



Maxim Kazmin, 123RF

SMS-Nachrichten vom Rechner aus versenden

Funkpost

Sich vom Server direkt per SMS über Probleme benachrichtigen zu lassen ist praktisch. Wer dafür keinen Gateway-Dienst verwenden will und ein Mobiltelefon übrig hat, kann selbst eine Lösung basteln. Oliver Frommel

Wer sich über SMS zum Beispiel von einem Monitoring-System benachrichtigen lassen möchte, wie in dem Artikel zuvor beschrieben, dem stehen mehrere Wege offen. So können Sie, abhängig vom Baujahr Ihres Handys, das Gerät über einen seriellen Anschluss, eine Infrarot-Schnittstelle, Bluetooth, WLAN oder USB anbinden. Der Zuverlässigkeit wegen sollten Sie von drahtlosen Verbindungen eher absehen. Dieser Artikel verwendet zur Demonstration ein übrig gebliebenes Nokia 8210, das zum Anschluss über die serielle Schnittstelle ein spezielles Kabel erfordert, das Sie im Batterieraum des Telefons verankern.

Die typischen Softwarepakete, um unter Linux und Unix mit Mobiltelefonen zu arbeiten, heißen Gammu [1] und Gnokii [2]. Letzteres hat sich, wie der Name schon andeutet, auf Nokia-Geräte spezialisiert. Zuerst müssen Sie die verwendete Verbindung konfigurieren. Die benutzer-spezifischen Einstellungen legt Gammu in »~/gammurc« ab, Gnokii in »~/gnokii.rc«, die systemweiten Dateien liegen wie gewohnt unter »/etc«.

So sieht bei Gnokii die Konfiguration für das seriell angebundene Nokia 8210 aus:

```
[global]
model = 8210
port = /dev/ttyS0
connection = serial
```

Bei Gammu erfolgt die Konfiguration ähnlich, aber der Verbindungstyp heißt

```
gnokii --identify
GNOKII Version 0.6.28
IMEI       : 1223342424244
Manufacturer : Nokia
Modell     : 8210
Product name : NSM-3
Revision   : SW 5.28, HW1910
```

Gammu gibt etwas mehr Informationen aus (Abbildung 1).

Verbindungstest

Um zu testen, ob das Telefon auch im Netz eines Providers eingeloggt hat, können Sie wiederum ein Gnokii-Kommando verwenden:

```
$ gnokii --getnetworkinfo
GNOKII Version 0.6.28
Network       : T-Mobile / D1 - DeTe Mobil ..
Network Code  : 262 01
LAC           : 8a04 (35332)
Cell id       : 0000cf09 (53001)
```

Wie Sie sehen, gibt das Tool sogar die ID der Funkzelle aus. Nun lässt sich eine SMS mit einem simplen Befehl verschicken:

```
$ echo "Server down" | >
gnokii --sendsms >
0162123451234
GNOKII Version 0.6.28
Send succeeded with >
reference 0!
```

Ähnlich einfach funktioniert das Verschicken auch über Infrarot oder

Bluetooth, wenn die Programme das verwendete Gerät unterstützen. Konsultieren Sie im Zweifelsfall die Gerätelisten auf den Programm-Homepages. Ironischerweise gestaltet sich das Ganze mit Android-Telefonen viel schwieriger. Dafür ist es nötig, erst auf dem Telefon eine passende Anwendung aus dem Android-Market zu installieren. Für den Rechner gibt es dann meist ein grafisches Programm, mit dem das Verschicken von SMS möglich ist.

Damit arbeiten beispielsweise Android-Easy-SMS [3], Desktop-SMS [4] und Talk-My-Phone [5]. Mit etwas Java-Kenntnissen dürfte sich der Quellcode eines dieser Programme leicht für eigene Zwecke verwenden lassen. ■

Damit arbeiten beispielsweise Android-Easy-SMS [3], Desktop-SMS [4] und Talk-My-Phone [5]. Mit etwas Java-Kenntnissen dürfte sich der Quellcode eines dieser Programme leicht für eigene Zwecke verwenden lassen. ■

Damit arbeiten beispielsweise Android-Easy-SMS [3], Desktop-SMS [4] und Talk-My-Phone [5]. Mit etwas Java-Kenntnissen dürfte sich der Quellcode eines dieser Programme leicht für eigene Zwecke verwenden lassen. ■

Infos

[1] Gammu: [<http://wammu.eu/gammu/>]

[2] Gnokii: [<http://www.gnokii.org/>]

[3] Android-Easy-SMS: [<http://code.google.com/p/androideasysms/>]

[4] Desktop-SMS: [<http://code.google.com/p/desktopsms/>]

[5] Talk-My-Phone: [<http://code.google.com/p/talkmyphone/>]

```
oliver@marx:~$ gammu --identify
Hersteller   : Nokia
Modell       : 8210 (NSM-3)
Firmware     : 5.28 C (24-05-01)
Hardware     : 1910
IMEI         : 350601104330809
Original IMEI : 350601104330809
Hergestellt  : 0701
Produkt-Code : 0504436
Simlock 1    : MCC+MNC 00101, opened, user , counter 0
Simlock 2    : GID1 0000, opened, factory, counter 0
Simlock 3    : GID2 0000, opened, factory, counter 0
Simlock 4    : MSIN 0000000001, opened, factory, counter 0
MSID        : B32c38fc29777204f30ff30e67
MCU checksum : 5344
DSP ROM      : 6
oliver@marx:~$
```

Abbildung 1: Gammu weiß eine Menge über das gefundene Mobiltelefon zu erzählen.