Technische Fakultät
Albert-Ludwigs-Universität, Freiburg

Lehrstuhl für Kommunikationsysteme
Prof. Dr. Gerhard Schneider

Master thesis

Mobile Assisted GPS Localization in GSM Networks

June 13, 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

I would like to thank my supervisors Konrad Meier and Dennis Wehrle for their encouraging talks during the thesis. Things which have not been done before are intellectually seductive in a way. Beside the help from the supervisors I would like to thank my family and friends who supported me through my master studies, and the entire Communication systems department for their support, free coffee and to Prof. Dr. Gerhard Schneider for making all the required hardware available. I would like to thank Sebastian Schmelzer for his LaTeX tips, Michael Neves Pereira and Jonathan Bauer for borrowing me their cell phones to test my system with and Johan Latocha for patiently explaining me words I did not understand in the German language.

Contents

1. Introduction to GSM and GPS **1**
1.1. Motivation 1
1.2. Goals of the thesis 1

2. Assisted GPS **3**
2.1. GPS Principles 3
2.2. GPS signal modulation 3
2.3. GPS signal demodulation 3
2.4. Distance and position estimation 9

3. Radio Resource Location Protocol **11**

4. Working **13**
4.1. Zitieren.. 13

5. System **15**

6. Software **17**

7. Hardware **19**
7.1. GSM BTS - nanoBTS 19
7.2. GPS Receiver - NL-402U 23
7.3. Cable configuration 24

8. Implementation **25**

9. Future work **27**

10. Summary **29**

Appendix **33**
A. Installation and configuration guide 33
A.1. Installation of OpenBSC 33
A.2. Configuring nanoBTS for OpenBSC 35

A.3. Installation and configuration of GNSS assistance software . . 37
B. Sourcecode 42
C. GPS Constants 42

Bibliography **43**

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

2.1. GPS Principles

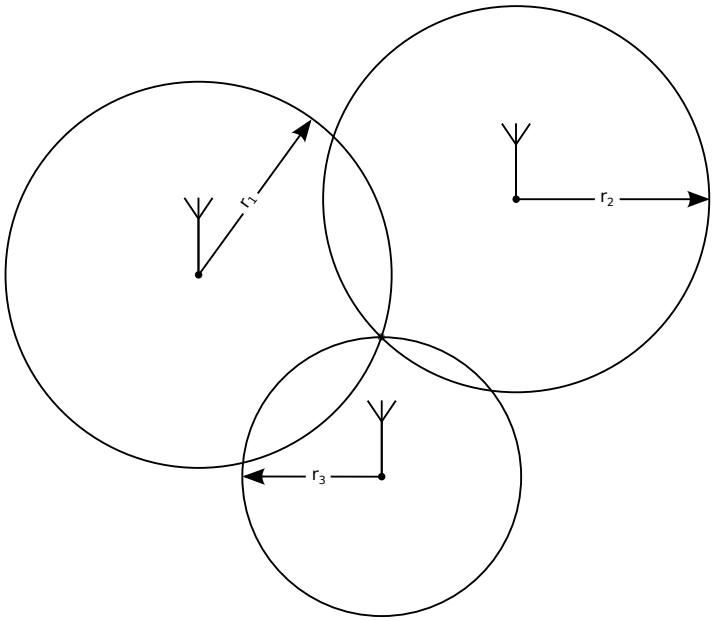

Figure 2.1.: nanoBTS with its plastic cover. Image courtesy of ip.access ltd

2.2. GPS signal modulation

2.3. GPS signal demodulation

The GPS satellites[1] orbiting our planet, at a distance of approximately 20,200 km, are equiped with precise atomic clocks [5, Chapter 2.7]. These atomic clocks are

[1]Satellites are named as space vehicles and the abrevation SV is used in the equation notations to denote a parameter related to the satellite itself.

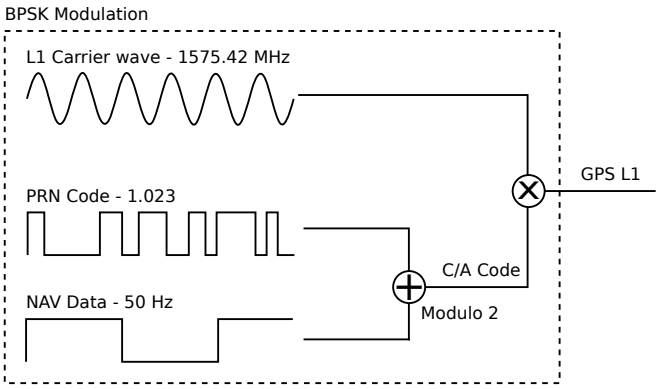

Figure 2.2.: Modulation of the GPS signal L1

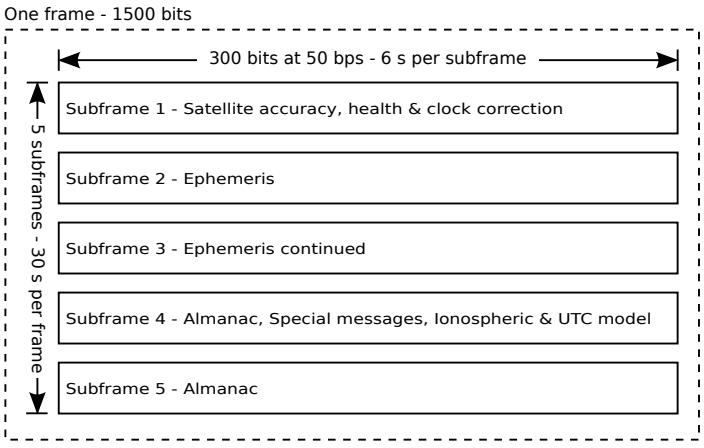

Figure 2.3.: One frame of 1500 bits on L1 frequency carrier

calibrated and maintained on a daily basis by the U.S. Air Force, [7]. The time the clock generates is called *GPS system time*, denoted as t_{SV}, and it is generated as a time stamp at the moment of the frame broadcast [2]. Each satellite signs the frame with its exact broadcast time. The broadcast time is encapsulated in the subframe 1 of the 1500 bit long frame. In addition to the broadcast time, subframe 1 contains parameters to account for the deterministic clock errors embedded in the broadcasted GPS system time stamp. These errors can be characterized as bias, drift and aging errors [2]. The correct broadcast time, denoted as t, can be estimated using the model equation given in (2.1) [2]. In equation (2.2), where the GPS receiver is required to calculate the satellite clock offset, denoted as Δt_{SV}, a number of unknown terms can be seen. These terms are encapsulated in the subframe 1 or they can be estimated using predefined equations. The polynomial coefficients: a_{f0} - *clock offset*, a_{f1} - *fractional frequency offset*, a_{f2} - *fractional frequency drift*; and t_{0c} -

reference epoch are encapsulated inside of subframe 1. Finally, the only unknown term left in equation (2.2) is the *relativistic correction term*, denoted as Δt_r. Δt_r can be evaluated by applying the equation given in (2.3). F is a constant calculated from the given parameters in (C.0.6) and (C.0.7), whereas e, $\sqrt{A}$ and E_k are *orbit parameters* encapsulated in subframe 2 and 3 [2].

$$t = t_{SV} - \Delta t_{SV} \tag{2.1}$$

$$\Delta t_{SV} = a_{f0} + a_{f1}(t_{SV} - t_{oc}) + a_{f2}(t_{SV} - t_{oc})^2 + \Delta t_r \tag{2.2}$$

$$\Delta t_r = F e \sqrt{A} \sin E_k \tag{2.3}$$

$$F = \frac{-2\sqrt{\mu_e}}{c^2} = -4.442807633 \cdot 10^{-10} \frac{s}{\sqrt{m}} \tag{2.4}$$

However, the broadcast satellite time information is not sufficient to estimate the precise time at the moment of the signal arival. Even though the signal arives in approximately 77 ms, the precision of the atomic clock is in the range of 10 ns [5, Chapter 2]. Undoubtedly the signal propagation (travel) time, denoted as t_{prop}, has to be taken into account. Then the exact time at the moment of arival, denoted as t_{exact}, is given in equation (2.5). The signal propagation time must be known to estimate the distance from the satellite as well as to estimate the position of the GPS receiver.

$$t_{exact} = t_{prop} + t \tag{2.5}$$

In order to calculate the signal propagation time between the satellite and the receiver, the internal clock wave of the of the receiver crystal needs to be synchronized with the carrier clock wave of the satellite [17]. In other words, the identical carrier wave replica has to be generated on the receiver as on the satellite. Due to the nature of wave propagation and various errors the signal arives phase disordered at the receiver [17]. The observed phase at the receiver antenna, denoted as φ_o, can be described using the equation given in (2.6), where φ_{GPS} represents the known satellite carrier wave phase, $\delta\varphi_{SV}$ the clock instabilities on the GPS satellite, φ_a the phase shift error caused by propagation delays in the ionosphere and troposphere respectively and $\delta\varphi_w$ is the wideband noise.

$$\varphi_o = \varphi_{GPS} + \delta\varphi_{SV} + \varphi_a + \delta\varphi_w \tag{2.6}$$

The task of the syncrhonization process is to generate a replica carrier wave with the matching phase shift. In the ideal case, the observed phase on the antenna and

the generated phase on the receiver, denoted as φ_r, cancel each other out, in other words, equation (2.7) equals to zero.

$$\Delta\varphi = \varphi_o - \varphi_r \tag{2.7}$$

If this property is not satisfied, it is not possible to demudalte the C/A code from

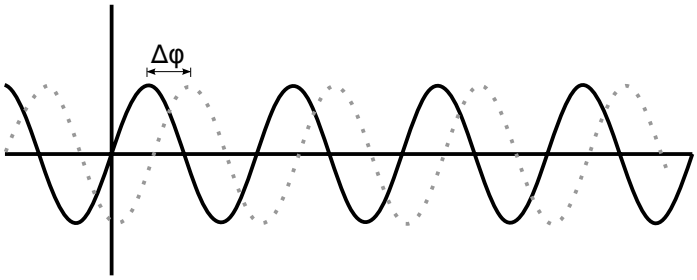

Figure 2.4.: Two equivalent carrier waves with phase shift

the received signal.

More importantly, t_{exact} is used to synchronize various system dependent.

$$
\begin{aligned}
A &= (\sqrt{A})^2 \\
n_0 &= \sqrt{\frac{\mu}{A^3}} \\
t_k &= t - t_{oe} \\
n &= n_0 + \Delta n \\
M_k &= M_0 + nt_k \\
M_k &= E_k - e\sin E_k \\
v_k &= \tan^{-1}\left(\frac{\sin v_k}{\cos v_k}\right) = \tan^{-1}\left(\frac{\frac{\sqrt{1-e^2}\sin E_k}{1-e\cos E_k}}{\frac{\cos E_k - e}{1-e\cos E_k}}\right) \\
v_k &= \tan^{-1}\left(\frac{\sin v_k}{\cos v_k}\right) = \tan^{-1}\left(\frac{\sqrt{1-e^2}\sin E_k/(1-e\cos E_k)}{(\cos E_k - e)/(1-e\cos E_k)}\right) = \tan^{-1}\left(\frac{\sqrt{1-e^2}\sin E_k}{\cos E_k - e}\right) \\
E_k &= \cos^{-1}\left(\frac{e+\cos v_k}{1+e\cos v_k}\right) \\
\Phi_k &= v_k + \omega \\
\delta u_k &= c_{us}\sin 2\Phi_k + C_{us}\cos 2\Phi_k \\
\delta r_k &= c_{rc}\cos 2\Phi_k + C_{rs}\sin 2\Phi_k \\
\delta i_k &= c_{ic}\cos 2\Phi_k + C_{is}\sin 2\Phi_k \\
u_k &= \Phi_k + \delta u_k \\
r_k &= A(1 - e\cos E_k) + \delta r_k \\
i_k &= i_0 + \delta i_k + (IDOT)t_k \\
x_k^{'} &= r_k\cos u_k \\
y_k^{'} &= r_k\sin u_k \\
\Omega_k &= \Omega_0 + (\Omega - \Omega_e)t_k - \Omega_e t_{oe} \\
x &= x_k^{'}\cos\Omega_k - y_k^{'}\cos i_k\sin\Omega_k \\
y &= x_k^{'}\sin\Omega_k - y_k^{'}\cos i_k\cos\Omega_k \\
z &= y_k^{'}\sin i_k
\end{aligned}
\tag{2.8}
$$

The received signal after the RF frontend is given in equation (2.9) [8].

$$S(t) = \sqrt{\frac{P}{2}}D(t)C(t)cos(2\pi f_c + \varphi_{SV}) + n(t) \tag{2.9}$$

Each tracked GPS satellite signal is demodulated seperately using the same PRN code, code chipping rate and carrier frequency phase for the given satellite [6, Chapter

4]. The PRN codes for each GPS satellite are well defined and known by the GPS receiver. The receiver has to generate the same PRN code with matching code chipping rate (phase) of the C/A code, this is depicted in figure 2.5 [6, Chapter 5]. For the particular example, the matching phase shift was achieved with the second

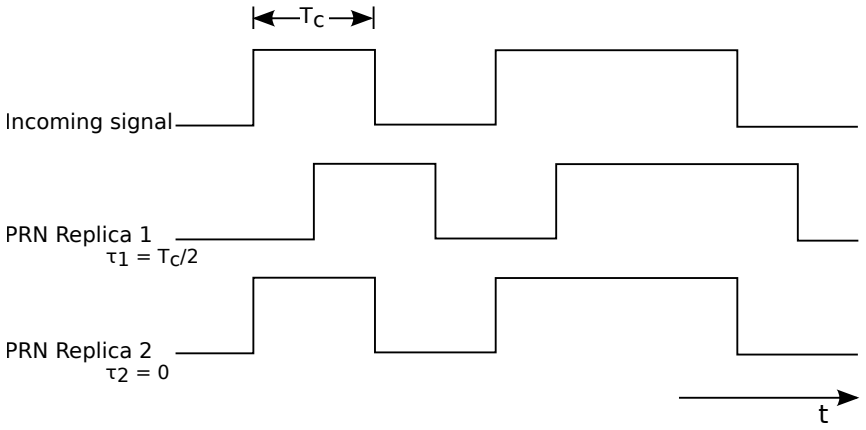

Figure 2.5.: Comparison of original C/A code generated on the GPS satellite with two synthesized PRN codes with phase shift on the receiver

replica PRN code, with a phase shift of $\tau = 0$ but there could be a case with any other value of τ, $\tau \in [0, 1023]$. The PRN code synthesizer implementation depends on the GPS receiver manufacturer but it is usually implemented as a linear feedback shift registers (LFSR) that produces an output according to a predefined function $f(\tau)$. This function generates an PRN code, that is delayed in phase by τ, where τ is a multiple of the chipping period $T_c = 977.5ns$. The chipping period T_c can be derived from equation (2.10). The time required to find a matching PRN code shift (τ) is proportional to the amount of LFSR on the system [4, Chapter 3]. Particularly with more LFSRs the required time for finding the matching phase shift increases.

$$T_c = \frac{1}{f_{PRN}} = \frac{1}{1.023 \cdot 10^6} \tag{2.10}$$

To determine whether the synthesized PRN code, matches the incoming C/A code from the satellite, known correlation properties of PRN codes are used. Since the signal is modeled as a sequence of +1's and -1's, the autocorrelation of a signal is at its maximum if it is in phase, i.e. summing up the sequence products yields the absolute maximum value. As an illustration of the idea, an example is given in figure 2.6. The cross-correlation of the incoming C/A code with the first synthesized PRN produces a result of $-3 = (+1)\cdot(-1)+(-1)\cdot(+1)+(+1)\cdot(-1)+(+1)\cdot(+1)+(-1)\cdot(+1)$, whereas the cross-correlation of the incoming C/A code and the second synthesized PRN code

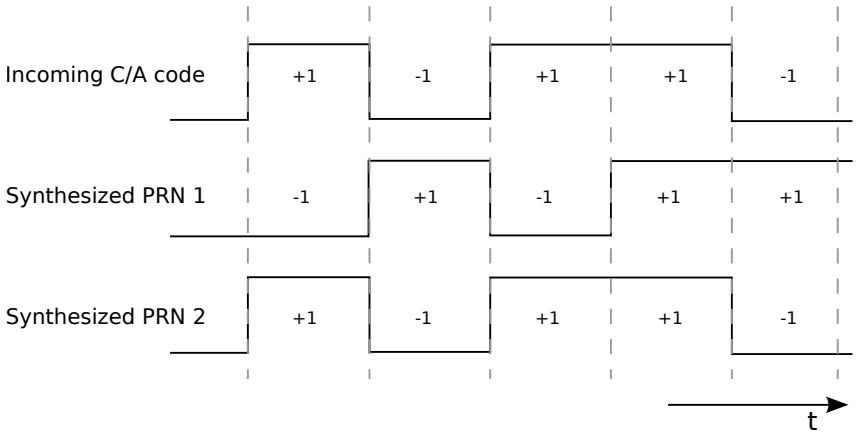

Figure 2.6.: Cross-correlation on three different signals

yields a result of $+5 = (+1)\cdot(+1)+(-1)\cdot(-1)+(+1)\cdot(+1)+(+1)\cdot(+1)+(-1)\cdot(-1)$. The same principle applies to the sent C/A and PRN code sequences in the GPS receiver and thus can be modeled using the equation given in (2.11), where, $G_i(t)$ is the C/A code Gold code sequence as a function of time, t, for the GPS satellite i; $T_{C/A}$ is the C/A chipping period of $977.5ns$ and τ is the phase shift in the auto-correlation function [6, Chapter 4].

$$R_i(t) = \frac{1}{1023 \cdot T_{C/A}} \int_{t=0}^{1023} G_i(t) G_i(t+\tau) d\tau \tag{2.11}$$

Another correlation property of the PRN codes comes in useful, the fact that in the ideal case the cross-correlation of two different PRN codes yields a result of zero. The ideal case can be modeled as in equation (2.12),

$$R_{ij}(\tau) = \int_{-\infty}^{+\infty} PRN_i(t) PRN_j(t+\tau) d\tau = 0 \tag{2.12}$$

where PRN_i is the PRN code waveform for GPS satellite i and PRN_j is the PRN code waveform for every other GPS satellite other than i, $i \neq j$ [6, Chapter 4]. Equation (2.12) "states that the PRN waveforms of satellite i does not correlate with PRN waveform of any other satellite for any phase shift τ" [6, Chapter 4]. Without this property, the GPS receiver would not be able to smoothly differentiate between best phase shifts.

2.4. Distance and position estimation

2. ASSISTED GPS

3. Radio Resource Location Protocol

3. Radio Resource Location Protocol

4. Working

4.1. Zitieren..

citep: [15]
citet: Kopka [15]

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 [19]. 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 [12].

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" [11]. The multiple TRX cell configuration will not be further

[2]TIB stands for Timing Interface Bus

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

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 [13].

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 [18]. 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 [6]. 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.

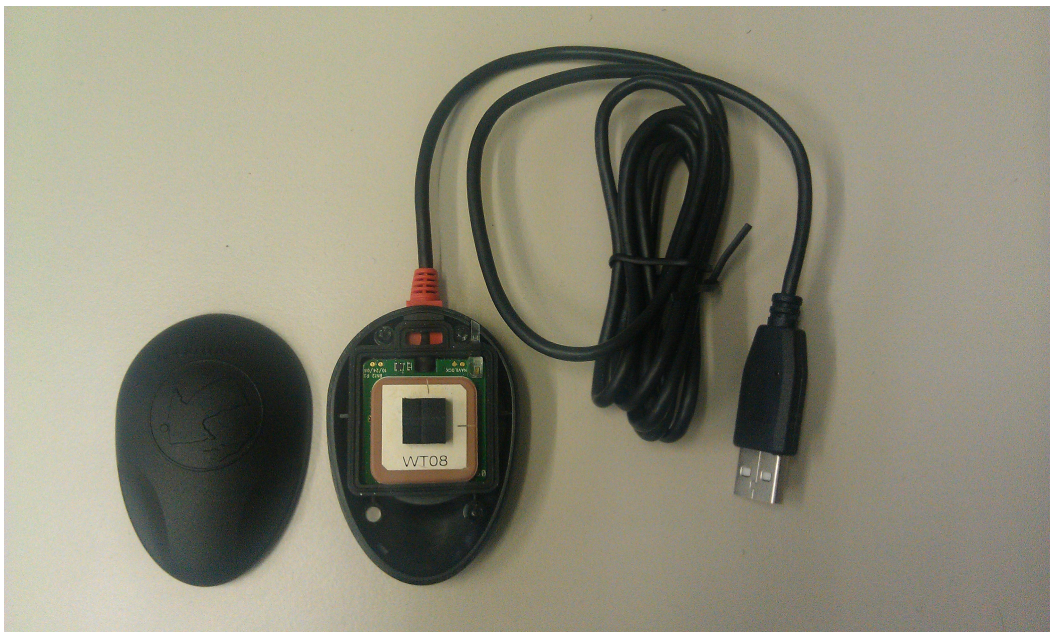

Figure 7.3.: Navilock NL-402U, opened up with the antenna and USB cable

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 [13].

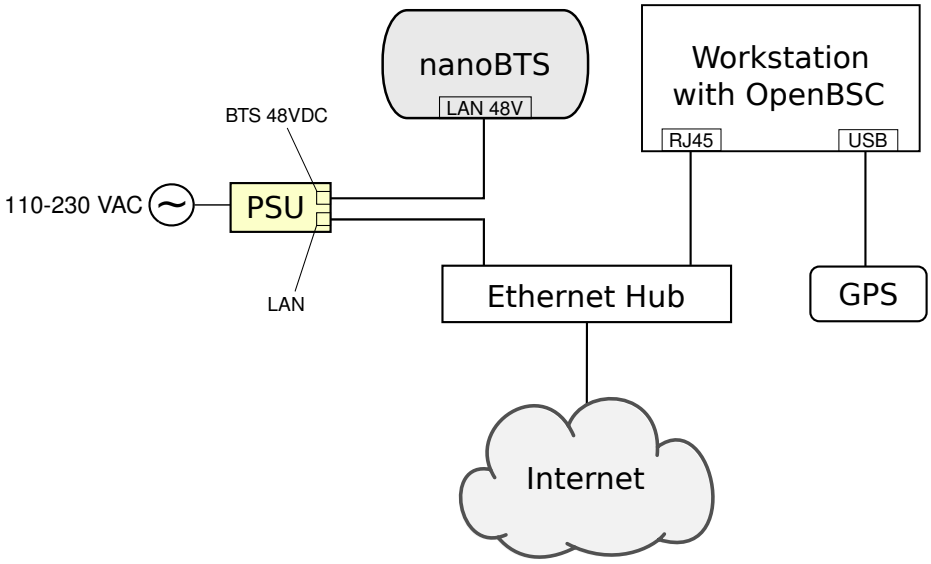

Figure 7.4.: Cable connections, showing interconnection diagram

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. There is a marking difference between text given in light and dark grey background color, the first ought to be typed in into the terminal window or it may be an output produced by an application, whereas the later emphasizes a file modification case.

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

[1] If more details are required for the installation process a guide can be found at [16].

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.

```
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.

```
arfcn 877
```

The ARFCN channel value can be calculated using the given formula in (A.2.1), where f_{start} is the starting frequency of the uplink bandwitdh for DCS1800, f_{CB} is the channel bandwidth and *Offset* is the offset[6].

$$f_{up}(\text{ARFCN}) = f_{start} + f_{CB} \cdot (\text{ARFCN} - \text{Offset})$$

$$where \begin{cases} f_{start} &= 1710.2\,\text{MHz} \\ f_{CB} &= 200\,\text{KHz} \\ \text{Offset} &= 512 \end{cases} \tag{A.2.1}$$

On line numbered 53, the last configuration file modification has to be made for the final configuration of the OpenBSC software. 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 $f_{CB} = 200\,\text{KHz}$ or higher values 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

[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. The configuration file can be found in the same directory as the RRLP modules under the name: "gnssrrlp.cfg". The sample configuration file is already preconfigured for the location of "Angewandte Mathematik und Rechenzentrum" building. Latitude and longitude of the BTS are expressed in decimal degrees and are bounded by $\pm 90°$and $\pm 180°$respectively. Positive latitudes are north of the equator, whereas negative are south of the equator. It is alike for longitude coordinates, positive longitudes are east of Prime Meridian and negative are west of the Prime Meridian. If the position in decimal degrees of the BTS is unknown, it is straightforward to derive them using the formula given in (A.3.1), where D are

[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/

degrees, M are minutes and S are seconds[9].

$$DD = D + \frac{M}{60} + \frac{S}{3600} \tag{A.3.1}$$

Describe other parameters as well.

The altitude may be left as it is, set to 0, since it is not used in the current measurement technique[10].

```
// 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;
    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 ; } );
};
```

CHECK IF THIS IS CORRECT

CHECK IF THIS IS CORRECT

The target user, one wants to locate, has to be inside of a geometric estimated shape. This shape can be described using an ellipsoid point with altitude and uncertainty ellipsoid. The uncertainty of the latitude and longitude correctness can be described using equation (A.3.2) [3]. The uncertainty of r is expressed in meters, it defines how accurate is the specified location of the BTS. In the configuration file, K is set to 7, which corresponds to $r = 9.4872$ m. Instead of using the integer parameter

[9] An online converter of the Federal Communication Commission can be used as well to convert from degrees, minutes and seconds to decimal degrees and vice versa http://transition.fcc.gov/mb/audio/bickel/DDDMMSS-decimal.html

[10] If the value is set to zero, it is important to set it to 0.0 because *libconfig* would otherwise convert it to an integer however it is a floating point number.

K as the known variable, the equation (A.3.2) can be rewritten as in (A.3.3), where we can get the integer value K for a previously selected r.

$$r = C((1+x)^K - 1)$$

$$where \begin{cases} C &= 10 \\ x &= 0.1 \\ K &\in [0, 127] \cap \mathbb{N}_0 \end{cases} \tag{A.3.2}$$

$$K = \left\lceil \frac{ln(\frac{r}{C} + 1)}{ln(1+x)} \right\rceil$$

$$where \begin{cases} C &= 10 \\ x &= 0.1 \\ r &\in [0, 1800]\,\text{km} \end{cases} \tag{A.3.3}$$

A set of uncertainties r is given in table A.3.1 for various integer values of K.

Value of K	Value of uncertainty r
0	0 m
1	1 m
2	2.1 m
3	3.3 m
-	-
20	57.3 m
-	-
60	3.0348 km
-	-
100	137.8 km
-	-

Table A.3.1.: Example uncertainties (latitude and longitude) for various integer values of K

Altitude uncertainty can be described using the same Binomial expansion method, as given in (A.3.4), however with altered constant values [3]. The altitude uncertainty ranges between 0 m and 990.5 m ($h \in [0, 990.5]$ m). Although the same constant name K is used, it describes the altitude uncertainty, (A.3.5).

$$h = C((1+x)^K - 1)$$

$$where \begin{cases} C & = & 45 \\ x & = & 0.025 \\ K & \in & [0,127] \wedge \|K\| \end{cases} \tag{A.3.4}$$

$$K = \left\lceil \frac{ln(\frac{h}{C}+1)}{ln(1+x)} \right\rceil$$

$$where \begin{cases} C & = & 45 \\ x & = & 0.025 \\ h & \in & [0,990.5]\,\mathrm{m} \end{cases} \tag{A.3.5}$$

A set of uncertainties h is given in table A.3.2 for various integer values of K.

Value of K	Value of uncertainty h
0	0 m
1	1.13 m
2	2.28 m
3	3.46 m
-	-
20	28.74 m
-	-
60	152.99 m
-	-
100	486.62 m
-	-

Table A.3.2.: Example uncertainties (altitude) for various integer values of K

Confidence level is the next parameter in the configuration file that needs to be set. It can take any integer value between 0 and 127. The confidence level defines the percentage of the confidence that the target entity, the GSM user one wants to locate, is within the geometric shape defined earlier. A value of 0 and between 100 and 127, may be interpreted as "no information" [3]. The reason why the values are not limited to 100 is because of the nature of binary numbers and that 2^6 bits is not sufficient to represent the number 100, but rather requires one bit more.

Confidence level is followed by the ephemeris repair option. Ephemeris repair is a variable of the boolean type, it can take two different values *true* or *false*. Ephemeris

data may contain errors or miss some satellite information [14] [10] and the ephemeris repair function, if set to true, will take data of the previous measurement report. This introduces an error as well.

To increase the speed of measurement report, reference time can be used to provide extra information for the A-GPS in the MS of target entity. This field is of boolean type, if set to true, reference time is included in the sent packets.

Since the sent packets are not transmitted in real time but put on a stack and then sent to the MS, a time delay exists. A solution to this problem is to add extra seconds to the reference time being sent. In order to assess the amount of extra seconds to add, the GSM operator is required experimentally to verify his/her findings. .

see how much the reference time can deviate from current time

The reference time being sent to the MS is Coordinated Universal Time (UTC). The GPS device receives UTC time from the satellites and adjusts the computer time. To set the correct time, time zone offset of the BTS ought to be set correctly.

Finally, the refresh time of downloading new almanac and ephemeris data has to be set. The variable uses the hour unit, how often the data are being downloaded. If the data are used from a local GNSS station, refresh time of the ephemeris data should be set to every 30 minutes or 0.5 hours. The almanac data are valid for up to 180 days [1] but are updated usually every day[11] [7].

[11] Almanac update times can be found here: http://www.navcen.uscg.gov/?pageName=currentNanus&format=txt

B. Sourcecode

Example:

```
#include <stdio.h>

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

C. GPS Constants

$$\mu_e = 3.986004418 \cdot 10^{14} \frac{m^3}{s^2} \quad \Longleftarrow \quad \text{Geocentric gravitational constant} \tag{C.0.6}$$

$$c = 2.99792458 \cdot 10^{8} \frac{m}{s} \quad \Longleftarrow \quad \text{speed of light} \tag{C.0.7}$$

Bibliography

[1] NAVSTAR GPS USER EQUIPMENT INTRODUCTION. Online, Sept. 1996. URL http://www.navcen.uscg.gov/pubs/gps/gpsuser/gpsuser.pdf.

[2] INTERFACE SPECIFICATION IS-GPS-200. Online, Mar. 2006. URL NavstarGPSSpaceSegment/NavigationUserInterfaces.

[3] 3GPP-Coordinates. 3GPP TS 23.032 V6.0.0 (2004-12), 3rd Generation Partnership Project; Technical Specification Group Core Network; Universal Geographical Area Description (GAD) (Release 6). Technical report, Dec. 2004.

[4] A. Bensky. *Wireless positioning technologies and applications.* Artech House, Boston, Mass, 2008. ISBN 1596931302.

[5] V. Diggelen. *A-GPS assisted GPS, GNSS, and SBAS.* Artech House, Boston, 2009. ISBN 1596933747.

[6] C. H. Elliott D. Kaplan. *Understanding GPS : principles and applications.* Artech House, Boston, 2006. ISBN 1580538940.

[7] J. G. Grimes. GLOBAL POSITIONING SYSTEM STANDARD POSITIONING SERVICE PERFORMANCE STANDARD. Online, Sept. 2008. URL http://www.gps.gov/technical/ps/2008-SPS-performance-standard.pdf.

[8] X. Guan, D. Hu, and J. Chen. Design and implementation of the acquisition circuit in software gps receiver. In *Mobile Technology, Applications and Systems, 2005 2nd International Conference on*, pages 4 pp. –4, nov. 2005. doi: 10.1109/MTAS.2005.243823.

[9] N. Harper. *Server-side GPS and assisted-GPS in Java.* Artech House, Boston, 2010. ISBN 9781607839859.

[10] L. Heng, G. X. Gao, T. Walter, and P. Enge. GPS Ephemeris Error Screening and Results for 2006-2009. *ION Institute of Navigation Global Navigation Satellite Systems Conference*, 2010.

[11] ip.access ltd. GSM-over-IP picocells for in-building coverage and capacity, 2005. URL http://www.hexazona.com/nexwave/docs/ipaccess/nanoBTS;1800-1900.pdf.

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

[13] ip.access ltd. nanoBTS Installation Manual, 2009. URL http://subversion.assembla.com/svn/bxpgfKRFar3O9EeJe5afGb/PP/ipaccess/NGSM_INST_300_nanoBTS_Install_v3_0.pdf.

[14] D. C. Jefferson and Y. E. Bar-Sever. Accuracy and Consistency of Broadcast GPS Ephemeris Data. *ION Institute of Navigation International Technical Meeting*.

[15] H. Kopka. LATEX Band 1: Einführung. 1997.

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

[17] A. Razavi, D. Gebre-Egziabher, and D. Akos. Carrier loop architectures for tracking weak gps signals. *Aerospace and Electronic Systems, IEEE Transactions on*, 44(2):697 –710, april 2008. ISSN 0018-9251. doi: 10.1109/TAES.2008.4560215.

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

[19] R. M. Zahoransky. Localization in GSM Mobile Radio Networks. Master's thesis, University of Freiburg, 2011.