



REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE



MINISTERE DE L'ENSEIGNEMENT SUPERIEUR  
ET DE LA RECHERCHE SCIENTIFIQUE  
LABORATOIRE DE TELECOMMUNICATIONS

MEMOIRE  
Pour l'obtention du

DIPLOME DE MASTER

EN

TELECOMMUNICATION

OPTION

RESEAUX ET SYSTEMES DE TELECOMMUNICATIONS

***THEME***

# Intégration d'un service de localisation GSM/GPS

Soutenu le 24 Juin 2013

**Réalisé par :**

BELDJILALI Bilal

BELKHITER Abdessalam

**Devant les membres du jury composée de :**

- |                              |              |
|------------------------------|--------------|
| • Professeur F.T. BENDIMERAD | Président    |
| • Dr A. ABDELMALEK           | Examineur    |
| • Dr A. BOUACHA              | Examineur    |
| • Dr B. BENADDA              | Encadreur    |
| • Mr G. ABDELLAOUI           | Co-Encadreur |

**Année universitaire : 2012-2013**

# *Remerciement*

---

## *Remerciement*

Nous remercions tout d'abord le Dieu Tout Puissant de nous avoir armés de force et de courage pour mener à terme ce projet.

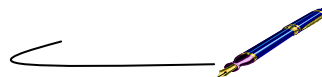
Au terme de ce travail, nous adressons nos remerciements les plus sincères à notre encadreur Mr **Belkacem BENADDA**, pour nous avoir permis de bénéficier de son grand savoir dans la matière, pour sa pédagogie, ses compétences, sa modestie et son aide précieuse tout au long de ce projet même pendant les moments les plus difficiles. Vraiment merci pour une qualité d'encadrement si sérieuse et si consistante.

Notre profonde gratitude s'adresse à Monsieur le Professeur **Fathi Tarek BENDIMERADE**, responsable du laboratoire de Télécommunications de Tlemcen (**LTT**).

En fin, nous exprimons notre profonde gratitude à tous les enseignants de l'université de Tlemcen pour la qualité de l'enseignement qu'ils nous ont bien voulu prodiguer durant nos études.

Enfin nos vifs remerciements aux membres du jury de nous honorer de leur présence et de porter leur jugement sur ce modeste travail.

**BELDJILALI BILAL ET BELKHITER ABDESSALAM**





## *Dédicaces*

*A mon père et ma mère qui n'ont jamais détourné leurs regard sur moi et ont toujours guidé mes pas sur tous les sentiers et aidé à surmonter les différents obstacles sur mon chemin, je dédie ce travaille.*

*A mon frère et mes Sœurs et à toute la famille Beldjilali.*

*A tous mes amis surtout mes amis de la promo RST : Imane, Yasmine, Ismail, Ibrahime, Youcef, Tarike, Adila, Wahiba, Assia, Sara, Alae, Lila..... et tous mes amis Anes Aissa yacine Islame Zaki Amin pour les moments agréables que nous avons passés ensemble.*

*Beldjilali Bilal*



*Au nom d'Allah le tout Miséricordieux le très Miséricordieux.*

*Je dédie ce modeste travail :*

*A mes trs chers parents qui taient prsents pour moi durant toute ma vie, qu'il trouve ici le témoignage de mon amour profond et de ma gratitude certaine. Que Dieu les préserve de toute peine et de tout malheur.*

*A mes sœurs et mon frre : Je vous remercie pour le soutien moral et l'encouragement que vous m'avez accords. Je vous souhaite tout le bonheur que vous mritiez*

*A mes très chers grands-parents et à toute la famille BELKHITER et BENAMAR*

*A mes amis : Hakim, Sid ahmed, Abdou, Mohammed, Yassine, Aissa, youssef Redah, Hadjer et à tous ceux qui ont su m'apport l'aide et le soutient aux moments propices.*

*A tous mes amis de la promotion RTS : Bilal, Ismail, Ibrahim, Youssef, Alaa, Mokhetar Tarek, Amine, Mohammed, Hocine, Imane, Ysmine, Adila, Wahiba, Leyla, Assia, et à tous mes chers amis de la promotion RST*

*BELKHITER Abdessalam*





## Sommaire :

Remerciement.....	I
Dédicace .....	II
Liste des figures.....	III
Introduction générale.....	8
<b>Chapitre1 : Description de système de localisation GPS/GSM</b>	
1.1- Introduction.....	10
1.2- Description de système.....	10
1.3- Présentation du système.....	11
1.3.1- Sonde embarquée.....	12
1.3.2- Système de télésurveillance.....	13
1.4- Impact attendu.....	14
1.5- Conclusion.....	14
<b>Chapitre2 : Description des systèmes GPS et GSM</b>	
2.1- Introduction.....	16
2.2- Système GPS.....	16
2.2.1- Description du système.....	16
A- Secteur spatial.....	17
B- Secteur de contrôle.....	18
C- Secteur Utilisateur.....	18
2.3- Sources d'erreurs.....	19
2.3.1- Erreurs d'origine naturelle.....	19
2.3.2- Erreurs d'origine technique.....	19
2.4-Correction d'erreur.....	20
2.5-Signal GPS.....	20
2.5.1- Code pseudo aléatoire.....	21
2.5.2-Contenu du Signal.....	22
2.6- Principe de fonctionnement.....	23
2.6.1- Calcul de la position.....	23
2.6.2- Conversion des coordonnées.....	25
2.7- GSM.....	25
2.7.1- Le concept cellulaire.....	26

2.7.2-Architecture GSM.....	26
A-Mobile station.....	27
B-Sous-système radio.....	27
C-Sous-système réseau.....	27
D-Sous-système d'exploitation.....	28
2.7.3-GPRS.....	28
2.7.4- EDGE.....	29
2.8- Conclusion .....	29

### **Chapitre3 : la sonde GM862-GPS**

3.1- Introduction .....	31
3.2- Définition .....	31
3.3- Caractéristique .....	31
3.3.1- caractéristique du module .....	31
3.3.2- Caractéristiques de l'antenne GPS .....	32
3.3.3- Caractéristique de l'antenne GSM.....	33
3.4- Configuration et programmation du module .....	34
3.4.1- Les commande AT .....	34
3.4.2- configuration du module .....	34
A- Préparation de l'environnement .....	34
B- Configuration .....	36
3.4.3- Programmation du module .....	37
A- Langage Python .....	37
B- Le script Python.....	38
C- Les trames NMEA .....	42
3.5- Conclusion.....	45

### **Chapitre4 : la station de surveillance**

4-1-Introduction.....	47
4-2-Description de la station de surveillance.....	47
4.3- Le rôle de la station de surveillance.....	47
4.3.1- la réception des données.....	48
A- Réception des données dans URL.....	49
B- Traitement des données.....	49
C- Enregistrement des données dans la base de données .....	54
4.3.2- l'affichage de la position sur une carte GoogleMaps.....	57

A-Réception d'une demande d'un client.....	58
B-Récupération des coordonnées enregistrées dans la BD.....	59
C-Chargement de la carte GoogleMaps.....	60
D-L'affichage des coordonnées dans GoogleMaps.....	61
E-Affichage d'un parcours.....	62
4.4- Conclusion.....	63

### **Chapitre5:Application**

5.1- Introduction .....	66
5.2- Administration de l'application .....	66
5.3- Suivi en temps réel .....	68
5.4- Suivi en temps différé .....	69
5.5- Conclusion .....	70
Conclusion générale .....	71
Bibliographie.....	72

## Liste des figures :

Fig1.1: Schéma global du dispositif à réaliser.....	11
Fig1.2: Schéma fonctionnel global.....	12
Fig1.3: Composantes de la sonde de localisation.....	12
Fig1.4: La station de surveillance .....	13
Fig2.1:Secteurs du GPS.....	16
Fig2.2:Satellite GPS.....	17
Fig2.3:Répartition des satellites GPS mai 2006.....	17
Fig2.4:Répartition des secteurs de contrôle GPS. ....	18
Fig2.5:Génération du signal.....	20
Fig2.6:Code C/A et P.....	21
Fig2.7:Contenu de message de navigation.....	22
Fig2.8:Différents paramètres pour calculer la position.....	24
Fig2.9:Position en Longitude, latitude et hauteur.....	25
Fig2.10: motif du réseau GSM.....	26
Fig2.11: Architecture GSM.....	26
Fig3.1: module GS862-GPS.....	32
Fig3.2: Antenne GPS pour GM-862-GPS.....	33
Fig3.3: Antenne GSM pour GM862-GPS.....	33
Fig3.4 Interface de HypeTerminal.....	35
Fig3.5: Format d'une trame NMEA.....	42
Fig4.1:Algorithme de la réception des données.....	48
Fig4.2:Description de la méthode de réception des données.....	48
Fig4.3:Les bases des données créées par MySQL Administrator.....	55
Fig4.5:Les tableaux créés par MySQL Administrator.....	56
Fig4.6:Algorithme pour l'affichage des positions sur une carte .....	67

Fig4.7:Description de la méthode pour l’affichage des positions sur la carte .....	68
Fig4.8:Principe de fonctionnement d’un serveur Web.....	68
Fig4.9:Récupération des données à partir d’une base de données .....	59
Fig5.1: Fenêtre d’Authentification.....	66
Fig5.2: Espace administrateur.....	67
Fig5.3 : Fenêtres des différentes opérations gérer par l’administrateur.....	67
Fig5.4: Position du véhicule en temps réel.....	68
Fig5.5: Parcours d’un véhicule pendant une durée.....	69

**Liste des tableaux :**

Tableau2.1: Les erreurs et leurs corrections.....	20
Tableau3.1: Les champs de la trame GPSACP.....	44
Tableau4.1: Les valeurs récupéré avec PHP.....	52

## **Introduction générale:**

Ce projet présente une conception Matériel et logiciel d'un dispositif de localisation GPS à installer sur un engin destiné au transport publique ou de marchandises. Le réseau GSM est utilisé pour effectuer un transfert des coordonnées calculées par le GPS vers une machine serveur connectée au réseau Internet et dotée elle même d'une interface GSM ou éventuellement une connexion internet. Il s'agit ainsi d'une machine communicante qui effectue des transmissions de données avec un protocole spécifique GSM. Notre choix va être porté vers les protocoles qui s'adaptent à la troisième génération des réseaux mobiles qui sera largement déployées d'ici peu de temps en Algérie, et reste compatible avec la seconde génération GSM. Un tel choix est effectué pour des raisons de simplicité, efficacité, faible coût et très haute confidentialité.

Notre travail est divisé en cinq parties :

Le premier chapitre est une description fonctionnelle du système de localisation GSM/GPS que nous prévoyons de concevoir et d'implémenter à ce stade. Dans notre cas, ce système se présente comme une intégration de deux systèmes GPS et GSM : il utilise le système GPS pour la détermination des positions puis le système GSM comme un moyen de transmission de ces dernières.

Le deuxième chapitre est une description des systèmes GPS et GSM : l'architecture, les différents sous-systèmes qui les composent ainsi que les principes de fonctionnement de chaque système. Les informations présentées permettent de dimensionner le système ainsi que les algorithmes à implémenter pour notre solution.

Le troisième chapitre est consacré au module GM862-GPS utilisé pour réaliser notre solution et son contrôle à travers des commandes AT. Le langage Python, est présenté comme un langage de script qui permet d'automatiser la réception des coordonnées GPS au format NMEA et les envoyer à travers une connexion de données GSM tel que le GPRS.

Le quatrième chapitre, présente les outils utilisés pour concevoir notre station de surveillance. Cette station est responsable du traitement des données reçues à partir de chaque sonde, pour atteindre les objectifs tracés. Pour cela nous avons travaillé avec plusieurs logiciels et langages de programmation.

Le cinquième chapitre figure la structure de notre portail WEB accessible à travers internet via l'URL : [www.localisationrst.netii.net](http://www.localisationrst.netii.net). Ce dernier, est composé de deux parties, l'une dédiée aux utilisateurs et l'autre aux clients avec en plus des privilèges de contrôle sur les clients standards. Nous avons également doté notre serveur de surveillance de la possibilité d'effectuer un suivi en temps réel, en plus de celui dit en temps différé.

# Chapitre 1

## Description de système de localisation GPS/GSM



**1.1- Introduction :**

Le système de localisation GPS/GSM est un procédé permettant de positionner un objet (une personne, des engins etc.) sur un plan ou une carte à l'aide de ses coordonnées géographiques.

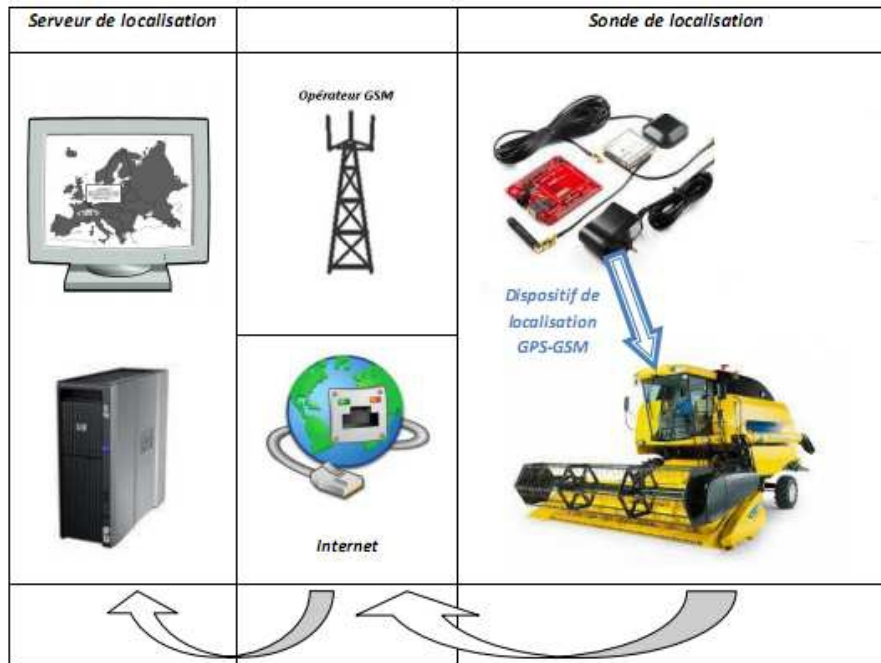
Cette opération est réalisée à l'aide d'un terminal capable d'être localisé (grâce à un système de positionnement par satellites ( un récepteur GPS par exemple) et de publier (en temps réel ou de façon différée) ses coordonnées géographiques (latitude/longitude). Ces positions seront transmises en temps réel vers une plateforme logicielle de géo-localisation. La transmission temps réel nécessite un terminal équipé d'un moyen de télécommunication de type GSM, GPRS lui permettant d'envoyer les positions à des intervalles réguliers. Ceci permet de visualiser la position du terminal au sein d'une carte (GoogleMaps).

Dans ce chapitre nous allons présenter une description globale d'un système de localisation GPS/GSM.

**1.2- Description de système :**

Le prototype que nous voulons concevoir sera un système embarqué doté d'une carte à microcontrôleur, d'un module GPS, d'un module GSM quadri bandes, avec en plus le système d'exploitation qu'il faut développer pour assurer les tâches nécessaires au bon fonctionnement de la totalité du système.

Le serveur réalisé par une technologie Internet doit comporter une base de données pour mémoriser la trace des déplacements et chemins parcourus, avec la possibilité d'afficher la position courante des engins sélectionnés. L'affichage des positions sera effectué sur une carte d'état major sous forme d'image pour avoir une vision globale des positions et faciliter la prise d'une décision.



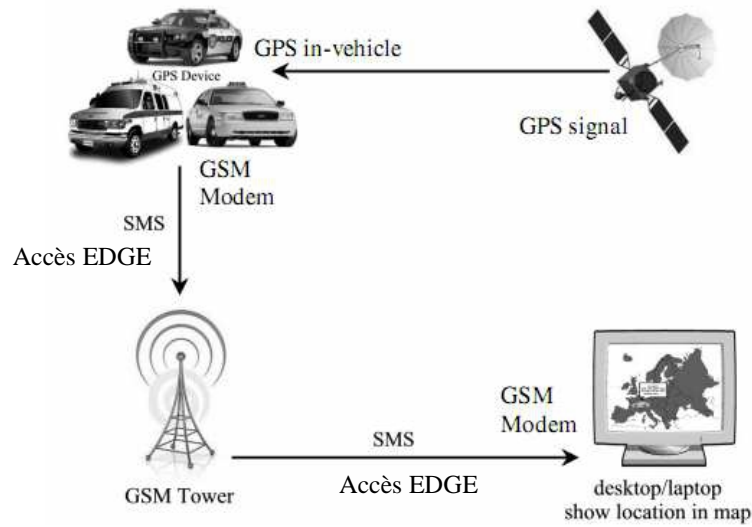
**Fig1.1: Schéma global du dispositif à réaliser**

### 1.3- Présentation du système :

Le "Système de positionnement GPS /GSM" (Figure1.1) utilise, le système de positionnement par satellites GPS pour situer géographiquement des récepteurs GPS, le réseau GSM est utilisé pour transmettre ces positions à une station de surveillance.

Le dispositif est constitué de deux systèmes :

- une sonde embarquée installée dans la cible à suivre.
- un système en mode stationnaire situé dans le centre de surveillance.

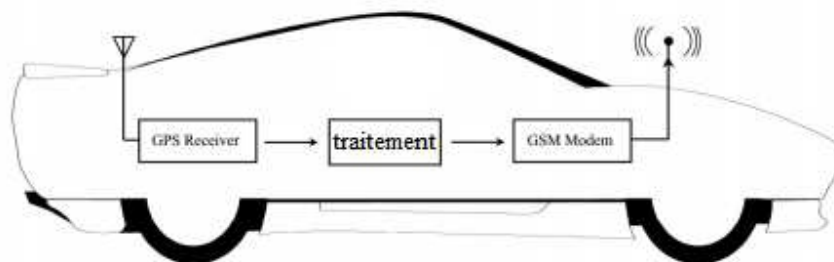


**Fig1.2 : Schéma fonctionnel global**

**1.3.1- Sonde embarquée :**

La sonde embarquée (Figure 1.3) se compose de trois modules :

- Le récepteur GPS.
- Un module de traitement de données.
- Le modem GSM.



**Fig1.3: Composantes de la sonde de localisation**

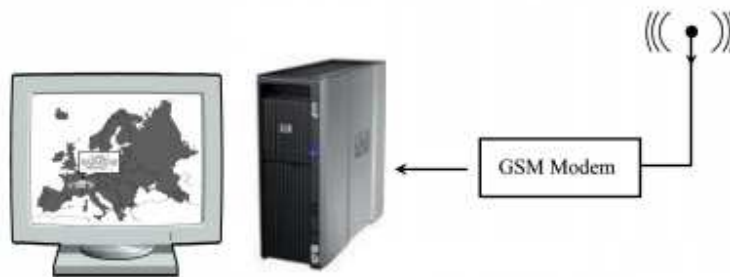
Le système embarqué sur les véhicules doit être capable:

- De calculer avec précision les coordonnées de l'engin, à travers une interface GPS ou bien GSM.
- De communiquer périodiquement ou bien à la demande, les coordonnées calculées à travers une interface GSM.

### 1.3.2- Système de télésurveillance :

Le système de télésurveillance compose en :

- Un modem GSM.
- Un serveur avec une base de données connecté à internet.



**Fig1.4: La station de surveillance**

Le modem GSM permet de recevoir les données envoyées par la sonde.

La machine serveur et de supervision connectée au réseau internet permet :

- D'enregistrer les traces des déplacements effectués,
- D'afficher sur une carte d'état major la position des engins sélectionnés,
- De communiquer des alertes pour optimiser le déplacement des engins,
- Communiquer des alertes aux responsables pour une prise rapide de décisions.

**1.4- Impact attendu :**

Le transport public en communs et de marchandises, est devenu un domaine essentiel pour le développement d'un pays. Le projet que nous proposons entre dans le cadre d'un développement de ce secteur. En effet nous visons :

- Une gestion optimisée d'une flotte d'engins destinés au transport en commun, ceci en assurant que les engins gardent une distance suffisante entre eux. De permettre la circulation du nombre nécessaire selon l'encombrement de la ligne.
- Gestion efficace des engins destinés au transport de marchandise ou de travaux publics.

Ceci est effectué en évitant par exemple qu'un engin reste sans charge utile et prévoir des déplacements améliorés de la flottes ; éviter les déplacements et détour inutiles.

- Visualiser l'activité et la position en temps réel des engins.
- Signaler en temps réel d'éventuel panes ou alarmes.

**1.5- Conclusion :**

Il va falloir à présent matérialiser les lignes de la conception que nous avons proposée dans ce chapitre, avec des technologies Internet, GSM et GPS de nature logiciel et matériel. Pour cela une étude de l'ingénierie des réseaux GSM et GPS doit être effectuée.

# Chapitre2

## Description des systèmes GPS et GSM

## 2.1- Introduction :

Nous allons présenter les principes fondamentaux des réseaux GPS et GSM, ceci pour nous aider à sélectionner les outils et module de notre dispositif de localisation. Nous commencerons par une description du réseau GPS suivie par une étude du réseau GSM.

## 2.2- Système GPS : [1] [2] [3]

Le GPS (Global Positioning System) est un système de radionavigation par satellites qui permet aux utilisateurs de déterminer leurs positions que ce soit sur la terre, en mer ou dans les airs à l'aide d'un récepteur électronique.

Il a été conçu pour des raisons militaires mais maintenant il est couramment utilisé dans des applications civiles telles que le transport, l'agriculture, opérations de sauvetage et des autres applications ou services qui ne cessent de voir le jour.

### 2.2.1- Description du système : [4]

Le GPS se compose de trois secteurs :

- Secteur spatial.
- Secteur de contrôle
- Secteur utilisateur

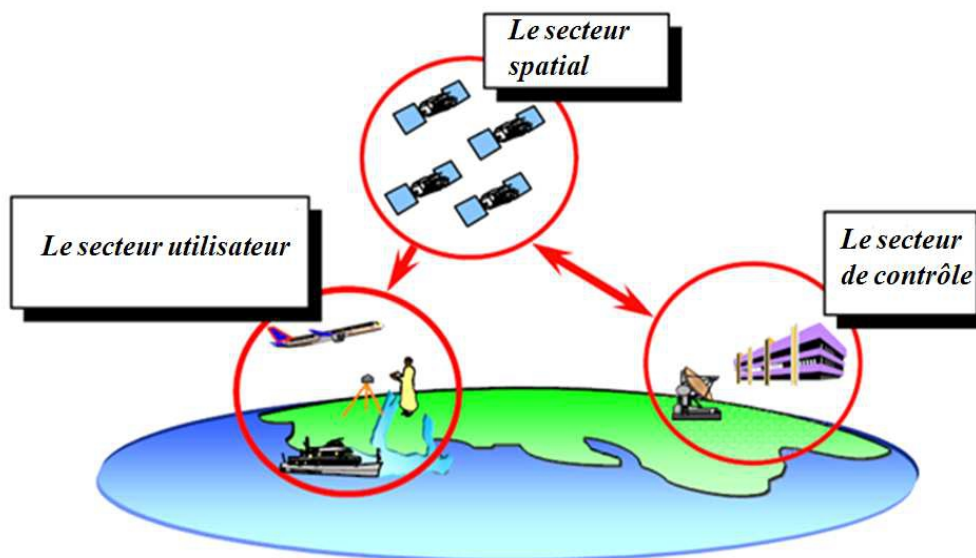


Fig.2.1: Secteurs du GPS

### A- Secteur spatial :

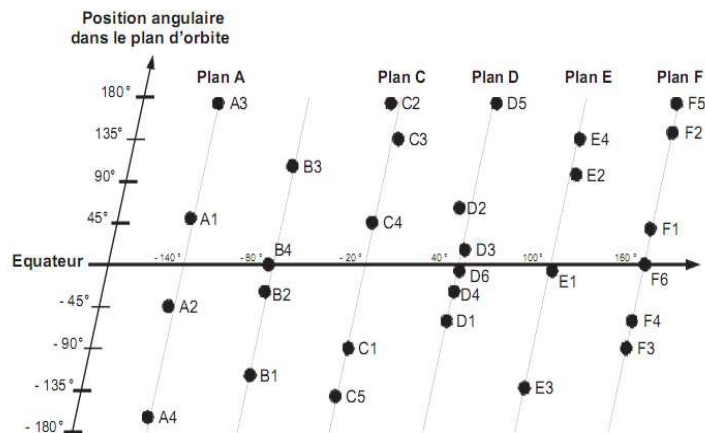
Le secteur spatial permet de transmettre les signaux, lesquels donnent le positionnement et l'heure du réseau GPS. Il est constitué de constellation nominale de 24 satellites opérationnels.



**Fig2.2: Satellite GPS**

#### a) Constellation :

Le secteur spatial contient 24 satellites qui tournent dans des orbites MEO à une altitude de 20200 km et qu'ils parcourent le tour de la terre en 11 heures 58 minutes 02 secondes à une vitesse d'environ 4 km/s, soit 14 000km/h. L'augmentation de nombre de satellites qui a atteint 28 satellites en 2003 et 31 actuellement réparties sur 6 plans.



**Fig2.3: Répartition des satellites GPS mai 2006**



### b) Caractéristique des satellites :

Un satellite GPS est constitué d'un émetteur et récepteur radioélectrique, d'horloge atomique, de calculateurs, d'un système de propulsion auxiliaire et des panneaux solaires lui fournissant l'énergie.

### B- Secteur de contrôle : [1]

C'est le système responsable de contrôle des satellites son rôle est de calculer les données diffusées aux utilisateurs et d'opérer l'ensemble des composants. Il contient plusieurs éléments notamment les suivant :

- La station de control principal, ou MCS (Master Control Center), utilisé au sein d'une base de l'US Air Force dans Colorado.
- Les stations de surveillance, ou MS (Monitor Station), qui effectuent des mesures sur les signaux diffusés par les satellites.
- Les antennes sol (Ground Antennas) qui permettes de communiquer avec les satellites



**Figure2.4: répartition des secteurs de contrôle GPS**

### C- Secteur Utilisateur : [5]

C'est le système permettant d'accéder aux services GPS par les utilisateurs, il a deux types de services :

- SPS (Standard Positioning Service) : un service libre dédié aux applications civiles utilisant un récepteur mono fréquence L1 : 1575.42 MHz.

- PPS (Precise Positioning Service) le service précis utilisant aux applications militaires, il utilise deux fréquences L1 (1575,42 MHz) et L2 (1221,60 MHz), ainsi que sur le code P(Y) qui est modulé sur ces deux fréquences.

### **2.3- Sources d'erreurs : [6]**

Liées aux phénomènes physiques et facteurs techniques qui ont un effet sur la performance et le bon fonctionnement. Il y a plusieurs types de sources d'erreurs :

#### **2.3-1. Erreurs d'origine naturelle :**

Dues aux phénomènes naturelle notamment la température, la pression et l'humidité de l'air ainsi qu'aux couches de l'atmosphère (Troposphère, Ionosphère) qui introduisent des retard sur les signaux GPS.

Les obstacles (Immeubles ou montagnes) causent des trajets multiples à travers des réflexions, réfraction et diffraction des signaux, ils perturbent les signaux se qui cause l'estimation de positions erronées.

#### **2.3-2. Erreurs d'origine technique :**

Ces erreurs viennent de la précision des satellites dues à la géométrie des satellites nous appelons ça la diminution de précision. Nous trouvons aussi la validité sélective SA, c'est l'erreur introduit volontairement par le département de la défense des États-Unis pour des raisons de sécurité, qui a été arrêté en 2000.

Il y a d'autre source d'erreur : le nombre de visible des satellites, le mauvais fonctionnement des composants, le bruit de récepteur.

## 2.4- Correction d'erreur :

Cependant la précision peut être améliorée en combinant le récepteur GPS à un récepteur GPS différentiel (DGPS), qui peut fonctionner à partir de plusieurs sources possibles pour aider à réduire certaines erreurs.

Source d'erreurs	GPS	DGPS
Ionosphère	10 m	-
Troposphère	1 m	-
Multi trajets	0.5 m	0.5
Bruit du code	3 m	1 m
Bruit du récepteur	3 m	1 m
Horloge de bord	5 m	-
éphémérides	5 m	-
SA	32 m	-
DOP	69 m	3 m
Total	128.5 m	5.5 m

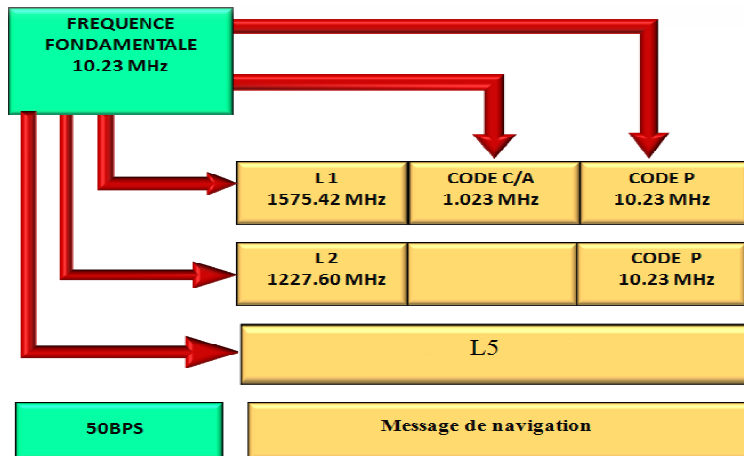
**Tableau2.1: Les erreurs et leurs corrections**

## 2.5- Signal GPS : [4]

Les récepteurs GPS reçoivent deux types de codes selon le service utilisé : un code pour les services militaires américains (signaux PPS) et un code civil.

Chaque satellite génère deux ondes porteuses utilisant la même fréquence fondamentale  $f_0=10.23$  MHz (figure.2.4).

- $L1=154.f_0=1575,42$  MHz (soit une longueur d'onde d'environ 19 cm)
- $L2=120.f_0=1227,60$  MHz (soit une longueur d'onde d'environ 24 cm)

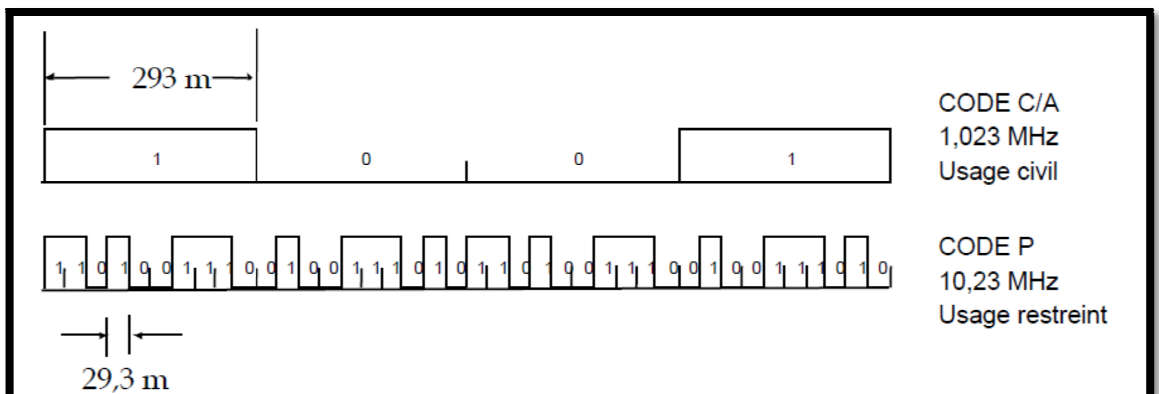


**Fig.2.5: Génération du signal**

**2.5-1. Code pseudo aléatoire :**

Les fréquences émettant par les satellites GPS sont modulés par des codes pseudo-aléatoires qui sont les suivants (figure.2.5) :

- Le code C/A (Code Acquisition -code) disponible sur les signaux L1 pour les applications civiles.
- Le code P : disponible sur les deux signaux L1 et L2 pour les applications militaires



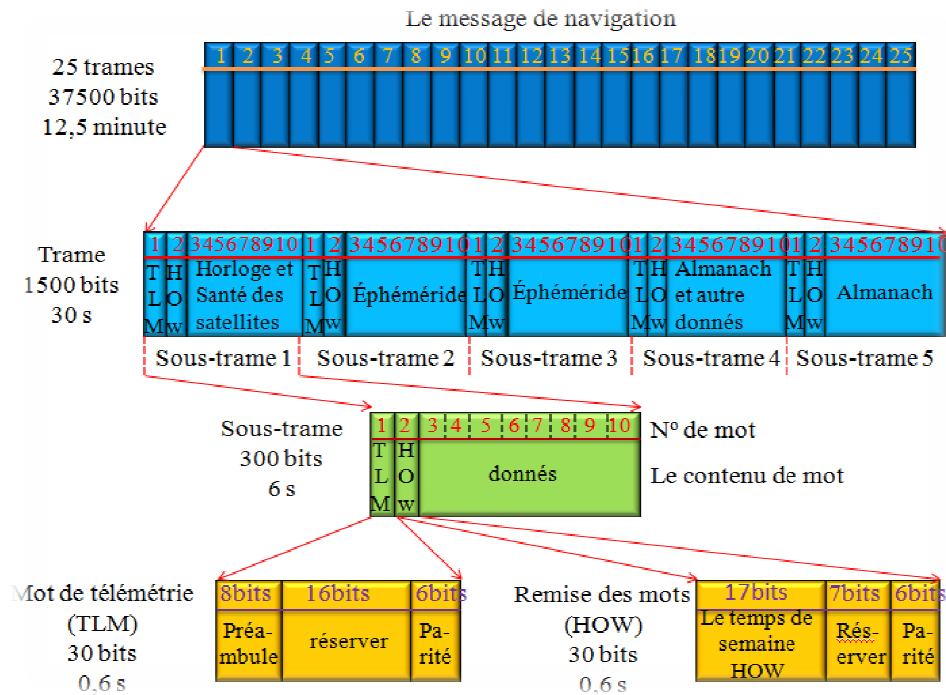
**Fig2.6: Code C/A et P[5]**

**2.5.2- Contenu du Signal :**

Le signal GPS se compose de trames de 1500 bits, chaque trame contient 10 mots de 30 bits (figure.2.6). Le débit du message est de 50 bits/sec, une trame est donc émise en 30 secondes.

Les sous trames 1,2 et 3 contiennent des informations (messages) qui ne changent pas et indispensables à la navigation et elles seront répétées chaque 30 secondes.

Les sous trames 4 et 5 contiennent 25 messages différents, elles sont réservées pour les messages les plus longs et les plus importants elles durent 12,5 minutes.



**Fig2.7: Contenu de message de navigation**

Almanach : données diffusées dans les signaux GPS permettant aux récepteurs de déterminer les positions approximatives à tout instant de tous les satellites de la constellation. Son rôle est de réduire le temps de démarrage des récepteurs.

Éphémérides : donnent les positions du satellite avec une précision de 1 km. Chaque satellite diffuse les éphémérides qui les concernent.

TLM (Télémetry Word) : mot contient 30 bit : un profil de 8 bits servant de préambule, une suite de 14 bits adressés par le secteur de control concernant le chargement des messages, 2 bits réservés suivis d'une parité calculée sur 6 bits.

HOW (Hand Over Word) : mot de 30 bits répété tous les six secondes. Il contient les bits les plus significatifs du comptage hebdomadaire, ces bits donnent l'heure et le jour et permettent de pré positionner les registres générateurs du code P, les bits suivants sont utilisés comme drapeaux pour signifier l'état du satellite.

## **2.6 Principe de fonctionnement : [8]**

Le principe du positionnement GPS est très similaire du principe de triangulation. On mesure la distance entre l'utilisateur et un certain nombre de satellites de positions connues. Pour cela, le récepteur mesure la différence de temps entre l'émission et la réception du signal.

Le signal voyageant pratiquement à la vitesse de la lumière, on peut appliquer la relation :

$$D = c * t.$$

Avec :

- D : distance entre le récepteur et le satellite (c'est en fait le rayon d'un cercle où le satellite serait le centre).
- c : vitesse de propagation des signaux radios (la vitesse retenue par convention pour le système GPS est  $c = 299\,792\,458$  m/s),
- t : durée de signal entre l'émission et la réception.

On obtient donc la distance entre le récepteur GPS et les satellites.

### **2.6.1 Calcul de la position :**

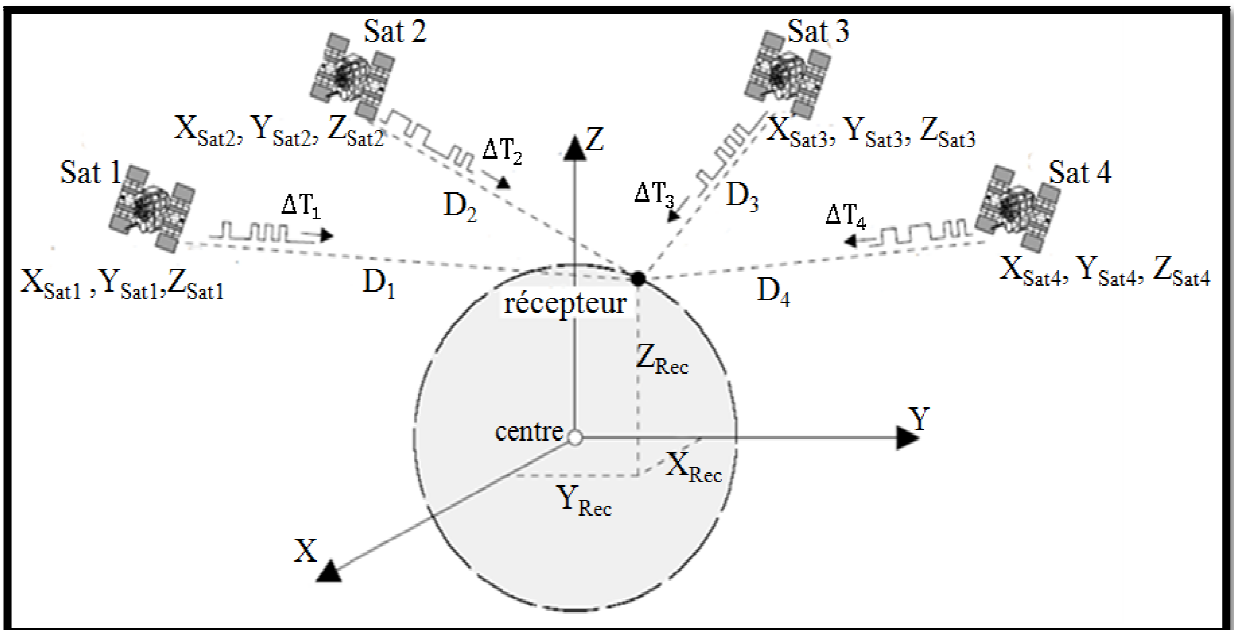
La position de l'observateur est donnée par le vecteur :

$$\overrightarrow{OA} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

L'équation fondamentale pour mesurer la distance entre le satellite et le récepteur:

$$P = \sqrt{(X_s - X)^2 + (Y_s - Y)^2 + (Z_s - Z)^2}$$

- distances  $P$  entre le satellite le récepteur.
- $(X, Y, Z)$  coordonnées de l'utilisateur.
- $(X_s, Y_s, Z_s)$  celles des satellites .



**Fig2.8: Différents paramètres pour calculer la position**

Pour quatre satellites c'est -à -dire quatre mesures, les équations de navigation du système GPS sont dans un repère cartésien :

$$P_1 = \sqrt{(X_{S1} - X)^2 + (Y_{S1} - Y)^2 + (Z_{S1} - Z)^2}$$

$$P_2 = \sqrt{(X_{S2} - X)^2 + (Y_{S2} - Y)^2 + (Z_{S2} - Z)^2}$$

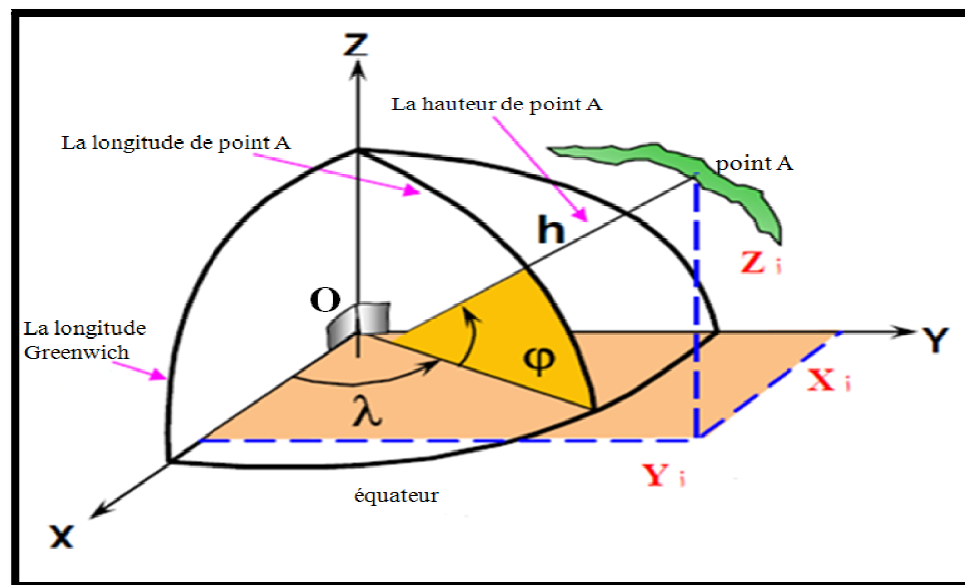
$$P_3 = \sqrt{(X_{S3} - X)^2 + (Y_{S3} - Y)^2 + (Z_{S3} - Z)^2}$$

$$P_4 = \sqrt{(X_{S4} - X)^2 + (Y_{S4} - Y)^2 + (Z_{S4} - Z)^2}$$

### 2.6.2 Conversion des coordonnées :

Les satellites émettent des informations permettant de calculer leur position à chaque instant. Puis le récepteur GPS convertie ces données en (X, Y, Z) en donnée NMEA c'est-à-dire en latitude, longitude et altitude.

Le récepteur GPS qui effectue cette conversion grâce au système géodésique WGS 84 (World Geodesic System 1984), le système le plus utilisé au monde, ou alors à l'aide d'un autre système géodésique. Cependant, cette conversion n'est pas aussi simple et peut être à la source de nombreuses erreurs dues à une mauvaise programmation.



**Fig.2.9: Position en Longitude, latitude et hauteur**

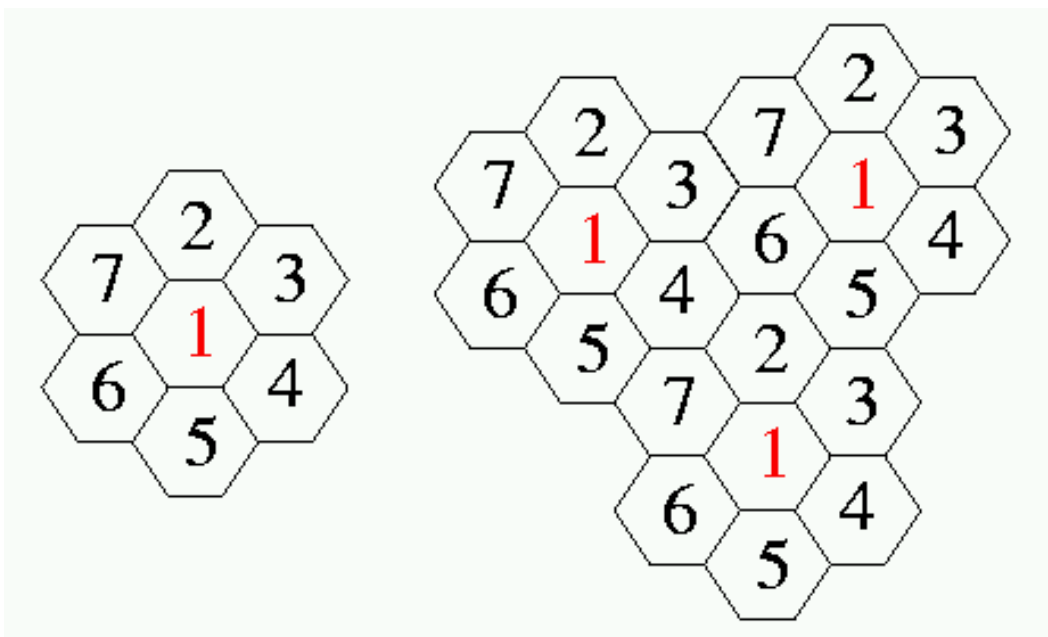
### 2.7- GSM : [9]

Le GSM (Global System For Mobile Communication) est un système de communication mobile cellulaire. Il utilise une liaison radioélectrique entre le terminale et le réseau, selon les systèmes, plusieurs bande de fréquence sont utilisées : parmi principalement, on compte les bandes 900MHz et 1800MHz.



### 2.7.1- Le concept cellulaire [10]

Ce concept consiste à diviser un territoire en cellules dont chacune est couverte par une station radio ou station de base (BTS) du réseau. Et ainsi la réutilisation d'une même fréquence que celle des cellules différentes, c'est-à-dire qui sont adjacentes ou sécantes afin d'éviter les phénomènes d'interférences sur le signal utile reçu par le terminal mobile pour la station de base.

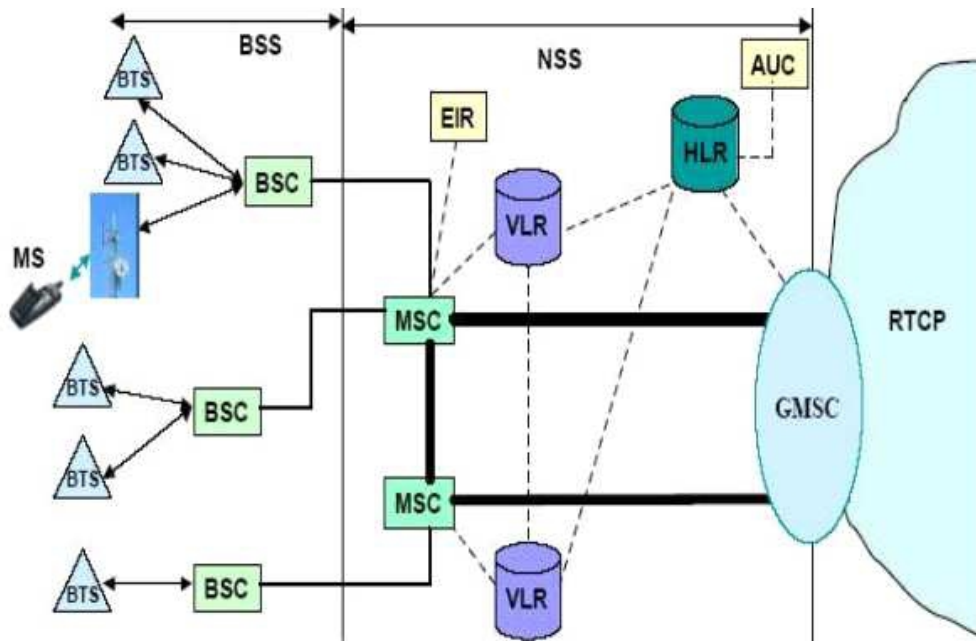


**Fig.2.10: motif du réseau GSM**

### 2.7.2- Architecture GSM : [10]

Le réseau GSM se compose de trois sous systèmes

- Sous-systèmes radio (BSS)
- Sous-système réseau (NSS)
- Sous-système d'exploitation (OSS)



**Fig2.11: Architecture GSM**

#### **A- Mobile station:**

Elle est composée d'un terminal (Mobile equipment) et une carte SIM qui sert à identifier l'abonné indépendamment de terminal utilisé.

#### **B- Sous-système radio BSS (Base Station Sub-System) :**

Son rôle est de gérer l'attribution des ressources radio, indépendamment des abonnés, de leur identité ou de leur communication. Nous distinguons dans le BSS :

- BTS : (Base Transceiver Station) qui contient des émetteurs récepteurs appelés TRX reçoit les informations sur le canal radio.
- BSC : (Base Station Controller) elle gère une ou plusieurs BTS.

#### **C- Sous-système réseau NSS (Network Station Sub-System) :**

Il gère les fonctions de commutation et de routage, il permet d'accéder aux autres réseaux RTCP et RNIS, il prend en charge les fonctions d'analyse et de contrôle d'informations et aussi l'authentification, le chiffrement et le roaming. Il est constitué de :

- MSC : Mobile Switching Center
- HLR : Home Location Register / AUC : Authentication Center
- VLR : Visitor Location Register
- EIR : Equipement Identify Register
- 

#### **D- Sous-système d'exploitation OSS (Operating Station Sub-System) :**

Il assure la gestion et la supervision du réseau. C'est la fonction dont l'implémentation est laissée avec le plus de liberté dans la norme GSM. La supervision du réseau intervient à de nombreux niveaux :

- Détection de pannes.
- Mise en service de sites.
- Modification de paramétrage.
- Réalisation de statistiques.

#### **2.7.3- GPRS : [11]**

Le GPRS (General Packet Radio Service) il s'agit d'une norme de téléphonie de seconde génération permettant de faire la transition vers la troisième génération (3G), on parle généralement de la 2.5G pour classer le standard GPRS.

Le GPRS permet d'étendre l'architecture du standard GSM, afin d'autoriser le transfert de données par paquets, avec des débits théoriques maximums de l'ordre de 171,2 kbit/s (en pratique jusqu'à 114 kbit/s).

**2.7.4- EDGE : [11]**

La norme EDGE (Enhanced Data for GSM Evolution) est une évolution de la norme GSM, modifiant le type de modulation. Tout comme la norme GPRS, le standard EDGE est utilisé comme transition vers la troisième génération de téléphonie mobile (3G). On parle ainsi de la 2.75G pour désigner le standard EDGE.

L'EDGE permet ainsi de multiplier par un facteur 3 le débit des données avec une couverture plus réduite. Dans la théorie EDGE permet d'atteindre des débits allant jusqu'à 384 kbit/s pour les stations fixes (piétons et véhicules lents) et jusqu'à 144 kbit/s pour les stations mobiles (véhicules rapides).

**2.8- Conclusion**

Le système GPS offre des services de positionnement et de localisation. Le réseau GSM permet d'établir des communications et transfert de données indépendamment de la position. Combinée les deux systèmes permet la création d'un nouveau dispositif et service de localisation à distance et télésurveillance.

# Chapitre 3

## La sonde GM862-GPS

### **3.1- Introduction :**

Les modules GSM/GPS sont utilisés dans les applications mobiles. Nous distinguons plusieurs services offerts par ses modules : la géolocalisation, la télématique, gestion de flottes, le suivi, la sécurité et la navigation des véhicules et des gens. Il y a plusieurs marques des modules : SIM902 de Simcom, WISMO0228 de Sierra Wireless, GM862-GPS de Telit.

### **3.2- Définition:[16]**

Dans notre projet on a utilisé le module GM862-GPS, ce système compose d'un récepteur GPS et un modem GSM.

Ce module permet de communiquer sur les réseaux de téléphonie cellulaire et supporte les protocoles d'envoi de SMS, Fax, accès à internet (GPRS, EDGE...), la voix et la vidéo - grâce a un modem GSM. Il est capable de déterminer sa position grâce à un récepteur GPS. Il est piloté, manuellement depuis un ordinateur via un HyperTerminal, à l'aide de commandes dites AT, ou automatiquement avec un programme embarqué dans le module ou à travers un ordinateur de bord.

### **3.3- Caractéristique :[16]**

#### **3.3.1- caractéristique du module :**

Il est réalisé sur une dimension: 43.9 x 43.9 mm le GM862-GPS contient 13 ports entrée/sortie en plus des ports de communications série. Il possède les caractéristiques suivantes :

- GSM Quad Band : Supporte les fréquences 850/900/1800/1900MHz.
- Poids : 19 gr
- 17mA en veille, 3.5mA en mode d'Economie d'Energie.

- Lecteur carte SIM intégré.
- Pile TCP/IP Intégrée.
- FTP et SMTP client Intégré.
- Interpréteur de Script PYTHON.



**Fig3.1: Le module GM862-GPS**

### 3.3.2- Caractéristiques de l'antenne GPS :

Le module ne contient pas d'antenne interne, il a besoin d'une antenne GPS externe pour recevoir les signaux émis par les satellites GPS, donc il doit être muni d'une antenne GPS avec les caractéristiques suivantes :

- Alimentation : 55mA, y compris 35mA GPS pour le matériel GPS et 20 mA pour l'antenne LNA.
- 20 canaux GPS utilisant la fréquence L1 1575.42 MHz.
- Précision < 2,5 m
- GPS NMEA 0183 format de sortie
- Faible consommation d'énergie
- Date de WGS-84

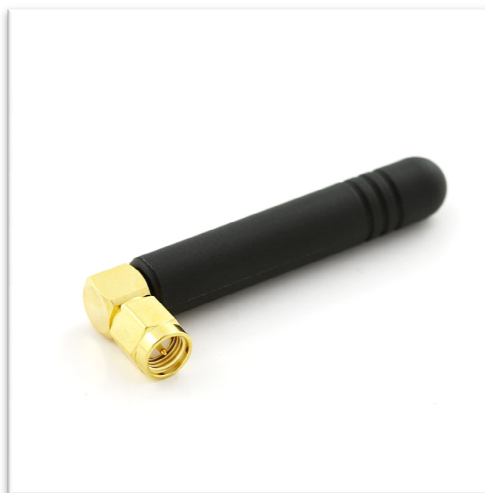


**Fig3.2: Antenne GPS pour GM-862-GPS**

### 3.3.3- Caractéristique de l'antenne GSM :

Le module a également besoin d'une antenne GSM pour pouvoir communiquer. Cette antenne possède les caractéristiques suivantes :

- Bande passante : 80 MHz dans EGSM 900, 70 MHz si GSM 850, 170 MHz dans DCS 140
- Gain : Gain <3 dB
- Puissance d'entrée : 2 W
- Impédance d'entrée : 50  $\Omega$



**Fig3.3: Antenne GSM pour GM862-GPS**



### **3.4- Configuration et programmation du module :**

La configuration du modem GM862-GPS se base sur des commandes se nomment les commandes AT qui seront programmées dans un script Python pour l'utilisation mobile du modem.

#### **3.4.1- Les commandes AT:[17] [18]**

Les commandes AT sont des chaînes de caractères en code ASCII utilisées pour contrôler les modems et activer ses fonctionnalités. Les applications qui commandent les modems génèrent les plus souvent ces commandes AT, mais nous pouvons également contrôler un modem à l'aide d'un programme de communication dans le terminal.

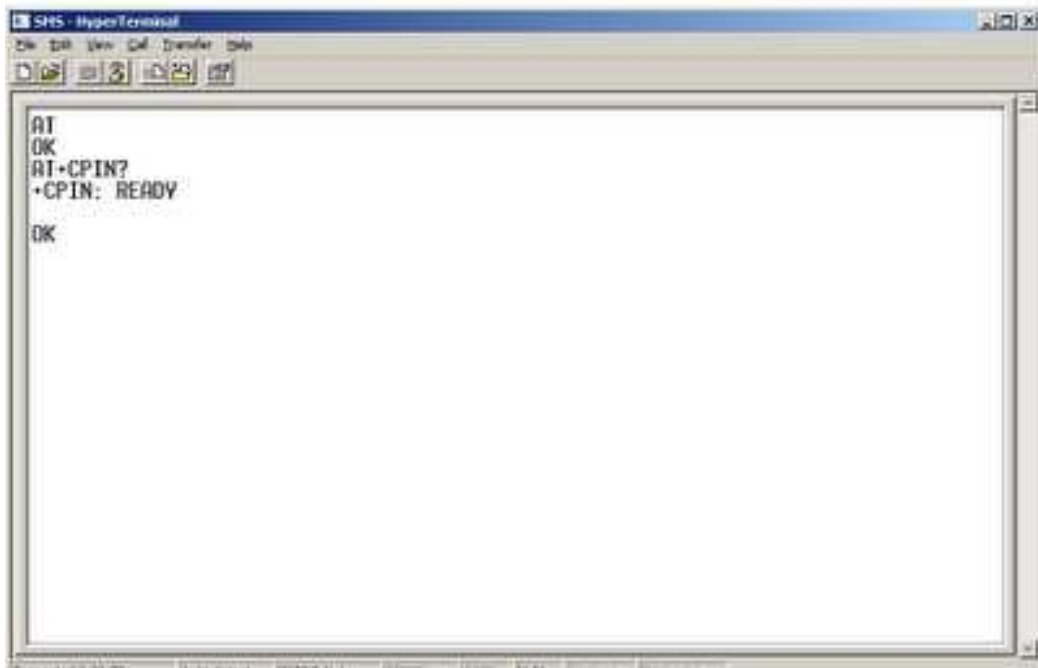
Parmi les rôles de ces commandes nous citons :

- Configuration des modems
- Affichage des paramètres du modem
- Diagnostiquer le réseau de téléphonie
- Exploitation des ressources de communication : téléphonie, SMS, GPRS, EDGE, 3G...

#### **3.4.2- configuration du module**

##### **A- Préparation de l'environnement :**

Dans la configuration du modem GM862- GPS nous avons utilisé le programme Hyperterminal qui a été développé localement au sein du laboratoire de télécommunications LTT. Ce petit logiciel nous permet de dialoguer avec des Modems via des commandes AT en exploitant le port USB émulé par des ports série COM.



**Fig3.4: Interface de HypeTerminal**

Pour pouvoir utiliser HyperTerminal nous devons configurer certains paramètres. Les étapes de configuration sont les suivantes :

- Étape 1 : nous devons lancer Hypertterminal et sélectionné le port utilisé
- Étape 2 : dans cette étape nous allons définir les paramètres de connexion entre le PC et le modem :
  - Le nombre de bits par seconde (9600)
  - Bits de donnés : 8
  - Parité : aucune
  - Bits d'arrêt : 1
  - Contrôle de flux : aucun

Après l'introduction des paramètres de connexion et son activation, nous confirmons la communication avec le modem en envoyant la commande **AT**, si la réponse est **OK** la communication se déroule normalement.

### **B- Configuration du modem:[18] [19]**

A travers HyperTerminal le GM862-GPS est prêt à recevoir les commandes AT et les exécuter.

- Code PIN de la carte SIM : pour entrer le code nous tapons la commande :

**AT+CPIN= "code PIN"**

- Test de réception : envoyer :

**AT+CREG ?**

Si la réponse est

+GREG: 0,2 : module cherche du réseau.

+GREC: 0,1 : module connecté au réseau.

- Configuration du mode de fonctionnement GPS :

**AT\$GPSPS=1**

Avec :

0 : puissance maximale

1 : économie d'énergie

2 : GPS éteint jusqu'à ce qu'une position soit demandée. Le GPS collecte périodiquement les données éphémérides pour pouvoir fournir une position rapidement.

- Utilisation embarquée : nous devons activer l'exécution automatique du script Python.

### **AT#STARTMODESCR=1.10**

La première valeur correspond au mode de départ choisi. La deuxième valeur correspond au délai d'exécution du script en secondes.

Les valeurs possibles sont :

0 : le script s'exécute en l'absence de signal DTR (module non connecté à un port COM).

1 : le script s'exécute si aucune commande AT n'est émise dans le délai spécifié.

2 : le script s'exécute immédiatement (non recommandé)

### **3.4.3- Programmation du module:[19]**

#### **A- Langage Python :**

Le GM862 va être programmé par un script en utilisant le langage de programmation Python.

Python est un langage de programmation orienté objet utilisé particulièrement comme langage de script pour automatiser des tâches complexes, comme une tâche chargée de récupérer des coordonnées GPS et les associer avec une carte géographique.

Le module à notre disposition utilise la version Python 1.5.2+ et seules les bibliothèques spécifiques au GM862 peuvent être importées dans le script. Nous devons utiliser une syntaxe compatible pour nos programmes. La taille du programme est limitée à 16kb, ce qui est finalement largement suffisant pour réaliser un système avec plusieurs fonctionnalités.

**B- Le script Python :**

- Les librairies

Il existe plusieurs librairies permettant de développer des scripts python,

Les principales librairies que nous utiliserons sont :

- **MDM** : permet au script python d'envoyer des commandes AT au module.
- **GPS** : Permet de contrôler le récepteur GPS
- **GPIO** : Interface de contrôle des GPIOs (General Purpose Input Output – Entrées sorties numériques)
- **MOD** : Permet d'utiliser certaines fonctions comme les compteurs.

La première chose à faire est d'importer dans notre script les librairies que nous allons utiliser :

```
import MDM
```

```
import GPIO
```

```
import MOD
```

```
import GPS
```

- Code PIN : quand le module s'allume, il attendra son code PIN avant de pouvoir accéder au réseau :

```
MDM.send('AT+CPIN="0000"\r',0)
```

Avec : le chiffre après la virgule correspond au délai d'exécution de la commande

- Récupération des coordonnées GPS: demander une position GPS s'effectue avec la syntaxe suivante :

**GPS.getActualPosition()**

La valeur retournée est une chaîne de caractère avec toutes les données du GPS séparées par des virgules. Si le GPS manque de données, Python retournera des virgules vides. Dans notre exemple il pourrait retourner cette chaîne : 202415.000,,0,,,,,230911,0. Ces données appelées Les trames NMEA.

- Définir le mode de fonctionnement du GPS : Il est possible de changer régulièrement le mode du GPS avec la commande :

**GPS.powerSavingMode(2,600)**

Le deuxième paramètre correspond au taux de rafraîchissement des données en secondes (utile uniquement si le mode 2 est sélectionné).

Envoi les données sur internet : ce script Python nous permet d'envoyer les données reçues sur internet (GPRS) et les traiter dans une base de données MySql afin de les afficher sur une carte (Google Maps)

Nous avons besoin de connaître quelques informations données par l'opérateur : son APN (Access Point Name), le nom d'utilisateur et le mot de passe. Dans notre cas la puce utilisée est celle de l'opérateur nedjma avec les paramètres :

APN : nedjma.dz

Nom d'utilisateur : nedjma

Mot de passe : nedjma

Paramètres de connexion : Nous pouvons maintenant paramétrer une connexion GPRS, à travers le réseau de l'opérateur Nedjma, en envoyant la commande AT suivante :

```
MDM.send('AT+CGDCONT=<cid> , " <pdp-type> " , "<apn> " , " <pdp-addr> " , <d-comp> , <h-comp> \r' , 0)
```

Avec :

<cid> : contexte du GPRS (1 à 5)

<pdp-type> le prorocole utilisé

<apn> nom de point d'accès au GPRS

<pdp-addr> l'adresse demandée, (0.0.0.0) adressage dynamique

<d-comp> paramètre de contrôle de données 0 par défaut

<h-comp> paramètre de contrôle d'en-tête 0 par défaut

Dans notre cas le script Python et le suivent :

```
MDM.send('AT+CGDCONT=1,"IP","nedjma.dz","0.0.0.0",0,0\r', 0)
```

Nous envoyons le nom et mot de passe de l'utilisateur par la commande :

```
MDM.send('AT#USERID="nedjma"\r',0)
```

```
MDM.send('AT#PASSW="nedjma"\r', 0)
```

Après nous paramétrons les tailles des paquets et le socket : la première commande définit la taille de paquet et la 2eme le temps du socket

```
MDM.send('AT#PKTSZ=300\r',0)
```

```
MDM.send('AT#DSTO=50\r', 0)
```

Maintenant activer la connexion GPRS :

```
MDM.send('AT#GPRS=1\r', 0)
```

Cette commande retournée l'adresse IP assignée par le réseau, sous la forme (exemple) : +IP: 10.32.87.56 Cette IP est cependant temporaire.

Atteindre un page web : nous utilisant la méthode POST :

```
MDM.send('POST/mes.php HTTP/1.1\r\n',0)
```

```
MDM.send('Host: localisation.netii.net\r\n', 0)
```

Nous pouvons également envoyer les données à une page web en utilisant la méthode GET

```
MDM.send('GET /mes.php?longitude=valeur&latitude=valeur2  
HTTP/1.1\r\n', 0)
```

```
MDM.send('Host:localization.netii.net \r\n', 0)
```

Nous pouvons juste atteindre un page sans envoyer des données en utilisant la méthode GET :

```
MDM.send('GET / HTTP/1.1\r\n', 0)
```

```
MDM.send('Host: monsite.fr\r\n', 0)
```

Nous pouvons maintenant **fermer la connexion GPRS** :

```
MDM.send('AT#GPRS=0\r', 0)
```

Pour recevoir les données d'une page web nous introduisons la commande

MDM.receive, en utilisant un délai de réponse assez important (par exemple 10 secondes) : réponse = **MDM.receive(100)** La réponse sera codée sous le format html.

Nous téléchargerons ce programme sur le module, les données GPS seront enregistrées dans la base de données et affichés sur une carte

### **C- Les trames NMEA :[20][21]**

Un récepteur GPS trouve sa position grâce à la réception de signaux émis par des satellites. Le récepteur GPS détermine par calcul sa position et la transmet sous forme d'une trame NMEA.

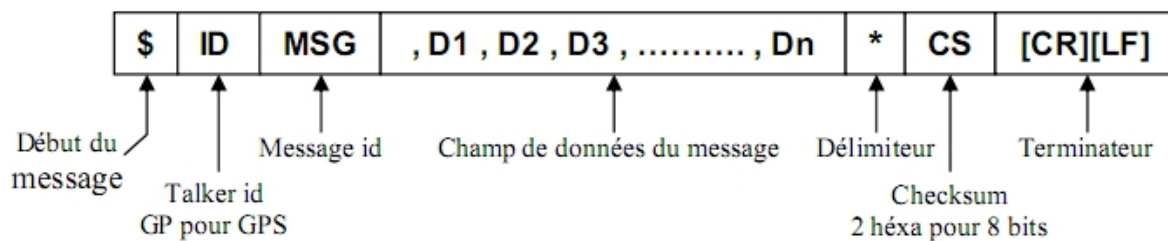


**Définition :**

Le protocole NMEA-0183 (National Marine Electronic Association) est un langage employé par le récepteur GPS pour discuter avec son environnement (ordinateur, autre GPS, traceur de route, etc.).

Chaque trame NMEA se compose d'une séquence de lettres et de chiffres. Elle se compose :

- D'un préfixe (un symbole, le \$ en principe),
- D'un en-tête de longueur fixe (5 lettres) qui indique au récepteur (un microprocesseur par exemple) le sens des informations qui le suivent,
- D'un texte de longueur variable

**Forma d'une trame NMEA :****Fig3.5: Format d'une trame NMEA**

Il y a plusieurs type de donnée NMEA telle que :

\$GPRMB = Information minimales de navigation

\$GPRMC = Données spécifiques minimum de travail

\$GPVTG = Piste (track) bonne sur le sol

\$GPGGA = Données globales du système de positionnement

La séquence la plus importante pour connaître sa position est celle qui débute par: **\$GPSACP**.

**Format d'une séquence \$GPSACP :**

**\$GPSACP:<UTC>,<latitude>,<longitude>,<hdop>,<altitude>,<fix>,<cog>,<spkm>,<spkn>,<date>,<nsat>**

Exemple d'une séquence reçue :

\$GPSACP:094624.000,3455.1614N,00118.1186W,1.2,605.4,3,7.65,0.03,0.01,270213,07

Nom	Valeur	Format	Description
<b>Message ID</b>	\$GPSACP		
<b>UTC</b>	094624.000	hhmmss	hh (heure) 0 à 24 mm(minute) 0 à 59 ss(seconde)0 à 59
<b>Latitude</b>	3455.1614N	ddmm.mmmmN/S	dd(degré) 0 à 90 mm.mmmm(minute) 0 à 59.9999 N= nord S= sud
<b>Longitude</b>	00118.1186W	dddmm.mmmmW/E	ddd(degré) 0 à 180 mm.mmmm(minute) 0 à 59.9999 W=ouest E=est
<b>Hdop</b>	1.2		
<b>altitude</b>	605.4		Mètre
<b>Fix</b>	3		
<b>Cog</b>	7.65	ddd.mm	ddd(degré) 0 à 360 mm.mmmm(minute) 0 à 59

<b>Spkm</b>	0.03		Vitesse (Km/h)
<b>Spkn</b>	0.01		Vitesse (nœud)
<b>Date</b>	270213	ddmmyy	dd jour mm mois yy année
<b>Nsat</b>	07		Nombre satellite visible

**Tableau3.1: Les champs de la trame GPSACP**

Nous allons avoir besoin d'extraire les données qui nous intéressent. L'interpréteur Python autorise certaines opérations sur les chaînes de caractères (mais pas toutes, la librairie string ne peut pas être importée). Nous allons créer une fonction qui se chargera de faire une recherche sur les virgules, et nous allons retenir notre paramètre.

### Code Python :

```
def positionGPS(position):
    index1 = position.find(',')
    index2 = position.find(',',index1+1)
    index3 = position.find(',',index2+1)
    index4 = position.find(',',index3+1)
    index5 = position.find(',',index4+1)
    index6 = position.find(',',index5+1)
    index7 = position.find(',',index6+1)
    index8 = position.find(',',index7+1)
    index9 = position.find(',',index8+1)
    index10 = position.find(',',index9+1)
    lat = position[index1+1:index2]
    lon = position[index2+1:index3]
    heure = position[0:5]
    d = position[index9+1:index9+7]
```

Le code suivant nous permet alors d'obtenir nos données :

```
pos = GPS.getActualPosition()  
MOD.sleep(2)  
pos = positionGPS(pos)
```

Si tous les paramètres sont séparés on peut maintenant l'envoyer comme nous avons déjà indiqué.

### **3.5- Conclusion :**

Le développement des systèmes de géo-localisation ne cessent de voir le jour, l'utilisation de ces différents systèmes (GSM, GSP) grâce aux systèmes embarqués nous a permis d'innover avec la création des nouveaux services de localisations qui répondent aux besoins économiques et sociaux de la vie quotidienne. En effet, nous avons dans notre cas exploité le module GM862-GPS combiné avec plusieurs techniques et protocoles pour dépêcher les informations sur une position vers un serveur web. Dans le prochain chapitre nous allons développer une telle plateforme.

# Chapitre4

## La station de surveillance

**4-1-Introduction :**

Pour la station de surveillance nous avons développé un portail web responsable de la gestion des bases de données , de la localisation et de la gestion des utilisateurs.

Deux bases de données MySQL, accessible à travers le protocole TCP/IP, ont été implémentées :

- la première dédiée à la récolte des informations de localisation émises par les sondes
- la seconde dédiée à la gestion des utilisateurs du portail avec deux privilèges ; administrateur et client de visualisation.

Des scripts PHP et JavaScript ont été implémentés pour traduire les coordonnées NMEA envoyés par les sondes sur des cartes GOOGLE MAPS, GOOGLE EARTH. En effet, voir simultanément l'évolution des positions de plusieurs engins (chaque engin avec une couleur), est très utile pour prendre un jugement ou une décision.

**4-2-Description de la station de surveillance :**

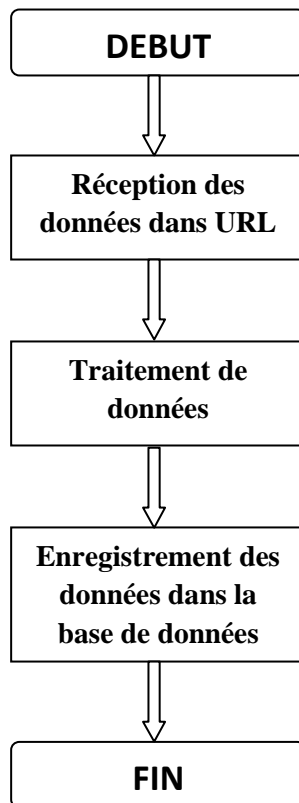
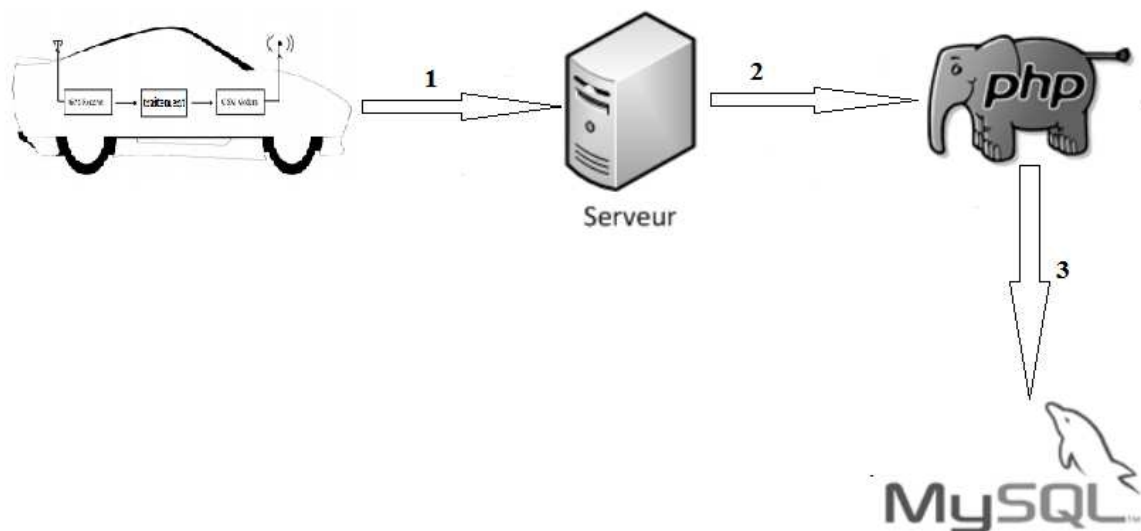
La station de surveillance se compose :

- D'un modem GSM.
- D'un serveur web.
- D'un langage PHP.
- D'un langage MySQL.
- L'API GoogleMaps.

**4.3- Le rôle de la station de surveillance :**

La station de surveillance fait en même temps la réception des données envoyées par les sondes et l'affichage de la position de ces données sur une carte à la demande des observateurs (clients).

Dans cette partie nous allons détailler les méthodes, les algorithmes et les structures de la base de données pour chaque rôle.

**4.3.1- la réception des données :****Fig4.1: Algorithme de la réception des données****Fig4.2: Description de la méthode de réception des données**

**A- Réception des données dans URL :[12]**

La station de surveillance est équipée par un modem GSM qui est utilisé pour la relier au réseau internet.

Les données de localisation seront envoyées à travers le réseau EDGE, actuellement en exploitation en Algérie, avec la possibilité d'extension vers le réseau mobile 3G.

**Transmission des données avec le réseau EDGE :**

Avec le réseau EDGE nous pouvons utiliser plusieurs méthodes pour la transmission de données à travers le réseau internet. Dans ce projet nous avons utilisé l'adresse URL pour la transmission de coordonnées des engins à cause de simplicité, efficacité, faible coût et très haute confidentialité de cette méthode.

Une URL représente l'adresse d'une page web (commençant généralement par http://).

Lorsqu'on nous faisons un lien vers une page, il est possible d'ajouter des paramètres sous la forme :

**<http://www.localisation.php?latitude=3412.1247S&longitude=00124.1587W>**  
qui sera transmis à la page.

– La page localisation.php dans l'exemple précédent recevra ces paramètres dans un array nommé \$\_GET :

– \$\_GET['latitude'] aura pour valeur «**3412.1247S**»;

– \$\_GET['longitude'] aura pour valeur «**00124.1587W**».

Cette technique est très pratique pour transmettre des valeurs à une page.

**B- Traitement des données :**

Le traitement des données se fait par le langage PHP mais pour que les sondes puissent communiquer avec la station pour faire ces opérations il faut qu'un logiciel d'un serveur web (Apache) doit être installé dans cette station.



**Serveur web:[13]**

Un serveur web est un logiciel permettant à des clients d'accéder à des pages web, c'est-à-dire en réalité des fichiers au format HTML à partir d'un navigateur (aussi appelé browser) installé sur leur ordinateur distant.

Un serveur web est donc un « simple » logiciel capable d'interpréter les requêtes HTTP arrivant sur le port associé au protocole HTTP (par défaut le port 80), et de fournir une réponse avec ce même protocole.

Les principaux serveurs web sur le marché sont entre autres :

- Apache
- Microsoft IIS (Internet Information Server)
- Microsoft PWS (Personal Web Server)
- Xitami
- ...

Dans notre cas nous avons utilisé un serveur Apache.

**Apache** est le serveur le plus répandu sur Internet. Il s'agit d'une application fonctionnant à la base sur les systèmes d'exploitation de type Unix, mais il a désormais été porté sur de nombreux systèmes, dont Microsoft Windows.



Apache (prononcez à la française ou bien pour les puristes à l'anglophone « Apatchy ») tire son nom de la façon dont il a été mis au point (« A patchy server » traduisez « un serveur rafistolé ») car il est le fruit d'une multitude de correctifs logiciels afin d'en faire une solution très sûre. En effet Apache est considéré comme sûr dans la mesure où peu de vulnérabilités le concernant sont publiées.

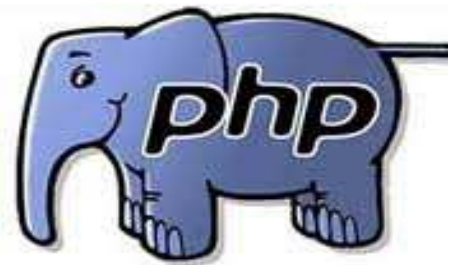
Ainsi, dès qu'un bug ou une faille de sécurité est décelée, ceux-ci sont rapidement corrigés et une nouvelle version de l'application est éditée.

Quand Apache sera installé sur la station, elle devient capable de recevoir à tous moments les données envoyées par les sondes, maintenant dans la station. Le traitement des données est fait par le langage PHP.

## Langages PHP :[14]

PHP est un langage de scripts généraliste et Open Source, spécialement conçu pour le développement d'applications web. Il peut être intégré facilement au HTML.

Au lieu d'utiliser des tonnes de commandes afin d'afficher du HTML (comme en C ou en Perl), les pages PHP contiennent des fragments HTML dont du code qui fait "quelque chose" . Le code PHP est inclus entre une balise de début `<?php` et une balise de fin `?>` qui permettent au serveur web de passer en mode PHP.



Ce qui distingue PHP des langages de script comme le Javascript, est que le code est exécuté sur le serveur, générant ainsi le HTML, qui sera ensuite envoyé au client. Le client ne reçoit que le résultat du script, sans aucun moyen d'avoir accès au code qui a produit ce résultat. Vous pouvez configurer votre serveur web afin qu'il analyse tous vos fichiers HTML comme des fichiers PHP. Ainsi, il n'y a aucun moyen de distinguer les pages qui sont produites dynamiquement des pages statiques.

Les données seront envoyées dans l'adresse URL de la station, le premier rôle de langage PHP et de séparer les différents champs dans l'adresse, cette opération est faite par la balise `$_GET`.

Exemple des données envoyées dans URL :

**[www.localisation.php?sonde=sonde1&time=094624&latitude=3455.1614N&longitude=00118.1186W&altitude=605.4&distance=7.65&vitesse=0.03&date=270213](http://www.localisation.php?sonde=sonde1&time=094624&latitude=3455.1614N&longitude=00118.1186W&altitude=605.4&distance=7.65&vitesse=0.03&date=270213)**

Le programme PHP qui récupère les données :

```
$sonde=$_GET['sonde'];
$date=$_GET['date'];
$time=$_GET['time'];
$latitude=$_GET['latitude'];
$longitude=$_GET['longitude'];
$altitude=$_GET['altitude'];
$distance=$_GET['distance'];
$vitesse=$_GET['vitesse'];
```

Valeur	Script PHP	Résultat
\$sonde	\$_GET['sonde ']	sonde1
\$date	\$_GET[' date ']	270213
\$time	\$_GET[' time']	09 :46 :24
\$latitude	\$_GET['latitude ']	3455.1614N
\$longitude	\$_GET[' longitude ']	00118.1186W
\$altitude	\$_GET[' altitude ']	605.4
\$distance	\$_GET['distance']	7.65
\$vitesse	\$_GET['vitesse']	0.03

**Tableau4.1 : les valeurs récupérées avec PHP**

Après la réception des données nous allons convertir quelques valeurs, cette opération est nécessaire soit pour que ces valeurs deviennent compatibles avec la base de données (le cas de la date) soit pour devenir compatibles avec GoogleMaps (cas de longitude et latitude).

Dans le cas de la date, la valeur envoyée par le récepteur GPS est de la forme ddmmyy qui n'est pas compatible avec celle de la base de données qui est de la forme yymmdd. Nous allons devoir convertir la date avant l'enregistrement.

Le script PHP qui fait la conversion :

```
<?PHP
$a=$date[5];
$b=$date[4];
$date[5]=$date[1];
$date[4]=$date[0];
$date[1]=$a;
$date[0]=$b;
?>
```

Le GPS utilise un système de coordonnées sexagésimal qui n'est pas directement compatible avec celui utilisé par Google Maps (décimal). Nous allons devoir convertir les coordonnées avant de les transmettre.

- Traitement des coordonnées avec PHP :

```
<?PHP
$altitude=$_GET['altitude'];
$distance=$_GET['distance'];
$vitesse=$_GET['vitesse'];
$deglat=$latitude[0].+$latitude[1];
$minlat=$latitude[2].+$latitude[3];
$seclat=$latitude[5].+$latitude[6];
$milseclat=$latitude[7].+$latitude[8];
$NSind=$latitude[9];
$minlat=$minlat/60;
$seclat=$seclat/3600;
$milseclat=$milseclat/3600000;
if ($NSind=='N'){
$latitude=$deglat+$minlat+$seclat+$milseclat;
}
else { $latitude=-($deglat+$minlat+$seclat+$milseclat);
}
$deglon=$longitude[0].+$longitude[1].+$longitude[2];
$minlon=$longitude[3].+$longitude[4];
$seclon=$longitude[6].+$longitude[7];
$milseclon=$longitude[8].+$longitude[9];
$EWind=$longitude[10];
$minlon=$minlon/60;
$seclon=$seclon/3600;
$milseclon=$milseclon/3600000;
if ($EWind=='E'){
$longitude=$deglon+$minlon+$seclon+$milseclon;
}
else { $longitude=-($deglon+$minlon+$seclon+$milseclon);
}
?>
```

### C- Enregistrement des données dans la base de données :

Après le traitement ces données seront enregistrées dans une base de données, dans cette étape on a utilisé un logiciel MySQL.

#### Langage MySQL :[12][15]

**MySQL** c'est ce qu'on appelle un SGBD (Système de Gestion de Base de Données). Pour faire simple, son rôle est d'enregistrer des données de manière organisée afin de nous aider à les retrouver facilement plus tard. C'est grâce à MySQL que nous pourrions enregistrer la liste des administrateurs et utilisateurs de notre site, les coordonnées des véhicules, etc. Le langage qui permet de communiquer avec la base de données s'appelle le SQL (Structured Query Language).



**MySQL** fonctionne sur pratiquement toutes les plates-formes, y compris Linux, Unix et Windows. Il est entièrement multi-thread avec un noyau de threads, et fournit des API (Application Programming Interface) pour de nombreux langages de programmation, notamment C, C ++, Eiffel, Java, Perl, PHP, Python, et Tcl.

**MySQL** est utilisé dans une large gamme d'applications, Le commerce électronique, les bases de données Web...

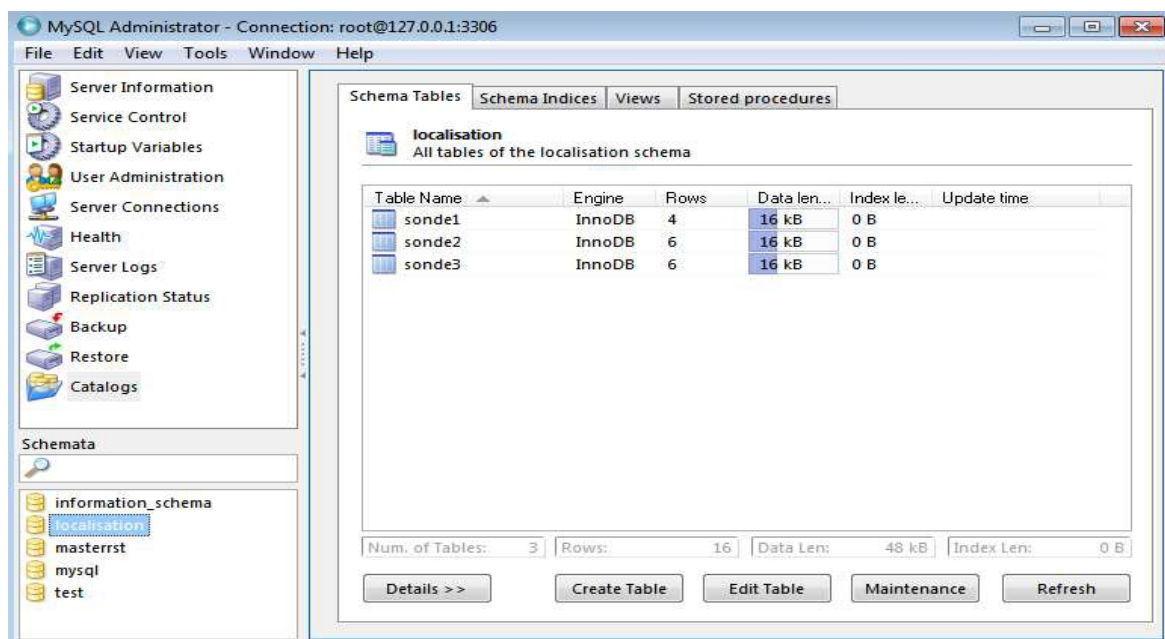
Selon **MySQL AB**, avec plus de dix millions de serveurs MySQL installés dans le monde entier, MySQL est devenu le leader mondial du marché des Bases de Données. MySQL compte des clients prestigieux comme Google, la NASA ou Suzuki...

#### Les avantages de MYSQL :

- **Il est facile à comprendre:** Sa syntaxe simple en fait un langage facile à comprendre pour les programmeurs et des débutants.
- **Le Langage est fonctionnel:** MySQL fonctionne sur de nombreuses plates-formes différentes.

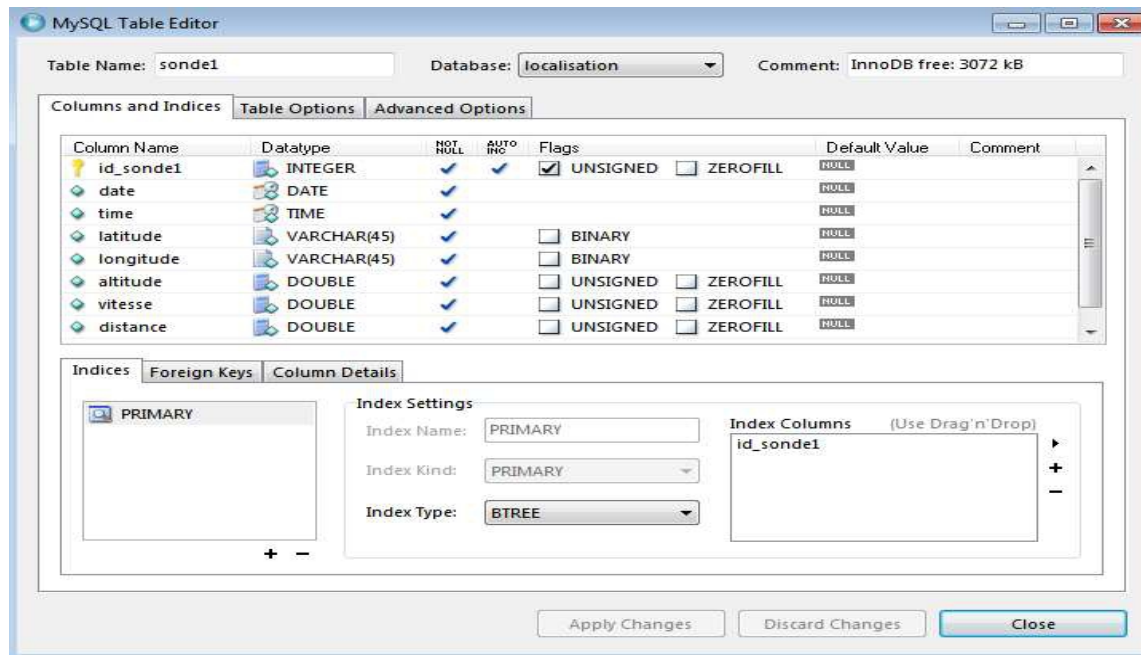
- **Dispose d'une vaste bibliothèque de fonctions et d'API:** API pour C, C++, Eiffel, Java, Perl, PHP, Python, Ruby et Tcl sont disponibles. Les fonctions SQL sont mises en place en utilisant une bibliothèque de classes optimisées.
- **Multi Thread:** Complètement multi-thread utilisant un noyau de threads.
- **Haute capacité de stockage:** Pour vous donner une idée: De grosses entreprises actuelles utilisent le serveur MySQL avec plus de 1.00.000 tables et 1.000.000.000 d'enregistrements.

Pour cette opération nous allons créer une base de données contient des tableaux utiliser pour enregistrer les coordonnées des sondes, nous allons dédiée pour chaque sonde un tableau. Ces tableaux sont créés par un logiciel « **MySQL Administrator** » qui est un logiciel utilisé pour créer des bases de données graphiquement.



**Fig4.3: Les bases des données créés par MySQL Administrator**

Dans cette figure nous avons présenté l'interface graphique de « **MySQL Administrator** » qui contient à gauche les bases de données, chacune contient un ou plusieurs tableaux, dans le cas de la base « localisation », elle contient plusieurs tableaux sonde, sonde2, sonde3..... Chaque un enregistre les informations d'une sonde c'est-à-dire la position (latitude, longitude et l'altitude) la vitesse, la date et la distance parcourue.



**Fig4.5 Les tableaux créés par MySQL Administrator**

Après la création des bases de données l'enregistrement ce fait par des scripts SQL, nous allons intégrer ces scripts dans une page web à l'aide des scripts PHP, cette méthode facilite notre travail, la page qui fait la réception des données fait en même temps l'enregistrement, pour ce le script PHP établie en premier une liaison avec la base de donnée, cette opération fait par le script suivant :

```
$link = mysql_connect("mysql17.000webhost.com","a4785593_U","")
or die ('Could not connect:' . mysql_error());
```

Quand la relation est établie donc on fait maintenant l'enregistrement des données avec le scripte suivant :

```
$sql="INSERT INTO `localisation`.`$sonde`
(`date`, `time`, `latitude`, `longitude`, `altitude`, `distance`, `vitesse`)
VALUES ('$date', '$time', '$latitude', '$longitude', '$altitude', '$distance', '$vitesse');";
```

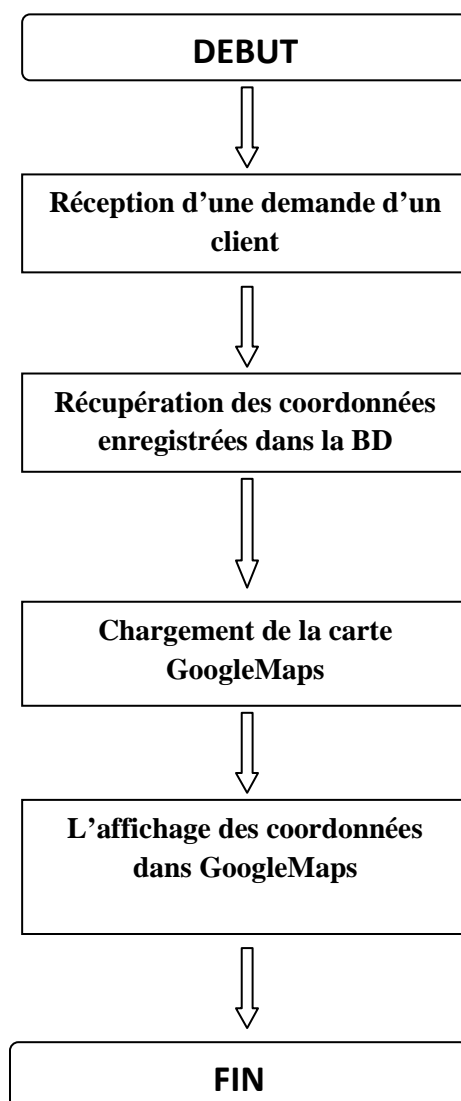
Si l'enregistrement est terminé on coupe la liaison avec BD par le script :

```
mysql_query($sql);
mysql_close($liendb);
```

### 4.3.2- l'affichage de la position sur une carte GoogleMaps :

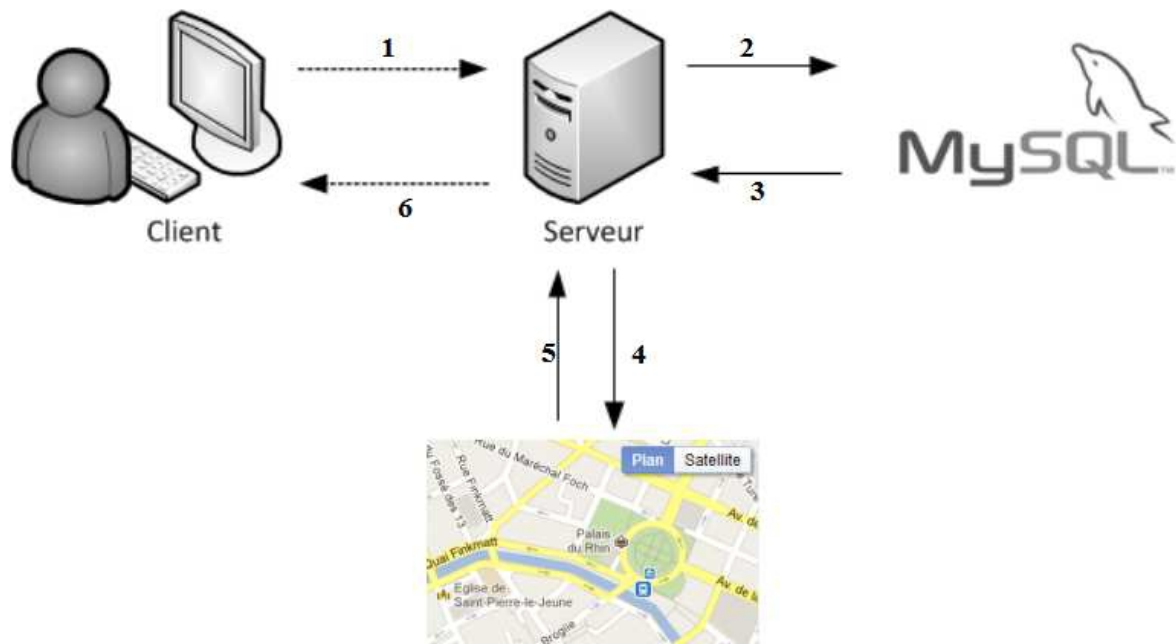
Dans notre projet nous avons développé un site web responsable de l'affichage d'une ou de plusieurs position (engin) sur une carte GoogleMaps a la demande des utilisateurs.

Les utilisateurs sont divisés en deux grande partie selon leur privilèges ; administrateur et client de visualisation. Pour ce la nous avons créé une base de données contient deux tableaux, un pour les administrateurs et l'autre pour les clients de visualisation.



**Fig4.6: Algorithme pour l'affichage des positions sur une carte**





**Fig4.7:** Description de la méthode pour l'affichage des positions sur la carte

#### A- Réception d'une demande d'un client :



**Fig4.8:** Principe de fonctionnement d'un serveur Web

Lorsqu'un client fait une demande pour récupérer une page web par un navigateur web le code PHP est exécuté en premier sur le serveur et fait ce qu'on lui demande.

Une fois toutes les instructions PHP exécutées, la page demandée devient une page HTML (contient que du HTML).

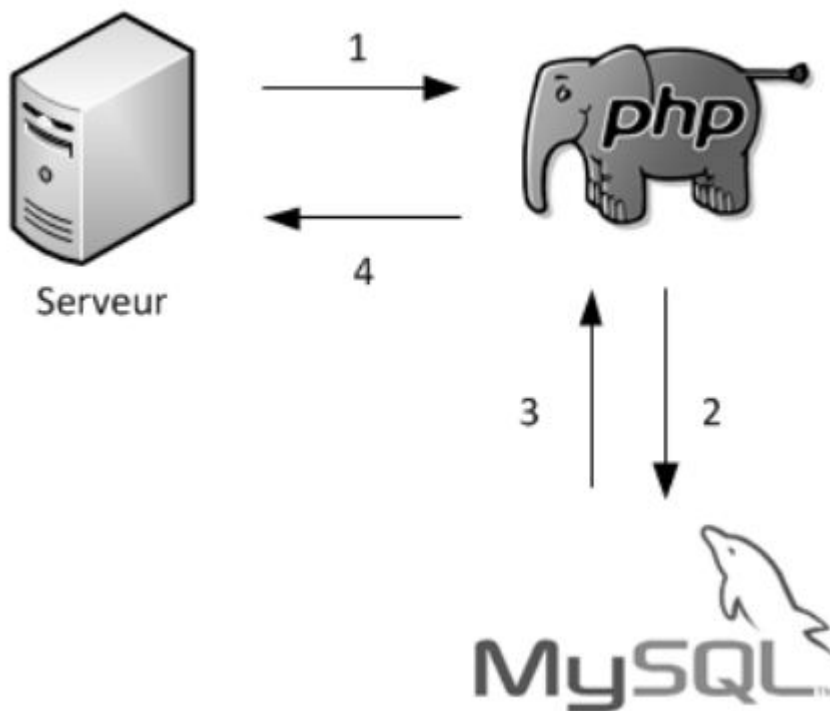
Si la page demandée contient aussi des scripts java (JavaScript) elles seront exécutées sur la machine client.

C'est cette page «résultat» qui est envoyée au visiteur, car celui-ci ne sait lire que le HTML et de JavaScript.

Dans notre cas le client (administrateurs, clients de visualisation) fait une demande pour la visualisation des positions des différents engins, tous les codes PHP seront exécuté sur la machine serveur, les code PHP contient surtout les scripts de communication avec les bases de données, donc cette méthode est très efficace pour la sécurité.

La carte GoogleMaps est incluse dans un script JavaScript qui sera exécuté sur la machine client.

### B- Récupération des coordonnées enregistrées dans la BD :



**Fig4.9: Récupération des données à partir d'une base de données**

Le client fait une demande pour voir la position d'un ou de plusieurs engins.

Le serveur doit récupérer les coordonnées de ces engins qui sont enregistrées dans la base de données BDD. Pour effectuer cette opération PHP doit faire l'intermédiaire entre nous et MySQL.

Pour pouvoir travailler avec la base de données en PHP, il faut d'abord s'y connecter.

En effet, MySQL demande d'abord un nom d'utilisateur et un mot de passe. S'il ne le faisait pas, tout le monde pourrait accéder à notre BDD et lire les informations qu'elle contient.

Il va donc falloir que PHP s'authentifie: on dit qu'il établit une connexion avec MySQL.

```
$link = mysql_connect("mysql17.000webhost.com","a4785593_U","")  
or die ('Could not connect:' . mysql_error());
```

Une fois que la connexion est établie, nous pourrions lire les coordonnées d'un engin enregistré sur la BDD.

```
$sql="SELECT * FROM sonde1";
```

Donc maintenant les positions des engins sont récupérées.

### **C- Chargement de la carte GoogleMaps:**

#### **GoogleMaps :**

GoogleMaps est un service web cartographique gratuit créé par Google en 2004.

On peut naviguer dans GoogleMaps facilement avec la souris, on peut zoomer pour voir le détail d'une rue par exemple. L'utilisateur peut entrer une adresse ou le nom d'une place connue pour pouvoir l'identifier rapidement sur la carte.

GoogleMaps permet aussi de faire le routage. Il donne à l'utilisateur une liste des trajets pour arriver à sa destination avec une estimation de la durée de chaque trajet et la distance qui sépare entre les deux locations.

#### **Définition de l'API GoogleMaps :**

Une API est une interface de programmation. Dans le cas de GoogleMaps, il s'agit d'un ensemble de fonctions et classes JavaScript qui permettent de manipuler une carte dynamiquement au sein d'un site web.

#### **Inclure GoogleMaps :**

Pour inclure les fonctions GoogleMaps dans notre application, nous ajoutons une balise script dans la partie Head de notre page:

```
<script type="text/javascript"
src="http://maps.google.com/maps/api/js?sensor=false"></script>
```

On va inclure la bibliothèque GoogleMaps directement dans le code JavaScript de la page. Cette URL permet d'accéder à l'ensemble des fonctionnalités de l'API.

Maintenant nous allons afficher la carte dans notre site, pour ce la nous utilisons premièrement le code suivant :

```
<div id="carte" style="width:100%; height:75%"></div>
```

Ce code définit la taille de la carte.

#### D- L'affichage des coordonnées dans GoogleMaps:

Pour faire l'affichage des coordonnées dans une carte, il faut premièrement lire ces coordonnées à partir de la base de données, pour cela on utilise un simple script PHP qui est le suivant :

```
$sql="SELECT * FROM $tab[$i]";
```

Ce code nous permet de récupérer le dernier enregistrement d'une sonde bien déterminé.

Nous souhaitons afficher la carte une fois notre page html chargée. Pour cela nous allons ajouter l'attribut "onload" à notre balise "body" :

```
<body onload="initialiser()">
```

Puis créer la fonction "initialiser()" dans un script javascript. Cette fonction va, comme son nom l'indique, initialiser la carte Google Maps.

```
function initialiser() {
    var latlng = new google.maps.LatLng(34.921115,-1.303070);
    var options = {
        center: latlng,
        zoom: 13,
        mapTypeId: google.maps.MapTypeId.ROADMAP
    };
    var carte = new google.maps.Map(document.getElementById("carte"), options);
}
```

Dans cette fonction nous définissons une position (latitude, longitude) qui sont le centre de cette carte, nous allons définir aussi le Zoom.

Maintenant nous allons afficher les positions des engins sur la carte, ces positions seront indiquées par des marqueurs, dans la fonction “initialiser()” nous allons définir les marqueurs et nous allons donner leurs positions sur la carte par le Script suivant :

```
var marqueur= new google.maps.Marker({
  position: new google.maps.LatLng(tab[j][2], tab[j][3]),
  map: carte,
  title: ''+tab[j][7]+'',
  icon: 'http://maps.google.com/mapfiles/ms/icons/'+tableau[tmp-1]+'-dot.png'
});
```

Avec cette méthode on peut afficher une ou plusieurs positions en même temps.

### E- Affichage d'un parcours :

Par la même procédure on peut afficher un parcours d'un engin dans un intervalle de temps bien déterminer, pour cela nous récupérons tous les positions de cet engin enregistrée dans la base de données dans cet intervalle par le code suivant :

```
$query="SELECT latitude FROM $tab[0]
WHERE date= '$dat' AND time>='$tim1' AND time<='$tim2'" ;
```

Puis dans la fonction “initialiser()” nous affichons ces positions et tracer un parcours entre ces points en utilisant le code suivant :

```
for (var k=0;k<=j;k++){
parcoursBus[k]= new google.maps.LatLng(tab1[k],tab2[k]);
}

var traceParcoursBus = new google.maps.Polyline({
  path: parcoursBus,//chemin du tracé
  strokeColor: "#FF0000",//couleur du tracé
  strokeOpacity: 1.0,//opacité du tracé
  strokeWeight: 2//grosseur du tracé
});
```

#### 4.4-Hébergement de notre application :

Maintenant après la création de notre application, nous allons héberger notre site web dans un serveur web. Quand notre site est hébergé tous les utilisateurs peuvent accéder à cette application à travers une connexion internet. De même, l'avantage majeur est que notre portail web va être associé avec une adresse IP fixe et publique, grâce à cette adresse la sonde à base du module GM862-GPS peut communiquer avec la station de surveillance.

Notre site web est hébergé dans le serveur [www.000webhost.com](http://www.000webhost.com), l'inscription pour ce serveur est gratuite, et en plus il nous donne la possibilité d'héberger aussi des bases de données.

Après notre inscription nous avons obtenu des privilèges limités à travers un nom d'utilisateur et un mot de passe. Nous avons utilisé les commandes FTP à travers l'invite de commandes DOS, pour les télécharger sur le serveur. Les commandes FTP sont des simples instructions qui nous permettent à travers l'invite de commande d'exécuter des actions sur le serveur distant : créer, envoyer, récupérer et supprimer des fichiers.

Nous commençons par l'établissement d'une connexion avec le serveur, pour ce la nous avons utilisé la commande **open** et l'adresse IP de ce serveur, le nom d'utilisateur et le mot de passe :

```
>>Open 31.170.160.93
```

```
>> a4785593
```

```
>>*****
```

Une fois la connexion est bien établie, nous transférons les fichiers vers le répertoire **Public\_html** activé avec la commande **cd** :

```
>> cd Public_html
```

Le transfère des fichiers se fait à l'aide de la commande **put** :

```
>> put < chemin vers le fichier >
```

```
>> put c:\ monsite\index.PHP
```

Il est possible d'envoyer plusieurs fichiers en même temps par la commande **mput** :

```
>> mput c:\monsie\*.PHP
```

Le paramètre « \*.PHP » signifie tous les fichiers avec l'extension PHP.

Pour la création des répertoires, nous utilisons la commande **mkdir** avec le nom de ce répertoire :

```
>> mkdir administrateur
```

La commande **delete** supprime des fichiers :

```
>> delete index.PHP
```

Les images sont envoyées en mode **binary** :

```
>> binary
```

```
>>put c:\monsie\im1.jpg
```

Pour fermer la connexion nous utilisons la commande **close** :

```
>> close
```

#### 4.5- Conclusion :

La station de surveillance offre des services qui nous permettent de suivre un ou plusieurs engins en temps réel, en plus elle nous permet de voir le parcours d'un engin.

La station de surveillance fait aussi la communication avec toutes les sondes pour recevoir les positions de ces engins.

# Chapitre 5

## Application



### 5.1- Introduction :

Dans ce chapitre nous allons présenter les résultats de notre travail, ce travail est devisé en deux grandes parties :

- 1- Réalisation d'un système capable de faire le suivi des véhicules soit en temps réel soit en temps différé (historique)
- 2- Création d'un site web pour la télésurveillance de ces véhicules, les utilisateurs sont divisés en deux grandes parties (administrateurs, client) avec des privilèges différents

### 5.2- Administration de l'application :

Ce module contient la partie authentification qui permet de gérer l'accès à l'application, chaque utilisateur introduit son nom et son mot de passe pour accéder à l'application.

L'application définit deux niveaux d'utilisateur : Les administrateurs, les clients.



Fig5.1: Fenêtre d' Authentification

Lorsque l'administrateur de l'application s'authentifie, le système lui redirige vers la fenêtre suivante :

bienvenue ! [Déconnexion](#)

pour ajouter un observateur utiliser "ajouter un observateur"

[ajouter un observateur](#)

pour supprimer un observateur utiliser "supprimer un observateur"

[supprimer un observateur](#)

pour modifier un observateur utiliser "modifier un observateur"

[modifier un observateur](#)

pour ajouter une appareil utiliser "ajouter une appareil"

[ajouter une appareil](#)

pour voire les traces utiliser "voire les traces"

[voire les traces](#)

**Fig5.2: Espace administrateur**

Dans cet espace, nous donnons à l'administrateur la possibilité de gérer l'application.

L'administrateur peut ajouter, modifier ou supprimer un utilisateur comme il a la possibilité d'ajouter un engin à partir des fenêtres suivantes :

**Veuillez remplir ce formulaire**

nom de l'observateur:

prénom de l'observateur:

Nouveau nom de l'observateur:

Nouveau prénom de l'observateur:

Nouveau email de l'observateur:

Nouveau Mot de passe:

[enregistrer](#)

[Home](#)

**Veuillez remplir ce formulaire**

Nom de l'observateur:

prénom de l'observateur:

email de l'observateur:

Mot de passe:

[enregistrer](#)

[Home](#)

**Veuillez remplir ce formulaire**

Nom de l'observateur:

prénom de l'observateur:

[supprimer](#)

[Home](#)

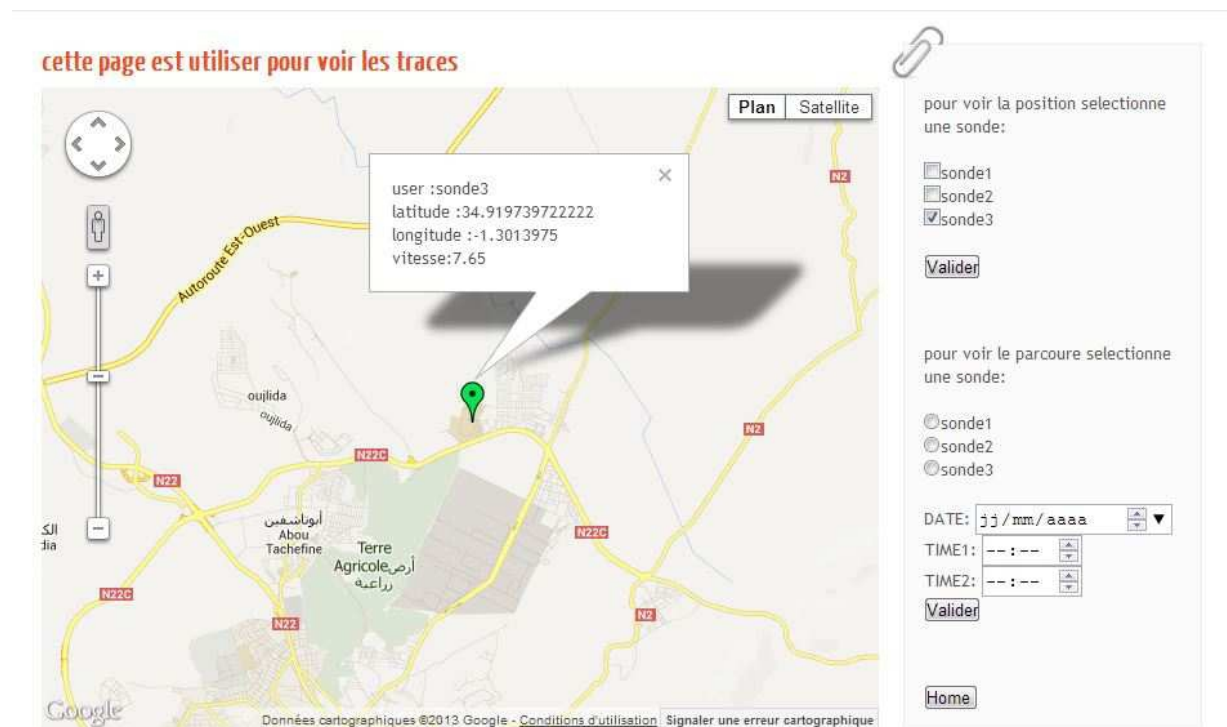
**Fig5.3 : Fenêtres de différentes opérations gérer par l'administrateur**

### 5.3- Suivi en temps réel :

Dans cette partie, nous donnons à chaque administrateur et utilisateur la possibilité de suivre en temps réel les véhicules.

L'administrateur ou utilisateur doit sélectionner un ou plusieurs engins qui sont indiqués dans une liste qui contient toutes les sondes. Chaque sonde est associée avec un véhicule.

Chaque engin sera indiqué par un marqueur de couleur différente à celle des autres engins (marqueurs), un simple cliquer sur le marqueur donne tous les informations de ce véhicule (nom, latitude, longitude et la vitesse) comme indique la figure suivante :



**Fig5.4: Position du véhicule en temps réel**

### 5.4- Suivi en temps différé :

Ce mode ressemble au premier, sauf qu'on consulte les données enregistrées dans un historiques. L'utilisateur choisit le véhicule, en plus la date et la durée de visualisation.

La figure suivante présente un exemple de la visualisation de l'historique d'un véhicule :



**Fig5.5: Parcours d'un véhicule pendant une durée**

**5.5- Conclusion :**

Dans ce chapitre nous avons exposé la partie de notre travail qui consiste à réaliser une plateforme logiciel (PHP, MySQL, JavaScript) combinée avec les travaux effectués précédemment avec le module GM862GPS pour la télésurveillance des engins soit en temps réel ou bien en temps différé. La solution globale se présente sous un site web qui offre notre service avec des différents privilèges. La solution est actuellement disponible sur internet : [www.localisationrst.netii.net](http://www.localisationrst.netii.net) via le serveur de site web gratuit [www.000webhost.com](http://www.000webhost.com).

## **Conclusion générale:**

Ce projet nous permet d'aborder une expérience réelle dans le domaine de la conception, le design, le développement ainsi que la gestion d'un projet de conception d'un système embarqué avec une exploitation directe sur le marché. En effet, en plus de l'aspect recherche et technique, ce projet vise la réalisation dans un avenir à moyen terme d'un prototype opérationnel destiné au commerce, ce qui nous oblige à prendre en compte le volet économique en minimisant les prix de fabrication de service et d'exploitation.

Encourager l'utilisation publique de la technologie d'information géographique. En effet, des engins dotés d'un dispositif de localisation peuvent avoir d'innombrables retombés sur le secteur public à savoir la surveillance des déplacements, assurer la sécurité des passagers....

Encourager le secteur privé à intégrer les technologies d'information géographique à faibles coûts.

A l'image des pays développés, avec un projet de localisation GSM/GPS il est possible de dynamiser la création de nouvelles entreprises de haute technologies pour la fabrication de modules, systèmes, logiciels et systèmes d'exploitation destinés à s'intégrer sur le produit créé.

## **Bibliographie :**

- [1] GPS et Galileo Systèmes de navigation par satellites, Jean-Marc Piéplu, Eddition Eyrolles : 2006
- [2] le système GPS « thèse de diplôme d'ingénieur »; KEDDARI Ahmed et REGGAS Mehdi
- [3] « [http://fr.wikipedia.org/wiki/Global\\_Positioning\\_System#Autres\\_syst.C3.A8mes\\_de\\_positionnement\\_par\\_satellites](http://fr.wikipedia.org/wiki/Global_Positioning_System#Autres_syst.C3.A8mes_de_positionnement_par_satellites) »
- [4] Etude du GPS et ses techniques de localisation « thèse de diplôme d'ingénieur »
- [5] Etude d'un système GPS hybride pour localisation en conditions de visibilité « thèse de diplôme d'ingénieur » ; Mr Ould M<sup>ed</sup> Mahmoud Hafed et Mr Hachemaoui Smaine
- [6]-Etude et modélisation des effets de la ionosphère sur le Système de navigation par satellite « projet de fin d'études pour l'obtention du diplôme d'ingénieur d'Etat en télécommunication par Mr REHOUMA M<sup>ed</sup> Larbi et Mr LAHCENE Abdelkader
- [7] « <http://www.gps.gov/french.php> »
- [8] Guide GPS pour débutant ; « Garmin,International » 2002/2003
- [9] Réseau GSM « thèse de diplôme d'ingénieur » Par OMBANG Mathurin.
- [10] Réseau GSM « thèse de diplôme d'ingénieur » Par MEYENG ABATH Christian.
- [11] Sécurité et gestion de la mobilité dans le réseau GSM « thèse de diplôme d'ingénieur » Par BOUTIOUTA Aboubakr.
- [12] Concevez votre site web avec PHP et MySQL, Mathieu Nebra alias
- [13] Maitrisez PHP de A à Z, François-Xavier BOIS, Editeur : MICRO APPLICATION, SA
- [14]www.PHP.com
- [15]PHP et MySQL, Luke Welling & Laura Thomson, Pearson Education France
- [16]:Telit GM862 Family Hardwar user guide: 2ème edition 15/09/2011
- [17]:[http://fr.wikipedia.org/wiki/Commandes\\_Hayes](http://fr.wikipedia.org/wiki/Commandes_Hayes)
- [18]: Telit AT commands reference guide
- [19] :www.evola.fr
- [20] : Programmation trames GPS : auteur : G.VALET, date: octobre 2010
- [21] : <http://www.gpsinformation.org/dale/nmea.htm>

## RESUME :

Ce projet vise l'intégration d'un service innovant en télécommunications mobile. Un tel service consiste à exploiter les paramètres de localisation GPS, suivant un format imposé, à travers les réseaux mobiles du type GSM. Une étude des systèmes GPS avec les différents formats des coordonnées position est à présenter et détailler. Le service GSM est proposé comme support de transmission du service GPS. Une variante d'un système embarqué intégrant les deux services a été conçue, intégrée et testée avec le module GM862-GPS et un portail WEB.

## ABSTRACT:

This project deals with the integration of new innovating service in mobile telecommunications. This service consists in exploiting the GPS localization parameters, through the GSM mobile networks. A survey of GPS coordinate's systems with different position formats has been presented and retailed. The GSM service is proposed as transmission medium of the GPS positions, an embedded system with the GM862-GPS integrating the two services has been designed, built and tested.

## الملخص:

ينطوي هذا المشروع على دمج عدة خدمات في الاتصالات اللاسلكية. هذه الخدمات تتمثل في استخدام نظام تحديد المواقع GPS ونظام شبكات الهاتف المحمول GSM. من خلال هذا العمل قمنا بدراسة مختلف أنواع إحداثيات GPS و قد قمنا باستخدام نظام GSM كوسيلة لنقل هذه الإحداثيات. وقد قمنا بتصميم نظام لدمج خدمتي GPS و GSM مستخدمين النظام المدمج GM862-GPS و موقع الكتروني.