Technische Fakultät
Albert-Ludwigs-Universität, Freiburg

Lehrstuhl für Kommunikationsysteme
Prof. Dr. Gerhard Schneider

Master thesis

Mobile Assisted GPS Localization in GSM Networks

May 25, 2012

Refik Hadžialić

Supervised by
M.Sc. Konrad Meier
M.Sc. Dennis Wehrle

First Examiner
Prof. Dr. Gerhard Schneider

Second Examiner
Prof. Dr. Christian Schindelhauer

Erklärung

Hiermit erkläre ich, dass ich diese Abschlussarbeit selbständig verfasst habe, keine anderen als die angegebenen Quellen/Hilfsmittel verwendet habe und alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten Schriften entnommen wurden, als solche kenntlich gemacht habe. Darüber hinaus erkläre ich, dass diese Abschlussarbeit nicht, auch nicht auszugsweise, bereits für eine andere Prüfung angefertigt wurde.

Ort, Datum
(Place, Date)

Unterschrift
(Signature)

Acknowledgment

The author would like to thank his supervisors Konrad Meier and Dennis Wehrle for their help and support during the thesis work. Beside the help from the supervisors the author would like to thank his family and friends who supported him through his master studies and the entire department for the support, free coffee and to Prof. Dr. Gerhard Schneider for making available all the required hardware.

Contents

CONTENTS

1. Introduction to GSM and GPS

What use is knowledge if there is no understanding?

(Stobaeus)

1.1. Motivation

1.2. Goals of the thesis

The goal of the following thesis is to: - implement the Radio Resource Location Protocol inside of OpenBSC, to the extent of delivering correct GPS assistance data to cell phone subscribers inside the GSM network - test the protocol on 5-10 different smart phones - describe and analyze the background processes taking place inside of the cell phone

1. Introduction to GSM and GPS

2. Assisted GPS

2. ASSISTED GPS

3. Radio Resource Location Protocol

3. Radio Resource Location Protocol

4. Working

4.1. Zitieren..

citep: [Kopka 1997]
citet: Kopka [1997]

5. System

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.Test test

Referenz für lorem ipsum

6. Software

Author's test system operated on the ARFCN 877 channel. ARFCN (Absolute Radio Frequency Channel Number) defines the uplink and downlink channel frequency insdide the GSM network [Zahoransky 2011]. ARFCN 877 corresponds to the uplink frequency of 1,783.2 MHz and a downlink frequency of 1,878.2 MHz, where the uplink direction represents the direction from the nanoBTS to the mobile stations and downlink the opposite direction. The decision to use the ARFCN 877 channel was derived from the fact that the channel was free, measurements were carried out with a spectrum analyzer built on the USRP hardware.

6. SOFTWARE

7. Hardware

In the following chapter the author will introduce the reader to the hardware components used in the thesis. The hardware components will be presented according to their importance of building an operational and functional GSM network with GPS localization capabilities. Firstly the nanoBTS will be introduced since it is the main hardware component used for building a basic GSM network infrastructure. Then a short insight into the used GPS receiver will be given. Additionally the mobile stations used for testing of the system will be reviewed. Finally, a hardware connection diagram will be given.

7.1. GSM BTS - nanoBTS

In recent years, there has been an increasing interest in deployment of private cellular networks in remote areas or for research which lead to the devolopment of diverse "low-cost" GSM hardware solutions. According to ip.access[1], the manufacturer of nanoBTS, their hardware product is deployed for coverage of "hard-to-reach places; in-buildings; remote areas; marine and aviation; and public spaces". A nanoBTS with its plastic cover can be seen in Figure 7.1. Our University GSM network consists of three nanoBTS stations. The deployed nanoBTS in author's thesis works in the 1800 MHz frequency range, for which the University of Freiburg had obtained a licence from the Federal Network Agency (German: *Bundesnetzagentur*). The transmission frequencies range between 1805-1880 MHz, with 200 KHz channel spacing and maximal output power of +13 dBm (≈20 mW), whereas the receiving frequencies lie in the range between 1710-1785 MHz and same channel spacing as for transmission of 200 KHz [ip.access ltd 2007].

Check the output powere 20 dBm

Add the Abis over IP protocol

The nanoBTS is equiped with an internal 0 dBi (nominal) omni-directional antenna. However, two external antennas sized 30x36 mm, one for transmission (TX) and the other one for reception (RX) of radio waves were used to extend the coverage area. These antennas are connected via the SMA connectors. By using an RF

[1] http://www.ipaccess.com

Figure 7.1.: nanoBTS with its plastic cover. Image courtesy of ip.access ltd

Check for what NWL is

amplifier and larger antennas, for these frequency ranges, the covered area with the GSM signal reception can be increased. For the gain estimation and radiation angle of the used antennas the measurement equipment was missing and therefore was not conducted and described in this work.

At the bottom of the nanoBTS there are 5 ports, as seen in Figure 7.2. The ports from left to right are: voltage supply, ethernet cable with power supply, USB port, TIB-IN and TIB-OUT. In the next paragraph a brief overview of each port will be given.

The left most port is the power supply port used for supplying the nanoBTS with 48 V DC and is optionally used depending on the cable configuration. In author's hardware configuration the power supply port is not used. The following port is for the ethernet connection with 48 V DC power supply. This port is connected to a power supply that is supplied with the nanoBTS. It extends the ethernet connection with 48 V DC for the normal operation mode of the nanoBTS which is in the range between 38-50 V DC. The power consumtion of the nanoBTS is 13 W. More details on how to interconnect the cables will be given in section 7.3. In the middle of the five port region, the mini USB port can be found. It is used by the manufacturer to write the firmware software to the nanoBTS. The last two ports are the TIB-IN and TIB-OUT port[2]. These two ports are used if the GSM network operator requires more than 11 channels to increase the overall capacity of the network. "Up to 4 nanoBTS can be combined into a multiple TRX cell, increasing the number of supported users per TRX by up to 200%. The TIB-OUT from the Master TRX must be connected to the TIB-IN of the slave TRX. This in turn has its TIB-OUT connected to the next TRX in the chain" [ip.access ltd 2005]. The multiple TRX cell configuration will not

[2]TIB stands for Timing Interface Bus

Figure 7.2.: nanoBTS with two external antennas and five connection ports

be further discussed in this work since the purpose of the work was not to boost the capacity of a GSM network but implementation and testing of the RRLP protocol.

To determine the working state of the nanoBTS, an indicator status LED is located on the left side of the five ports region. After the nanoBTS is connected to the power suplly with the ethernet cable, it will change its color and blink speed according to the state it is in. The states can be seen in the Table given in 7.1 [ip.access ltd 2009].

One of the key limitations of gathering more technical data and the critical aspect of this description lies in the fact, that nanoBTS is not an open source hardware platform and ip.access does not offer more details on their product. The lack of systematic hardware analysis can be seen as a major drawback of working with the nanoBTS hardware. However, the given technical data are sufficient for reproducing and conducting the RRLP tests described in this thesis.

Table 7.1.: Indicator LED status on the nanoBTS

State	Color & Pattern	When	Precedence
Self-test failure	Red - Steady	In boot or application code when a power on self-test fails	1 (High)
Unspecified failure	Red - Steady	On software fatal errors	2
No ethernet	Orange - Slow flash	Ethernet disconnected	3
Factory reset	Red - Fast blink	Dongle detected at start up and the factory defaults have been applied	4
Not configured	Alternating Red/Green - Fast flash	The unit has not been configured	5
Downloading code	Orange - Fast flash	Code download procedure is in progress	6
Establishing XML	Orange - Slow blink	A management link has not yet been established but is needed for the TRX to become operational. Specifically: for a master a Primary OML or Secondary OML is not yet established; for a slave an IML to its master or a Secondary OML is not yet established.	7
Self-test	Orange - Steady	From power on until end of backhaul powe on self-test	8
NWL-test	Green - Fast flash	OML established, NWL test in progress	9
OCXO Calibration	Alternating Green/Orange - Slow blink	The unit is in the fast calibrating state [SYNC]	10
Not transmitting	Green - Slow flash	The radio carrier is not being transmitted	11
Operational	Green - Steady	Default condition if none of the above apply	12 (Low)

7.2. GPS Receiver - NL-402U

In the next paragraphs the used GPS device will be described. In contrast to the earlier described hardware, nanoBTS, which the University of Freiburg already owned, the budget for the GPS receiver was limited and the Navilock NL-402U was bought considering only the single criterion, the price. The Navilock NL-402U GPS receiver is based on the u-blox UBX-G5000 single chipset and is a one chip solution [u-blox AG 2009]. It can be seen on Figure 7.3 with its passive ceramic patch antenna. 1575,42 MHz is the operating frequency of the receiver which corresponds to the L1 civil frequencies and Coarse/Acquisition (C/A) code. The GPS chipset consists of 50 channels, each channel tracks the transmission from a single satellite [Elliott D. Kaplan 2006]. It is important to note, the number of channels inside a GPS receiver interrelates with the amount of time required to get the first fix. Receiver tracking sensitivity is -160 dBm (10^{-16} mW). The GPS receiver communicates with the computer ovet the USB port. Although the GPS receiver uses an USB interface, on the computer it emulates 2 UART ports, which are serial communication interfaces.

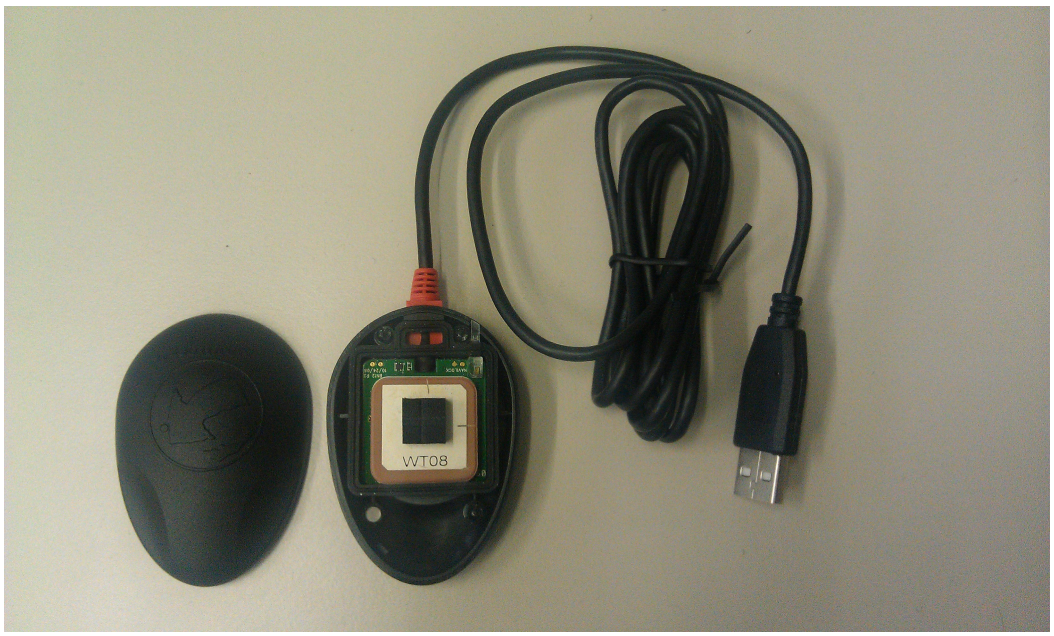

Figure 7.3.: Navilock NL-402U, opened up with the antenna and USB cable

7.3. Cable configuration

In the next section, the author will focus on properly connecting the hardware. At least 4 ethernet cables with RJ45 connectors, on both sides, were required and one switch or hub connected to the internet. One should take notice of the cabling between the nanoBTS and the ethernet switch or hub, since wrong cabling with the power supply unit (PSU) could damage one of the devices. In Figure 7.4, the junction points are label according to the used configuration setting. The ethernet cables between the switch/hub, PSU and nanoBTS should not be longer than 100 m [ip.access ltd 2009].

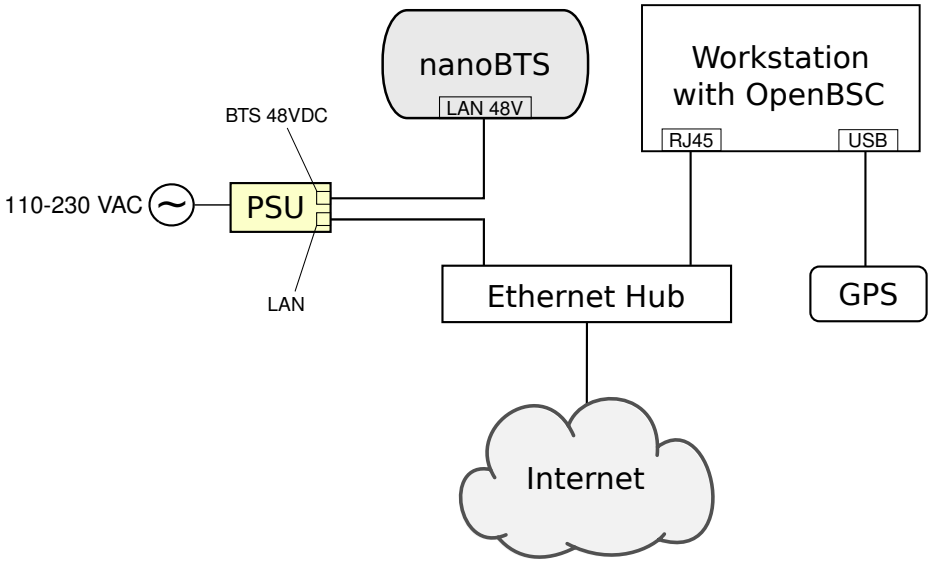

Figure 7.4.: Cable connections, showing interconnection diagram

8. Implementation

8. Implementation

9. Future work

9. Future work

10. Summary

10. SUMMARY

Dictionary of acronyms

- *ARFCN* - Absolute Radio Frequency Channel Number - The channel number specifies the physical frequency channel used for transmission and reception of radio waves inside of an BTS covered area.
- *BTS* - Base Transceiver Station -
- *DC* - Direct Current
- *GNSS* - Global Navigation Satellite System - A satellite navigation system that allows a specialized receive to determine its location on Earth.
- *LED* - Light Emitting Diode - A diode that emitts light.
- *IP Address* - .
- *PCB* - Printed Circuit Board - The board where electronic components are soldered onto and wired through conductive tracks.
- *RRLP* - Radio Resource Location Protocol - The employed protocol in GSM, UMTS and other wireless networks for providing and exchange of geolocation information.
- *SMA* - SubMiniature version A - SMA is a connector used for interconnecting coaxial cables or PCB electronics that work in the frequency range between 0-18 GHz.
- *TIB* - Time Interface Bus - The TIB is used to provide the synchronization of the clock, frequency and frame number between the nanoBTS when operating in a single 2-4 BTS configuration.
- *TRX* -
- *UART* - Universal Asynchronous Receiver Transmitter - A serial communication interface used by computers or other peripheral devices to communicate.
- *UMTS* - Universal Mobile Telecommunications System - Third generation mobile network based on the GSM standards.

Write what an IP address is

10. SUMMARY

Appendix

A. Installation and configuration guide

In order to evaluate the localization system, it is required to install OpenBSC and to modify the proper source files and compile the system. The aim of this section is to describe that process in such detail that the presented material is sufficient to reproduce equivalent or similar results. The guide was successfully tested out on the following operating systems: Ubuntu 10.04 LTS 64 bit and Ubuntu 12.04 LTS 64 bit. A self-bootable test USB system is supplied with the thesis and it can be evaluated without executing the given steps in A.1.

A.1. Installation of OpenBSC

In order to compile OpenBSC it is required to install the following precompiled packages[1]:

- libdbi0
- libdbi0-dev
- libdbd-sqlite3
- libortp-dev
- build-essential
- libtool
- autoconf
- automake
- git-core
- pkg-config

Before installing the required packages and libraries, to keep the installation process clean and free of modifying other files, the author will create a new directory.

[1] If more details are required for the installation process a guide can be found at [osmocom].

```
mkdir gsm_localization
cd gsm_localization
```

By executing the following instructions the required libraries will be installed.

```
sudo apt-get install libdbi0-dev libdbd-sqlite3 build-essential
sudo apt-get install libtool autoconf automake git-core
sudo apt-get install pkg-config libortp-dev
```

After the packages were installed, *libosmocore* library must be downloaded, compiled and installed. By executing the following instructions:

```
git clone git://git.osmocom.org/libosmocore.git
cd libosmocore
autoreconf -fi
./configure
make
sudo make install
sudo ldconfig
cd ..
```

In the next step *libosmo-abis* will be installed.

```
git clone git://git.osmocom.org/libosmo-abis.git
cd libosmo-abis
autoreconf -fi
./configure
make
sudo make install
sudo ldconfig
cd ..
```

After the previous steps have finished successfully, the author will proceed with downloading, compiling and installing OpenBSC.

```
git clone git://git.osmocom.org/openbsc.git
cd openbsc/openbsc
autoreconf -i
sudo export PKG_CONFIG_PATH=/usr/local/lib/pkgconfig
./configure
make
```

At this point, OpenBSC should be successfully compiled.

A.2. Configuring nanoBTS for OpenBSC

To enable the nanoBTS and OpenBSC to be fully operational, the last configuration steps have to be made. It is necessary to inform the nanoBTS of the IP address of the server that is running OpenBSC since it must connect to OpenBSC. We need to find a free ARFCN channel where our system is expected to operate[2].

To find the ID and the IP address of the nanoBTS it is required to start *ipaccess-find*[3].

```
cd ~/gsm_localization/openbsc/openbsc/src/ipaccess
./ipaccess-find
```

ipaccess-find will produce an output similar to the one given:

```
Trying to find ip.access BTS by broadcast UDP...
MAC_Address='00:02:95:00:61:70'  IP_Address='132.230.4.63'
Unit_ID='1801/0/0'  Location_1=''  Location_2='BTS_NBT131G'
Equipment_Version='165g029_73'
Software_Version='168a352_v142b30d0'
Unit_Name='nbts-00-02-95-00-61-70'
Serial_Number='00110533'
```

In the next step, the nanoBTS is informed of the OpenBSC IP address by typing the following commands (the first IP address belongs to the server running OpenBSC and the second to the nanoBTS):

```
cd ~/gsm_localization/openbsc/openbsc/src/ipaccess
./ipaccess-config -o 132.230.4.65 132.230.4.63 -r
```

It is required to create the directory where the configuration file will be located and to modify the configuration file.

```
sudo mkdir /usr/local/lcr
cd ~/gsm_localization/openbsc/openbsc/doc/
cd examples/osmo-nitb/nanobts
sudo cp openbsc.cfg /usr/local/lcr
sudo vim /usr/local/lcr/openbsc.cfg
```

[2]A licence has to be obtained from the Federal Network Agency (German: *Bundesnetzagentur*), otherwise it is ilegal and may be considered as a criminal act.

[3]The nanoBTS ought to be blinking in orange color before starting *ipaccess-find*.

A free ARFCN channel can be found using a spectrum analyzer and by setting the frequency range to the GSM frequency band. One has to slide through the frequencies shown on the X-axis, and by looking at the Y-axis with appropriate frequency resolution[4], where the received power is represented[5]. By patiently observing the Y-axis it can be easily seen on the X-axis which channels are taken by other GSM service providers and which are free. The chosen channel ought to be peak free. Once a free frequency channel has been found, it is necessary to instruct the nanoBTS to operate in that frequency range. The line, numbered 58, has to be modified with the correct free ARFCN channel,in this case 877[6].

```
arfcn 877
```

On line 53, the last configuration file modification has to be made. The Unit ID from the output above has to be set[7].

```
ip.access unit_id 1801 0
```

At this point the nanoBTS and OpenBSC configuration is done.

[4]The frequency resolution must be set to f_{CB} = 200 KHz or more for faster movement in the frequency spectrum.

[5] Dependent of the manufacturer and settings of the spectrum analyzer, it can show signal amplitude, magnitude and power.

[6] A table with frequency channels can be found at the following URL: https://gsm.ks.uni-freiburg.de/arfcn.php or it can be calculated using the given formulas $f_{up}(ARFCN) = f_{start} + f_{CB} \cdot (ARFCN - Offset)$, for the uplink where f_{start} = 1710.2 MHz is the starting frequency of the uplink bandwitdh for DCS1800, f_{CB} = 200 KHz is the channel bandwidth and $Offset$ = 512; whereas for downlink $f_{down}(ARFCN) = f_{start} + f_{CB} \cdot (ARFCN - Offset)$ where f_{start} = 1805.2 MHz and the rest of the variables remain same.

[7]Indentation has to match the one of the configuration file.

A.3. Installation and configuration of GNSS assistance software

To install the RRLP software that generates GNSS assistance data several libraries are required to be installed, *cURL*[8], *libconfig* and *SQLite*. *cURL* was used for the purpose of safely downloading GNSS data from the Navigation Center of the US Coast Guard and Trimble server. *libconfig* library is used for reading in the configuration file, this way compiling of the software whenever one changes the settings was avoided. The *SQLite* library was employed to access the database used by OpenBSC to store the respondence data from the mobile stations.

```
cd ~/gsm_localization
sudo apt-get install libsqlite3-dev
wget http://curl.haxx.se/download/curl-7.25.0.tar.gz
wget http://www.hyperrealm.com/libconfig/libconfig-1.4.8.tar.gz
tar -xvzf curl-7.25.0.tar.gz
tar -xvzf libconfig-1.4.8.tar.gz
cd curl-7.25.0
make
sudo make install
cd ..
cd libconfig-1.4.8/
./configure
make
sudo make install
```

Once the libraries have been successfully installed, the user may proceed with the configuration and compiling the GNSS assistance software, which is the key software produced in this thesis.

```
// An example configuration file for the GNSS RRLP software.
name = "Configuration for GNSS and RRLP";

// Change the settings if required:
settings =
{
  config = ( {
    ephemeris_url  = "ftp://ftp.trimble.com/pub/eph/CurRnxN.nav";
    almanac_url = "http://www.navcen.uscg.gov/ ↙
       ↳ ?pageName=currentAlmanac&format=yuma";
    latitude_of_BTS  = 48.003601;
```

[8] It may happen that the given download URLs are wrong and in the meantime have changed, but one can easily find the latest versions on http://curl.haxx.se/ and http://www.hyperrealm.com/libconfig/

```
    longitude_of_BTS    = 7.848056;
    altitude_of_BTS  = 0.0;
    uncertainty_of_lat_long  = 7;
    uncertainty_of_alt  = 7;
    confidence_level    = 0;
    ephemeris_repair  = false;
    use_reference_time  = false;
    extra_seconds_to_add  = 7;
    timezone_of_BTS  = 1;
    time_to_refresh_ephem = 1;
    time_to_refresh_alm = 1 ; } );
};
```

B. Sourcecode

Beispiel:

```
#include <stdio.h>

int main(void)
{
    printf("Hallo Welt!\n");
    return 0;
}
```

APPENDIX

Bibliography

u-blox AG 2009
AG u-blox: *UBX-G5010, G5000/G0010*. http://www.texim-europe.com/promotion/560/ubx-g5010%20datasheet_te.pdf. 2009. – [Online; accessed 5-April-2012]

Diggelen 2009
Diggelen, Van: *A-GPS assisted GPS, GNSS, and SBAS*. Boston : Artech House, 2009. – ISBN 1596933747

Elliott D. Kaplan 2006
Elliott D. Kaplan, Christopher H.: *Understanding GPS : principles and applications*. Boston : Artech House, 2006. – ISBN 1580538940

Harper 2010
Harper, Neil: *Server-side GPS and assisted-GPS in Java*. Boston : Artech House, 2010. – ISBN 9781607839859

ip.access ltd 2005
ip.access ltd: *GSM-over-IP picocells for in-building coverage and capacity*. 2005. – URL http://www.hexazona.com/nexwave/docs/ipaccess/nanoBTS%201800-1900.pdf

ip.access ltd 2009
ip.access ltd: *nanoBTS Installation Manual*. 2009. – URL http://subversion.assembla.com/svn/bxpgfKRFar3O9EeJe5afGb/PP/ipaccess/NGSM_INST_300_nanoBTS_Install_v3_0.pdf

Kopka 1997
Kopka, H.: LATEX Band 1: Einführung. (1997)

ip.access ltd 2007
ltd ip.access: *The world's most deployed picocell*. http://www.ipaccess.com/en/nanoGSM-picocell. 2007. – [Online; accessed 3-April-2012]

osmocom
OSMOCOM: *OpenBSC build guide*. Web. – URL http://openbsc.osmocom.org/trac/wiki/Building_OpenBSC. – [Online; accessed 22-May-2012]

Zahoransky 2011
ZAHORANSKY, Richard M.: *Localization in GSM Mobile Radio Networks*, University of Freiburg, Diplomarbeit, 2011