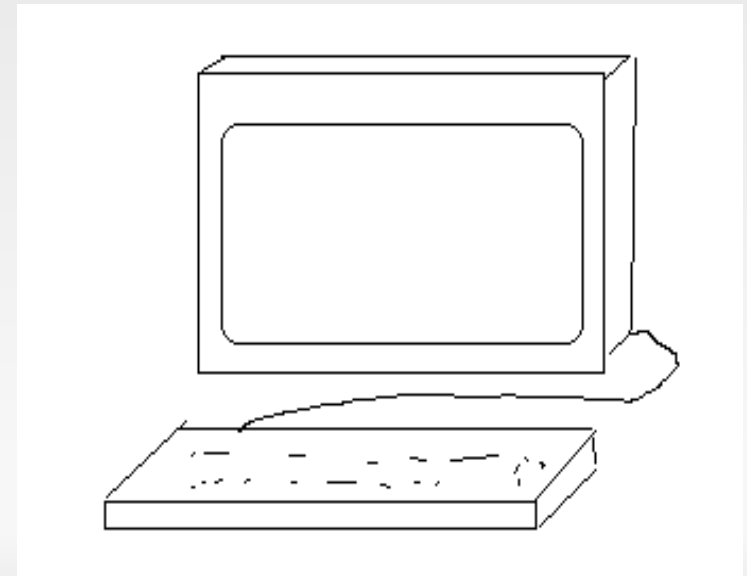
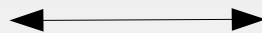
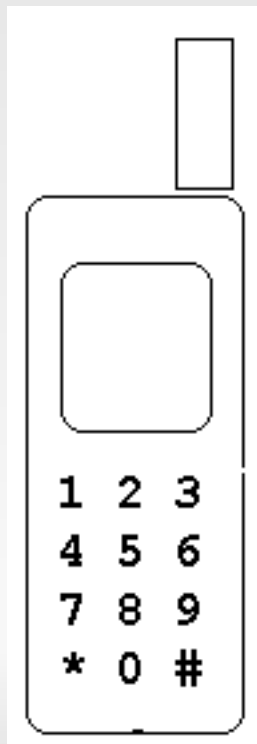


# Accessing Phone Data by Perl and Bluetooth

*Detlev Hauschildt*



# Why?

Lots of data on mobile phones:

- addresses
- SMS'es
- call log
- ...

But:

- fiddly to edit with phone keys...
- phones come with clumsy data exchange packages

# Could this be true...?

## Widely reported Dutch survey:

(August 2008; 1017 respondents between 16 and 30)

- Apparently, 1 in 3 women hand-copy SMS'es before deleting them...
- Apparently, 17% of men keep SMS'es that can be used to blackmail somebody!

### Jongeren koesteren oude sms-berichten (video) ◦

Uitgegeven: 29 augustus 2008 08:09

Laatst gewijzigd: 29 augustus 2008 11:58

**AMSTERDAM - Waren het vroeger liefdesbrieven die eeuwig in de la bleven liggen, nu zijn het sms-berichten waar jongeren moeilijk afstand van kunnen doen.**



[Bekijk video](#)

Een op de drie vrouwen vindt het opschonen van hun inbox zo erg dat ze berichten eerst overschrijven, ontdekte Hi na onderzoek.

Bijna alle jongeren laten sms-berichten in hun telefoon staan om ze later nog eens na te kunnen lezen. Als de telefoon daardoor vol loopt, is het wissen vervolgens een zorgvuldig en tijdrovend karwei. Bijna de helft van de jongeren loopt de sms'jes stuk voor stuk na om te bepalen welke in aanmerking komen voor de prullenbak.

Aan het onderzoek deden 1017 jongeren tussen de 16 en 30 jaar mee. Zij gaven aan vooral sms-verkeer van hun geliefde, vrienden of familie te bewaren.

#### Exen

Acht op de tien vrouwen bewaart vooral lieve sms'jes, tegenover zeven op de tien mannen. Sommige mannen (17 procent) bewaren sms'jes waarmee ze iemand kunnen chanteren. Tegenwoordig kunnen ze dat ook op internet doen in de Hi bewaarbox, die zelfs een categorie voor exen biedt.

De meeste sms'jes lijken te verdwijnen bij de aanschaf van een nieuwe telefoon, want de oudste sms is in 58 procent van de gevallen niet ouder dan zes maanden. Toch is er ook een vrij grote groep van 19 procent die aangeeft dat het oudste sms-bericht in hun telefoon een tot drie jaar oud is.

© ANP

# Bluetooth Serial Port

Conventional serial ports:

- look like:
- have names like:  
`/dev/ttyS0`, `/dev/ttyS1...` or `COM1`, `COM2...`



Bluetooth serial ports:

- nice exercise in backwards compatibility
- names like `/dev/rfcomm0`, `/dev/rfcomm1...` or `COM5`, `COM6...`
- make Bluetooth connections look like any conventional serial port to software accessing them

# How to get a Bluetooth serial port?

- Local device (if present) can always do it
- Remote device must offer it, but GSMs usually should
- Creating the connection is a task of the operating system, for example:

Windows:

Install “Widcomm Bluetooth Stack”

Find Remote Device under “Entire Bluetooth Neighborhood”

Click on it, to see whether it says “Serial Port on <device>”

Right-click, connect

Check the COM-port under Properties

Ubuntu/Debian:

```
sudo apt-get install bluetooth
```

```
sudo hcitool scan
```

(find `bluetooth addr` of device)

```
sdptool browse 00::11::22:33:44:55
```

```
locate channel nr for “Serial Port”
```

```
rfcomm connect /dev/rfcomm0
```

```
00:11::22:33:44:55 2
```

(port nr chosen above)

# Query phone in terminal program

# Query a phone interactively in terminal program 'minicom':

# (Windows: hyperterminal)

# First set it up properly

```
C-Ubuntu:/home/detlev> sudo minicom -s
```

```
# interactively:
```

```
# Serial port setup
```

```
# Serial Device: /dev/rfcomm0
```

```
# Bps/Par/Bis: 9600 8N1
```

```
# Save setup as dfl
```

```
# Exit from Minicom
```

```
C-Ubuntu:/home/detlev> minicom
```

```
Welcome to minicom 2.3
```

```
[...]
```

```
OK
```

```
AT+CGMM
```

```
SAMSUNG SGH-G600
```

```
OK
```

```
--> minicom_log_00
```

# How to do this in Perl?

## Three layers:

**Device::SerialPort** (non-Windows) or

**Win32::SerialPort** (Windows)

- replaces the terminal program,  
i.e. writes raw bytes to and reads raw bytes from the serial port

**Device::Modem**

- communicates through Device::SerialPort or Win32::SerialPort
- handles any AT-commands,  
i.e. sends them to the port and keeps checking for answers until “OK” or “ERROR” is found
- also keeps track of timeouts

**Device::Gsm**

- subclass of Device::Modem
- parses the answers to GSM-specific AT-commands and processes them into more user-friendly data structures

# Demo of Device::Modem

Short script that sends "AT+CCLK?" to the phone and shows the answer

→assumes that phone is connected to /dev/rfcomm0

```
$ perl modem.pl -p 0 -c "AT+CCLK?"
```

```
-----
```

```
question: 'AT+CCLK?'
```

```
-----
```

```
+CCLK: "09/03/05,18:47:56+00"
```

```
OK
```

```
-----
```

--> [source of modem.pl](#)

# Demo of Device::Gsm

Short script that shows the answer to “AT+CCLK?”, the current network and lists unpacked SMS messages:

```
$ perl gsm.pl -p 0 -c "AT+CCLK?"
```

```
    [...] same as before
```

```
$ perl gsm.pl -p 0 -c network
```

```
-----
```

```
question: 'network'
```

```
-----
```

```
NL KPN
```

```
-----
```

```
$ perl gsm.pl -p 0 -c messages -b ME
```

```
"","POSTBANK","Totaalbedrag overboekingen E 363,00.. Volgnummer 97; TAN-code  
737308.", "29/06/08 23:36:40 04", "0"
```

```
"+33686811111", "", "Mi jxus sendis msg-on... Detlev@", "", "1"
```

--> [source of gsm.pl](#)

# Summary

## Steps:

- Turn on Bluetooth on PC and phone
- Associate a serial port (/dev/rfcomm0, COM5...) with the phone
- To query phone interactively start up terminal program like
  - minicom or hyperterminal
- To query phone from Perl use **Device::SerialPort** or **Win32::SerialPort**
- To make exchanging AT commands a bit easier use **...::SerialPort** indirectly via:
  - Device::Modem**
- For GSM phones, add even more convenience by using subclass of **Device::Modem**
  - Device::Gsm**
- Questions...?