur notre travail :

Notre travail consiste à concevoir et développer un système à embarquer dans un véhicule afin de pouvoir le localiser à n'importe quel moment et à n'importe quel instant. Ainsi, comme on peut le voir sur la figure III.2, en cas de vol, via le réseau mobile GSM, l'utilisateur envoie un message vers le numéro de la carte SIM du système (1).

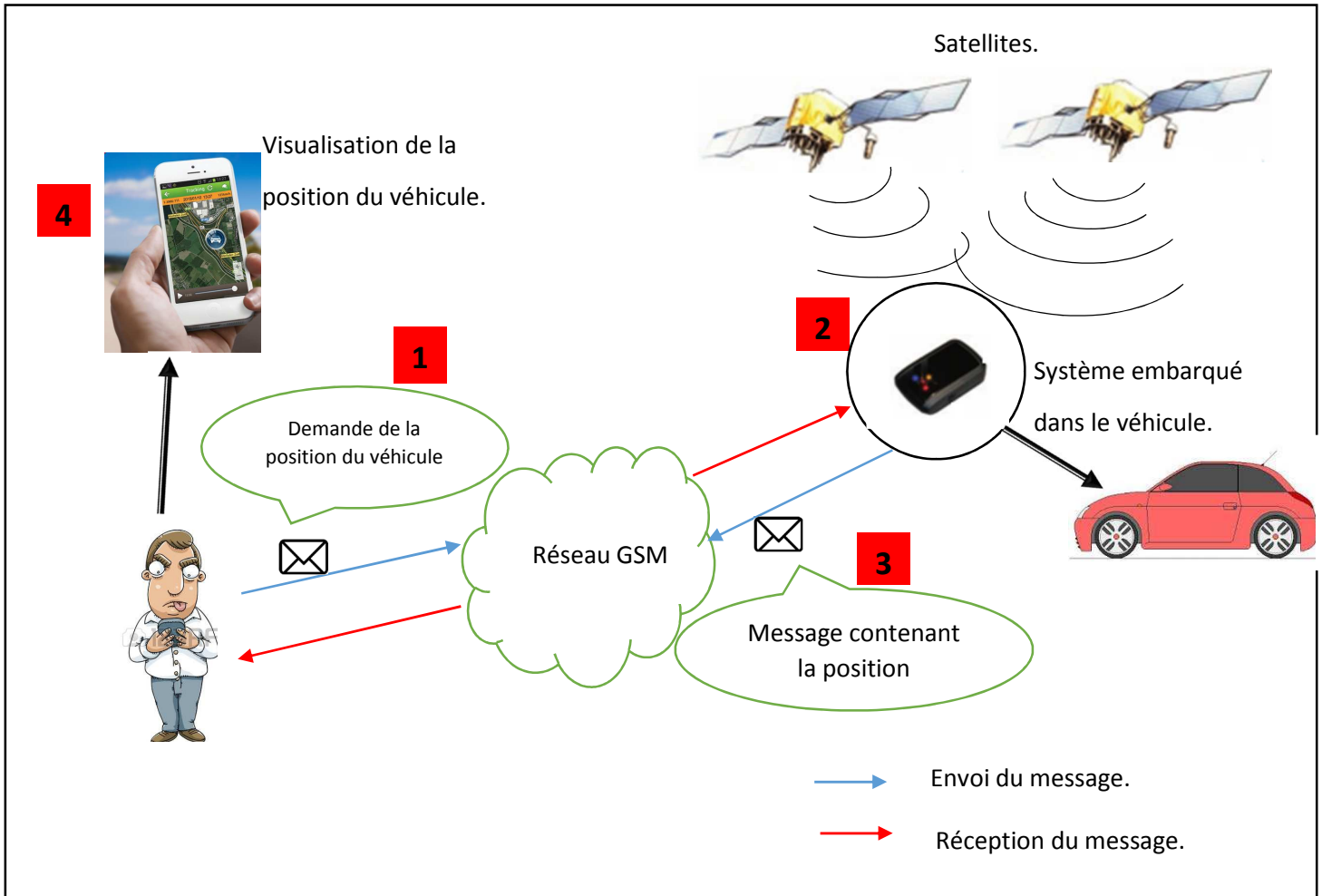


Fig. III.2. Fonctionnement du système de géolocalisation.

Ce dernier calcule sa position avec précision grâce aux signaux qu'il reçoit des satellites de la constellation du système de positionnement (2). Le système communique ensuite cette position sous forme de message (3), le message contient deux coordonnées : la latitude et la longitude, mais aussi un lien vers google maps pour afficher la position sur une carte. Une fois le message reçu, le propriétaire peut visualiser l'endroit où se trouve son véhicule, en utilisant un logiciel de cartographie (4).

En plus de la localisation de son véhicule, l'utilisateur du système peut procéder à la configuration du système. Il pourra par exemple changer le mot de passe en envoyant un message spécifique contenant l'ancien et le nouveau mot de passe. La figure III.3 donne le diagramme des cas d'utilisation de notre système.

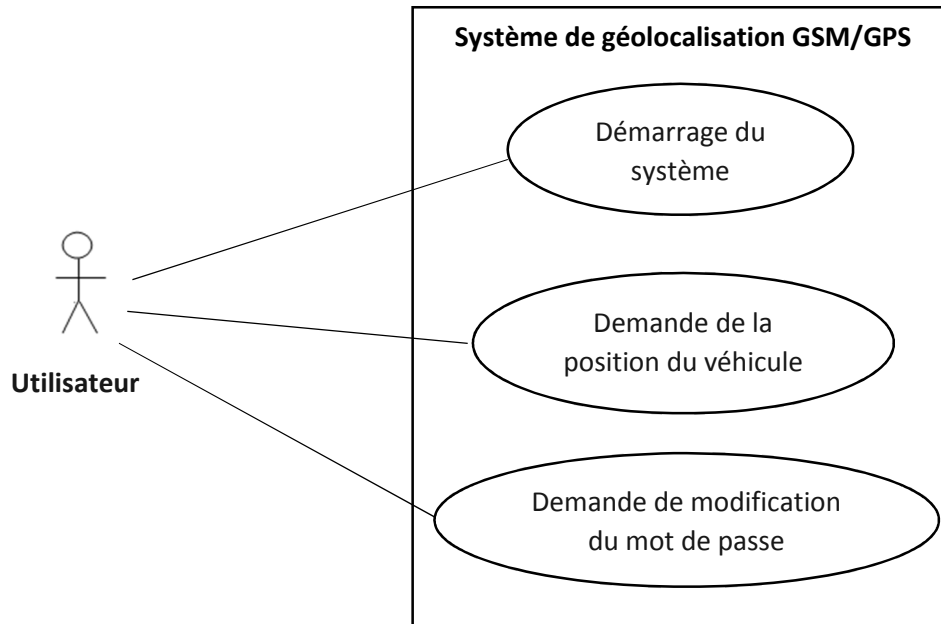


Fig. III.3. Diagramme des cas d'utilisation.

II.2. spécification des exigences fonctionnelles :

Lors de la demande de la position du véhicule, l'utilisateur doit être authentifié, pour cela, son message doit contenir un mot de passe. Pour modifier le mot de passe, le message doit contenir une commande spécifique. Le système logiciel doit alors remplir les fonctionnalités suivantes :

- Initialisation du système lors du démarrage.
- Attente d'un nouveau message.
- Traitement du nouveau message (lecture, extraction du contenu, suppression)
- Vérification du contenu du message.
- Récupération de la position actuelle, dans le cas de réception du mot de passe.

- Changement du mot de passe actuelle, dans le cas de réception de la commande de modification.
- Dans le cas de demande de position, envoi d'un message contenant la position (coordonnées géographiques et un lien vers google maps) vers le numéro ayant envoyé le message traité.
- Dans le cas de demande de modification de mot de passe, envoi d'un message de confirmation vers le numéro de téléphone ayant envoyé la demande.
- En cas d'erreur, informer l'utilisateur par message.

II.3. spécification des besoins matériels :

Afin de réaliser le système décrit précédemment, on doit utiliser un ensemble de composants matériels en interaction, chacun réalisant une tâche précise (Fig.III.4).

Pour pouvoir recevoir et envoyer les messages, le système doit disposer d'un module de communication radio, sur une longue distance, le choix se tourne alors vers un réseau mobile, mais celui-ci doit assurer la meilleure couverture possible, et être accessible par le grand public, d'où le choix d'utilisation d'un module GSM.

Le système doit pouvoir se localiser de manière précise, chose qui est permise par les systèmes de positionnement par satellite, le GPS est un système de géolocalisation accessible et entièrement opérationnel mais aussi le plus répandu, d'où le choix d'utilisation d'un récepteur GPS.

Un composant pour contrôler les modules cités, traiter les différentes informations (messages et trames) et assurer le bon fonctionnement du système en générale est nécessaire, d'où l'utilité d'utilisation d'un module de contrôle.

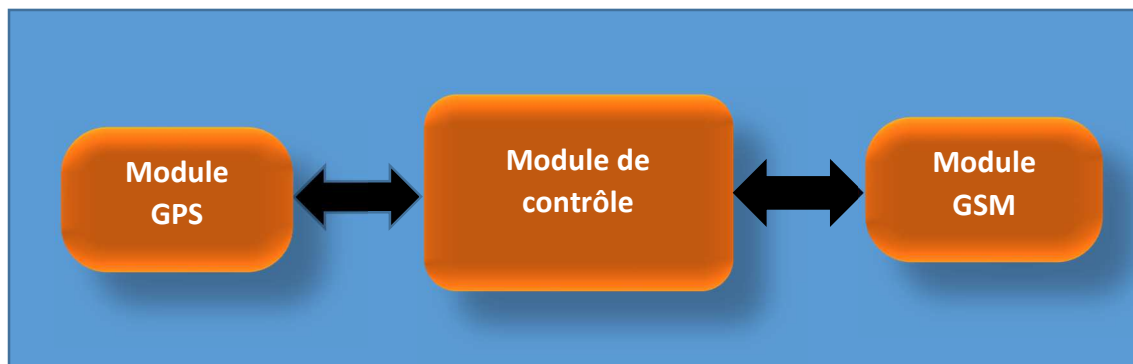


Fig.III.4. Vue simplifiée de l'architecture matérielle du système.

III. Conception :

Dans cette partie, nous allons choisir les composants matériels, les présenter en détail et voir leur schéma d'interconnexion. On définira le fonctionnement du système logiciel et les différents modules nécessaires à la réalisation des différentes tâches vues lors de l'analyse.

III.1. Conception matérielle :

Comme on l'a souligné au chapitre précédent, il existe des équipements intelligents dit de technologie M2M, qui intègrent, en même temps, un module GPS et un module GSM et qui présentent beaucoup d'avantages, notre choix se porte alors sur le module SIM908 qui en fait partie. Nous utilisons aussi un microcontrôleur de la famille Atmel, l'AT91SAM7S256. Celui-ci dispose d'une RAM suffisamment large (256Ko) pour éviter d'avoir recours à des mémoires externes, d'une bonne puissance de calcul et de différentes interfaces série (USART, I2C, USB, SPI). Les composants (le microcontrôleur et le sim908) disposant tous les deux d'interfaces série de type USART, la communication est ainsi assurée par cette interface (Fig.III.5).



Fig.III.5. Composants matériels utilisés.

Il faut noter que le microcontrôleur a accès aux différentes fonctions du SIM908 à l'aide des commandes AT (plus de détails sur les commandes AT sont donnés en Annexe D).

III.1.1. Description des composants matériels utilisés:

III.1.1.1. Description de la carte AT91SAM7S256 : [ATM1]

L'ATSAM7S256 (Fig.III.6) est un microcontrôleur produit par la société Atmel, basé sur un processeur ARM7 développé par la société ARM. Dans ce qui suit, nous allons voir les spécificités de ce composant.

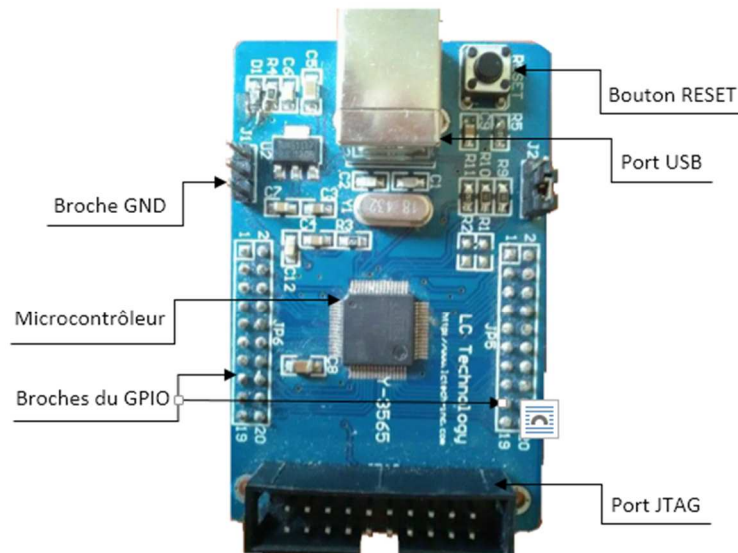


Fig.III.6. La carte AT91SAM7S256.

- **La mémoire :**

La carte dispose de

- Une RAM statique de 64 Ko (organisée en 16K mots de 32 bits) accessible en un seul cycle processeur.
- Une mémoire Flash de 256 Ko organisée en 1024 pages de 256 octets, accessible en lecture en un seul cycle et en écriture en deux cycles jusqu'à 30 MHz.
- Une ROM non mappée dans l'espace mémoire permettant le démarrage du système.

Le processeur a accès à ces ressources mémoire via le contrôleur de mémoire (MC).

- **Processeur ARM7TDMI : [ARM1]**

Le cœur ARM7TDMI est un processeur RISC 32 bits qui a fait naissance dans la version 4 de la famille ARM, il utilise l'architecture de Von Neuman, avec un pipeline à trois étages : Fetch (récupération de l'instruction), Decode (décodage de l'instruction et récupération des paramètres) et Execute (exécution de l'instruction et enregistrement du résultat).

➤ **Type d'instructions** : les instructions sont de longueur fixe et de structure uniforme afin d'accélérer le décodage, on trouve :

- Les instructions « load/store » : seules les instructions de chargement et rangement accèdent à la mémoire; toutes les autres instructions (traitements) travaillent sur les registres.
- Des instructions de multiplication longues, susceptibles de produire des résultats sur 64 bits.
- Des instructions spécifiques au traitement du signal. Ces instructions comprennent entre autres des opérations sur 16 bits ainsi que des opérations en arithmétique saturée.

➤ **Type de données** : Les types de données manipulables par le processeur sont les suivants :

- Word (mot) : Un mot est une donnée codée sur 32 bits, soit 4 octets. Il doit impérativement être aligné sur une adresse divisible par 4.
- Halfword (demi-mot) : est une donnée codée sur 16 bits, soit 2 octets. Il doit impérativement être aligné sur une adresse paire.
- Byte (octet) : est une donnée codée sur 8 bits. Il peut être rangé n'importe où en mémoire.

➤ **Les modes d'exécution** : Le processeur dispose de 7 modes de fonctionnement qu'on peut partager selon deux grandes catégories : Mode d'exécution normal (USR, SYS) et mode d'exceptions (IRQ, FIQ, SVC, ABT, UND) :

- **USR (USeR mode)** : mode utilisateur. Il s'agit du mode de fonctionnement des programmes utilisateurs. Certaines ressources sont inaccessibles (il s'agit du seul mode de fonctionnement non privilégié), et il est impossible de changer de mode autrement qu'en causant une interruption logicielle.
- **IRQ (Interrupt ReQuest mode)** : mode interruption qui est utilisé pour gérer les interruptions matérielles externes : il devient actif lorsqu'un signal actif est reçu sur l'entrée IRQ du processeur.

- *FIQ* (Fast Interrupt reQuest mode) : mode interruption rapide qui est actif lors de la réception d'un signal sur l'entrée FIQ du processeur. Ce mode se distingue du précédent par des capacités supérieures en termes de transfert de données, et un nombre de registres privés supérieur. La priorité du mode FIQ est supérieure à celle du mode IRQ.
- *SVC* (Supervisor mode) : mode Superviseur qui est un mode protégé, utilisé pour l'exécution du système d'exploitation.
- *ABT* (Abort mode) : signalé par le gestionnaire de la mémoire, utilisé pour implémenter des mécanismes de mémoire virtuelle et/ou de protection mémoire.
- *UND* (Undefined mode) : utilisé pour l'émulation logicielle de coprocesseur.
- *SYS* (System mode) : mode utilisé par le système d'exploitation pour effectuer des tâches nécessitant des privilèges étendus. Ce mode est donc similaire au mode superviseur, à ceci près que les registres sont ceux accessibles en mode utilisateur.

➤ **Les registres** : le processeur propose 31 registres généraux de 32 bits et six registres d'état qui cependant, seuls 16 registres généraux et deux registres d'état sont visibles à chaque instant ; les autres sont masqués selon le mode de fonctionnement du processeur et qui peuvent être déclenchés sur instruction ou sur évènement.

- **Les unités fonctionnelles de base :**

Les éléments de base nécessaires au bon fonctionnement du processeur sont :

- *Le PMC* (Power Management Controller) contrôle les différentes horloges du système :
 - MCK (Master Clock) alimente les périphériques qui fonctionnent en permanence (AIC et MC).
 - PCK (Processor Clock) alimente le processeur (arrêtée lorsque le processeur est en mode oisif «idle »).
 - UDP Clock (USB Device Port Clock) alimente le port USB.
 - Les horloges destinées aux périphériques (USART, SPI, etc.) peuvent être stoppées pour désactiver de manière sélective ces périphériques et ainsi économiser la consommation de l'énergie.
 - Des horloges en sorties, disponibles sur les broches PCK.

- *Le RC* (ou RSTC, Reset Controller) gère les lignes reset de la carte : le reset du processeur, le reset des périphériques.
- *Le PIT* (Periodic Interval Timer) destiné à gérer le temps partagé au niveau du système d'exploitation.
- *Le WDT* (Watchdog Timer) évite les verrouillages du système grâce à une interruption périodique programmable jusqu'à 16 secondes.
- *Le DBGU* (Debug Unit) gère tout l'aspect débogage de la carte.
- *Le PIO* (Parallel Input Output), chargé de gérer les entrées/sorties.
- *L'AIC* (Advanced Interrupt Controller), le contrôleur d'interruptions qui prend en charge la gestion de la priorité et de la vectorisation des interruptions pour le processeur.

- **Les USARTs :**

L'AT91SAM7S256 implémente deux USARTs (US0 et US1 que nous utilisons) pour la communication série avec les périphériques.

Schématiquement, les USARTs sont bâtis autour d'un générateur d'horloge (le baud rate generator) qui alimente les unités d'émission et de réception. Ils dialoguent avec l'AIC pour générer, des interruptions. Comme pour les autres périphériques internes, les broches des USARTs sont multiplexées avec les broches du PIO (ce dernier nécessite un paramétrage adéquat avant d'utiliser un de ces circuits). Enfin, l'unité de gestion d'alimentation (le PMC) permet d'activer ou non ces unités pour diminuer la consommation globale lorsque ces dernières ne sont pas utilisées.

III.1.1.2. Description du module SIM908 : [GPS/GSM2]

Le sim908 (Fig.III.7) est un produit de la société SimCom, il s'agit d'un module GPS/GSM quadri-bande travaillant sur les fréquences 850/900/1800/1900 MHz, il est équipé d'un récepteur GPS et une prise d'antenne est prévue pour chaque partie.

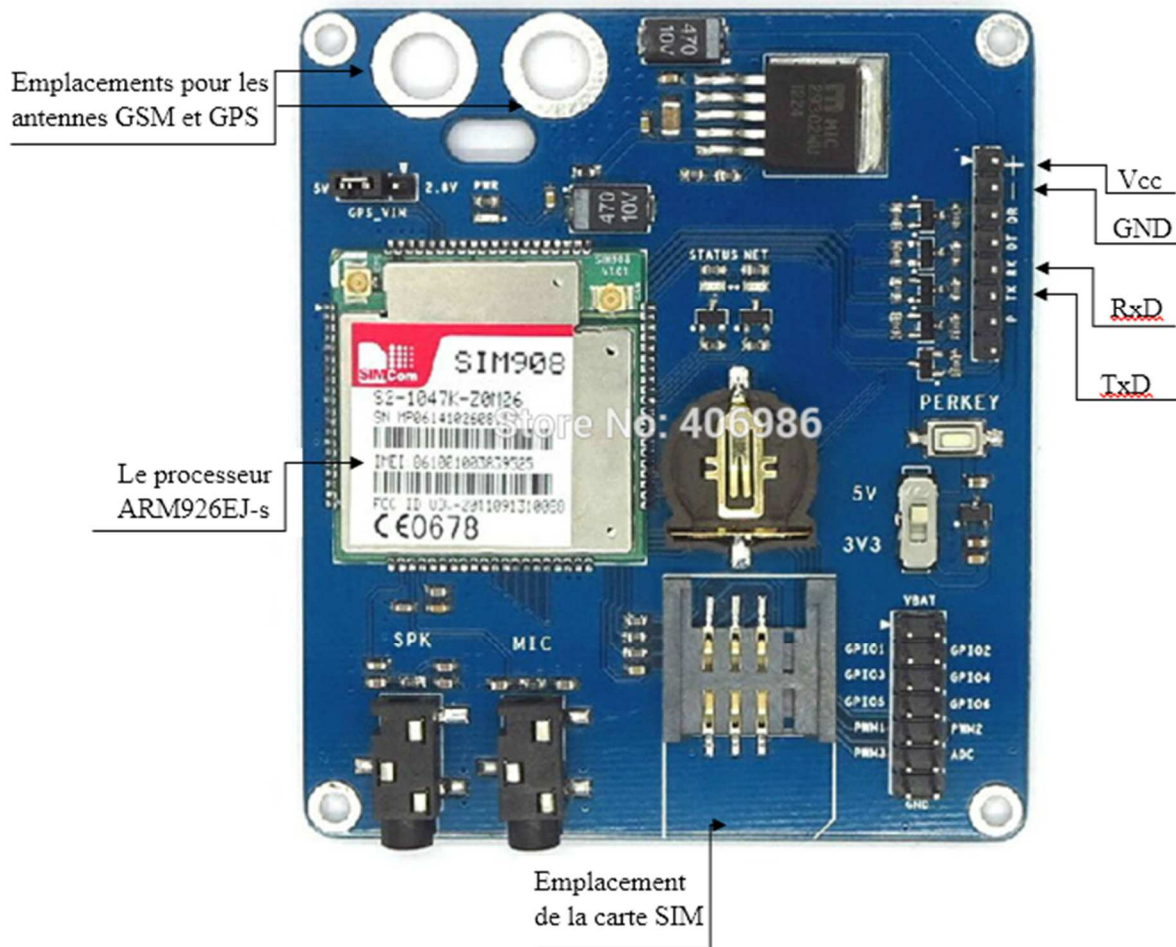


Fig.III.7. Le module SIM908.

La partie GSM permet la gestion des SMS, des communications de type « voix », et des communications de type « data » (transfert de données) et un accès direct à internet via le GPRS. Le module GSM est de classe 4 (2W) sur la bande 850/900 Mhz et de classe 1 (1W) sur la bande 1800/1900 Mhz.

Le récepteur GPS est doté de 42 canaux et sa sensibilité de suivi (tracking) lors d'un démarrage à chaud (hot start) est de -160 dBm pour l'acquisition d'une localisation avec une précision de seulement 2.5m et un temps d'une seconde. Pour un démarrage à froid (cold start), sa sensibilité est de -143 dBm et son temps d'acquisition est de 30 secondes.

Le module est géré par un processeur ARM926EJ-s basé sur l'architecture de version 5 contenant des instructions dédiées au traitement de signal. Il s'occupe de la communication téléphonique et des données (via le protocole TCP/IP) et dialogue avec la partie GSM. Le processeur gère également la carte SIM en 3V ou 1,8V qui est à l'extérieur du module.

Le module dispose de deux ports série TTL¹ : un principal (Tx²/Rx³) qui est utilisé pour transmettre les données d'un SMS et la transmission des données au cours d'une session TCP/IP en GPRS, mais permet aussi de recevoir des commandes du circuit (dans notre cas, celles provenant du module de control) qui peuvent être à la fois des commandes de type AT standards, commandes de type propriétaire AT étendues de SIMCOM. Celui du GPS (GPST_x/GPSR_x) est plutôt utilisé pour communiquer la trame NMEA. Le module fonctionne avec une tension allant de 3,4 à 4,5 Vcc et consomme un courant maximum de 0,4 A avec le GSM en mode de transmission et le GPS en fonction.

Le terme démarrage à froid décrit la performance d'un récepteur GPS à sa mise sous tension lorsqu'il n'y a aucune donnée de navigation disponible. Le démarrage à froid signifie que le récepteur n'a pas encore d'almanach à jour en mémoire, ni de données éphéméride de satellite, de position initiale, ni temps de référence. Le récepteur choisit automatiquement un ensemble de satellites et dédie un canal individuel afin de rechercher le signal de chaque satellite de la constellation de satellites visibles. Si aucun des satellites n'est acquis après une période prédéterminée (time out), le récepteur choisit un nouvel ensemble de recherche de satellites et ensuite il répète le processus jusqu'à ce que les données d'un premier satellite soient acquises. Un almanach complet du système n'est pas exigé pour réaliser un premier positionnement.

III.1.2. L'UART :

Comme dans toute transmission en série, l'information à envoyer est décomposée en bits, qui sont ensuite transmis un à un sur la ligne de transmission. L'UART est une interface classique de communication série en mode asynchrone, c.à.d. que l'émetteur et le récepteur disposent chacun de leurs propres horloges qui doivent être préalablement réglées à la même cadence.

La liaison série la plus simple utilise 3 fils : Un fil pour la transmission (Tx_D), un fil pour la réception (Rx_D) et un fil de masse (GND) (Fig.III.8). L'UART reçoit du processeur une donnée parallèle sur 8 bits, la sérialise et l'envoie bit par bit sur la ligne Tx_D, la réception se fait au niveau de la ligne Rx_D, l'ensemble de bits reçu en série est reconstitué en une donnée sur 8 bits et fournit au processeur.

¹ TTL : Transistor-Transistor Logic.

² Tx_D (Transmit Data) : broche de transmission de données.

³ Rx_D (Receive Data) : broche de réception de données.

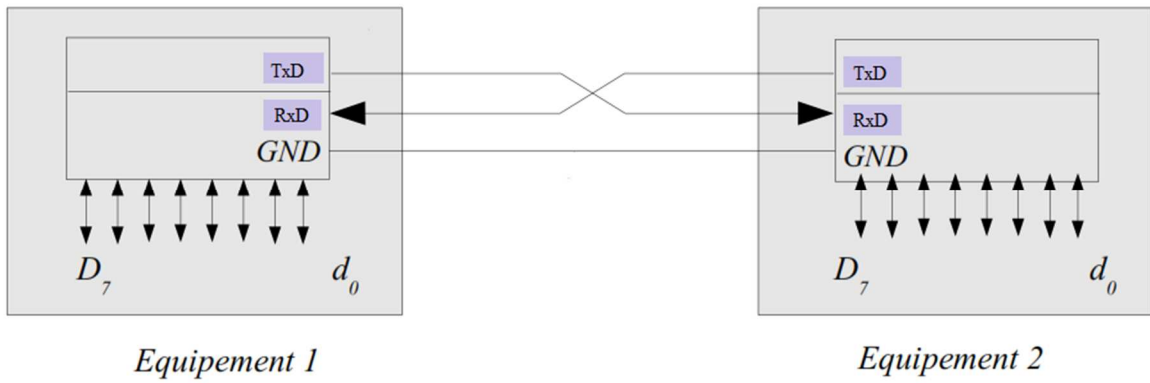


Fig.III.8. Schéma de connexion entre deux équipements à l'aide d'une liaison UART.

Après avoir vu en détail les différents composants, on présente sur la figure III.9, le schéma synoptique de notre système qui montre l'interconnexion des différents composants cités plus haut, et l'alimentation qui va servir à faire fonctionner ce système.

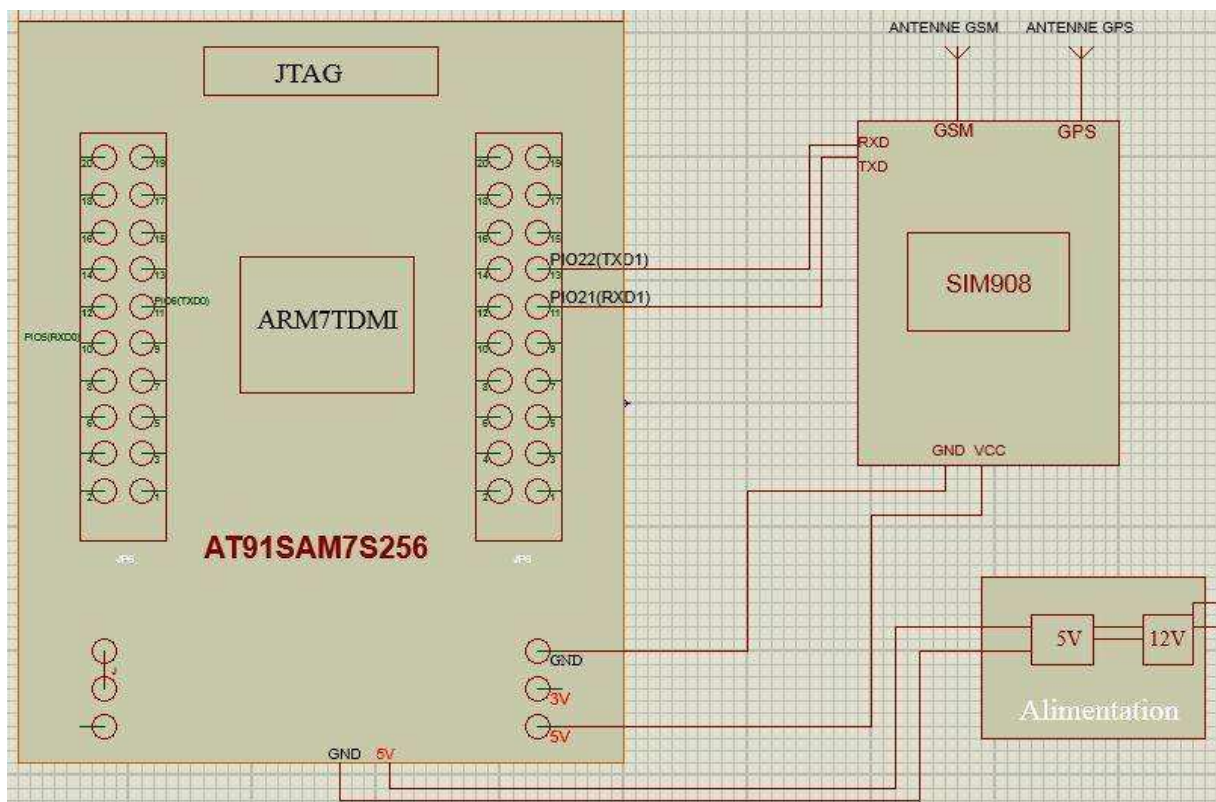


Fig.III.9. Schéma synoptique du système.

III.2. Conception Logicielle :

On peut résumer le fonctionnement global du système dans le schéma de la figure III.10.

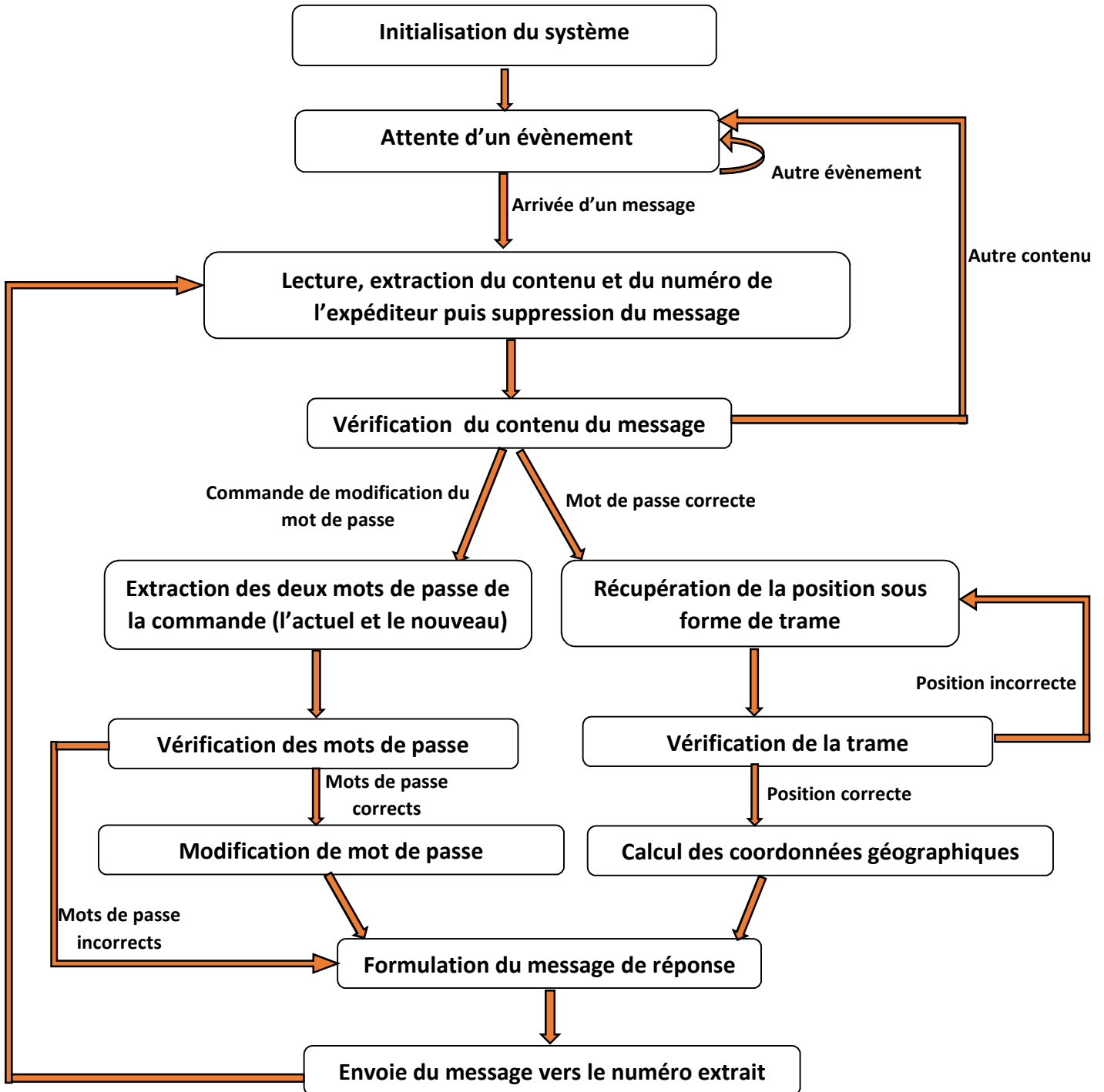


Fig.III.10. Schéma de fonctionnement global du système.

Afin de mieux structurer notre système, nous allons procéder à son abstraction en regroupant les tâches et procédures communes à chaque unité dans un seul module. Ainsi, nous allons utiliser les modules suivants :

- **Initialisation** : L'utilisation de composants électroniques nécessite une configuration initiale d'où la définition de ce module. Il va donc regrouper les procédures permettant la définition des paramètres initiaux.
- **Message** : Ce module va englober toutes les procédures traitant les messages.
- **Position** : Ce module contiendra toutes les procédures permettant de recevoir et traiter la trame NMEA depuis le SIM908.
- **MotDePasse** : Ce module contiendra les procédures qui traitent la demande de modification du mot de passe.
- **Communication** : Ce module disposera des différentes procédures permettant la communication entre le microcontrôleur et le module GPS/GSM via la liaison série.
- **Principal** : Ce module permettra de faire le lien entre les autres composants en étant le point de lancement et de terminaison de l'application, il s'occupera aussi de la détection des événements arrivant et de leur test.

III.2.1. Comportement du système logiciel :

Les diagrammes suivants vont nous permettre de voir avec précision l'interaction entre les différents modules du système logiciel dans les différents cas d'utilisation définis précédemment.

III.2.1.1. Le cas d'utilisation « démarrage du système » :

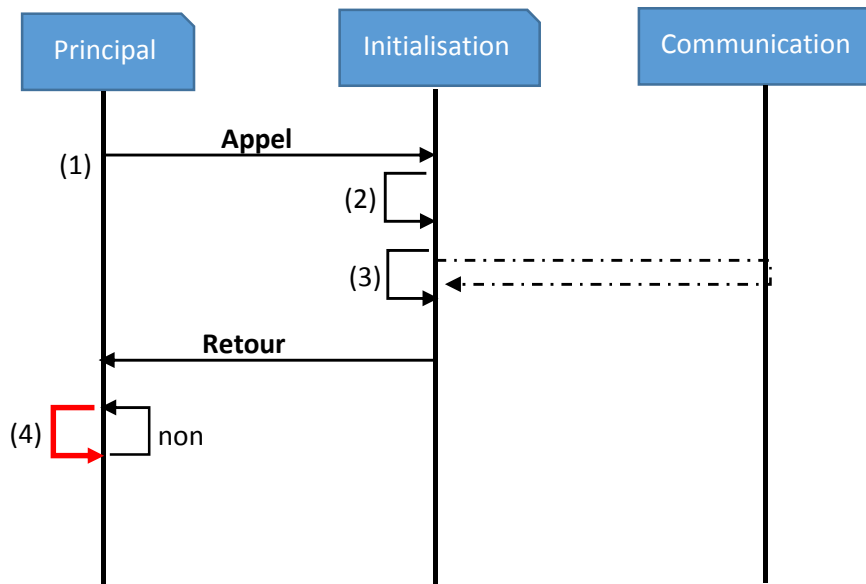


Fig.III.11. Diagramme de séquence pour le cas « Démarrage du système »

Description:

- (1) Lors de la mise en marche du système, le module « Principal » fait appel au module « Initialisation ».
- (2) Dans le module « Initialisation », l'appel est fait à la procédure qui initialise la liaison série.
- (3) Le même module fait appel à la procédure d'initialisation du module SIM908.
- (4) Le module « Principal » boucle pour attendre un évènement arrivant sur la liaison série, dans la boucle, un test est effectué pour savoir si l'évènement correspond à l'arrivée d'un message. Si c'est le cas, alors on fait appel au module « Message », sinon on continue à scruter le port série.

Plusieurs procédures des différents modules feront appel au module « Communication » afin d'envoyer des commandes vers le module SIM908 et recevoir leur résultat.

III.2.1.2. Le cas d'utilisation « Demande de la position du véhicule » :

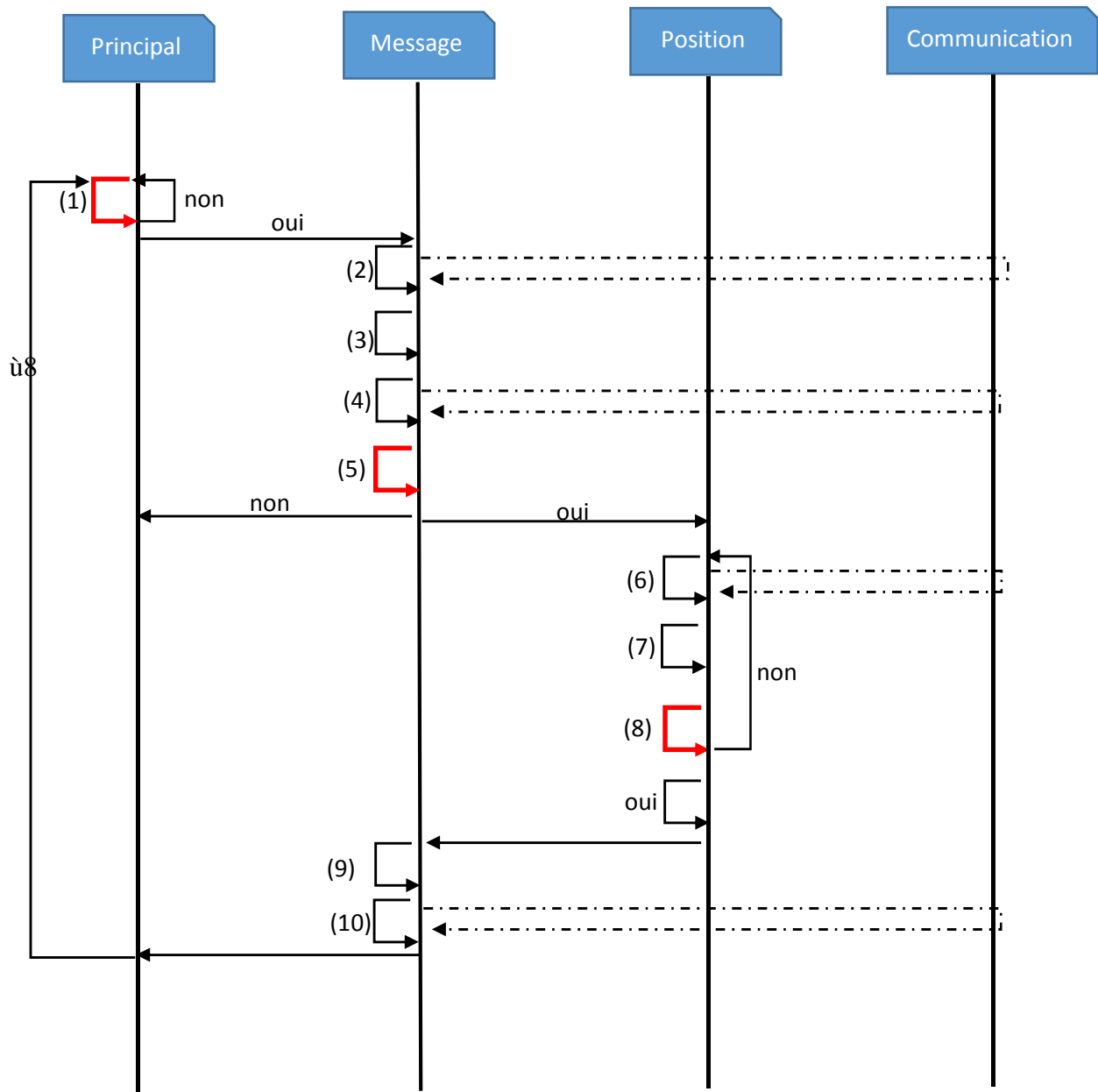


Fig.III.12. Diagramme de séquence pour le cas « Demande de la position du véhicule».

Description :

- (1) Le module « Principal » scrute l'arrivée d'un nouvel évènement, dès qu'il s'agit d'un message il fait appel au module « Message ».
- (2) Suite à l'arrivée d'un message, le module « Message » fait appel à la procédure permettant de lire le message à partir de la carte SIM et de le sauvegarder en mémoire.
- (3) Une fois le message sauvegardé, on fait appel à une procédure pour en extraire le contenu et le numéro de téléphone de l'émetteur et les sauvegarde en mémoire.
- (4) Afin de ne pas encombrer la mémoire de la carte SIM, le module « Message » supprime le message juste après qu'il soit sauvegardé.
- (5) On récupère le contenu reçu et on le compare au mot de passe prédéfini, s'il y a correspondance, on fait appel au module « Position ». sinon, on revient à la scrutation.
- (6) Le module « Position », récupère la trame NMEA et l'enregistre en mémoire.
- (7) On extrait la longitude et la latitude de la trame et on les sauvegarde en mémoire.
- (8) La procédure de test, compare les valeurs des coordonnées récupérées à la valeur nulle, dans le cas où les valeurs sont incorrectes une autre demande de trame est effectuée, sinon le module construit les coordonnées finales.
- (9) Retour au module « Message » où on construit le message en réponse à l'utilisateur.
- (10) On envoie ensuite le message construit vers le numéro de téléphone sauvegardé.

III.2.1.3. le cas d'utilisation « Demande de modification du mot de passe » :

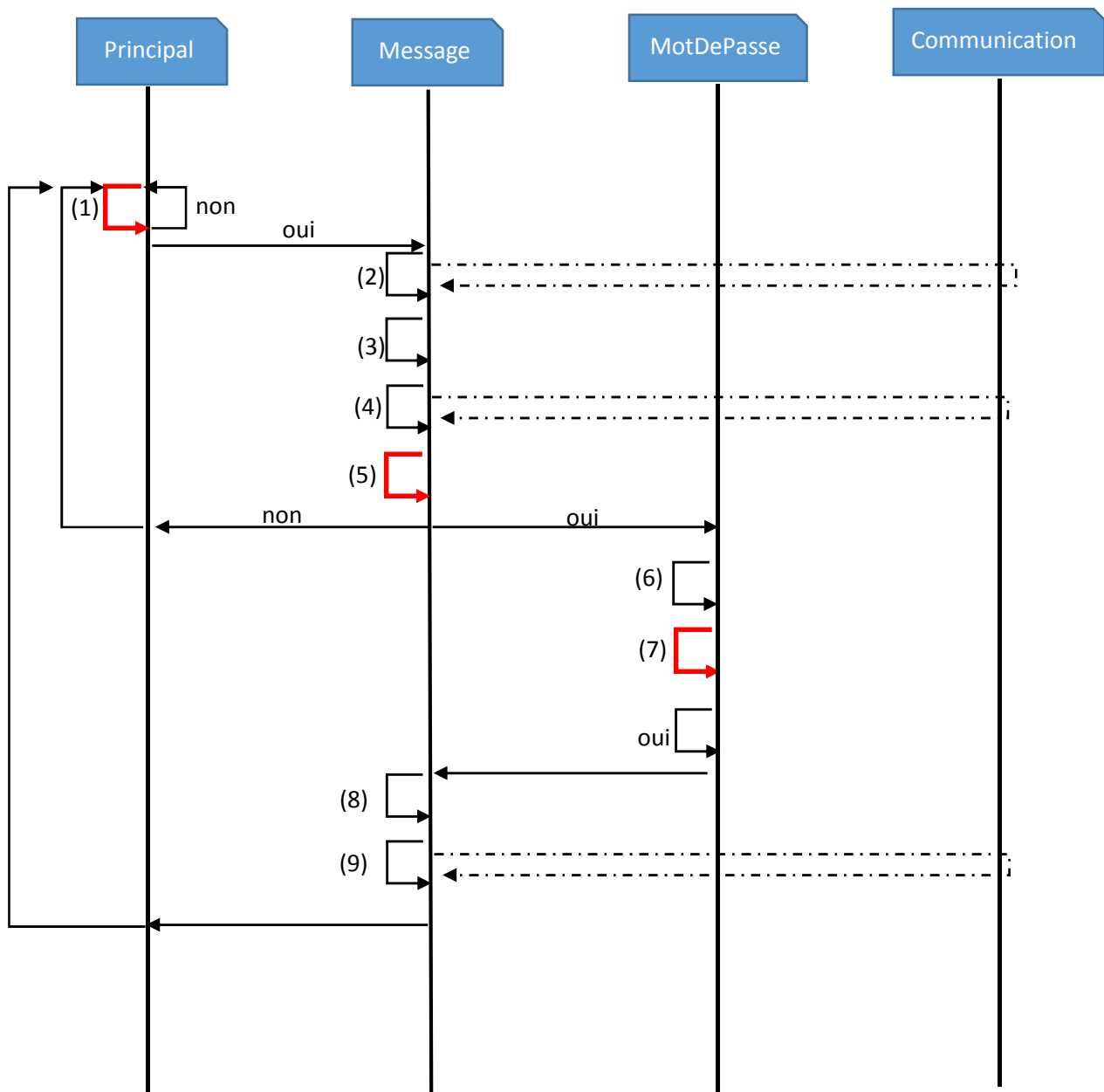


Fig.III.13. Diagramme de séquence pour le cas « Demande de modification du mot de passe ».

Description:

- (1) (2) (3) (4) sont les mêmes étapes que pour le cas « demande de la position du véhicule ».
 (5) On récupère le contenu reçu et on vérifie si c'est une commande de changement de mot de passe, si c'est le cas, on fait appel au module « MotDePasse». Sinon, on revient à la scrutation.

- (6) Au niveau du module « MotDePasse », on extrait les mots de passe (l'ancien et le nouveau) de la commande et on les sauvegarde en mémoire.
- (7) une procédure de test vérifie l'exactitude de l'ancien mot de passe, s'il est correct, alors on le modifie en le remplaçant par le nouveau, Sinon on revient au module « Message » pour envoyer un message d'erreur.
- (8) Après modification du mot de passe, retour au module « Message » où on construit le message de confirmation en réponse à l'utilisateur.
- (9) On envoie ensuite le message construit vers le numéro de téléphone sauvegardé.

III.2.2. Structure du système logiciel :

On peut représenter la structure modulaire du système logiciel par le diagramme de classes de la Figure III.14.

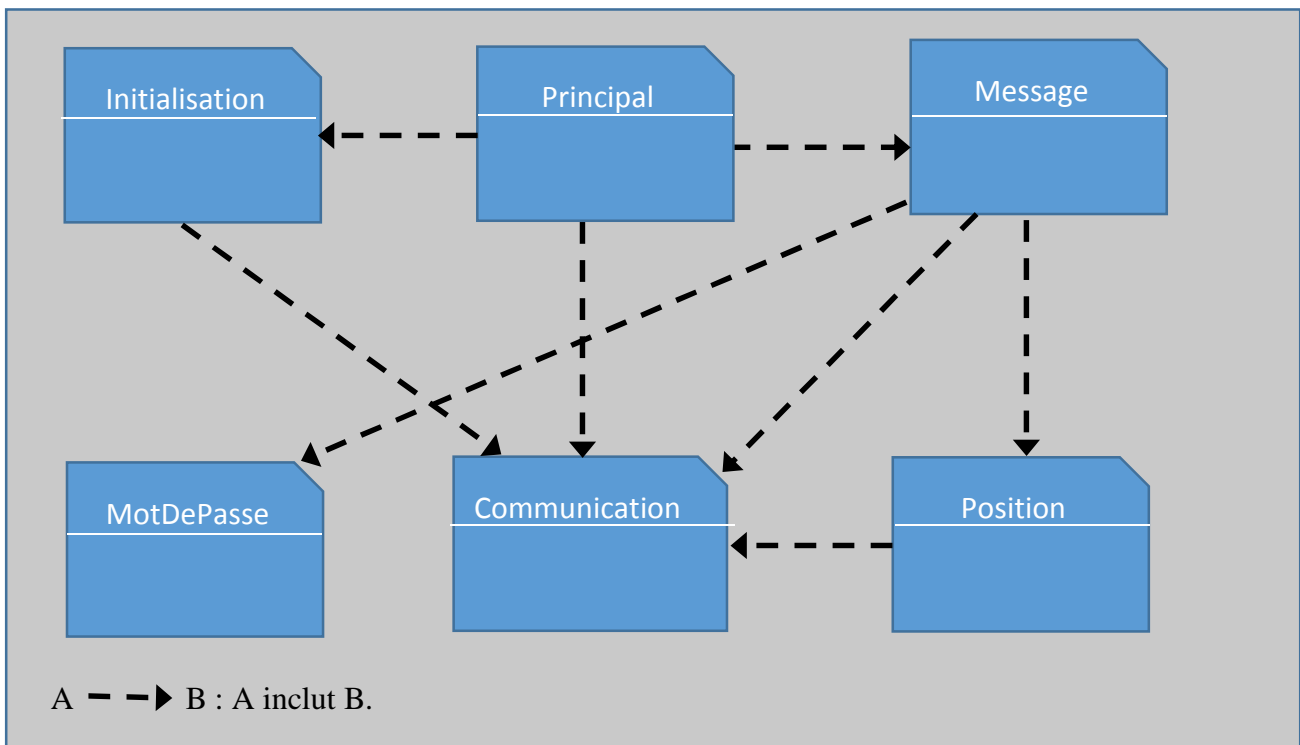


Fig.III.14. Diagramme de classes.

On présente dans ce qui suit, quelques procédures utilisées par les modules.

- **Initialisation:**

- **Init_Liaison_Serie** : elle permet d'initialiser la communication à travers le port série avec les paramètres suivants : 8 bits de données, 1 bit stop, vitesse de transmission à 57600 bps, pas de control de parité, émission des bits de poids faible d'abord et pas de control de flux.

• **Init_SIM908** : elle permet d'initialiser le composant SIM908, comme suit :

- Synchronisation de la vitesse de communication avec le module ;
- communication à l'aide des messages sous format texte ;
- définir le numéro du centre de messagerie ;
- réinitialiser le module GPS à froid.

- **Message :**

- **Traitement_Message** : cette procédure récupère le message reçu de la mémoire de la carte SIM du SIM908, en extrait le contenu et le numéro de téléphone qui l'a envoyé. Les données récupérées sont sauvegardées en mémoire.
- **Test_Message** : cette procédure permet de récupérer le mot de passe reçu en message et le comparer au mot de passe prédéfini et renvoie le résultat du test dans un registre.
- **Suppression_Message** : cette procédure permet de supprimer un message de la carte SIM du SIM908.
- **Formulation_Message** : dépendamment du paramètre qu'elle reçoit, cette procédure permet soit de récupérer les coordonnées (longitude et latitude), de former le lien vers google maps et de les intégrer dans le message à envoyer à l'utilisateur. Soit de formuler un message d'erreur.
- **Envoi_Message** : cette procédure permet d'envoyer un message via le module GSM du SIM908.

- **Position :**
 - **Demande_Trame :** cette procédure permet d'interroger le module SIM908 sur sa position et celui-ci lui répond en lui envoyant une trame NMEA que la procédure sauvegarde en mémoire.
 - **Traitement_Trame :** cette procédure permet de récupérer la longitude et la latitude depuis la trame reçue et de les sauvegarder en mémoire.
 - **Formulation_N_E :** cette procédure permet de former les coordonnées à envoyer à partir de celles reçues du SIM908.

- **MotDePasse:**
 - **Extraction_Passe :** permet d'extraire les mots de passe (l'ancien et le nouveau) de la commande reçue et de les sauvegarder en mémoire.
 - **Test_Passe :** permet de tester la validité de l'ancien mot de passe.
 - **Modification_Passe :** permet de modifier l'ancien mot de passe par le nouveau.

- **Communication :**
 - **Envoi_Chaine :** procédure qui permet d'envoyer, sur le port série, une chaîne de caractères préalablement enregistrée en mémoire, dans notre cas, on l'utilise pour l'envoi des commandes AT vers le SIM908.
 - **Reception_Chaine :** procédure qui permet de récupérer et de sauvegarder en mémoire, les différents caractères d'une chaîne reçue sur le port série.

III.2.3. Algorithmes de quelques procédures:

- **La procédure Traitement_Trame :**

Sachant que les données de la trame NMEA nécessaires à notre système sont envoyées par le SIM908 sous la forme suivante :

0,357.469332,3643.701357,
 (1) (2) (3)

- (1) Type de la trame
- (2) La longitude
- (3) La latitude

L'objectif de la procédure est de récupérer la longitude et la latitude de la trame, pour l'exemple précédent, on aura au final deux chaînes de caractères comme suit :

Nbrute : contiendra la latitude brute envoyée par le sim908 (357.469332).

Ebrute : contiendra la longitude brute envoyée par le sim908 (3643.701357).

Procédure Traitement_Trame

Trame : *chaîne de caractères* // contient la trame reçue du SIM908

Ebrute, Nbrute : *chaînes de caractères.*

i : entier initialisé à 0

c : caractère

debut

Tantque (trame[i] <> ',') faire

i = i +1 ;

fait

```

Tantque (trame[i] <> ',') faire
    Ebrute= Ebrute + trame [i];
    i = i +1 ;
fait
i = i +1 ;
Tantque (trame[i] <> ',') faire
    Nbrute= Nbrute + trame [i];
    i = i +1 ;
fait
fin

```

- **La procédure formulation_N_E**

La procédure sert à calculer la longitude et la latitude sous la forme suivante : dd° mm' s''
 Et cela à partir des coordonnées reçues (Ebrute et Nbrute), pour l'exemple précédent, on a la longitude sous la forme suivante :

$$\begin{array}{c} 3 \quad 57.469332 \\ \underbrace{\quad} \quad \underbrace{\quad} \quad \underbrace{\quad} \\ (1) \quad (2) \quad (3) \end{array}$$

(1) La position en degré (3°)

(2) Les minutes (57')

(3) La partie fractionnelle des minutes qu'on multiplie par 60 pour obtenir les seconde (0.469332*60=28,15992'')

Donc la longitude de la position finale est : 3°57'28,15992'' E

Procédure formulation_N_E

Nbrute : *chaîne de caractères* //contient la latitude brute envoyée par le sim908

Ebrute : *chaîne de caractères* //contient la longitude brute envoyée par le sim908

E : *chaîne de caractères* //nouvelle longitude après les calculs (à envoyer)

N : *chaîne de caractères* //nouvelle latitude après les calculs (à envoyer)

P, S, secondeC: *chaînes de caractères*

K, i, j, secondeE, m : *entiers* initialisés à 0

d : *caractère*

```

debut
m <= 2
p <= Ebrute
tantque ( m <> 0)
    tantque (P[k] <> '.') faire
        k <= k+1
    fait
    pour (i <= 0 jusqu'a (k-2))
        S[i] <= P[i]
    finpour
    d <= ' ° '
    S[i] <= d
    i = i + 1
    S[i] <= P[i]
    i = i + 1
    S[i] <= P[i]
    i = i + 1
    d <= ' ' '
    S[i] <= d
    j <= i
    k <= 0
    pour (j <= i jusqu'a longueur (P)) faire
        secondeC[k] <= P[j]
        k = k + 1
    fait
    secondeE <= int de (secondeC)
    secondeE = secondeE/ 60
    secondeC <= chaine de (secondeE)

```

```

d = ' '
S<= S + secondeC+d
E <= S
P <= Nbrute
m <= m - 1

```

fait

N <= S

Fin

- **La procédure Envoi_Chaine:**

Procédure Envoi_Chaine (E : chaîne de caractères)

// E contient la chaîne à envoyer sur le port série.

A : caractère

i : entier

début

pour (i <= 0 jusqu'à longueur(E)) faire

A <= E [i]

Envoyer A sur la TxD de l'usart

fait

fin

- **La procédure Envoi_Message:**

Procédure Envoi_Message

commandeAT : *chaîne de caractères* // la commande permettant l'envoi d'un message sur le SIM908.

NumTel : *chaîne de caractères* // le numéro de téléphone vers lequel on envoie le message.

Message : *chaîne de caractères* // le contenu du message.

Debut

commandeAT <= " AT+CMGS=" + **NumTel**

Envoi_Chaine(CommandeAt)

Reception

Fin

- **La procédure Modification_Passe:**

Procédure Modification_Passe

nouveau_passe : *chaîne de caractères* // contient le nouveau mot de passe

ancien_passe : *chaîne de caractères* // contient l'ancien mot de passe

début

pour ($i \leq 0$ jusqu'à *longueur (nouveau_passe)*) **faire**

ancien_passe[i] <= **nouveau_passe**[i]

fait

$i <= i + 1$

nouveau_passe[i] <= 0

fin

Conclusion :

Dans ce chapitre, nous avons modélisé les deux parties de notre système. On a choisi les composants matériels nécessaires et déterminé les différents modules logiciels qui vont assurer le bon fonctionnement du système. Suite au travail accompli au niveau de cette partie, il reste les étapes d'implémentation et des tests qui vont faire l'objet du prochain chapitre.

Chapitre 4

***Implémentation et tests du
système.***

Introduction :

Une fois le travail d'analyse et de conception achevé, vient l'étape du codage, qui consiste en la traduction dans un langage de programmation des différentes fonctionnalités précédemment définies. Et afin d'avoir en résultat un système qui réalise correctement toutes les tâches attendues, il faut bien choisir les logiciels ainsi que le langage adapté pour l'implémentation.

Dans ce chapitre, nous allons présenter les différents outils de développement utilisés, ainsi que les figures présentant les résultats des tests effectués sur notre système.

I. Outils de développement utilisés :

I.1. Outils logiciels :

I.1.1. Keil - μ vision :

Keil - μ vision (Fig.IV.1) est un environnement de développement pour systèmes embarqués. La plateforme fournit un ensemble d'outils tels que:

- Un éditeur de code (assembleur, C, C++) avec coloration syntaxique ;
- Un compilateur croisé pour architectures ARM7, ARM9, Cortex-M3, C166, 8051;
- Un simulateur supportant plusieurs processeurs et cartes SOC ;
- Un débogueur intégré fournissant l'état du processeur et des périphériques des cartes supportées en mode simulation;
- Un ensemble de fichiers de démarrage (startup) pour accélérer le développement des applications ;
- Des projets exemples complets pour des fins d'apprentissage.

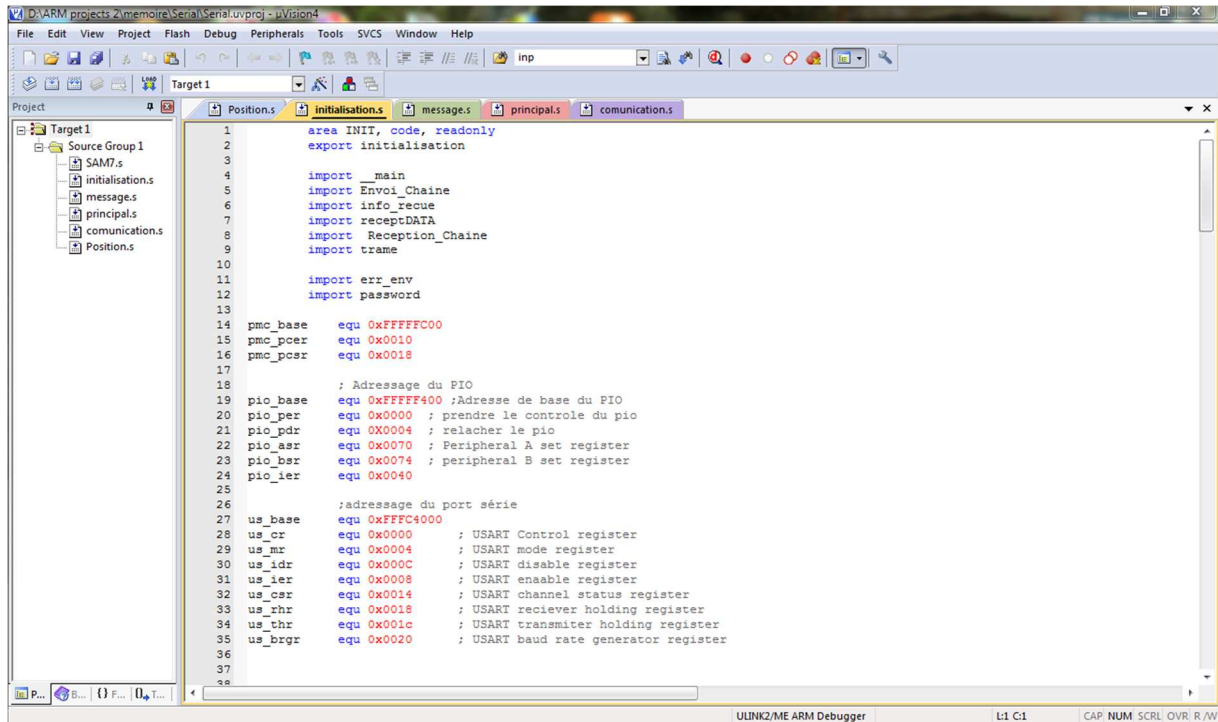


Fig.IV.1. La plateforme Keil- µ vision.

I.1.2. Putty :

Putty est un logiciel libre écrit et maintenu principalement par Simon Tatham. À l'origine disponible uniquement pour Windows, il est à présent porté sur diverses plates-formes Unix. Il s'agit d'un émulateur de terminal doublé d'un client pour des protocoles tels que SSH¹, Telnet². Tous ces protocoles servent à lancer une session sur une machine distante, via le réseau. Putty constitue la partie client de cette session, autrement dit l'extrémité de la liaison où les commandes sont tapées, et où les résultats sont affichés, par opposition à celle où sont exécutés les traitements. Il permet également d'établir des connexions directes par liaison série (Fig. IV.2), Dans ce mode de fonctionnement, après avoir choisi les paramètres de la communication, le texte tapé au clavier dans la fenêtre du logiciel est envoyé tel quel par le port série et les données reçues par ce port sont affichées telles quelles dans la même fenêtre.

¹ SSH (Secure Shell) est un protocole qui offre un service de connexion sécurisée à distance.

² TELNET est l'un des protocoles les plus anciens de l'environnement TCP/IP. Il facilite la connexion à distance.

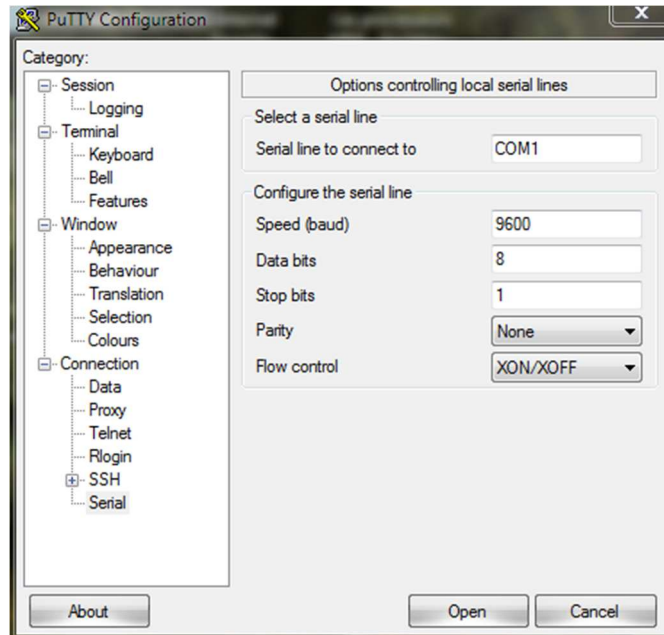


Fig. IV.2. La fenêtre d'établissement d'une connexion série du logiciel Putty.

I.2. Langage de programmation (Assembleur ARM) :

Le langage machine, est la suite de bits qui est interprétée par le processeur d'un ordinateur. Chaque type de processeur comprend son propre langage machine, chaque instruction dans ce langage a son propre code numérique unique appelé code opération. Un programme en langage assembleur est stocké sous forme de texte (comme un programme dans un langage de plus haut niveau) et chaque instruction assembleur représente exactement une instruction machine. Par exemple l'instruction d'addition est représentée comme suit :

Add r1 r2, r3

Qui signifie : additionner le contenu de r2 avec celui de r3 et sauvegarder le résultat dans r1.

Le mot Add est une mnémotique pour l'instruction d'addition. La forme générale d'une instruction assembleur est : ***Mnémotique opérande1 [,opérande2,...].***

ARM est le langage assembleur propre aux processeurs de la famille ARM, son jeu d'instruction est composé de 5 types d'instruction :

- Instructions de traitement de données ;
- Instructions d'accès à la mémoire ;
- Instructions de contrôle de séquences (branchement) ;
- Instructions de gestion d'interruption logicielle ;
- Instruction manipulant le registre d'état.

I.3. Outils matériels :

- **L'AT91SAM7S256** : le microcontrôleur qui a servi comme unité de contrôle du système.
- **Le SIM908** : le module intégrant un module GPS et un module GSM.
- **Le JTAG** : la sonde ULINK2 qui nous a permis d'injecter le code dans l'AT91SAM7S256 et le débogage de notre système.
- **Le multimètre digital** : Le multimètre digital UNI-T modèle UT60D a été utilisé pour mesurer et vérifier nos circuits.
- **Le lab d'essai** : Le lab d'essai nous a permis de réaliser et de tester l'interfaçage de nos modules au niveau matériel.
- **L'alimentation stabilisée et régulée** : L'alimentation stabilisée et régulée a été utilisée pour répondre aux besoins des diverses tensions (5V).

II. Les tests :

II.1. Les tests unitaires :

Dans ce qui suit, nous allons présenter les résultats des tests effectués sur les composants matériels utilisés.

II.1.1. Test de l'AT91SAM7S256 :

Nous avons procédé à l'essai du port série asynchrone (USART) de la carte, et cela en exécutant un programme qui permet soit :

- D'écrire un message sur l'émulateur Putty, de l'envoyer via le port. Le message est sauvegardé en mémoire pour qu'il soit renvoyé ensuite via le même port, on peut alors le visualiser sur la fenêtre de l'émulateur.
- De faire la somme de deux chiffres, et cela en écrivant la commande suivante :

$$= op1 + op2$$

La carte reçoit la commande via l'usart, la teste, effectue le calcul puis renvoie le résultat via le même port pour qu'on puisse visualiser le résultat sur l'émulateur.

On peut voir le branchement des différents outils sur la figure IV.3.

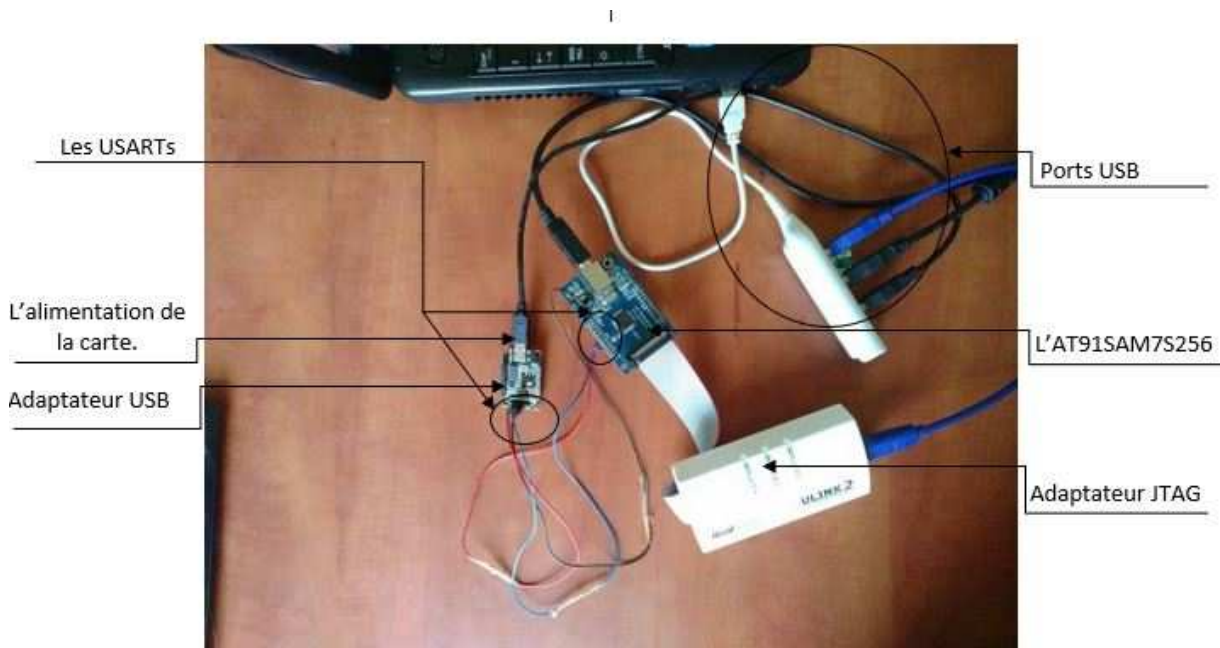
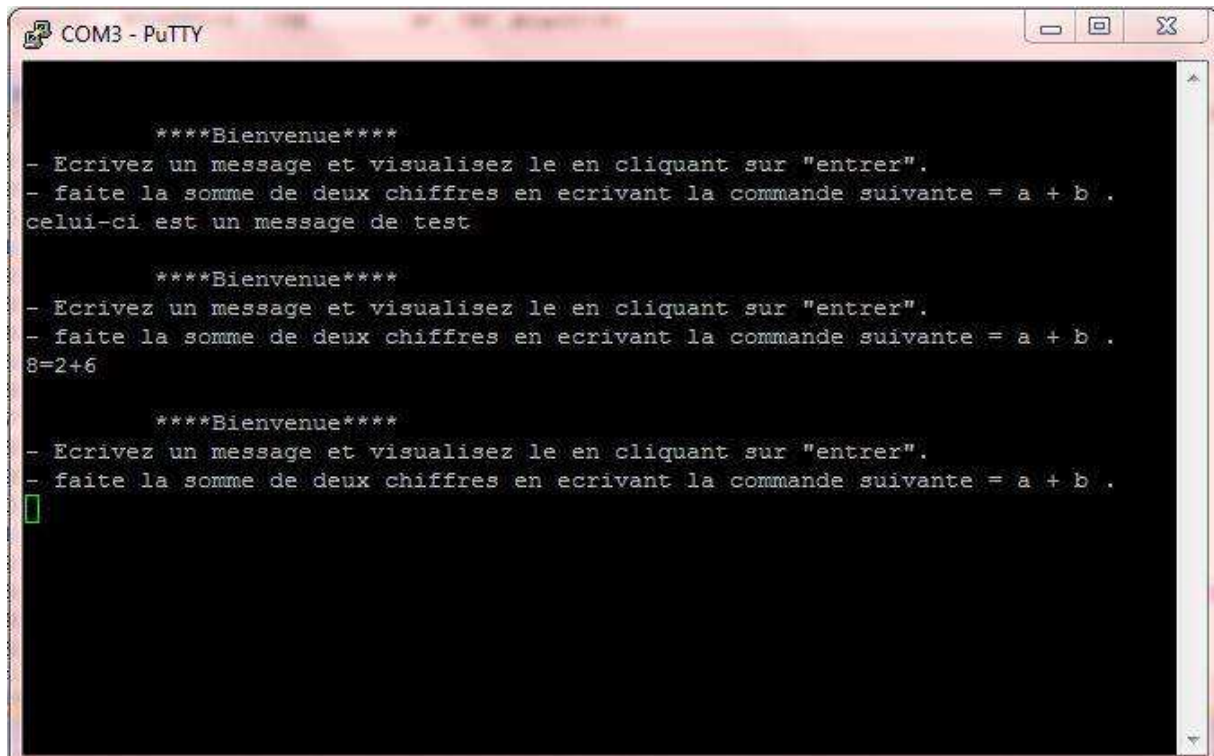


Fig. IV.3. Test de l'AT91SAM7S256.

On peut voir les résultats du test sur la figure IV.4.



```
****Bienvenue****
- Ecrivez un message et visualisez le en cliquant sur "entrer".
- faite la somme de deux chiffres en ecrivant la commande suivante = a + b .
celui-ci est un message de test

****Bienvenue****
- Ecrivez un message et visualisez le en cliquant sur "entrer".
- faite la somme de deux chiffres en ecrivant la commande suivante = a + b .
8=2+6

****Bienvenue****
- Ecrivez un message et visualisez le en cliquant sur "entrer".
- faite la somme de deux chiffres en ecrivant la commande suivante = a + b .
█
```

Fig.IV.4. Résultats du test de l'AT91SAM7S256.

II.1.2. Test du module SIM908 :

Afin de tester le module, nous avons procédé à l'envoi des commandes AT directement à partir de l'émulateur Putty et via la liaison série (USART).

On peut voir le branchement des différents outils sur la figure IV.5.

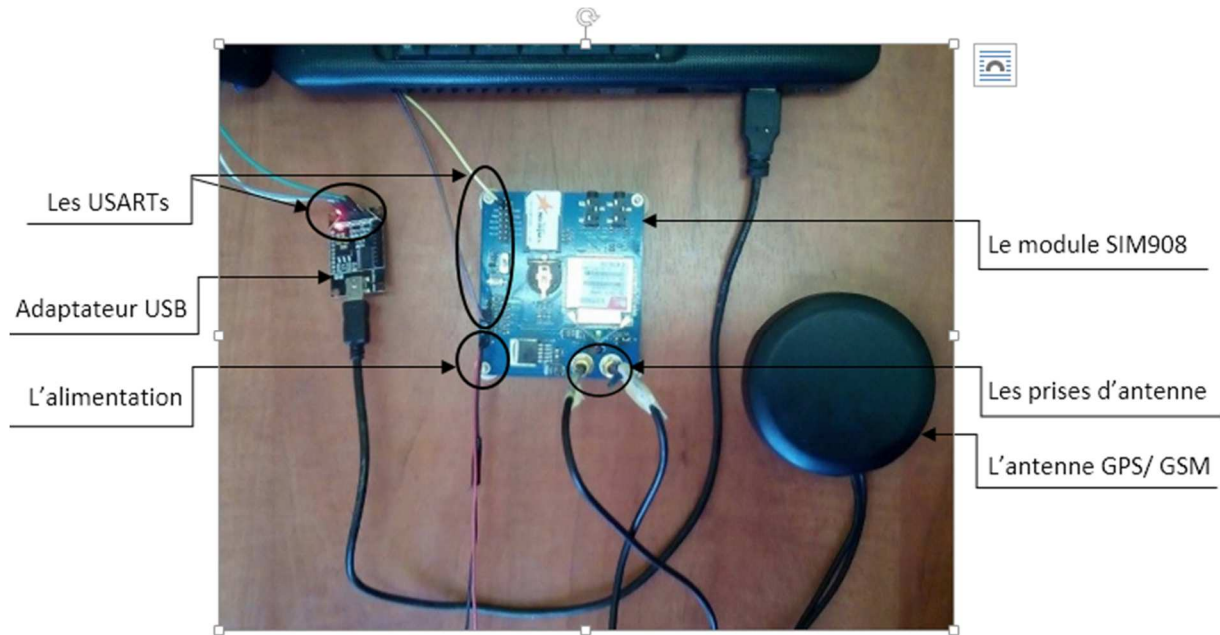


Fig. IV.5. Test du module SIM908.

Dans un premier temps, on a procédé à la configuration initiale du module, nous avons synchronisé la communication, configuré les messages au format texte, réinitialisé le module GPS à froid et défini le numéro du centre de messagerie. Les résultats des différentes commandes sont présentés sur la figure IV.6.

```

COM3 - PuTTY
RDY

+CFUN: 1

+CPIN: READY

GPS Ready

Call Ready
AT      } Synchronisation de la communication.
OK      }
AT+CMGF=1 } Configuration des messages au format texte.
OK      }
AT+CGPSRST=0 } Réinitialisation du module GPS à froid.
OK      }
AT+CSCA="+21350001701" } Définition du numéro du centre de messagerie.
OK      }

```

Fig. IV.6. Résultats de la configuration initiale du module SIM908.

Nous avons ensuite testé la partie GPS du module en demandant la trame NMEA, le résultat est présenté sur la figure IV.7.

```

COM3 - PuTTY
AT+CGPSRST=0
OK
AT+CGPSINF=0
0,357.469332,3643.701357,135.272675,20150530101952.000,42,11,0.000000,185.987579
OK

```

Demande de la trame NMEA.

Fig. IV.7. Résultats de la demande de la trame via le module SIM908.

Au final, nous avons testé la partie GSM du module, en envoyant le message « celui-ci est un message de test du SIM908 » au numéro de téléphone « +213561345652 » (Fig.VI.8). Le résultat de l'envoi est présenté sur la figure IV.9.

```

COM3 - PuTTY
AT+CSCA="+21350001701"
OK
AT+CMGS="+213561345652"
> celui-ci est un message de test du SIM908
+CMGS: 110
OK

```

Envoi d'un SMS

Fig. IV.8. Envoi d'un SMS via le module SIM908.

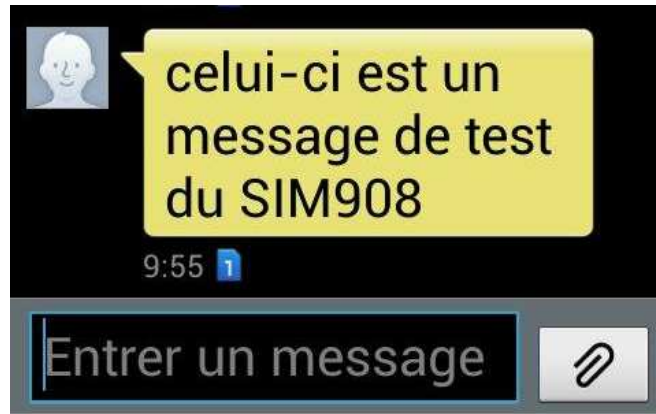


Fig. IV.9. Résultat de l'envoi d'un SMS via le module SIM908.

II.2. Test de notre système :

La figure IV.10 montre le branchement final de notre système, après avoir injecté le code de notre application dans l'AT91SAM7S256, on l'a reliée avec le module SIM908 via la liaison série USART et on a alimenté le tout à l'aide d'un transformateur de l'allume-cigare (12v vers 5 v). Le test a été effectué à l'aide d'une voiture garée au niveau du parking de notre département et d'un téléphone tenu à l'intérieur du département.

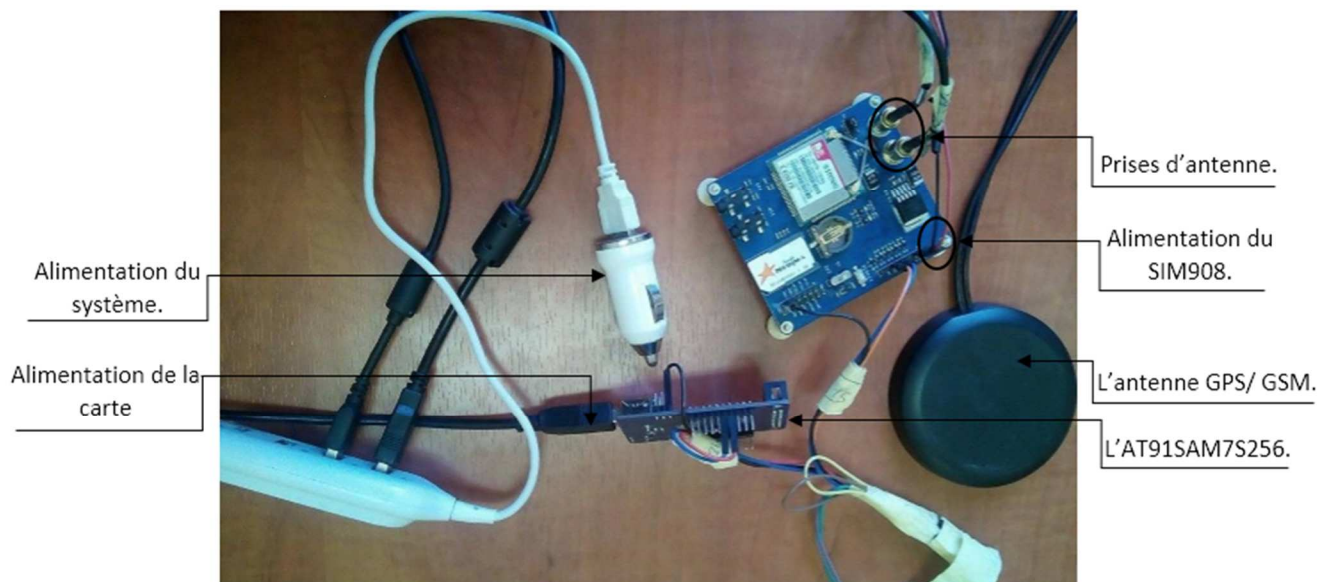


Fig. IV.10. Branchement du système.

II.2.1. Demande de la position du véhicule :

Nous avons testé la tâche principale du système qui consiste à savoir la localisation exacte de notre véhicule, comme le montre la figure IV.11 , nous avons envoyé un SMS contenant le texte « Appelle » qui est le mot de passe prédéfini, et au bout de quelques secondes le système répond par un SMS contenant les coordonnées géographiques de la position du véhicule et un lien vers google maps.



Fig. IV.11. Envoi du mot de passe par SMS et réception de la position.

Suite à la réception de la position, nous avons visualisé la localisation du véhicule sur google maps en suivant le lien reçu, comme on peut le voir sur les deux captures d'écran du téléphone de la figure IV.12.

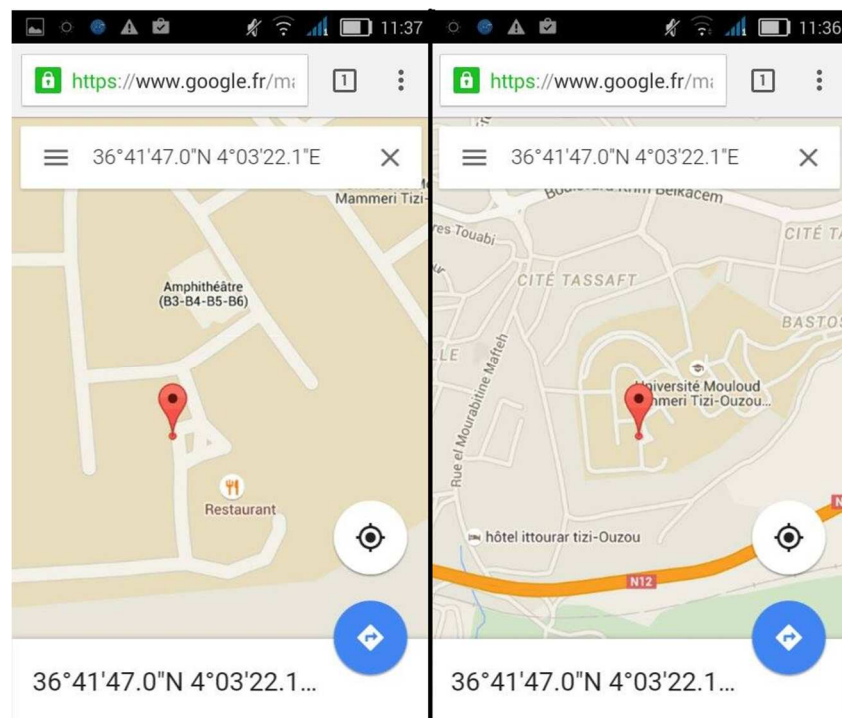


Fig. IV.12. Visualisation de la position du véhicule.

II.2.2. Demande de modification du mot de passe :

Nous avons testé la tâche de modification du mot de passe en envoyant un SMS contenant la commande permettant cela. Comme le montre la figure IV.13, le SMS envoyé contient le texte « \$ Appelle Ummto; » et au bout de quelques secondes nous avons reçu un SMS de confirmation de la modification de l'ancien mot de passe (Appelle) par le nouveau (Ummto).

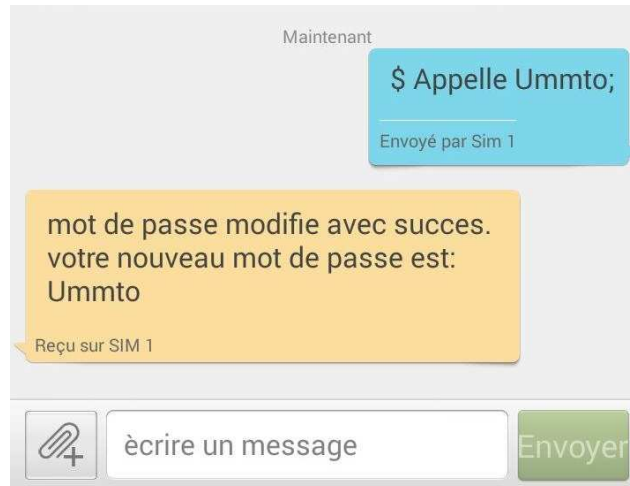


Fig.IV.13. Envoi de la commande de modification du mot de passe et réception de la confirmation.

Conclusion :

Lors de la phase de la réalisation, nous avons implémenté la partie logicielle de notre système à l'aide de l'environnement de développement Keil- μ vision, en utilisant le langage assembleur ARM et nous avons testé individuellement les différents composants matériels utilisés grâce au logiciel Putty. Suite à cela nous avons testé l'intégralité du système en réunissant les conditions nécessaires. En ayant les résultats attendus, on peut conclure par la validation des deux parties (logicielle et matérielle) du système.

Conclusion
générale

L'objectif du présent travail étant de concevoir et développer un système permettant de localiser un véhicule en cas de vol. L'utilisation étendue des réseaux téléphoniques sans fil et la disponibilité de divers moyens de localisation, notamment par satellites ont permis le développement d'un grand nombre de systèmes embarqués pour conforter, sécuriser et gérer les véhicules. Ainsi, notre attention s'est orientée vers l'utilisation d'un équipement intégrant à la fois un module radio pour le réseau mobile GSM et un module de positionnement par satellites GPS : le module SIM908 de la famille SimCom. Un dispositif pour le contrôle de ce module étant nécessaire, on a choisi le microcontrôleur AT91SAM7S256 qui dispose d'un processeur d'une grande performance : l'ARM7TDMI. Ces équipements permettront le calcul des coordonnées géographiques de la position du véhicule et leur envoi. Pour être en interaction avec le système, le propriétaire utilise son smartphone pour demander la position de son véhicule et la visualiser sur un logiciel de cartographie.

Afin de mieux cerner et comprendre les caractéristiques de notre matériel, nous avons présenté dans le premier chapitre des généralités sur les systèmes embarqués, leur architecture, leurs types et leurs utilisations. Ayant utilisé les modules GPS et GSM du module SIM908, le principe de la géolocalisation GPS/GSM et son utilisation ont été introduit au deuxième chapitre. Dans cette partie, nous avons détaillé la structure du système de positionnement GPS et son fonctionnement, celle du réseau GSM et son service SMS. Pour développer notre système, nous avons suivi une méthodologie typique : analyse, conception, codage et tests. Les deux premières phases ont fait objet du troisième chapitre dans lequel les deux parties du système (logicielle et matérielle) ont été traitées et éclairées. Le quatrième chapitre a donc porté sur les deux dernières phases, l'implémentation et les tests. Les outils de développements utilisés ont été mis en avant et différents tests confortés de figures ont été présentés.

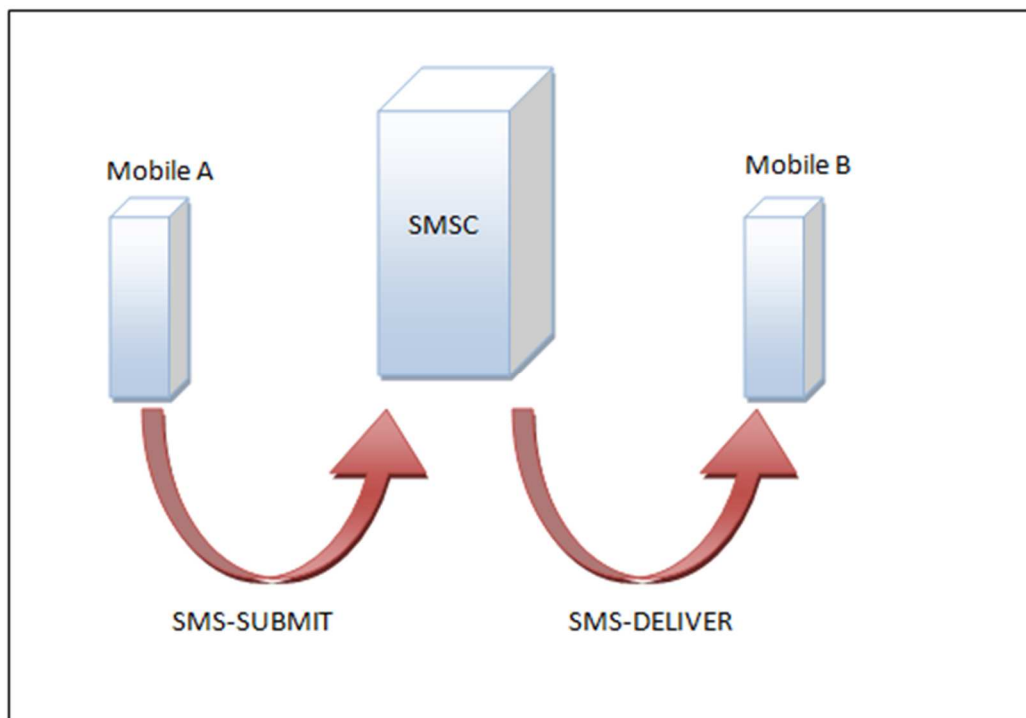
L'objectif fixé au début du travail est atteint, néanmoins, des améliorations peuvent être apportées. Le système pourrait donner plus d'informations, par exemple : la date, l'heure, la vitesse...etc. En plus du rôle de localisateur, notre système pourrait être configuré pour jouer le rôle d'un traceur permettant de voir l'itinéraire suivi par le véhicule en envoyant sa position de manière répétitive. Il peut aussi être programmé de manière à ce que dès que le véhicule quitte un rayon prédéfini, un message soit envoyé vers l'utilisateur.

Pour une utilisation plus aisée du système, une application cliente au niveau de l'ordinateur ou du Smartphone du client pourrait être développée.

Annexe

A- Les protocoles SMS-SUBMIT et SMS-DELIVER du réseau GSM.

Lorsqu'un mobile A envoie un SMS à un mobile B, le message est acheminé vers le centre de messagerie (SMSC) c'est ce qu'on appelle le **SMS-SUBMIT**. Une fois traité par le SMSC le message est délivré au mobile B, on parle alors du **SMS-DELIVER** (Fig A.1).



FigA.1 les protocoles SMS-SUBMIT et SMS-DELIVER

I- SMS-SUBMIT

La trame envoyée par un téléphone GSM vers le SMSC est comme suit :

1-10 octets	1 octet	1 octet	2-12 octets	1 octet	0-7 octets	1 octet	1 octet	0-140 octets
SCA	PDU	MR	DA	PID	DCS	VP	UDL	UD

La taille maximale de la trame d'un SMS-SUBMIT est de 173 octets. Le champ le plus important en termes de taille est le champ qui codifie le corps du message qui peut atteindre 140 octets.

- **SCA: Service Center Adress**

Adresse du centre de messagerie, sa taille dépend de la longueur du numéro du SMSC utilisé.

- **Type de PDU**

Le champ PDU toujours codé sur 1 seul octet, il a pour fonction principale de définir s'il s'agit d'un SMS-DELIVER ou d'un SMS-SUBMIT. Dans le cas du SMS-SUBMIT les deux premiers bits de ce champ sont positionnés à « 01 ».

- **MR: Message reference**

Chaque message envoyé par le mobile au SMSC est identifié par un numéro compris entre 0 et FF_{hex} (Référence du Message).

- **DA: Destination Adress**

Indique l'adresse du destinataire.

- **PID: Protocol Identifier**

Le champ PID indique à quel type de service télématique est destiné le message. Dans le cas du réseau GSM ce champ sera toujours positionné à 00_{hex} c.à.d. que la trame est traitée comme un message court.

- **DCS: Data Coding Scheme**

Le champ DCS indique de quelle manière est codé le champ UD qui correspond au corps du message. Dans la pratique, tous les bits sont positionnés à zéro c.à.d. que le champ UD est codé avec l'alphabet GSM (avec l'alphabet GSM chaque caractère est codé sur 7 bits).

- **VP : Validity Period**

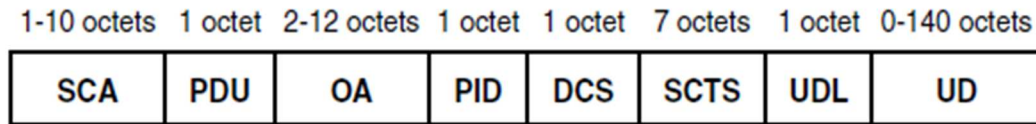
Permet d'indiquer au SMSC la durée de validité du SMS.

- **UDL : User Data Length, UD : User Data**

UDL contient la taille en octets utilisés pour codifier le message dans UD.

II- SMS-DELIVER :

La trame envoyée par le SMSC vers un téléphone GSM est comme suit :



- **SCA : Service Centre Adresse**
Adresse du centre de messagerie, il indique quel est le SMSC qui a traité le SMS.
- **PDU : Protocol Data Unit**
Les deux premiers bits de ce champ sont positionnés à « 00 » pour indiquer qu'il s'agit d'un SMS-DELIVER.
- **OA : Originator Adress**
Indique l'adresse de l'émetteur.
- **PID : Protocol Identifier, DCS: Data Coding Scheme**
Ces champs sont identiques à ceux présentés dans la partie SMS-SUBMIT.
- **SCTS : Service Centre Time Stamp**
Le champ SCTS indique au destinataire la date et l'heure à laquelle le SMS est arrivé au SMSC.
- **UDL : User Data Length, UD : User Data**
Ces champs sont identiques à ceux présentés dans la partie SMS-SUBMIT.

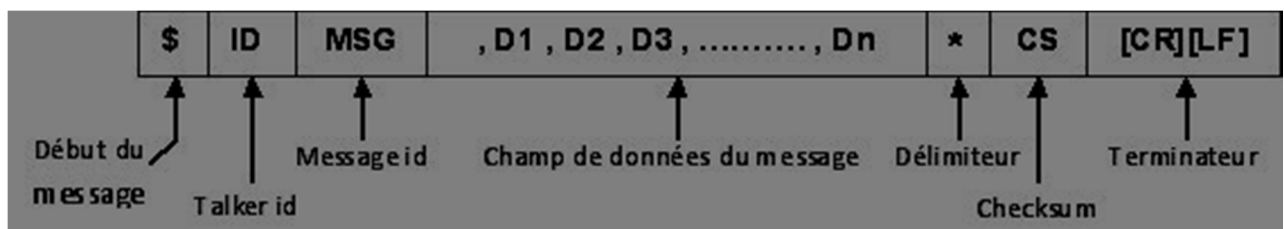
B- Le standard NMEA.

NMEA (National Marine Electronics Association) est une association américaine de fabricants d'appareils électroniques maritimes, basée à Severna Park au Maryland (États-Unis d'Amérique). Elle est à l'origine de nombreux standards et en particulier du Standard NMEA-0183.

NMEA 0183 est une spécification pour la communication entre équipements marins, dont les équipements GPS. Sous ce standard, toutes les données sont transmises sous la forme des caractères ASCII, tous imprimables, ainsi que les caractères [CR] Retour Chariot et [LF] Retour à la ligne. Les données sont transmises sous forme de trames.

Dans ce qui suit, le Standard NMEA est défini comme étant le protocole de transmission des données entre les équipements électroniques liés au GPS.

I- Format de la trame NMEA 0183 :



Chaque trame commence par le caractère \$

Suivi par un groupe de 2 lettres pour l'identifiant du récepteur. ID

Exemples :

- GP pour Global Positioning System.
- LC Loran-C receiver.
- OM Omega Navigation receiver.
- II Integrated Instrumentation (eg. AutoHelm Seatalk system).

Puis un groupe de 3 lettres pour l'identifiant de la trame (MSG).Exemples :

- GGA : pour GPS Fix et Date.
- GLL : pour Positionnement Géographique Longitude - Latitude.
- GSA : pour DOP et satellites actifs.
- GSV : pour Satellites visibles.
- VTG : pour Direction (cap) et vitesse de déplacement (en noeuds et Km/h).
- RMC: pour données minimales exploitables spécifiques.

Suivent ensuite un certain nombre de champs séparés par une "virgule". La virgule permet la dé-concaténation des données dans le programme de traitement des données, calculateur, navigateur.

Et enfin un champ optionnel dit checksum précédé du signe *, qui représente le « OU exclusif » de tous les caractères compris entre \$ et * (sauf les bornes \$ et *), certaines trames exigent le checksum.

Suit la fermeture de la séquence avec un [CR][LF].

Une trame comporte au maximum 82 caractères.

II- Exemple de trame (GLL) :

\$GPGLL,4916.45,N,12311.12,W,225444,A

4916.46,N = Latitude 49 deg. 16.45 min. Nord.

12311.12,W = Longitude 123 deg. 11.12 min. West (ouest)

225444 = Acquisition du Fix à 22:54:44 UTC

A = Données valides

Pas de checksum

III- La trame manipulée par notre système :

0,357.469332,3643.701357,135.272675,20150530101952.000,42,11,0.000000,185.987579

(1) (2) (3) (4) (5) (6) (7) (8) (9)

- (1) Type de la trame
- (2) La longitude
- (3) La latitude
- (4) L'altitude
- (5) Date et heure aaaammjjhhmmss
- (6) TTFF (time to first fix) temps la 1^{ère} fixation en seconde.
- (7) nombre de satellites en vue
- (8) Vitesse
- (9) route sur le fond.

C- La technologie Machine To Machine (M2M).

Le M2M résulte de la convergence de trois familles de technologies : des objets intelligents reliés par des réseaux de communication avec un centre informatique capable de prendre des décisions. Les usages du Machine To Machine sont multiples : La gestion de flotte, la gestion de la chaîne d'approvisionnement (Supply Chain Management), la télésurveillance, la surveillance des biens et des personnes...etc.

I. Définition :

M2M consiste en l'utilisation d'un équipement (capteur, compteur, etc.) pour capturer un événement (température, mesure sismique, etc.) qui est relayé à travers un réseau de communication mobile, fixe ou hybride à une application (ex. du système d'information de l'entreprise qui utilise ces équipements M2M). Cette dernière traduit l'événement capturé en des informations significatives. Une solution M2M est le résultat d'une interaction continue entre les équipements M2M, les réseaux de communication et les applications.

II. Architecture :

Comme indiqué sur la figure C.1, le M2M permet un échange bidirectionnel d'informations entre l'équipement M2M et l'application. L'information reçue est ensuite traitée. Dans de nombreux cas, le M2M implique un groupe de « devices » semblables interagissant avec une application.

Dans certains cas, les équipements dans le groupe ne peuvent pas directement communiquer avec l'application du fait de leurs capacités limitées. Dans ces conditions, un équipement de médiation (GW, Gateway) est nécessaire, permettant de consolider la communication.

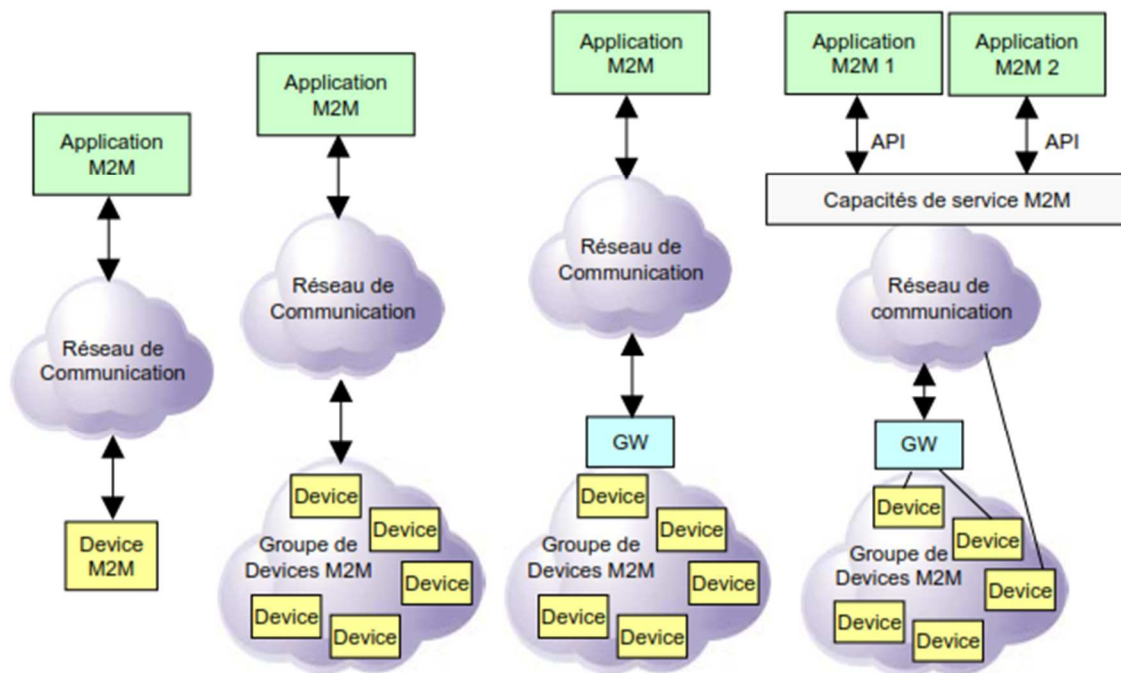


Fig.C.1. Architecture M2M.

III. Opérations principales :

Dans toute solution M2M, il y a quatre opérations principales: la récolte de données, son transport et son traitement et le réveil du device afin qu'il puisse émettre un rapport de données non programmé.

- Réveil pour envoi de données (domaine des télécommunications) : Les devices M2M sont généralement programmés pour se réveiller à intervalle de temps fixe (e.g., toutes les heures), réaliser des mesures, s'attacher au réseau, établir une connexion de données, transférer leur rapport, puis libérer leur connexion et se détacher du réseau. Il peut arriver que l'application souhaite que le device M2M lui communique un rapport de données non programmé. L'application réveille donc le device par exemple par SMS et ce dernier transmet les données au serveur M2M.
- Récolte de données (domaine de l'électronique) : La remontée d'informations se fait grâce aux capteurs embarqués dans les devices M2M. Les évolutions technologiques dans ce domaine donnent naissance à des dispositifs de moindre taille, moins coûteux et moins consommateurs en énergie.
- Transport des données (domaine des télécommunications) : Plusieurs technologies de réseaux, radio ou filaire, peuvent coexister dans une même solution M2M. Le choix technologique dépendra de la couverture requise, du mode de connectivité, de la quantité de données à transmettre, de sa fréquence et du modèle économique.
- Traitement des données (domaine de l'informatique) : L'application reçoit les données, les traite et intègre les données résultantes dans le système d'information de l'entreprise.

D- Les commandes « AT ».

Les commandes AT ne sont rien d'autres que des instructions qui permettent l'accès aux fonctions d'un téléphone portable par l'intermédiaire d'un terminal. Spécifiées par un standard de télécommunication européen (ETS), ces commandes s'inspirent des commandes universelles permettant de piloter un modem. Les téléphones portables doivent être construits de manière à respecter les deux normes suivantes :

- **GSM07.07** qui permet l'accès aux fonctions générales du téléphone ;
- **GSM07.05** qui concerne la gestion des SMS.

Trois entités sont définies dans les textes officiels qui traitent le GSM :

TE : Terminal Equipment qui peut être un ordinateur ou un microcontrôleur (envoi et affiche les commandes).

TA : Terminal Adaptator qui est l'interface entre l'utilisateur et le mobile.

ME : Mobile Equipment qui correspond par exemple à un téléphone portable.

La figure suivante (Fig.D.1) illustre les entités nécessaires pour accéder aux fonctions d'un terminal mobile (notamment les fonctions du module GSM) en utilisant les commandes AT.

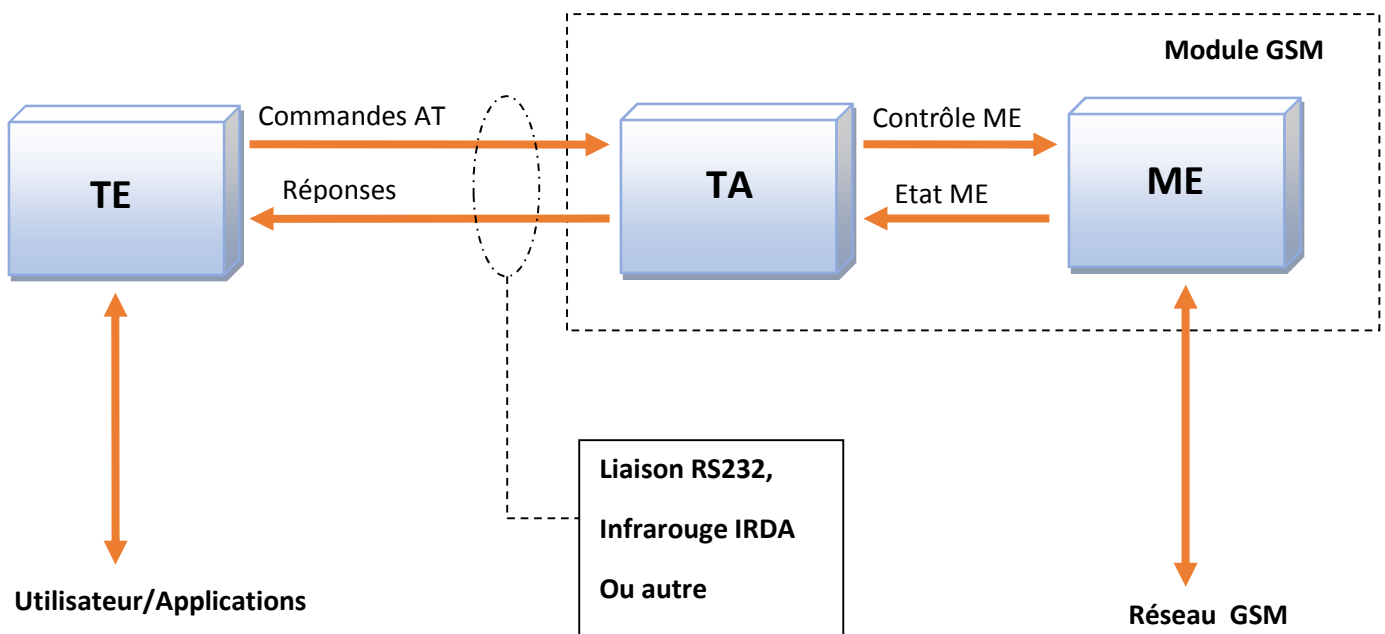


Fig.D.1. Gestion du module GSM à l'aide des commandes AT.

I. Syntaxe d'une commande AT :

Le préfixe «AT» doit être mis au début de chaque commande, et pour terminer la ligne de commande on entre <CR>.

On peut envoyer une commande AT de trois manières :

- **AT+CXXX=? (Commande de test)**, retourne la liste des paramètres utilisables avec la commande CXXX.
- **AT+CXXX? (Commande de lecture)**, retourne le ou les paramètres en cours associés à la commande CXXX.
- **AT+CXXX=<xxx> (Commande d'écriture)**, applique le ou les paramètres <xxx> à la commande CXXX.

Une fois la commande envoyée, le téléphone doit y répondre favorablement ou non et la réponse retournée est de la forme :

- **<CR><LF>OK<CR><LF>** si la commande est acceptée ;

- `<CR><LF>ERROR<CR><LF>` (accompagné éventuellement d'un message décrivant la nature de l'erreur), si la commande n'est pas reconnue, ou que le ME rencontre un problème lors de son exécution.

Il faut noter que le `<CR>` signifie *Carriage Return* (retour chariot) et le `<LF>` signifie *Line Feed*. (nouvelle ligne)

II. La norme GSM07.07 :

La norme GSM07.07 regroupe environ 80 commandes permettant d'accéder à toutes les fonctions du ME, le tableau suivant nous donne quelques exemples.

Commandes	Fonction
AT+CGSN	Identification numéro de série (IMEI)
AT+CGMI	Identification fabricant
AT+CPBS	Sélectionne un répertoire téléphonique
AT+CPBR	Lecture du répertoire téléphonique
AT+CSCS	Alphabet utilisé par le TE
AT+CCLK	Horloge
AT+CLIP	Présentation du numéro

Exemple de réponse : (la commande AT+CGSN)

Commande de teste : AT+CGSN= ?

OK

Si erreur

+CME ERROR :<err>

Commande de lecture : AT+CGSN

+CGSN :<sn>

Retourne le numéro de série du ME nommé IMEI (International Mobile station Equipment Identity)

Si erreur

+CGMI ERROR :<err>

III. La norme GSM07.05 :

La norme GSM07.05 spécifie les commandes AT permettant la gestion des SMS comme on peut le voir sur les exemples du tableau suivant :

Commandes	Fonction
AT+CSMS	Sélection du service de messagerie
AT+CSCA	Définition de l'adresse du centre de messagerie
AT+CMGF	Sélection du format du SMS (PDU ou TEXT)
AT+CSDH	Affiche en mode TEXT le paramétrage des SMS
AT+CMGR	Lecture d'un SMS
AT+CMGS	Envoie un SMS
AT+CMGW	Écriture d'un SMS

Exemple de réponse : (la commande : AT+ CSCA)

Commande de teste : AT+CSCA= ?

OK

Si erreur

+CMS ERROR : <err>

Commande de lecture : AT+CSCA ?

+CSCA :<sca> , <tosca>

OK

Si erreur

+CMS ERROR : <err>

Commande d'écriture : Si mode TEXTE (+CMGF=1) : AT+CSCA=<sca> [, <tosca>]

OK

Définit l'adresse du centre de messagerie à utiliser pour l'envoi des SMS.

Si erreur

+CMS ERROR : <err>

Bibliographie

[SE1] : Les systèmes embarqués Introduction, Pr Richard Grisel, Université de Rouen, Pr Nacer Abouchi, ESCPE Lyon.

[SE2] : Une Approche Basée Architecture pour la Spécification Formelle des Systèmes Embarqués, Thèse de Doctorat, Malika BENAMMAR, Université de Constantine, 2011.

[SE3] : Les Systèmes Embarqués, thèse d'Ingénieur, Ramzi BOULKROUNE, université d'Annaba, 2009.

[SE4] : Linux embarqué 2^{ème} édition, pierre ficheux , Editions EYROLLES.

[SE5] : Le livre blanc des systèmes embarqués, Eric Bantegnie, SYNTEC INFORMATIQUE.

[SE6] : <http://www.technologuepro.com>.

[SE7] : Dossier de Presse « Systèmes embarqués pour l'automobile : une nouvelle dynamique pour accélérer l'innovation dans la filière électronique automobile », septembre 2012, CEA / Service Information-Media, Coline VERNEAU.

[GSM1] : Interfaces du GSM 2^{ème} édition Montages pour téléphones portables par David REY, DUNOD, Paris 2004-2010.

[GPS1] : Découvrir l'univers du GPS & exploiter son potentiel par Luc AEBI, INLIBROVERITAS, 2007.

[GPS2] : Techniques et technologies de localisation avancées pour terminaux mobiles dans les environnements indoor par Frédéric EVENNOU.

[GPS3] : GPS et Galiléo Systèmes de navigation par satellites Jean-Marc Piéplu, EYROLLES.

[GPS/GSM-1] : La gestion de flotte de véhicules par GPS et GSM, Stéphane SOLEILHAVOUP, technologie 165, janvier-février 2010.

[GPS/GSM-2] : GSM/GPS, Localisateur, ELECTRONIQUE magazine - n°123.

[GEO1] : Application des techniques d'apprentissage à la géolocalisation par radio fingerprint, thèse de Doctorat, Iness Ahriz Roula, université pierre et marie curie, 2010.

[GEO2] : <http://help.arcgis.com/fr>

[RES] : http://www.memoireonline.com/07/10/3741/m_Evolution-technologique-de-la-3G-et-3G1.html

[Con1] : Architecture logicielle et méthodologie de conception embarquée sous contraintes temps réel pour la radio logicielle, Thèse de Doctorat, Noël Bertrand TCHIDJO MOYO, université de rennes 1, 2011.

[ARM1] : PROCESSEURS ARM, Architecture et langage d'assemblage, Jacques Jorda, Dunod, Paris, 2010.

[ATM1] : manuel d'utilisation des cartes SAM7S [datasheet].