



# Hardware configuration for SMS sending



URL: <https://pandorafms.com/manual/!779/>  
Permanent link: [https://pandorafms.com/manual/!779/en/documentation/pandorafms/technical\\_annexes/02\\_pandorafms\\_sms\\_gateway](https://pandorafms.com/manual/!779/en/documentation/pandorafms/technical_annexes/02_pandorafms_sms_gateway)  
2025/01/22 19:13





# Hardware configuration for SMS sending

## About the GSM device

A special GSM device is used to send SMS through a serial port (USB). You may use either another similar GSM model, or a cell phone with USB or serial connection. The device used here is a MTX 65 v3. It is a very small and GNU/Linux compatible device, which has different optional components such as an external GSM antenna for greater signal coverage.

## Device installation

The first step is to install the hardware device. This device consists of the following parts:

- Standard USB cable, with a small connector and a lock.
- Power supply. In this example it is the European 220 volts, if you live in America (except Chile) make sure that the power supply is for 110 or 120 volts.
- SIM card.
- Pandora FMS SMS gateway device.

Procedure:

- Open Pandora FMS SMS gateway device.
- Place the SIM card inside.
- Connect to the network at the socket labeled power, insert the USB cable into the SMS gateway device.
- Connect the other end to Pandora FMS server using a standard USB port.
- When you connect the device to the server, wait a few seconds and launch the `dmesg` command from the command line. You should see something similar to this screen:

```
[ 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/usb3/3-2/3-2.1/3-2.1.0/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
```

- This means that the device has been recognized by the kernel and is ready to accept commands on a device such as `/dev/ttyACM0`.

If there is a problem, check all the steps and make sure that:

- The device is connected and the cable is flashing green.
- The device is connected to the USB port, on both sides of the cable, one side with the SMS device and the other with Pandora FMS host server.
- The device has a SIM card inside, and the SIM card is properly inserted.

## Configuring SMSTools to use the new device

This device is managed by a software package called SMSTools. You may install SMSTools using the package provided by your chosen GNU/Linux distribution or use the RPM package (*Red Hat Package Manager* or *RPM Package Manager*) provided by PFMS.

### RPM-based systems

To use the RPM provided by PFMS just install it with the following command:

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

## Configurar SMStools

Edit with the user root, or equivalent, the base configuration file with a text editor (Vi is used here):

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

The following contents have to be added (if the output or result of the dmesg command differs from `ttyACM0`, use the device `tty` detected by your system):

```
# 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 =
```

On the last line, enter the PIN assigned to the SIM used.

Then smstools must be started manually:

```
/usr/bin/smsd start
```

You may now send a test SMS with a user other than root (34 is the international code for Spain, fill in the asterisks with the test phone number):

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

After a minute of waiting, monitor the logs to make sure that everything is correct. The SMS should be received within a few seconds. Depending on the network, the first SMS may be checked every 10 to 20 seconds, the next SMS should be almost immediate. SMStools uses a queue to send messages, so you may send as many messages as needed, *and they will be broadcasted when the cell phone network can handle them.*

To see logs:

```
cat /var/log/smsd.log
```

Finally, some tasks to secure this operation for the future:

1. Set 1 for loglevel in `/etc/smsd.conf` to avoid an excessively large and unnecessary `log` file.

2. Make sure smsd is configured to start automatically when the system reboots (this implies a link from `/etc/init.d/sms` to `/etc/rc2.d/S90sms` or `/etc/rc.d/rc2.d/S90sms`). If you installed it from a package, it probably already exists on the system, it will be a matter of checking it.

## Configure the Alert in Pandora FMS

For more information see [“Alert system in Pandora FMS”](#).

- First the alert command must be created:

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."/>

- And then create the alert action:

## Configure alert action

## Alerts

<b>Name</b>	<b>Group</b>
<input type="text" value="SMS to admin"/>	<input type="text" value="All"/>
<b>Command</b>	<b>Threshold</b>
<input type="text" value="SMS"/>	<input type="text" value="5 minutes"/>
Create Command <input type="button" value="⊕"/>	<input type="button" value="✎"/>
Send SMS using the standard SMS device, using smstools. Uses field2 as text message, field1 as destination phone (include international prefix!)	

The action is associated to a module using a previous alert template. In this case, the alert template will be triggered when the module status is CRITICAL.

[Back to Pandora FMS Documentation Index](#)