

PANDORAFMS



Configuración de hardware para el envío de SMS

01-08-2021





Configuración de hardware para el envío de SMS

[Volver al Índice de Documentación Pandora FMS](#)

Configuración de hardware para el envío de SMS con Pandora FMS

Acerca del dispositivo GSM

En esta documentación utilizamos un dispositivo GSM especial para enviar SMS a través de un puerto serie (usb). Puede utilizar otro modelo GSM similar, o un teléfono móvil con conexión USB o serie, esto no es lo realmente importante. El dispositivo empleado aquí es un MTX 65 v3. Este dispositivo puede ser adquirido comercialmente por aproximadamente 100 USD\$ en diferentes sitios web:

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

Como puede ver en Youtube, es un dispositivo muy pequeño y compatible con Linux, que tiene diferentes componentes opcionales, como una antena GSM, muy útil si su centro de datos está en el subsuelo, por ejemplo.

Instalación del Dispositivo

El primer paso es instalar el dispositivo hardware. Este dispositivo se compone de las siguientes partes:

- Cable USB Standard, con un pequeño conector y un cierre.
- Fuente de alimentación (en este ejemplo es la europea de 220v, si usted vive en EEUU, por favor asegúrese de que la fuente de alimentación valga para 110v).
- Tarjeta SIM.
- Dispositivo SMS gateway de Pandora FMS.



Abra el dispositivo SMS gateway de Pandora FMS y coloque dentro la tarjeta SMS.



Conecte a la red en la toma "power", inserte el cable USB en el dispositivo SMS Gateway y conecte el otro extremo con el servidor de Pandora FMS utilizando un puerto USB estándar.



Cuando conecte el dispositivo con el servidor, espere algunos segundos y lance el comando “dmesg” desde la línea de comandos. Debería ver algo similar a esta pantalla. Esto significa que el dispositivo ha sido reconocido por el kernel y está listo para aceptar comandos en un dispositivo, como por ejemplo /dev/ttyACM0

```
[ 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/u
sb3/3-2/3-2.1/3-2.1:1.0/bluetooth/hci0/hci0:11/input12
[ 155.006505] generic-bluetooth 0005:046D:B006.0003: input,hidraw2: BLUETOOTH H
ID 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 a
nd ISDN adapters
```



Si usted ha llegado hasta aquí, la configuración está hecha. Si no, por favor, revise todos los pasos y asegúrese de que:

- El dispositivo está conectado y el cable está parpadeando en un color verde.
- El dispositivo está conectado al puerto USB, por ambos lados del cable, uno de sus lados con el dispositivo SMS y el otro con el servidor host de Pandora FMS.
- El dispositivo tiene una tarjeta SIM dentro, y ésta está colocada adecuadamente.

Configurar SMSTools para Utilizar el Nuevo Dispositivo

Este dispositivo está gestionado por un paquete de software llamado SMSTools. Puede instalar SMSTools usando el paquete provisto por su Distribución Linux o usar el paquete RPM provisto por Ártica (sólo para distribuciones RPM).

Debian / Ubuntu

En Debian/Ubuntu, necesitará “personalizar” el script sendsms que usará Pandora FMS.

En primer lugar, instale el paquete desde los repositorios APT.

```
$ sudo apt-get install smstools
```

Después, necesitará usar uno de los ejemplos del script provisto para enviar sms desde la línea de comandos y “Personalizarlo”.

```
cp /usr/share/doc/smstools/examples/scripts/sendsms /usr/bin  
chmod 750 /usr/bin/sendsms
```

Edite /usr/bin/sendsms y añada la siguiente línea al final del script:

```
chmod 666 $FILE
```

Sistema basado en RPM (SUSE, Redhat)

Utilizar nuestro RPM es muy sencillo. Sólo tendrá que instalarlo:

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

Configurar SMSTools



Edite el fichero de configuración base:

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

Ponga estos contenidos. Si su output dmesg no es ttyACMO, utilice el dispositivo tty detectado por su sistema.

```
# 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
```

Utilice el PIN asignado a su SIM. En este ejemplo, el PIN es "2920".

Después, inicie manualmente smstools:

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

Envíe un test SMS. ATENCIÓN: los números de teléfono deben tener el prefijo internacional completo. En este ejemplo, +34 es el prefijo para España, y mi número de teléfono es 627934648:

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

Espere un minuto y vigile sus logs para comprobar que todo esta correcto. Debería recibir el SMS en unos segundos. Dependiendo de la red, el primer SMS puede chequear cada 10-20 segundos. Después de esto, espere. El siguiente SMS debería ser casi inmediato. SMSTools utiliza una cola para enviar mensajes, por lo que puede enviar tantos mensajes como desee, y se emitirán cuando la red de su móvil pueda gestionarlos

Para ver los 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: ->
001100099126974346F900F1FF5CC8373BCC0295E7F437A83C07D5DDA076D93D0FA
BCBA069730A2297417079BD2C0EBB406779789C0ECF41F0B71C44AF83C66FB7391D
76EBC32C503B3C46BFE96516081E7693DFF230C8D89C82E4EFF17A0E0
2009-11-12 11:30:27,7, GSM1: Command is sent, waiting for the
answer
2009-11-12 11:30:31,7, GSM1: <-
001100099126974346F900F1FF5CC8373BCC0295E7F437A83C07D5DDA076D93D0FA
BCBA069730A2297417079BD2C0EBB406779789C0ECF41F0B71C44AF83C66FB7391D
76EBC32C503B3C46BFE96516081E7693DFF230C8D89C82E4EFF17A0E0
+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
```

Finalmente, algunas tareas para asegurar esta operación para el futuro:

1. Fije 1 para loglevel en /etc/smsd.conf para evitar un fichero log excesivamente grande e innecesario.
2. Asegúrese de que smsd esta configurado para iniciarse automáticamente cuando el sistema se reinicia (esto implica un link de /etc/init.d/sms con /etc/rc2.d/S90sms or /etc/rc.d/rc2.d/S90sms). Si lo ha instalado desde un paquete, probablemente exista ya en su sistema. Sólo compruébelo.

Configurar la Alerta de Pandora FMS

Estos pasos reproducen los pasos básicos para crear alertas SMS en Pandora FMS 3.x

Cree el comando:

Alerts > Configure alert command

Name: SMS

Command: /usr/bin/sendms_field1_field2

Group: All

Description: This command sends a SMS using Pandora FMS SMS gateway attached to USB port in the Pandora FMS server.

Cree la acción:

Alerts > Configure alert action

Name: SMS to admin

Group: All

Command: SMS [Create Command](#)

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

Threshold: If words

Triggering: sendms +3462000000_field2

Recovery: sendms_field1_field2

Destination number: Field 1: +3462000000

Message: Field 2:



Asocie la acción a un módulo usando una plantilla de alerta anterior. En este caso, la plantilla de alerta se lanzará cuando el estado del módulo sea CRITICO.

//Gateway// de envío de SMS con Gnokii

Este método de envío de SMS era el propuesto en las versiones 1.x y 2.x de Pandora FMS. En la versión 3.x proponemos el uso de smstools, tal como se puede leer más arriba. El método basado en smstools es mejor y se recomienda no utilizar Gnokii por su complejidad y posibilidad de fallos. No obstante se referencia aquí para poder disponer de métodos alternativos a smstools.

Esta sección describe como montar una pasarela de envío de SMS basado en una cola de envíos. De esta forma se puede implementar un servidor de envío de SMS, conectado a un móvil y enviando los SMS a través del software del proyecto Gnokii, y diferentes servidores remotos, pueden enviar sus mensajes para que los procese el servidor de envío de SMS. Esto permite que diferentes servidores Pandora FMS (u otras máquinas que quieran usar el *gateway*) puedan enviar mensajes de forma centralizada, sin tener que tener un móvil por cada servidor.

Primeramente se debe crear un usuario «sms» en la máquina donde quiere instalar la pasarela de envío de SMS. Luego cree los directorios *home/sms* y */home/sms/incoming*. Si quiere utilizar el *gateway* SMS desde otras máquinas, necesitará hacer accesible el directorio */home/sms/incoming* a otros servidores por medio de cualquier sistema de envío de ficheros o de compartición de sistemas de archivos: NFS, SMB, SSH (scp), FTP o Tentacle.

El mecanismo de la pasarela de envío de SMS es sencillo: Por cada fichero que se encuentre en el directorio */home/sms/incoming* se procesará, eliminará y enviará un SMS con el contenido del fichero. Este fichero tiene que tener un formato específico, el cual se detalla a continuación:

Phonenumber|SMSText

Implementación del //gateway// con Gnokii

Tiene que crear cuatro *scripts*:

SMS: *Script* que envía el SMS usando Gnokii através de un cable de datos USB. Este *script* está únicamente en el sistema donde está el *gateway* de envío (el sistema que tiene el cable de datos conectado a un móvil GSM).

SMS_GATEWAY: *Script* que procesa de forma periódica el directorio de entrada (*/home/sms/incoming*) procesando ficheros pendientes de envío. Este *script* está únicamente en el sistema que hace de *gateway* de envío.



SMS_GATEWAY_LAUNCHER: *Script* lanzador para el *script* SMS_GATEWAY (demonio de inicio y parada). Este *script* está únicamente en el sistema que hace de *gateway* de envío.

COPY_SMS: Copia un SMS utilizando el comando **scp** desde un sistema cliente al sistema *gateway*. Utiliza el TELEFONO como primer parámetro, y el segundo como texto para enviar (utilizando "" para especificar cada parámetro). El *script* confía en la autenticación automática SSH y el usuario «sms» para la transferencia. En el sistema local se puede reemplazar el «scp» por el comando «cp» o utilizar un sistema como Tentacle para transferir el fichero.

sms

Este es el *script* que envía SMS utilizando Gnokii. Debe tener Gnokii correctamente configurado (usando el fichero */etc/gnokii.conf* o similar). Probablemente deba ser el usuario *root* para poder lanzar el *script*, o establecer *SETUID0* en el binario de 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

Este es el *script* del demonio *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
```

sms_gateway_launcher

Este es el *script* lanzador para el *sms_gateway*:

```
#!/bin/bash

# SMS Gateway, startup script
# Sancho Lerena, <slerena@gmail.com>
# Linux Version (generic)

# Configurable path and filenames
SMS_GATEWAY_HOME=/home/sms
SMS_PID_DIR=/var/run
SMS_PID=/var/run/sms.pid

# Main script

if [ ! -d "$SMS_PID_DIR" ]
then
    echo "SMS Gateway cannot write it's PID file in
$SMS_PID_DIR. Please create directory or assign appropriate perms"
    exit
fi

if [ ! -f $SMS_GATEWAY_HOME/sms_gateway ]
then
    echo "SMS Gateway not found, please check setup and read
manual"
    exit
```



```
fi
case "$1" in
  start)
    OLD_PATH=`pwd`
    if [ -f $SMS_PID ]
    then
      CHECK_PID=`cat $SMS_PID`
      CHECK_PID_RESULT=`ps aux | grep -v grep | grep
"$CHECK_PID" | grep "sms_gateway" | wc -l`
      if [ $CHECK_PID_RESULT == 1 ]
      then
        echo "SMS Gateway is currently running on
this machine with PID ($CHECK_PID). Aborting now..."
        exit
      fi
    fi

    nohup $SMS_GATEWAY_HOME/sms_gateway > /dev/null 2>
/dev/null & 2> /dev/null > /dev/null
    sleep 1

    MYPID=`ps aux | grep "$SMS_GATEWAY_HOME/sms_gateway" | grep
-v grep | tail -1 | awk '{ print $2 }'`
    if [ ! -z "$MYPID" ]
    then
      echo $MYPID > $SMS_PID
      echo "SMS Gateway is now running with PID $MYPID"
    else
      echo "Cannot start SMS Gateway. Aborted."
    fi
    cd "$OLD_PATH"
    ;;
  stop)
    if [ -f $SMS_PID ]
    then
      echo "Stopping SMS Gateway"
      PID_2=`cat $SMS_PID`
      if [ ! -z "`ps -F -p $PID_2 | grep -v grep | grep
'sms_gateway'`" ]
      then
        kill `cat $SMS_PID` 2> /dev/null > /dev/null
      else
        echo "SMS Gateway is not executing with PID $PID_2,
skip Killing step"
      fi
    fi
  fi
fi
```



```
        rm -f $SMS_PID
    else
        echo "SMS Gateway is not running, cannot stop it."
    fi
    ;;
force-reload|restart)
    $0 stop
    $0 start
    ;;
*)
    echo "Usage: sms_gateway {start|stop|restart}"
    exit 1
esac
```

copy_sms

Este pequeño *script* crea un fichero de envío SMS en una máquina cliente y lo copia al *gateway* SMS utilizando *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/$SERIAL1
```

[Volver al Índice de Documentación Pandora FMS](#)



From:

<https://pandorafms.com/manual/> - **Pandora FMS Documentation**

Permanent link:

https://pandorafms.com/manual/es/documentation/07_technical_annexes/02_pandorafms_sms_gateway

Last update: **2021/07/27 08:49**