Implementation of Contemporary Cellular Network using USRP B200 SDR and OpenBTS

Submitted by

Nahid Mahumud Rakib (ID: 011161102)
Nahim Hasan (ID: 011161255)
Ruhul Amin (ID: 011161020)
Mamunur Rashid (ID: 011161263)
Ratri Saha (ID: 011153035)

Supervised by

Dr. A.K.M. Muzahidul Islam
Professor, Department of Computer Science and Engineering
United International University (UIU)

Submitted in partial fulfilment of the requirements
of the degree of Bachelor of Science in Computer Science and Engineering

May 13, 2022

Department of Computer Science and Engineering
United International University

Abstract

In this paper, we describe the design and implementation of a Global System for Mobile Communications (GSM + GPRS) mobile network using Software Defined Radio (SDR). The network Base Transceiver System (BTS) based on Software Dened Radio (SDR) is build using Universal Software Radio Peripheral (USRP) B200 board and OpenBTS. OpenBTS, GNU radio with USRP Hardware Driver (UHD), Asterisk and Fosphor are used to control and configure the USRP B200 SDR on the software side. After establishing the cellular GSM network, short message service, data service and voice call can be realized. In addition to OpenBTS, a single typical application SMQueue and Asterisk are also used for the correct routing of messages and calls. Owning a free GSM cell phone can extend the connection to almost everyone in the disaster area. This low-cost and economical system will encourage governments to apply this model in disaster-prone areas. The results show that this autonomous communication infrastructure is suitable for the rapid installation of a complete mobile communication system with low power consumption, low-cost investment and maintenance.

Acknowledgements

We would like to start by expressing our deepest gratitude to the Almighty Allah for giving us the ability and the strength to finish the task successfully with in the scheduled time.

We would like to express our deep gratitude to our supervisor, Dr. A.K.M. Muzahidul Islam, Professor, Department of Computer Science and Engineering (CSE), United International University (UIU), for his continuous support, advice and care. His endless patience, guidance and energetic supervision, valuable advice at all state have made it possible to complete this project.

We would like to thank out entire course mate who took part in this discuss while completing the course work.

Finally, we would like to thank all the faculty member and staff of the CSE department of UIU, for their various support and constant cooperation.

https://www.overleaf.com/project/5de11e1f35faa10001ac7ab7

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Software-defined radio is a type of radio in which signals are processed by software on a computing platform (such as a PC). In an ideal SDR only the antenna, ADC and DAC is used as hardware for receiving and transmitting. The rest of the digital signal processing is done by a general purpose processor (PC or laptop)[1][2]. Being a re−configurable hardware, it supports operation using multiple protocols on different frequencies, and can adapt to different situations based on the needs of the user. We are using USRP B200 SDR throughout this project.

GSM is a second-generation network architecture for digital cellular networks[3][4][5]. GSM is used by mobile phones, tablets and IoT devices. We are using OpenBTS version 5's implementation of a GSM architecture.

Frequency modulation is a technique or a process of encoding information on a particular signal (analogue or digital) by varying the carrier wave frequency in accordance with the frequency of the modulating signal[6]. FM Radio broadcasting is analog frequency modulation. Where audio is FM encoded and transmitted for an FM receiver to demodulate the signal and reproduce the original audio signal. Using FM we can get lower noise and also provides high fidelity audio compared to AM[6]. FM radio was first developed by Edwin Armstrong in the 1930s[7]. Here we are developing FM receiver and transmitter using software radio.

For detecting signals in a given frequency area a spectrum analyzer can be used. It displays the signal amplitude on the vertical axis and the frequency on the horizontal axis. So, the activated RF signal is displayed as a horn on the vertical axis[8].

A Spectrum Analyzer can be used to measure frequency response, noise and distortion of RF hardware. Spectrum analyzers are also used to determine occupied bandwidth and interference source. Spectrum analyzers can also be used to test RF shielding, as it will be able to detect any RF signals that leaked through. It is also the basis of cognitive

radio, where the radio is aware of its environment and other radio signals and uses this information to reconfigure itself for avoiding interference with other devices.

1.1 Problem Statement

Implement a 2.5G GSM+GPRS network that is capable of registering devices with existing operator SIMs to provide SMS, voice and GPRS services. Voice and SMS services can be used for emergency communication during disasters or in a state of emergency. 850 MHz frequency and Absolute Radio Frequency Channel Number(ARFCN) 128 are used for setting up our base. Implementing an FM receiver to receive local FM transmissions.

1.2 Motivation

Technology has the power to do many things, and changing the world is one of them. Now a days, wireless communication plays a significant role in our life. Wireless communication uses radio wave to transfer data. The invention of radio waves has changed the way we share information.

According to the BTRC in October, 2019 Bangladesh has 164.170 million mobile phone subscribers[9] and 99.569 million internet subscribers[10] among 5 carriers. If somehow these carrier's operation is hampered by external forces such as natural disaster, it will disable communication for a lot of people. Even if we assume each mobile device has 2 SIM's there are 80.085 million devices in use. And all of them must have GSM, and FM radio receivers.

Bangladesh suffers a significant number of natural disasters each year, and because of climate change these disasters will increase day by day. Communication is an important requirement after a disaster. Disasters make it difficult for the general public to communicate with each other by disabling infrastructure in place for communication. For example, in case of Japans Fukushima nuclear plant disaster it disabled all cellular networks in the area. If something like this happens, we need a cheap and easy solution for reestablishing mobile communication. In such conditions we can deploy the GSM network and FM transmitter described in this report as a way of re-establishing emergency communication. There are a number of remote locations in Bangladesh where there are no cellular services. On such remote and small location, it is not viable for the big carriers to operate in because of low number of users and high cost. It is viable for these local communities to establish their own community networks which will be operated and financed by themselves[11][12]. We also like the idea of Software Defined Radio, as an open platform it holds enormous potential in wireless communication and research. We want more people to be excited about this device. For this reason, we are implementing practical and simple projects with the SDR.

1.3 Objectives

As it is our first project related to SDR, We need to explore and understand basic concept about all the resources about SDR. We set our initial target to implement GSM and further we will try to implement GPRS that will be our final goal. Our initial and final objectives are given below:

Initial Objectives : We are trying to implement a GSM cellular Network where we can communicate via SMS or Phone Call. For SMS and Voice call service we are using SMQueue and Asterisk respectively. The system will allow registration for new devices by SMS automatically.

Final Objectives : We selected our last target to implement 3G Cellular Network system, but it is not happening. Because our group members are in separate places for this pandemic situation. In 3G system there is no Phone call and SMS services but only have different GPRS system to increase the transfer speed than 2G. So we decided to include GPRS system into our implemented GSM base and got a better system having GSM + GPRS services.

Chapter 2

Background Studies

In this section, we discuss the GSM, SDR and OpenBTS architectures.

2.1 OpenBTS Architectures

OpenBTS is a software-based GSM access point that allows standard GSM compatible mobile phones to make calls without using existing telecommunications providers. OpenBTS software adopts GSM architecture, which has various components as shown in Figure 2.1[13]. OpenBTS manages the overall functions of the system, and implements GSM from the time division multiple access (TDMA) part of the L3/L4 boundary to the L3 (network layer) and L1 (physical layer) parts. It implements a transceiver at the bottom of the L1. You don't have to use the transceiver that with OpenBTS, and other transceivers can use OpenBTS. A PBX connects audio calls. The SIP (Session Initiation Protocol)

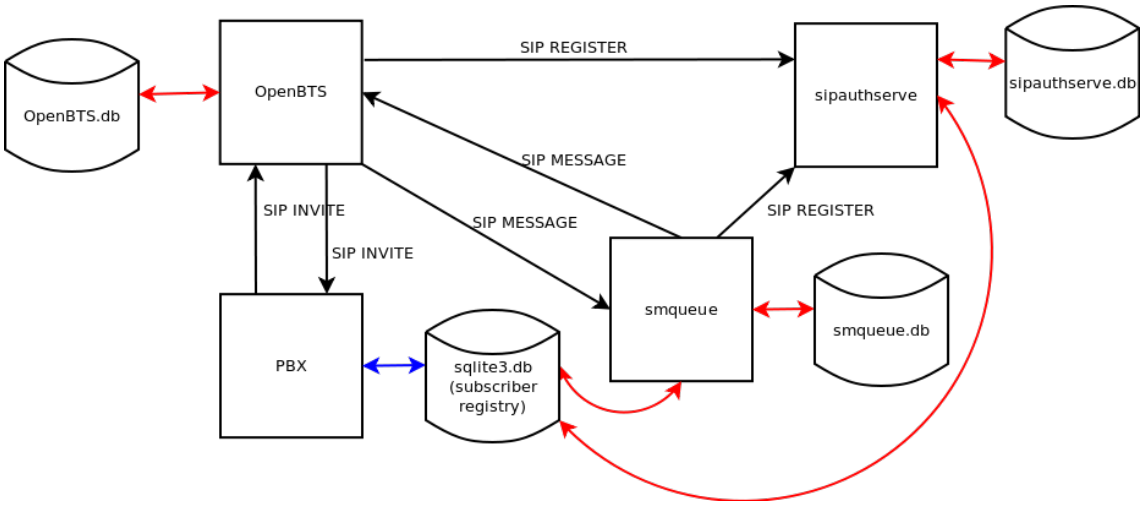

Figure 2.1: OpenBTS Architecture

Authentication Server is used for processing location updating requests from OpenBTS and save the updates to the subscriber registry database. The subscriber registration database contains network information based on the IMSI of the registered user. Smqueue

is an SMS quotation service that can enable SMS on the web. It saves the message and forwards it. If the target device is not currently accessible, it will store the message and deliver it when the target device is available. Network is not required, but text messaging is not supported without a network.

2.2 Global System for Mobile Communations (GSM)

The Global System for Mobile Communications (GSM) is a second-generation digital cellular communication standard developed by the European Telecommunications Standards Association (ETSI)[3][4][5]. It operates over the 900MHz or 1800MHz band. Figure 2.2 show the GSM Architechture. GSM (Global System for Mobile Communications) is a

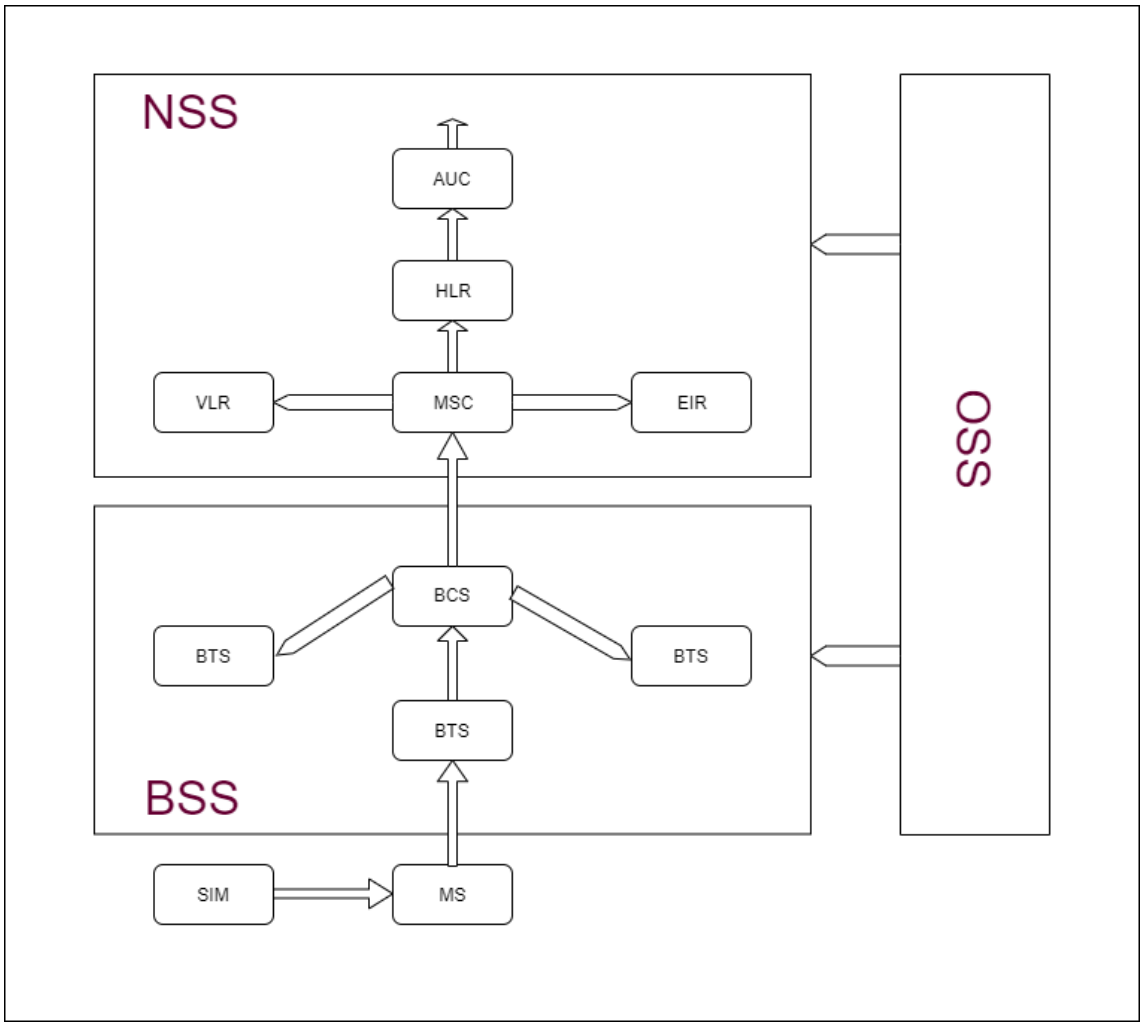

Figure 2.2: GSM Architecture

digital mobile network that has been widely used by mobile phone users in Europe and other parts of the world. The GSM time division department uses various multiple access (TDMA) and three digital wireless telephone technologies, namely TDMA, GSM and code division multiple access (CDMA). GSM digitizes and compresses the data, and then

sends it to a channel in each time slot, including two user data streams. GSM is an open digital cellular technology used to transmit mobile voice and data services operating in the 850MHz, 900MHz, 1800MHz and 1900MHz frequency bands. The GSM network is composed of four independent parts that work together to work as a whole, including the mobile device itself, the base station subsystem (BSS), the network switching subsystem (NSS), and the operation and support subsystem (OSS).

2.3 General packet Radio Service (GPRS)

GPRS architecture works on the same procedure like GSM network, but, has additional entities that allow packet data transmission. The data network overlaps with the second-generation GSM network and provides packet data transmission at a rate of 9.6 to 171 kbps.. Figure 2.3 shows all components of GPRS and how does it work.

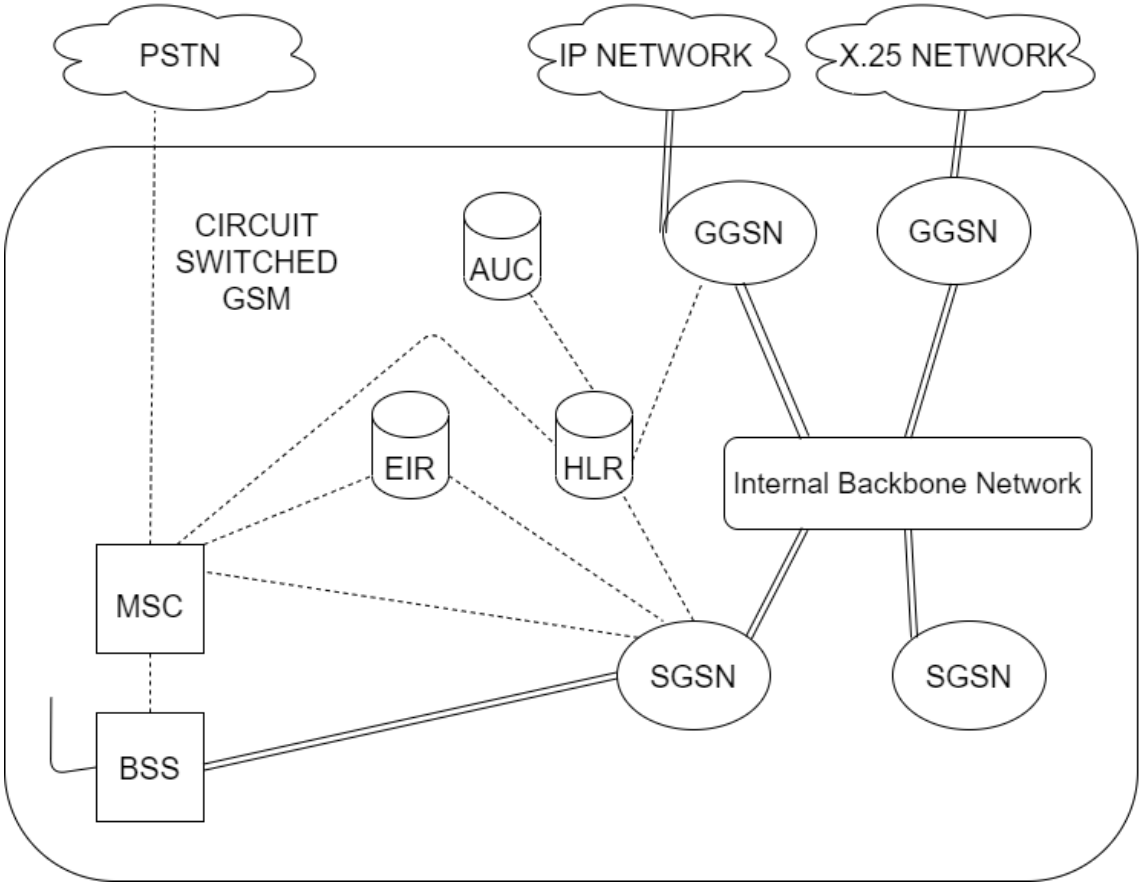

Figure 2.3: GPRS Architecture

Two new components are followed, called Gateway GPRS Support Node (GSN) and Serving GPRS Support Node (SGSN):

GPRS Support Node (GGSN)

The gateway GPRS support node acts as an interface and router in the external network. It contains routing information for GPRS mobile devices, which is used to tunnel data packets to the correct service through an IP-based internal backbone through a GPRS support node.GGSN also collects charging data related to the use of external data networks, and can act as a packet filter for incoming traffic.

Serving GPRS Support Node (SGSN)

The service GPRS support node is responsible for the identity verification of GPRS mobile phones, mobile phone registration on the network, mobile management and collection of air interface charging information

Internal Backbone

The internal backbone network is an IP-based network used to carry data packets between different GSNs. A tunnel is used between SGSN and GGSN, so the internal backbone network does not need any information about the external domain of the GPRS network. Any MSc, HLR or EIR signal from GSN is done using SS7.

2.4 Software Defined Radio (SDR)

Radio development re−configurability is not a new technique. 1980's re−configurable receivers were developed since the publication of the special issue on software radios of the IEEE Communication Magazine in April 1995.

If communication functions are realized as programs running on a suitable processor,then we call that as software radio (SR) Digital radio (DR) is invariably implemented on a digital processor by the base−band signal processing.The antenna output is named by SR.The practical version of an SR is mainly SDR.

It is often argued that SDR is the current perceptible version of SR, because the most advanced analog-to-digital (A/D) converter may not be used in SRs. Although this statement is correct, it may lead to a completely wrong conclusion that directly digitizing the SR output from the antenna should be the main goal of future development.

A cognitive radio (CR) is an SDR that additionally senses its environment, tracks changes, and reacts upon its findings.From our point of view, a CR is a refined SDR while this again represents a refined DR[14].

2.5 Comparison with some published papers

We are trying to show some differences between papers already published and our work. We are implementing the table in terms of output that we are achieving. Here is the table

2.1 that shows the differences as follows,

Paper Name	Operating System	Implementation	Messaging	Voice	GPRS
Reconfigurable cellular GSM network using USRP B200 and OpenBTS for disaster-hit regions[15]	Ubuntu 14.05 LTS	Yes	Yes	Yes	No
Cost effective restoration of wireless connectivity in disaster hit areas using OpenBTS[16]	Ubuntu 12.04 LTS	Yes	Yes	No	No
SDR-Based Reliable and Resilient Wireless Network for Disaster Rescue Operations[17]	Undefined	Yes	Yes	Yes	No
Implementation of Contemporary Cellular Network using USRPB200 SDR and OpenBTS (running)	Ubuntu 16.04 LTS	Yes	Yes	Yes	Yes

Table 2.1: Comparison Table

Chapter 3

Hardware and Software Tools Used

In this project we have used a number of hardware and software to implement the different systems. Discussion on those hardware and software follow.

3.1 Computer

A Lenovo yoga 500 laptop with a core i5 5200U processor and Intel HD Graphics 520 in processor running Ubuntu 16.04 LTS is used for running all the software. A USB 3.0 type A to micro-B cable is used to connect the USRP B200 with the laptop. In Figure 3.1 shows the operating system of our PC.

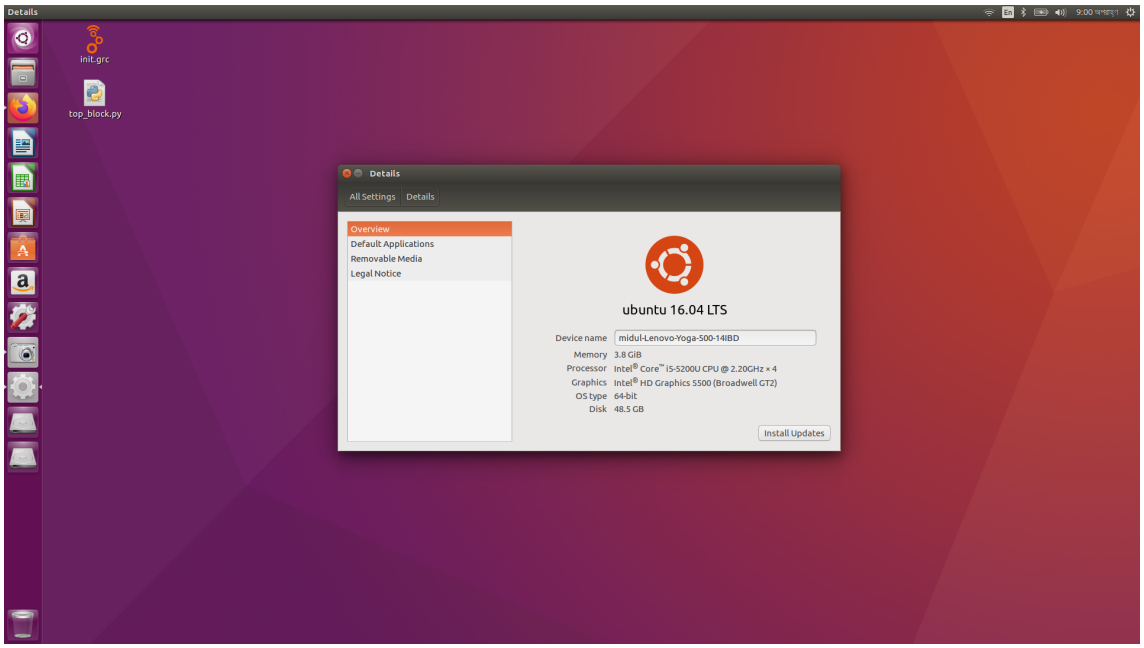

Figure 3.1: Computer Configuration

3.2 SDR Module

There are a lot of SDR modules available now-a-days at different price points. It is especially cost effective to buy cheap RTL-SDR dongles. However, those dongles usually only have a frequency range of 1 MHz to around 1200-2000 MHz and are incapable of transmitting signal. Consequently, they are not on the supported hardware list of OpenBTS. So, we could not use them in our project. We have also explored HackRF One which only supports half duplex operation so cannot be used for a GSM network, as GSM is full duplex. BladeRF 2.0, is capable of full duplex operations and is supported by OpenBTS. We also explored USRP B210, B200, N200, USRP 2900, USRP 2901 and other USRP SDRs. We can see USRP B200 SDR in Figure 3.2.

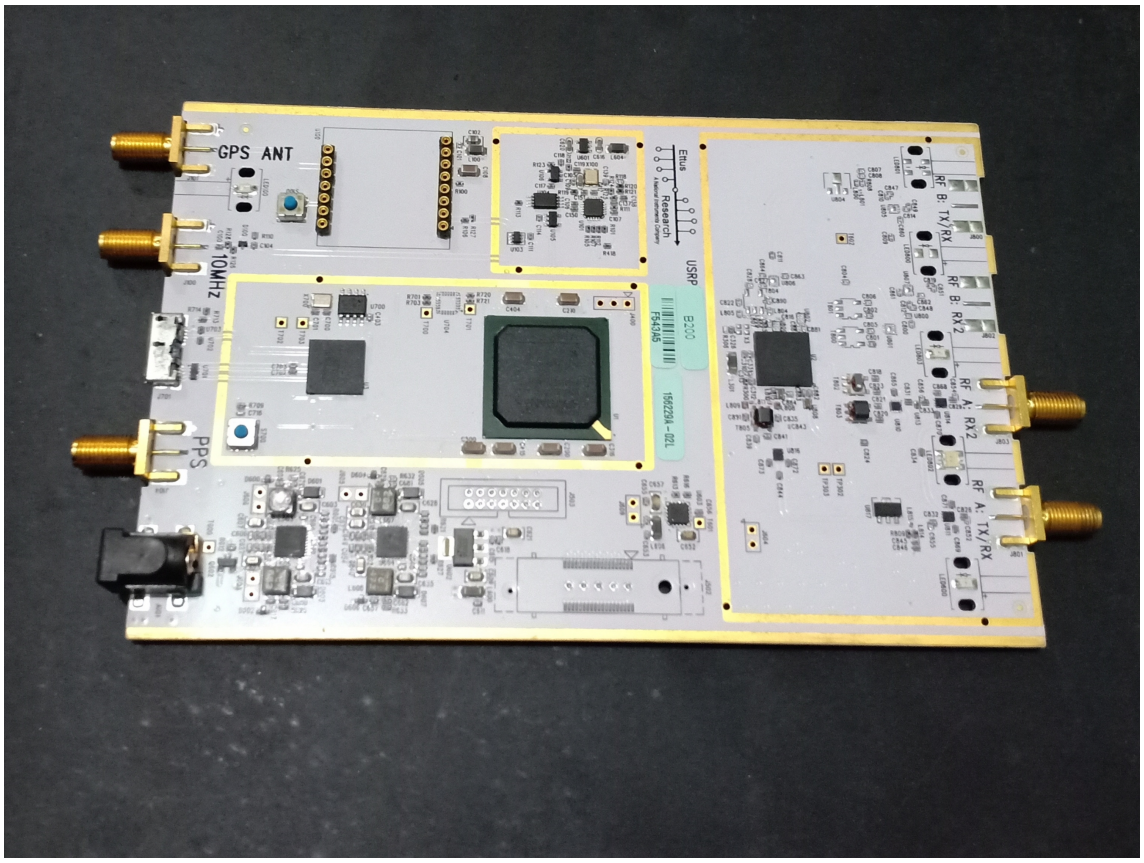

Figure 3.2: USRP B200 SDR Module

Here we will use USRP B200, which has a continuous frequency range of 70 MHz to 6 GHz, capable of full duplex operation. It has one receive and one transmit channel, capable of up to 56 MHz of instantaneous bandwidth. It interfaces with computer over a USB 3.0 connection.

3.3 OpenBTS

OpenBTS or Open Base Transceiver Station is an open source software implementation of the GSM architecture. It allows GSM enabled devices to be used as SIP endpoints in Voice over IP (VoIP) networks. It contains a Transceiver for the reception and transmission of the analog signals, SIP authentication server for user registration on the network, SMS queue for short message service, and asterisk as a PBX[18][19]. It is developed and maintained by Range Networks. The latest OpenBTS-GSM is version 5. There is also a UMTS 3GPP version which is in alpha.

3.4 Asterisk

Asterisk is an open source software operating as a PBX (Private Branch Exchange). It is used for voice calls between telephone endpoints[20]. Here asterisk enables OpenBTS to support audio calls between phones connected to the network.

3.5 Antenna

We have used 2 dual band antennae, VERT900 and VERT2450, both from USRP.

3.5.1 VERT-900

VERT900 is specified for 824–960 MHz and 1710–1990 MHz bands where most amateur radios and GSM networks operate. Although it works best within these ranges, in practice we are able to use it as a sub 2 GHz antenna in general. Figure 3.3 shows the VERT-900 antenna.

Figure 3.3: USRP VERT-900 dual-band antenna

3.5.2 VERT-2450

VERT2450 supports 2.4-2.5 GHz and 4.9-5.9 GHz bands. Wi-Fi, Bluetooth and many other services operate within these ranges. We use this antenna as the high frequency antenna. Figure 3.4 shows the VERT-2450.

Figure 3.4: USRP VERT-2450 dual-band antenna

3.6 GNU Radio

GNU radio is a free and open-source software development toolkit that provide signal processing blocks to implement software radios. The blocks are written in python and C++. The built-in blocks are well optimized and has been reviewed by the community, so using them are preferred rather than writing our own blocks. But writing the blocks are not difficult, although you do need good knowledge in signal processing and must know either python or C++. Users work with blocks on the GNU radio companion (GRC) for creating flow graphs, where the different blocks are connected one after the other to accomplish different tasks. The connections describe the dataflow between blocks.

3.7 UHD

UHD is provided by Ettus Research, currently a part of National Instruments[21]. It allows GNU radio to work with our B200. It installs some additional UHD blocks that enable operation using USRP devices. Like GNU radio UHD can also be either downloaded using the package manager or built from source. UHD version 003.007.002-release is installed.

Chapter 4

Methodology

4.1 Setting up the Environment

For setting up the environment, Firstly we connected the VERT900 and VERT2450 antenna with USRP B200's TX/RX port and RX2 port. Then We connected USRP B200 SDR with PC via USB 3.0. We completed the hardware setup. Figure 4.1 shows that all hardware components are connected successfully.

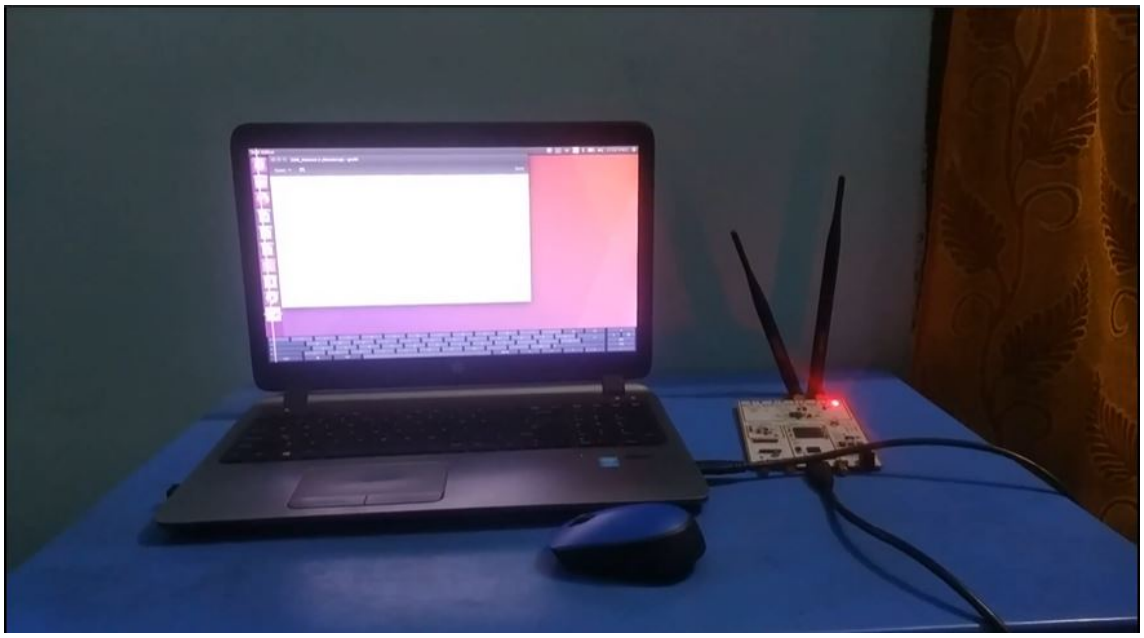

Figure 4.1: USRP B200 Connection with PC

Now we have to set up our software components. For setting up the software environment, we firstly have to install Ubuntu 16.04 LTS in our PC. OpenBTS works best with Ubuntu 16.04 LTS, despite the official build and install guides being written for Ubuntu 12.04 32-bit[13]. Installation process for Ubuntu can be easily found online, so not de-

scribed here.

Git version has to be at least newer than 1.8.2. Update commands for git follows,

$ sudo apt-get install software-properties-common python-software-properties
$ sudo add-apt-repository ppa:git-core/ppa
(press enter to continue)
$ sudo apt-get update
$ sudo apt-get install git

Next, we will clone the github repository by running,

$ git clone https://github.com/RangeNetworks/dev.git

This will save the downloaded files on a folder named 'dev' on /home, running

$ cd dev

we will move to the dev folder.
Now running the clone.sh script will download all the components automatically.

$ ./clone.sh

By default, the master branch is selected, and we will build from the master branch, so we don't need to use the switchto.sh script for selecting the desired branch.
Now to install all the components and all the dependencies including GNUradio, UHD etc. we need to run the build.sh script along with the radio being used. Here we are using USRP B200,

We have to download a .py file as uhd_images_downloader.py in build folder before run the build command.
$ ./build.sh B200

This should compile everything needed for B200 on a new directory named, BUILDS/timestamp within the dev folder.

Now we will use dpkg to install the packages, it may complain about dependencies being missing. Run the second command underneath to resolve the dependencies.

$ sudo dpkg -i BUILDS/timestamp/*.deb
$ sudo apt-get -f install

This will ask for confirmation to change some things, we selected yes for all prompts.

4.2 GSM and GPRS Configuration in OpenBTS

First, we need to confirm the B200 is connected properly with the PC, we can run the following command to make sure the device is properly connected and also to upload the image and FPGA image to the SDR,

$ uhd_usrp_probe

Next, we need to start up the network. Running OpenBTS automatically runs an instance of the transceiver. And data is then transferred between OpenBTS and the transceiver using a UPD (User Datagram Protocol) socket.

$ cd /OpenBTS

$ sudo ./OpenBTS

Next we check the current band and ARFCN of our network. This command will be run on the OpenBTSCLI interface,

OpenBTS>config GSM.Radio

By default, Band is 900MHz and ARFCN is 51. So, downlink frequency 945.2MHz, and uplink frequency 900.2MHz.We can change the band using,

OpenBTS>config GSM.Radio.Band <desired-band>

And ARFCN by,

OpenBTS>config GSM.Radio.c0 <desired-channel>

Both of the command above are static, for them to take effect we need to restart OpenBTS.

After restarting, as we are using USRP hardware we need to change the value for GSM.Radio.RxGain to something between 1 and 10. Otherwise received signal will overdrive the demodulator.

OpenBTS>devconfig GSM.Radio.RxGain 10

This key is also static and requires a restart of OpenBTS to take effect.

Now we now use a handset to search for the newly created network. This confirms the downlink is functional.

Next we need to connect to the network using a handset. For this we need the IMSI (International Mobile Subscriber Entry) of our SIM. We now start the SIPAuthServe daemon,

$ sudo /OpenBTS/sipauthserve

Now we can click on the networks name on our handset and this will perform a LUR (Location Update Request) on the network, which will allow us to see the IMSI of the SIM trying to connect. After the handset shows error saying couldn't connect to network. After more than one times, it should be connected. For confirmation the connection, we run the following on the OpenBTS interface,

OpenBTS>tmsis

If AUTH is 1 that means the device is connected. If not then the device is not connected.

```
OpenBTS> tmsis
IMSI            TMSI IMEI            AUTH CREATED ACCESSED TMSI_ASSIGNED
470070730069224 -    862305030431650 1    55m     42s      0
470073550091670 -    867325032649330 1    48m     8m       0
```

Figure 4.2: Registered user with their IMSI and IMEI

In Figure 4.2 shows that devices are connected with network.
In order to manually add a subscriber to the network we can use nmcli.py NodeManager which is in dev/NodeManager,
$cd dev/NodeManager

$./nmcli.py sipauthserve subscribers create <name> <imsi> <msisdn>
Example: ./nmcli.py sipauthserve subscribers create "XXX" IMSI..... \12345678910
Now to start our communication over text messaging we need to run following command.
$ sudo /OpenBTS/smqueue

The messaging service is ON. Now we can sent message between connected devices. Figure 4.3 shows that message system is working properly..

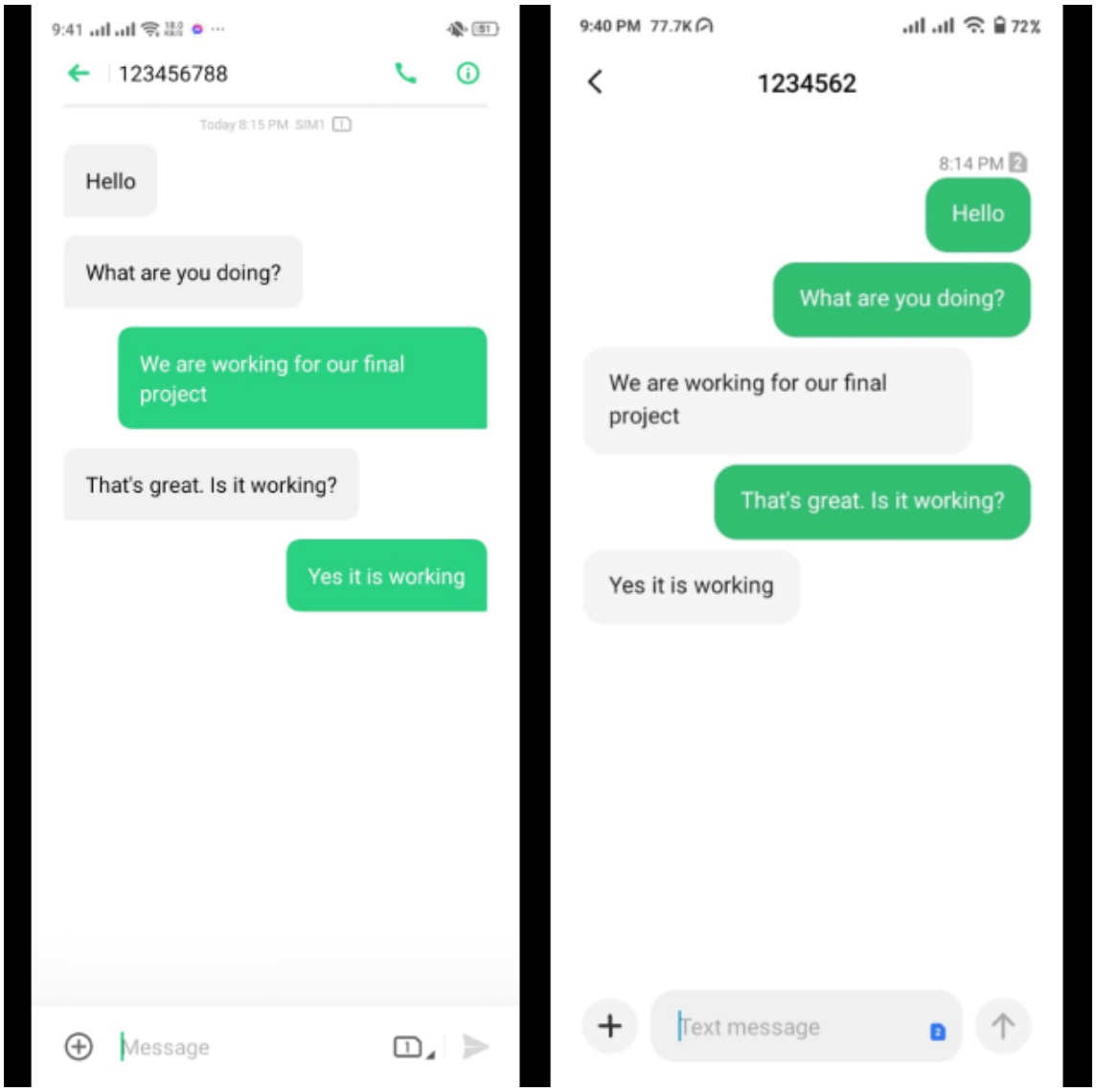

Figure 4.3: SMS Between Connected Devices

For voice call service the following command should be followed.
$ sudo asterisk -vvvv

The Voice service is ON. Now we can communicate through voice call. Figure 4.2 shows that the voice call system is working.

We can now check that our network is ready to use for text messaging and call system. To ensure that SMQUEUE is working properly we can text message to system by texting to 411. For voice call system asterisk we can dial 2600 which will echo our context.

GPRS system is already installed along with OpenBTS. Now we just have to enable this. By default GPRS system stays disable. To enable GPRS the following command

Figure 4.4: Calling Between Devices

must be run in terminal. This is a static parameter. Thus we need to restart OpenBTS after run the command.

OpenBTS>config GPRS.Enable 1

Now we can see a assigned list of gprs connected with the system. Users should have to turn on mobile data for using internet. If someone wants to see GPRS list then he or she should run the following command.

OpenBTS>gprs list

While your handset should perform a Location Update Request and join the network once its break. There may be additional step to make sure the GPRS service is usable and recognized.

OpenBTS > config GGSN.DNS.8.8.8.8 GGSN.DNS is static ; change take effect on restart.

The GPRS service is ON. Figure 4.2 shows that GPRS service is working properly. At top of the left side; the red circle show that the GPRS is connected.

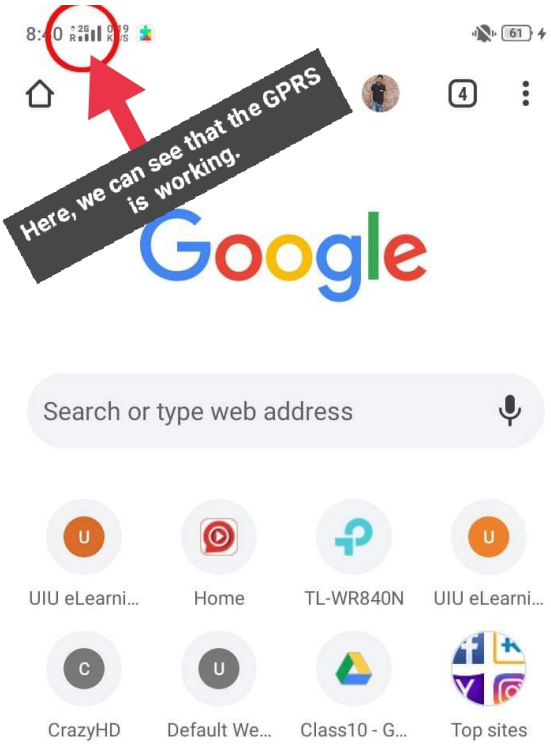

Figure 4.5: GPRS connection

Chapter 5

Standards Constraints

5.1 Compliance with the Standards

In this section we discuss the standards and impacts constraints.

5.1.1 Software Standard

SIP authentication - Adopted from Base Station to register and authenticate user namely sipauthserve.

SMS Queue - Standard called by smqueue to serve massaging service supported by OpenBTS.

PBX - By the help of Asterisk from the station PBX works for voice call service.

LUR - TO active location service we configured gprs along with DNS server.

5.1.2 Hardware Standard

Base Station - B200 SDR module is being used as physical base station.

OS - Configured the system in Ubuntu 16.04 LTS.

Antenna - VERT 900 was connected with the station as we supposed to implement GSM and GPRS and the antenna raughly work between 824 MHz to 1990 MHz.

5.1.3 Communication Standard

USB v3.0 (Universal Serial Bus)

5.1.4 Programming Languages

C/C++
Python

5.2 Impacts and Constraints

5.2.1 Economical Impact

If any private corporation want to communicate over a secured personal communication system inside the industry, they can purse our project. This is not any cheaper system in short term usages but the users can be beneficial when they use for long term. This is an one time investment system and this is not going to replace anyone's job.

5.2.2 Environmental Impact

Our project will use a very minimum energy to setup but as it is continuous it will use energy continuously. This will not create any harm to our environment or nature. But this can affect to our ecosystem as this will change regular communication system.

5.2.3 Health and Safety impact

This system will secure users communication as there will be no outside users who could use their system for communication. The owner will maintain the system after setup. This will not hard to any user health condition.

5.2.4 Manufacture ability

Now we are trying to build the project by configuring GSM architecture. If we can success, this can be build this in large number. This is not depend on upcoming technological advances.

Chapter 6

Conclusion

The text and Voice over communication as well as GPRS list are working properly, thus we can claim that our GSM and GPRS system are implemented successfully. Although in theoretically the average speed of GPRS is 56 to 114 kbps, the practical the speed is around 35 kbps. So for better communication through GPRS, we need to extend our system up−to 3G and we are planning for that. To be more specified in our system, there are a little bit delay while communicating via SMS or Voice call. In the end of that conclusion we can hope that we are looking forward to UMTS(3G).

References

[1] Joseph Mitola. Software radios: Survey, critical evaluation and future directions. *IEEE Aerospace and Electronic Systems Magazine*, 8(4):25–36, 1993.

[2] R. H. Hosking, Software Defined Radio Handbook, vol. 7. 2016.

[3] Guifen Gu and Guili Peng. The survey of gsm wireless communication system. In *2010 international conference on computer and information application*, pages 121–124. IEEE, 2010.

[4] Moe Rahnema. Overview of the gsm system and protocol architecture. *IEEE Communications magazine*, 31(4):92–100, 1993.

[5] John Scourias. Overview of the global system for mobile communications. *University of Waterloo*, 4, 1995.

[6] electronics notes, "What is Frequency Modulation — Electronics Notes." [Online]. https://www.electronics-notes.com/articles/radio/modulation/frequency-modulation-fm.php. Accessed: 2019-11-28.

[7] Electronics Notes, "Edwin Armstrong — FM Development — Electronics Notes." [Online]. https://www.electronics- notes.com/articles/history/pioneers/edwin-armstrong-fm-radio.php. Accessed: 2019-11-28.

[8] "What is a Spectrum Analyzer — RF Spectrum Analyzer — Electronics Notes." [Online]. https://www.electronics-notes.com/articles/test- methods/spectrum-analyzer/spectrum-analyser-overview.php. Accessed: 2019-11-28.

[9] BTRC, "Mobile Phone Subscribers in Bangladesh October, 2019—BRTC." [Online]. http://www.btrc.gov.bd/content/mobile-phone-subscribers-bangladesh-.Accessed: 2020-03-20.

[10] BTRC, "Internet Subscribers in Bangladesh October, 2019— BTRC." [Online]. http://www.btrc.gov.bd/content/internet-subscribers-bangladesh-october-2019. Accessed: 2020-03-20.

[11] "Open source project builds mobile networks without big carriers — Network World." [Online]. https://www.networkworld.com/article/2226543/open-source-project-builds- mobile-networks-without-big-carriers.html. Accessed: 2019-11-29.

[12] "InfoconDB." [Online]. https://infocondb.org/con/hope/hope- x/community-owned-and-operated-cellular-networks-in-rural-mexico. Accessed: 2019-11-29.

[13] openbts.org,"Build-Install-Run-OpenBTS." [Online]. http://openbts.org/w/index.php? Accessed: 2019-11-25.

[14] Friedrich K Jondral. Software-defined radio: basics and evolution to cognitive radio. *EURASIP journal on wireless communications and networking*, 2005(3):275–283, 2005.

[15] Kinjal Aggrawal and Khyati Vachhani. Reconfigurable cellular gsm network using usrp b200 and openbts for disaster-hit regions. In *2017 IEEE 13th Malaysia International Conference on Communications (MICC)*, pages 141–146. IEEE, 2017.

[16] Kunal Sankhe, Chandan Pradhan, Sumit Kumar, and Garimella Rama Murthy. Cost effective restoration of wireless connectivity in disaster hit areas using openbts. In *2014 Annual IEEE India Conference (INDICON)*, pages 1–6. IEEE, 2014.

[17] Dereje Mechal Molla, Hakim Badis, Alemayehu Addisu Desta, Laurent George, and Marion Berbineau. Sdr-based reliable and resilient wireless network for disaster rescue operations. In *2019 International Conference on Information and Communication Technologies for Disaster Management (ICT-DM)*, pages 1–7. IEEE, 2019.

[18] D. a. burgess and h. s. samra, the open bts project, 2008.

[19] M. iedema and h. samra, getting started with openbts. 2014.

[20] "C. Rhodes, M. Spencer, and M. Allison, The Asterisk Handbook Version 2, vol. 2. 2003.

[21] "Ettus-Research,"UHD-Ettus-Knowledge-Base."[Online]. https://kb.ettus.com/UHD. Accessed: 2019-11-28.