



# Configuration matérielle pour l'envoi de SMS



h:  
<https://pandorafms.com/manual/!current/>  
manent link:  
[https://pandorafms.com/manual/!current/fr/documentation/pandorafms/technical\\_annexes/02\\_pandorafms\\_sms\\_gateway](https://pandorafms.com/manual/!current/fr/documentation/pandorafms/technical_annexes/02_pandorafms_sms_gateway)  
4/06/10 14:36



# Configuration matérielle pour l'envoi de SMS

## À propos de l'appareil GSM

Un périphérique GSM spécial est utilisé pour envoyer des SMS via un port série (USB). Vous pouvez utiliser soit un autre modèle GSM similaire, soit un téléphone portable avec connexion USB ou série. L'appareil utilisé ici est un MTX 65 v3. Cet appareil peut être acheté commercialement pour environ 100 \$ USD sur différents sites Web :

- <http://matrix.es> .
- <http://www.youtube.com/watch?v=OxcKAarS2M0> .

Comme vous pouvez le voir sur YouTube, c'est un appareil très petit et compatible GNU/Linux, qui a différents composants optionnels, comme une antenne GSM qui est très utile, par exemple, si le centre de données est dans le sous-sol.

## Installation de l'appareil

La première étape consiste à installer le périphérique matériel. Ce dispositif se compose des parties suivantes :

- Câble USB standard, avec un petit connecteur et une fermeture.
- Alimentation électrique. Dans cet exemple, il s'agit de l'euro-pennante 220 volts, si vous habitez en Amérique (sauf au Chili), veuillez vous assurer que l'alimentation électrique est de 110 ou 120 volts.
- Carte SIM.
- Dispositif de passerelle SMS Pandora FMS.



Ouvrez le périphérique Pandora FMS SMS Gateway :



Placez la carte SIM à l'intérieur.



Connectez-vous au réseau sur la prise étiquetée power et insérez le câble USB dans le périphérique de passerelle SMS.



Connectez l'autre extrémité au serveur Pandora FMS à l'aide d'un port USB standard



Lorsque vous connectez l'appareil au serveur, attendez quelques secondes et lancez la commande `dmesg` à partir de la ligne de commande. Vous devriez voir quelque chose de similaire à cet écran :

```
[ 22.814094] pci 0000:00:02.0: irq 2298 for MSI/MSI-X
[ 22.814180] [drm] Initialized i915 1.6.0 20080730 on minor 0
[ 24.688037] [drm:i915_setparam] *ERROR* unknown parameter 4
[ 27.212863] tg3 0000:09:00.0: irq 2297 for MSI/MSI-X
[ 27.265652] /dev/vmnet: open called by PID 2716 (vmnet-bridge)
[ 27.265663] /dev/vmnet: hub 0 does not exist, allocating memory.
[ 27.265672] /dev/vmnet: port on hub 0 successfully opened
[ 27.265692] bridge-eth0: up
[ 27.266040] ADDRCONF(NETDEV_UP): eth0: link is not ready
[ 27.266055] bridge-eth0: attached
[ 27.273643] bridge-eth0: disabling the bridge
[ 27.275862] bridge-eth0: down
[ 27.275917] bridge-eth0: detached
[ 28.725052] vmnet1: no IPv6 routers present
[ 28.916037] vmnet2: no IPv6 routers present
[ 32.205041] eth1: no IPv6 routers present
[ 91.000154] Clocksource tsc unstable (delta = -183026827 ns)
[ 154.992861] Bluetooth: HIDP (Human Interface Emulation) ver 1.2
[ 154.995497] input: Dell BT Travel Mouse as /devices/pci0000:00/0000:00:1a.0/usb3/3-2/3-2.1/3-2.1:1.0/bluetooth/hci0/hci0:11/input12
[ 155.006505] generic-bluetooth 0005:046D:B006.0003: input,hidraw2: BLUETOOTH HID v1.24 Mouse [Dell BT Travel Mouse] on 00:1F:3A:D6:6F:7C
[ 1124.784176] usb 5-1: new full speed USB device using uhci_hcd and address 2
[ 1124.978913] usb 5-1: configuration #1 chosen from 1 choice
[ 1125.079813] cdc_acm 5-1:1.0: ttyACM0: USB ACM device
[ 1125.082273] usbcore: registered new interface driver cdc_acm
[ 1125.082281] cdc_acm: v0.26:USB Abstract Control Model driver for USB modems and ISDN adapters
```

Cela signifie que le périphérique a été reconnu par le noyau et est prêt à accepter des commandes sur un périphérique, comme `/dev/ttyACM0`.

Si vous êtes arrivé jusqu'ici, la configuration est faite. Si non, veuillez vérifier toutes les étapes et assurez-vous que :

- L'appareil est connecté et le câble clignote en vert.
- L'appareil est connecté au port USB, des deux côtés du câble, l'un de ses côtés avec le périphérique SMS et l'autre avec le serveur hôte Pandora FMS.
- L'appareil a une carte SIM à l'intérieur, et celle-ci est correctement placée.

## Configurer SMSTools pour utiliser le nouvel appareil

Cet appareil est géré par un logiciel appelé SMSTools. Vous pouvez installer SMSTools en utilisant le package fourni par votre distribution GNU/Linux choisie ou en utilisant le package **RPM** (*Red Hat Package Manager* ou *RPM Package Manager*) fourni par Ártica PFMS (uniquement pour les distributions RPM telles que Fedora, Mandriva, Mageia, PCLinuxOS, entre autres).

### Systemes basés sur RPM

L'utilisation du RPM fourni par Ártica PFMS sur SUSE et Red Hat est très simple. Il vous suffit de l'installer avec la commande suivante :

```
# rpm -i smstools*.rpm
```

### Configurer SMStools

Modifiez le fichier de configuration de base :

```
# vi /etc/smsd.conf
```

Mettez ces contenus. Si la sortie ou le résultat de la commande `dmesg` diffère de `ttyACM0`, *utilisez le périphérique tty détecté par votre système.*

```
# Example smsd.conf. Read the manual for a description
```

```
devices = GSM1
logfile = /var/log/smsd.log
loglevel = 10
```

```
[GSM1]
device = /dev/ttyACM0
incoming = no
pin = 2920
```

Utilisez le code PIN attribué à votre carte SIM. Dans cet exemple, le code PIN est 2920.

Ensuite, lancez manuellement smstools :

```
# /usr/bin/smstools start
```

Envoyez un SMS de test. Attention : les numéros de téléphone doivent avoir le préfixe international complet. Dans cet exemple, +34 est le préfixe pour l'Espagne, et le numéro de téléphone est 627934648 :

```
$ sendsms 34***** "Pandora FMS rocks"
```

Attendez une minute et surveillez vos *journaux* pour vérifier que tout est correct. Vous devriez recevoir le SMS dans quelques secondes. Selon le réseau, le premier SMS peut être vérifié toutes les 10 à 20 secondes. Après ça, attendez. Le prochain SMS devrait être presque immédiat. SMSTools utilise une file d'attente pour envoyer des messages, de sorte que vous pouvez envoyer autant de messages que vous le souhaitez, *et ils seront émis lorsque le réseau de votre mobile peut les gérer.*

Pour voir les *logs* :

```
# cat /var/log/smsd.log
2009-11-12 11:30:12,2, smsd: Smsd v2.2.20 started.
2009-11-12 11:30:12,6, smsd: outgoing file checker has started.
2009-11-12 11:30:12,6, GSM1: Modem handler 0 has started.
2009-11-12 11:30:13,6, smsd: Moved file /var/spool/sms/outgoing/send_mNZxHa to
/var/spool/sms/checked
2009-11-12 11:30:13,6, smsd: I have to send 1 short message for
/var/spool/sms/checked/send_iUegPD
2009-11-12 11:30:13,6, GSM1: Sending SMS from to 627934648
2009-11-12 11:30:13,6, GSM1: Checking if modem is ready
2009-11-12 11:30:13,7, GSM1: -> AT
2009-11-12 11:30:13,7, GSM1: Command is sent, waiting for the answer
2009-11-12 11:30:14,7, GSM1: <- AT
OK
2009-11-12 11:30:14,6, GSM1: Checking if modem needs PIN
2009-11-12 11:30:14,7, GSM1: -> AT+CPIN?
2009-11-12 11:30:14,7, GSM1: Command is sent, waiting for the answer
2009-11-12 11:30:14,7, GSM1: <- AT+CPIN?
+CPIN: SIM PIN
OK
2009-11-12 11:30:14,5, GSM1: Modem needs PIN, entering PIN...
2009-11-12 11:30:14,7, GSM1: -> AT+CPIN="2920"
2009-11-12 11:30:14,7, GSM1: Command is sent, waiting for the answer
2009-11-12 11:30:15,7, GSM1: <- AT+CPIN="2920"
OK
2009-11-12 11:30:15,7, GSM1: -> AT+CPIN?
2009-11-12 11:30:15,7, GSM1: Command is sent, waiting for the answer
2009-11-12 11:30:15,7, GSM1: <- AT+CPIN?
+CPIN: READY
OK
2009-11-12 11:30:15,6, GSM1: PIN Ready
2009-11-12 11:30:15,6, GSM1: Checking if Modem is registered to the network
2009-11-12 11:30:15,7, GSM1: -> AT+CREG?
2009-11-12 11:30:15,7, GSM1: Command is sent, waiting for the answer
2009-11-12 11:30:16,7, GSM1: <- AT+CREG?
+CREG: 0,2
OK
2009-11-12 11:30:16,5, GSM1: Modem is not registered, waiting 10 sec. before
retrying
```



```
2009-11-12 11:30:26,7, GSM1: -> AT+CREG?
2009-11-12 11:30:26,7, GSM1: Command is sent, waiting for the answer
2009-11-12 11:30:26,7, GSM1: <- AT+CREG?
+CREG: 0,5
OK
2009-11-12 11:30:26,6, GSM1: Modem is registered to a roaming partner network
2009-11-12 11:30:26,6, GSM1: Selecting PDU mode
2009-11-12 11:30:26,7, GSM1: -> AT+CMGF=0
2009-11-12 11:30:26,7, GSM1: Command is sent, waiting for the answer
2009-11-12 11:30:26,7, GSM1: <- AT+CMGF=0
OK
2009-11-12 11:30:26,7, GSM1: -> AT+CMGS=94
2009-11-12 11:30:26,7, GSM1: Command is sent, waiting for the answer
2009-11-12 11:30:27,7, GSM1: <- AT+CMGS=94>

2009-11-12 11:30:27,7, GSM1: ->
001100099126974346F900F1FF5CC8373BCC0295E7F437A83C07D5DDA076D93D0FABCBA069730A22
97417079BD2C0EBB406779789C0ECF41F0B71C44AF83C66FB7391D76EBC32C503B3C46BFE9651608
1E7693DFF230C8D89C82E4EFF17A0E0
2009-11-12 11:30:27,7, GSM1: Command is sent, waiting for the answer
2009-11-12 11:30:31,7, GSM1: <-
001100099126974346F900F1FF5CC8373BCC0295E7F437A83C07D5DDA076D93D0FABCBA069730A22
97417079BD2C0EBB406779789C0ECF41F0B71C44AF83C66FB7391D76EBC32C503B3C46BFE9651608
1E7693DFF230C8D89C82E4EFF17A0E0
+CMGS: 0
OK
2009-11-12 11:30:31,5, GSM1: SMS sent, To: 627934648
2009-11-12 11:30:31,6, smsd: Deleted file /var/spool/sms/checked/send_iUegPD
2009-11-12 11:30:32,6, smsd: I have to send 1 short message for
/var/spool/sms/checked/send_mNZxHa
2009-11-12 11:30:32,6, GSM1: Sending SMS from to 34*****
2009-11-12 11:30:32,7, GSM1: -> AT+CMGS=29
2009-11-12 11:30:32,7, GSM1: Command is sent, waiting for the answer
2009-11-12 11:30:33,7, GSM1: <- AT+CMGS=29>

2009-11-12 11:30:33,7, GSM1: ->
0011000B914326974346F900F1FF11D0B09BFC968741C6E614247F8FD7730
2009-11-12 11:30:33,7, GSM1: Command is sent, waiting for the answer
2009-11-12 11:30:36,7, GSM1: <-
0011000B914326974346F900F1FF11D0B09BFC968741C6E614247F8FD7730
+CMGS: 1
OK
2009-11-12 11:30:36,5, GSM1: SMS sent, To: 34*****
2009-11-12 11:30:36,6, smsd: Deleted file /var/spool/sms/checked/send_mNZxHa
```

Enfin, quelques tâches pour assurer cette opération pour l'avenir :

1. Fixez 1 pour loglevel dans /etc/smsd.conf pour éviter un fichier *journal* excessivement volumineux et inutile.
2. Assurez-vous que smsd est configuré pour démarrer automatiquement lorsque le système



redémarre (cela implique un lien de `/etc/init.d/sms` avec `/etc/rc2.d/S90sms` ou `/etc/rc.d/rc2.d/S90sms`). Si vous l'avez installé à partir d'un paquet, il existe probablement déjà sur votre système. *Il suffit de vérifier.*

## Configurer l'alerte dans Pandora FMS

Ces étapes reproduisent les étapes de base pour créer des alertes SMS dans Pandora FMS 3.x.


Pour plus d'informations, consultez "[Système d'alertes chez Pandora FMS](#)".

Créez la commande

Configure alert command

Alerts

<b>Name</b>	<b>Group</b>
<input type="text" value="SMS"/>	<input type="text" value="All"/>
<b>Command</b>	<b>Description</b>
<input type="text" value="/usr/bin/sendsms_field1_field2_"/>	<input type="text" value="This command sends a SMS using Pandora FMS SMS gateway attached to USB port in the PFMS server."/>


Create 

Créez l'action :

## Configure alert action

## Alerts

<b>Name</b>	<input type="text" value="SMS to admin"/>	<b>Group</b>	<input type="text" value="All"/>
<b>Command</b>	<input type="text" value="SMS"/>	<b>Threshold</b>	<input type="text" value="5 minutes"/>

Create Command 

Send SMS using the standard SMS device, using smstools. Uses field2 as text message, field1 as destination phone (include international prefix!)

Associez l'action à un module à l'aide d'un modèle d'alerte précédent. Dans ce cas, le modèle d'alerte sera lancé lorsque l'état du module devient CRITIQUE.

## Passerelle d'envoi de SMS avec Gnokii

Il est recommandé de maintenir vos systèmes à jour avec la [dernière version de Pandora FMS](#). Ces informations sont conservées à des fins historiques.

Cette méthode d'envoi de SMS était celle proposée dans les versions 1.x et 2.x de Pandora FMS. Dans la version 3.x [smstools](#) est utilisé. La méthode basée sur smstools est meilleure et il est déconseillé d'utiliser Gnokii en raison de sa complexité et de sa possibilité de défaillance. Cependant, il est référencé ici pour pouvoir disposer de méthodes alternatives à smstools.

Cette section décrit comment monter une passerelle d'envoi de SMS basée sur une file d'attente d'envoi. De cette façon, vous pouvez déployer un serveur d'envoi de SMS, connecté à un mobile et en envoyant des SMS via le logiciel du projet Gnokii, et différents serveurs distants. Vous pouvez envoyer vos messages pour être traités par le serveur d'envoi de SMS. Cela permet à différents serveurs Pandora FMS (ou à d'autres machines utilisant la *passerelle*) d'envoyer des messages de manière centralisée, *sans avoir besoin d'avoir un mobile pour chaque serveur*.

Tout d'abord, vous devez créer un utilisateur SMS sur la machine où vous installerez la passerelle d'envoi de SMS. Créez ensuite les répertoires `home/sms` et `home/sms/incoming`. Si vous

souhaitez utiliser la *passerelle* SMS à partir d'autres machines, vous devez rendre le répertoire `/home/sms/incoming` accessible à d'autres serveurs via n'importe quel système d'envoi de fichiers ou de partage de systèmes de fichiers : NFS, SMB, SSH (scp), FTP ou **Tentacle**.

Le mécanisme de la passerelle d'envoi de SMS est simple : pour chaque fichier situé dans le répertoire `/home/sms/incoming`, un SMS contenant le contenu du fichier sera traité, supprimé et envoyé. Ce fichier doit avoir un format spécifique, qui est détaillé ci-dessous :

```
Numéro de téléphone|texte du SMS
```

## Déploiement de la passerelle avec Gnokii

Il est recommandé de maintenir vos systèmes à jour avec la **dernière version de Pandora FMS**. Ces informations sont conservées à des fins historiques.

Vous devez créer quatre *scripts* :

**SMS** : *Script* qui envoie le SMS à l'aide de Gnokii via un câble de données USB. Ce *script* est uniquement sur le système où se trouve la *passerelle* d'envoi (le système qui a le câble de données connecté à un mobile GSM).

**SMS\_GATEWAY** : *Script* qui traite périodiquement le répertoire d'entrée (`/home/sms/incoming`) en traitant les fichiers en attente d'envoi. Ce *script* se trouve uniquement sur le système qui fait office de *passerelle* d'envoi.

**SMS\_GATEWAY\_LAUNCHER** : *Script* de lancement pour le *script* SMS\_GATEWAY (démon de démarrage et d'arrêt). Ce *script* se trouve uniquement sur le système qui fait office de *passerelle* d'envoi.

**COPY\_SMS**: Copiez un SMS à l'aide de la commande `scp` d'un système client vers le système *passerelle*. Utilisez le `TÉLÉPHONE` comme premier paramètre, et le second comme texte à envoyer (en utilisant `""` pour spécifier chaque paramètre). Le *script* s'appuie sur l'authentification automatique SSH et l'utilisateur `sms` pour le transfert. Sur le système local, vous pouvez remplacer la commande `scp` par la commande `cp` ou utiliser un système tel que **Tentacle** pour transférer le fichier.

### **sms**

Il est recommandé de maintenir vos systèmes à jour avec la **dernière version de Pandora FMS**. Ces informations sont

conservées à des fins historiques.

C'est le *script* qui envoie des SMS en utilisant Gnokii. Vous devez avoir Gnokii correctement configuré (en utilisant le fichier `/etc/gnokii.conf` ou similaire). Vous devez probablement être l'utilisateur `root` pour pouvoir lancer le *script*, ou définir `SETUID0` sur le binaire Gnokii.

```
#!/bin/bash
texto=$1
number=$2
if [ $# != 2 ]; then
echo "I need more parameters"
exit 1;
fi
/bin/echo $1 | /usr/local/bin/gnokii --sendsms $2
```

### sms\_gateway

Il est recommandé de maintenir vos systèmes à jour avec la [dernière version de Pandora FMS](#). Ces informations sont conservées à des fins historiques.

Voici le *script* du démon *passerelle*:

```
#!/bin/bash

INCOMING_DIR=/home/sms/incoming
HOME_DIR=/home/sms

while [ 1 ]
do

    for a in `ls $INCOMING_DIR`
    do
        if [ ! -z "$a" ]
        then
            NUMBER=`cat $INCOMING_DIR/$a | cut -d "|" -f 1`
            MESSAGE=`cat $INCOMING_DIR/$a | cut -d "|" -f 2`
            TIMESTAMP=`date +"%Y/%m/%d %H:%M:%S"`
            echo "$TIMESTAMP Sending to $NUMBER the message
$MESSAGE">> $HOME_DIR/sms_gateway.log
            $HOME_DIR/sms "$MESSAGE" "$NUMBER"
            echo "$TIMESTAMP Deleting $a">>
$HOME_DIR/sms_gateway.log
            rm -Rf $INCOMING_DIR/$a
            sleep 1
```

```
                fi
            done
            sleep 5
done
```

## sms\_gateway\_launcher

Il est recommandé de maintenir vos systèmes à jour avec la [dernière version de Pandora FMS](#). Ces informations sont conservées à des fins historiques.

Voici le *script* de lancement pour le sms\_gateway:

```
#!/bin/bash

INCOMING_DIR=/home/sms/incoming
HOME_DIR=/home/sms

while [ 1 ]
do

    for a in `ls $INCOMING_DIR`
    do
        if [ ! -z "$a" ]
        then
            NUMBER=`cat $INCOMING_DIR/$a | cut -d "|" -f 1`
            MESSAGE=`cat $INCOMING_DIR/$a | cut -d "|" -f 2`
            TIMESTAMP=`date +"%Y/%m/%d %H:%M:%S"`
            echo "$TIMESTAMP Sending to $NUMBER the message
$MESSAGE">> $HOME_DIR/sms_gateway.log
            $HOME_DIR/sms "$MESSAGE" "$NUMBER"
            echo "$TIMESTAMP Deleting $a">>
$HOME_DIR/sms_gateway.log
            rm -Rf $INCOMING_DIR/$a
            sleep 1
        fi
    done
    sleep 5
done
```

## copy\_sms

Il est recommandé de maintenir vos systèmes à jour avec la [dernière version de Pandora FMS](#). Ces informations sont conservées à des fins historiques.

Ce petit *script* crée un fichier d'envoi SMS sur une machine cliente et le copie sur la *passerelle* SMS à l'aide de scp:

```
#!/bin/bash

SERIAL=`date +"%j%M%s"`
SERIAL=`hostname`_${SERIAL}

TEL=$1
TEXT=$2

echo $TEL\|$TEXT>> /tmp/${SERIAL}
scp /tmp/${SERIAL} sms@192.168.1.1:/home/sms/incoming
rm -Rf /tmp/${SERIAL}
```