



**68 rue musselburg - 94500 Champigny sur  
marne – France**

**Tél : +33 1 70 81 93 29**

**Email : [info@kt-centrex.com](mailto:info@kt-centrex.com)**

**[www.kt-centrex.com](http://www.kt-centrex.com)**

---

# RAPPORT DE STAGE

Sujet de stage : Configuration des Gataway et les programmes autour  
d' Asterisk



**IUT Villetaneuse**

**99, avenue Jean Baptiste Clément**

**93430 VILLETANEUSE**

**Tél : 01 49 40 30 00**

**Fax : 01 49 40 20 97**

**[www.iutv.univ-paris13.fr](http://www.iutv.univ-paris13.fr)**

UTHAYARASA  
Surekananth  
LP – Réseau et  
télécommunication  
2011/2012



---

## REMERCIEMENT

Je tiens tout particulièrement à remercier mon maitre de stage M. Kalonji Tony pour son accueil chaleureux et aussi de m'avoir accepté en tant que stagiaire au sein de KT-CENTREX.

Je remercie également Mr. Marc Vervel et Mme SABINE Bodioga de m'avoir aidé à trouver ce stage.

D'une façon plus générale, je remercie l'ensemble de l'équipe de KT-CENTREX de m'avoir aidé à bien dérouler mon stage et à bien finir mon projet.

# Table de matière

RAPPORT DE STAGE .....	1
Remerciement.....	3
Abstract .....	5
Introduction .....	6
Présentation de l'entreprise .....	7
Présentation du projet .....	9
1) Introduction .....	9
2) Interface client du soft Switch .....	9
3) Interface administrateur .....	13
A) Rubrique « paramétrage du group » .....	14
B) Rubrique « Monitoring 2 » .....	14
4) Groupes.....	15
A) qu'est ce qu'un serveur smsc ?.....	15
B) qu'est ce qu'un modem gsm ?.....	16
C) Les commandes AT .....	17
D) Envoi de sms depuis un script Perl .....	19
F) l'application smssoft .....	22
Problèmes rencontrés.....	32
Conclusion .....	33
Index.....	34
Annexe 1 : Rubrique paramétrage du group.....	35
Annexe 2 : Rubrique monitoring .....	36
Annexe A : script de confirmation.....	37
Annex B : script de changement .....	40

---

## ABSTRACT

As part of the "Professional License in Network and in Telecommunication", we have to perform an internship in a company for a period of three months.

I choose to do my internship in a society named KT-CENTREX. I worked as a software developer on a project soft Switch sms.

This report presents the work that I did during my internship. This training course was held from May 25th to August 17th, 2012 in KT-CENTREX. During my course, I got used to the programming language and materials of the network and the telecommunication.

My work had two main objectives: the first one was to create a program which allows sending and receiving messages through SMS Gateway. The second part was to change the web interface SoftSwitch sms which already existed.

In a first part, I set out, in this report, a presentation of the company. Then, I will explain the different aspects of my work over the last few months and, in conclusion, I summarize the contributions of this internship.

# INTRODUCTION

Dans le cadre de la Licence Professionnelle Réseau et Télécommunication, nous devons effectuer un stage dans une entreprise d'une durée de 3 mois.

J'ai choisi d'effectuer mon stage dans l'entreprise KT-CENTREX qui fournit une solution complète pour un centre d'appel. J'ai travaillé en tant que développeur sur un projet soft Switch sms.

Ce projet sera expliqué plus tard en détails dans ce rapport.

Ce rapport présente le travail que j'ai effectué lors de mon stage. Ce stage s'est déroulé du 25 mai au 17 août 2012 au KT-CENTREX. Pendant la période de mon stage, je me suis familiarisé avec le langage de la programmation et les matériaux concernant le réseau et la télécommunication.

Ma mission comportait deux principaux objectifs : le premier était de faire un programme qui permet d'envoyer et recevoir les messages grâce à des Gateway sms. La deuxième partie consistait à modifier l'interface web du SoftSwitch sms qui existait déjà.

Je vous expose dans ce rapport en premier lieu une présentation des entreprises. Ensuite, je vous explique les différents aspects de mon travail durant ces quelques mois et enfin, en conclusion, je résume les apports de ce stage.

## PRESENTATION DE L'ENTREPRISE

KT-CENTREX est une entreprise qui offre une solution complète pour les centres d'appels et qui offre aussi une solution Asterisk, c'est-à-dire installer, programmer ou configurer un module dans un environnement Asterisk sur lequel les clients rencontrent des difficultés.

KT-CENTREX fournit un support en ligne sur les produits suivants : Soft-Switch, Call Shop, Calling, Card VoIP, crm, ViciDial, A2Billing, asterCC, CDRTTool, HylaFAX, ipbx, IPBX, Asterisk, trixbox, RTPproxy, MediaProxy, elastix.

KT-CENTREX a trois bureaux de représentations dans le monde : un bureau en Tunisie, un bureau au Maroc et un autre bureau à l'Île Maurice. Le bureau en France se situe à l'adresse suivante : 68 rue musselburgh - 94500 Champigny sur marne - France

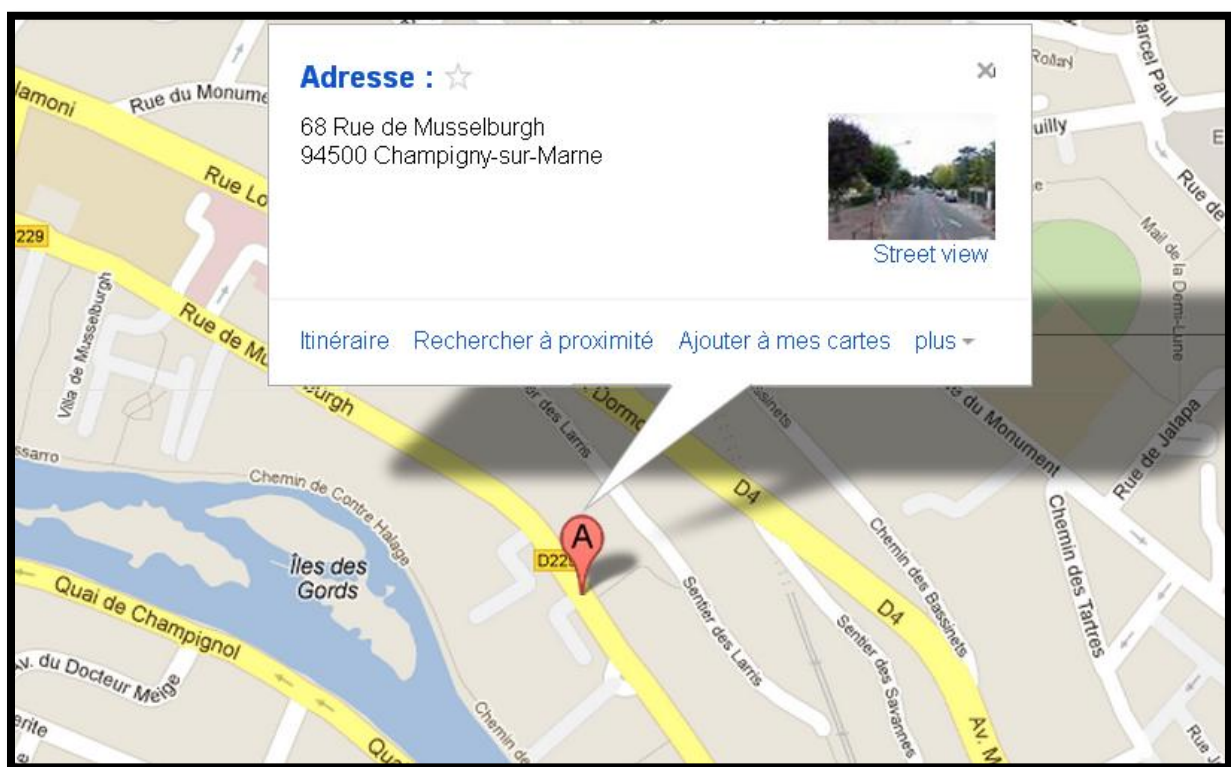


Figure 1 : Emplacement géographique de l'entreprise

Pourtant le déroulement du stage s'est déroulé à l'adresse suivante : 32 Rue Raspail  
92270 Bois-Colombes

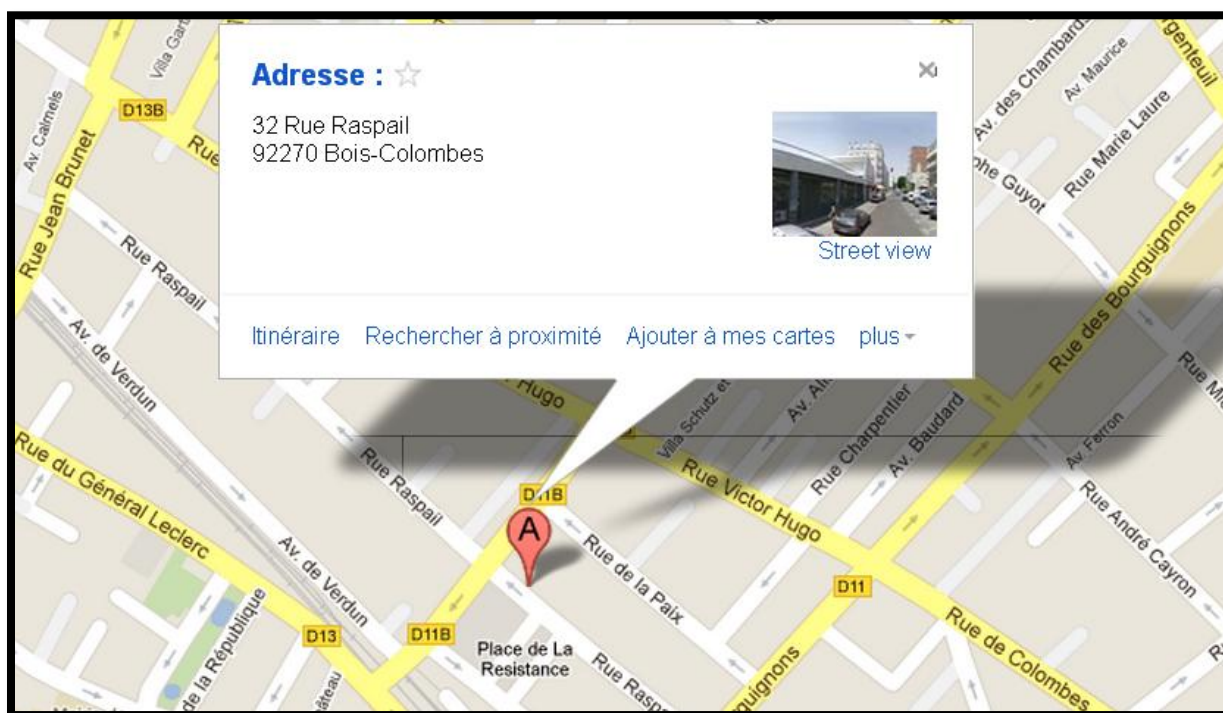


Figure 2 : Lieu du stage



# PRESENTATION DU PROJET

## 1) INTRODUCTION

Ce projet consiste à créer un soft Switch sms : ce soft Switch permet d'envoyer 50000 sms par jour grâce à des modems GSM. Nous avons créé une interface web pour administrer le soft Switch et une interface web pour les clients afin qu'ils puissent uploader et créer des campagnes de sms. L'interface administrateur est faite pour faire le monitoring sur les « groupes » de modems (un « groupe » est un ordinateur dans lequel on branche les modems) et voir les statistiques d'envois de sms.

## 2) INTERFACE CLIENT DU SOFT SWITCH

Comme je l'ai expliqué dans la partie Introduction, le but de ce projet est d'envoyer plus de 50000 sms par jour. Donc si nous voulons réussir ce projet, nous aurons besoin de 50000 contacts. Cela est plus difficile si l'on demande au client de le saisir manuellement. Pour faciliter les tâches, le client aura sa propre interface web dans lequel il pourra se connecter en utilisant son identifiant et son mot de passe. Ensuite il aura la possibilité d'uploader un fichier Excel des contacts afin de pouvoir envoyer des sms.

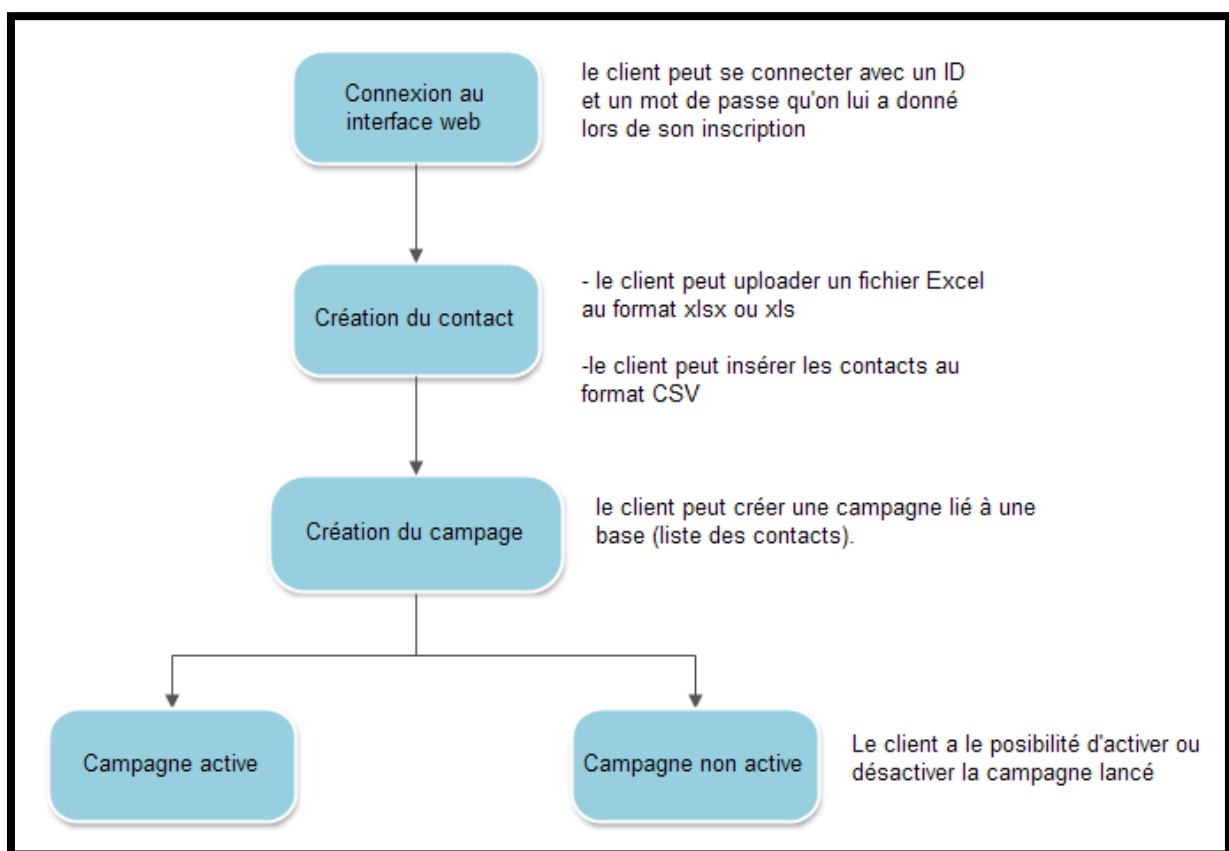


Schéma structurel

Donc pour uploader les contacts, il doit se rendre dans la rubrique de « crm sms » puis choisir l'option « import base & gestion ».

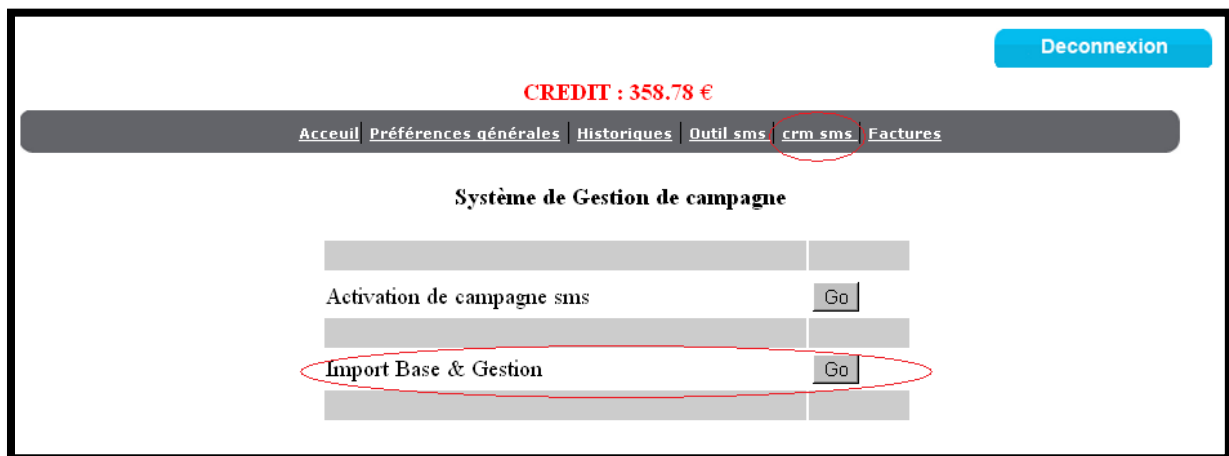


Figure 1 : Crm sms

La phase de création du contact peut se faire de deux manières différentes. Soit il uploade une base Excel au format xls ou xlsx, soit il peut insérer les contacts au format CSV.

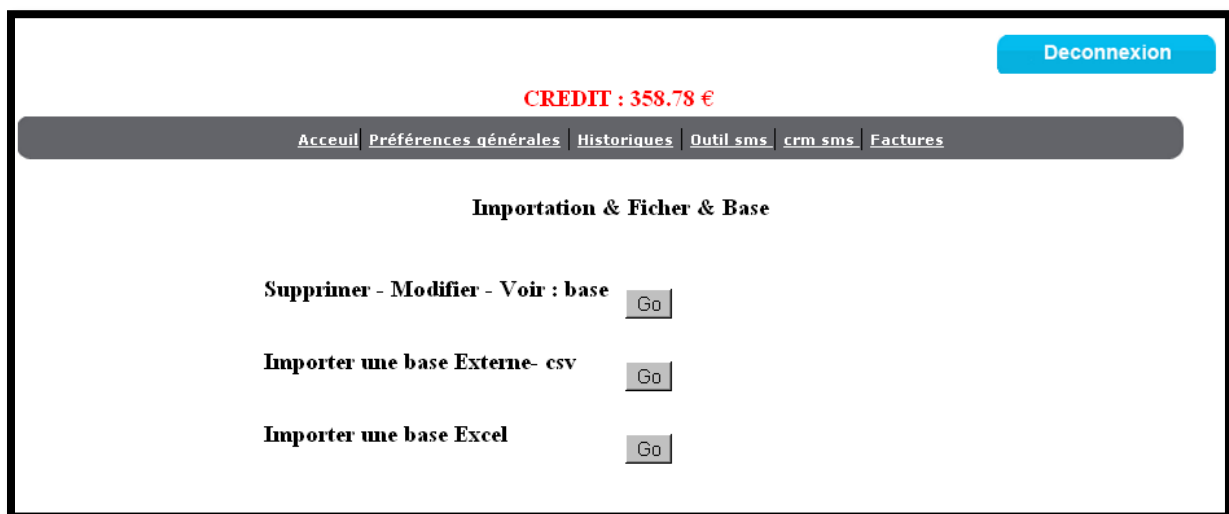


Figure 2 : Importation du contact

Les options « Supprimer-Modifier-Voir : base » et « Importation une base Externe csv » ont déjà été créés. J'ai travaillé uniquement sur la partie « importer une base Excel ».

Lorsque le client veut uploader une base Excel, nous lui demandons de saisir un nom pour cette base afin de faciliter la démarche de la création de la campagne de sms.

Figure 3 : Upload du fichier Excel

Une fois que le client a uploadé le fichier Excel, il y a un script de perl qui récupère le fichier qui, en même temps, traite le fichier puis propose au client de confirmer les champs comme dans la figure 4.

Nom	prenom	Telephone	variable 1	variable 2
Nom	Nom	Nom	Nom	Nom
Prenom	Nom	telephone	varibale2	variable1
Surekananth	Uthayarasa	06 50 90 82 33	95400	Villier-le-bel
Gmutha	Muturaja	06 34 48 23 83	95200	Louvre
Tony	Kalonji	06 32 56 89 45	92270	Bois-Colombes
				suivant

Figure 4 : Demande de changements des champs

S'il existe des changements que le client veut effectuer, cela peut se faire grâce au menu déroulant puis en cliquant sur le bouton suivant. Lorsque le changement des champs est validé par le client, il y a un autre script de Perl qui récupère le fichier à traiter et les changements des champs saisis par le client afin d'insérer les contacts dans une base de données.

Une fois le traitement du fichier terminé, le client pourra voir ou modifier les contacts ou il pourra créer une campagne associée à cette base uploadée en incluant le message à envoyer. Une fois la campagne créée, il peut ensuite l'activer ou la désactiver.

Nom base	voir	Supprimer
3000l	<a href="#">Go</a>	<a href="#">Go</a>
base dd	<a href="#">Go</a>	<a href="#">Go</a>
espace	<a href="#">Go</a>	<a href="#">Go</a>
rrrrrrr	<a href="#">Go</a>	<a href="#">Go</a>
test_1	<a href="#">Go</a>	<a href="#">Go</a>
test_3000_1	<a href="#">Go</a>	<a href="#">Go</a>
tony	<a href="#">Go</a>	<a href="#">Go</a>

Figure 5 : Voir ou modifier les bases

Creation d'une campagne

Nom de la campagne :

Date de début : 1 / 1 / 2012

Date de fin : 1 / 1 / 2012

Nom du produit :

\* Associer une base a votre campagne :

Message :

Variable :

Texte :

variable :

Texte :

variable :

Texte :

variable :

Envoyer

Figure 6 : Création d'une campagne

[Deconnexion](#)

**CREDIT : 358.78 €**

[Accueil](#) [Préférences générales](#) [Historiques](#) [Outil sms](#) [crm sms](#) [Factures](#)

Liste des campagnes
Créer Une Campagne

### Liste des campagnes

Page: 1

NOM CAMPAGNE	NOM PRODUIT	DATE DEBUT	DATE FIN	BASE ASSOCIEE	MESSAGE	ETAT	ACTION
		1/1/2012	1/1/2012	rrrrrrrr	[nul] [nul] [nul] [nul]	Activée	<div>Desactiver ▾</div> <div>Go</div>
ricky2	popop	20/7/2012	28/7/2012	test_1	ricky1 [nom] ricky2 [nul] [nul] [nul]	Activée	<div>Desactiver ▾</div> <div>Go</div>
halousolaire	basesolaire2	19/7/2012	24/7/2012	test_1	bonjour [nom] je vien le [date] je serais a [variable1] puis a [variable2]	Activée	<div>Desactiver ▾</div> <div>Go</div>
basesolaire2toto	basestitoo	1/1/2012	1/1/2012	test_1	hello [nul] [nul] [nul] [nul]	Activée	<div>Desactiver ▾</div> <div>Go</div>
		1/1/2012	1/1/2012	rrrrrrrr	[nul] [nul] [] []	Activée	<div>Desactiver ▾</div> <div>Go</div>
campagne123	delinso	15/7/2012	18/7/2012	rrrrrrrr	salut [nom] du [variable1] [] []	Activée	<div>Desactiver ▾</div> <div>Go</div>

Figure 7 : Liste des campagnes

Le client peut vérifier l'état de la campagne et activer ou désactiver celle-ci.

### 3) INTERFACE ADMINISTRATEUR

L'interface d'administrateur est faite pour ajouter ou supprimer des « groupes ». Un « groupe » est un ensemble de modems branchés sur un ordinateur. Cette interface permet aussi de faire le monitoring en temps réel et voir les statistiques, les historiques, les factures, la liste des clients, les paramétrages, etc...

De plus, des rubriques ont été créées : j'ai travaillé sur les rubriques suivantes :

- ➔ Monitoring 2
- ➔ Paramétrage -> Paramétrage du group

A) RUBRIQUE « PARAMETRAGE DU GROUP »

Dans cette rubrique, l'administrateur peut ajouter ou supprimer les groupes. Nous lui demandons le numéro MAC de la machine, un nom et le nombre modem branché sur cette machine. Une fois que l'administrateur a saisi ces informations, il peut démarrer ou arrêter le group et suivre les procédures d'envoie grâce a la rubrique monitoring (voir annexe 1).

B) RUBRIQUE « MONITORING 2 »

Le monitoring est l'anglicisme du terme surveillance. Il définit la mesure d'une activité (humaine, économique, électrique, d'un organe, etc.). Dans ce cadre, cette rubrique « monitoring » est la partie la plus importante de ce projet car elle permet de suivre les opérations effectuées par les groupes en temps réel. (Voir annexe 2)

Dans cette partie « monitoring », il y a deux parties différentes : une sur l'information sur les « groupes » et une sur l'information des modems par « groupes ».

Information sur les « groupes » :

Dans cette sous-partie, nous affichons quelques informations importantes telles que le numéro MAC du « groupe » (ordinateur), Alias liée au « group », le nombre de modems branchés sur le « groupe », l'état du « groupe », la vitesse du « groupe », les sms envoyés, les sms reçus, le pourcentage reçu, les sms en attente et les actions liées au « group ».

Id	Numéro MAC	Alias MAC	Nombre de modem	Etat	Vitesse	Sms envoyé	Sms reçu	% Reçu	Sms en queue	Action
1	64:31:50:1b:e9:e9	machine 1	1	demarre	0	0		0	0	Démarrer Arrêter Détails
2	e8:11:32:e9:ab:36	notebook	2	arrete	0	0		0	0	Démarrer Arrêter Détails
3	00:15:c5:b4:1c:8a	porta_tony	2	demarre	0	0		0	0	Démarrer Arrêter Détails

Figure 1 : Information sur le « groupe »

L'administrateur peut arrêter ou démarrer le « groupe » et, en même temps, il a la possibilité de modifier les informations du « groupe » telles que alias MAC, le nombre de modems, la vitesse ou même supprimer le « groupe » définitivement.

machine 1	
Numero MAC	64:31:50:1b:e9:e9
Alias MAC	machine 1
Nombre de modem	1
Vitesse	0
Etat	demarre
<div> <div>Editer</div> <div>Supprimer</div> </div>	

Figure 2 : Suppression ou modification du « groupe »

machine 1	
Numero MAC	64:31:50:1b:e9:e9
Alias MAC	machine 1
Nombre de modem	1
Vitesse	0
Etat	demarre
Appliquer	

Figure 3 : Modification du « group »

### Information des modems par « groupe »

Dans cette sous-partie, nous affichons les informations sur les modems par « group ». Les informations affichées sont : le numéro IMEI du modem, l'opérateur, l'état, les sms envoyés par le modem, les sms reçus, le pourcentage reçu et les sms en queue.

Info modems par groups							
machine 1							
Id	Numero IMEI	Opérateur	Etat	Sms envoyé	Sms reçu	% Reçu	Sms en queue
1	358191012692190	orange	on				
notebook							
Id	Numero IMEI	Opérateur	Etat	Sms envoyé	Sms reçu	% Reçu	Sms en queue
1	355083001234480	orange	on				
2	355083001234630	orange	on				
porta_foxy							
Id	Numero IMEI	Opérateur	Etat	Sms envoyé	Sms reçu	% Reçu	Sms en queue
1	355083001234780	orange	on				
2	355083001234410	indisponible	off				

Figure 4 : Information des modems par groupe

## 4) GROUPES

Comme nous en avons parlé précédemment, un « groupe » est un ensemble de modems branchés sur un ordinateur.

Sur cette machine là, nous allons installer l'application qui gère l'envoi des sms. Pour envoyer des sms, nous avons que deux choix : l'utilisation d'un serveur smsc ou l'utilisation d'un modem GSM.

### A) QU'EST CE QU'UN SERVEUR SMSC ?

Un serveur SMSC (*Short Message Service Center*) est un serveur qui permet de gérer le transfert de message SMS (textes ou binaires) entre des téléphones mobiles. En particulier, quand un abonné envoie un SMS vers un autre, le téléphone transmet en réalité le SMS vers le SMSC. Le SMSC stocke le message puis le transmet au destinataire lorsque celui-ci est présent sur le réseau (mobile allumé) : Le SMSC fonctionne sur le mode Store & Forward.

Un SMSC joue également le rôle de « passerelle » entre le réseau IP et le réseau mobile. En particulier, un serveur peut y accéder par connexion Internet TCP afin d'envoyer des SMS vers des numéros MSISDN de destination.

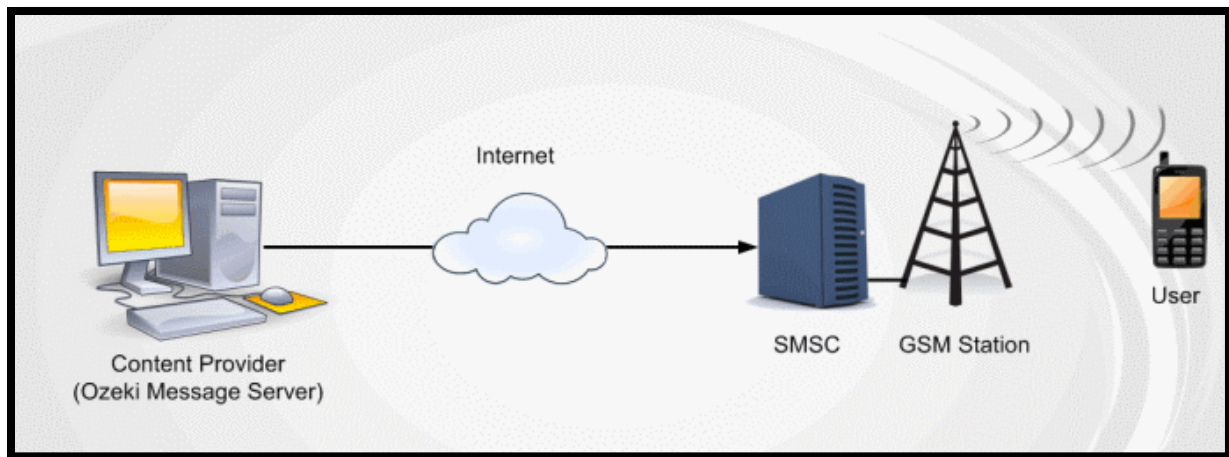


Figure 1 : Fonctionnement d'un serveur SMSC

Pour communiquer entre les serveurs et le SMSC, en TCP/IP, les protocoles les plus utilisés sont SMPP et CMG EMI. Le SMPP est un protocole standard d'échange qui permet d'envoyer des SMS vers des opérateurs téléphoniques.

Avoir un serveur SMSC est très coûteux et il doit aussi avoir l'accord d'un opérateur téléphonique comme Orange, SFR, etc...Donc nous avons choisi d'utiliser des modem GSM.

#### B) QU'EST CE QU'UN MODEM GSM ?

Un modem GSM est un modem qui permet la transmission des données sur le réseau de téléphonie mobile GSM. Il peut se présenter sous diverses formes :

- Modem logiciel, à installer dans l'ordinateur auquel est relié le téléphone mobile;
- Carte PMCIA, à insérer dans le slot approprié d'un ordinateur portable ;
- Modem intégré dans le téléphone (Tester avant tout achat et s'assurer de disposer des pilotes).

Dans ce projet nous avons utilisé le modem suivant : Huawei E220



Figure 2 : Huawei E220



Ce modem est fabriqué par le fameux fabricant chinois Huawei. Le modem Huawei E220 est un modem USB. E220 fonctionne bien avec Linux, surtout que son soutien a été ajouté au noyau Linux 2.6.20. Et il y a des solutions de contournement pour les distributions avec les anciens noyaux.

En plus de l'antenne cellulaire, le modem Huawei E220 dispose aussi d'une mémoire de stockage de 22 Mo (10 Mo sur les versions plus anciennes) accessible par le système d'exploitation comme un périphérique de stockage de masse et contient le driver d'installation du modem conçu par les opérateurs mobiles 3G.

Les modems Huawei sont conçus pour fonctionner sous Windows et Linux. Sachant que la plupart des clients de KT CENTREX utilise Linux, OS du programme est imposé. Tous les scripts de ce rapport fonctionnent uniquement sous Linux. Les modems Huawei sont très simples à installer sous Linux. Il suffit juste de brancher les modems sur le système grâce au mini câble USB, on n'a pas besoin d'installer manuellement les pilotes.

Une fois que le modem est branché sur le système, il ouvre deux ports USB virtuels. Exemple: si on branche un seul modem on aura deux ports virtuels : /dev/ttyUSB0 et /dev/ttyUSB1. Le premier port est utilisé pour l'envoi de sms et le deuxième port est utilisé pour la réception de sms. On peut utiliser les programmes comme Hyperterminal (Windows) GTKTerm (Linux) pour établir la communication avec le modem afin d'exécuter les commandes AT.

### C) LES COMMANDES AT

Les commandes AT sont définies dans la norme GSM 07.07 (pour les SMS). AT est l'abréviation d'ATtention. Ces 2 caractères sont toujours présents pour commencer une ligne de commande sous forme de texte (codes ASCII). Les commandes permettent la gestion complète du mobile ou un modem GSM.

Trois entités sont définies :

- TE : Terminal Equipment (envoi et affiche les commandes)
- TA : Terminal Adaptateur (interface entre l'utilisateur et le mobile)
- ME : Mobile Equipment

#### Schéma de fonctionnement

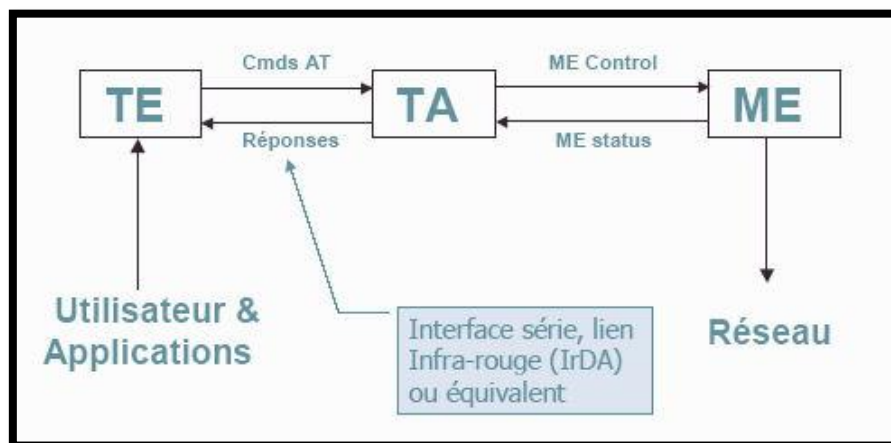


Figure 3 : Schéma de fonctionnement

### Comment établir la connexion avec le modem?

Une fois le modem branché sur la machine, nous pouvons vérifier la connectivité à l'aide de la commande Linux suivant : `ls /dev/ | grep ttyUSB`  
Lorsque la vérification terminée, il faut lancer le programme « GTKTerm » (équivalent à Hyperterminal sous Windows). Ensuite on définit les paramètres de connexion.

Bits par seconde : 9600

Bits de données : 8

Parité et bits d'arrêt : à laisser par défaut

Matériel : Xon / Xoff

### *Exemple AT commande*



Figure 4 : Exemple AT commande sur Hyperterminal

Quelque AT command :

Pour connaître le statut d'enregistrement du réseau GSM :

3.5.4.3.3. Network Registration Report - +CREG	
+CREG - Network Registration Report	SELINT 0 / 1
AT+CREG=[<mode>]	<p>Set command enables/disables network registration reports depending on the parameter &lt;mode&gt;.</p> <p>Parameter: &lt;mode&gt;            0 - disable network registration unsolicited result code (factory default)            1 - enable network registration unsolicited result code            2 - enable network registration unsolicited result code with network Cell identification data</p> <p>If &lt;mode&gt;=1, network registration result code reports:</p> <p>+CREG: &lt;stat&gt;</p> <p>where            &lt;stat&gt;            0 - not registered, ME is not currently searching a new operator to register to            1 - registered, home network            2 - not registered, but ME is currently searching a new operator to register to            3 - registration denied            4 - unknown            5 - registered, roaming</p> <p>If &lt;mode&gt;=2, network registration result code reports:</p> <p>+CREG: &lt;stat&gt;[,&lt;Lac&gt;,&lt;Ci&gt;]</p> <p>where:            &lt;Lac&gt; - Local Area Code for the currently registered on cell</p>

Envoie d'un sms :

Pour envoyer un sms, tout d'abord, il faut définir le format d'envoi. Les modems Huawei supportent deux formats : PDU et Texte. Dans notre cas nous utiliserons le format Texte.

Afin de choisir le format Texte, on utilise la commande suivante : AT+CMGF = 1  
 Une fois le format bien défini, on peut envoyer un sms grâce à la commande suivante : AT+CMGS=0650908233

Ensuite si le modem répond avec « > » un symbole supérieur, on écrit le message à envoyer sur le terminal et on valide l'envoi par « CTRL+Z ».

Nous avons vu comment envoyer un sms avec un modem GSM en utilisant les commandes AT depuis un programme comme Hyperterminal.

Maintenant nous allons voir comment envoyer un sms depuis un programme/script écrit en langage Perl.

#### D) ENVOI DE SMS DEPUIS UN SCRIPT PERL

Il existe un module de Perl Device:SerialPort qui permet d'établir une connexion au port COM. J'ai utilisé CPAN pour installer ce module.

Je vais vous expliquer avec un exemple de code la connexion avec le modem depuis un script Perl.

La connexion se fait avec la ligne suivante, où \$ob est un objet :

```
$ob=Device::SerialPort->new("/dev/ttyUSB$_[0]");
```

Une fois la connexion établie, nous pouvons utiliser l'objet \$ob afin de communiquer avec le modem. La méthode « *write* » permet d'envoyer les caractères ASCII vers le modem et la méthode « *input* » permet de récupérer la réponse du modem.

J'ai défini deux fonctions « *use\_device* » et « *at\_command* » afin de faciliter la programmation. La fonction « *use\_device* » prend en argument le numéro port de l'USB et il effectue la connexion sur ce port. La fonction « *at\_command* » prend en argument les commandes AT et retourne la réponse liée à cette commande.

La fonction « *use\_device* » :

```
use strict;
use warnings;
use Device::SerialPort;

my $ob;

#ici je déclare la fonction permet d'établir la connexion avec les modem
sub use_device {
    $ob = Device::SerialPort->new("/dev/ttyUSB$_[0]");
}
```

La fonction « *at\_command* » :

```
sub at_command {
    my $flag = 0;
    my @rep;
    my $rep;
    my $i = 0;
    my $var;
    $ob->write($_[0]);

    while ($flag == 0){
        $rep[$i] = $ob->input;
        if ($rep[$i] =~ /OK/){
            $flag = 1;
        }
        if ($rep[$i] =~ />/){
            $flag = 1;
        }
        if ($rep[$i] =~ /NOT/){
            $flag = 1;
            print "ATCommande invalide\n";
        }
        if ($rep[$i] =~ /ERROR/){
            $flag = 1;
            #print "Erreur\n";
        }
        if ($i == 1320735){
            $flag = 1;
        }
        $i++;
    }

    foreach $var (@rep){
        $rep .= $var ;
    }

    return $rep;
}
```

Nous avons vu les fonctions qui permettent d'effectuer la connexion avec le modem et la fonction qui permet d'exécuter les commandes AT.

Maintenant nous allons voir un script du Perl qui permet d'envoyer un sms. Pour cela, nous allons utiliser les deux fonctions dont on a parlé précédemment.

```
1.use_device(0);
2.
3.my $rep = at_command("AT+CMGF=1\r");
4.
5.my $rep = at_command("AT+CMGS=\"0650908233\"\r");
6.if ($rep == '>'){
7.    my $rep = at_command("C'est un message de test".\cZ\r");
8.    if ($rep =~ /OK/){
9.        print "Message est envoyé.";
10.    }
11.
12.    elsif ($rep =~ /500/){
13.        print "Pas de credit! :\n";
14.    }
15.
16.    else {
17.        print "Erreur d'envoi! :\n";
18.    }
19.    }
20.else{
21.    print "Erreur d'envoi!\n"; at_command("\e");
22.}
```

**Explication:**

Ligne 1 : nous appelons la fonction « *use\_device* » en lui fournissant un argument. Cet argument correspond au numéro port du modem

Ligne 3 : cette ligne permet d'initialiser le format du message. La commande « AT+CMGF=1 » correspond au format texte du message

Ligne 5 : on saisit le numéro du destinataire

Ligne 6 : si le modem répond avec un symbole supérieur, on écrit le message à envoyer

Ligne 7 : on écrit le message et on valide par « Ctrl+Z »

Ligne 8 : si tout se passe bien le modem répond « OK »

Ligne 9 : si la réponse du modem est « OK », on confirme l'envoi du sms

Ligne 12 : si la réponse est une erreur « 500 »

Ligne 13 : on affiche un message

Jusqu'à maintenant nous avons vu comment effectuer la connexion avec le modem et comment envoyer un sms depuis un script du Perl.

Maintenant nous allons aller un peu plus loin, nous allons voir comment récupérer les campagnes lancées par un client et comment effectuer les envois de sms si on a plusieurs modems.

#### F) L'APPLICATION SMSOFT

L'application « smssoft » est l'ensemble du script Perl qui permet d'effectuer les envois de message et de faire le monitoring sur les modems. L'ensemble du script doit être placé dans un seul et unique répertoire : /home/script/perl/sms. Certain script doit être lancé au démarrage du « groupe » (ordinateur). Pour cela nous devons ajouter la ligne suivante dans le fichier « /etc/rc.local » : /home/script/perl/sms/'nom de script'.

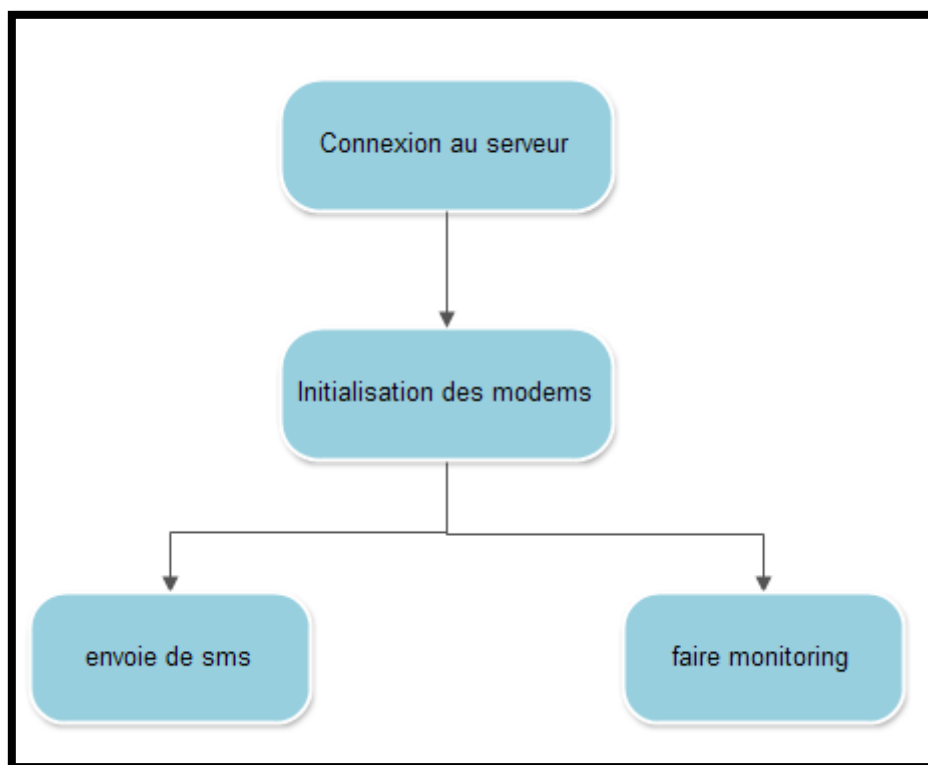


Figure 5 : Schéma structurel du « groupe »

Il y a 3 étapes importantes sur le « groupe » : « connexion au serveur », « envoi de sms », « faire monitoring ». La « connexion au serveur » permet d'effectuer la communication avec le serveur distant et de récupérer les paramètres saisis par l'administrateur. Les éléments à récupérer sont : le numéro MAC, l'alias MAC et le nombre de modem. Une fois que le « groupe » a récupéré ces informations, le smssoft initialise les modems branchés sur la machine. Une fois l'initialisation terminée, le « groupe » commence à envoyer les sms et faire le monitoring, autrement dit, à écrire les informations et les défauts des modems sur l'interface de l'administrateur.

Nous avons vu le fonctionnement global du « groupe » (smssoft). Maintenant nous allons voir en détails la partie « envoi de sms » et « faire monitoring ».

## Envoie de sms

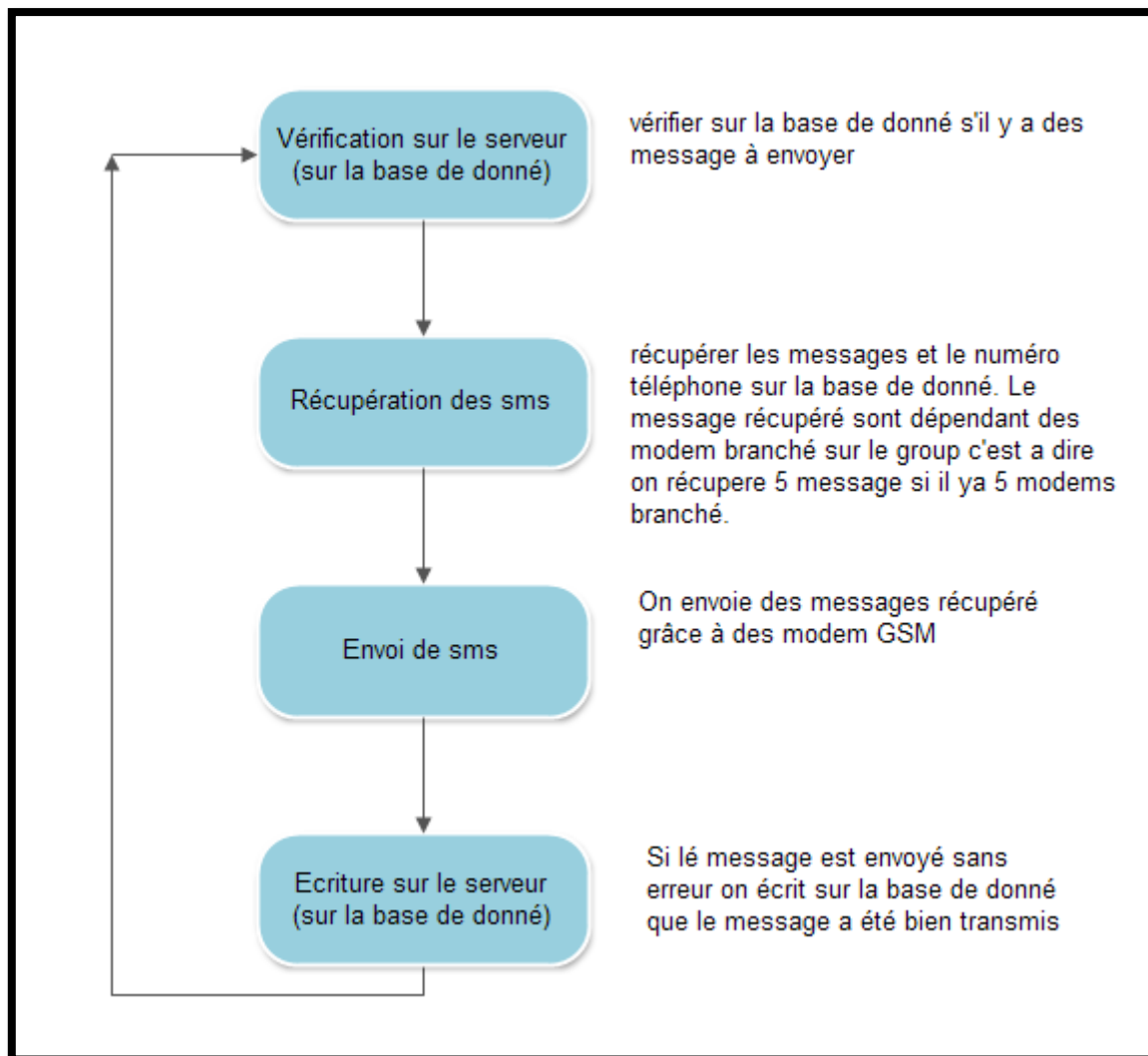


Figure 6 : Schéma structurel d'envoi de sms

Lorsque le client active la campagne, il y a un script sur le serveur qui écrit les messages à envoyer sur la base de données avec le numéro de téléphone. Une fois cette étape terminée, le « groupe », connecté sur le serveur, vérifie s'il y a des sms à envoyer. S'il trouve les messages, il les récupère grâce aux requêtes SQL, ensuite il effectue l'envoi des messages récupérés. Si le message est envoyé sans erreur, on écrit sur la base de données que le message est envoyé et on écrit aussi le numéro IMEI du modem par lequel le message est envoyé.

J'ai utilisé les threads en Perl pour effectuer les envois parallèles. Autrement dit, si on branche 5 modems sur le « groupe », lorsque le groupe récupère les messages, il récupère uniquement les 5 premiers messages, ensuite il envoie les messages en parallèle.

## Monitoring

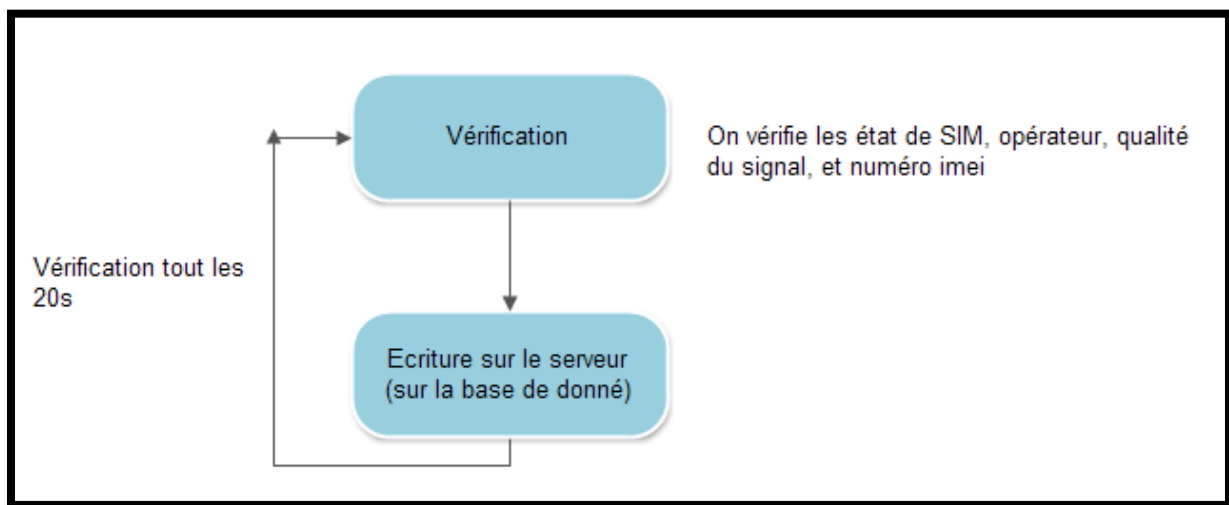


Figure 7 : Schéma structurel de monitoring

La partie « monitoring » se fait toutes les 20 secondes. Le but de cette partie est de collecter les informations concernant les modems et d'afficher sur le panel de l'administrateur. Les informations collectées sont : le numéro IMEI, l'état de la puce, l'opérateur et la qualité du signal. J'ai défini les fonctions en Perl qui permet de collecter ces informations.

Maintenant, je vais vous expliquer les fonctions que j'ai définies pour faire le « monitoring ».

### Les fonctions

Une fonction pour connaître le numéro IMEI :

```
sub imei{
    my $imei = at_command("AT+CGSN\r");
    my @imei = split(' ', $imei);
    my $size = @imei;
    for (my $i=0; $i<=$size; $i++) {
        if (length($imei[$i]) == 15){
            $imei = $imei[$i];
        }
    }

    return $imei;
}
```

Cette fonction « imei » récupère le numéro IMEI du modem grâce à la commande AT suivant : AT+CGSN.



Une fonction pour connaître le nom de l'opérateur :

```
sub nom_opérateur{
    my $nom_op;
    my %op_hash =
(20820,'bouygues',20810,'sfr',20801,'orange');
    my $num_op = at_command("AT+COPS?\r");
    my @num_op = split(' ', $num_op);
    my $size = @num_op;
    for (my $i=0; $i<=$size; $i++) {
        if (length($num_op[$i]) == 13){
            $num_op = $num_op[$i];
        }
    }

    @num_op = split(',', $num_op);
    if ($num_op[2] =~ /\r/){
        my @tmp = split(' ', $num_op[2]);
        $num_op[2] = $tmp[1];
    }

    if ($op_hash{$num_op[2]}){ $nom_op =
$op_hash{$num_op[2]} }
    else {$nom_op = "indisponible";}

    return $nom_op;
}
```

Cette fonction « *nom\_opérateur* » permet de connaître l'opérateur sur lequel la carte SIM est enregistrée. J'ai récupéré le numéro de l'opérateur grâce à la commande AT « AT+COPS ? » et ensuite j'ai retrouvé le nom de l'opérateur à l'aide de hash « %op ». Si la carte SIM n'est pas enregistrée auprès d'un opérateur, la fonction retourne « indisponible ».

Une fonction pour connaître la qualité du signal :

3.5.4.4. Signal Quality - +CSQ	
+CSQ - Signal Quality	SELINT 0 / 1
AT+CSQ	<p>Execution command reports received signal quality indicators in the form:</p> <p><b>+CSQ: &lt;rss&gt;,&lt;ber&gt;</b>            where  <b>&lt;rss&gt;</b> - received signal strength indication            0 - (-113) dBm or less            1 - (-111) dBm            2..30 - (-109)dBm..(-53)dBm / 2 dBm per step            31 - (-51)dBm or greater            99 - not known or not detectable</p> <p><b>&lt;ber&gt;</b> - bit error rate (in percent)            0 - less than 0.2%            1 - 0.2% to 0.4%            2 - 0.4% to 0.8%            3 - 0.8% to 1.6%            4 - 1.6% to 3.2%            5 - 3.2% to 6.4%            6 - 6.4% to 12.8%            7 - more than 12.8%            99 - not known or not detectable</p> <p>Note: this command should be used instead of the %Q and %L commands, since GSM relevant parameters are the radio link ones and no line is present, hence %Q %L and have no meaning.</p>

La commande « AT+CSQ » permet de connaître la qualité du signal en dBm et les bits erreurs. Dans notre cas, nous avons juste besoin de la qualité du signal.

D'après la documentation de « AT commande », la qualité du signal varie entre 0 et 31. Si la qualité n'est pas détectable, le terminal répond avec <rss> = 99.

J'ai défini les intervalles pour faciliter la lecture des informations sur la qualité du signal.

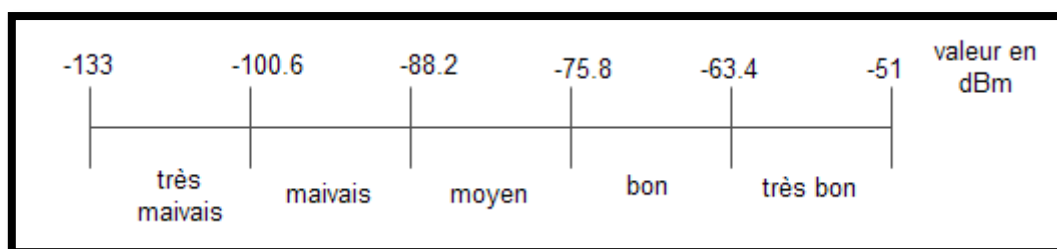


Figure 8 : intervalle de la qualité du signal en dBm

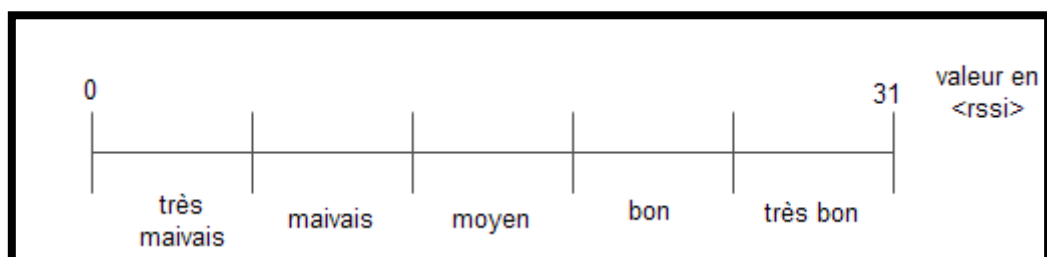


Figure 8 : Intervalle de la qualité du signal en <rss>

Les valeurs <rss> et dBm ne sont pas proportionnelles donc pour trouver la formule pour effectuer la conversion, je dois tracer la courbe dBm en <rss>.

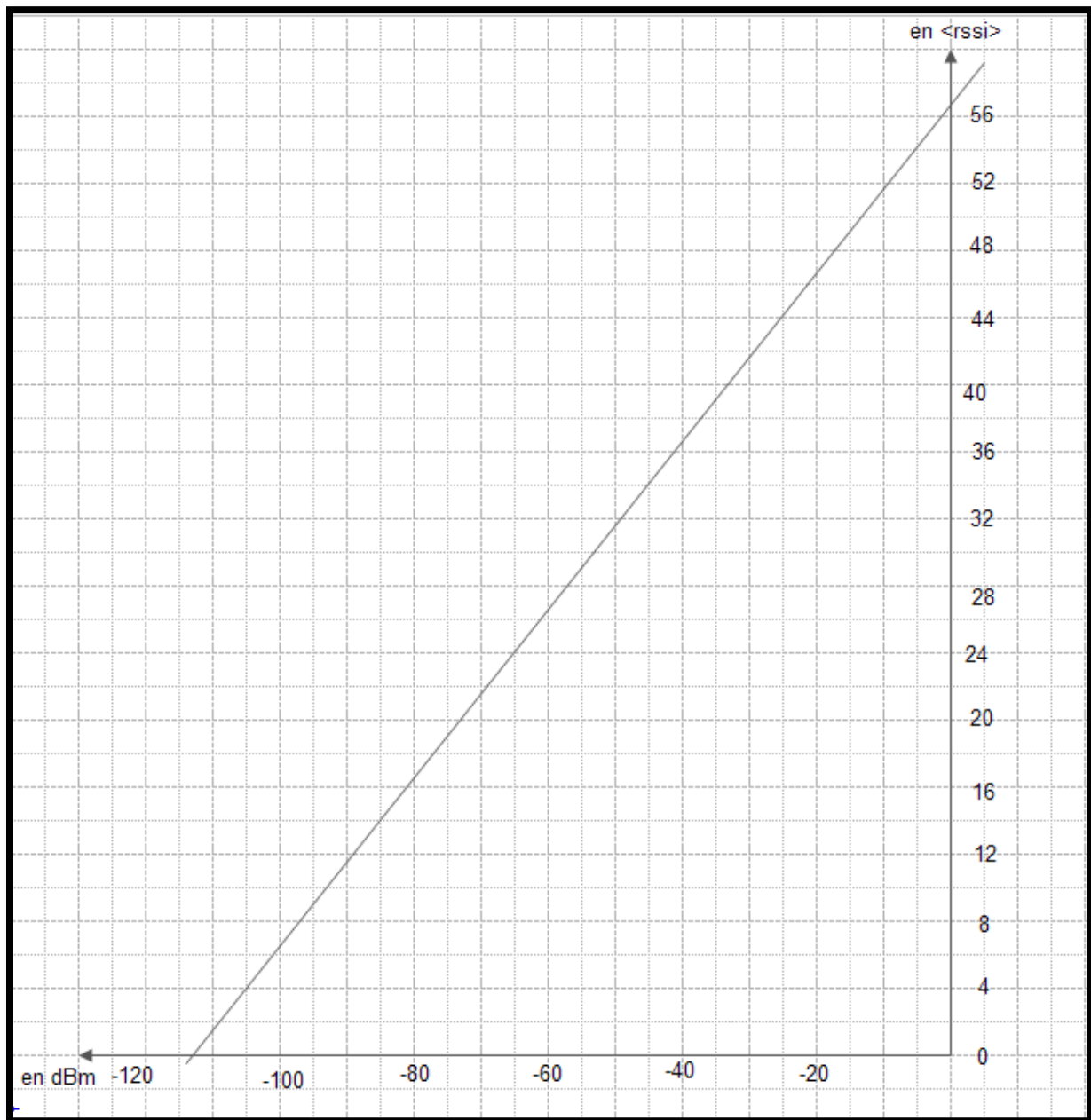


Figure 9 : Courbe de la qualité du signal

**Calcul :**

$$f(x) = ax + b$$

$$a = \frac{31 - 0}{-51 - (-133)} = \frac{31}{62} = 0.5$$

$$b = x - ax = x(1 - a) = -133(1 - 0.5) = 56.25$$

$$\text{Donc l'équation de la droite est : } f(x) = \frac{x}{2} + 56.25$$

$$\text{donc la formule de conversion est : } \text{dBm} = (<\text{rss}> - 56.25) * 2$$

Voici le programme de Perl qui permet de trouver la qualité du signal :

```
sub qualite_signal{
    my $qs_rssi = at_command("AT+CSQ\r");
    my @qs_rssi = split(' ', $qs_rssi);
    my $size = @qs_rssi;
    for (my $i=0; $i<=$size; $i++) {
        unless ($qs_rssi[$i] =~ /[A-Z]/){
            if ($qs_rssi[$i] =~ /\./) { $qs_rssi =
$qs_rssi[$i]; }

        }

    }
    @qs_rssi = split(',', $qs_rssi);
    $qs_rssi = $qs_rssi[0];
    my $qs_dbm = ($qs_rssi-56)*2;
    my $quality;

    if ($qs_rssi == 99) {
        $quality = "introuvable"
    }
    if (($qs_dbm > -113) && ($qs_dbm < -100.6)) {
        $quality = "tres mauvais";
    }

    elsif (($qs_dbm > -100.6) && ($qs_dbm < -88.2)) {
        $quality = "mauvais";
    }

    elsif (($qs_dbm > -88.2) && ($qs_dbm < -75.8)) {
        $quality = "moyen";
    }

    elsif (($qs_dbm > -75.8) && ($qs_dbm < -63.4)) {
        $quality = "bon";
    }

    elsif (($qs_dbm > -63.4) && ($qs_dbm < -51)) {
        $quality = "très bon";
    }

    return $quality;
}
```

Et une autre fonction pour déterminer l'état de la carte SIM :

```
sub etat {
    my %etat_hash =
    (0,'not_registered',1,'registered_HN',2,'not_registered',3,'re
gistration_denied',4,'unknown',5,'registered_roaming');
    my $etat;
    my @etat;
    $cpin = at_command("AT+CPIN?\r");
    if ( $cpin =~ /13/){$etat = 'SIM failure';}
    elsif ( $cpin =~ /failure/){$etat = 'SIM failure';}
    elsif ( $cpin =~ /READY/) {
        $etat = at_command("AT+CREG?\r");
        @etat = split(',', $etat);
        my $size = @etat;
        for (my $i=0; $i<=$size; $i++) {
            unless ($etat[$i] =~ /[A-Z]/){
                if ($etat[$i] =~ /,/){ $etat =
$etat[$i]; }
            }
        }
        @etat = split(' ', $etat[1]);
        $etat = $etat_hash{$etat[0]};
    }

    else {$etat = 'Erreur!';}
    return $etat;
}
```

Pour vérifier si la carte SIM est présente dans l'appareil, nous utilisons la commande suivante : « AT+CPIN ? ». Si le terminal nous répond « Error 13 » cela signifie l'absence de la puce dans le modem. Si le terminal nous répond « SIM faillure » cela signifie que la puce est défectueuse.

Si le terminal nous répond « READY » cela veut dire que la carte SIM est bien installée dans le modem donc on peut vérifier l'état de l'enregistrement. Pour vérifier l'état de l'enregistrement, nous utilisons la commande suivante : « At+CREG ? ».

## 3.5.4.3.3. Network Registration Report - +CREG

+CREG - Network Registration Report	SELINT 0 / 1
AT+CREG[= [<mode>]]	<p>Set command enables/disables network registration reports depending on the parameter &lt;mode&gt;.</p> <p>Parameter: &lt;mode&gt;            0 - disable network registration unsolicited result code (factory default)            1 - enable network registration unsolicited result code            2 - enable network registration unsolicited result code with network Cell identification data</p> <p>If &lt;mode&gt;=1, network registration result code reports:</p> <p>+CREG: &lt;stat&gt;</p> <p>where &lt;stat&gt;            0 - not registered, ME is not currently searching a new operator to register to            1 - registered, home network            2 - not registered, but ME is currently searching a new operator to register to            3 - registration denied            4 - unknown            5 - registered, roaming</p> <p>If &lt;mode&gt;=2, network registration result code reports:</p> <p>+CREG: &lt;stat&gt;[,&lt;Lac&gt;,&lt;Ci&gt;]</p> <p>where: &lt;Lac&gt; - Local Area Code for the currently registered on cell</p>

Suite à la réponse du terminal pour la commande « AT+CREG ? », on va pouvoir déterminer le statut d'enregistrement sur le réseau. J'ai utilisé la documentation ci-dessus pour déterminer le statut d'enregistrement.

Nous avons fait le tour des fonctions, grâce à ces fonctions je récupère les informations concernant le modem et la carte SIM. Une fois ces informations récupérées, j'ai utilisé les requêtes SQL afin d'écrire ces informations dans une base de données sur laquelle les scripts du serveur récupèrent les informations pour faire le monitoring.

Jusqu'à maintenant nous avons vu comment fonctionne les scripts sur un « groupe » et nous avons vu en détails les fonctions permettant de faire le monitoring des modems. Pour faciliter l'utilisation de l'application smssoft, nous avons décidé de faire une interface graphique. Cette interface graphique est faite par un autre stagiaire. Le but de cette interface est de faire le lien entre les commandes de l'interface (buttons) et les scripts de Perl.



Figure 10 : Interface graphique de smssoft

#### *Explication :*

Cette interface facilite le client à se connecter ou se déconnecter du serveur. L'adresse du serveur doit être renseignée par le client. Et on affiche l'adresse MAC du « groupe » afin de faciliter l'inscription du « groupe » sur le serveur à distant.

#### *Installation du smssoft :*

Pour faciliter l'installation de l'application, nous avons fait un script de bash qui permet d'installer les Library et de créer les répertoires et les fichiers pour le fonctionnement de l'application. Donc le client de KT-CENTREX peut télécharger cette application sur internet au format tar.gz. Ensuite, il décompresse le fichier et lance le script « installe.sh ».

---

## PROBLEMES RENCONTRES

Durant ce projet, j'ai rencontré des problèmes liés à la programmation. Surtout lors de l'installation des modules de Perl. Lorsqu'on installe les modules de Perl manuellement, j'ai trouvé des erreurs de version. Donc j'ai utilisé CLI de CPAN pour installer les modules.

Pour rentrer dans Shell de CPAN, j'ai utilisé la commande suivante : `perl -MCPAN -e shell`

Cette ligne de commande permet de rentrer dans CLI de CPAN. Une fois que je suis rentré dans CLI, j'utilise la commande suivante pour installer les modules : `install « nom de module »`

Pendant mon stage dans l'entreprise, nous avons des problèmes de connexion internet. Donc nous avons décidé d'installer le serveur en local. Pour cela, nous avons installé un serveur LAMP sur une machine. Cette opération m'a donné une chance pour installer un serveur web sur une machine et de le configurer correctement. Suite à cet événement j'ai appris à installer et à configurer un serveur apache, à installer et à configurer PHP, à installer et à configurer la base de données MySQL et phpmyadmin.

Lors de mon stage, je n'ai pas que travaillé sur ce projet, j'ai travaillé sur le CRM qui existait déjà. Sur ce CRM, il y avait des « bugs » sur certaines pages donc j'ai essayé de les corriger. Quand j'ai travaillé sur le CRM, j'ai appris à utiliser Html, PHP JavaScript et Ajax. Lorsque j'ai travaillé sur ce CRM, j'ai rencontré des problèmes sur les langages de programmation mais grâce à des nombreux cours qui existent sur internet, j'ai trouvé rapidement des solutions à ce problème.



---

## CONCLUSION

Pendant le déroulement de mon stage, j'ai eu l'opportunité de travailler sur des différentes problématiques. Le travail réalisé s'est avéré très enrichissant pour mon expérience professionnelle. Le fait de travailler avec les différents langages de programmation, m'a permis d'avoir une notion sur la programmation et sur le développement.

En effet, le projet smsswitch m'a permis de bien organiser les différents travaux, à bien faire les algorithmes avant de coder et bien planifier préalablement les tâches. Il m'a aussi donné l'occasion de travailler en équipe sur un projet.

D'autre part grâce à des problèmes liés à la connexion internet, j'ai appliqué la notion du réseau telle que la configuration d'un Switch, l'adressage d'IP, la transformation d'une machine en routeur etc...Et aussi cela m'a permis d'installer et configurer un serveur LAMP.

Le fait de travailler en équipe et d'utiliser des applications existantes m'a permis de m'intégrer dans un groupe de travail et de voir en quoi consistait le travail d'ingénieur.

# INDEX

CPAN .....	17, 30
CRM .....	30
CSV .....	8
GSM .....	7
GSM .....	13, 14, 15
IMEI .....	13, 21, 22
IPBX .....	5
LAMP .....	30, 31
MAC .....	12, 20, 29
PDU .....	17
SIM .....	23, 27, 28
SMSC .....	13, 14
TCP/IP .....	14

## Annexe 1 : Rubrique PARAMETRAGE DU GROUP

PARAMETRAGES																																		
Facturation	Paramètre Société	Template	Connexion	Gestion Campagne	Paramètre du group																													
<p>Numero de MAC non valide OU nombre modem non valide</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 5%;">Id</th> <th style="width: 20%;">Numéro MAC</th> <th style="width: 15%;">Alias MAC</th> <th style="width: 15%;">Nombre de modem</th> <th style="width: 10%;">Etat</th> <th style="width: 10%;">Vitesse</th> <th style="width: 15%;">Action</th> </tr> </thead> <tbody> <tr> <td>57</td> <td>64:31:50:1b:e9:e9</td> <td>machine 1</td> <td>1</td> <td>demarre</td> <td>0</td> <td> <div style="display: flex; justify-content: space-around;"> <div>Démarrer</div> <div>Arrêter</div> <div>Détails</div> </div> </td> </tr> <tr> <td>55</td> <td>e8:11:32:c9:ab:36</td> <td>notebook</td> <td>2</td> <td>arrete</td> <td>0</td> <td> <div style="display: flex; justify-content: space-around;"> <div>Démarrer</div> <div>Arrêter</div> <div>Détails</div> </div> </td> </tr> <tr> <td>58</td> <td>00:15:c5:b4:1c:8a</td> <td>porta_touy</td> <td>2</td> <td>demarre</td> <td>0</td> <td> <div style="display: flex; justify-content: space-around;"> <div>Démarrer</div> <div>Arrêter</div> <div>Détails</div> </div> </td> </tr> </tbody> </table>							Id	Numéro MAC	Alias MAC	Nombre de modem	Etat	Vitesse	Action	57	64:31:50:1b:e9:e9	machine 1	1	demarre	0	<div style="display: flex; justify-content: space-around;"> <div>Démarrer</div> <div>Arrêter</div> <div>Détails</div> </div>	55	e8:11:32:c9:ab:36	notebook	2	arrete	0	<div style="display: flex; justify-content: space-around;"> <div>Démarrer</div> <div>Arrêter</div> <div>Détails</div> </div>	58	00:15:c5:b4:1c:8a	porta_touy	2	demarre	0	<div style="display: flex; justify-content: space-around;"> <div>Démarrer</div> <div>Arrêter</div> <div>Détails</div> </div>
Id	Numéro MAC	Alias MAC	Nombre de modem	Etat	Vitesse	Action																												
57	64:31:50:1b:e9:e9	machine 1	1	demarre	0	<div style="display: flex; justify-content: space-around;"> <div>Démarrer</div> <div>Arrêter</div> <div>Détails</div> </div>																												
55	e8:11:32:c9:ab:36	notebook	2	arrete	0	<div style="display: flex; justify-content: space-around;"> <div>Démarrer</div> <div>Arrêter</div> <div>Détails</div> </div>																												
58	00:15:c5:b4:1c:8a	porta_touy	2	demarre	0	<div style="display: flex; justify-content: space-around;"> <div>Démarrer</div> <div>Arrêter</div> <div>Détails</div> </div>																												
<p><b>Ajouter un group:</b> <input style="width: 150px;" type="text" value="Numero MAC"/> <input style="width: 150px;" type="text" value="Alias MAC"/> <input style="width: 150px;" type="text" value="Nombre de modem"/> <input style="width: 50px;" type="button" value="Ajouter"/></p>																																		

## ANNEXE 2 : RUBRIQUE MONITORING

Info groups

Id	Numero MAC	Alias MAC	Nombre de modem	Etat	Vitesse	Sms envoyé	Sms reçu	% Recu	Sms en queue	Action		
1	64:31:50:1b:e9:e9	machine 1	1	demarre	0	0		0	0	<div>Démarrer</div>	<div>Arrêter</div>	<div>Détails</div>
2	e8:11:32:c9:ab:36	notebook	2	arrete	0	0		0	0	<div>Démarrer</div>	<div>Arrêter</div>	<div>Détails</div>
3	00:15:e5:b4:1c:8a	porta_tony	2	demarre	0	0		0	0	<div>Démarrer</div>	<div>Arrêter</div>	<div>Détails</div>

Info modems par groups

machine\_1

Id	Numero IMEI	Opérateur	Etat	Sms envoyé	Sms reçu	% Recu	Sms en queue
1	358191012692190	orange	on				

notebook

Id	Numero IMEI	Opérateur	Etat	Sms envoyé	Sms reçu	% Recu	Sms en queue
1	355083001234480	orange	on				
2	355083001234630	orange	on				

porta\_tony

Id	Numero IMEI	Opérateur	Etat	Sms envoyé	Sms reçu	% Recu	Sms en queue
1	355083001234780	orange	on				
2	355083001234410	indisponible	off				

## ANNEXE A : SCRIPT DE CONFIRMATION

Ce script permet de traiter les 5 premières lignes de la base Excel et demande de confirmer les colonnes au client.

```
use DBI;
use Spreadsheet::ParseExcel;
use Spreadsheet::XLSX;
require "connect_base.pl";

#récupérer les infos sur la base "tableatt"
my $dbh = connect_db();

while(1){

    #Initialisation des variables...
    my $i = 0;
    my $max_ligne = 4;
    my @tab_full;
    my @cell_vide;
    my $tmp;
    my $TotalCol;
    my $TotalRow;
    my $length_tab;
    my $j;

    print "Vérification...\n";
    my $sth = send_prepare("SELECT * FROM tableatt WHERE etat = 'att'");
    while(my @row = $sth->fetchrow_array) {
        $id = $row[0];
        $nom_base = $row[2];
        $nom_fichier = $row[3];
        $logine_client = $row[4];
    }

    if($id){
        $repertoire = '/home/ktcentrex/prepays_cards/adminclient/monupload/';
        system "chmod -R 777 /home/ktcentrex/prepays_cards/adminclient/monupload/";
        print "Fichier à analyser : $nom_fichier\n";

        #Traitement du fichier XLSX
        if ($nom_fichier =~ /\.xlsx/){
            print "Fichier xlsx\n";
            my $excel = Spreadsheet::XLSX -> new ($repertoire . $nom_fichier);
            foreach my $sheet (@{$excel -> {Worksheet}}) {
                $sheet -> {MaxRow} ||= $sheet -> {MinRow};

                foreach my $row ($sheet -> {MinRow} .. $sheet -> {MaxRow}) {
                    $TotalRow = ($sheet->{MaxRow} - $sheet->{MinRow})+1;
                    $sheet -> {MaxCol} ||= $sheet -> {MinCol};
                    $TotalCol = ($sheet->{MaxCol} - $sheet->{MinCol})+1;
                    foreach my $col ($sheet -> {MinCol}
.. $sheet -> {MaxCol}) {
                        my $cell = $sheet -> {Cells} [$row] [$col];
                        if ($cell) {
                            $tmp = $cell -> {Val};
                            push (@tab_full, $tmp);
                        }else { push(@tab_full, '_');}

                        if ($i == $max_ligne){$i = 0; last;}
                        $i++;
                    }
                }
            }
        }

        #Traitement du fichier XLS
        elsif ($nom_fichier =~ /\.xls/){
            my $filename = shift || $repertoire . $nom_fichier;
            my $e = new Spreadsheet::ParseExcel;
```

```

my $eBook = $e->Parse($filename);
my $sheets = $eBook->{SheetCount};
my ($sheet, $sheetName);
foreach my $sheet (0 .. $sheets - 1) {
$sheet = $eBook->{Worksheet}[$sheet];
$sheetName = $sheet->{Name};
next unless (exists ($sheet->{MaxRow}) and (exists ($sheet->{MaxCol})));
    foreach my $row ($sheet->{MinRow} .. $sheet->{MaxRow}) {
        $TotalRow = ($sheet->{MaxRow} - $sheet->{MinRow})+1;
        foreach my $column ($sheet->{MinCol} .. $sheet->{MaxCol}) {
            $TotalCol = ($sheet->{MaxCol} - $sheet->{MinCol})+1;
            if (defined $sheet->{Cells}[$row][$column]){
                $tmp = $sheet->{Cells}[$row][$column]->Value;
                push(@tab_full, $tmp);
            } else {
                push(@tab_full, '_');
                push (@cell_vide, $memoire+1);
            }

            $memoire++;

        }

        if ($i == $max_ligne){$i = 0; last;}
        $i++;

    }

}

else {
    #print "error extension \n";
    @tab_full = ();
    $resultat = "Format inconnu\n";
    goto END;
}

#Si le fichier contient plus de 5 colonnes
if ($TotalCol > 5) {
    print "Le fichier contient plus de 5 colonnes\n";
    $resultat = 'Erreur sur le fichier';
    send_do("UPDATE tableatt SET etat = 'trop_col' WHERE id = '$id'");
}

# Si il ya des colonnes vide
if (($cell_vide[0] + $TotalCol) == $cell_vide[1]){
    if (($cell_vide[1] + $TotalCol) == $cell_vide[2]){
        if (($cell_vide[2] + $TotalCol) == $cell_vide[3]){
            $resultat = "Erreur sur le fichier, colonne $cell_vide[0] est vide";
            send_do("UPDATE tableatt SET etat = '$cell_vide[0]_vide' WHERE id =
'$id'");
        }
    }
}

elseif (($cell_vide[0] + $TotalCol) == $cell_vide[2]) {
    if (($cell_vide[2] + $TotalCol) == $cell_vide[4]){
        if (($cell_vide[4] + $TotalCol) == $cell_vide[6]){
            $resultat = "Erreur sur le fichier, colonne $cell_vide[0]
est vide";
            send_do("UPDATE tableatt SET etat = '$cell_vide[0]_vide'
WHERE id = '$id'");
        }
    }
}

#Supprimer tous les espace
foreach $tmp (@tab_full){
    if ($tmp eq '_') {$tmp = ' ';}
    #print "$tmp\n";
}

```

```

    }

    #Ecriture sur la base
    $length_tab = @tab_full;
    print "@tab_full\n";
    print "length du tab_full : $length_tab\n";
    print "row : $TotalRow\n";
    print "col :$TotalCol\n";
    if ($length_tab > 0) {
        $j = 0;

        for(my $i=0; $i<$TotalCol; $i++){

            if($TotalCol == 2){

                send_do("INSERT INTO basetempo (numclient, etat, prenom,
nom) VALUES ('$logine_client','off','$stab_full[$i+$j]','$stab_full[$i+1+$j]')");
                $resultat = 'Opération est déroulé avec succès';
                send_do("UPDATE tableatt SET etat = 'en_cours' WHERE id =
'$id'");

            }

            elseif($TotalCol == 3){
                send_do("INSERT INTO basetempo (numclient, etat, prenom,
nom, variable1) VALUES
('$logine_client','off','$stab_full[$i+$j]','$stab_full[$i+1+$j]','$stab_full[$i+2+$j]')");
                $resultat = 'Opération est déroulé avec succès';
                send_do("UPDATE tableatt SET etat = 'en_cours' WHERE id
= '$id'");

            }

            elseif($TotalCol == 4){
                send_do("INSERT INTO basetempo (numclient, etat, prenom,
nom, variable1, variable2) VALUES
('$logine_client','off','$stab_full[$i+$j]','$stab_full[$i+1+$j]','$stab_full[$i+2+$j]','$stab_full[$i+3+$j]')");
                $resultat = 'Opération est déroulé avec succès';
                send_do("UPDATE tableatt SET etat = 'en_cours'
WHERE id = '$id'");

            }

            elseif($TotalCol == 5){
                send_do("INSERT INTO basetempo (numclient, etat, prenom,
nom, variable1, variable2, telephone) VALUES
('$logine_client','off','$stab_full[$i+$j]','$stab_full[$i+1+$j]','$stab_full[$i+2+$j]','$stab_full[$i+3+$j]','$stab_full[$i+4+$j]')");
                $resultat = 'Opération est déroulé avec succès';
                send_do("UPDATE tableatt SET etat = 'en_cours' WHERE id =
'$id'");

            }

            $j += $TotalCol-1;

        }

    }

END:
    print "$resultat\n";

}

else {print "Pas de fichier!\n";}
$id = 0;
sleep(20);
}

```

## ANNEX B : SCRIPT DE CHANGEMENT

Ce script permet de traiter un fichier excel et de écrire dans une base de donnée.

```
use DBI;
use Spreadsheet::XLSX;
require "connect_base.pl";

#Connexion à la base
my $dbh = connect_db();

while(1) {
#Les variable globale
    my $idclient;
    my $prenom;
    my $nom;
    my $variable1;
    my $variable2;
    my $telephone;
    my $nom_base;
    my $nom_fichier;
    my $var1;          #prenom
    my $var2;          #nom
    my $var3;          #variable1
    my $var4;          #variable2
    my $var5;          #telephone
    my @tab_variable;
    my @tab_full;
    my $id = 0;
    my $i = 0;

    print "Verification...\n";
    #Récupérer l'ordre, idclient dans la base "baseclassement"
    my $sth = send_prepare("SELECT * FROM baseclassement WHERE etat = 'off'");
    while (my @row = $sth->fetchrow_array){
        $id = $row[0];
        $idclient = $row[1];
        $prenom = $row[3];
        $nom = $row[4];
        $variable1 = $row[5];
        $variable2 = $row[6];
        $telephone = $row[7];
    }

    if($id) {
        #remettre etat "on", si on a fini de récupérer
        send_do("UPDATE baseclassement SET etat = 'on' WHERE numero = '$id'");

        #Récupérer le fichier initiale et nom de la base dans la base "tableatt"
        $sth = send_prepare("SELECT * FROM tableatt WHERE etat = 'en_cours' AND
login_client = '$idclient'");
        while(my @row = $sth->fetchrow_array){
            $id = $row[0];
            $nom_base = $row[2];
            $nom_fichier = $row[3];
        }

        send_do("UPDATE tableatt SET etat = 'on' WHERE id = '$id'");

        #Changement de l'ordre
        $var1 = $prenom;
        $var2 = $nom;
        $var3 = $variable1;
        $var4 = $variable2;
        $var5 = $telephone;
        #print "var : $var1 : $var2 : $var3 : $var4 : $var5 \n";

        if ($var1 ne 'null'){ push (@tab_variable, $var1);}
        if ($var2 ne 'null'){ push (@tab_variable, $var2);}
        if ($var3 ne 'null'){ push (@tab_variable, $var3);}
        if ($var4 ne 'null'){ push (@tab_variable, $var4);}
        if ($var5 ne 'null'){ push (@tab_variable, $var5);}
```



```

#Traitement de fichier
print "Fichier à analyser : $nom_fichier\n";
print "Analyse en cours...\n";
$repertoire = '/home/ktcentrex/prepaids_cards/adminclient/monupload/';
system "chmod -R 777 /home/ktcentrex/prepaids_cards/adminclient/monupload/";

#Traitement du fichier XLSX
if ($nom_fichier =~ /\.xlsx/){
    print "Fichier xlsx\n";
    my $excel = Spreadsheet::XLSX -> new ($repertoire . $nom_fichier);
    foreach my $sheet (@{$excel -> {Worksheet}}) {
        $sheet -> {MaxRow} ||= $sheet -> {MinRow};
        foreach my $row ($sheet -> {MinRow} .. $sheet -> {MaxRow}) {
            $TotalRow = ($sheet->{MaxRow} - $sheet->{MinRow})+1;
            $sheet -> {MaxCol} ||= $sheet -> {MinCol};
            $TotalCol = ($sheet->{MaxCol} - $sheet->{MinCol})+1;
            foreach my $col ($sheet ->
{MinCol} .. $sheet -> {MaxCol}) {
                my $cell = $sheet -> {Cells} [$row] [$col];
                if ($cell) {
                    #printf("( %s , %s ) => %s\n", $row, $col, $cell ->
{Val});
                    $tmp = $cell -> {Val};
                    push
(@tab_full, $tmp);
                }
            }
        }
    }
}

#Traitement du fichier XLS
elsif ($nom_fichier =~ /\.xls/){
    my $filename = shift || $repertoire . $nom_fichier;
    my $e = new Spreadsheet::ParseExcel;
    my $eBook = $e->Parse($filename);
    my $sheets = $eBook->{SheetCount};
    my ($eSheet, $sheetName);
    foreach my $sheet (0 .. $sheets - 1) {
        $eSheet = $eBook->{Worksheet}[$sheet];
        $sheetName = $eSheet->{Name};
        next unless (exists ($eSheet->{MaxRow}) and (exists ($eSheet-
>{MaxCol})));
        foreach my $row ($eSheet->{MinRow} .. $eSheet->{MaxRow}) {
            $TotalRow = ($eSheet->{MaxRow} - $eSheet-
>{MinRow})+1;
            foreach my $column ($eSheet->{MinCol} ..
$eSheet->{MaxCol}) {
                $TotalCol =
($eSheet->{MaxCol} - $eSheet->{MinCol})+1;
                if (defined
$eSheet->{Cells}[$row][$column]){
                    $tmp =
$eSheet->{Cells}[$row][$column]->Value;
                    push(@tab_full, $tmp);
                } else { push(@tab_full,
' ');}
            }
        }
    }
}

}
else {
    #print "error extension \n";
    @tab_full = ();
    $resultat = "Format inconnu\n";
}

```

```

        goto END;
    }
    #Supprimer tous les espace dans le champ phone
    my $position_phone = '';
    my $p = 0;
    foreach my $var(@tab_variable){
        if($var eq 'phone'){ $position_phone = $p;}
        $p++;
    }

    $p = 0;
    my $incr = 0;
    foreach $tmp (@tab_full){
        if ($position_phone == ($p-$incr)) {
            $tmp = join '', split ' ', $tmp;
            $incr += 4;
        }

        if ($tmp eq '_') { $tmp = ' ';}
        $p++;
        #print "$tmp\n";
    }

    $length_tab = @tab_full;
    print "ligne : $length_tab\n";
    print "row : $TotalRow\n";
    print "col : $TotalCol\n";

    #Ecrire dans la base "base_client"
    print "Ecriture dans la table \"base_client\" \n";
    $j = 0;

    for(my $i=0; $i<$TotalRow; $i++){
        my $m = 0;
        foreach my $var_tmp (@tab_variable) {
            if ($var_tmp eq $tab_variable[0]){
                send_do("INSERT INTO base_client ($var_tmp,
numcli, nombase) VALUES ('$tab_full[$i+$j+$m]', '$idclient', '$nom_base')");
            }

            else {
                send_do("UPDATE base_client SET $var_tmp =
'$tab_full[$i+$j+$m]', numcli = '$idclient', nombase = '$nom_base' WHERE $tab_variable[0] =
'$tab_full[$i+$j]' AND nombase = '$nom_base'" );
            }

            $m++;
        }

        $j += $TotalCol-1;
    }

    print "Effacer la table basetempo \n";
    send_do("DELETE FROM basetempo");

    }else {print "Il n'ya pas de classement\n";}

    print "Fin de traitement\n";
    sleep(20);
}

```