

Nutzung des internen GPS Empfängers eines (Nokia) Symbian Handy

Die hier beschriebene Vorgehensweise wurde erfolgreich mit einem Nokia N82 getestet.

Im Normalfall ist es nicht möglich die GPS Daten eines internen GPS Empfängers eines Handys per Bluetooth an externe Geräte weiter zu leiten.

Gefunden habe ich aber dieses auf Java basierende Tool für Symbian Handys:
http://www.symarctic.com/beta/static.php?page=extgps_download

ExtGPS mit dem Handy runterladen, installieren und starten.
Danach sind folgende Schritte als Root auszuführen:

```
sdptool search SP
```

Die Ausgabe schaut bei mir so aus:

```
Inquiring ...
Searching for SP on XX:XX:XX:XX:XX:XX ...
Service Name: Data Transfer
Service Description: Version
Service RecHandle: 0x10019
Service Class ID List:
  "Serial Port" (0x1101)
Protocol Descriptor List:
  "L2CAP" (0x0100)

Service Name: Symarctic ExtGPS
Service Description: Share phone's built-in GPS module via Bluetooth
Service Provider: Symarctic Solutions
Service RecHandle: 0x1001f
Service Class ID List:
  "Serial Port" (0x1101)
Protocol Descriptor List:
  "L2CAP" (0x0100)
  "RFCOMM" (0x0003)
    Channel: 5
Language Base Attr List:
  code_ISO639: 0x656e
  encoding:    0x6a
  base_offset: 0x100
```

Wichtig sind:

```
Searching for SP on XX:XX:XX:XX:XX:XX
Channel: 5
```

Die "XX" müßt ihr natürlich mit der Ausgabe für eure Mac Adresse ersetzen.
Weiter gehts mit dem Anlegen der Schnittstelle in Linux für ExtGPS. Die "0"
steht für
die Nummer der Schnittstelle, danach folgt die MAC Adresse und dann die
Kanalnummer

```
rfcomm bind 0 XX:XX:XX:XX:XX:XX 5
```

Nun können wir unsere GPS Verbindung testen:

```
cat /dev/rfcomm0
```

Es sollten nun eine Menge GPS Daten angezeigt werden. Die Anzeige mit Strg + C beenden.

Die Datei /etc/bluetooth/rfcomm.conf editieren:

```
#
# RFCOMM configuration file.
#

rfcomm0 {
#   # Automatically bind the device at startup
    bind yes;
#
#   # Bluetooth address of the device
    device XX:XX:XX:XX:XX:XX;
#
#   # RFCOMM channel for the connection
    channel 5;
#
#   # Description of the connection
    comment "N82 GPS";
#}
```

Wenn im Handy ExtGPS gestartet ist und dann den Computer startet, sollte nun automatisch das Device /dev/rfcomm0 vorhanden sein. Bei mir hat das nicht funktioniert.

Deshalb habe ich in /etc/rc.d/rc.local den folgenden Eintrag hinzugefügt:

```
rfcomm bind 0 XX:XX:XX:XX:XX:XX 5
```

Damit klappt die Autostartverbindung von /dev/rfcomm0.

Nutzung eines „normalen“ Bluetooth-GPS-Empfängers

- Empfänger einschalten

```
hcitool scan
Scanning ...
XX:XX:XX:XX:XX:XX NOKIA N82
```

Da ich leider keinen BT-GPS Empfänger besitze, muß ich hier mit den Ausgabe meines Handys weiter machen. Wir suchen den seriellen Port:

```
sdptool browse
Inquiring ...
Browsing XX:XX:XX:XX:XX:XX ...
Service Name: Symarctic ExtGPS
Service Description: Share phone's built-in GPS module via Bluetooth
Service Provider: Symarctic Solutions
Service RecHandle: 0x10011
Service Class ID List:
```

```
"Serial Port" (0x1101)
Protocol Descriptor List:
  "L2CAP" (0x0100)
  "RFCOMM" (0x0003)
    Channel: 5
Language Base Attr List:
  code_ISO639: 0x656e
  encoding:    0x6a
  base_offset: 0x100
```

Wichtig sind:

```
Searching for SP on XX:XX:XX:XX:XX:XX
Channel: 5
```

Die "XX" müßt ihr natürlich mit der Ausgabe für eure Mac Adresse ersetzen. Weiter geht es mit dem Anlegen der Schnittstelle in Linux für ExtGPS. Die "0" steht für die Nummer der Schnittstelle, danach folgt die MAC Adresse und dann die Kanalnummer

```
rfcomm bind 0 XX:XX:XX:XX:XX:XX 5
```

Nun können wir unsere GPS Verbindung testen:

```
cat /dev/rfcomm0
```

Es sollten nun eine Menge GPS Daten angezeigt werden. Die Anzeige mit Strg + C beenden.

Die Datei /etc/bluetooth/rfcomm.conf editieren:

```
#
# RFCOMM configuration file.
#
rfcomm0 {
#   # Automatically bind the device at startup
  bind yes;
#
#   # Bluetooth address of the device
  device XX:XX:XX:XX:XX:XX;
#
#   # RFCOMM channel for the connection
  channel 5;
#
#   # Description of the connection
  comment "N82 GPS";
#}
```

Wenn ihr nun den Bluetooth-GPS-Empfänger einschaltet und dann den Computer gestartet, sollte nun automatisch das Device /dev/rfcomm0 vorhanden sein. Bei mir hat das nicht funktioniert.

Deshalb habe ich in /etc/rc.d/rc.local den folgenden Eintrag hinzugefügt:

```
rfcomm bind 0 XX:XX:XX:XX:XX:XX 5
```

Damit klappt die Autostartverbindung von /dev/rfcomm0.