Martin Erb, BSc

A NFC Protocol Analyzer based on the RedPitaya Platform

MASTER'S THESIS

to achieve the university degree of
Master of Science
Master's degree program: Information and Computer Engineering

submitted to

Graz University of Technology

Supervisor

Dipl.-Ing. Dr.techn.
Quaritsch Markus

Institute Director
Univ.-Prof.Dipl.-Inform. Dr.sc.ETH Kay Uwe Römer
Institute for Technical Informatics

Graz, December 2015

Kurzfassung

Die der RFID (Identifizierung mit Hilfe elektromagnetischer Wellen) zugrunde liegende Technologie ist relativ alt. In den Bereichen Transport und Logistik, Zugangskontrolle, Überwachung von Personen und Tieren, kontaktlose Zahlung, E-cards, Datenübertragung uvm. wird sie zunehmend vermehrt eingesetzt. Jeder Anwendungsfall hat unterschiedliche Anforderungen an die Technologie. Je nach Einsatzgebiet werden verschiedene Ansprüche gestellt; wie größere Kommunikationsreichweite, strengere Sicherheitsbestimmungen, höhere Übertragungsrate, niedrigerer Energieverbrauch usw. Um zu überprüfen, ob neu entwickelte Produkte diesen Anforderungen entsprechen, werden verschiedene Messinstrumente benötigt. Diese sollen auch Kompatibilität und Einhaltung aktueller Standards garantieren. Die RFID Technologie wird zum Beispiel bei "PayPass" angewendet und erlaubt es dem Nutzer bis zu 25 Euro kontaktlos zu bezahlen, indem er die Bankomatkarte über das Lesegerät hält. Es wird versucht diese Anwendung in Smart Phones zu integrieren. Um zu garantieren, dass jede Bankomatkarte und jedes Smart Phone weltweit für kontaktloses Bezahlen genutzt werden kann, müssen Messinstrumente und Software entwickelt werden. Diese überprüfen, ob das neue Produkt den standardisierten Protokollen und Tests entspricht. Diese Masterarbeit erklärt, wie es möglich ist, die Kommunikation zwischen Bankomatkarte und Lesegerät aufzuzeichnen. Signalabtastung, Signalfilterung, Signaldekodierung und Protokollanalyse werden im Detail erklärt. Es ist damit allerdings nicht möglich, sensible Data zu extrahieren da die Kommunikation kryptografisch verschlüsselt ist. Verschiedene Probleme, die bei der Entwicklung derartiger Messsysteme auftreten, werden analysiert und Lösungen aufgezeigt. Das Hauptaugenmerk wird auf die Möglichkeit der Entwicklung eines 'Protocolanalyzer' gelegt unter Verwendung der neuen Plattform RedPitaya.

Abstract

The technology behind Radio-Frequency Identification (RFID) is relatively old but the usage has enormously increased in the last couple of years. It is used for tracking persons and animals, access control, transportation and logistics, passports, contactless payment, e-cards, data transmission, and much more. Each use-case has different requirements of the technology. Some of them need a wider communication range, stricter security policy, higher transmission rate, low power consumption etc. To develop new products which accomplish these requirements and to ensure interoperability between different devices measurement instruments are needed. Such instruments are also needed to check if the developed devices satisfy the given standards. One of the newest technologies is the so called "PayPass" where the user can pay up to 25 Euros by tapping his payment card on a terminal reader. Many companies try to port this technique to the smart phone. To ensure that each payment card and each smart phone is compatible with terminal readers all over the world, organizations developed standardized protocols and tests which now have to be checked by measurement instruments and software. This thesis shows how it is possible to use an additional passive participant to capture the communication between payment card and reader. It explains in detail the sampling and filtering process, the decoding task and the protocol analyzing software. It is not possible to extract any sensitive data because the communication is cryptographically encoded. The main purpose of this research is to try to develop a protocol analyzer using the new platform RedPitaya.

Statutory Declaration

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly indicated all material which has been quoted either literally or by content from the sources used. The text document uploaded to TUGRAZonline is identical to the present master‘s thesis dissertation.

Graz, _____ _____

Date Signature

Contents

1. Introduction

The increasing interest in the use of Near Field Communication (NFC) technology is rapidly growing in the area of payment systems sector. The possibility to pay by tapping the payment card or smart phone on a terminal reader (Proximity Coupling Devices (PCD)) is becoming increasingly popular in our society. In the Eurozone it is possible to pay up to 25 Euros with contactless technology if the payment card and the terminal reader provide this new functionality. Various credit providers, terminal reader and smart phone producers want to sell their products with this new feature. This leads to new problems this master thesis deals with. Are the different payment cards and smart phones compatible with all different terminal readers? Do the payment cards and the terminal reader support the needed ISO standards [5] [9] [13]? To answer those question expensive hardware and software is needed to run the interoperabilitytests. The motivation of this research is to develop a better and cheaper tool to run those test using the new platform RedPitaya.
The next few chapters will show the differences between hardware and software of the NFC-sniffer used by companies, in this particular case CISC Semiconductor and which I developed [2].

1.1. Overview

This section provides a brief overview of how communication between reader terminal and smart card or smart phone is captured and analyzed. Figure 1.1 shows the measurement setup. The main part of the setup is the RedPitaya platform shown in the middle of figure 1.1. This platform is described in section 2.2 in more detail [17]. The terminal reader is connected to the PC via Universal Serial Bus (USB) and is responsible for communication with the smart card. The RedPitaya is connected to the PC via

Ethernet to start the sample acquisition and to transmit the captured signal to the PC. To measure the communication signal a sniffer coil is connected via a coaxial cable to the second input channel of the RedPitaya. To complete the measurement setup a smart card is placed on top of the sniffer coil as shown in figure 1.1. The software running on the PC is also an important part of the measurement setup enabling the whole process to be controlled. Figure 1.2 shows the main screen of the software. The button "Start acquisition" starts the recording process on the RedPitaya and the analyzing task on the PC as shown in the flow diagram 1.3. The communication between terminal reader and smart card can be started by using the corresponding driver software for the reader. The software on the RedPitaya saves the captured signal and sends it to the PC afterwards where it will analyzed, decoded and displayed for the user. With the software on the PC the user can save the captured communication and load it in the future on demand.

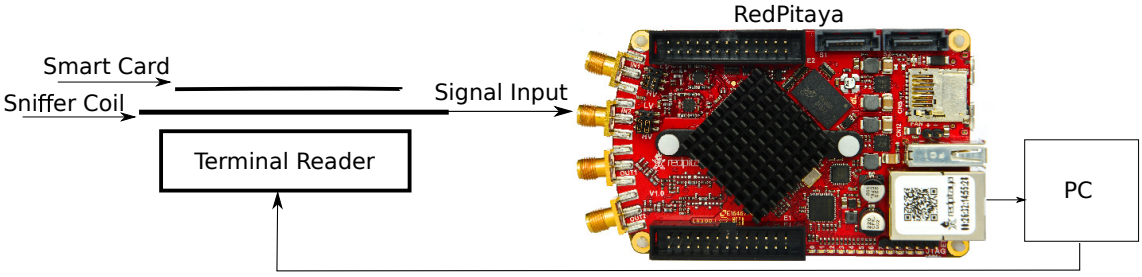


Figure 1.1. – Measurement setup

1.2. Outline

In chapter 2, some background information about the most important parts of a NFC-sniffer system are explained for a better understanding of the ongoing topics. Chapter 3 presents commercial solutions which act as model for the implementation of the NFC-sniffer developed in this master thesis. Chapter 4 shows the main problems this research had to deal with. The structure of the software and how this cooperates with the hardware is described in chapter 5. Chapter 6 contains more information about the hardware used and compares it with the commercial hardware. Chapter 7 explains the implemented software for the RedPitaya and the PC. These two chapters contain the explanation of most problems incurred in this project. The next chapter 8,

shows the evaluation of the implemented system in comparison to the professional system used by CISC. The implemented system is not complete and therefore chapter 9 gives some possible ideas for further improvement. Chapter 10 provides the conclusion to this master thesis.

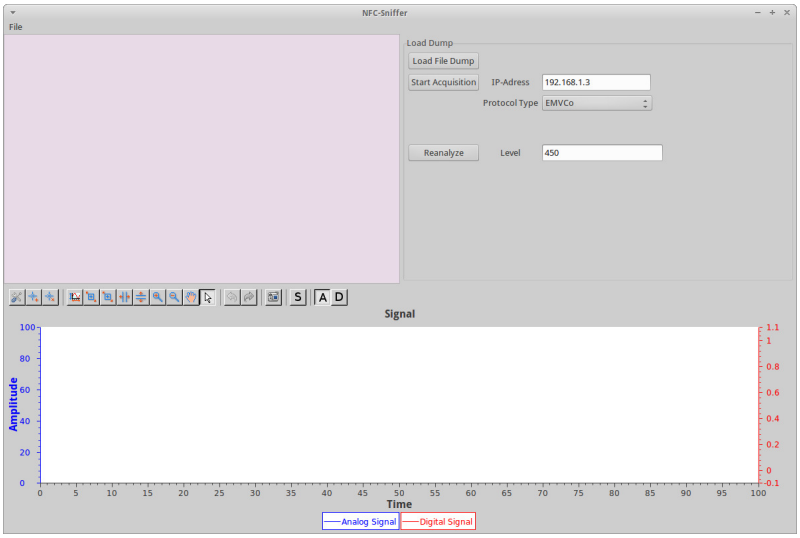


Figure 1.2. – Software start screen

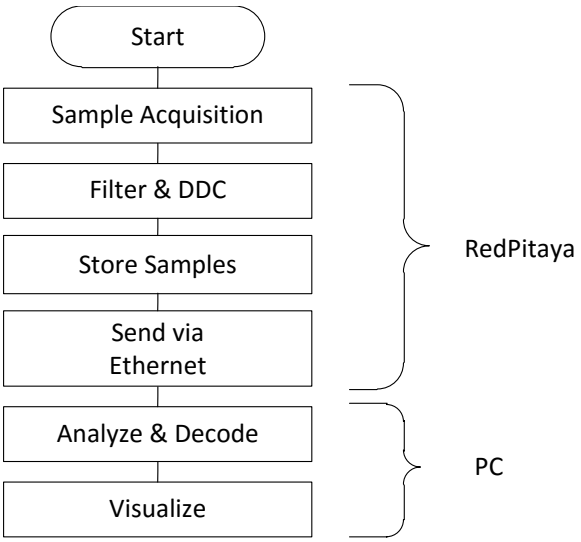


Figure 1.3. – Work-flow

2. Background

This chapter gives detailed background information about the most important hardware, software and protocol parts of the NFC-sniffer.

2.1. Near Field Communication

There exist a lot of different standards around the Radio-Frequency Identification (RFID) technology. Three of them are interesting talking about NFC and the popular frequency of 13.56 MHz [8]. The first standard ISO/IEC 18092 defines the peer-2-peer mode. With this operation mode two devices can establish a bidirectional connection and exchange data. The vicinity cards (according to ISO/IEC 15693) and the Proximity Integrated Circuit Cards (PICC) (according to ISO/IEC 14443) are the other two interesting standards. The vicinity cards can operate up to a distance of 1.5 m with a data rate of 10 kBit/s. The PICC can only operate within a distance of 15cm with a data rate of 106 kBit/s. During the standardization of ISO/IEC 14443 the major player Infineon and NXP could not agree on the modulation schema and therefore now exist ISO/IEC 14443-A (NXP) and ISO/IEC 14443-B (Infineon). NXP uses the easier Amplitude Shift Keying (ASK) and Infineon the more complex Phase Shift Keying (PSK).
The NFC Forum, an organization formed in 2004 is responsible to certificate tags which are compliant to certain standards to ensure interoperability between these devices. The NFC Forum has agreed on four different tag types.

Type 1 Type 1 is based on ISO/IEC 14443A. The communication speed of these tags is 106 kBit/s and the memory size is 96 Byte. An example of this tag type is Innovision Topaz.

Type 2 This type is based on the same standard as type 1 and also the communication speed is the same. The memory size can be either 48 Byte or 144 Byte. Examples for this type are: NXP Mifare Ultralight and NXP Mifare Ultralight C.

Type 3 The so called FeliCa tags from Sony are based on a Japanese standard and have a memory size between 1 kByte and 9 kByte. The communication speed can either be 212 kBit/s or 424 kBit/s.

Type 4 The last type is fully compatible with the ISO/IEC 14443A /& B standard. The memory size can be up to 32 kByte and the communication rate is 106 kBit/s. Tags with this type are commonly used to store applications on them based on ISO/IEC 7816. This standard defines the application layer that can be used by other associations (e.g. EMVCo) to implement their functionality.

The NFC-sniffer developed in this master thesis deals with the ISO/IEC 14443A standard and the EMVCo application layer.

2.2. RedPitaya

RedPitaya is a young spin-off company from Instrumentation Technologies [1] and was presented the first time through a Kickstarter campaign [2] in September 2013 [17]. The company does not want to produce a professional measurement system for one purpose but a product that can be used in a wide range at a high technical level. The Field Programmable Gate Array (FPGA) is a Xilinx Zynq 7010 System on Chip (SoC) with a Dual ARM Cortex-A9 CPU and 512 MByte RAM. The RedPitaya platform provides 16 digital input/output channels, four analog input channels, four analog output channels, two Radio Frequency (RF) inputs and two RF outputs [16]. RedPitaya provides a Beta Linux as Operating System (OS) for the platform but some community members have already ported a more comfortable Debian Linux.
There are three possible user interfaces to control the RedPitaya: The first one is the web interface enabling the possibility to install application from the web and control

[1] http://www.i-tech.si/
[2] https://www.kickstarter.com/projects/652945597/red-pitaya-open-instruments-for-everyone

them. Some examples of those applications are the oscilloscope, the LCR-meter, the spectrum analyzer and many more. With this application interfaces it is only possible to control the web application programmed for this platform. The other two interfaces are almost the same and are SSH connections, one via TCP/IP the other via the second micro USB connector beside the power supply. With these two interfaces it is possible to control the OS and the programs running on it. The Ethernet is also used to load custom FPGA software on the RedPitaya and than on the FPGA over a preconfigured char device provided by the OS.

2.3. RFID Technology

The RFID technology allows to read and write information from a tag by using RF. In the specific case of payment systems the passive actor is the payment card (PICC) because it is only readable by the active actor represented by the terminal reader. Since the passive actor has no power supply the active actor has to provide the energy. This is done via an electromagnetic field as shown in figure 2.1. The system, PCD and PICC behave like a transformer. The alternating current passes through the primary coil of the PCD and creates an electromagnetic field. This electromagnetic field induces a voltage in the secondary coil which is the antenna of the PICC. The PICC converts the induced voltage into a DC voltage to power the internal circuit of the PICC. This technology operates at 13.56 MHz and the communication between PCD and PICC should work within the operating volume shown in figure 2.2. The RF energy transmitted by the PCD is not only used to power the PICC but also to transport data through the modulated carrier. The PCD uses a 100% ASK modulation to send data to the PICC. The PICC uses On-Off keying (OOK) load modulation to send the requested information back. The functionality of these two modulation methods are described in more detail in section 2.5.

2.4. Field Programmable Gate Array

FPGAs are semiconductor devices where configurable logic blocks are arranged around a matrix and can be connected with programmable connections. An FPGA can be

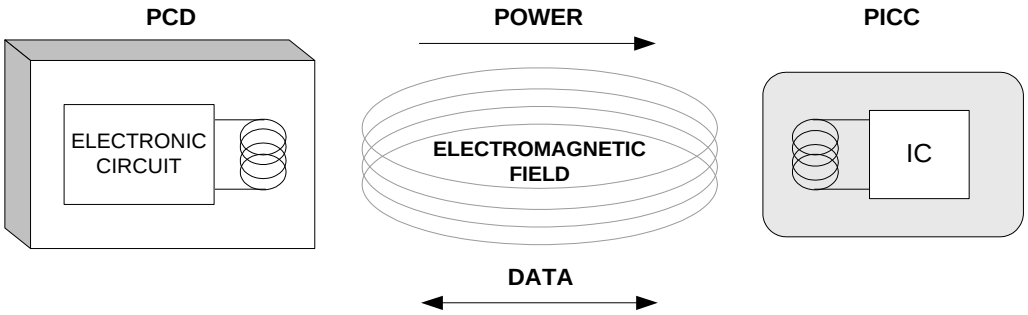


Figure 2.1. – PCD and PICC Configuration [5]

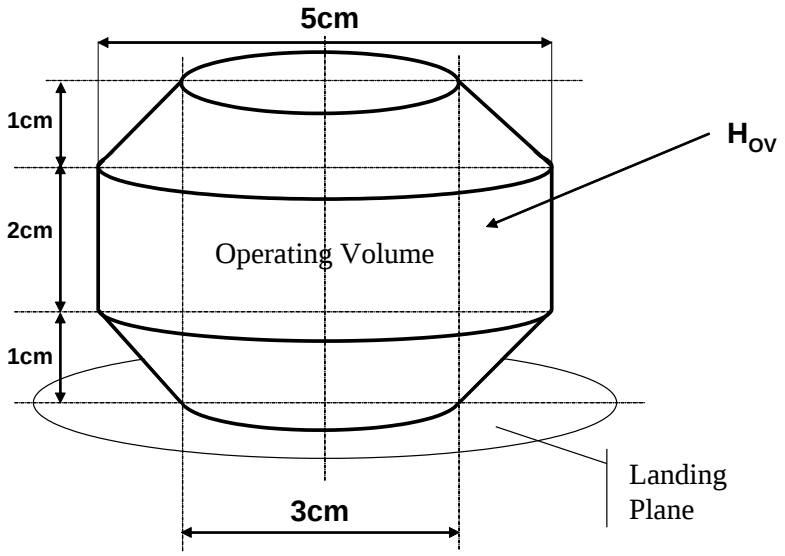


Figure 2.2. – Operating Volume according to [5].

reprogrammed to a specific application or functionality after manufacturing. This is the main difference between FPGA and Application Specific Integrated Circuit (ASIC). To program an FPGA a hardware description language is used: e.g. vhdl, verilog. FPGAs have the advantage of changing design very late in the design cycle. In comparison to ASIC the design cycle is simpler because the software handles the routing, placement and timing. FPGA may also have some incorporated hard blocks of commonly used functionalities such as RAM, Digital Signal Processor (DSP) and clock management. The DSP can be used to handle complex signal processing tasks. On the other hand the basic components of an FPGA are:

CLB The Configurable Logic Block (CLB) is the fundamental building block of an FPGA and is usually laid out within a logic block array. The number of CLBs on an FPGA vary from device to device but they all consist of a configurable switch

matrix with four or six inputs, some selection circuits such as multiplexer (MUX) and flip-flops. The switch matrix can be configured to handle combinatorial logic, shift register or RAM.

Interconnect The CLB described earlier, provides the logic capability and the programmable interconnect routing routes the signal between CLBs and between CLBs and Input/Output (I/O)s. The interconnect routing task is invisible for the user because the design software handles the whole process. This reduces the complexity during the design cycle enormously.

Select I/O Todays FPGAs provide a wide spectrum of I/O standards providing a perfect interface for different systems. I/Os in FPGAs are grouped in banks, where each bank can support different I/O standards. Todays FPGAs provide over a dozen of those I/O banks. RedPitaya also has a lot of different I/Os accessible though the following I/O-banks: RF-I/O, auxiliary analog I/O-channels, general purpose digital I/O, Serial Advanced Technology Attachment (SATA)-type connector and Insulation Displacement Connector (IDC).

Memory Embedded Block RAM (BRAM) is available in most FPGAs for on-chip memory. Xilinx FPGAs provide up to 10 MBits of this memory in 36 kBits blocks [18].

2.5. Amplitude Modulation and Encoding

The ISO/IEC 14443 standard allows two different modulation types for communicating between PCD and PICC, called Type A and Type B. In this master thesis only Type A is discussed. For the communication from PCD to PICC Amplitude Shift Keying is used. The amplitude of the carrier is switched between two voltage levels. The level of the lower voltage is defined by the percentage number of the modulation. Type A uses a 100% ASK meaning that the lower voltage is nearly zero.
For communication from PICC to PCD Type A uses load modulation. By switching load on and off a subcarrier with a frequency of about 847 kHz is created. The information is modulated on the subcarrier by using On-Off keying and will be transmitted with about 106 kBit/s. The modulated signal causes a different current flow through the antenna of the PICC. When the PICC is not sending the load is switched off. As a consequence the different current flow through the antenna has a different induced

voltage in the PCD antenna and so the PCD can decode the sent information.
The two communication directions use different bit codings, from PCD to PICC modified Miller with ASK 100% and Manchester with On-Off keying from PICC to PCD. Figure 2.3 and figure 2.4 show the two different modulation types with the corresponding bit codings. The two codings allows to use a counting method to verify the conformity of the number of bits in a received frame. The main advantage is the possibility to detect transmission errors in the lower layers without the need of parity bits or Cyclic Redundancy Check (CRC).

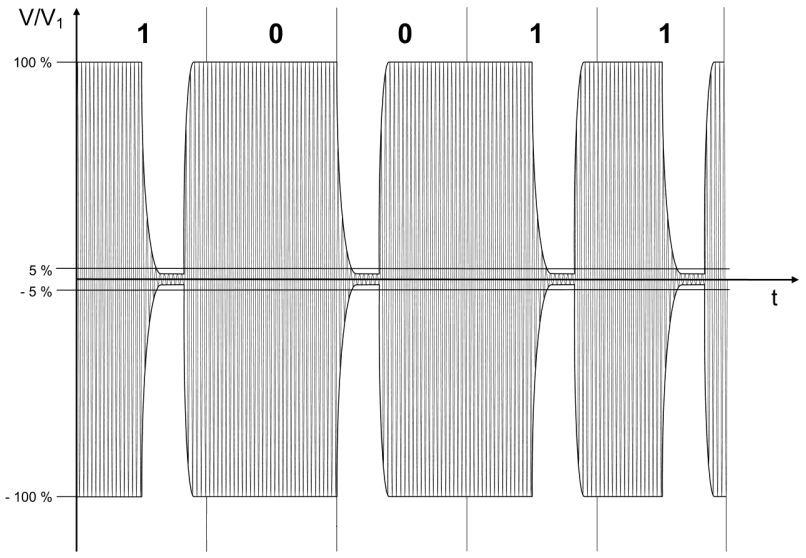


Figure 2.3. – Modified Miller Coding with 100% ASK [5].

2.6. Sampling

An important part of the measurement system shown in figure 1.1 is the sample acquisition. As described in section 4.2 there are two usable methods to represent a signal, the serial samples and the I/Q-samples. The NFC-sniffer, talked about in this master thesis and commercial systems use the I/Q-samples. The main differences and advantages are described in more detail in this section and shown in figure 2.5 and 2.6. It is observable, that the I/Q-data plot has much more information about the signal than the series data plot. Both figures show the amplitude of the signal but only the I/Q-data plot also has information about the phase of the signal. The I-data is called "in-phase" and represents the real component of the signal. The Q-data, also called

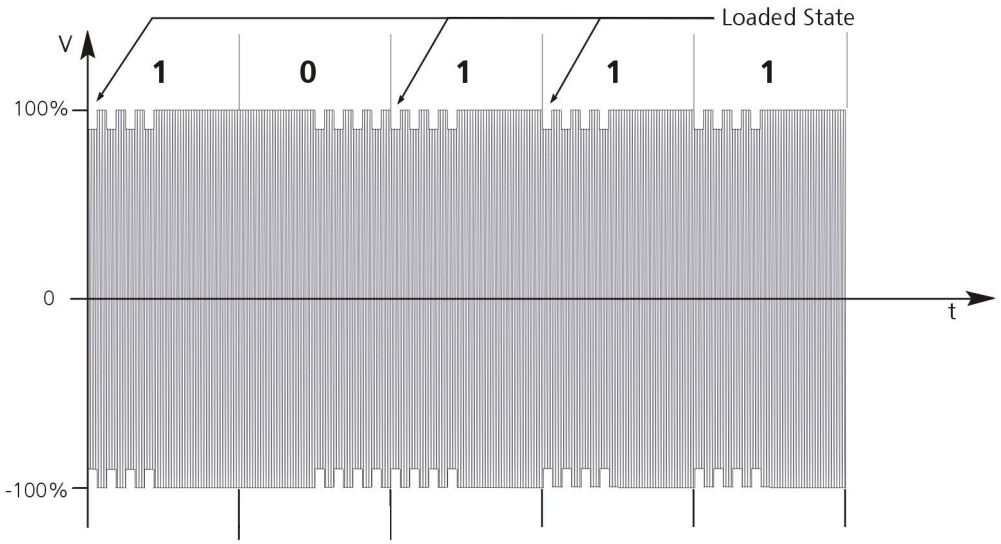


Figure 2.4. – Manchester Coding with OOK [5].

"quadrature", represents the imaginary component and is the real signal shifted by -90°. If the signal is a single sine curve the Q-data is the composite of the I-data but if the signal is more complex and has more than one sine component the Q-data shows the shifted, individual components. The following equations show the formulas to calculate some important parameters of the I/Q-data [14][11].

$$\text{Peak Amplitude} \ldots A = \sqrt{(I^2 + Q^2)} \tag{2.1}$$

$$\text{Phase Angle} \ldots \Phi = \tan^{-1} \frac{I}{Q} \tag{2.2}$$

$$I = A * \cos(\omega * t) \tag{2.3}$$

$$Q = A * \sin(\omega * t) \tag{2.4}$$

In systems like the NFC-sniffer it is highly recommended to do the I/Q-data sampling with an FPGA. There are many algorithms for FPGA to calculate hyperbolic and trigonometric functions if no hardware multiplier is available. In this system the Coordinate Rotation Digital Computer (CORDIC) algorithm is used [1]. The only operations this algorithm requires to compute the needed trigonometric functions are addition, subtraction, bit shift and table lookup. To get the in-phase samples equation 2.3 is needed and for the quadrature samples equation 2.4.
To reduce the amount of digitalized samples a Digital Down Converter (DDC) is used.

The DDC consists of a Cascade Integrator Comb (CIC) filter to convert a high sampling rate into a lower one and a Finite Impulse Response (FIR) filter to limit the bandwidth. The specific values of these filters and how they are used in this sniffer system are described in chapter 6.

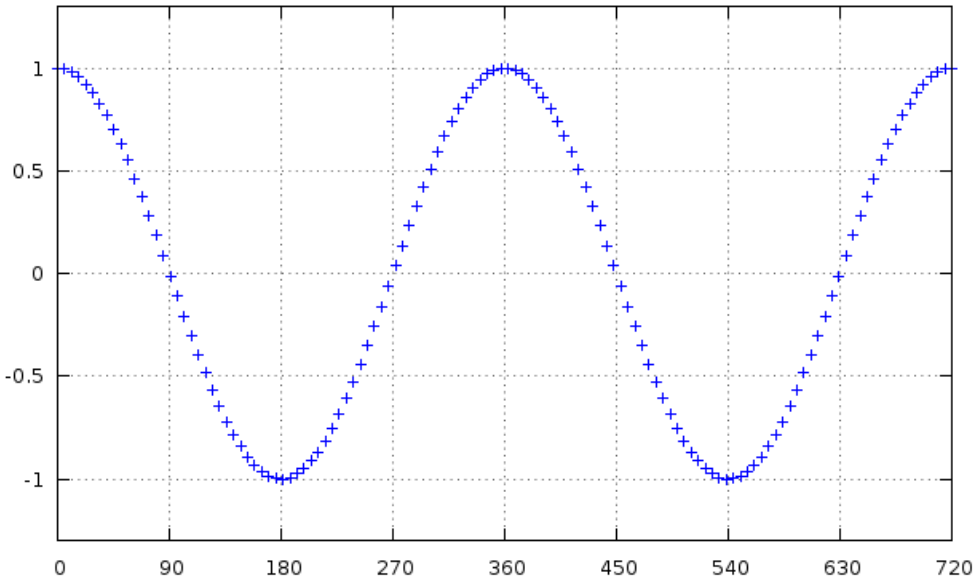


Figure 2.5. – Series Samples [14]

2.7. Protocol

The basic protocol implemented by the NFC-devices is the ISO/IEC 14443 standard [9]. This standard defines the different frequencies, the modulation types, the different bit codings, how the CRC is computed and how the different frames have to look. Furthermore the state machine demonstrating how the connection is established and the commands up to the application layer are defined in this standard. The flow chart shown in figure 2.7 depicts the commands involved to start a communication between PCD and PICC. For the application layer this standard provides the R/S/I-blocks. Each different application has to independently define how the data packed in these blocks, must be interpreted. The I-block is used to convey information for the application layer. The R-block is used to convey positive or negative acknowledgements and the S-block to deselect the card or to expand the waiting time on the PCD for the next answer.

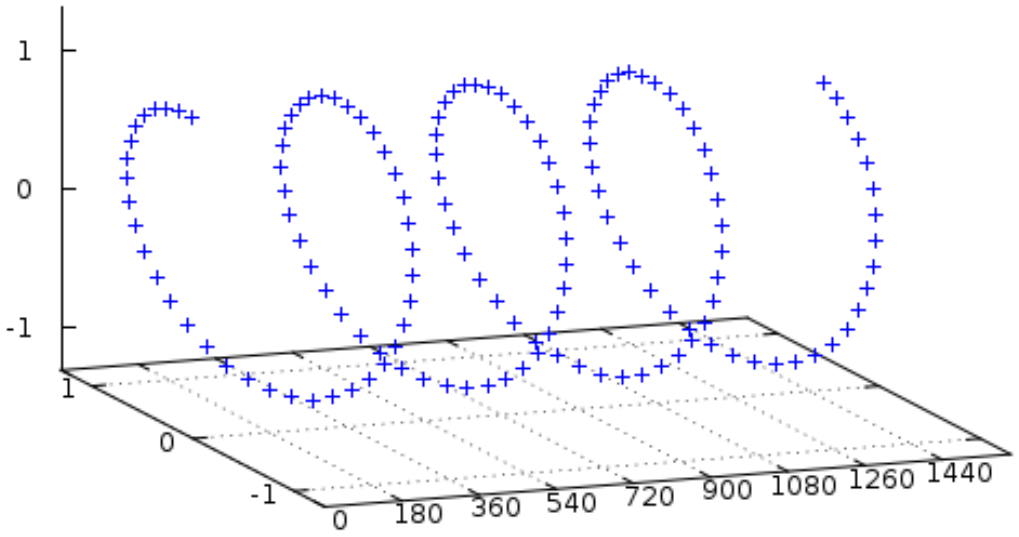


Figure 2.6. – I/Q Samples [14]

The communication between terminal reader and payment card works as follows: The terminal reader powers its antenna and starts sending the “wake-up” command (REQA, WUPA). The communication starts when the payment card taps the terminal reader. The PICC starts in IDLE-state and sends the acknowledge (ATQA) to the terminal reader when the “wake-up” command is registered. After this wake-up process the PICC is in READY-state and the anticollision process starts. In this process the terminal reader checks if only one card is in its electromagnetic field range and answers to the requests. Within this process the PICC has to send his Unique Identifier (UID). This transmission can take up to three request-response cycles if the UID has the maximum possible length. If the anticollision process is finished and only one payment card is within the operating volume, the terminal reader selects this PICC and the payment card switches to the ACTIVE-state. At this moment the application layer starts and only R/S/I-blocks will be transmitted. The NFC-sniffer this master thesis is talking about handles only the EMVCo application protocol whose data is packed into the ISO7816 standard [13]. The EMVCo application layer selects the right application with the PPSE command and afterwards it starts to read the needed information. At the end the terminal reader deselects the card and stops powering the antenna to clear the electromagnetic field.

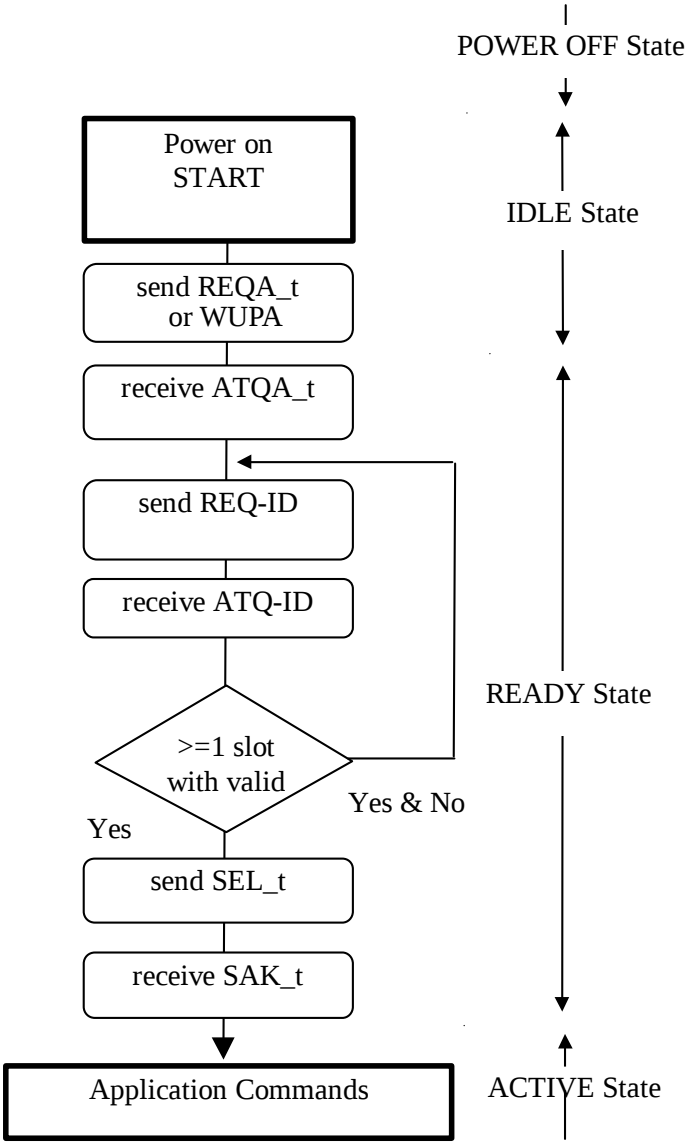


Figure 2.7. – Protocol flow chart [9]

3. State of the Art

Thanks to the increasing interest in the use of RFID technology many companies were keen to develop, maintain, sell and test new NFC tags and read/write terminals. The first established companies developed their own product for a specific purpose but nowadays there are various standards. These standards can be assigned to three categories: Low Frequency (LF) from 30 kHz to 300 kHz, High Frequency (HF) from 3 MHz to 30 MHz and Ultra High Frequency (UHF) from 300 MHz to 3 GHz. For each category different hardware and software tools are needed for developing and testing. During development processes different parameters are important than in the testing phase. To cover the whole spectrum many companies, with the focus on a category or on a specific standard within one of those three frequency ranges are needed. This chapter shows some companies, their focus and tools they use. The tools used by CISC, the partner company supporting this research are described in more detail at the end of this chapter.

3.1. Voyantic

The company Voyantic[3] is specialized in RFID measurement and testing. The developed solutions serve the industrial and academic market. With the hardware and software tools provided by this company it is possible to measure and test tags and readers which operate in the HF band and in the UHF band. With the various tools the effect of different tag materials, tag shapes, the performance, the signal strength during a power sweep on selected frequencies, the optimal placement for a tag on a reader and the protocol can be studied. Some developed systems are also applicable for Integrated Circuit (IC) functional testing, RFID application testing and conformance

[3] https://voyantic.com/

testing in both frequency bands with many different protocol standards. The tools of the company should reduce the time to market, speed up the development of tags and use standardized methods during the measuring and testing process. The developed academic hardware and software helps to gain a better knowledge of HF and UHF RFID tags.

3.2. At4Wireless

At4Wireless[4] provides certification and testing services of all telecommunication devices. This company has also developed complete solutions of localization, identification, management, control, monitoring and automation systems. Some of the supported wireless technologies are: Global System for Mobile Communications (GSM), General Packed Radio Service (GPRS), Wideband Code Division Multiple Access (WCDMA), High Speed Packed Access (HSPA), Long Term Evolution (LTE), Worldwide Interoperability for Microwave Access (WiMAX), Bluetooth, Wi-Fi, RFID, NFC, and many more. Regarding the RFID and NFC technology in mobile phones, At4Wireless performs the following tests: conformance, performance, electrical safety, interoperability and field trials for the standards defined by the NFC Forum.

3.3. Arsenal Testhouse

The four different main services provided by the company Arsenal Testhouse[5] are: Mifare training, Mifare certification, test cards and readers for contactless RFID communication and developing new products regarding contactless testing. The supported application layers are EMVCo, Mifare and NFC-Forum. Arsenal Testhouse further provides test tools according to international standards such as ISO/IEC 14443A/B, ISO/IEC 15693, ISO/IEC 18000-3, ISO/IEC 18092, ICAO e-passport and ECMA-340. The test assemblies and reference PICC can also be ordered. The certification service includes the Mifare classic/ultralight, Mifare desfire and Mifare plus standards.

[4] www.at4wireless.com/
[5] http://www.arsenal-testhouse.com/

3.4. Keysight(Agilent)

The company Keysight[6] is a well known manufacturer of oscilloscopes but also produces NFC conformance test system. This system supports analog RF and digital protocol parts of NFC, EMVCo and ISO test specifications, especially the EMV Level 1, NFC Forum and ISO/IEC standards 14443, 18092, 15693 and 18000-3. FIME[7] developed some of the used test cases providing numerous testbenches for contactless payment systems. Using this software it is not possible to show the single commands of the communication but it is deft in analyzing the analog curve. It is important, that the generated signal of the two communicating partners is within the specified limitations regarding the timing, rise time, fall time and modulation index.

3.5. KEOLABS

The company KEOLABS[8] has developed a system facilitating the running of conformance tests of smart cards and card readers in accordance with industry standards. This system enables communication and analysis of most protocols based on the ISO/IEC 14443 standard beginning from the analog signal up to the application layer. Furthermore the system can undertake the communication part of the reader or the smart card in the form of an emulator. This feature allows testing a smart card or reader with a predefined behavior of the communication partner. The captured signal between PICC and PCD is visualized as a tree view with much additional information. The illustration of the analog and digital signal shows the decoded command name, the hexadecimal values of the data split in transport layer and application layer. This view is extendable with additional filters and representation modes. It is possible to display only the application layer or transport layer for better understanding. With the automatic test program different tests can be loaded and run automatically, generating a personalized report.

[6] www.keysight.com
[7] www.fime.com/
[8] www.keolabs.com/

3.6. CISC Semiconductor

CISC [2] serves products and engineering services to the semiconductor, automotive, wireless communication and RFID market. CISC was founded in 1999 with its headquarters in Klagenfurt. CISC, the supporting company of this master thesis is operating as an R&D office in Graz since 2007. Some products regarding RFID technology are:

RFID Xplorer The RFID Xplorer is a compact and easy to use RFID test instrument for the UHF band. The main features are performance and conformance tests but also the sniffer function to analyze the communication signal is available. The captured complex waveforms can be saved on the hard-drive to make them available for further analysis in the time and frequency domain. This analysis includes: message parsing, waveform analysis, frequency spectrum and link time analysis.

RFID MeETS The CISC RFID MeETS is a National Instruments and LabVIEW based measurement system. Conformance and performance tests for tags and readers in the complete HF and UHF band can be carried out. With this system it is possible to emulate tags and test readers with specific signal parameters. The measurements include very detailed information about the captured communication in the time and frequency domain. The software supports many different protocols e.g. ISO/IEC 14443 A/B, ISO/IEC 15693/18000-3 and ISO/IEC 18092 NFC. Through the extendable LabVIEW executable and the modular hardware and software the system can be adjusted to many different environments.

RFID Field Recorder The RFID Field Recorder is a battery powered system to measure the electromagnetic field. This instrument helps to develop and setup RFID tagging systems. A possible measurement scenario would be to find critical tag locations on a pallet moving through an antenna gate. With the help of the graphical analyzing software it is possible to improve the reader position.

RFID Tag Emulator The CISC RFID Tag Emulator is a portable device and can emulate up to four UHF RFID tags in real-time. With this system it is possible to vary tag parameters exceeding the limitations of the standards. The extensive logging capabilities allow an insight into the tags in order to understand what is going on in mixed tag populations. Possible use cases for the Tag Emulator

are: reader evaluation, physical layer tests and optimization, data management application tests, reader development support and customer commands and protocol support.

NFC Xplorer The NFC Xplorer is a compact test system, developed for HF RFID and NFC devices for measurements and performance and conformance tests. The hardware and software described in this research correspond to the version - NFC Xplorer 100. The hardware specifications and the new software features of the new version are described on the homepage[9].

The hardware used for this system was developed by the company Ettus and named USRP N200. Some of the important hardware specifications are shown in chapter 6. With this instrument and the additional LabVIEW software developed by CISC it is possible to sniff the communication between a terminal reader and a smart card. The software can display the envelope and the frequency spectrum of the signal. To analyze the communication it is possible to select between the ISO/IEC 14443 A/B, ISO/IEC 15693, ISO/IEC 18092 standards. The output of the analysis are all requests and responses, with the transmitted data displayed by the software and stored in a text file. To get a better graphical overview of the whole communication history the text file can be loaded by means of additional software. Not only analysis output can be stored but also the captured complex signal wave can be saved on the hard-drive for further analysis. For improved analysis of the signal curve the software provides the possibility measuring each rising and falling edge, the overshooting and many more signal relevant parameters. A disadvantage of the old version of the software is that it is only possible to analyze chunks of an approximate 300 ms duration. The trigger functionality during the capturing process is also available if the signal duration is approximately 300 ms. This disadvantage is no longer existent in the newer version because the software is capable to analyze the signal in real-time.

[9] www.cisc.at

4. Technical Challenges

This chapter describes the main problems of the hardware and software parts of a NFC-sniffer. I encountered mostly the same problems with my solution as the developers of the commercial systems. The main problems are: choice of the sniffer coil, sampling technique, data transfer and signal decoding.

4.1. Sniffer Coil

It is very important that the sniffer coil is properly terminated with the same resistor as the cable which connects the sniffer coil and the Analog Digital Converter (ADC) of the RedPitaya. The orientation and distance between sniffer coil and the payment card (PICC) change the direction and the amplitude of the induced voltage in the sniffer coil. Not only these two parameters of the sniffer coil influence the induced voltage, but also the antenna size of the PICC. The size of the antenna is specified by the class of the card [7]. The three different class types are shown in figure 4.1. These potential changes of the induced voltage make it very difficult for the decoder software to analyze the voltage curve. Not only the potential changes of the voltage may lead to problems, a too small peak-to-peak amplitude of the signal can have the same effect.

4.2. Sampling

As described in section 1.1 the induced voltage curve in the sniffer coil is measured with the RedPitaya. There are two ways to measure the signal. The first and straightforward method is to use a series of samples of the momentary amplitude of the signal. This

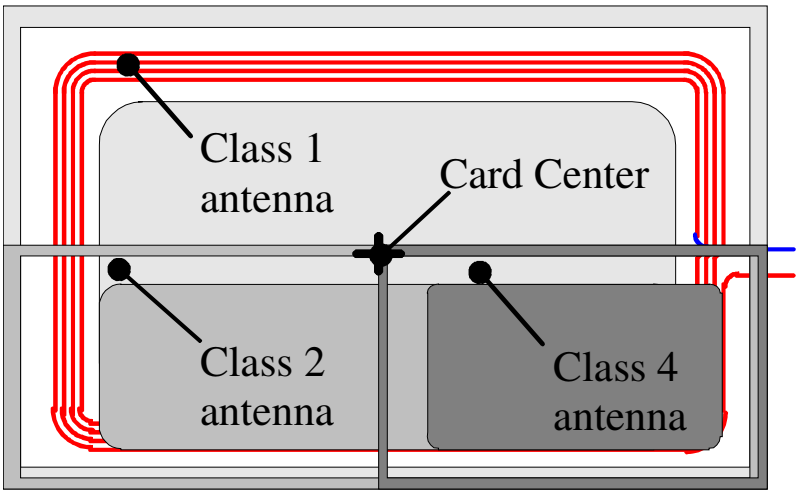


Figure 4.1. – Card antenna areas for the different classes [7].

method leads to a higher amount of data and makes the decoder much more complex and error-prone. The second possibility used by the NFC-sniffer and the professional systems is the in-phase quadrature demodulator and a DDC. The advantages are: a smaller amount of data, it is easier to calculate the envelope of the signal and it is very easy to perform an AM-demodulation. The disadvantage is that the quadrature demodulator and the DDC are more complicated and need more effort to implement them for the FPGA of the RedPitaya. Section 2.6 describes in more detail the two different methods of the signal representation.

4.3. Data Transfer

The sampled signal is sent over Ethernet to the PC. With a 1 GBit/s Ethernet connection it is not possible to send the samples in real time without using a DDC. With a DDC it is possible to reduce the sampling rate and therefore the amount of data transmitted to the PC. One possible implementation of a DDC is explained in chapter 6.

4.4. Signal Decoding

To make the signal decoder software on the PC simpler, the FPGA part of the RedPitaya filters the signal so that the outgoing signal of the RedPitaya is the envelope of the measured signal. Calculating the envelope in software is much more complex, CPU-intensive and needs signal processing software. Even if the decoder software receives the envelope as input there remain some difficult parts when digitalizing the signal. First the decoder has to find the start of the communication in the data set. For communication from PCD to PICC the system uses ASK and a load modulation is used for the communication from PICC to PCD. These different modulation types implicate that the decoder has to distinguish between request signal (PCD to PICC) and response signal (PICC to PCD). The threshold for the request signal where the signal switches between zero and one, is easier to calculate automatically than the threshold for the response signal. The threshold for the response signal depends on the distance between sniffer coil and smart card and between smart card and terminal reader. The higher the distances are the lower the amplitude is and so the threshold varies too. Another problem occurs when the direction of the smart card changes in comparison to the terminal reader because the direction of the induced voltage may change and so the decoder has to handle many different options. The solution to those problems in this NFC-sniffer system is explained in chapter 7.

5. Design

The four different designs explained in this chapter may lead to a good solution. The third and the fourth design presented in this chapter were implemented for this master thesis. All four solutions contain the RedPitaya as the measurement and precomputation unit and the PC as the computation and visualization unit. The FPGA has to provide in all designs at least the BRAM as an accessible register to copy the stored samples with a C-program running on the RedPitaya into the Random Access Memory (RAM). Figure 5.1 shows the high-level software work-flow remaining the same for all four designs. The differences between the designs are the responsibility of the single tasks shown in figure 5.2 and the software used.

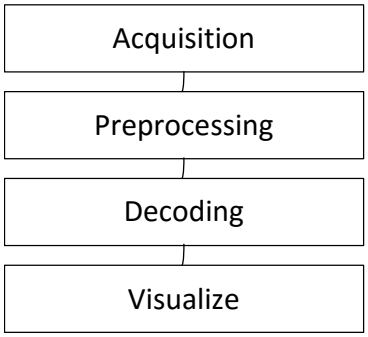


Figure 5.1. – Software high-level work-flow

5.1. Matlab as decoder

The first design is an easy and fast solution. It is very helpful to gain a quick overview of how the communication between PCD and PICC works. It is also useful to have a glimpse of how the whole system behaves by modifying parameters regarding the

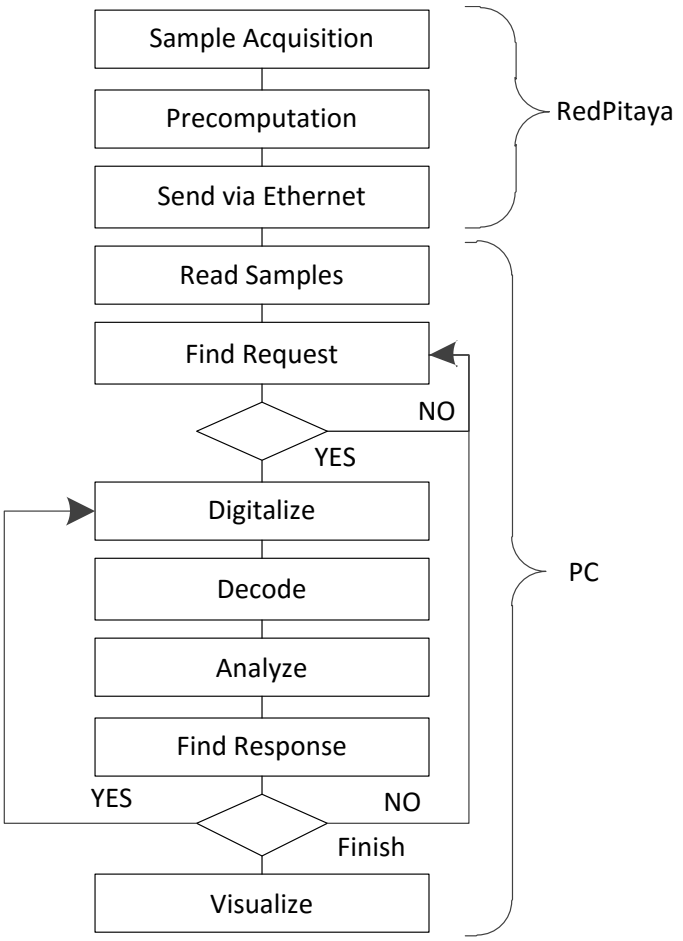


Figure 5.2. – Software work-flow

operating volume and the direction of the sniffer coil. The design is as follows: the RedPitaya samples the signal with serial samples with 31.25 MS/s in order to reconstruct the whole signal including the carrier frequency of 13.56 MHz and the subcarrier frequency of 847 kHz without losing information. The second process shown in figure 5.2 "Precomputation" in this design consists of preparing and saving the samples into a text file and to send them via Ethernet to the PC. On the PC this text file is imported into Matlab. Matlab is a good software to obtain a fast overview of the signal because it is easy to display signal traces and to modify them if needed. Matlab is a powerful tool for signal processing tasks but it is also possible to implement the signal decoding and protocol analysis. Since there are no DDC and filter implemented on the RedPitaya, the text file with the samples is very large. Therefore the arrays used by Matlab to handle signal traces are very large but if the PC has enough RAM it is not a problem for Matlab. The amount of data is not the only reason why this design is not used in this system and

in commercial systems. To receive the protocol information out of the signal trace it has to be demodulated, filtered, digitalized and decoded. All these tasks can be done in Matlab but the combination of filtering and digitalizing is a very complex process. The amplitude of the signal changes if the sniffer coil moves around the operating volume or when another terminal reader is used and then the threshold for the digitalizing process needs to be adjusted. This adaption of the threshold to the amplitude of the signal is very complex to implement automatically. To set the threshold manually after each measurement however is not difficult but inconvenient for the user. In this first design Matlab is responsible for finding the request and responses within the signal digitalize, decode and analyze them. The analyzed signal and the found protocol commands should be written into an output file for the user.

Advantages The advantage of this design is the low effort to obtain a relatively good result. The FPGA part is not very complex and the Matlab-script can be implemented within a few days.

Disadvantages The main disadvantage is the error-proneness of the combination of the filter and digitalization tasks and the threshold which cannot be calculated automatically. Furthermore the high sampling rate leads to large files and high data transfer rate between RedPitaya and PC.

5.2. RedPitaya only

The second design is a very compact design and with this hardware very hard to feasible. The majority of the tasks are handled by the RedPitaya. The PC only has to visualize the information for the user. This design was not implemented completely because the required performance of the signal processing tasks has not been reached. The sample acquisition task remains the same as in the first design but the following tasks shown in figure 5.2 are implemented on the RedPitaya except the "Send via Ethernet" because it is not needed anymore in this design. It is possible to use signal processing tools on the RedPitaya e.g. Octave, but they are not able to handle such large arrays as is needed in this system. Also the experiment to implement the filter in another programming language (e.g. C,C++) failed due to performance reasons. Another

problem may be the performance of the visualization because the RedPitaya only has the Ethernet connection to send the Graphical User Interface (GUI). It would also be possible to avoid this problem by preparing a save file on the RedPitaya and a second software running on the PC could decode the information to build a GUI.

Advantages The advantage of this design is that the measurement is independent of the software on the PC, if the web server on the RedPitaya is fast enough to transmit the needed information for the GUI.

Disadvantages Disadvantages however are the limited GUI possibilities if no second software for the PC is developed and the resulting very low performance.

5.3. RedPitaya as DDC

This design is one of the two implemented versions of this master thesis. The FPGA part of the RedPitaya with the DSPs is used and therefore some problems of the previous design can be avoided. The sample acquisition remains the same again but the precomputation task makes the main difference. In this task the FPGA of the RedPitaya is used to do the whole signal processing needed in this system. Furthermore a DDC is implemented. With the implementation described in section 7.1 the amount of data transmitted between RedPitaya and PC is reduced by a significant factor. With some more optimizations described in chapter 9 it would also be possible to send the sampled signal in real-timer via the 1 GBit Ethernet to the PC. In this master thesis the filtered samples are saved and then transmitted over Transmission Control Protocol/Internet Protocol (TCP/IP) to the user software running on a PC. The software on the PC takes care of the remaining tasks: read samples, find the requests and responses within the signal trace, digitalize the found signal parts, decode the signal either “Manchester” or “Modified Miller”, analyze the protocol flow and finally visualizes the result to the user. This design is described in more detail in chapter 6 and 7.

Advantages The main advantage compared to the previous two designs is the better usage of the FPGA as a signal processing unit. Consequently the amount of data is decreased and the software on the PC does not have to handle that huge amount of data. This increases the performance of the program and the usability.

Disadvantages The main remaining disadvantage is that the capture time of the RedPitaya is limited to the available RAM on the board because the samples have to be saved first before they are sent over TCP/IP.

5.4. RedPitaya as DDC with trigger

The last design contains further improvements but also a restriction regarding the third one. The main improvement and a very useful feature is the trigger functionality. An additional register allows to set a trigger level to decide when the system should start to store the signal. The signal is then saved in the BRAM and is accessible by a C-program when the acquisition stops. To check if the signal amplitude oversteps the trigger value the amplitude of each I/Q-sample pair has to be calculated by the FPGA with equation 2.1. The earlier mentioned restriction is the maximum amount of storable samples. With the actual FPGA and acquisition software configuration and the used OS it is only possible to stored about 8M values containing the peak amplitude shown in equation 7.5. It would be possible to increase the amount of stored samples by changing the boot configuration and the device-tree of the used OS. The maximum of available memory for the ADC values depends on the free RAM of the running OS. In both this and the previous design it is possible to switch the capturable signal duration between 1.342 s and 3.355 s by changing the decimation rate of the CIC-filter

Advantages The trigger functionality makes the capturing process more convenient.

Disadvantages If the communication duration between PCD and PICC is longer than 3.355 s this design cannot be used to capture the signal.

6. Hardware

This chapter provides a more technical overview of the platform RedPitay. It contains two tables 6.1 and 6.2 showing the important hardware specifications of the RedPitaya and the hardware used by CISC (Universal Software Radio Peripheral (USRP) N200). This section describes only the most important and for this work most relevant information, more information can be found on the RedPitaya homepage [17] and on the Ettus homepage [6].

6.1. Hardware Specifications

With dimensions of only 107 x 60 x 21 mm, the RedPitaya is a very small and flexible measurement tool. The measurement setup shown in figure 1.1 is very easy and can be accomplished within minutes. The relative low power consumption allows to use the RedPitaya with a battery pack instead of the power adapter. With these properties the RedPitaya is an easy to transport and a user friendly system. Also the hardware used by CISC the USRP N200 has these positive characteristics and is therefore also very flexible and easy to transport. As table 6.1 shows the bandwidth of the RF inputs of the RedPitaya is 50 MHz and therefore high enough for measurements up to RF frequencies. The USRP however has 6 GHz of bandwidth and is therefore also usable for UHF applications. To sample the whole spectrum of a signal without losing information the Nyquist-Shannon criteria, shown in equation 6.1 has to be satisfied. As table 6.1 shows the highest possible sampling rate of the RedPitaya is 125 MS/s and is therefore high enough to sample the communication with the carrier frequency of 13.56 MHz without losing information. In this specific use case it would also be possible to undersample the signal, which means that the Nyquist-Shannon criteria is not satisfied because the carrier frequency is not needed for further analysis of the

signal. This would decrease the amount of data that has to be filtered and sent to the user software. Instead the data rate is reduced using a digital down conversion technique described in the next section 7.1.

$$f_{sampl} \geq 2 * f_{max} \tag{6.1}$$

Another important value is the high input impedance of the ADC-converter shown in table 6.1. The high impedance at the input is important because the connected sniffer coil should have as less as possible influence on the electromagnetic field during the communication. This is only possible if the measurement setup consumes minimal current from the electromagnetic communication field. The sniffer coil has to have only a few important characteristics to receive good measurement results. These characteristics are: the size of the coil should be about the same size as the PICC, the number of windings should not lead to an overly high induced voltage for the ADC and the sniffer coil has to terminate with the same resistor as the cable connecting the sniffer coil with the ADC. It is also possible to use a copper wire and measure the induced voltage with a probe. The advantage is that the probe is terminated with the right resistor and the copper wire can be easily turned to the needed shape. If the induce voltage is too high some windings should be removed.

A great advantage of the RedPitaya is the SoC including the Central Processing Unit (CPU) and the FPGA. The built-in ARM-processor allows running an OS on the platform. The best choice of OS is a Unix either a Beta Linux or Debian adjusted for the ARM processors. With the OS it is possible to run scripts and custom programs very easily on the RedPitaya. The signal processing part is done by the FPGA as described in section 7.1 and the analyzing part on a PC connected via Ethernet. The analyzing part could also be done on the RedPitaya by the ARM-processor but this would take too long because the software has to handle large amounts of data. More reasons why the ARM-processor in adequate computational power and why commercial systems use a PC to solve the analysis problem are described in chapter 7.

		RedPitaya
RF input		
	Channels	2
	Bandwidth	50 MHz
	ADC resolution	14 Bit
	Sample rate	125 MS/s
	Input impedance	1 MΩ
	Full scale voltage	$46V_{pp}$
CPU/FPGA		
	Xilinx Zynq 7010 SoC	
	Dual ARM Cortex-A9	
	Logic Cells	28.000
	DSP slices	80
	BRAM	240 kByte
	Memory	512 MByte
SD card		
	Type	micro SD
	Size	max 32 GByte
Power supply		
	Voltage	5V
	Current	2A
	Connector	micro USB
Dimensions		107mm x 60mm x 21mm
Power consumption		0.9A
Interface		Gigabit Ethernet

Table 6.1. – RedPitaya hardware specification

		USRP N200
RF input		
	Channels	2
	Bandwidth	6 GHz
	ADC resolution	14 Bit
	Sample rate	100 MS/s
	Input impedance	1 MΩ
	Full scale voltage	
FPGA		
	Spartan-3A DSP	
	18 Bit Multiplier	18x
	Logic Cells	37.440
	DSP slices	84
Daughterboard		
	Type	LFTX 0-30 MHz Tx
Power supply		
	Voltage	6V
	Current	1.3A
Dimensions		220mm x 160mm x 50mm
Interface		Gigabit Ethernet

Table 6.2. – USRP N200 hardware specification[6]

7. Implementation

This chapter explains how the software for the NFC-sniffer is implemented for the RedPitaya and for the user on the PC in more detail. As described in chapter 5 two programs for the RedPitaya and one program for the user software are needed to handle the whole process. The FPGA implementation is described in section 7.1. The first software described in this chapter is the FPGA-implementation, followed by the used OS, then the implemented acquisition-software for the RedPitaya and at the end the user software for the PC.

7.1. FPGA Implementation

During the implementation and testing process two different versions of FPGA designs came up and will be described in this section. Additional this section describes the advantages and the disadvantages of the use of the FPGA on the RedPitaya. Furthermore the different filters, signal processing units and other blocks used in the FPGA designs are described in the following paragraph. Some of the general advantages have been described in section 2.4 but there are also some more specific advantages used in this NFC-sniffer system.

7.1.1. FPGA without trigger

To receive the envelope of the Amplitude Modulation (AM)-modulated signal of the communication between PCD and PICC a signal processing tool is needed. A lot of programs could solve this problem e.g. GNURadio [10], Matlab [12], LabVIEW [15] but the signal processing step can also be done in hardware with a FPGA. A great

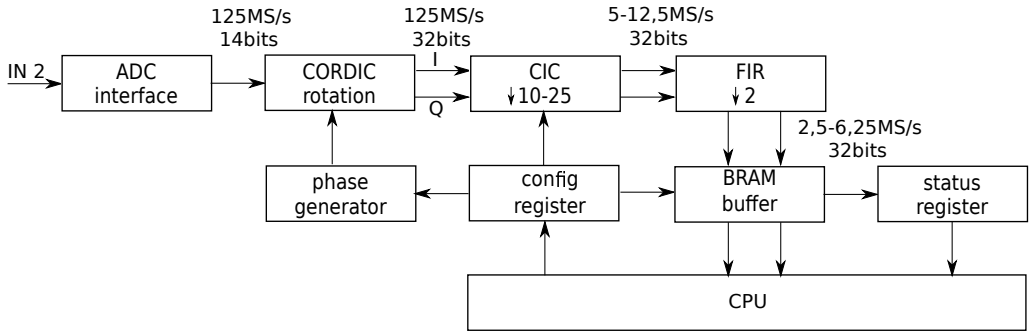


Figure 7.1. – Block-diagram of the sample acquisition

advantage when the signal processing step is done on the FPGA is, that the amount of samples transferred from the RedPitaya to the user software is significantly smaller because the DDC can also be done in hardware. The main advantage of the FPGA part in this system is the high data throughput. The FPGA design, depicted in figure 7.1, can transfer 125 MS from the input through the CORDIC-algorithm, the filter bank to the CPU in real time with a small latency. In this design the back-end can choose the sample rate between 2,5-6.5 MS/s. This relative low sample rate contains the information of the whole signal without the carrier frequency because it is not needed for the protocol analysis task. At the beginning of the sample acquisition the user program has to set some configuration parameters for the FPGA circuit. These parameters are:

Phase increment This parameter specifies the step size for the CORDIC algorithm. With this phase increment the sine and the cosine will be computed correctly for each ADC measurement point.

Sample rate The adjustable sampling rate is not the sampling rate of the ADC but the amount of I/Q-data points stored in the BRAM within one second. This sampling rate is variable between 2.5-6.5 MS/s.

Reset mode With this parameter in the configuration register it is possible to start and stop the acquisition and filter process.

After these parameters are set to the right values the signal processing starts with the sample acquisition of the ADC with a sample rate of 125 MS/s and a resolution of 14 Bit as shown in figure 7.1. As the name already suggests the phase generator generates the phase for the CORDIC-algorithm with the previously set phase increment in the configuration register. The CORDIC-block computes the sine and the cosine with

the phase received from the phase generator as shown in equation 7.1. The output of the CORDIC-block are the I- and Q-samples, computed by multiplying the signal with the calculated sine and cosine. After this computation the data throughput is 125 MS/s with a data bandwidth of 32 Bits. The next block is the CIC filter. This type of filter is used in digital signal processing to increase (interpolating CIC) or to decrease (decimating CIC) the sample rate of a signal. CIC-filters are very easy to implement for FPGAs because all their coefficients are "1" and therefore no complicated multiplication is needed. The used decimation CIC-filter passes only every 25th sample to the next block. To avoid aliasing a lowpass filter suppresses the signal components with a higher frequency than half of these sample rate of the decimated output signal. The last block in the signal processing tool chain is a FIR-filter with the band edge at 1.1 MHz. The magnitude response of the used filter is shown in figure 7.2. The FIR-filter has a decimation rate of two. With these filter settings the usable bandwidth is about 2 MHz as shown in equation 7.3. How these values were computed is shown in the following equations.

$$\begin{aligned} &\text{A}\ldots\text{digitized signal amplitude after the ADC} \\ &\text{f}\ldots\text{frequency from the phase generator} \\ &I = A * \cos(\omega * t) \\ &Q = A * \sin(\omega * t) \end{aligned} \tag{7.1}$$

$$\begin{aligned} \text{Band edge frequency}\ldots f_e &= \frac{\text{sampling rate}}{\text{decimation factor}} * 0.22 \\ &= \frac{125\,\text{MS/s}}{25} * 0.22 = 1.1\,\text{MHz} \end{aligned} \tag{7.2}$$

$$\begin{aligned} \text{Bandwidth} B &= f - f_e \text{ to } f + f_e \\ &= f - 1.1\,\text{MHz} \text{ to } f + 1.1\,\text{MHz} = 2.2\,\text{MHz} \end{aligned} \tag{7.3}$$

Why is this low bandwidth enough to sample the communication between PCD and PICC?
Section 2.5 introduces the subcarrier frequency with 847 kHz and the transfer rate with 106 kBit/s. With this information the smallest possible bandwidth can be calculated as shown in equation 7.4.

$$\text{Bandwidth } B = 2 * 847\,\text{kHz} + 2 * 106\,\text{kBit/s} = 1906\,\text{kHz} \tag{7.4}$$

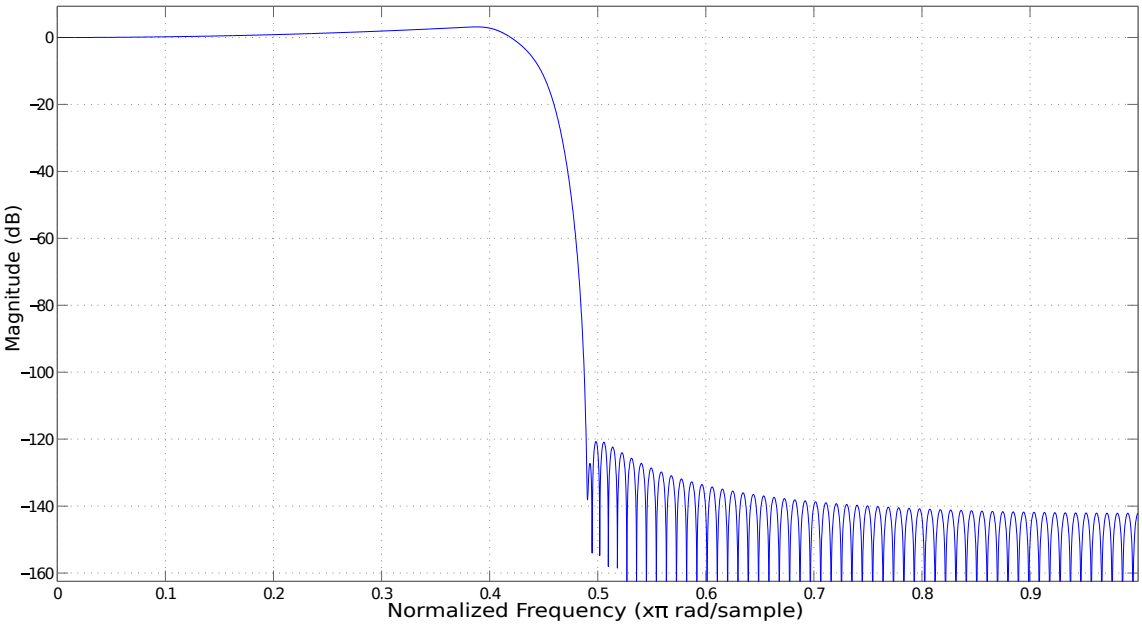


Figure 7.2. – Magnitude response FIR-filter

With the shown calculation of the I/Q-samples and the use of the filter bank it is possible to restrict the needed bandwidth to about 2 MHz and decrease the sampling rate from 125 MS/s to 2.5 MS/s. This reduces the amount of data that has to be saved by the RedPitaya, then send over the network to the PC and finally the software has to handle by a factor of 50.
The last important aspect discussed in this section is the resource utilization of the FPGA. The different resources of an FPGA are limited and therefore also the functionality range. The FPGA used on the RedPitaya platform is the Xilinx Zynq 7010 SoC as shown in table 6.1 and it is the smallest within the Zynq-7000 family. Table 7.1 shows how many of the different FPGA parts are used. It is observable, that the percentage of used DSPs handling the signal processing task is about 90% and therefore the design could not be much larger regarding signal processing tasks. The other listed resources in the table are under 25%, except the connected but not used I/Os, therefore the FPGA can handle more tasks but no signal processing functions.

7.1.2. FPGA with trigger

The FPGA code version with trigger functionality is more improved regarding the implementation described previously. The differences between the two FPGA versions

Resource	Utilization	Available	Utilization [%]
Flip Flop	6488	35200	18.43
Look Up Table (LUT)	3630	17600	20.63
Memory LUT	1275	6000	21.25
I/O	60	100	60.00
BRAM	2.50	60	4.17
DSP	71	80	88.75
Global Clock Buffer	2	32	6.25

Table 7.1. – FPGA utilization without trigger

are shown in figure 7.1 and figure 7.3. As the second figure shows the main differences are the amplitude calculation and the trigger functionality. The ADC interface, the CORDIC block and the filters remain the same, the data width changes. The configuration registers were extended by one register to set the trigger-level on the FPGA with the acquisition-software running on the RedPitaya. The first difference between the two versions appears after the signal processing step immediately after the FIR-filter. The amplitude computation block calculates the signal amplitude out of the I-data and the Q-data of each sample using equation 7.5. At least the trigger block checks if the computed amplitude oversteps the defined value in the configuration register. If that is the case the ongoing samples will be stored in the BRAM and can be read by the acquisition software. It is also possible to save an adjustable number of samples before the signal amplitude oversteps the trigger level.
Within this version of FPGA software the BRAM is not implemented as a circular buffer and therefore it is not possible to load more samples into the CPU than can be stored on the BRAM. The small memory on the FPGA limits the maximum capturable samples to 8 MS. Since it is possible to change the sample rate between 2.5-6.25 MS/s, the maximum duration is either 1.342 s or 3.355 s. These two values can be calculated as shown in equation 7.6.

$$Amplitude = I^2 + Q^2 \tag{7.5}$$

$$\begin{aligned} \text{Duration} &= \frac{8\,\text{MS}}{6.25\,\text{MS/s}} = 1.342\,\text{s} \\ \text{Duration} &= \frac{8\,\text{MS}}{2.5\,\text{MS/s}} = 3.355\,\text{s} \end{aligned} \tag{7.6}$$

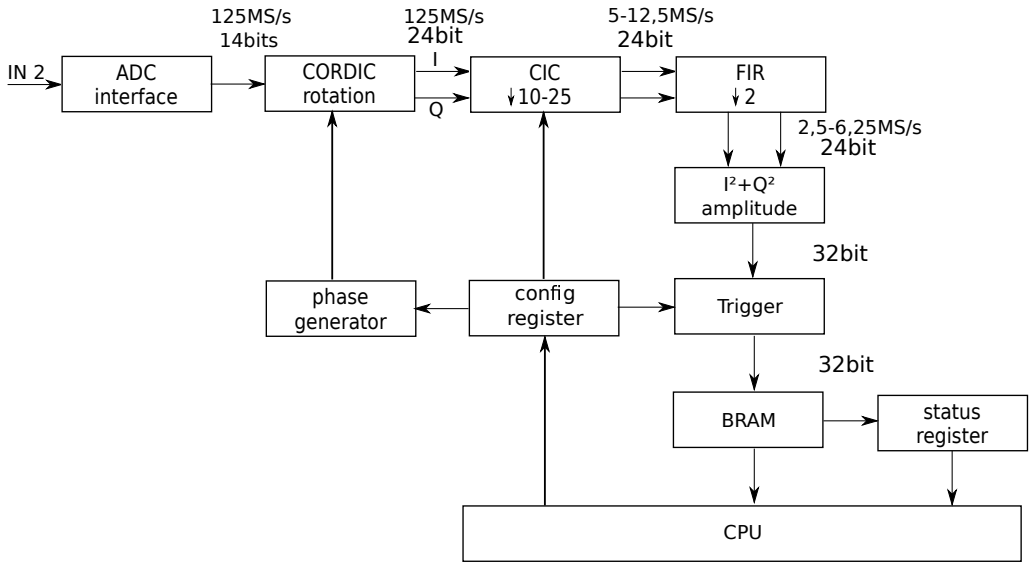


Figure 7.3. – Block-diagram of the sample acquisition with trigger

Even if the maximum capturable signal duration is limited, the trigger functionality is a great advantage regarding the usability of the system. If the sampling rate is set to 2.5 MS/s the resulting signal duration of 3.355 s is enough for all cases without any waiting time extensions during the communication and for the most successful communications within the ISO-standard.
Comparing the two tables 7.1 and 7.2 it is observable that the second table, which represents the FPGA utilization with the implemented trigger functionality is nearly the same except for the line with the DSP usage. Since the data width is smaller this implementation needs less DSP slices.

Resource	**Utilization**	**Available**	**Utilization [%]**
Flip Flop	6614	35200	18.79
LUT	3728	17600	21.18
Memory LUT	972	6000	16.20
I/O	60	100	60.00
BRAM	2.50	60	4.17
DSP	43	80	53.75
Global Clock Buffer	2	32	6.25

Table 7.2. – FPGA utilization with trigger

7.2. Operating System

The RedPitaya homepage provides a running OS which can be downloaded, copied onto the SD-card and it will work but with some limitations in comparison to the full installation of an OS. The provided OS is a kind of Linux version without a package manager and also many other functionalities are missing. It is a lightweight OS and adequate if no programs or libraries are missing for the needed implementation. With this OS the new developed program running on the RedPitaya have to be cross-compiled with a toolchain on another platform. To avoid the installation of the toolchain and to provide more flexibility on the RedPitaya a user has ported Linux to the RedPitaya platform [3]. In the NFC-sniffer system Debian 8 also called Jessie OS is used. Originally this distribution of Linux is developed for 32-bit x86-based PCs but today also runs on many other platforms including ARM architecture. Both, the provided Linux from the RedPitaya homepage and Debian version make the RedPitaya accessible via Secure Shell (SSH) over Ethernet or the second micro-USB-port. To port the Debian distribution on the RedPitaya platform some changes in the device-tree and start-up scripts are needed. In the start-up scripts the configuration file for the hardware components have to be adjusted and the parameters set to the right values. The device-tree received new char devices to load the FPGA software and to connect the PC with the RedPitaya over the micro-USB-port. The first char device allows the user to copy the compiled FPGA-code onto the SD-card with a SSH over the USB-port and the second to load the copied file on the FPGA while the RedPitaya is running. With the package manager it is easy to install new software and therefore the C or C++ compiler can be installed and used on the RedPitaya. If someone likes to program with Perl or Python it is possible to develop the software directly on the platform. The reason for additional software to the FPGA-software is describe in the next section. For the NFC-sniffer system the compiled FPGA-code is saved on the SD-card and loaded automatically after the RedPitaya starts. To ensure that the sampling and filter process works smoothly and no error occurs the clock frequency has to be increased at start-up. To do that the lines shown in listing 7.1 have to be inserted at the end of the file $/etc/rc.local$.

7.3. Acquisition-software

The software concept described in this section works for both versions of FPGA code introduced in section 7.1. In this system the acquisition-software is written in C because it is easy, fast, the tasks handled by this software are not very complex and the software is not comprehensive. The software running on the RedPitaya is called acquisition-software because after the required samples are stored in an array it offers a TCP/IP server socket on which the user program can connect and download the data. The acquisition-software loads the sampled and filtered signal out of the BRAM where the FPGA program saves it. The acquisition-software also expedites possible configurations and receives information via the status registers. To give the user access to this memory space and the registers there are two possibilities. The first method is to add a new char device to the device-tree of the OS and to write a kernel driver to map the hardware address of the FPGA registers to the userspace. With this method the required kernel driver has to be implemented. The second possibility used by this NFC-sniffer system is to address the hardware registers directly in the C-program with the *mmap* function. This function maps the registers to the userspace and makes the data accessible with the function $memcopy$. Listing 7.2 shows how the hardware registers are mapped. The first variable "cfg" is used to set parameters for the FPGA like sampling rate, trigger edge and trigger level. The flags "PROT_READ" and "PROT_WRITE" enable the possibility to read and write the allocated memory with the given size of "PAGESIZE". The last parameter is the address configured in the FPGA software. The second line defines the memory space for the status register. The last variable "ram" allocates the memory space for the captured samples. Listing 7.3 shows how the data is copied to a large array.

How command and data transfer between client-software and acquisition-software is implemented is shown in figure 7.4. This diagram shows that at the beginning the acquisition-software creates the server-socket and waits until the client software

```
1 echo fclk0 > /sys/devices/soc0/amba/f8007000.devcfg/fclk_export
2 echo 1 > /sys/devices/soc0/amba/f8007000.devcfg/fclk/fclk0/enable
3 echo 143000000 > /sys/devices/soc0/amba/f8007000.devcfg/fclk/fclk0/set_rate
4 cat /home/rfid_ddc.bit > /dev/xdevcfg
```

Listing 7.1 – Load FPGA software

connects to this socket. After the connection is established, the acquisition-software starts with the sample acquisition and the filtering process. At the end the sampled signal data will be sent over the created connection to the client-software where the signal analysis starts.

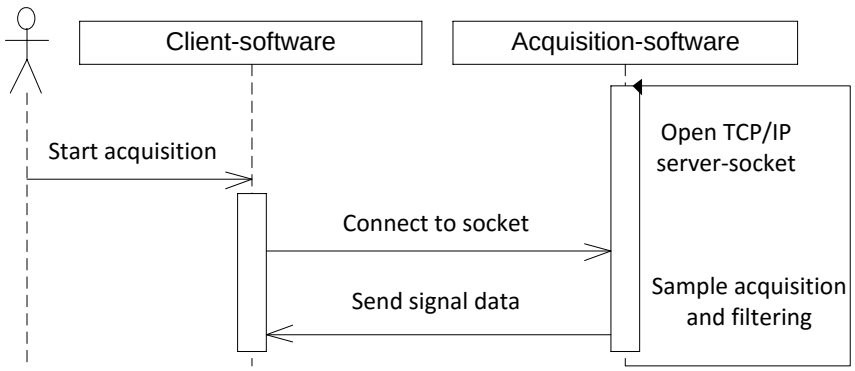


Figure 7.4. – Sequence diagram client and RedPitaya interaction

7.4. Client-software

The client-software runs on the PC and undertakes the task of the analysis of the captured signal. The general work-flow of the software is shown in figure 5.2 and will be described in more detail in this section. The client software should show the sampled signal in a graph, the single protocol commands used during the captured communication and also act as control panel of the whole capturing process. To solve this task the software is programmed in java with some additional libraries for the GUI and the graphical signal representation. The previously named tasks are split into

```
1 cfg = mmap(NULL, PAGESIZE, PROT_READ | PROT_WRITE, MAP_SHARED,fd, 0x40000000);
2 sts = mmap(NULL, PAGESIZE, PROT_READ | PROT_WRITE, MAP_SHARED,fd, 0x40001000);
3 ram = mmap(NULL, 8192 * PAGESIZE, PROT_READ,MAP_SHARED, fd, 0x1E000000);
```

Listing 7.2 – Map the hardware registers into the userspace

```
1 memcpy(sig_buf + transferred, ram + offset, 1024);
```

Listing 7.3 – Copy the data out of the BRAM to a large array the sig_buf

three different parts on the main screen of the client-software shown in figure 1.2. The control panel of the software is shown in the right upper part and the signal graph in the lower part of the screen. The white space in the left upper part of the figure shows the reserved space for the found and decoded commands during the communication between PCD and PICC displayed within a tree view. With the client-software it is possible to load the sampled signal directly from the RedPitaya in case of a new measurement or from a saved dump-file. In both cases the samples are loaded into an array and the decoder starts to iterate over the data points. The decoder first searches the beginning of the communication and then the first request of the PCD. If a request is found this part of the signal will be digitalized and decoded with the modified Miller decoder function. If the decoder could decode the digitalized request without any errors the protocol analyzer tries to match the decoded data with a command given by the set protocol type. Once the decoder is finished with the first request it starts to search for the first response before the next request occurs in the signal. The analyzing task of the response remains almost the same only the digitalizing task is much more complex and the modified Miller function needs to be replaced with the Manchester decoder.
Figure 7.5 shows the PCD request with the higher amplitude jumps and the PICC response with the smaller amplitude difference. For a human eye it is easy to digitalize the request and the response of the signal but not for software. The request signal will not be changed if the PICC is moved around in the operating volume and therefore the threshold which decides if the analog signal represents a logical “1” or a logical “0” can be implemented as a constant value. But if the distance between PCD and sniffer coil changes the amplitude value of the request signal may be lower and therefore the software provides an additional field to enter the threshold and a button to reanalyze the signal with the new entered threshold. A characteristic of the modified Miller coding is that the duration of a high pulse can be 0.5 Bit, 1 Bit or 1.5 Bit long. However the duration of a low pulse can only vary between 0.5 Bit and 1 Bit. This duration is measured using the software by counting the number of samples recognized as logical "0" or logical "1" and then comparing the count with predefined constants. To digitalize the response part of the signal it is not possible to use a predefined threshold because the amplitude changes if the position of the PICC relative to the sniffer-coil changes or if the PCD uses a stronger or weaker electromagnetic field or if the class of the PICC is different. One possible solution to determine the threshold is to calculate the mean value of the last 200 samples of the signal. The amplitude difference between

logical "0" and logical "1" can be so small that a low amplitude oscillation of the last 200 samples leads to many errors during the digitalizing process. If the digital signal representation of the responses is wrong the software cannot match the signal to the right protocol commands. Figure 7.6 shows the possible response of the PICC. The red line shows the calculated threshold. The software cannot distinguish between the three possibilities until the analyzing task is finished. If the software had to analyze the whole signal with the three possibilities it would require considerable CPU power and time. If the difference between threshold and logical "0" value is large enough this solution would lead to an adequate result. But if this is not the case, as shown in figure 7.6b the software could not digitalize and decode the signal.
To avoid the problem with the three different analyzing functions and the calculation of the mean value a different digitalizing approach is used. A small window of about five samples is moved over the signal between two located requests. Each time a new sample is added to the ring buffer, representing the moving window a function checks if the difference between the smallest and the highest amplitude of these five samples is greater than the predefined value of ten. If this is the case then a logical high pulse is found and the software counts how many samples are shifted into the window until the difference between all five samples is smaller than ten which represents a logical "0". As previously described the count value defines the duration of the high and the low pulses. With this improved approach it is possible to digitalize all the three different signal shapes shown in figure 7.6 even if the peak-to-peak amplitude is very low. The success of the digitalizing functionality is shown in chapter 8 where the results of six different measurements are described in more detail.
After the signal is split into request and response, digitalized and decoded without errors the protocol analyzer tries to match the found commands with the predefined commands depending on the selected protocol type. The commands defined by the protocol standard are embedded with additional information in an xml-file. It is easy to extended the file with more commands or protocol types. One of the important requirements for this software is that it should be as extendable as possible and therefore the different protocols can be written into the xml-file. Using a short example, listing 7.4 shows how the xml-file is organized. The element “Types” holds all the different protocols defined by the attribute of the “Type” element. After the protocol is specified the commands are split into “RequestCommands” and “ResponseCommands”. These two elements hold all the required information of the commands needed by the software to match them with the decoded signal. The next element in the xml-file is the

command element with two attributes the commands length, specified by the number of hexadecimal values and the commands name. The command element hold as many child elements as the command length specifies. This child element named “Byte_x” may have more value elements to specify the different bit-pattern the specified byte may have. These values are used to find the corresponding command name. These values may also have an additional name attribute to give more detailed information about the command.

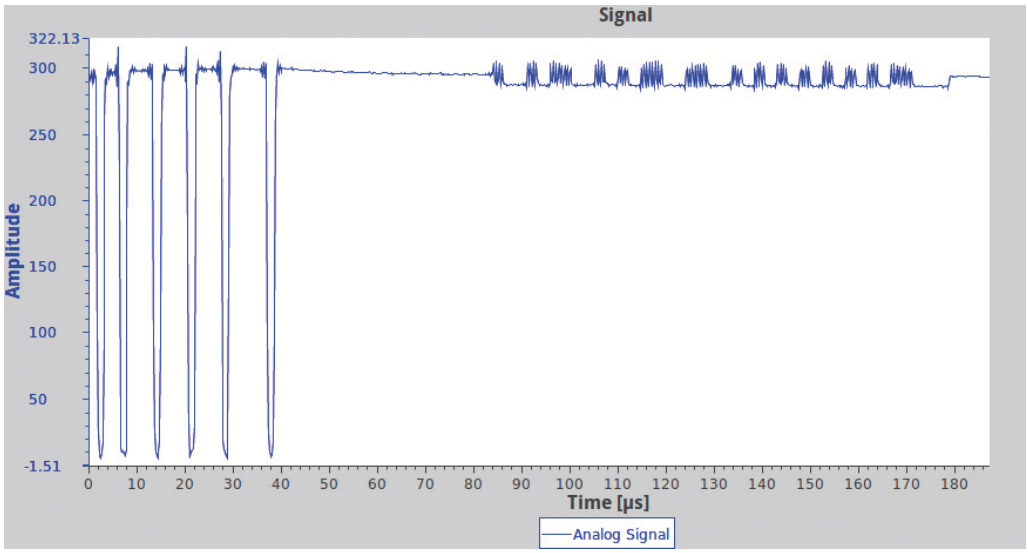


Figure 7.5. – Analog request and response signal

After the signal analysis is finished the software presents the results to the user with a tree view and the signal graph within a canvas element. The tree view lists the found commands in chronological arrangement with some additional information about the structure of the command. Figure 7.7 shows how the “Select” command and the corresponding responses are listed within the tree view. In the first line the green arrow and “Term” indicates that the PCD sent the request followed by the timestamp stating at the beginning of the acquisition. After the timestamp the command name in capital letters is located and the exact command at the end. The second line shows the command in hexadecimal notation. Next comes the expandable header and body. The header element shows how it is formatted within the different blocks in this case the I-Block. The body element shows the content of the body with further information if available. The received response is listed in the same way only the body includes more

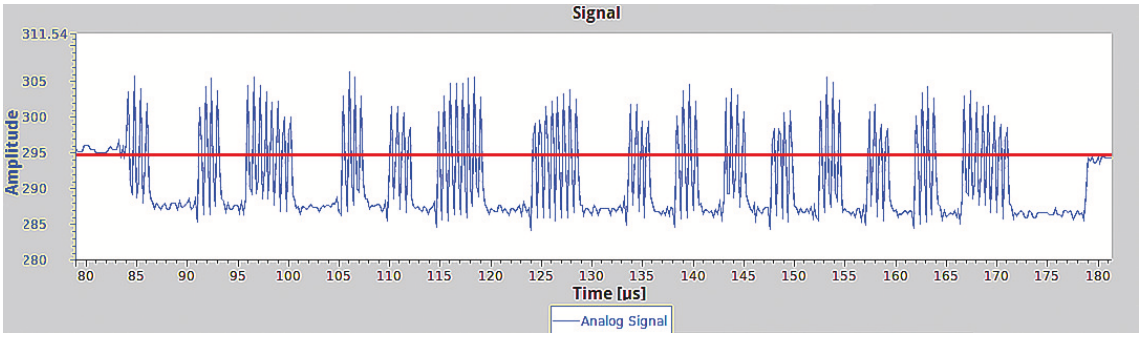
details.

The second graphical view is the signal graph shown in figure 7.8. In this figure the whole analog signal of one capture communication is shown. The view of the whole signal provides a imprecise view of the single commands but with the included toolbar shown in figure 7.9 it is possible to zoom into the interesting parts of the signal. The toolbar also provides also some standard functionality to change the color, description etc. of the axis and the graph.Furthermore, it is possible to add and remove annotations to the graph view and to save the plot as PNG-file. To give the user a better view of the single commands listed in the tree view it is possible to double click on the commands to show the corresponding signal in the graph-view. It is possible to show both request

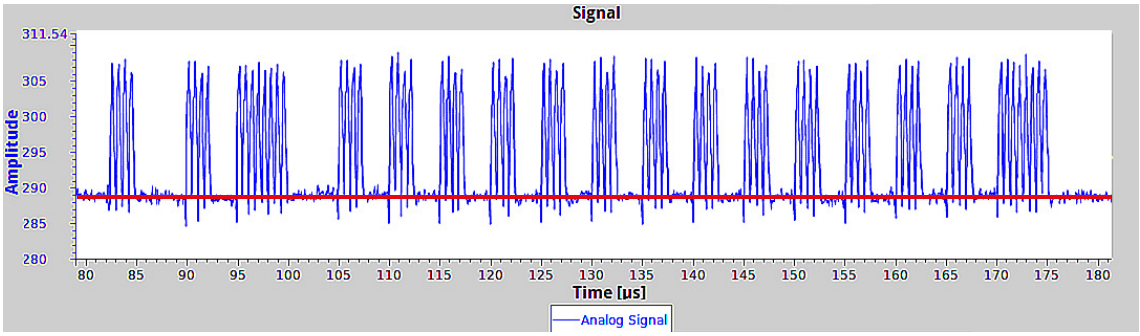
```
<?xml version="1.0" encoding="UTF-8"?>
<!-- cmdLength = number of hex values; 0x52 => length = 1 -->
<Types>
  <Type name="EMVCo">
    <RequestCommands>
      <Command cmdLength="1" cmdName="WUPA">
        <Byte_1>
          <cmdValue>
            <value>01010010</value>
          </cmdValue>
        </Byte_1>
      </Command>
    </RequestCommands>
    <ResponseCommands>
      <Command cmdLength="2" cmdName="ATQA">
        <Byte_1>
          <cmdValue name="UID_double">
            <value>01000001</value>
            <value>01000010</value>
            <value>01000100</value>
            <value>01001000</value>
            <value>01010000</value>
          </cmdValue>
        </Byte_1>
        <Byte_2>
          <cmdValue>
            <value>00000000</value>
          </cmdValue>
        </Byte_2>
      </Command>
    </ResponseCommands>
  </Type>
  <Type name="MIFARE">
    <RequestCommands></RequestCommands>
    <ResponseCommands></ResponseCommands>
  </Type>
</Types>
```

Listing 7.4 – Code snipped of the xml-file

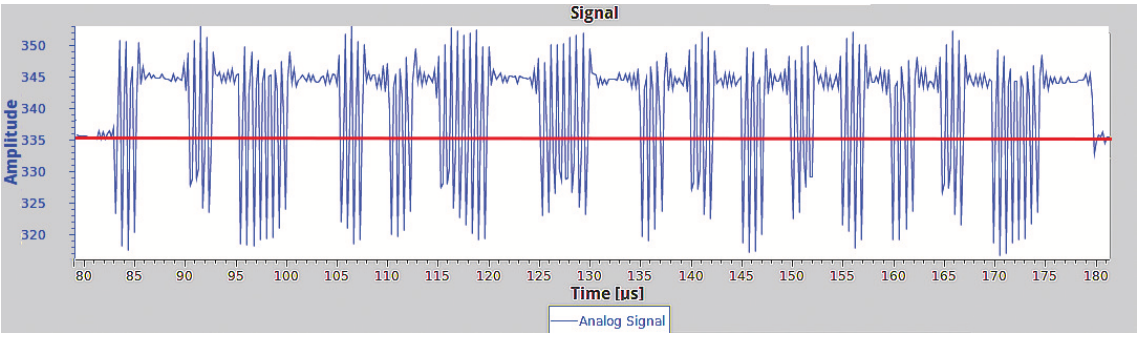
and response or only request or only response. This functionality allows the user improved analysis of the analog signal in case of an error. If the user is interested in how the signal is digitalized by the software the button “D” in the toolbar switches the digital signal on and off but only in the single view of a response or request. This feature is shown in figure 7.10.



(a) Logical “0” value lower than threshold



(b) Logical “0” value equal than threshold



(c) Logical “0” value higher than threshold

Figure 7.6. – Possible response signals

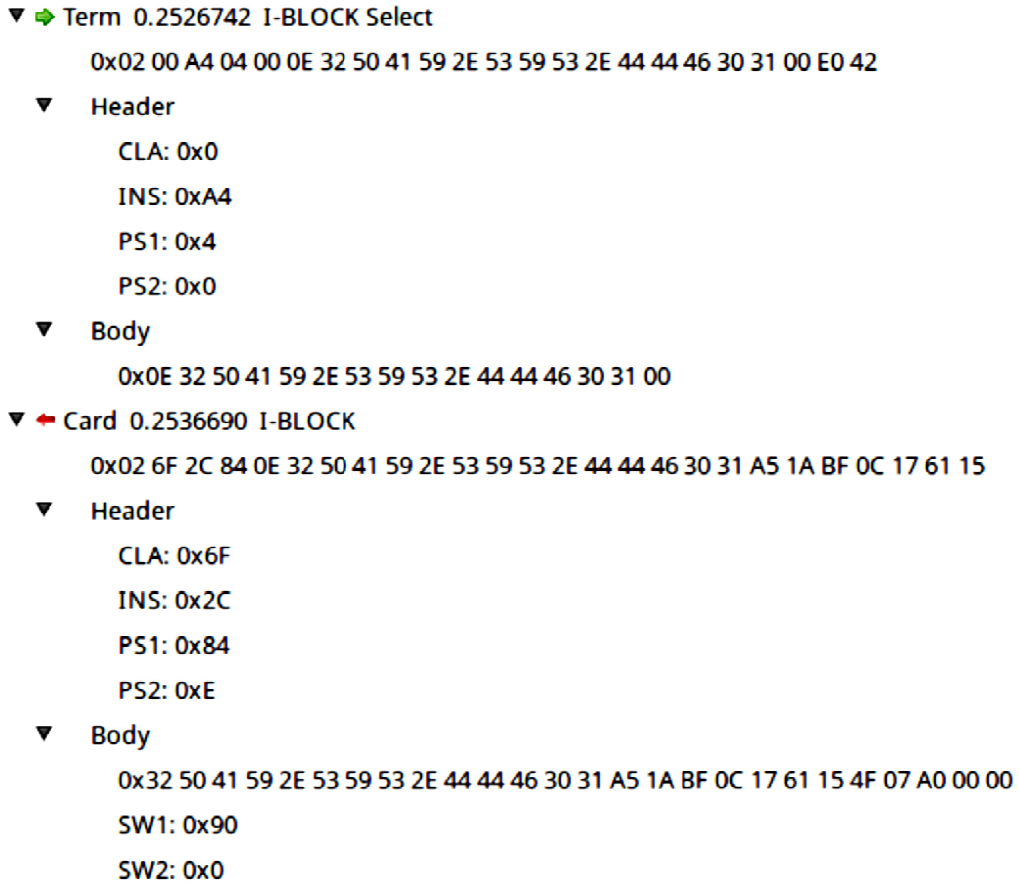


Figure 7.7. – Tree view example

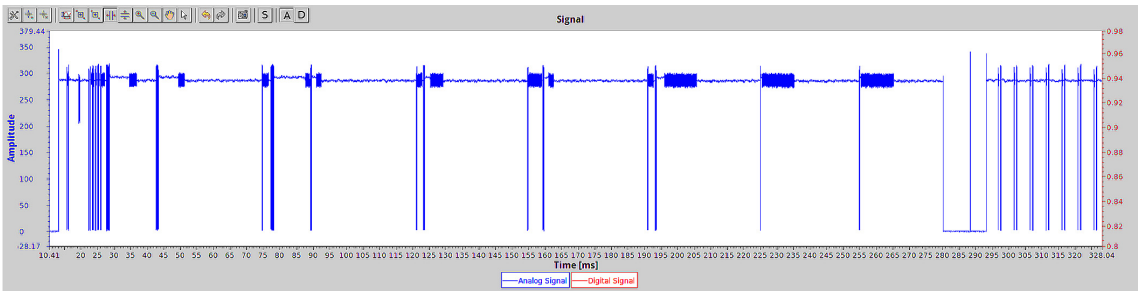


Figure 7.8. – Whole analog signal graph

Figure 7.9. – Graph toolbar

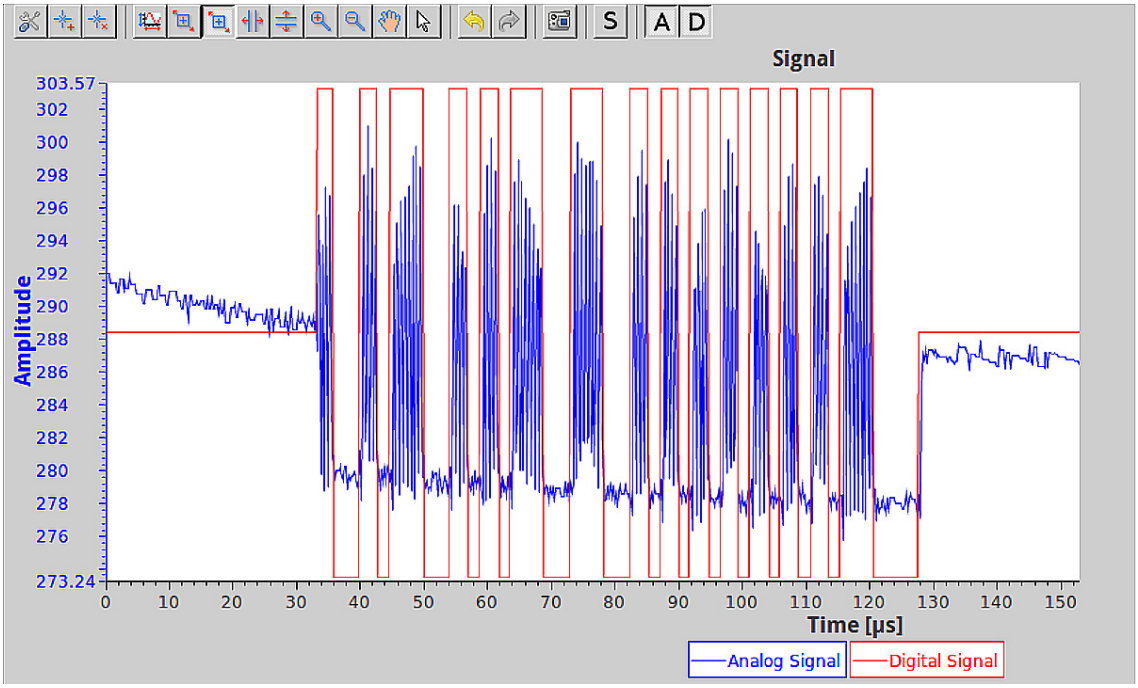


Figure 7.10. – Analog and Digital signal view

8. Evaluation

This chapter shows the measurement result from the NFC-sniffer comparing them with the values from the commercial system NFC Xplorer 100. To receive meaningful and comparable results the two systems captured the same communication in parallel. Overall this evaluation compares six captured communications with three various terminal reader. To test how robust the system is regarding the distance between sniffer-coil and PCD or PICC, two measurements with different distances were done. The two measurement setups are shown in figure 8.2. The first constellation is without any additional distance plate between sniffer-coil and terminal reader or payment card to test the ideal use-case. The second one shows the measurement setup with one distance plate between terminal reader and sniffer-coil and another one between sniffer-coil and payment card to test the robustness of the system. "Sniffer-coil 1" is a probe where the ground and the measurement pin is connected. The NFC-sniffer uses the antenna of the NXP-blue-board PNEV512B [4] shown in figure 8.1.
The next three sections describe the measurement results of the different terminal reader used for this evaluation.

Figure 8.1. – Used sniffer-coil

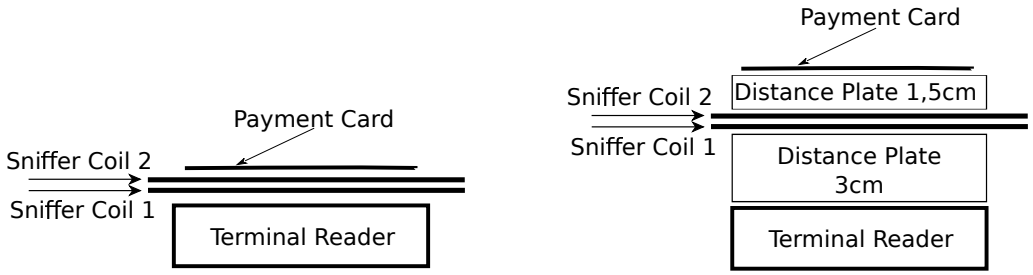


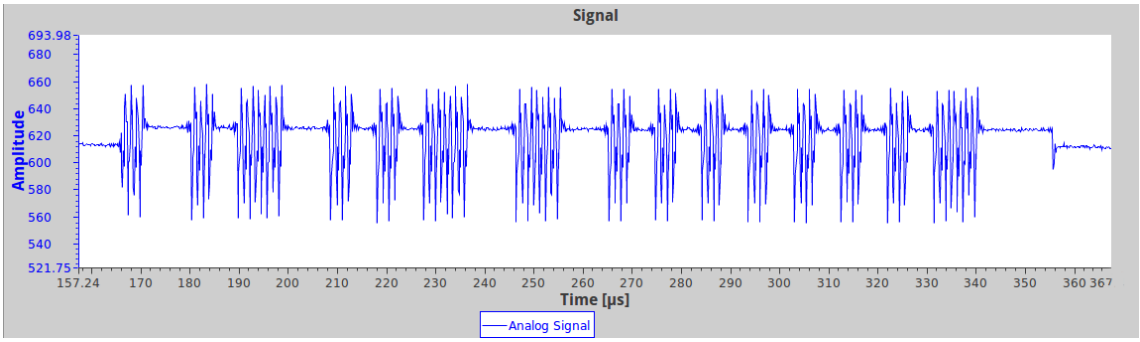
Figure 8.2. – Measurement setup

8.1. Reader CM100

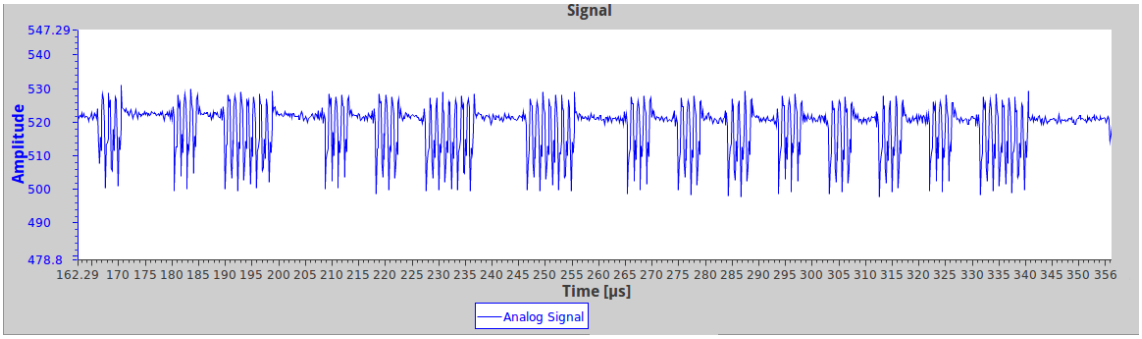
The terminal reader ID TECH Xpress CM100 was taken as first device for the evaluation because this reader was provided by CISC as test device for the research process. The setup for the first measurement is shown in figure 8.2 at the left side. The communication between the PCD and a standard payment card took about 300 ms therefore it was possible to use the trigger functionality. The maximum capturing time for the commercial system, implemented with LabView including the trigger function is only about 300 ms but even enough for this communication. Both systems, the professional and the NFC-sniffer could decode the whole captured signal and display the total protocol behavior of this communication.
To check if the NFC-sniffer is able to analyze a captured communication with a lower peak-to-peak amplitude the measurement setup at the right side of figure 8.2 is used. Again both system could decode and analyze the whole communication. Figure 8.3 shows the response for the first "WUPA" command. It is observable, that the peak-to-peak amplitude and the mean value of the response in figure 8.3b is much lower than in figure 8.3a. The results of this two measurements show that the maximum duration of the capturing time with trigger functionality is enough and the digitalizing software can also handle lower amplitude differences.

8.2. Reader QX1000

The two measuring setups used to test this reader remain the same as described in the previous section and shown in figure 8.2. To test this reader a different PICC was used because the PCD could not start a communication with the previously used payment



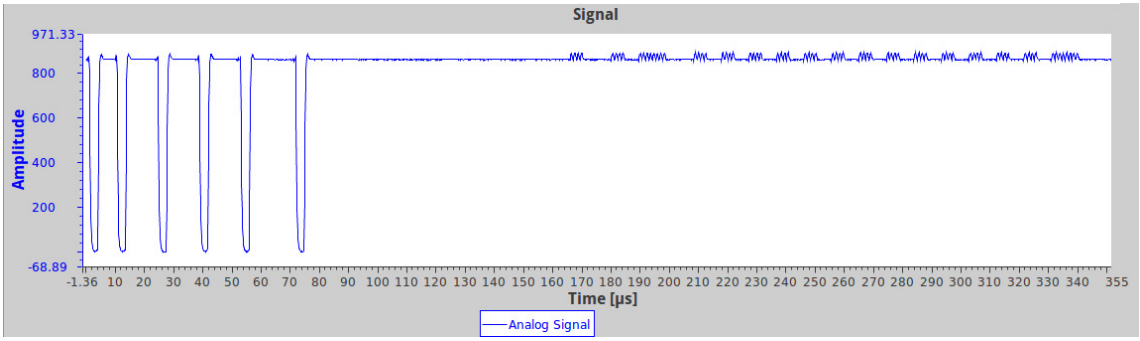
(a) ATQA command with no distance plates



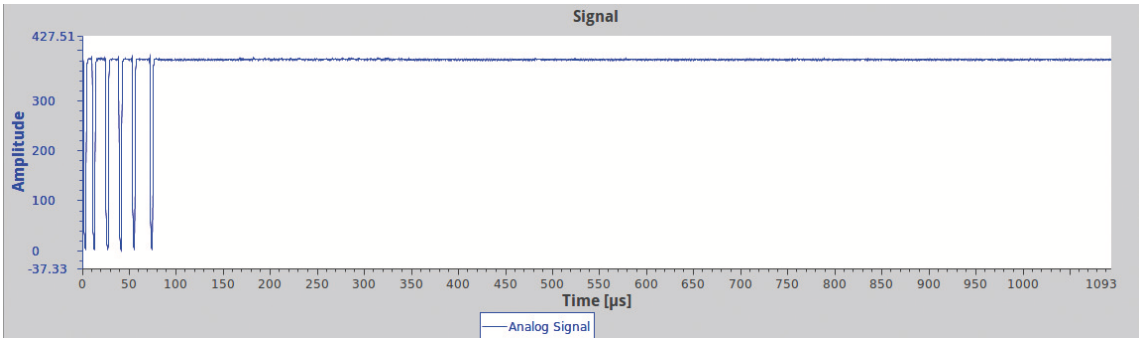
(b) ATQA command with inserted distance plates

Figure 8.3. – CM100-response to "WUPA" command

card. Using the measuring setup without any distance plates both systems could analyze the whole protocol correctly. Figure 8.4a shows the first "WUPA" command with the corresponding answer. The peak-to-peak amplitude of the response is relative low but observable. For the next measurement the distance plates were placed on the terminal reader and on the sniffer-coil as shown in the right part of figure 8.2. The commercial system was able to analyze the communication correctly. The NFC-sniffer however could only decode all request but no response. Figure 8.4b shows that the response is not even visible for human eye and therefore not possible to decode. The reason why the commercial system could capture the response signal could be either the different sniffer-coil which leads to another induced voltage or the more enhanced signal processing unit.



(a) Decodable small peak-to-peak amplitude response



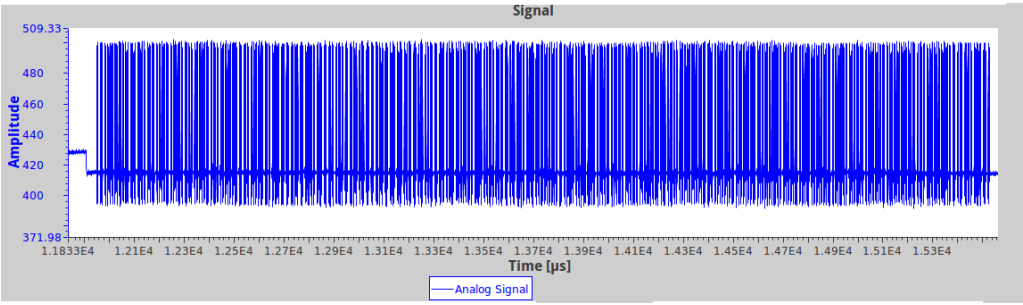
(b) Request with no visible response

Figure 8.4. – QX1000-"WUPA" command with corresponding response

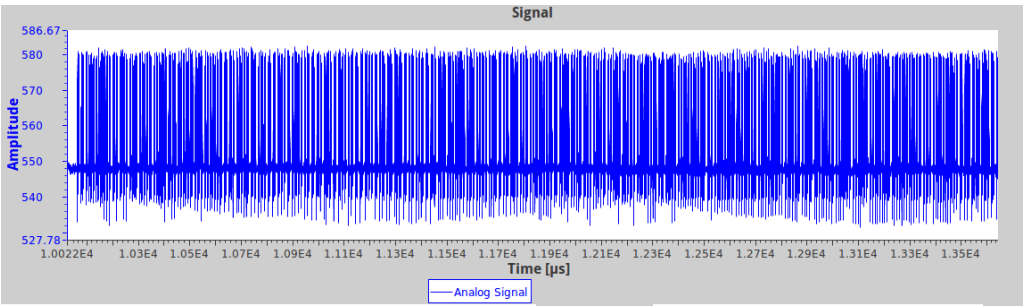
8.3. Reader QP1000SL

The last terminal reader used for this evaluation is the QP1000SL. The results of the NFC-sniffer with no distance plates in the measurement setup are not as good as the previously described. The NFC-sniffer was able to decode the whole communication except the four responses of the application layer. Figure 8.5a shows the first not decodable response. It is observable that the PICC had to transmit a lot of data and the signal level is good. The peak-to-peak amplitude is much higher than in previously successfully decoded signals. The result of a more accurate post-analysis found that the digitalizing function made some errors and therefore it is not possible for the Manchester-decoder to deliver the proper solution. It may well be that this error would not occur in some further measurements with the same setup.
The setup for the last measurement was a bit different than shown in figure 8.2 at the right side. The about 3 cm high distance plate between terminal reader and sniffer coil was removed in this setup because the PCD was not able to establish a communication with the PICC. To reduce the signal level of the responses the smaller distance plate

between sniffer-coil and PICC remains. Figure 8.5b shows the same response as figure 8.5a captured with the described changed measuring setup. It is observable that the peak-to-peak amplitude is only about the half. The mean value however is higher which means that the sniffer-coil changed direction and therefore the induced voltage. The results of the captured communication are good. Both systems, the commercial and the NFC-sniffer were able to analyze the whole communication even the four not decodable responses from the previous measurement. The communication between this reader and the PICC takes longer than 300 ms and therefore it is not possible to use the trigger function of the commercial system. With a second version of the program it is possible to capture the signal as long as enough free memory is available on the hard disk. After the capturing process the software provides the functionality to load data chunks with the size of 300 ms and analyze them. With the NFC-sniffer such a workaround is not needed and therefore an advantage regarding the usability.



(a) Not decodeable response



(b) Decodable response

Figure 8.5. – QP1000SL-response on application layer

8.4. Evaluation Summary

The evaluation shows that the NFC-sniffer is a competitive product to the professional NFC Xplorer 100. All the requests and the major parts of responses can be analyzed correctly. If the captured peak-to-peak amplitude is very low the enhanced signal processing unit in combination with the amplifier delivers better results. The commercial software is more robust regarding the digitalizing function as described in section 8.3 because the used tools are developed for signal processing tasks. The digitalizing function used by the NFC-sniffer however is written in Java which is definitively no professional signal processing software.
During the evaluation it becomes apparent that the trigger functionality is an important improvement of the system. Additionally the longer capture time of the NFC-sniffer in comparison to the commercial system is a great advantage. Another advantage of the NFC-sniffer is the tree view with the whole analyzed communication and the possibility to show the corresponding signal part by double clicking a request or response. This functionalities are not possible with the commercial capturing software. To receive the communication history another software analyzes the output file of the capturing program and displays them with some additional information equal to the tree view in the NFC-sniffer software.

9. Future Work

The prototype developed during this research is competitive regarding the cost effectiveness of the commercial system. As shown in the evaluation chapter it is possible to use this system to analyze the communication between various terminal reader and payment cards. Even if the system delivers good results some improvement tasks remain. Some of these tasks are described in the following chapter. There are certainly many more interesting topics in relation to this system but the following points came up during the project phase.

Decoder

As described in chapter 7 the decoder is one of the most complex parts of the software. The captured response signal has to have a specific form to be analyzed correctly. To avoid fail records the digitalizing software should be more robust and be able to convert signals even if the peak-to-peak voltage of the response signal is low. The difficult implementation task is to keep the calculation time for the digitalizing task as low as possible. To do that it is not possible to try several functions one after the other.

Capture time

As described in section 7.1 the maximum capture time is 3.355 s. This could be insufficient if the communication between PCD and PICC is unstable and full of retransmission because of transmission errors.

Data transfer

It should be possible to transfer the sampled data in real-time over the network to a PC. Therefore the server-software has to be implemented more efficiently. One possible solution is to store more samples on the FPGA and reduce the number of transfers between the FPGA BRAM and the server-software. Another good approach could be multi-thread architecture for the server-software to split the data loading task and the data sending task. If this feature is implemented successfully the maximum capturable signal duration would not be limited by the amount of free RAM on the RedPitaya anymore but by the free RAM on the PC. Since a PC has a lot more available RAM this feature would allow to capture a much longer communication duration.

Transmitter

Since the RedPitaya has not only RF-inputs but also two RF-outputs it should be possible to implement a transmitter. The RedPitaya should operate as terminal reader and as sniffer tool. With the combination of these two tasks it would be possible to test very specific use-cases without any other driver software for the terminal reader. This system would have many advantages if the aspect of portability is important because there is no need of further hardware. If however the functionality between a terminal reader and a PICC should be tested this functionality is obsolete and only payload for the software. Before implementing this feature it should be ensured that the remaining free space on the FPGA is sufficient.

Modulation types

In this master thesis only “Type A“ of payment card is discussed. In the future it would be impressive if the FPGA code could handle the second type ”Type B“ modulated with a PSK. To handle this type of modulation the digitalize software needs to be able to understand a new format of the incoming data. The decoding and protocol analyzing software remains the same. If it is possible to implement this feature the NFC-sniffer would be usable for a wider range of payment card technologies.

MAC-address

The user has to enter the IP-address of the RedPitaya each time the software starts. The IP-address may change every time the RedPitaya or the device with the DHCP-server restarts if it is not set statically. To avoid the additional work of finding the IP-address of the RedPitaya it would be possible to save the MAC-address and search the local network for the address. This would make the users life easier if the RedPitaya is connected to a big local network. If the RedPitaya is directly connected to a PC this feature is not that important but would still be helpful to have.

10. Conclusion

The aim of this research was to implement a NFC-sniffer by using the platform RedPitaya and to find restrictions by comparing it with the commercial device used by CISC. There should be some restrictions because of the lower price of the hardware. The result of this research shows that the RedPitaya is a good measurement instrument and as good as the professional one with some limitations regarding the signal modulation and protocol types. The evaluation in chapter 8 shows that the implemented system is able to decode most of the required communications of the ISO/IEC 14443 Type A standard. The implementation regarding the FPGA has significant advantages in contrast to the commercial system. The capturing time with the trigger functionality and the signal duration which can be analyzed by the software is higher. However the possibilities for detailed measurements of the signal are not as good as in the professional software. To conclude this thesis it can be said that the emerging prototype is a great success.

Bibliography

[1] M. Chinnathambi, N. Bharanidharan, and S. Rajaram. “FPGA implementation of fast and area efficient CORDIC algorithm.” In: *Communication and Network Technologies (ICCNT), 2014 International Conference on*. Dec. 2014, pp. 228–232. DOI: 10.1109/CNT.2014.7062760.

[2] *CISC Semiconductor homepage*. Nov. 23, 2015. URL: www.cisc.at.

[3] *Creator of the FPGA code and OS for this master thesis.* Nov. 23, 2015. URL: http://pavel-demin.github.io/red-pitaya-notes/.

[4] *Documentation of the NXP-blueboard PNEV512B*. Nov. 23, 2015. URL: http://www.nxp.com/documents/application_note/AN11308.pdf.

[5] *EMVCo Specification*. Nov. 23, 2015. URL: www.emvco.com/specifications.aspx.

[6] *Ettus Research homepage*. Nov. 23, 2015. URL: www.ettus.com.

[7] M. Gebhart et al. “From power to performance in 13.56 MHz Contactless Credit Card technology.” In: *Communication Systems, Networks and Digital Signal Processing, 2008. CNSDSP 2008. 6th International Symposium on*. July 2008, pp. 301–305. DOI: 10.1109/CSNDSP.2008.4610746.

[8] *General information about NFC*. Nov. 23, 2015. URL: www.nfc.cc.

[9] “Identification cards – Contactless integrated circuit(s) cards – Proximity cards.” In: ISO 14443. International Organization for Standardization, 2000.

[10] *Information about GNURadio*. Nov. 23, 2015. URL: www.gnuradio.org.

[11] *Information about I/Q samples*. Nov. 23, 2015. URL: http://www.ni.com/tutorial/4805/en/.

[12] *Information about Matlab*. Nov. 23, 2015. URL: de.mathworks.com.

[13] *ISO7816 Specification*. Nov. 23, 2015. URL: www.cardwerk.com/smartcards/smartcard_standard_ISO7816-1.aspx.

[14] M. Q. Kuisma. *Information about I/Q samples*. Nov. 23, 2015. URL: http://whiteboard.ping.se/SDR/IQ.

[15] *National Instruments homepage for detailed LabVIEW information*. Nov. 23, 2015. URL: www.ni.com/labview/d/.

[16] *RedPitaya hardware specifications*. Nov. 23, 2015. URL: https://www.dropbox.com/s/b2pxkwj1ljzyw71/Red_Pitaya_HW_Specs_V1.1.1.pdf.

[17] *RedPitaya homepage*. Nov. 23, 2015. URL: http://redpitaya.com/.

[18] *Xilinx homepage*. Nov. 23, 2015. URL: www.xilinx.com/fpga.

List of Tables

List of Figures

List of Acronyms

NFC	Near Field Communication
ISO	International Organization for Standardization
ADC	Analog Digital Converter
PICC	Proximity Integrated Circuit Cards
PCD	Proximity Coupling Devices
FPGA	Field Programmable Gate Array
DDC	Digital Down Converter
ASK	Amplitude Shift Keying
SoC	System on Chip
RF	Radio Frequency
OS	Operating System
RAM	Random Access Memory
USB	Universal Serial Bus
TCP/IP	Transmission Control Protocol/Internet Protocol
SSH	Secure Shell
RFID	Radio-Frequency Identification
DC	Direct Current
OOK	On-Off keying
CORDIC	Coordinate Rotation Digital Computer
CIC	Cascade Integrator Comb
FIR	Finite Impulse Response
CRC	Cyclic Redundancy Check
UID	Unique Identifier
CPU	Central Processing Unit
LUT	Look Up Table
DSP	Digital Signal Processor
USRP	Universal Software Radio Peripheral

I/O	Input/Output
ASIC	Application Specific Integrated Circuit
CLB	Configurable Logic Block
MUX	multiplexer
SATA	Serial Advanced Technology Attachment
IDC	Insulation Displacement Connector
BRAM	Block RAM
AM	Amplitude Modulation
GUI	Graphical User Interface
PSK	Phase Shift Keying
LF	Low Frequency
HF	High Frequency
UHF	Ultra High Frequency
IC	Integrated Circuit
GSM	Global System for Mobile Communications
GPRS	General Packed Radio Service
WCDMA	Wideband Code Division Multiple Access
HSPA	High Speed Packed Access
LTE	Long Term Evolution
WiMAX	Worldwide Interoperability for Microwave Access

Appendix

A. Class Diagram

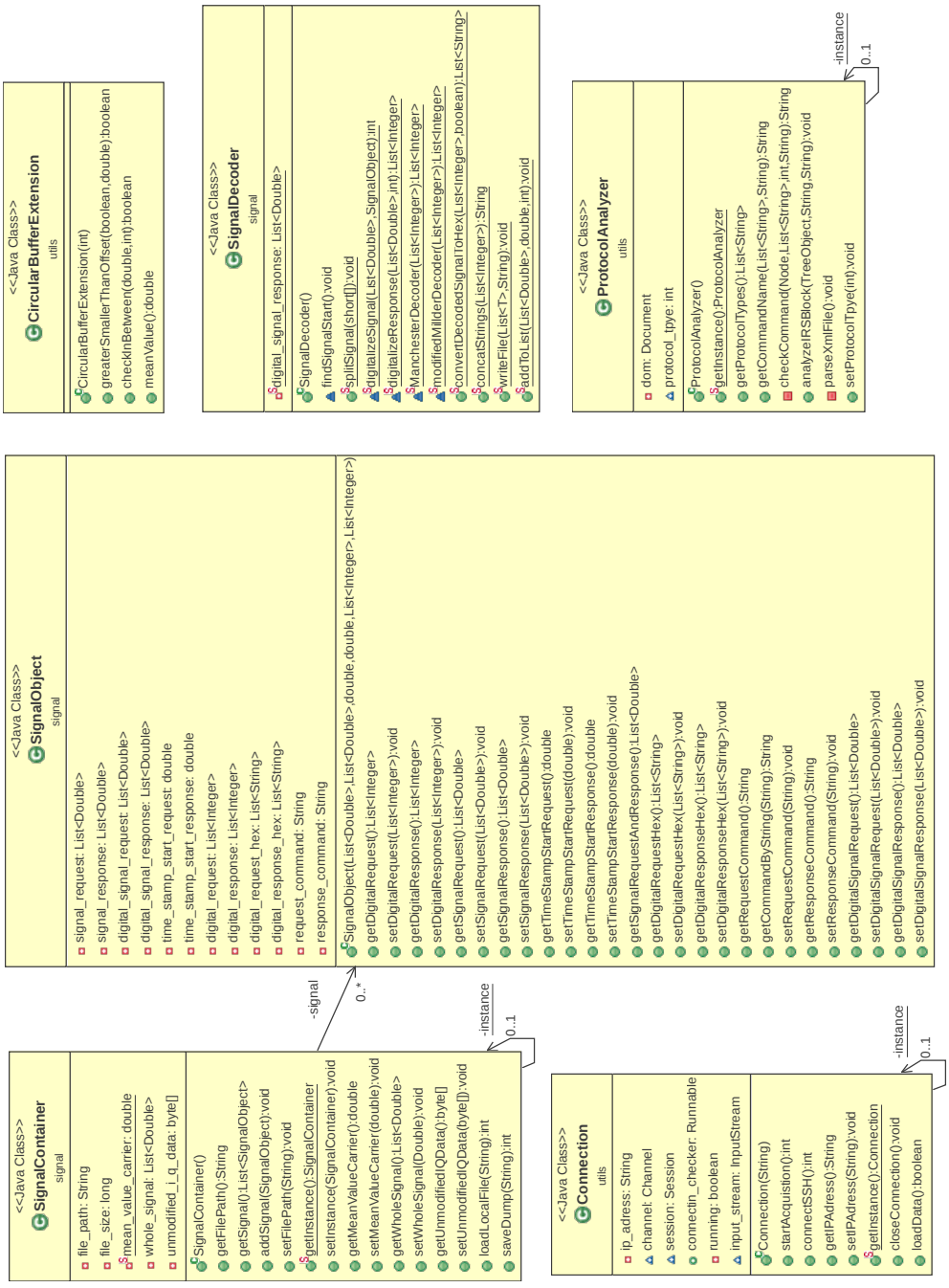


Figure 10.1. – Decoder software class-diagram

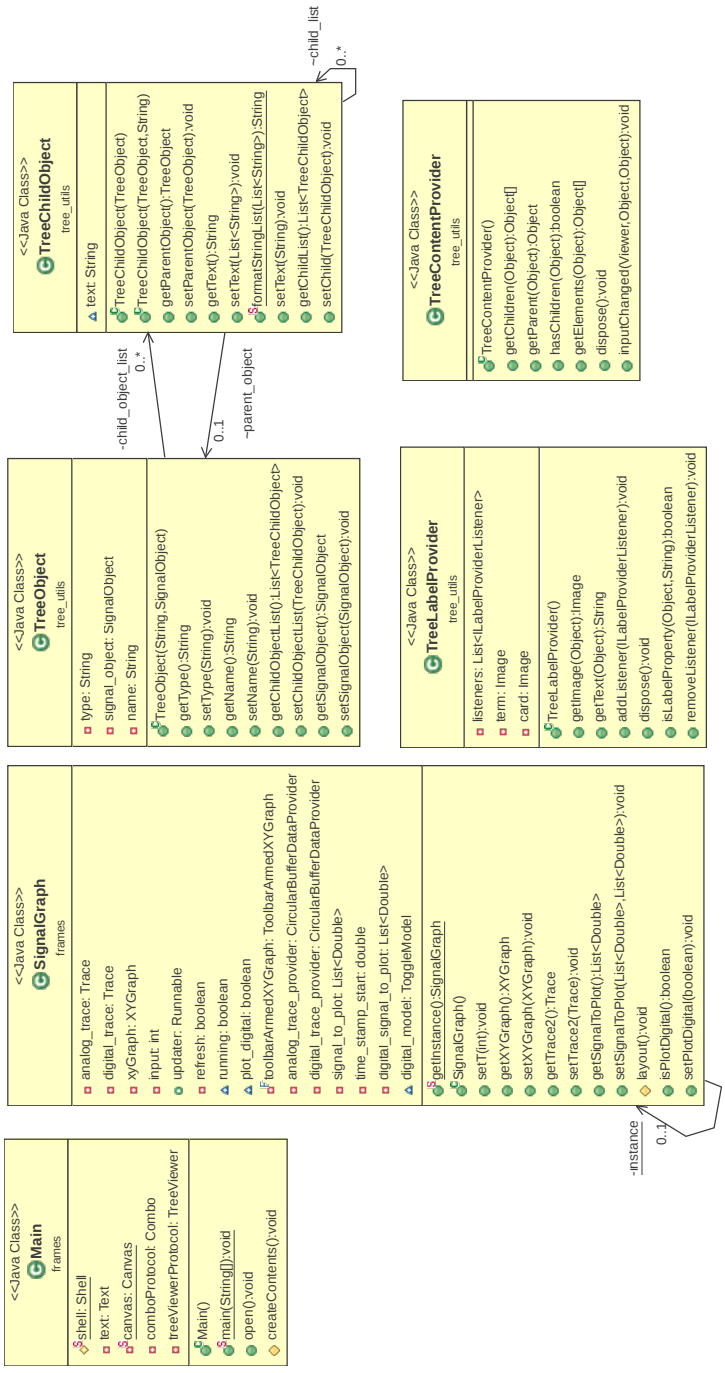


Figure 10.2. – GUI class-diagram