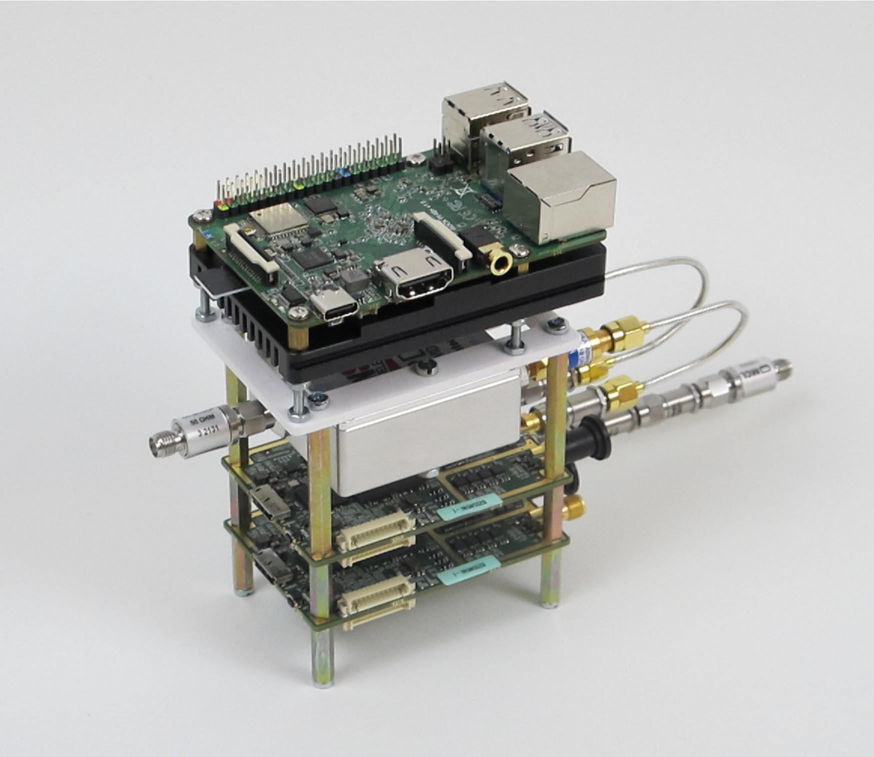


Testbed Implementation for Functional Verification of a Cellular Modem

Master's Thesis in Embedded Electronic System Design

JOHANNES LAMPELA
LUDVIG SVENSSON

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden 2022

Master's Thesis 2022

Testbed Implementation for Functional Verification of a Cellular Modem

JOHANNES LAMPELA
LUDVIG SVENSSON

UNIVERSITY OF
GOTHENBURG

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Computer Science and Engineering
Chalmers University of Technology
University of Gothenburg
Gothenburg, Sweden 2022

Testbed Implementation for Functional Verification of a Cellular Modem
JOHANNES LAMPELA
LUDVIG SVENSSON

Supervisor: Arne Linde, Department of Computer Science and Engineering
Assistant Supervisor: Erik Börjeson, Department of Computer Science and Engineering
Company Advisor: Thom Zakariassen, Host Mobility at SETEK groups AB
Examiner: Per Larsson-Edefors, Department of Computer Science and Engineering

Master's Thesis 2022
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Picture of the implemented testbed.

Typeset in LATEX
Gothenburg, Sweden 2022

Testbed Implementation for Functional Verification of a Cellular Modem
JOHANNES LAMPELA
LUDVIG SVENSSON
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg

Abstract

Equipment for testing the 4G and GPS functionality in a cellular modem can be expensive for a small company. This master's thesis investigates how a simple testbed can be implemented in an affordable way. The investigation concluded that a software defined radio together with open-source software could be used for a low-cost testbed implementation. The proposed solution was implemented and the testing of the system showed that it indeed can be used for functional verification of 4G as the testing is quick, stable, and successfully simulates a realistic scenario. The implemented GPS test is stable and can successfully be used for functional verification, but cannot simulate a realistic worst-case scenario in terms of signal power. The main conclusion is that a software defined radio indeed can be used for simple testing of LTE and GPS functionality in a cellular modem.

Keywords: Testbed, SDR, LTE, GPS, srsRAN, GPS-SDR-SIM

Acknowledgements

We would like to thank our supervisors Arne Linde and Erik Börjeson as well as our examiner Per Larsson-Edefors for all the support and mentoring during the project. We would also like to thank our company advisor Thom Zakariassen and Host Mobility in Setek group AB for the opportunity to perform our thesis work there and the help they have provided in order for us to succeed in doing so. Lastly, we would like to thank our family and friends for supporting us.

Johannes Lampela and Ludvig Svensson, Gothenburg, March 2022

Contents

Abbreviations

Abbrevation	Declaration
3GPP	Third group partnership project
CP	Cyclic prefix
CPU	Central processing unit
C/A	Coarse acquisition
DUT	Device-under-test
eNodeB	Evolved Node B
EPC	Evolved packet core
E-UTRAN	Evolved UMTS terrestrial radio access network
FPGA	Field-programmable gate arrays
GNSS	Global navigation satellite systems
GPP	General-purpose processor
GPS	Global positioning system
HSS	Home subscribe server
IMSI	International mobile subscriber identification number
IP	Internet protocol
IMSI	International mobile subscriber identity
Ki	Key identifier
LTE	Long-term evolution
MCC	Mobile country code
MME	Mobility management entity
MNC	Mobile network code
OFDMA	Orthogonal frequency-division multiple access
OP	Operator code
OPc	Cyphered operator code
PAPR	Peak-to-average power ratio
PGW	Packet gateway
PSS	Primary synchronization signal
RB	Resource block
RF	Radio frequency
SBC	Single board computer
SC-FDMA	Single carrier frequency
SDR	Software defined radio
SGW	Serving gateway
SIM	Subscriber identity module
srsRAN	Software radio systems radio access network
SSS	Secondary synchronization signal
UE	User equipment
UMTS	Evolved universal mobile telecommunications system

1 Introduction

A cellular modem is a key component in any device that needs to be connected to the Internet or other devices from a remote location. It is used in common equipment such as cellphones, network adapters, 4G routers and other smart devices. Even though the modem is common, it is complex, which makes it difficult to verify full functionality of the component. Therefore, testing equipment for function verification is usually expensive, advanced, and may require a skilled operator [1].

Some companies offer purchasable or rentable testing equipment at huge costs in the range of hundreds of thousands SEK [2]. However, this equipment focuses on product development as well as production testing. The extensive functions of this equipment are not all necessary for an in-production function test.

In this thesis, we investigate how a simpler and more affordable test system can be implemented. Such a system would allow smaller companies, which might not afford the expensive equipment, to still have a reliable test process in the production of their devices. Such a test procedure would reduce the risks of customers receiving faulty devices and reduce the risk of functional devices being discarded due to a failed, unreliable test.

1.1 Aim and Research Questions

The aim of this project is to investigate an implementation solution of an affordable testbed for a built-in cellular modem, suitable for a production environment. The testing solution shall primarily focus on verifying the functionality of long-term evolution (LTE) communication and also verify the functionality of a global navigation satellite systems (GNSS) receiver.

The goal is that the resulting solution from the investigation should be implemented, tested and verified to be reliable enough to be used in production of devices with a built-in cellular modem.

The following research questions are used as the core of the investigation:

- Is a field-programmable gate array (FPGA) with a radio frequency (RF) peripheral or a software defined radio (SDR) with a general-purpose processor (GPP) best suited as a platform for the implementation of the test system in the terms of cost, reliability, and flexibility?
- How can academic research on LTE implementations on the chosen platform be applied for this industrial application?

- How can a test system be implemented to verify the functionality of a built-in cellular modem without the presence of a public downlink signal?
- How does a hardware implementation on the chosen platform meet the test requirements that we will identify, and is it suitable for use in a production environment?

1.2 Limitations

To be able to estimate the time required in the implementation process of the system, the investigation will be limited to platforms and development tools in which the developers have previous knowledge about. Therefore, the platform selection will be limited to FPGA and GPP.

Due to the current pandemic, long delivery times of hardware components can become a problem. Therefore, hardware choices will be affected by the availability of the different options and not only by the technical aspects such as performance.

Another limitation is the price of the platform. There is a limited budget dedicated to the project, so hardware components that cost more than 20,000 SEK will not be investigated.

One more potential limitation is the possibility of future development and maintenance of the test system. If a suggested solution requires paid licensed tools and/or special skills to allow for future development or maintenance, it will affect the choice of solution.

The final limitation is that the test system is designed to interact with the modem through the antenna connections and through a serial connection. This limits the possibility to, for example, verify the connection of all solder joints connecting the modem to the device. The serial connection with the modem will be designed to function with the modem used in this project. Additional work might be needed for compatibility with other brands of modems.

1.3 Thesis Outline

The structure of this thesis is to first, in Chapter 2, explain the theory about the relevant concepts. Chapter 3 presents the testbed investigation research and the resulting implementation suggestion. Chapters 4 and 5 explain the implementation process and the results of the performed tests on the testbed. Chapter 6 covers the discussion about design choices and testing results. Finally, Chapter 7 presents the conclusions of this work.

2

Theory

For the reader to fully understand the investigation, implementation, and discussion in this report, some knowledge about the concepts is needed. This chapter presents the theory of the technologies used in this report.

2.1 Long-Term Evolution Communication

Third Generation Partnership LTE (3GGP LTE) or simply LTE, is a communication technology known as 4G. LTE provides fast data rates in both the downlink and uplink with up to 300 Mbit/s and 75 Mbit/s respectively. It also increases the bandwidth in the radio channel up to 20 MHz [3]. LTE provides a fast roundtrip time, flexibility in bandwidth and frequencies, high spectral efficiency, and high peak data rates, which all are requirements for the access network [3].

2.1.1 System Architecture Overview

The architecture of the LTE network, called system architecture evolution (SAE) [4], consists of the evolved universal mobile telecommunications system (UMTS) terrestrial radio access network (E-UTRAN) and the evolved packet core (EPC). The E-UTRAN and EPC are called evolved packet system (EPS) and the architecture is shown in Fig. 2.1. In the E-UTRAN there exists only one network element which is the base station Evolved Node B (eNodeB), while in the EPC there exists many other [5].

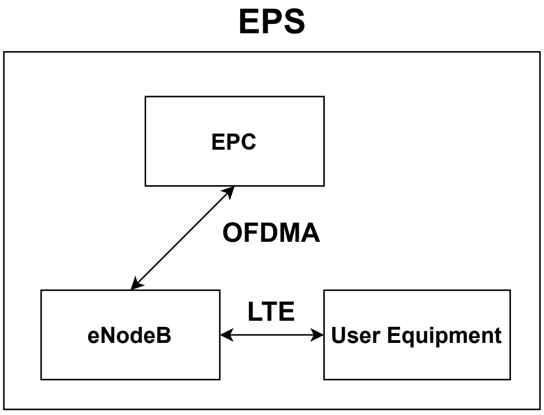


Figure 2.1: Picture that shows the EPS architecture, adapted from [3].

Evolved Packet Core

The LTE system's core network is the EPC which consists of four components: The Home Subscribe Server (HSS), Serving Gateway (SGW), Packet Data Network Gateway (PGW), and the Mobility Management Entity (MME) [6]. The internal structure can be observed in Fig. 2.2.

In order for the E-UTRAN to be granted access to the EPC, it needs to pass through the MME, which manages the security needed for E-UTRAN access and also handles the control plane. The MME also connects to the HSS to retrieve subscriber and user information for setting up call sessions, authentication, authorization, and the necessary support functions for completing those tasks [6].

The SGW handles the communication and routing of IP packets to the external networks from the User Equipment (UE) [6] such as cellular modems which can be seen in Fig. 2.2. The SGW is connected logically to the PGW which performs controls of IP addresses, IP prefixes, policies, charging and handles communication to and from the packed data networks [6].

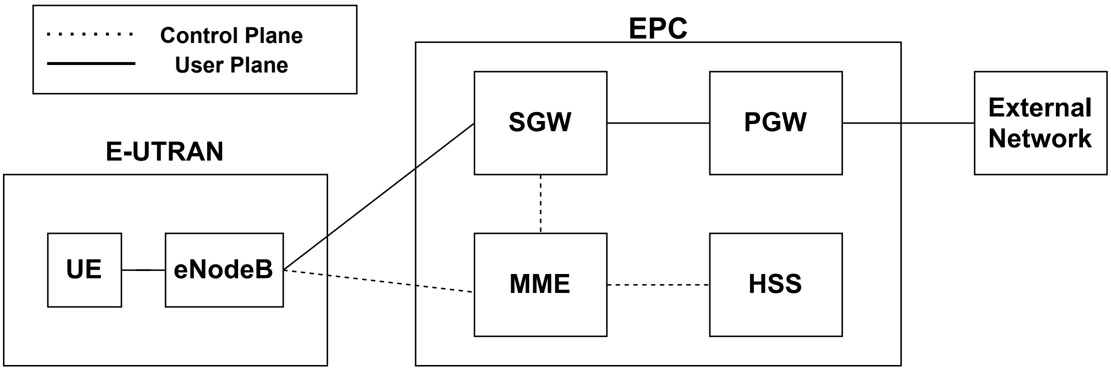


Figure 2.2: Picture over E-UTRAN and the EPC's internal architectures, adapted from [6].

Evolved Node B

An eNodeB is a base station in LTE E-UTRAN which connects to the UEs and also to the rest of the base stations spanning up the LTE access network [3]. The connection setup and handover time for communication with LTE is faster than for that of the previous generation, UTRAN, as some of the intelligence is distributed to the base stations [3]. In UTRAN systems, the radio network controller had to do all the work [7]. This improvement allows for a faster downlink communication in the network [3].

2.1.2 Data Transmission

LTE transmits data over the physical channel by using two different modulation techniques in the downlink and uplink [8].

Downlink

The LTE downlink transmits data by using a modulation technique called orthogonal frequency-division multiple access (OFDMA) [8]. OFDMA is a modulation scheme designed for multiple users and is a version of orthogonal frequency division multiplexing [9]. With OFDMA, problems such as multipathing and intersymbol interference (ISI) are reduced [10]. OFDMA handles ISI well because the bandwidth is broken up into subcarriers and information is sent in parallel over these subcarriers [10].

Providing protection against multipath spread delay is also important. Without protection ISI can be introduced as a consequence of symbols that overlap in the received signal [9]. However, OFDMA solves this problem by adding a guard band in-between the symbols, which is called the cyclic prefix (CP) [9].

LTE uses a constant subcarrier spacing of 15 kHz, independent of the selected bandwidth [8]. Twelve subcarriers span up 180 kHz in the frequency domain which correspond to 0.5 ms in the time domain. This is called a resource block (RB and it is the smallest available resource allocated in both the downlink and uplink. Using a higher bandwidth allows fore more RBs [8]. The whole RB is observable in Fig. 2.3.

Since the bandwidth is split into sub-carriers, the noise has a high dynamic range following the amplitude spikes power peak of overlapping carriers when in phase, which results in a high peak-to-average power ratio (PAPR) that needs to be controlled by power amplifiers [11].

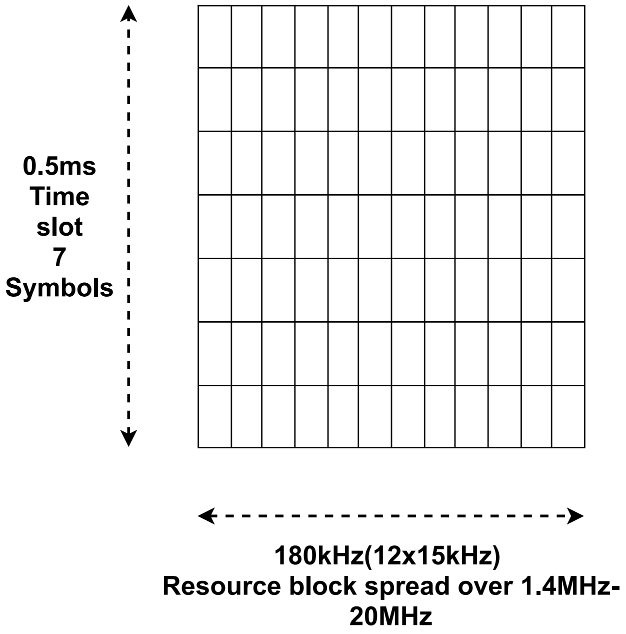


Figure 2.3: Illustration of the time and frequency grid for a resource block, adapted from [12].

.

Uplink

LTE does not use OFDMA in the uplink. Instead, it uses single-carrier frequency-division multiple access (SC-FDMA). The reason for this is to avoid the high PAPR that is generated from pure OFDM [8]. Reducing the PAPR is of great importance in the UEs as it determines the efficiency of the power amplifiers and also how much coverage the UEs will have [13].

Although the PAPR is reduced by using SC-FDMA. The performance is affected by the order of modulation used. Using a higher order in the modulation will reduce the PAPR advantage that SC-FDMA has [14].

2.1.3 UE and eNodeB Synchronization

In order for a random access (RA) procedure to be established between the UEs and eNodeBs. The UEs and the eNodeBs has to perform cell search and then synchronization with a primary and secondary synchronization signal. [15]. Initial cell search is performed by the UEs and refers to the search and synchronization of nearby base stations in order to set up an access in the downlink [15].

The base station periodically sends out primary synchronization signals (PSS) and secondary synchronization signals (SSS) which are detected by UEs in order to sync the time and frequency with each other. It is not possible for the UEs to be selected for transmission in the uplink if synchronization hasn't occurred. The PSS contains information about cell group ID while the SSS contains information about cyclic prefix (CP) length, cell-id, properties and frame timing [15].

From the PSS and SSS, system information can be downloaded to the UE, which are needed for the UE to set up a physical random access channel [15]. The cell search procedure can be seen in Fig. 2.4.

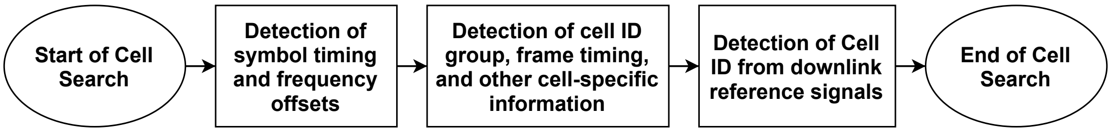

Start of Cell Search
Detection of symbol timing and frequency offsets
Detection of cell ID group, frame timing, and other cell-specific information
Detection of Cell ID from downlink reference signals
End of Cell Search

Figure 2.4: Procedure of the cell search, adapted from [16].

2.2 Global Navigation Satellite Systems

GNSS is a collective name of navigational satellite systems like GPS, Galileo, GLONASS and BeiDou which are developed by US, Europe, Russia, and China [17]. The systems consist of several satellites in orbit around the Earth on a distance of approximately 20,000 km [18].

GNSS uses the positioning, time and distance of several satellites in orbit to calculate the position of the receiver through trilateration that measures the distance between the satellites and the receiver [19]. The GPS system can provide synchronization services and positioning with a global coverage [18]. Due to the long distance between the satellite and the receiver on Earth, the received signal power is normally around -125 dBm [20].

The GPS system uses two RF-links, which are called L1 and L2. The carrier frequency of link L1 is 1575.42 MHz and the frequency of L2 is 1227.6 MHz. Both carriers are modulated as one or more bit-trains, using modulo-2 addition of the pseudo-random noise ranging code (PRN) and the system data of the downlink (NAV). The different PRN ranging codes that are sent are the Precision (P) code, the Y-code, and the Coarse/Acquisition (C/A) code [21].

2.3 Cellular Modem

A cellular modem is a component which can be installed in a system to add wireless connectivity. The key function of the modem is to add a 3G, 4G and potentially a 5G connection depending on the modem [22]. In addition to the cellular signals, modems can also include other wireless technologies, such as a GNSS engine and receiver for geopositioning [23]. A device with a cellular modem is called UE in the cellular network and has the required protocols implemented within the modem.

2.3.1 Subscriber Identity Module

Each cellular modem needs a subscriber identity module (SIM) to establish a connection with the cellular network. The SIM is a removable card which has a set of identification numbers and security keys, unique to each individual card. The international mobile subscriber identity (IMSI) is a 15-digit number which is the units' identification number in a network. Three of the digits is the mobile country code (MCC) which indicates which country the unit belongs to. Two other of the 15 digits is the mobile network code (MNC) which in turn indicates which mobile operator the unit belongs to. The key identifier (Ki) is a 128-bit secret encryption key for which the network uses to verify the legitimacy of the device. If the Ki does not match with what the network expects, it will reject the connection attempt [24].

In the same way that the Ki is used by the network to validate the device, the device uses operator code (OP) or cyphered operator code (OPc) to validate the network. These are also secret 128-bit keys similar to the Ki [25].

2.4 Software Defined Radio (SDR)

SDR is a dominant standard today in the industry [26]. It is a component which consists of a radio communication system that performs modulation and demodulation of radio signals under software control [27]. This is done by using re-configurable digital electronics and a GPP as controller, which opens up the potential of running new radio protocols just by exchanging the software [27].

Signal properties, such as modulation technique, signal bandwidth, and the carrier frequency, are implemented in software reconfigurable components [28]. Analog components such as antennas, power amplifiers, pre-filters, and switches are fixed hardware components [29]. An example of an SDR implementation is shown in Fig. 2.5.

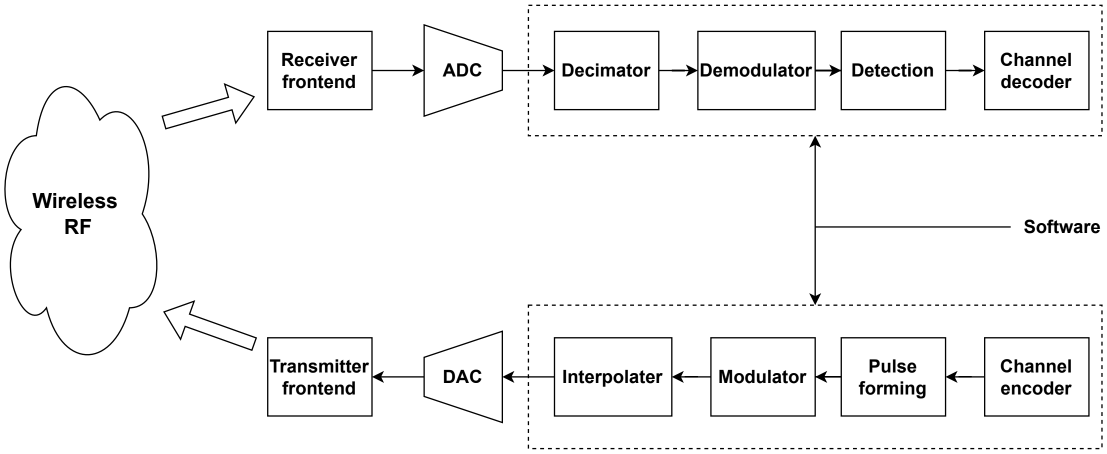


Figure 2.5: A block diagram of how an SDR is typically implemented, adapted from [30].

3

Testbed Investigation

To obtain an understanding on how the testbed can be implemented efficiently, inexpensively, as stable as possible, and with the possibility of further development, several solution paths were investigated. This chapter presents the investigated concepts, considers the advantages and disadvantages of them, and explains the design choices made from the investigation.

3.1 Test System Requirements

To decide the requirements of the testbed, the testing process needed to verify functionality of the modem was investigated. Additionally, requirements for being suitable for a production environment were specified.

To avoid any legal issues, it was decided that the testbed should only perform wired testing, since it is not legal to transmit signals with any significant amount of power in the licensed frequency bands [31].

Since the testbed is intended for testing during manufacturing of devices with a built-in modem and not for development, all functions of the modem do not need to be tested. A full function verification should have been done by the manufacturer of the modem prior to the installation in a system. This shifts the focus to verifying the data path between the modem and the antenna ports, as well as to verify that the modem can be used by the main processor in the device.

3.1.1 LTE Verification

To verify the functionality of the LTE communication, it is deemed to be enough to establish a connection between the modem and the eNodeB. Therefore, a functional downlink and uplink channel is required to complete the RA procedure. In addition to an established connection, the received signal strength in the uplink and the downlink should be checked. If a data path is damaged or has imperfections but still works, the fault can be detected by comparing the received signal strength to a reference value. The reference value depends on the wired RF path between the testbed and the device-under-test (DUT) as well as the gain values in the downlink and uplink channels.

For the verification to be correct, the transmitted signal from the testbed needs to be comparable to an official signal in terms of signal power.

To verify that the DUT signal detection is an actual signal and not a malfunction in the device, the network name can be verified to match the network of the testbed. This verification needs a serial communication with the modem.

3.1.2 GPS Verification

To verify the functionality of the GPS, a position needs to be established by the modem and the transmitted signal needs to have a realistic signal power. For additional verification, the IDs of the reported satellites in view can be verified to match the satellites in the simulated signal. This verification needs a serial communication to the modem, as the GPS only consists of a downlink.

3.1.3 Production Environment Requirements

To be suitable in a production environment, the testing needs to be done as time-efficiently as possible since a bottleneck could limit the entire production chain. However, it is critical that no faulty devices pass through the function verification and are shipped to the customer.

The testbed needs to be quick and simple to set up to be viable in a production environment since time is of the essence. Any user error causing a delay would be highly undesirable and needs to be avoided.

3.2 Platform Investigation

To investigate how the core of the test system with the signal generation could be implemented, a couple of possible solutions were looked into. The solution suggestions branched out from two concepts: implementing the signal generation on an FPGA evaluation board with an RF frontend or using an SDR together with a GPP. The considered aspects are hardware cost, development time/complexity and the possibility of further updates.

3.2.1 SDR Based LTE Implementation

By using a GPP-based implementation with an SDR, the full LTE protocol stack could be implemented to do the communication correctly. This would be a stable solution since the LTE communication between the DUT and the testbed would be performed as intended. It is not necessary to code the protocol stack implementation from scratch as there are multiple open-source projects which have already implemented it, such as *srsRAN* (formerly *srsLTE*), *OpenAirInterface* (*OAI*) and *OpenLTE*. However, if there is a need for custom functions or the available software is not satisfactory, one could build a custom software.

Complete LTE Protocol Stack Implementation

All the open-source projects *srsRAN*, *OAI* and *OpenLTE* are software implementations of the 3GPP LTE stack with EPC, eNodeB and UE [32, 33, 34] , with additional support for commercial off-the-shelf (COTS) UE devices [35, 36, 34]. The software can be built and executed on a computer running a Linux-based operating system and use an SDR as an RF frontend. The SDRs which can be used in these solutions varies in performance and range in price from approximately 2.000 SEK to 15.000 SEK. These are covered in detail in Section 3.4.

The *srsRAN* and *OpenLTE* software are published with *GNU Affero General Public License, Version 3* [37, 38] which allows for commercial use. This is not the case for *OAI* since the license only allows for personal or academic usage [39].

The documentation of the *srsRAN* software claims that both the EPC and the eNodeB are able to run in parallel on a *Raspberry Pi 4* with a low performance configuration [40]. This is also possible with *OpenLTE* since it is the simplest of the three [34]. *OpenLTE* is claimed to have the EPC included in the eNodeB software but is not officially confirmed. The ability to run both the EPC and the eNodeB on the same machine is not recommended with *OAI* in the later versions of the software [36]. Previous research indicates that *srsRAN* has shorter connection times than *OAI* [41].

Amarisoft LTE 100 is another software implementation of the 3GPP LTE stack. This software is the most advanced of the currently available software LTE implementations [42], but is not open-source and requires a paid license.

Custom LTE Implementation

Another possible way to implement the test would be to use *MATLAB* or *GNU Radio* instead of the previously mentioned solutions for the signal generation. This allows for a fully customizable implementation where the developer designs the complete system.

The *LTE Toolbox* in *MATLAB* has extensive documentation with several examples for which could be helpful if chosen as an implementation method. However, this would require a paid license of *MATLAB* and the *LTE Toolbox* in order for further development.

GNU Radio, which does not require a paid license, can also be used to implement LTE communication. This is shown by the open-source project *OpenLTE*, which is built with it. Although there is a large community working with *GNU Radio* and sharing their work, there is limited official documentation, example designs and plug-ins.

Both these solutions could be used for an LTE implementation, but would require a significant amount of implementation time compared to using one of the pre-built stack implementation.

3.2.2 FPGA-Based LTE Implementation

If the signal generation would be implemented on an FPGA development board with an RF frontend, the communication could possibly be done without the protocol stack by implementing a hard-coded RA procedure. This might however require long development time since it would require a custom implementation. The implementation could be done with *MATLAB* using the *LTE Toolbox* or with *Xilinx ISE Design Suite* [43].

Another possibility is to use the FPGA board and an RF frontend as an SDR unit together with a GPP controller like in [44]. This would however not be a cheaper alternative than using a pre-built SDR as the RF frontend *FMCOMMS4* [45] together with a *Xilinx ZedBoard* [46] cost approximately 10,000 SEK. It is also not supported by the LTE stack implementations described in Section 3.2.1 and would therefore require a custom implementation.

Both *MATLAB* and the tools from *Xilinx* require paid licenses [47, 48] and previous knowledge to be used efficiently. Therefore, both these solution paths are deemed less flexible for further development than previously mentioned solutions.

3.2.3 GPS Simulator Investigation

One available solution for the GPS test-implementation is to use an SDR with *GPS-SDR-SIM*, which is an open-source implementation of a GPS simulator. It uses real ephemeris data [49] to generate a GPS baseband data stream, which can be transmitted as RF using an SDR [50]. According to [51, 52], the software can generate a signal, transmit it with an SDR and have a GPS receiver to acquire a stable position from it.

Another possible solution, if using an *Ettus USRP* SDR, would be to use the GPS Simulation function in *LabVIEW*. The software, created by *National Instruments*, does however require a paid license to be used [53].

The GPS simulator could also be custom-built on an FPGA with *MATLAB/Simulink* like in [54]. This requires a significant development time and a paid license.

3.2.4 Platform Choice

With the gathered information of the different solutions, the SDR implementation with *srsRAN* was chosen to be the most suitable LTE implementation for this project and the *GPS-SDR-SIM* for the GPS implementation. The SDR was chosen as the most promising platform due to its short implementation time, relatively low cost, and it is deemed to be stable enough for this project. The reason why it was deemed stable enough is that there are several papers, such as [41, 55, 56], which has previously used SDRs for LTE implementations. Another benefit is the flexibility of SDRs which allows for implementation of both the LTE verification and GPS verification on the same hardware.

OpenLTE was not chosen as it has the least documentation and does not appear as commonly as the *srsRAN* or *OAI* in previous research. This leaves an uncertainty of how well the software performs.

The reason for why *srsRAN* was chosen above *OAI* was because, as already mentioned, has shorter connection times. The documentation claims that it can run both the EPC and eNodeB on a single *Raspberry Pi 4* [40] which keeps the total price of the implementation to a minimum.

A final reason is that the license of *srsRAN* allows for commercial use, while *OAI* has a stricter license. *OAI* would be a viable option as well and could be used as a backup if the *srsRAN* implementation would fail.

3.3 GPP Investigation

The chosen solution for further investigation, using an SDR along with a GPP, require the GPP to run a Linux-based operating system and recommends *Ubuntu* [57]. The documentation for *srsRAN* claims that a *Raspberry Pi 4 2GB* running *Ubuntu Server 20.04 LTS* is capable to run the *srsEPC* and the *srsENB* software with an *USRP* SDR [58]. The example setup is configured to operate in band 3 (1800 MHz) with a bandwidth of 3 MHz, which corresponds to 15 RBs. Since the data transfer rate is not important in this implementation, as described in Section 3.1.1, this bandwidth would be acceptable. This indicates that a computer with equal or better performance than a *Raspberry Pi 4* should be sufficient for the testbed implemented in this project.

Single-board computers (SBC) such as the *Raspberry Pi* are generally inexpensive and would keep the hardware costs low. An additional benefit would be the small form factor, allowing for a physically smaller implementation of the testbed compared to a full size PC, laptop or mini computer.

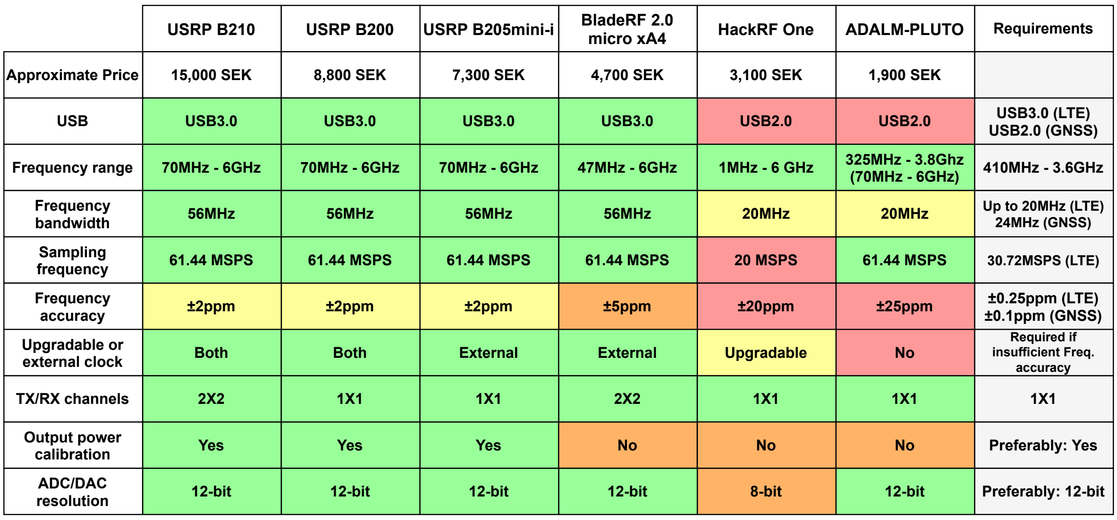
With this information, a *Rock Pi 4B 4GB* was chosen for the system implementation as it has the same SCB form factor as a *Raspberry Pi 4* but with additional computational power [59] and both devices costs approximately 1000 SEK.

3.4 SDR Investigation

Since there are several SDRs on the market with varying prices and specifications, the first selection on which SDRs to investigate was the devices supported by *srsRAN*. This is determined by which drivers it supports. Along with the specific SDR drivers *UHD* and *BladeRF*, *srsRAN* also supports *SoapySDR* which is a general driver for SDRs from several brands [60]. From this list, only the devices with the possibility to transmit and that support the required frequency range, were further investigated. Furthermore, only devices that cost less than 20,000 SEK are of interest, in order to keep the total cost low enough to be reasonable for a small company.

One SDR which was excluded from the comparison list was the *LimeSDR*. It has specifications compatible in many aspects for a good price. The reason for this is that it is currently unavailable due to the component shortage, partly caused by the current pandemic.

The specifications of the investigated SDR units can be seen in Fig. 3.1. The cells marked green in the figure are the ones that are preferred or fulfill the set requirements, while the red cells are deemed to be insufficient. The yellow cells are considered to be good-enough options, while the orange cells could work but may risk being insufficient.



	USRP B210	USRP B200	USRP B205mini-i	BladeRF 2.0 micro xA4	HackRF One	ADALM-PLUTO	Requirements
Approximate Price	15,000 SEK	8,800 SEK	7,300 SEK	4,700 SEK	3,100 SEK	1,900 SEK	
USB	USB3.0	USB3.0	USB3.0	USB3.0	USB2.0	USB2.0	USB3.0 (LTE) USB2.0 (GNSS)
Frequency range	70MHz - 6GHz	70MHz - 6GHz	70MHz - 6GHz	47MHz - 6GHz	1MHz - 6 GHz	325MHz - 3.8Ghz (70MHz - 6GHz)	410MHz - 3.6GHz
Frequency bandwidth	56MHz	56MHz	56MHz	56MHz	20MHz	20MHz	Up to 20MHz (LTE) 24MHz (GNSS)
Sampling frequency	61.44 MSPS	61.44 MSPS	61.44 MSPS	61.44 MSPS	20 MSPS	61.44 MSPS	30.72MSPS (LTE)
Frequency accuracy	±2ppm	±2ppm	±2ppm	±5ppm	±20ppm	±25ppm	±0.25ppm (LTE) ±0.1ppm (GNSS)
Upgradable or external clock	Both	Both	External	External	Upgradable	No	Required if insufficient Freq. accuracy
TX/RX channels	2X2	1X1	1X1	2X2	1X1	1X1	1X1
Output power calibration	Yes	Yes	Yes	No	No	No	Preferably: Yes
ADC/DAC resolution	12-bit	12-bit	12-bit	12-bit	8-bit	12-bit	Preferably: 12-bit

Figure 3.1: Investigated SDRs with specifications.

3.4.1 USB

The USB connection of the units is either USB 2.0 or 3.0 where the latter can transfer up to 10 times more data depending on the internal bus [61]. In the documentation for *srsRAN* [35] and in the user forums [62] it is expressed that USB 2.0 is not sufficient for the eNodeB implementation. This excludes the *HackRF One* and the *ADALM Pluto* from the selection of viable SDRs.

3.4.2 Frequency Bandwidth

All the *USRP* devices and the *BladeRF* can handle the required bandwidth, which is up to 20MHz for LTE and 24MHz for GPS (L1 C/A) [63]. The LTE implementation will most likely not be implemented with 20MHz, but instead a narrower bandwidth. This is because a higher bandwidth requires a higher central processing unit (CPU) usage in the eNodeB and the resulting higher data rates are not of interest in this implementation as described in Section 3.1.1.

3.4.3 Output Power Calibration

The possibility to calibrate the output power of the SDR is a feature which rarely is of interest for the common use of SDRs. However, in this project, it is desired since an accurate signal would result in a more accurate reference value described in Section 3.1.1. This is the value which the reported signal strength in the UE can be compared to during the test.

With the latest driver update, *UHD 4.0*, the *USRP* SDRs support input and output power calibration with reference power levels [64]. The SDR uses the calibration data and compensates for it automatically.

If output power calibration is not available, the calibration data would have to be manually created using a calibrated measuring instrument. The software in the testbed would then need to read the data and compensate for it. This would be acceptable, but would make the calibration process more inconvenient.

3.4.4 Frequency Stability

Having a decent frequency stability is an important factor when operating base stations for LTE. Being outside the recommended specifications for the minimum required stability is bound to fail and introduce all kinds of error patterns in the system according to [65]. However, other sources indicate that LTE communication is possible with a frequency accuracy outside the specified limit [41, 55, 56].

The recommended stability for LTE lies between 50 and 250 parts per billion (ppb) [66]. This means that all the SDRs investigated have crystal oscillators which are outside the recommendation. However, some SDRs have the possibility to upgrade their frequency stability by using an external reference clock instead. Depending on the oscillator in the reference clock, this option can significantly increase the frequency stability. A reference clock with an oven controlled crystal oscillator (OCXO) has a frequency stability in the range of parts per billion instead of parts per million (ppm), which is the range for temperature compensated crystal oscillators (TCXO) [67].

3.4.5 SDR Choice

From the investigated SDR devices, only the *USRP* SDRs and the *BladeRF* have sufficient performance to be a feasible choice for this implementation. Neither of the two meets the frequency stability requirement, but they both support the use of an external reference clock.

The chosen SDR was the *USRP B205mini-i* since it is the cheapest SDR that meets all the requirements except for the frequency stability for which an external clock can be used as a solution as mentioned in Section 3.4.4. It also supports output power calibration that is a desired feature. The final reason is that previous research [41, 55, 56] has shown that the more expensive *USRP B210*, which is equal in performance but includes two TX/RX channels, can successfully be used for LTE implementations with *srsRAN*.

The *USRP B200mini* could also be used, since the only difference to the *B205mini-i* is the FPGA used in the device [68]. Since the SDR will be used in the standard configuration, the additional FPGA resources is not needed in this implementation. The *B205mini-i* were available at a lower price than the *B200mini* which is the reason it was chosen instead.

3.5 RF Channel Investigation

As the modems in UE devices generally use combined antennas for the transmit and receive channels and the eNodeB has separated transmit and receive antennas, a solution on how to connect the devices in the LTE implementation was investigated. The RF channel for the GPS implementation was also investigated.

3.5.1 Combining LTE Uplink and Downlink channels

To combine the uplink and the downlink channels, a power combiner/divider can be used to merge the two channels into one. Another option is to use a duplexer which also includes a bandpass filter to suppress the downlink frequencies in the uplink channel and uplink frequencies in the downlink channel. A duplexer is specified to the frequencies of a single band. For this implementation, a power combiner/divider would be a better solution, since the use of a duplexer would limit the use to only one LTE band.

3.5.2 Channel Attenuation

Since both the LTE and the GPS are intended to be transmitted wirelessly, we can expect a significant power loss in the RF channel. This needs to be introduced in the wired channel in the form of attenuators. The minimum attenuation value required is decided by the transmitted power and the maximum safe input power limit of the components.

The chosen *USRP B205mini-i* SDR has a transmit power of approximately 10 dBm and a maximum safe input power of −15 dBm [69]. The *Quectel EG25-G* modem, which will be used as a DUT in the testing of the system, has a maximum safe input power of 35 dBm for the LTE antenna port and 15 dBm for the GNSS antenna port. The modem transmits LTE with a power of 23±2 dBm but GSM (2G) with a power of 33±2 dBm from the same antenna port [70]. Therefore, an attenuation value of at least 50 dB is required to not risk causing damage to the SDR.

An attenuation value of 60 dB is deemed suitable for the LTE test implementation as it results in a good margin to the 50 dB safety requirement. It also brings the received signal power down to a level to what a wireless LTE transmission would result in, which is less than −30 dBm [71]. Finally, it would result in an 120 dB attenuation between the transmit and receive ports of the SDR, which is considered a generous amount in a wireless setup [72].

For the GPS test implementation, additional attenuation is required since GPS signals are received with a power of −125 dBm. Therefore, 90 dB of attenuation is suitable.

3.5.3 DC-Component Blocker

It is common that the GNSS antenna port uses a bias tee to add a DC component in the channel. This is used to power the amplifier when using an active antenna, but it is not desired in a wired RF connection since the introduced power could damage the connected hardware. A DC blocker is therefore needed in the wired RF path to remove the DC-component.

3.6 Investigation Results

The testbed implementation solution that this investigation has resulted in is shown in Fig. 3.2. The test-system features two *USRP-B205mini-i* SDRs: One for the

LTE test with *srsRAN* and the other one for GPS test with *GPS-SDR-SIM*. The reason two SDRs are used is to run the LTE and GPS test in parallel to reduce the testing time. This is to make it as viable as possible for the production environment specified in Section 3.1.3.

The SDRs are connected to a *Rock Pi 4* running *Ubuntu Linux*. The SMA ports of the LTE test SDR are connected to 60 dB attenuators which are connected to a power divider/combiner with a DC blocker. The DC blocker is in turn connected with SMA RF-cables to the UE. The GPS test SDR are connected to the modem with 90 dB of attenuation and a DC blocker.

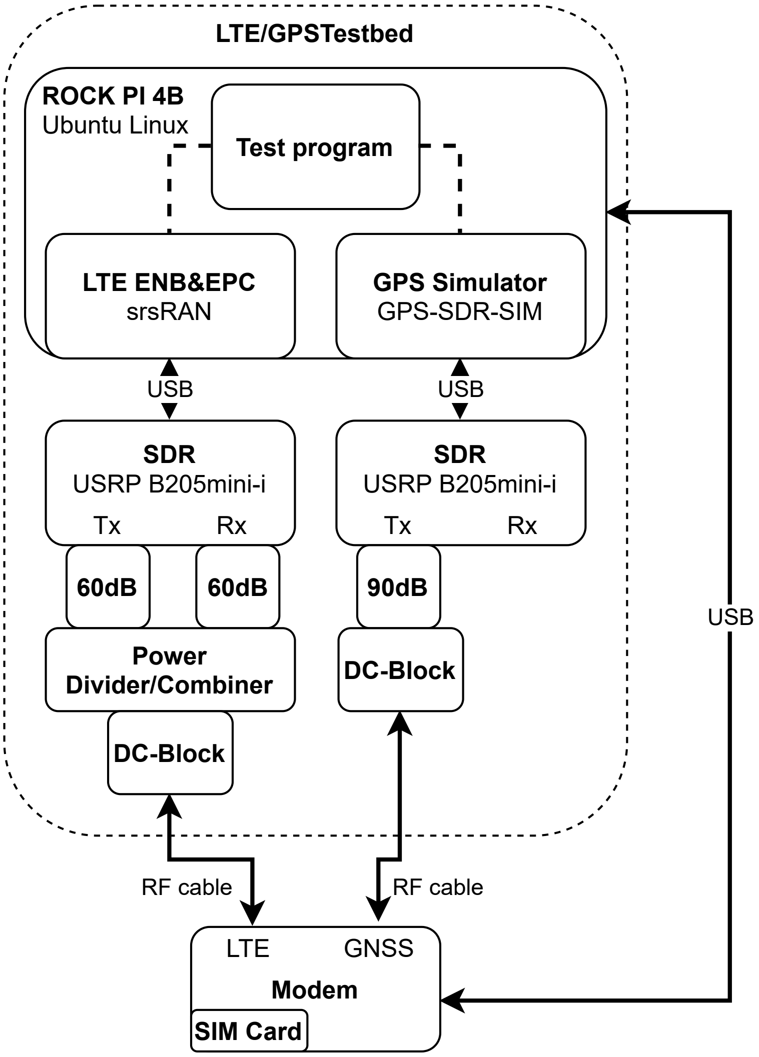


Figure 3.2: Testbed implementation overview.

3.7 Solution Feasibility

To test the feasibility of the chosen solution, an *ADALM Pluto* SDR was used as a first test implementation before investing in the chosen SDR. The reason this device was used despite it was previously deemed to be insufficient was because it was already available to the development team.

The host computer used for the initial tests was an *HP EliteBook 840 G3* running *Ubuntu 20.04*. The reason the testing was not done on the *Rock Pi 4* was because the use of a GUI would speed up the testing process but would also require extra resources which might be too much for the *Rock Pi 4*. *SoapySDR* and the other required drivers were installed, and the software *IIO Oscilloscope* [73] was used to verify the transmit and receive functionality of the SDR.

During the tests, the devices were connected with an RF cable and a 30dB attenuator in order to not transmit any signals to the environment.

3.7.1 srsRAN Test with Pluto SDR

The *srsRAN* project and the required drivers were installed and configured as described in Section 4.1.2. The EPC was started and ran stably on the host computer and when the eNodeB software was started, it successfully identified the *Pluto* SDR and connected to the EPC. However, the modem did not detect the downlink.

The results of these tests can not be used to verify that *srsRAN* can be used for the LTE implementation. However, the results were to be expected since *srsRAN* recommends that a USB 3.0 connection is used for the SDR and is known to have issues with USB 2.0 as mentioned in Section 3.4.1.

Despite these results, *srsRAN* is still considered a viable solution, as the software itself ran without issues. However, a more capable SDR than the *ADALM Pluto* is most likely needed for a successful implementation.

3.7.2 GPS-SDR-SIM Test with Pluto SDR

Tests of the GPS implementation were then performed with the *Pluto*, but the *GPS-SDR-SIM* does not support transmitting the signal with a *Pluto* SDR using USB. The branched project *multi-gps-sdr-sim* was therefore used instead.

After connecting the SDR to the modem and transmitting the signal, the NMEA log, which is the serial channel where GPS information is reported, indicated that the receiver was able to identify the ID numbers of the simulated satellites. Occasionally the elevation information of some satellites was decoded, but the modem was unable to determine a position from the information.

The receiver also reported some satellite IDs which were not in the list of simulated satellites. These satellites were not reported when the SDR did not transmit. This could indicate a problem in the signal generation, but is most likely caused by the insufficient frequency stability of the SDR.

These results indicated that the *GPS-SDR-SIM* can generate a signal that the tested modem can decode information from, but the *Pluto* was not stable enough to transmit it correctly.

3.7.3 Feasibility Conclusion

From the performed tests, along with the results of previously done research on the subject presented earlier in this chapter, a conclusion was drawn that the suggested solution was feasible but could not be proven to work. It is deemed feasible since, for the LTE implementation, the EPC and eNodeB ran successfully but the transmitted signal by the *Pluto* SDR could not be identified by the modem. For the GPS simulation, the receiver could decode information from the generated and transmitted signal, but not enough to establish a position. Both these problems could be caused by the insufficient specifications of the *ADALM Pluto* and could be solved with an SDR with specifications within the requirements.

4

Implementation

This chapter explains how the software environment of the testbed was set up, how the LTE and GPS tests were implemented, and how the hardware components were assembled.

4.1 Software Environment Setup

As explained in Chapter 3, *Ubuntu Linux* is the recommended operating system for *srsRAN*. Since the *Rock Pi 4* is an ARM based platform, *Armbian* was chosen as it is an *Ubuntu Linux* image optimized for ARM based SBCs' [74]. The chosen version was *Armbian Focal EU* for the *Rock Pi 4 A/B/C* which is an *Ubuntu 20.04* image [75].

After downloading and flashing the *Armbian* image to an SD card, it was inserted into the *Rock Pi 4*. Before any new software was installed, the pre-installed software was updated. The CPU governor was also set to *performance* mode so that *srsRAN* would run reliably.

4.1.1 UHD

The *UHD* software was available as packages to be installed in *Ubuntu*. It was however an older version and since the power calibration feature added in the later releases was desired, the software was built and installed from source code.

The *UHD* source code for version 4.1.0.4 was therefore downloaded from *GitHub*, the dependencies were installed, and then the software was built with *CMake*. The built project was then installed and the FPGA images for the *USRP* devices were downloaded. With the *B205mini-i* SDR connected to the USB 3.0 port, the function of the drivers could be verified by probing the SDR.

4.1.2 srsRAN

The *srsRAN* software was also available as a package installation, but since it was built with an older version of *UHD*, *srsRAN* was instead built and installed from source code.

The source code for the latest release (21.10) were downloaded from *GitHub*, the dependencies were installed, and the software was built with *CMake*. The built software was installed in the system and the configuration files were installed with *user* configuration.

The EPC and the eNodeB were configured to operate with $MCC = 208$ and $MNC = 92$, which corresponds to a French test network. The reason for this was because the used SIM cards from *Open-Cells* [76], arrived pre-programmed with these values. Furthermore, a user with the SIM card's IMSI, Ki, OPc values was added to the HSS database.

The eNodeB was configured to operate in the middle block of LTE band 5. The reason band 5 was chosen was because it is a lower frequency band (around 800 MHz) and it was not used in Sweden. The LTE band can be changed to another band since the RF path is not restricted with a duplexer as described in Section 3.5.

The eNodeB was also configured to operate with 6 RBs which corresponds to a bandwidth of 1.4 MHz since less RBs requires less CPU resources. Another benefit of a narrow bandwidth is that the transmitted power was distributed over fewer frequency components. Therefore, a signal with narrower bandwidth has better SNR than a wider bandwidth if transmitted with the same power.

The final configuration for $srsRAN$ was to include the serial number of the dedicated SDR for LTE testing in the eNodeB configuration. If the serial number was not included, there was a risk that the eNodeB started transmitting and receiving with the wrong SDR. The SDR serial number could be reported by probing the device with *UHD*.

4.1.3 GPS-SDR-SIM

The source code for the *GPS-SDR-SIM* software was downloaded from *GitHub*. However, before building the software, *GNU Radio* was downloaded since it is required for operation with *USRP* SDRs. *GNU Radio* in turn requires *Volk* which also was downloaded. Note that *UHD* was needed to be installed before *GNU Radio* for it to work as intended.

All dependencies for the software were installed prior to the build process. The main branch of *Volk* (commit: $f0eb99e$) was downloaded, built and installed, followed by $v3.9.4.0$ of *GNU Radio*. Finally, the main branch of *GPS-SDR-SIM* (commit $366d4c4$) was then downloaded and built.

To be able to generate a signal file with the software, the latest updated GPS broadcast ephemeris file was needed. Files with the updated ephemeris data are available at *NASA's* website [77]. A GPS signal file could then be generated with a static or dynamic position. Since a *USRP* device was used, the sample rate was to be specified to 2500000 (2.5 MSPS) and the I/Q data format was to be specified to 8 bits.

The coordinates for a static location close to the development site in Gothenburg (Sweden) were then used when generating the signal file. This location was chosen to keep the simulation realistic and comparable to that of testing with an antenna. An alternative file was also created for a location in Uganda with better satellite placements.

A bash script for starting the signal transmission was created to always start with the correct input parameters, which are the sample rate and I/Q data format. These values would match the values specified when generating the signal file. The serial number for the SDR dedicated for the GPS testing was also included as a device argument to ensure that the correct SDR was used to transmit the signal.

4.2 Test Implementation

A set of bash scripts was created to automate the execution and configuration of *srsRAN* and *GPS-SDR-SIM* for the testing process.

Since several applications ran in parallel, *GNU Screen* was used as a terminal multiplexer for running the software and test scripts in. *GNU Screen* was configured to output a log file with timestamps, which was updated every second. It was also used to establish a serial connection with the modem for both an AT terminal and an NMEA log.

For both the LTE and GPS test, the scripts were implemented to generate a directory named with the date. In the directory, a subdirectory for each test was then generated to store the log files, which could be used for later analysis.

4.2.1 LTE Test Implementation

The bash script executing the test for LTE is shown in Fig. 4.1 where a flowchart for the overall process of the test is presented. First, a timer starts so that if the test would take too much time, it would be considered a fail and then be aborted. Then the AT terminal is set up so that AT commands can be transmitted to the modem in order to be able to extract response information from it. Afterwards, the EPC starts as a background process. This is done simultaneously as a timer is instantiated so that the time for important events can be reported. After a one-second delay, the eNodeB is initialized in the same way. This delay was introduced in order to give the EPC time to start, as the eNodeB can't successfully connect to the EPC otherwise.

After the eNodeB has started, the output log file for the eNodeB is probed every second to check if the eNodeB is ready to transmit or not. This is the "Read eNodeB status" process in Fig. 4.1. If the eNodeB isn't ready, it will perform another check after one second. When it is ready, it will print out the time it took to start transmit through the eNodeB as well as to keep moving forward with the check for the downlink.

The next step in the test is the downlink check that is performed every second until an available network is found. This step is the process "Read available networks". This check consists of sending AT commands to the modem and probing the output log files for the correct network.

After an available network is found, the test proceeds to check if the modem is registered to the network. The registration check is performed every 0.5 s with an

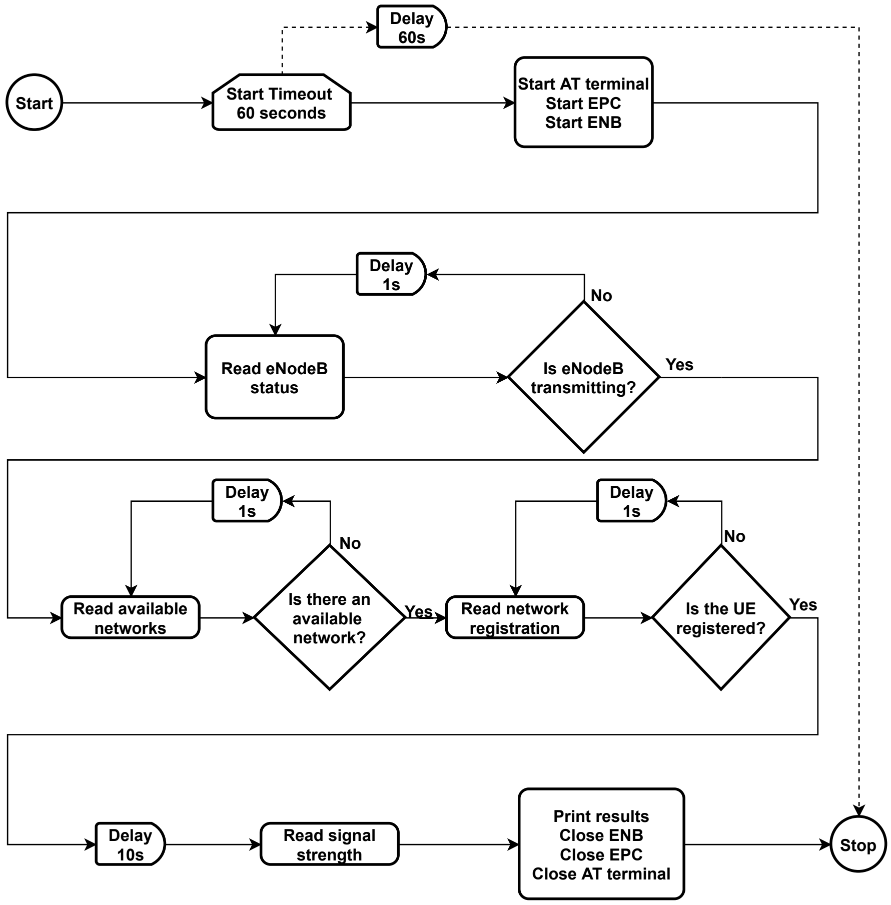


Figure 4.1: Flowchart of the implemented LTE test.

AT command that checks if the registration of the network has completed. This corresponds to the process “Read network registration”.

After a successful registration, the received signal strength and the reported network name is read in the AT terminal and then saved to the results file. Before the signal strength is read from the log file, a delay of 10 s is introduced since the signal strength in the modem takes a while to stabilize. The test is then considered complete and the background processes and the AT terminal are closed down.

After a test has completed, the time for the test, the time for eNodeB to get ready, the time for the downlink to be detected, the time for the uplink to be established, signal strength, and if the test was a success or fail, were written out to a results file in the folder for the specific date.

4.2.2 GPS Test Implementation

The function of the bash script implemented for the GPS test can be seen in Fig. 4.2. The script starts with introducing a timer and setting a 180 s timeout delay to stop the script and report a failed test if it has not completed within this time. A terminal with the NMEA log is opened to read information of the satellites the modem detects and the data decoded from the GPS signal. Then the GPS signal transmission is started and the status of *GPS-SDR-SIM* is continuously checked until the SDR has started to transmit the GPS signal. The startup time for the transmission is a few seconds and the time is noted for each test.

When the SDR is transmitting, the NMEA log is used to read the detected satellites and to see if a position has been calculated. When a position has been calculated, the elapsed time is noted and the test is considered complete. The script ends with closing the NMEA terminal and ending the GPS signal transmission. Then the time of the events, the detected satellites, and the test parameters are written out to a results file in the folder for the specific date.

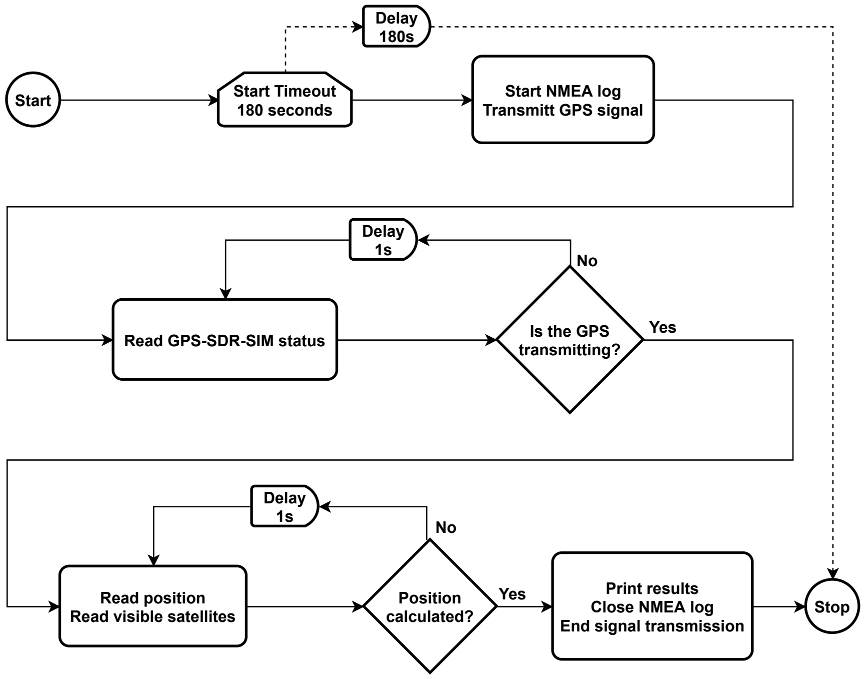


Figure 4.2: Flowchart of the implemented GPS test.

4.3 Hardware Assembly

The components of the testbed were connected together as illustrated in Fig. 4.3 where all individual hardware components are included. A list with all component names can be found in Appendix A. The total cost was approximately 22,000 SEK, and the price of the individual components is included in the list.

Instead of connecting the USB cables from the SDRs directly to the *Rock Pi*, they are connected through a powered USB 3.0 hub. This is because each of the *B205mini-i* devices draw a maximum of 4.5 W (0.9 A on 5 V) which is more than the power supply of the *Rock Pi* can deliver [78]. An alternative to using a USB hub is investing in a power supply which supplies enough power for the *Rock Pi* and the SDRs.

The main components of the testbed were mounted together as a stack with distance bolts to reduce the desk footprint. The SDRs were mounted at the bottom of the stack, with the power divider/combiner above them. On top of the stack is the

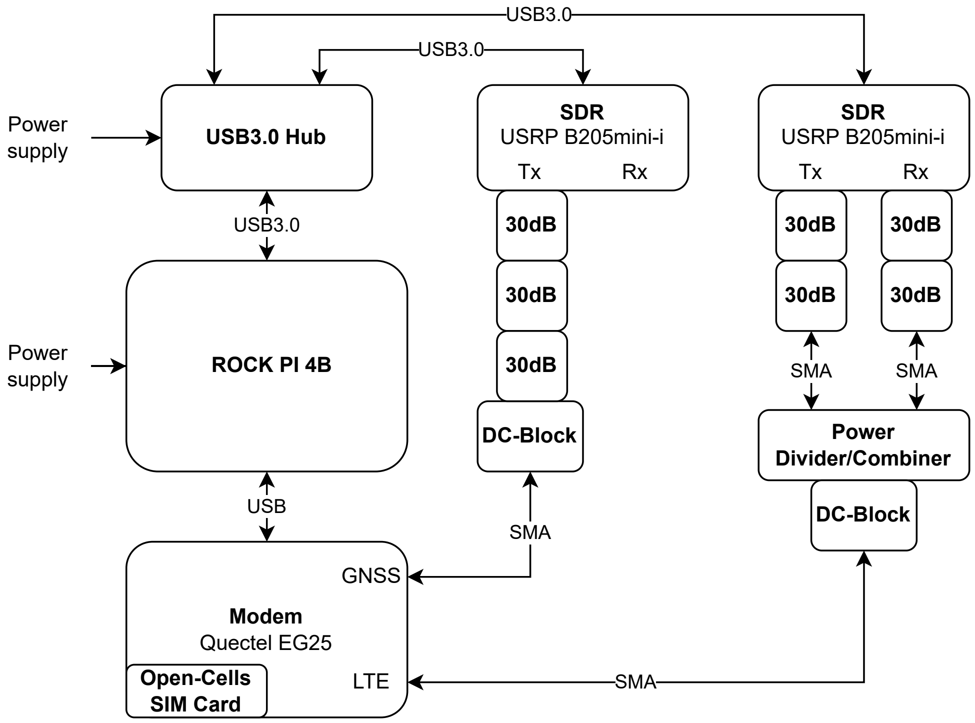


Figure 4.3: Block diagram of all the assembled components in the testbed including the modem used for testing.

Rock Pi with the GPIO pins mounted upwards for easier accessibility. The *Rock Pi* was mounted using long bolts to not restrict the cooling of the heat sink. To assemble the stack in this way, a mounting bracket was designed and 3D-printed. The designed bracket can be seen in Fig. 4.4.

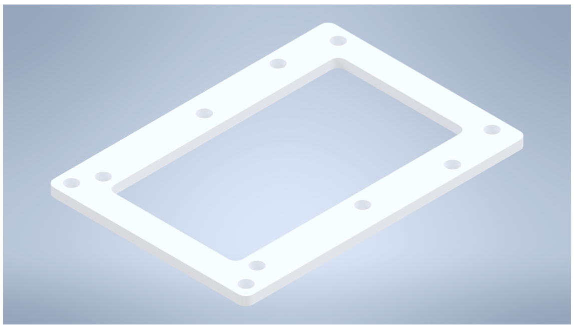


Figure 4.4: Mounting bracket for matching holes of the stacked components.

The final implementation of the testbed with all components is shown in Fig. 4.5 and the testbed connected to the *Quectel* evaluation board, can be seen in Fig. 4.6.

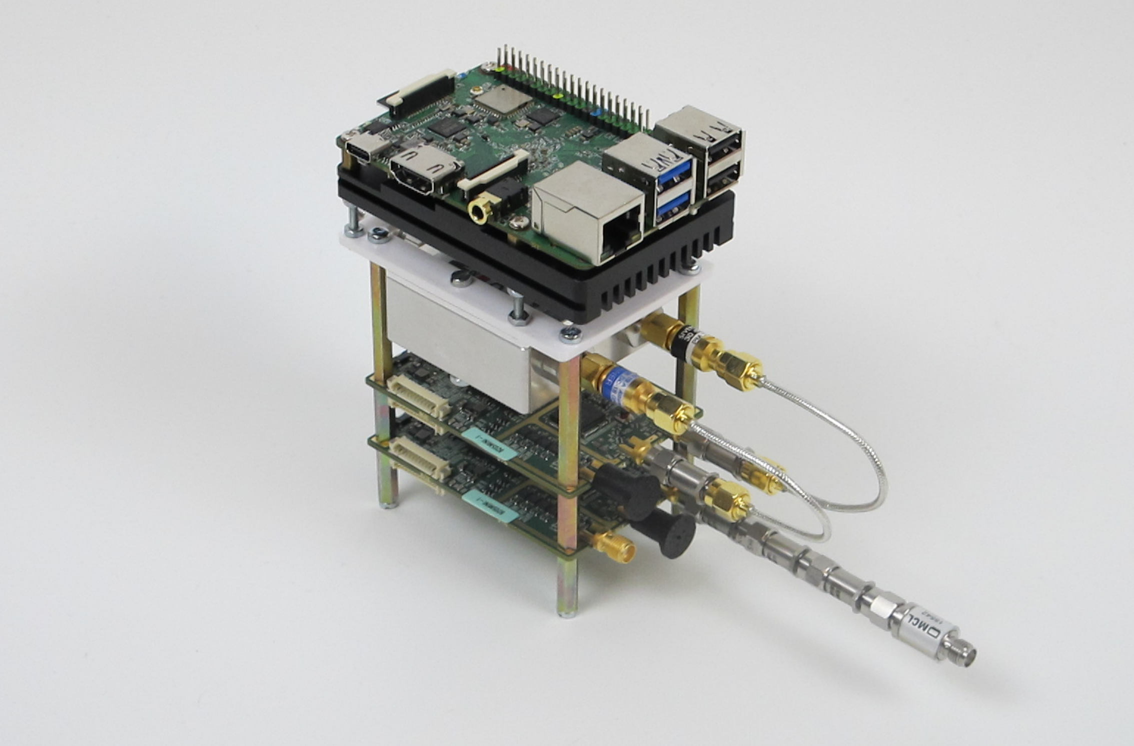


Figure 4.5: Final testbed implementation.

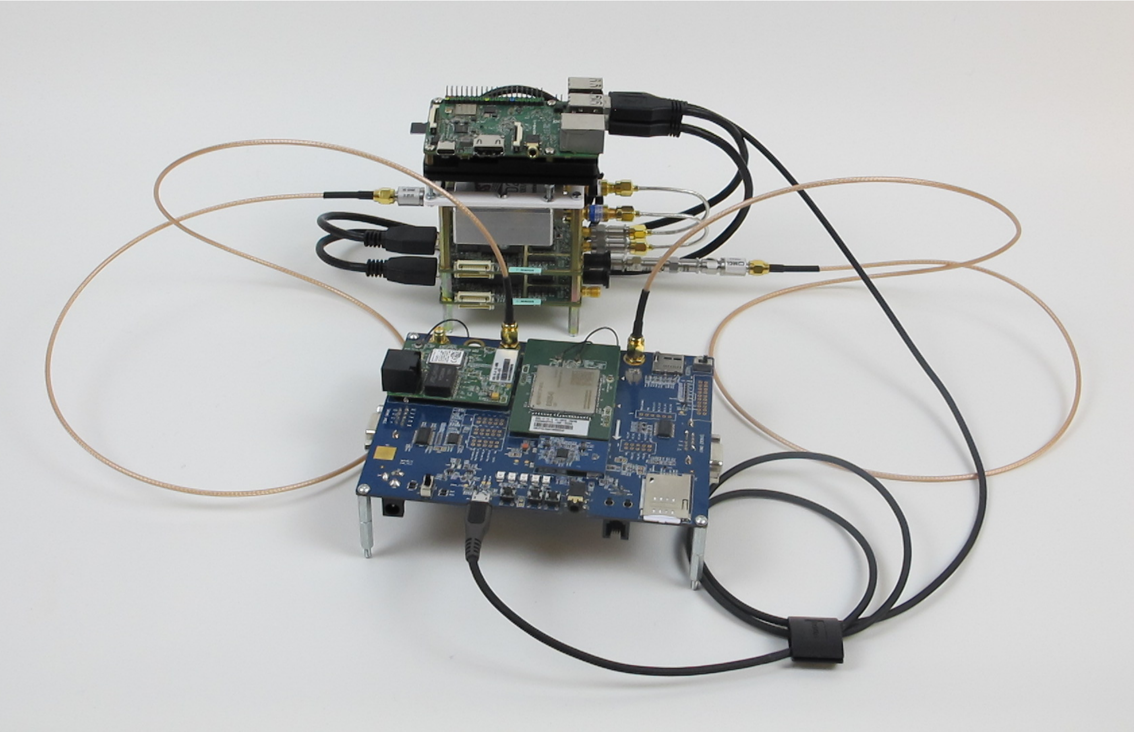


Figure 4.6: Final testbed implementation with the modem used for testing.

5
System Testing and Results

In this chapter, the tests performed on the implementation are presented along with the results of these tests. The tests focus on connection time, stability, and reliability to determine if the implementation is suitable for a production environment. The hardware was assembled as presented in Fig. 4.3 for all the tests that do not state otherwise. A *Tektronix MDO3104* oscilloscope, with an *MDO3SA* upgrade, was used to measure signal power and observe the characteristics of the transmitted signals.

During the testing process, the behavior of the system as a whole was studied. There were no crashes of either the OS or any of the software programs, and there were also no observed thermal issues.

5.1 General Function Testing

The initial tests were performed to verify that the core functions of the testbed functioned as intended with the hardware and software. The function tests were performed to verify that the *B205mini-i*, together with *srsRAN* and *GPS-SDR-SIM*, was able to transmit an LTE and a GPS signal which the *Quectel EG25-G* modem treats as an official signal. Another test was performed to verify that the *Rock Pi 4B* is able to run the tasks in parallel since it is how the testbed is intended to operate. In addition to this, the frequency stability of the SDR was investigated since it is known from the investigation to potentially cause issues.

5.1.1 Initial LTE Function Test

For LTE, the initial function test was performed by starting the EPC and the eNodeB, then using AT commands to extract the information of the recognized networks. The modem was configured to only search for networks in band 5 which is the band used by the eNodeB. After a short delay, a network in band 5 was recognized. The reported network name confirmed that it indeed was the network of the testbed. The reported signal strength was also feasible compared to the set gain value together with the hardware attenuation. After inserting the SIM card, with information added to the HSS, both the eNodeB and the EPC showed activity and the modem reported that it had connected to the network.

During the test, there were no reports of missed samples by the SDR. This initial test confirmed that the *USRP B205mini-i* SDR along with the software *srsRAN* can be used to transmit an LTE signal, which the *Quectel EG25-G* modem handles correctly. It can also receive the signal from the modem correctly and register the modem to the network.

5.1.2 Initial GPS Function Test

For GPS, the initial function test was performed with a hardware attenuation of 60 dB instead of 90 dB to ensure that the signal strength was sufficient. After starting the transmission of the GPS signal with a gain of 30, the NMEA log started reporting that there were detected satellites. All the reported IDs matched the IDs of the simulated satellites in the generated GPS signal file. Eventually, the degrees of elevation and azimuth were reported, and a position was calculated with coordinates that matches the coordinates used when generating the GPS signal file.

During the test, there were no reports of missed samples by the SDR. This test confirmed that the *USRP B205mini-i* SDR along with the software *GPS-SDR-SIM* can be used to transmit a GPS signal in which the *Quectel EG25-G* modem can calculate a position from.

5.1.3 CPU Usage and USB Data Rate Test

Another test was performed with both software programs running in parallel without issues, and the modem reported similar results as the initial LTE and GPS function tests. During this test, the *Htop* command was used to monitor the CPU usage, which is presented in Table 5.1.

Table 5.1: CPU usage of the software programs.

Application	Idle	GPS-SDR-SIM	srsENB	srsEPC
CPU usage [% of single core]	0.7 – 1.3	20 – 28	50 – 60	3 – 6

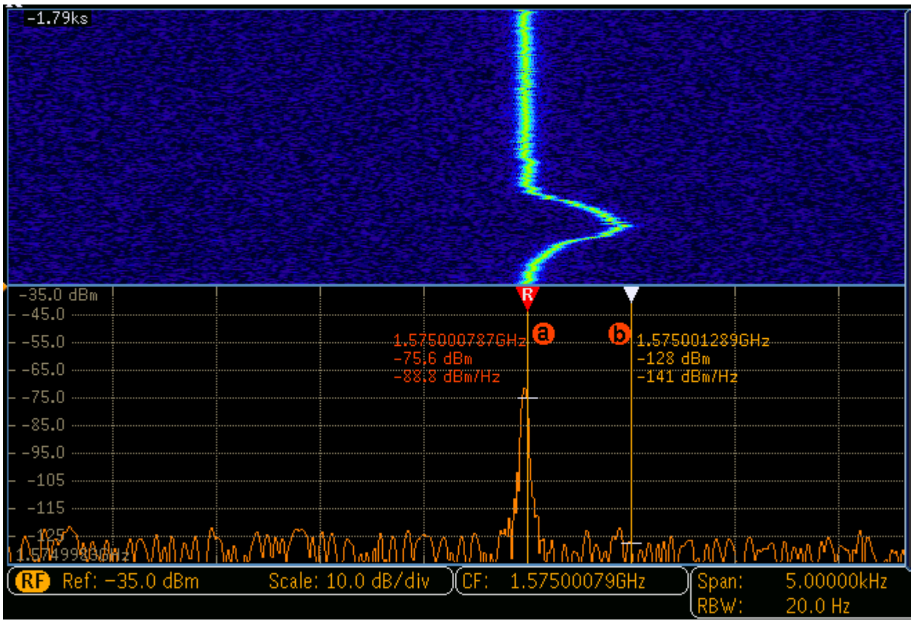
The CPU usage for both *GPS-SDR-SIM* and *srsENB* is distributed over multiple cores. This resulted in approximately 10 – 25 % usage on five of the six CPU cores and a low usage on the last core. Combining the numbers for the cores, the total usage was approximately 20 % of the available CPU resources.

During this test, there were no missed samples reported by any of the SDRs. These observations indicated that the *Rock Pi 4B* has sufficient CPU resources to run both the software in parallel and that the internal USB bus can handle the data transmission to the SDRs.

5.1.4 Frequency Stability Test

The frequency stability of the SDR was investigated since the *B205mini-i* did not meet the set requirement as described in Section 3.4.4. During the test, a single frequency component at 1575 MHz was transmitted by one of the SDRs and inspected with the oscilloscope with 60 dB attenuation.

At first the test ran for 20 minutes without interruption and the frequency remained stable with only a slightly jagged line in the waterfall diagram. Then airflow was added directly at the SDR board for approximately 5 minutes, which caused a significant frequency drift as can be seen in Fig. 5.1. The airflow was then removed and the frequency drifted back to the initial location. The markers were used to approximate the drift to 0.5 ppm.



Ref: -35.0 dBm
Scale: 10.0 dB/div
CF: 1.57500079GHz
Span: 5.00000kHz
RBW: 20.0 Hz

Figure 5.1: Temperature impact on frequency stability and the transmitted signal.

5.2 LTE Implementation Testing

The following tests were performed to investigate how long the connection times were and how stable the connection was. The testing was performed by repeatedly connecting and disconnecting the modem to the LTE network of the testbed and logging the results. In these tests, the presented times for the uplink and downlink are measured from when the transmission starts, which is approximately 5 s from when all tests start.

Tests were also performed to investigate if the communication is comparable to a real world scenario. To investigate this, the signal power was analyzed as well as the signal appearance.

5.2.1 Connection Time Testing

The test script mentioned in Section 4.2.1 was altered so that automated testing could be performed. Between each test, a delay of one minute was introduced for the modem to completely lose the previous connection. In the first test, the gain was swept over the interval 20-80 and the resulting uplink times are shown in Fig. 5.2. All the tests with a gain less than 35 resulted in a failed test and were therefore discarded. This revealed the least required signal strength necessary for the modem to be able to detect the network and connect to it.

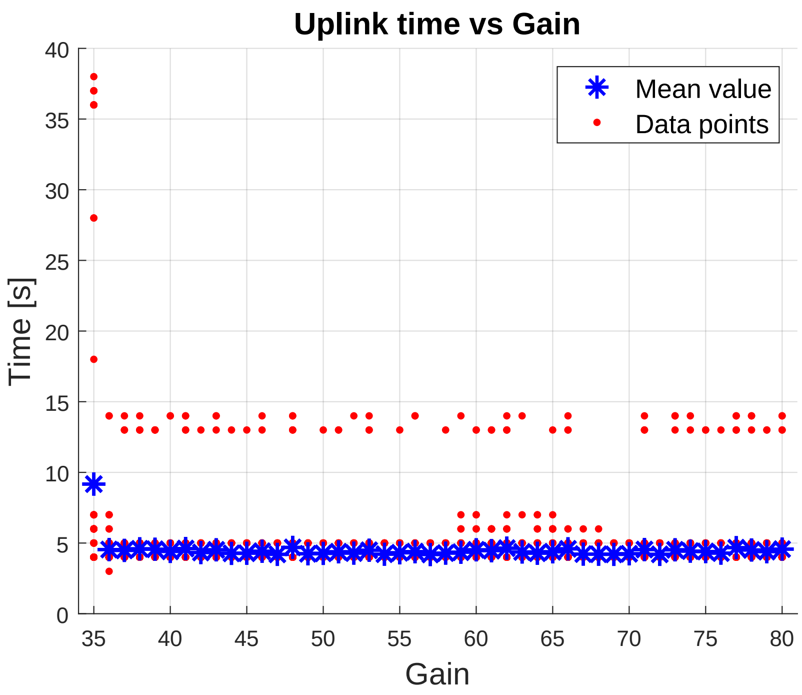


Figure 5.2: Uplink time for gains 35 to 80 with 100 data points per gain setting.

In Fig. 5.2, a stable level can be seen for the uplink times around 4.5 s, with only a few outliers. One deviation can be seen for gain 35 where the time for detecting the downlink has a mean time that is more than twice as long as the other gain values. Gain 35 also failed to detect the downlink four times and failed to establish an uplink three other times. Another deviation is that there is a set of outliers close to the stable level for gains 59 to 68 which are caused by a delay in the downlink detection.

For gain 36 and above, an uplink was successfully established every time and the mean values for the downlink are between 3.1 s and 3.3 s. The mean value for the uplink is between 4.2 s and 4.7 s. For gain 35 the downlink and uplink mean times are 7.5 s and 9.2 s respectively.

Another test with gain values of 75 to 80 was performed in order to collect more data points with a signal power comparable to a realistic scenario. This was because the highest gain value of 80 reported a CSQ value lower than the maximum in the previous test. A CSQ value is what the modem uses to report the signal strength and is further explained in Section 5.2.3. In this test, a total of 230 data points were collected for each gain value. The results are presented in Table 5.2 and shows a stable behavior with an uplink establishment at 4 s or 5 s with a few outliers at 13 s or 14 s . The two distinct levels for both of the cases are consequences of using a time resolution of 1 s.

Table 5.2: Uplink Times for gains 75 to 80 with 230 data points each.

Uplink Time [s] **Gain**	**4**	**5**	**13**	**14**
75	181 (79%)	43 (19%)	3 (1%)	3 (1%)
76	187 (81%)	39 (17%)	0 (0%)	4 (2%)
77	170 (74%)	53 (23%)	5 (2%)	2 (1%)
78	178 (77%)	46 (20%)	4 (2%)	2 (1%)
79	179 (78%)	44 (19%)	3 (1%)	4 (2%)
80	179 (78%)	46 (20%)	2 (1%)	3 (1%)

A final test was performed for a gain value of 80 where 2800 data points were gathered. This was done to get an accurate mean value and percentage for the occurrences of outliers. The results can be seen in Table 5.3. The mean times for the downlink and uplink are 3.1 s and 4.4 s respectively. Out of the 2800 connections were 55 above 5 s which corresponds to 1.96 %.

Table 5.3: Uplink Times for gain 80 with 2800 data points.

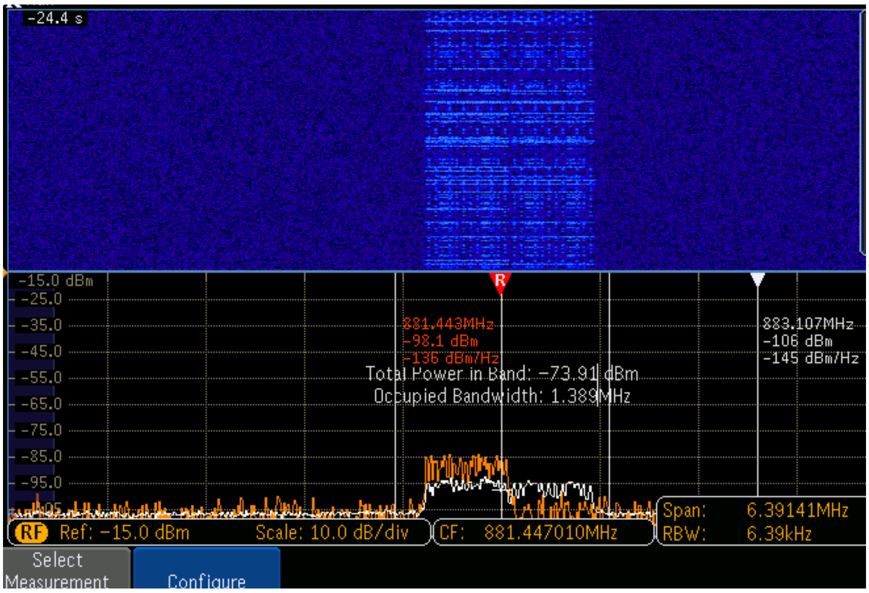
Uplink Time [s]	**4**	**5**	**13**	**14**
Occurrences	2155 (77%)	590 (21%)	38 (1.4%)	17 (0.6%)

5.2.2 LTE Signal Power Analysis

The signal characteristics of the transmitted LTE signal were studied with the oscilloscope. Measurements with a change in gain were performed, leaving the amount of allocated RB constant. A measurement with 80 gain using 6 RB (1.4 MHz bandwidth) in middle block of LTE band 5 was conducted. The EPC and the eNodeB were both started and when the eNodeB started to transmit, the signal was observable on the oscilloscope. The center frequency of the signal was 881.5 MHz and the bandwidth was slightly less than 1.4 MHz, as can be seen in Fig. 5.3. These settings resulted in a signal level at -73 dBm which is less than what a wireless communication with an official signal could result in.

5.2.3 Signal Power Stability Analysis

For the tests mentioned in Section 5.2.1, the reported signal power was analyzed and showed a stable behavior for gain 45 and above. The results of the test are shown in Fig. 5.4. It can be seen that the signal strength did not rise proportionally between gain levels 35 to 45.



881.443MHz
-98.1 dBm
-136 dBm/Hz

883.107MHz
-106 dBm
-145 dBm/Hz

Total Power in Band: -73.91 dBm
Occupied Bandwidth: 1.389MHz

RF Ref: -15.0 dBm Scale: 10.0 dB/div
CF: 881.447010MHz
Span: 6.39141MHz
RBW: 6.39kHz

Figure 5.3: Measurement of signal level using the oscilloscope at 80 gain using 6 RBs.

For gain values between 35 and 42, the mean CSQ value varies between 2.8 and 3.0. However, at gain 45 it starts to increase proportional to the increase in gain, with a mean value of 5 CSQ for gain 45 to 21.9 CSQ for gain 80. By using Eq. 5.1 where x is the CSQ value, 2.8 CSQ corresponds to a signal power of $-107.4\,$dBm and 21.9 CSQ corresponds to $-69.2\,$dBm.

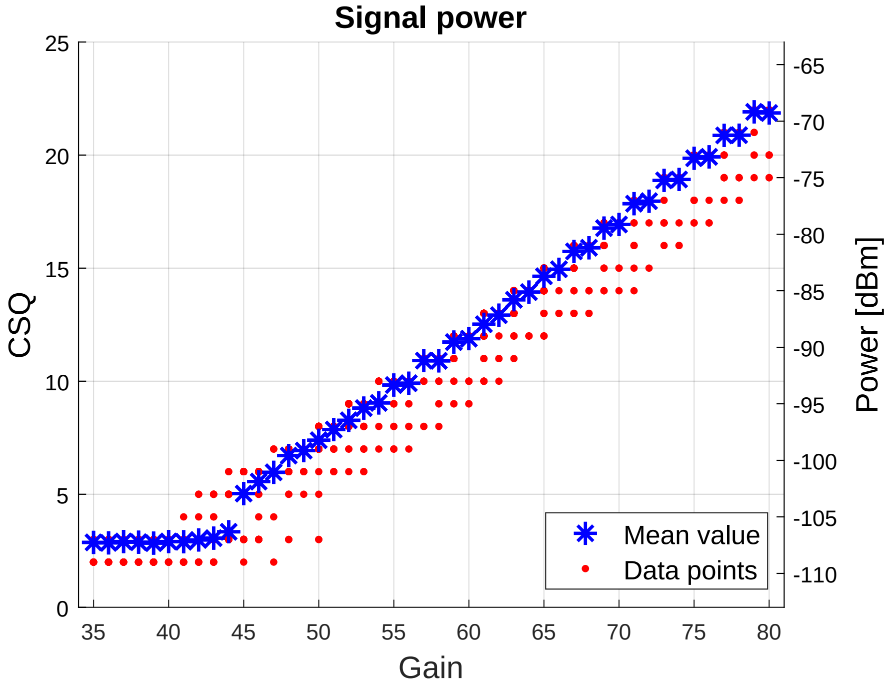


Figure 5.4: Reported signal power for gains 35 to 80 with 100 data points per gain setting.

$$f(x) = 2 \cdot x - 113 \ [dBm] \tag{5.1}$$

The data from the second test, with more data points for each gain value, was used to investigate the variation percentage. The used gain values were 75 to 80 and 230 data points were collected for each gain value. As previously explained, the range of 75 to 80 was chosen since they are in the top of the gain range and are comparable to a realistic scenario. The occurrences for the CSQ values are shown in Table 5.4 and shows a variation of 4 CSQ. Between 4 and 6 percent of the outcomes were not the highest CSQ value reported for that gain.

Table 5.4: Uplink CSQ Occurrences for gains 75 to 80 with 230 data points each.

CSQ Gain	**17**	**18**	**19**	**20**	**21**	**22**
75	5 (2%)	6 (3%)	2 (1%)	217 (94%)	0	0
76	3 (1%)	6 (3%)	0 (%)	221 (96%)	0	0
77	0	6 (3%)	5 (2%)	3 (1%)	216 (94%)	0
78	0	3 (1%)	4 (2%)	0	223 (97%)	0
79	0	0	5 (2%)	8 (3%)	2 (1%)	215 (93%)
80	0	0	4 (2%)	4 (2%)	0	222 (97%)

Analysis of the final test, with a gain value of 80 with 2800 data points, resulted in a mean CSQ value of 21.9. The occurrences for the CSQs are shown in Table. 5.5. The resulting outcome shows similar percentages as in the previous test for gain value 80. Out of the 55 outcomes with a CSQ lower than the maximum value for that gain did 42 correlate with an outlying uplink time.

Table 5.5: Uplink CSQ Occurrences for gains 75 to 80 with 230 data points each.

CSQ	**17**	**18**	**19**	**20**	**21**	**22**
Occurrences	0	0	33 (1%)	87 (3%)	0	2680 (96%)

The test with gains from 75 to 80 was performed again with only the downlink to investigate if the signal activity of connecting a UE affected the CSQ. The SIM card was therefore removed from the modem to only test the downlink and not the uplink The used gain values were again 75 to 80 and 150 data points were collected for each gain value. The results showed that all the CSQ values matched the highest value expected for that gain.

5.2.4 LTE Test with Power Cycle

A test was performed with a manual power cycle of the modem between each connection. This was done to investigate if the delay of one minute between the connections was enough for the modem to completely disconnect from the testbed. A gain value of 80 was used, and 30 data points were collected. The resulting times and behavior completely matched the previous test with 80 gain. In the 30 connections, one outlier appeared, which roughly matches the ratio for outliers in the previous connection tests.

5.2.5 LTE Test with PLS8 Modem

The final test of the LTE implementation was performed with another modem, which was the *PLS8-E*. For this test, the testbed had to be reconfigured to operate in LTE band 8 instead of band 5 as in previous tests. This was because the modem did only support bands used in Europe. The used gain value was 80 and a total of 10 data points were collected. The resulting uplink times can be seen in Table 5.6 and ranged from 3 s to 35 s with a mean value of 16.9 s.

Table 5.6: Uplink times with PLS8-E modem

Test Nr.	1	2	3	4	5	6	7	8	9	10
Uplink [s]	35	13	3	3	3	34	8	17	28	25

5.3 GPS Implementation Testing

To investigate how the GPS test implementation in the testbed performs, a set of tests was performed to analyze how long it takes to establish a position and if the transmitted signal is comparable to a wireless GPS transmission, in terms of signal power. The time-to-first-fix (TTFF), which is how long it takes for the modem to calculate a position from the signal, is measured from when the signal starts transmitting. The startup time for the SDR is approximately 5 s and is the same for all tests.

5.3.1 Time-To-First-Fix (TTFF) Testing

To test the TTFF and the stability of the GPS test implementation, an automated testing script was created to run the GPS test script described in Section 4.2.2 repeatedly with different configurations. The original test script also needed to be modified for the repeated testing since the stored GPS data in the modem needs to be cleared between each test. By clearing the GPS data, the modem is forced to do a cold start, which makes the testing conditions equal for all the tests. The modem is cleared by starting an AT terminal and sending AT commands to the modem before the GPS signal transmission is started. The script reads the AT log as well as the NMEA log to confirm that the GPS data is correctly cleared and that there is no stored satellite information from the previous test.

A test with gain setting ranging from 55 to 75 with 100 data points each was performed using the alternative GPS signal file with optimal satellite placements. The outcome of this test can be seen in Fig. 5.5. The lower gain setting of 55 was chosen since it was the lowest gain that connected reliably. With a gain value of 54 did approximately 10 % of the tests connect in 180 seconds.

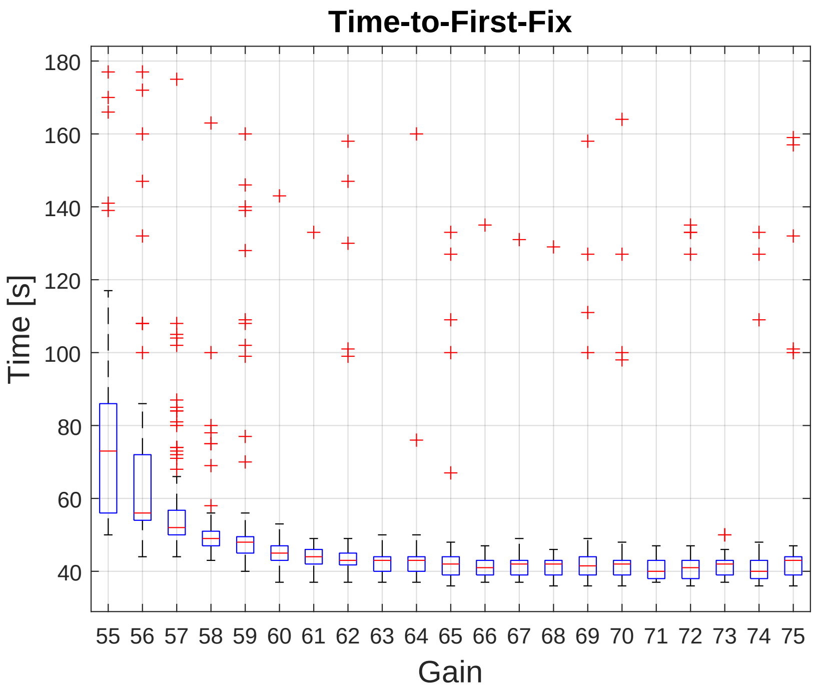


Figure 5.5: Time-To-First-Fix for gains 55 to 75 with 100 data points per gain setting.

As can be seen in Fig. 5.5, the time required for the modem to calculate a position from the GPS signal is affected by the gain value for the lower gain values, but not so much for the higher ones. The mean time is affected exponentially by the signal power, and remains close to constant for gain 60 and higher. It can also be seen that there are a number of outliers for most of the gain settings, which can be seen together with the mean TTFF in Table 5.7. The NMEA logs of the outliers show that the modem resets the satellites in view and restarts the calculations, causing the increase in time.

Table 5.7: Test results for gains 55 to 75 with 100 data points per gain setting.

Gain	**55**	**56**	**57**	**58**	**59**	**60**	**61**	**62**	**63**	**64**	**65**
Mean TTFF [s]	78	66	58	52	55	46	45	47	42	44	45
Above 90s	18	8	5	2	9	1	1	5	0	1	4
Not connected in 180s	10	-	1	-	-	2	-	-	-	-	-
Gain	**66**	**67**	**68**	**69**	**70**	**71**	**72**	**73**	**74**	**75**	
Mean TTFF [s]	42	42	42	45	44	41	44	41	43	46	
Above 90s	1	1	1	4	4	0	4	0	3	5	
Not connected in 180s	-	-	-	-	-	-	-	2	-	-	-

To get an accurate mean value and percentage of the times above the set time limit, a test with a single gain with more data points was performed. The gain value of 60 was chosen since it is the lowest gain value where the TTFF does not depend on the gain value. The outcome of this test with 2600 runs can be seen in Fig. 5.6.

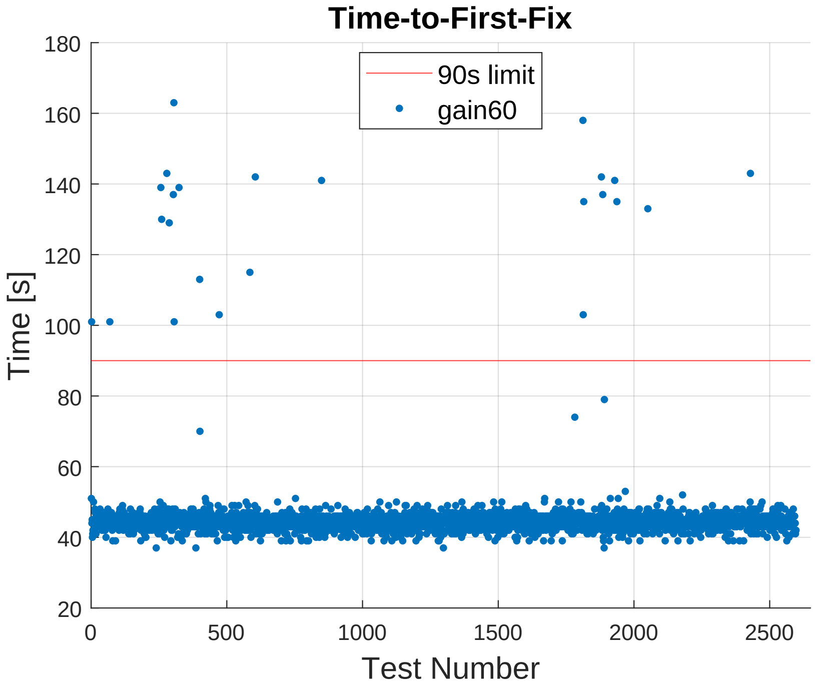


Figure 5.6: Time-To-First-Fix for gain 60 with 2600 data points.

This test with 2600 connections resulted in a mean TTFF of 45.2 s. During the test did 28 connections take more than 53 s. Of these were 25 above 90 s and one did not connect in 180 s. This means that in 98.9% of these cases, the modem did calculate a position in 53 s or less.

5.3.2 GPS Signal Power Analysis

To determine if the transmitted GPS signal is comparable with a real signal, the power of the transmitted GPS signal was analyzed. Since the power of public wireless GPS signals are below the noise floor, they can not be inspected with an oscilloscope. There is also no function in the modem to report the signal power, so the transmitted signal power can only be compared to the theoretical value.

To measure the power of the transmitted signal of the testbed, only 30 dB of attenuation was used when connecting the GPS test SDR to the oscilloscope. When transmitting with a gain of 60 the reported power was -42 dBm as can be seen in Fig. 5.7.

With an attenuation of 90 dB, which was used for testing the modem, the resulting power level would be approximately -102 dBm. This is 23 dB higher than the signal power of the official GPS signals with -125 dBm, as mentioned in Section 3.5.2. The measured signal power is therefore not comparable to what a passive antenna would

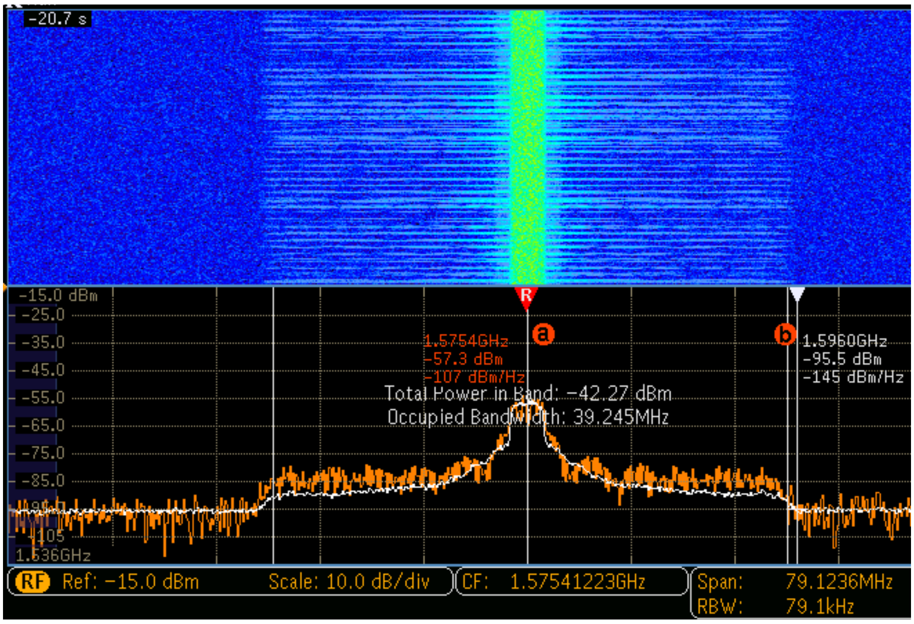


Figure 5.7: GPS signal power with 60 gain and 30dB attenuation.

deliver to the modem, but it is in the range of what an active antenna would. It is not uncommon for active antennas to amplify the signal with up to 28 dB [79].

5.3.3 GPS Test with Power Cycle

To investigate if testing a forced cold start is comparable to that of testing a freshly configured modem, a test was performed with manual power cycles of the modem between tests. The gain value was 60 as in previous tests and 10 data points were collected. The resulting TTFF was higher than for previous tests, ranging from 83 s to 142 s with a mean value of 113.6 s.

5.3.4 GPS Test with PLS8 Modem

The final test of the GPS implementation was performed with another modem, which was the *PLS8-E*. The test was also performed with a gain value of 60 and 10 data points were collected. The tests were manually performed due to the change of interface between the modems. The result is shown in Table 5.8 and the TTFF ranged from 72 s to 105 s, with a mean value of 90.6 s.

Table 5.8: TTFF with PLS8-E modem.

Test Nr.	1	2	3	4	5	6	7	8	9	10
TTFF [s]	75	72	98	101	97	97	77	103	81	105

6

Discussion

This chapter covers the discussion about choices made in the implementation, as well as the results of the testing performed on the testbed. There is also a discussion about the ethical aspects and some general thoughts about the work done.

6.1 Hardware Choices

Most of the design choices in Chapter 3 were made with cost efficiency as motivation. However, the choice of using two SDRs for running the tests in parallel was not. The implementation can be done with only one SDR, which would greatly reduce the total cost. This would increase the time required for testing, which is highly undesirable for a production environment. Another possibility of reducing the cost is to implement the GPS testing with a cheaper SDR, since it does not require USB 3.0. The main reason why we decided to use the same type of SDR for both implementations was because of the flexibility. If any of the test implementations wouldn't work or be too unstable, the SDR could be repurposed to perform the other test.

Another hardware choice was to include a DC blocker in the RF path for the LTE test even if there is no DC component for the LTE antenna. This was a safety feature so that if the operator of the testbed would connect the cable to the GNSS antenna port by mistake, the SDR wouldn't risk being damaged.

The power calibration feature, which was one of the reasons why an *Ettus USRP* SDR was chosen, was not utilized in the implementation. The reason for this was because the maximum transmitted signal power (which is determined by the gain value) does greatly affect the signal power, but it can not alone be used to predict the received power by the DUT. This was discovered in Section 5.2.3 where the reported signal power by the modem varied when an uplink was connected, which is discussed further in Section 6.3. The power calibration feature is still considered important for later implementation, since accurate testing of the received power could be important for the operator of the testbed. If it is not considered important, the use of a *BladeRF* SDR could be a possible alternative for reducing the total cost.

6.2 Test Implementation

The tests were implemented as bash scripts with a default configuration, which makes the testbed easy to use for the operator. For default configuration testing, the operator is only required to connect the USB and RF cables to the correct ports in the DUT and simply run the script. This is considered to fulfill the requirement of being easy to use by the operator, which was specified in Section 3.1.3. Both the hardware and software in the testbed are considered easy to maintain since it is easy to take apart and replace the hardware and the open source software allows for further development.

When the tests were implemented, a time resolution of one second was used for noting events. This is quite long considering that the LTE uplink times are generally less than 10 seconds as presented in Section 5.2.1. This resolution was however considered enough, and a higher resolution would be more resource intensive on the CPU. An increase in resolution would perhaps provide a bit more interesting data but would not improve the performance of the testbed, but only risk lowering it by increasing the CPU usage.

6.3 LTE Implementation Testing

The test results presented in Section 5.2 indicate that the LTE implementation in the testbed can reliably produce a signal which the modem can identify and connect to. This can be claimed because more than 8000 connections were presented with a gain value of 36 and above and none of these had an uplink time longer than 14 s.

When using gain 35 could the modem often identify the network, but not always, due to the low signal power. This gain value could therefore be used to test a worst-case scenario in terms of signal power for the *Quectel EG25-G* modem. However, for simple functional verification of the modem, there is no reason to use a gain value less than 80, since the signal produced is comparable to a realistic signal, as was established from the results presented in Section 5.2.2.

The time it took for the modem to identify the downlink had a mean time between 4.2 s and 4.7 s for the gain values 36 to 80 and 9.2 s for gain 35. These results were actually an improvement compared to the earlier tests we performed. An important discovery was made which indicate that if we configure the modem to check for networks only in a specific frequency band, the downlink detection time is lower. This most likely has to do with the processing of data in the modem.

The testing of the signal strength and the uplink times followed a stable behavior throughout the tests, even though there were some variation. However, at the gain levels 35 to 45 the reported signal strength did not rise proportionally to the gain value. This is most likely due to the fact that the transmitted signal is in proximity to the noise floor.

The reason for the variation of the signal strength, presented in Section 5.2.3, is probably due to the signal activity as indicated by the results for the downlink-only test. The synchronization signal sent out in the downlink for UEs to detect probably vary compared to the transmitted signal after a UE has been connected. Since there is variation in the reported signal strength, the “reference value”, which the reported value is compared to, is forced to be a range of 4 CSQ with the maximum value defined by the chosen gain. This range is predictable for the higher gain values, which is where the testbed should operate at for normal testing as previously mentioned. This is one of the reasons why the power calibration feature, discussed in Section 6.1, was considered less important than previously thought.

When looking at the results of the tests with a gain value of 75 to 80, the difference in time between the stable level and the outliers is always 9 s. A reason for this could be that if the uplink fails to be established, the modem might use a back-off duration before it attempts to establish a new one. Unfortunately, we found no way to confirm this or any way to reconfigure the modem to improve this delay.

Most of the tests with an outlying uplink time also had an outlying CSQ value. This indicates that there could be some activity in the modem which causes this outcome, or the problem might be located in the *srsRAN* software. If the problem is located in *srsRAN*, testing with another modem should show similar behavior.

The test performed with power cycling the modem between each connection indicates that the results from the other tests are comparable to testing a fresh modem. Therefore, similar times as presented in Section 5.2.3 could be expected when testing a *Quectel EG25-G*. The test performed with the *PLS8-E* modem indicates that different connection times are to be expected for different modems. However, the result for the *PLS8-E* test could also be affected by the change of frequency band from 5 to 8. This is because the official LTE signals possibly interferes with the transmitted signal from the testbed. All the tests successfully connected anyway, but depending on the tested modem and the used LTE band, this could potentially affect the connection times.

6.4 GPS Implementation Testing

In the results from the stability testing of the GPS implementation, we can see that there are some outliers, which take up to four times longer than the mean time. There are fewer outliers for the gains 60 and above than for the lower gains, but in the higher range it is seemingly unrelated to the gain value (signal power). This indicates that there are some stability issues somewhere in the setup. It may be located in the modem, but we deem it most likely to be caused by the frequency stability, since the logs showed that the modem did restart the position calculations. As testing on the frequency stability showed that the airflow caused a frequency drift, people moving in proximity to the testbed may affect the performance of it. This could explain why there are no outliers for almost 1000 runs in the single gain test in Fig. 5.6 if there was no or low activity in the office at that day. To improve this uncertainty, one could use a reference clock or to shield the testbed.

The results from the 2800 data points with gain value 60 showed that there was approximately one percent of the tests that differed from the otherwise tight grouping. This is considered stable enough for a small-scale production environment. However, since the power analysis showed that the signal is not comparable to an official signal using a passive antenna, it can not be used to test the worst-case scenario in terms of signal power. It is however comparable to a realistic scenario where an active antenna is used, so the testbed can still be used for function verification of the GNSS engine with a realistic scenario.

Also, the tests performed with the power cycling of the modem indicate that the TTFF from a forced cold start might not be directly comparable to the TTFF for a freshly configured modem. However, since testing with the *PLS8-E* modem showed that the TTFF depends on which modem is tested, the exact times are not as interesting as the general behavior.

6.5 Ethical Aspects

There is a low to no risk that faulty devices passes through the implemented test since a malfunction in the testbed would result in no signal or an incorrect one received by the modem. An incorrect signal is identified by the network name check for the LTE and the simulated satellite IDs for the GPS, as described in Section 3.1.1 and 3.1.2. This keeps the risks low that a unit with a malfunctioning cellular modem would be shipped out to a customer.

If the test system malfunctions or is not used as intended, there is a risk that fully-functioning devices fails the test, are mistaken as defect and risk being thrown away. This could happen if the operator connects the wrong RF cable from the testbed to the wrong antenna port in the modem. This needs to be kept in mind by those operating the system. If a device fails the test, the connected cables should be checked and the device should be re-tested at least once to ensure that it is indeed a faulty device before it is rejected.

The software in this testbed implementation is mostly open source projects and the licenses allows commercial use. However, we encourage anyone who might use it commercially, to contribute to the software or support the creators financially.

7
Conclusion

We have shown how a testbed for LTE and GPS in a cellular modem could be implemented in an affordable way. The proposed solution consisted of using an *USRP B205mini-i* SDRs together with open-source software *srsRAN* and *GPS-SDR-SIM* running on a *Rock Pi 4B*. The solution was implemented and testing done on the system indicated that it is feasible to use SDR technology for verifying LTE and GPS functionality.

The LTE test implemented with *srsRAN* is considered fast and stable enough to be suitable for a production environment, and investigation showed that the signal is indeed comparable to an official signal.

The GPS test implemented with *GPS-SDR-SIM* is considered stable, but could use more work before it is deemed fully suitable for a real production environment. The testbed can transmit a GPS signal which the modem reliably can calculate a position from, but since the signal strength is not low enough, it can not simulate a realistic scenario with a passive antenna. It is however comparable to what an active antenna would result in, so the GPS test can be used for realistic function testing but not the worst-case scenario.

7.1 Answers to the Research Questions

The research questions introduced in Section 1.1 are answered and presented below.

- **Is a field-programmable gate array (FPGA) with a radio frequency (RF) peripheral or a software defined radio (SDR) with a general-purpose processor (GPP) best suited as a platform for the implementation of the test system in the terms of cost, reliability, and flexibility?**

  The investigation done in Chapter 3 indicated that SDR is considered most suited for this task in terms of cost and flexibility. The result of the tests presented in Chapter 5 also indicates that it is reliable enough for this task.

- **How can academic research on LTE implementations on the chosen platform be applied for this industrial application?**

  Previous academic research was used as a primary way to determine which platform to choose in Chapter 3. By relying on previous research done on SDRs and *srsRAN* the solution was chosen even if the frequency stability of the SDR was suspected to be insufficient.

- **How can a test system be implemented to verify the functionality of a built-in cellular modem without the presence of a public downlink signal?**

  The investigation presented in Chapter 3 concluded that the solution with an SDR together with *srsRAN* and *GPS-SDR-SIM* running on a GPP can be used for this task with relatively short development time.

- **How does a hardware implementation on the chosen platform meet the test requirements that we will identify, and is it suitable for use in a production environment?**

  The identified requirements were stated in Section 3.1 and the results presented in Chapter 5 shows that the hardware implementation fulfills most of the requirements. The only requirement which was not fulfilled by the testbed was that the signal strength of the GPS signal is not low enough. The LTE test is deemed suitable for a production environment, but the GPS test could use more work to be considered suitable.

7.2 Recommended Future Work

We recommend that a reference clock with a frequency accuracy within the specifications for GPS are included in the testbed. The reason for this is to investigate if it would become more stable and if it allows for lower signal power to be used.

Another future work could be to compare performance with other software for the LTE and GPS implementations, as well as tests with other SDRs and modems. Especially, it would be interesting to test with a *LimeSDR* when they are available, since it could reduce the hardware cost of the testbed.

It could also be interesting to test the GPS implementation with another SDR since it does not require USB 3.0 as the LTE implementation does.

A final recommendation is to build a sturdy encapsulation of the testbed if it is to be used in an actual production environment.

Bibliography

[1] Long term evolution – LTE/LTE-advanced. (accessed: 2021-10-21). Anaritsu. [Online]. Available: https://www.anritsu.com/en-us/test-measurement/technologies/lte

[2] Leasametric. Test & measurement devices. Available at https://www.leasametric.com/en/product/aeroflex-3901/ (accessed: 2021-06-17). Leasametric.

[3] M. Nohrborg. LTE overview. (accessed: 2021-10-11). 3GPP a global initiative. [Online]. Available: https://www.3gpp.org/technologies/keywords-acronyms/98-lte

[4] M. Rumney, "System architecture evolution," in *LTE and the Evolution to 4G Wireless - Design and Measurement Challenges, Second Edition.* Wiley, 07 2013, pp. 195–228.

[5] Y. Chen and X. Lagrange, "Architecture and protocols of EPC-LTE with relay." hal-00830621, 06 2013, p. 25.

[6] F. Firmin. The evolved packet core. (accessed: 2021-10-15). 3GPP a global initiative. [Online]. Available: https://www.3gpp.org/technologies/keywords-acronyms/100-the-evolved-packet-core

[7] Artiza Network. LTE tutorials what is the LTE eNodeB? (accessed: 2021-10-19). Artiza Network. [Online]. Available: https://www.artizanetworks.com/resources/tutorials/what_lteenb.html

[8] Agilent Technologies. (2009) 3GPP long term evolution: System overview, product development, and test challenges. Agilent Technologies. pp.18-24.

[9] S. Prasad, C. Shukla, and R. F. Chisab, "Performance analysis of OFDMA in LTE," in *2012 Third International Conference on Computing, Communication and Networking Technologies (ICCCNT'12)*, 2012, pp. 1–7.

[10] J. Zyren. (2007, 01) Overview of the 3GPP long term evolution physical layer. (accessed: 2022-03-01). [Online]. Available: https://www.nxp.com/docs/en/white-paper/3GPPEVOLUTIONWP.pdf

[11] A. Chadha, N. Satam, and B. Ballal, "Orthogonal frequency division multiplexing and its applications," vol. 2, 01 2013, p. 325.

[12] ABACA. (2014, 05) File:resource-block LTE OFDMA.png. (accessed: 2021-11-15). [Online]. Available: https://commons.wikimedia.org/wiki/File:Resource-Block_LTE_OFDMA.png

[13] L. Kewen and X. Ning, “PAPR reduction of uplink for carrier aggregation in LTE-advanced,” in *The 2nd International Conference on Information Science and Engineering*, 2010, pp. 2224–2226.

[14] C. Ciochina and H. Sari, “A review of OFDMA and single-carrier FDMA,” in *2010 European Wireless Conference (EW)*, 2010, pp. 706–710.

[15] R. Buvaneswaran and S. Srikanth, “Cell search and uplink synchronization in LTE,” in *International Journal of Scientific and Engineering Research*, vol. 4, no. 5, 2013, pp. 1011–1016, ISSN 2229-5518.

[16] Y. Tsai, G. Zhang, D. Grieco, F. Ozluturk, and X. Wang, “Cell search in 3GPP long term evolution systems,” *IEEE Vehicular Technology Magazine*, vol. 2, no. 2, pp. 23–29, 2007.

[17] EUSPA. What is GNSS. (accessed: 2021-11-15). European Union Agency for the Space Programme. [Online]. Available: https://www.euspa.europa.eu/european-space/eu-space-programme/what-gnss

[18] P. Papadimitratos and A. Jovanovic, “GNSS-based positioning: Attacks and countermeasures,” in *MILCOM 2008 - 2008 IEEE Military Communications Conference*. IEEE, 2008, pp. 1 – 7.

[19] GISGeography. How GPS receivers work – trilateration vs triangulation. (accessed: 2021-10-21). GISGeography. [Online]. Available: https://gisgeography.com/trilateration-triangulation-gps/

[20] NXP, “GPS, LNA, sensitivity, jamming, cohabitation, TTFF,” Brochure, May 2009. [Online]. Available: https://www.nxp.com/docs/en/brochure/75016740.pdf

[21] P. Kwan, “NAVSTAR GPS space segment/navigation user segment interfaces,” (accessed: 2022-02-05), Incorporation of IRN-IS-200J-001, El Segundo, CA, 2019. [Online]. Available: https://www.gps.gov/technical/icwg/IS-GPS-200K.pdf

[22] USR IOT. What are cellular modems and what do cellular modems do? (accessed: 2021-10-21). USR IOT. [Online]. Available: https://www.pusr.com/news/what-is-a-cellular-modem-and-what-does-a-cellular-modem-do.html

[23] Quectel. LTE EG25-G. (accessed: 2021-10-21). Quectel. [Online]. Available: https://www.quectel.com/product/lte-eg25-g

[24] TitanWolf. Basic knowledge of SIM (PIN, PUK, IMEI, ICCID, Ki, IMSI, SMSP). (accessed: 2021-10-21). TitanWolf. [Online]. Available: https://titanwolf.org/Network/Articles/Article?AID=71dc9e23-4758-4d43-b368-079d8602e8ed

[25] Diameter Protocol. Usage of OP/OPc and transport key. (accessed: 2021-10-21). Diameter Protocol. [Online]. Available: https://diameter-protocol.blogspot.com/2013/06/usage-of-opopc-and-transport-key.html

[26] National Instruments. Software defined radio: Past, present, and future. (accessed: 2021-11-15). National Instruments. [Online]. Available: https://www.ni.com/sv-se/innovations/white-papers/17/software-defined-radio--past--present--and-future.html

[27] V. K. Garg, “Fourth generation systems and new wireless technologies,” in *Wireless Communications & Networking*, ser. The Morgan Kaufmann Series in Networking, V. K. Garg, Ed. Burlington: Morgan Kaufmann, 2007, pp. 23–1–23–22. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780123735805500570

[28] B. A. Fette, “History and background of cognitive radio technology,” in *Cognitive Radio Technology (Second Edition)*, second edition ed., B. A. Fette, Ed. Oxford: Academic Press, 2009, pp. 1–26. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780123745354000011

[29] Rohde and Schwarz, “Software defined radios,” in *News from Rohde & Schwarz*, Nov 2004, vol. 44, pp. 58 – 61. [Online]. Available: https://cdn.rohde-schwarz.com/pws/dl_downloads/dl_common_library/dl_news_from_rs/182/n182_radiocomunit.pdf

[30] F. Pinto, F. Afghah, R. Radhakrishnan, and W. Edmonson, “Software defined radio implementation of DS-CDMA in inter-satellite communications for small satellites,” in *2015 IEEE International Conference on Wireless for Space and Extreme Environments (WiSEE)*, 2015, pp. 1–6.

[31] Post- och telestyrelsen, “Post- och telestyrelsens författningssamling[The Swedish Post and Telecom Agency constitution],” (accessed: 2021-11-15), Dec 2020, ISSN 1400-187X. [Online]. Available: https://www.pts.se/globalassets/startpage/dokument/legala-dokument/foreskrifter/radio/foreskrifter_undantag_tillstandsplikt_2020.pdf

[32] Software Radio Systems. srsRAN 21.10 documentation. (accessed: 2021-11-15). Software Radio Systems. [Online]. Available: https://docs.srsran.com/en/latest/

[33] OpenAirInterface Software Alliance. OpenAirInterfaceTM (OAI): Towards open cellular ecosystem. (accessed: 2021-11-15). OpenAirInterface software Alliance. [Online]. Available: https://openairinterface.org/getting-started/openairinterface-an-open-cellular-ecosystem/#introduction

[34] B. Wojtowicz. openLTE, an open source 3GPP LTE implementation. (accessed: 2021-11-15). [Online]. Available: https://sourceforge.net/projects/openlte/

[35] Software Radio Systems. COTS UE application note. (accessed: 2021-10-21). Software Radio Systems. [Online]. Available: https://docs.srslte.com/en/latest/app_notes/source/cots_ue/source/index.html

[36] L. Ariza. How to connect OAI eNB (USRP B210) with COTS UE. (accessed: 2021-11-15). [Online]. Available: https://gitlab.eurecom.fr/oai/openairinterface5g/-/wikis/HowToConnectCOTSUEwithOAIeNBNew

[37] Software Radio Systems(SRS). srsRAN/LICENSE. (accessed: 2021-11-15). [Online]. Available: https://github.com/srsran/srsRAN/blob/master/LICENSE

[38] B. Wojtowicz. OpenLTE/LICENSE. (accessed: 2021-11-15). [Online]. Available: https://github.com/mgp25/OpenLTE/blob/master/LICENSE

[39] OpenAirInterface Software Alliance. OAI license model. (accessed: 2021-11-15). OpenAirInterface Software Alliance. [Online]. Available: https://openairinterface.org/legal/oai-license-model/

[40] Software Radio Systems. "raspberry pi 4 application note". (accessed: 2021-10-21). Software Radio Systems. [Online]. Available: https://docs.srsran.com/en/latest/app_notes/source/pi4/source/index.html

[41] F. Gringoli, P. Patras, C. Donato, P. Serrano, and Y. Grunenberger, "Performance assessment of open software platforms for 5G prototyping," *IEEE Wireless Communications*, vol. 25, no. 5, pp. 10–15, 2018.

[42] F. B. F. Spinelli and E. Puig. Technology. (accessed: 2021-11-15). Amarisoft. [Online]. Available: https://www.amarisoft.com/technology/

[43] S. Hassan and A. A. Zekry, "FPGA implementation of LTE-advanced downlink physical layer transceiver," *International Journal of Electronics & Communication Technology (IJECT)*, vol. Vol. 8, Issue 2, April - June 2017, 06 2017.

[44] F. Radu, A. Timofte, A. Balan, and F. Sandu, "LTE communications using an SDR platform," in *2020 13th International Conference on Communications (COMM)*. IEEE, 2020, pp. 393–396.

[45] Digi-Key Electronics. AD-FMCOMMS4-EBZ. (accessed: 2021-11-15). Digi-Key Electronics. [Online]. Available: https://www.digikey.se/product-detail/en/analog-devices-inc./AD-FMCOMMS4-EBZ/AD-FMCOMMS4-EBZ-ND/4754308?utm_adgroup=RF%2FIF%20and%20RFID&utm_source=google&utm_medium=cpc&utm_campaign=Shopping_Supplier_Analog%20Devices&utm_term=&productid=4754308

[46] ——. ZEDBOARD ZYNQ-7000. (accessed: 2021-11-15). Digi-Key Electronics. [Online]. Available: https://www.digikey.se/product-detail/en/digilent-inc/240-122/1286-1225-ND/9841710

[47] The MathWorks, Inc. Get pricing for MATLAB and toolboxes. (accessed: 2021-11-15). The MathWorks, Inc. [Online]. Available: https://se.mathworks.com/campaigns/offers/matlab-toolbox-price-request.html?ef_id=CjwKCAiAp8iMBhAqEiwAJb94z49SoPjn1z1cYXyUfafE1ppwuJ5iwnFklKgJP\kCfEAEHZr1oIH-RnRoCDFAQAvD_BwE:G:s&s_kwcid=AL!8664!3!552170978040!e!!g!!matlab%20price&s_eid=ppc_68036321255&q=matlab%20price

[48] Xilinx. Vivado. (accessed: 2021-11-15). Xilinx. [Online]. Available: https://www.xilinx.com/support/university/vivado.html

[49] Spirent. What is an ephemeris? (accessed: 2022-03-22). Spirent Communications. [Online]. Available: https://www.spirent.com/blogs/2010-08-16_what_is_an_ephemeris

[50] OSQZSS et al. GPS-SDR-SIM. (accessed: 2021-11-15). [Online]. Available: https://github.com/osqzss/gps-sdr-sim

[51] Y. Hu, "GNSS SDR signal generator implementation based on USRP N210," *Journal of Physics: Conference Series*, vol. 1314, p. 012016, 10 2019.

[52] K. K. Songala, S. R. Ammana, H. C. Ramachandruni, and D. S. Achanta, “Simplistic spoofing of GPS enabled smartphone,” in *2020 IEEE International Women in Engineering (WIE) Conference on Electrical and Computer Engineering (WIECON-ECE).* IEEE, 2020, pp. 460–463.

[53] National Instruments Corp. NI GPS simulation toolkit. (accessed: 2021-11-15). National Instruments Corp. [Online]. Available: https://www.ni.com/gate/gb/GB_EKITGPSSIMLTIN/US

[54] K. L. Pedrosa, “A GPS signal generator using a ROACH FPGA board.” Daytona Beach, Florida: Embry-Riddle Aeronautical University, 2017, Master Thesis.

[55] Z. Geng, X. Wei, H. Liu, R. Xu, and K. Zheng, “Performance analysis and comparison of GPP-based SDR systems,” in *2017 7th IEEE International Symposium on Microwave, Antenna, Propagation, and EMC Technologies (MAPE)*, Xi’an, China, 2017, pp. 124–129.

[56] S. Tripathi, C. Puligheddu, C. F. Chiasserini, and F. Mungari, “A context-aware radio resource management in heterogeneous virtual RANs,” *IEEE Transactions on Cognitive Communications and Networking*, p. 1–1, 2021. [Online]. Available: http://dx.doi.org/10.1109/TCCN.2021.3115098

[57] Software Radio Systems. Installation guide. (accessed: 2021-11-16). Software Radio Systems. [Online]. Available: https://docs.srslte.com/en/latest/general/source/1_installation.html#gen-installation

[58] ——. srsRAN on raspberry pi 4. (accessed: 2021-11-15). Software Radio Systems. [Online]. Available: https://docs.srslte.com/en/latest/app_notes/source/pi4/source/index.html

[59] Seeed The IOT hardware enabler. Rock pi 4 model b 4GB. (accessed: 2021-11-15). Seeed Technology Co. [Online]. Available: https://www.seeedstudio.com/ROCK-Pi-4-Model-B-4GB-p-4137.html

[60] M. Surligas. (2020) Supported devices. (accessed: 2021-10-21). GitLab. [Online]. Available: https://gitlab.com/librespacefoundation/gr-soapy/-/wikis/supported-devices

[61] I. Chiu. Superspeed USB 3.0 FAQ. (accessed: 2022-02-05). Everything USB. [Online]. Available: https://www.everythingusb.com/superspeed-usb.html

[62] A. Puschmann, “LimeSDR USB + srsENB fails to start #562,” (accessed: 2021-10-21), 2020. [Online]. Available: https://github.com/srsran/srsRAN/issues/562

[63] LabSat. GNSS frequency guide. (accessed: 2021-11-15). LabSat. [Online]. Available: https://www.labsat.co.uk/index.php/en/gnss-frequency-guide

[64] Ettus Research. USRP hardware driver and USRP manual, power level controls. (accessed: 2021-10-21). Ettus Research. [Online]. Available: https://files.ettus.com/manual/page_power.html

[65] H. Welte, “The limits of general purpose SDR devices.” Mildenberg Brick Work Park: Chaos Communication Camp 2019, 08, days 21-25. [Online].

Available: https://media.ccc.de/v/Camp2019-10248-the_limits_of_general_purpose_sdr_devices

[66] S. Fischer. Introduction to OTDOA on LTE networks. (accessed: 2021-10-21). Qualcomm technologies Inc. [Online]. Available: https://www.qualcomm.com/media/documents/files/introduction-to-otdoa-on-lte-networks-highlights.pdf?fbclid=IwAR0FbH_8y95GX5r_medIyDmXHkfI0o7cY403Aonys3ZjFCshfYn-JEeSiXs

[67] Electronics Notes. OCXO, oven controlled crystal oscillator. (accessed: 2021-12-2). Electronics Notes. [Online]. Available: https://www.electronics-notes.com/articles/electronic_components/quartz-crystal-xtal/ocxo-oven-controlled-crystal-xtal-oscillator.php

[68] E. Research. B200/B210/B200mini/B205mini. (accessed: 2021-11-15). Ettus Research. [Online]. Available: https://kb.ettus.com/B200/B210/B200mini/B205mini

[69] Ettus Research. Comparative features list - B200/B210/B200mini. (accessed: 2022-01-17). Ettus Research. [Online]. Available: https://files.ettus.com/manual/page_usrp_b200.html

[70] Quectel. EG25-G hardware design. (accessed: 2021-11-15). Quectel. [Online]. Available: https://www.quectel.com/download/quectel_eg25-g_hardware_design_v1-4

[71] R. Subramanian, K. Sandrasegaran, and X. Kong, "Benchmarking of real-time LTE network in dynamic environment," in *2016 22nd Asia-Pacific Conference on Communications (APCC)*, Yogyakarta, Indonesia, 2016, pp. 20–25.

[72] Commscope, Inc. (2018) Understanding the RF path. (accessed: 2021-11-15). Commscope, Inc. [Online]. Available: https://www.commscope.com/globalassets/digizuite/3221-rf-path-ebook-eb-112900-en.pdf

[73] D. Nechita et al. IIO oscilloscope. (accessed: 2021-11-15). [Online]. Available: https://github.com/analogdevicesinc/iio-oscilloscope

[74] Armbian. Armbian documentation - quick facts. (2022-01-04). [Online]. Available: https://docs.armbian.com/Quick_facts/

[75] ——. Rockpi 4 A / B / C. (2022-01-04). Armbian. [Online]. Available: https://www.armbian.com/rock-pi-4/

[76] Open Cells Project. Open-cells project - SIM cards. (2022-01-04). Open Cells Project. [Online]. Available: https://open-cells.com/index.php/sim-cards/

[77] Crustal Dynamics Data Information System (CDDIS DAAC). (2022, 03) International GNSS service, broadcast ephemeris data, GPS daily. (2022-01-04). [Online]. Available: https://cddis.nasa.gov/archive/gnss/data/daily/

[78] Radxa Team, "Introduce the new rock pi 4 - hardware," (2022-01-04), Radxa Limited, Shenzhen, China, 11 2018. [Online]. Available: https://wiki.radxa.com/News/2018/11/introduce-the-new-rockpi-4-hardware

[79] Adafruit. GPS antenna - external active antenna - 3-5V 28dB 5 meter SMA. (accessed: 2022-01-21). Adafruit. [Online]. Available: https://www.adafruit.com/product/960

A

Hardware Components

	Name	Quantity	Approximate Unit price [SEK]	Approximate Total cost [SEK]
SDR	USRP B205i-mini	2	7,200	14,400
GPP	Rock Pi 4B	1	1,000	1,000
SD Card	Samsung Evo 32GB	1	130	130
30dB Attenuator	Huber+Shuner 6630_SMA-50-2/199_N	7	760	5,300
DC Blocker	Mini-Circuits BLK-89-S+	2	180	180
Power Divider/Combiner	Instock Wireless PD2120	1	430	430
SMA RF Cable 10cm	Crystek Microwave CCSMA-MM-086-4	3	100	300
SMA RF Cable 100cm	Johnson / Cinch 415-0029-M1.0	2	120	120
SIM Card	Open-Cells SIM Card	1	50	50