Trigger Alarm: A Smart NFC Sniffer for High-Precision Measurements

Martin Erb, Christian Steger, Martin Troyer, Josef Preishuber-Pflügl

Trigger Alarm: A Smart NFC Sniffer for High-Precision Measurements*

Martin Erb[1], Christian Steger[1], Martin Troyer[2], and Josef Preishuber-Pflügl[3]

[1] Institute of Technical Informatics, Graz University of Technology
{martin.erb, steger, martin.troyer}@trugraz.at
[2] CISC Semiconductor GmbH, Lakeside B07, 9020 Klagenfurt, Austria
https://www.cisc.at
j.preishuber-pfluegl@cisc.at

Abstract. In this paper, we present a new design and proof of concept of a smart Near Field Communication (NFC) sniffer, including special trigger features for high-precision measurements during NFC interoperability testing. Even though interoperability testing is not mandatory for successful NFC-device certification, the fast increasing amount of electronic consumer devices providing NFC functionality strongly increases the need for interoperability testing. Nowadays, used automated interoperability test systems require time-consuming and expensive manual debug sessions in case of a communication error, to compensate missing test data for analyzing the failure root cause. To highly decrease costs and time required to perform these manual measurements, we developed a proof of concept of a sniffer tool providing intelligent trigger functionalities. It supports the test engineer during manual debug session and can be integrated into a fully automated interoperability test and analysis system. Hence, we drive the development of automated interoperability test systems and want to encourage standardization bodies to include interoperability testing to the certification procedure.

Keywords: Near Field Communication, Software Defined Radio, Measurement System, Test Platform, Interoperability

Near Field Communication has experienced a steep marked ramp up in recent years. The rapidly increasing amount of mobile phones pushed to the global market led to the increasing integration of NFC technology in consumer electronic devices. NFC provides a convenient, easy, and secure possibility to transfer wireless data between two electronic devices [6,12]. Integrating NFC, which has been evolved from the Radio Frequency Identification (RFID) technology, to smartphones enables the possibility to replace identity documents for access control

* This work is part of the ANITAS project in cooperation with CISC Semiconductor, NXP Semiconductors, and the Institute of Technical Informatics at the Graz University of Technology. This project is partly founded by the Kärntner Wirtschaftsförderungsfonds and the Steirischen Wirtschaftsförderungssgesellschaft mbH under the FFG (Austrian Research Promotion Agency) grant number 864323.

or debit cards for payment [17]. Furthermore, NFC allows initiating device connectivity such as Bluetooth or WLAN, thus allowing the user to easily exchange security-sensitive data between their devices [3, 7, 9, 18, 25].
To keep the NFC technology on the course of further success and to enlarge the influence to the market, the user acceptance needs to be increased. To achieve this and to encourage more people to use NFC-enabled devices for a wide range of different applications in their daily life, interoperability needs to be guaranteed. The ultimate goal is to achieve full interoperability among all devices from various manufactures. To reach this goal, the NFC Forum follows the approach of harmonizing the different standards and ensuring interoperability by extensive conformance testing [10, 23, 24].
However, interoperability testing cannot entirely be replaced by conformance testing because during the certification process one communication party is always simulated. Additionally, the setup for conformance testing has to be entirely defined to make the test procedure reproducible [13]. This leads to the drawback that conformance testing cannot represent all real-world scenarios and misses to include many user requirements [14, 15]. The problem of missing user requirements is shown in various current systems where certified products have interoperability lacks [1].
In contrast to other wireless technologies where interoperability issues may occur due to interference [21], such as Bluetooth or WiFi or test systems where the physical layer is excluded from testing [26], NFC interoperability test systems have to analyze the whole data transfer across all communication layers. CISC Semiconductor, a leading supplier of RFID measurement and test solutions, developed an automated interoperability test system to gain significant speedup for testing [2]. In the case of a communication error, however, manual debugging with different additional measurement and analysis equipment is required. We analyzed projects over the last years from our research partners (CISC Semiconductor and NXP Semiconductors) and found out that setting up the measurement devices and performing additional communications to detect the error root cause consumes a massive amount of the overall test time.
Lack of all-in-one solution. As defined by the NFC test standard [13], the sample rate of the measurement device to properly analyze communication parameters must be at least 500 MS/s. Existing test platforms such as presented by the works [5, 11, 16] have implemented protocol decoder and further analysis tasks based on the baseband signal. They use the baseband signal because the required bandwidth is only 847 kHz (the subcarrier generated by the Proximity Inductive Coupling Card (PICC)) plus the data rate such as 106 kHz used for payment applications. Therefore, a digital sample data rate of about 6 MS/s is sufficient to decode the protocol. Such measurement devices are essential and very useful for interoperability error debugging; however, not all communication relevant parameters can be analyzed. Therefore, an additional oscilloscope is still required, with which a test engineer can capture parts of the communication with a sufficiently high sample rate. To capture exactly the appropriate part of the communication mostly requires multiple tries to set up the oscilloscope.

The time to set up the measurement system correctly, perform the measurements again, and analyze the results prolongs the debug time and makes interoperability testing very expensive.

Oscilloscope for specific parameters. The trigger functionalities of an oscilloscope are sufficient to analyze simple parameters like the field strength or modulation index. For such measurements, no specific parts of an NFC communication are required, which means that one configuration of the measurement is sufficient, and only one communication needs to be performed per measurement point. In work [19] they present an automated test system with various measurement devices to analyze interoperability problems. In this case, the oscilloscope is used to measure the high frequency (HF)-field strength. Nevertheless, the system can analyze only the HF-field strength automatically. Otherwise, manual interaction from a test engineer would be required to reconfigure the oscilloscope, which would again increase the test and measurement time.

Lack of intelligent trigger features. During conformance testing, one communication partner is always simulated, meaning the test controller is generating either a Proximity Coupling Device (PCD) command or a PICC response. Therefore, it is straightforward to control a high precision analog measurement device with a high sampling rate, such as an oscilloscope, to capture exactly the communication parts required to check if the Device Under Test (DUT) fulfills the requirements defined by the standards. This is not true for interoperability testing where both communication partners have finalized products and have to be handled as black-box. Additionally, interoperability testing mostly requires to analyze a specific command or response within the whole communication. However, it is time-consuming to configure the oscilloscope each time to capture exactly the required command or response using a sample rate of at least 500 MS/s. The high sample rate prevents the user from capturing the whole communication. Furthermore, the time between the start of the communication and the occurrence of a certain command may differ between multiple test runs due to re-transmissions. To the best of our knowledge no analog measurement system including such features to trigger an external high-precision oscilloscope exists.

Our contribution. In this paper, we present a smart sniffer tool providing intelligent trigger functionality for NFC interoperability testing based on software-defined radio (SDR) platform. The sniffer tool presented does not close the gap of an all-in-one solution but implements intelligent trigger features for easy communication capturing using an external oscilloscope. Precise and easy triggering at the position before an error happens or where additional investigations should be made is essential to reduce the time for a manual debug session significantly. Therefore, we implemented three options for the communication types described in ISO-14443, which apply rising edges on different general-purpose input outputs (GPIO): (i) HF-field on, (ii) every command typeA or typeB, and (iii) selected command typeA or typeB. In Section 3, we present how the different trigger functionalities can be used to achieve significant speedup during testing. In this paper, we make the following contributions:

– We design three different trigger functionalities not yet used for interoperability testing to reduce the manual debug session duration (Section 1).
– We implement the designed features based on an SDR platform, the LimeSDR (Section 2).
– We experimentally show the functionalities and the performance of the three different trigger features (Section 3).

After describing related work in Section 4 we conclude our paper in Section 5, along with a discussion on future work.

1 Trigger Design

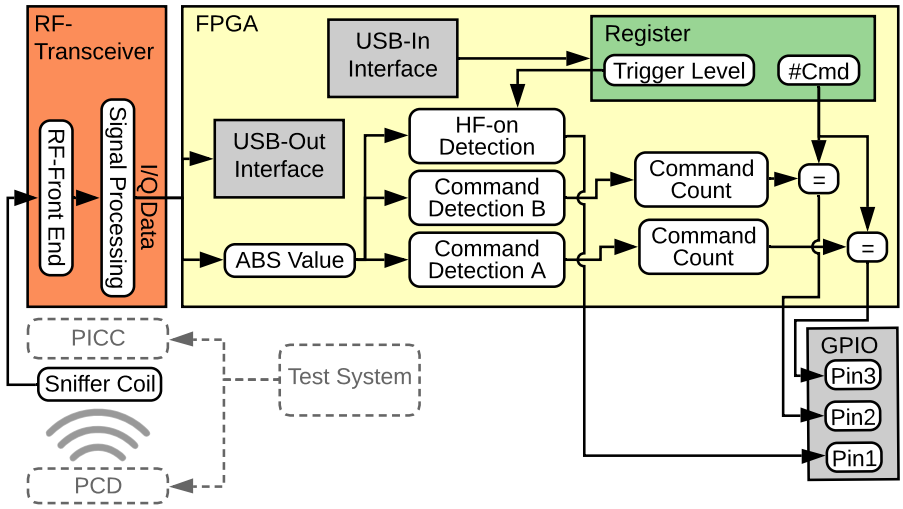

Fig. 1: Block Diagram of the FPGA Implementation

As a basis for our Field Programmable Gate Array (FPGA) design, we used the work [5] of the researcher from the University of Technology Graz. In their work, they implemented a simple HF-field on trigger using the RedPitay platform [28]. Our design is not restricted to any hardware platform. However, the input to the trigger block is the absolute value of the baseband of the captured communication. Nevertheless, it makes sense to include the trigger functionality to a similar measurement device as the RedPitaya. Figure 2 shows a simplified version of the design of the trigger functionality. The green blocks show the input values enabling the possibility to configure the behavior of the trigger. The HF-on block and the Command Detection block show the two main components of the trigger. We split up the FPGA and the RF-transceiver because different FPGA hardware platforms may not include an RF-transceiver but, for sure multiple connection possibilities for an RF daughterboard.

HF-on trigger. The HF-on trigger is the simplest version of a trigger and could also be realized using an oscilloscope. The HF-on trigger outputs a single rising edge on a GPIO pin when the absolute value exceeds the trigger level the first time after the user armed the trigger. The trigger level is configurable by the user. Implementing this simple trigger feature enables the possibility to synchronize the time between the captured I/Q data by the measurement device and the signal captured by the oscilloscope. This can be useful for further signal analysis tasks. Additionally, this trigger is internally connected to the streaming block. Meaning, the host PC starts receiving I/Q data from the measurement device when the HF-on trigger fires. This is useful during testing because, in most cases, a test engineer is not interested in the signal before a communication start.

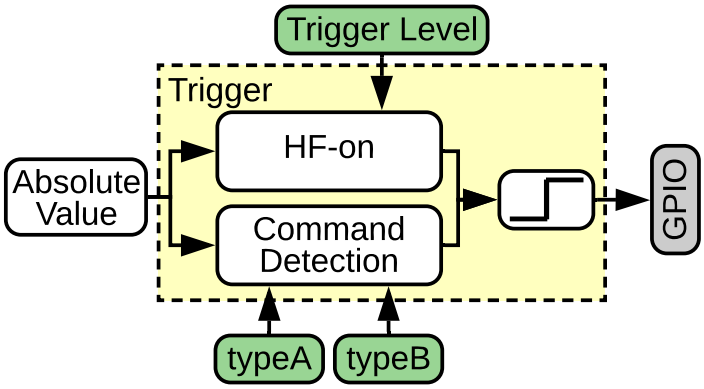

Fig. 2: Block Diagram of the FPGA Design

Command based trigger. The second trigger functionality is based on command recognition. The user can configure the communication type and the number of the command when the trigger should fire. For simplicity, only the number of the command can be defined because triggering on a specific command name would require a complete protocol decoder implemented on the FPGA. The command detection block is responsible for finding all commands, either typeA or typeB, within the absolute value of the baseband signal and counting them. The trigger fires whenever the number of found commands of one single communication type is equal to the configured value by the user. If the user does not define any number of commands to trigger on, typeA and typeB trigger fire on every detected command.
The command base trigger feature is handy and time saving for a test engineer when it comes to capturing a specific command within a communication. If an interoperability issue should be analyzing the error mostly happens in the same position within the protocol, but not at the same time after the communication start. Therefore, a standard oscilloscope trigger would not be sufficient to always trigger at the correct point in time. The functionality where every command

triggers a rising edge significantly simplifies error cases where either the PCD waits for a response of the PICC or the PICC does not understand the command. In these cases, the oscilloscope shows the last command sent by the PCD and can easily be analyzed by a test engineer.

2 Implementation

2.1 Hardware

Before thinking of implementation details, we used the design described in Section 1 to find a suitable hardware platform to implement our smart sniffer. As mentioned above, the absolute value of the baseband signal is required as input for the trigger function. NFC technology uses a 13.56 MHz carrier frequency and a sub-carrier with a frequency of 848 kHz. For our implementation, we are focusing on the payment application, which supports a data rate of 106 kBit/s, but the hardware should support higher data rates too. During our research, we tried to focus on only a few important hardware requirements: (i) Radio Frequency (RF)-front end capabilities such as center frequency, bandwidth, and sample rate, (ii) FPGA specification, (iii) configurable GPIOs, and (iv) connection interface to the host PC. Because we are using only the baseband of the signal, we decided to use a sample rate of about 10 MS/s for the I/Q data transmitted to the host PC. Therefore, the data throughput of the connection interface between measurement hardware and host PC is not critical, and a lot of different hardware platforms are available. After our evaluation phase, we decided to use the LimeSDR from Lime Microsystems [20] in the USB configuration. The advantage of this hardware platform is the powerful RF-transceiver chip, which includes already highly configurable analog and digital signal processing units. These units enable the possibilities to shift the signal to the baseband and to apply the required filters to have clean I/Q data ready even before sent to the FPGA. Using such a powerful RF-transceiver chip saves many hardware resources on the FPGA for other tasks. The FPGA provides enough computational power and memory resources to implement the trigger features described in Section 1. The LimesSDR provides a USB 3.0 interface to connect the host PC. The data throughput of this interface is far enough to transfer the I/Q data shifted to the baseband.

2.2 Software

Figure 1 shows the high-level block diagram of our FPGA implementation. The sniffer coil is connected to the RF-front end and placed between PICC and PCD. The signal processing units of the RF-transceiver convert the captured communication to the digital domain and shift the signal to the baseband. Furthermore, the RF-transceiver applies the corresponding filters and sends the I/Q data to the FPGA. Further, the FPGA sends the captured baseband I/Q data to the USB-Out interface and computes the absolute value of the signal. As shown in

Figure 1, the calculated absolute value is than used as input for the three different main functional blocks of the FPGA implementation: (i) HF-on detection, (ii) command detection typeA and typeB, and (iii) command count.
The HF-on detection block compares the received absolute value with the trigger level defined by the user using the configuration register. The first time the absolute value exceeds the threshold, the HF-on trigger fires a rising edge on GPIO Pin1. Additionally, this trigger signal is used to start the I/Q data streaming to the USB-Out interface, more precisely to the host PC. This feature gives a lot of flexibility during testing. It reduces the data to be transferred because the unwanted signal before a communication start is not sent to the host PC. Furthermore, this trigger allows a precise synchronization between captured I/Q data on the host PC and the recorded data on an external oscilloscope.
The command detection blocks for typeA and typeB use a min-max method with fixed thresholds based on the modulation values defined in the standard to detect rising edges within the received signal. Using the peak-to-peak values and the frequency of the rising edges, the detector blocks check if the absolute value belongs to a request sent by the PCD or not. If one of the detection blocks identifies a command, it increases the command count by one. The command count block counts the number of commands for a single type, which has been decoded after the acquisition has been started. The count of recognized commands is further compared with the number defined by the user in the configuration register. If the number counted by the command count block is equal as configured by the user, GPIO Pin2 is set to high for typeB and GPIO Pin3 for typeA, respectively. If the user configures the number of commands with zero, a short peak is applied to the corresponding GPIO pin whenever a command is detected. For our use case, which is the payment application, the data rate is always 106 kBit/s. Therefore, the communication between PCD and PICC takes place via Amplitude Shift Keying (ASK) 10 % amplitude modulation of the RF operating field for communication typeB and around 100 % for typeA. This is not true for data rates higher than 424 kBit/s for which the thresholds for the min-max method need to be adapted.
As depicted in Figure 1 the different trigger features are implemented in parallel. This means that the HF-on trigger can be used simultaneously to the different command detection features. Therefore, a test engineer can configure the oscilloscope in a way that triggers first on the HF-on trigger to show the start of the communication and later trigger again if a certain amount of commands were detected. This increases the flexibility of performing manual measurements using an oscilloscope and reduces the amount of required additional communications because multiple events can be observed at once.

3 Evaluation

We used the design described in Section 1 to implement the functionality as presented in Section 2 and performed an evaluation shown in this section. We used one PCD and two PICCs to show the trigger functionalities for communication

typeA and typeB. The three NFC-enabled devices implement the payment application according to the EMVCo standard. Therefore, they are communicating at 106 kBit/s. As shown in Figure 3, we placed the sniffer coil, which is connected to the LimeSDR and the oscilloscope, between PCD and PICC. The LimeSDR was connected to the host PC, which was responsible for configuring the device, controlling the measurement procedure as well as starting the communication on the PCD. Furthermore, we connected the three different GPIO pins of the LimeSDR to the remaining analog input channels of the oscilloscope. The trigger of the oscilloscope was configured in single-shot mode using one of the analog inputs as a trigger source. The sample rate of the oscilloscope was set to 2 GS/s.

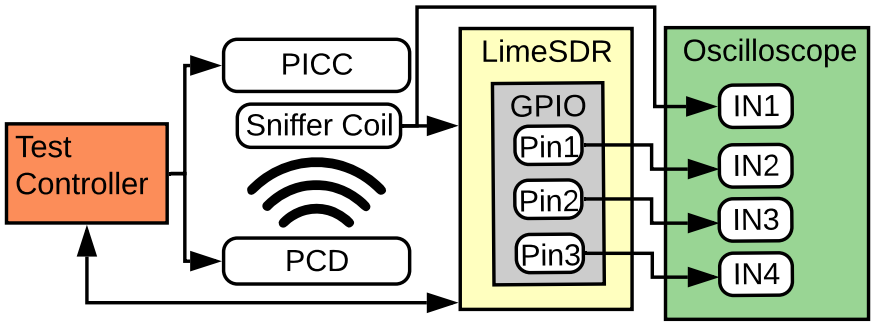

Fig. 3: Evaluation Setup

Figure 4 shows the beginning of an NFC communication between PCD and PICC using communication typeB. One can see the four captured signals by the oscilloscope: (i) *HF* signal capture by the sniffer coil, (ii) *typeA* trigger signal for typeA commands, (iii) *typeB* trigger signal for typeB commands, (iv) and *HF-on* trigger signal when the HF-field exceeds the configured threshold. The fifth signal (*SDR*) shown in Figure 4 is the absolute value captured and processed by the LimeSDR. The LimeSDR was configured to trigger on HF-on and on the second typeA and typeB command. The five red marked frames show three zoom regions, explained more detailed later, and the shape of the commands for the different communication types.

The default activation procedure for payment transactions starts with the HF-field activation by the PCD shown in the first red marked frame *Zoom 1*. Afterward, the PCD tries to activate the card using a typeA command, as shown in the second red-marked frame *TypeA cmd 1*. The activation was not successful because we used a typeB only PICC, and therefore, no typeA response can be found. Further on, the PCD tries to activate the card using a typeB command, as shown in the third red marked frame *TypeB cmd 1*. This time one response after the command can be found. The default procedure of the PCD is to check again if a typeA card is present. Therefore, a second typeA command is sent thus marked with a red frame *Zoom 2*. Since the PICC did not respond to the second

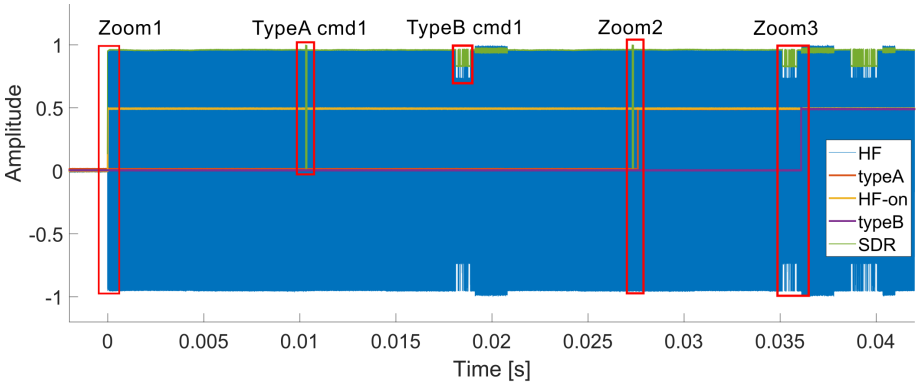

Fig. 4: TypeB Communication with the different Trigger Features

typeA command, the PCD starts with the anti-collision and further application-layer communication using typeB. The last red marked frame *Zoom 3* shows the second typeB command of this communication.

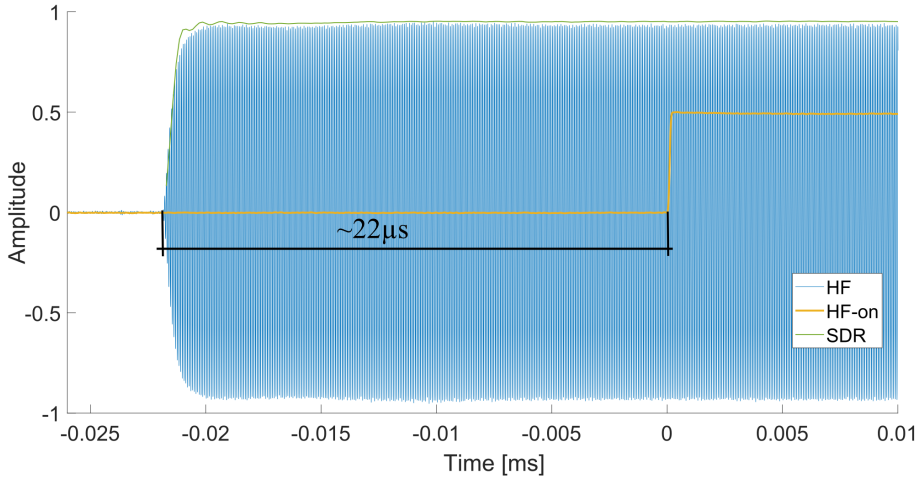

Fig. 5: TypeB Communication showing Zoom 1

Figure 5 shows a close-up of the HF-on trigger functionality more precisely the *Zoom 1* frame in Figure 4. On the time axis, one can observe that the oscilloscope triggered on the rising edge of the HF-on signals generated by the LimeSDR because time equals zero. The time delay between actual HF-field on captured by the oscilloscope and the rising edge of the signal generated by the LimeSDR of about 22 µs is introduced by the processing time of the implemented detection algorithm and the time required to control the GPIO pin. Nevertheless,

this time delay is constant and thus can be used to synchronize the captured signal from the oscilloscope and the LimeSDR. All the different figures show the SDR signal synchronized to the HF signal using this time delay. As explained in Section 2, the streaming of the I/Q data to the host PC is started at the same time when the HF-on trigger fires. Figure 5, however, shows that the SDR signal (depicted in green) starts about 22 µs earlier. This feature was set as an additional requirement not to lose any important information and was realized using ring buffers for data streaming to the host PC.

Figure 6 shows an enlarged view of the LimeSDR command trigger for typeA. One can see that the rising edge of the command recognition appears after around 220 µs. The reason, therefore, is, on the one hand, the implemented command detection algorithm, which analyzes the whole command and waits until the command is finished to reduce the probability of wrong command detection. On the other hand, the ring buffer, which has to be bigger than for the HF-on trigger, and the overall processing time of the LimeSDR, is larger. The constant time shift between command end and the rising edge of the trigger can be ignored because conventional oscilloscopes have a changeable pre-trigger time. This measurement shows that a user can trigger very precisely on a predefined command even if the between communication start and the specified command changes.

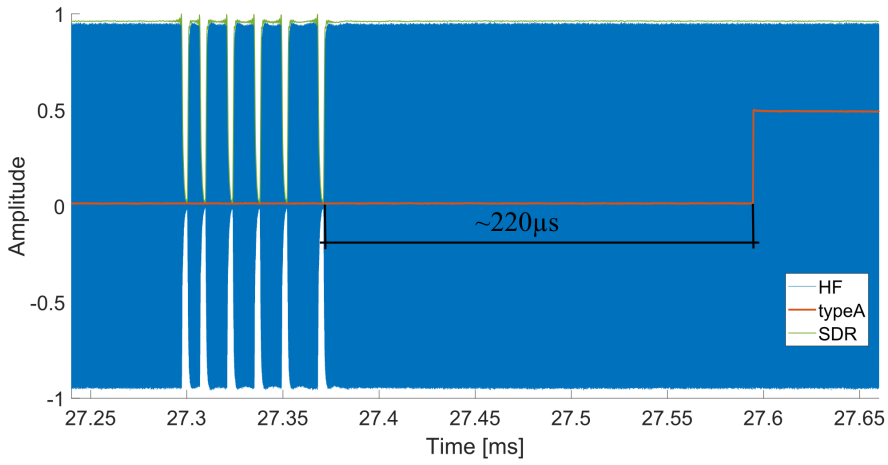

Fig. 6: TypeB Communication showing Zoom 2

The last red marked frame in Figure 4 shows the second typeB command, specifically the command the LimeSDR triggers on. Figure 7 displays a close-up of the mentioned frame. One can see that the rising edge of the trigger signal from the LimeSDR is again 220 µs after the end of the command. This has, of course, the same reasons as explained in the previous paragraph. However, we

tried to keep the time delay for both command triggers as similar as possible to ensure that the test engineer can use the same oscilloscope settings independent of the communication type.

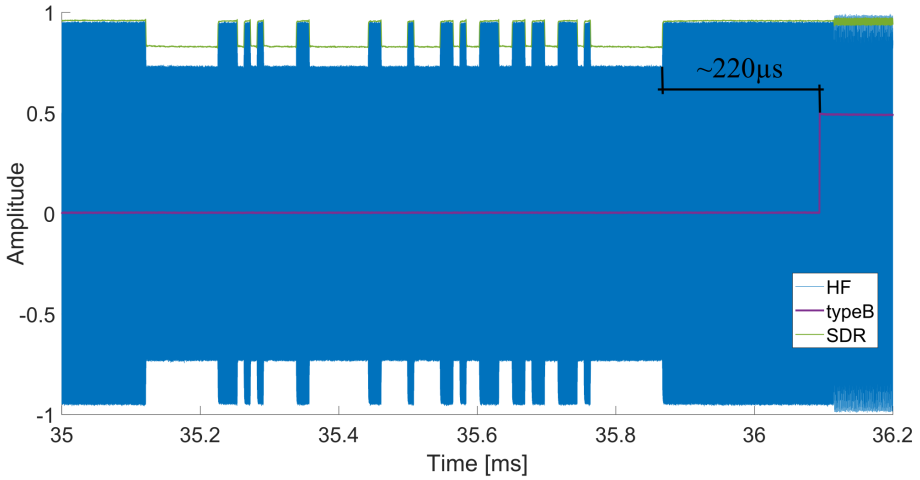

Fig. 7: TypeB Communication showing Zoom 3

Figure 8 shows the last missing trigger functionality, applying a short peak to the appropriated GPIO for every command within the whole communication. For clarity reason, the HF-on trigger is not shown in this figure, but it can be used in parallel for this configuration case as well. This figure shows a typeA communication to prove further that the trigger functionalities work the same way for typeA communications. Observing Figure 8, one can see a short peak after each command of typeA and typeB captured by two different input channels of the oscilloscope. This shows that the test engineer can decide to configure the oscilloscope to trigger either on typeB or typeA commands. Using this feature, it is possible to quickly capture the last command independent of the length of the communication.

Figure 4 in combination with the three close-up figures 5, 6, and 7 shows that the three trigger functionalities work in parallel. Furthermore, it is proven that triggering on a specific command within a communication is possible. All these features enable the possibility of convenient high-precision communication snippet capturing using an external oscilloscope and, therefore, reduce the time for manual debug sessions.

4 Related Work

In the field of NFC testing, a lot of research already exists. Research dealing with measurement methods of various communication parameters, design, and imple-

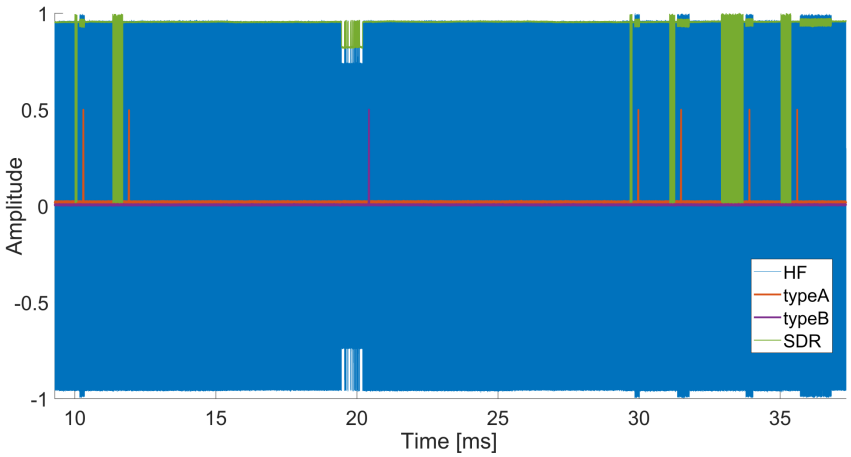

Fig. 8: TypeA Communication Trigger on all Commands

mentation of test systems focusing on different NFC applications, and design of measurement devices to be as accurate as possible. Furthermore, the development of automated conformance and interoperability test systems is ongoing.

Baseband measurement systems. For interoperability testing, first of all, it is essential to analyze the whole communication in general. However, for some specific communication relevant parameters, the baseband signal is not sufficient. The required information is lost during the mixing and filtering process. In the works [5,11,16], different hardware, and measurement methodologies are used to analyze the communication between two NFC-enabled devices regarding interoperability issues. These works have one thing in common: the complete signal analysis and error evaluation are based on the baseband signal. This can be very useful to find protocol errors or to analyze which communication partner led to the communication fault. Furthermore, it is helpful to find the position of the error within the communication quickly. However, it is not suitable to analyze specific parameters that require an exact representation of the HF-carrier signal of the PCD.

To find all possible interoperability issues a high-precision oscilloscope is essential. Therefore, the designed trigger functionality presented in this paper could be used to improve the already existing functionality presented in [5, 11, 16] as long as the used hardware provides configurable GPIOs and enough computation power.

Special setup. In contrast to conformance testing, where a well-defined measurement setup is required, this is not true for interoperability testing. The standard ISO/IEC 10373-6 [13] defines this specific setup, and it is taken over by the NFC-Forum in their analog parameter comparison and alignment [24]. Since interoperability testing is performed as black-box testing with finished products, the specific setup and most of the given specifications of the various standards cannot be applied anymore. Furthermore, the algorithms for many

parameter computations have to be adjusted [22,27]. The presented approach in work [4] implements some of the specific parameter computation but again uses the specific setup. Furthermore, the transmitter is included in the FPGA implementation, which simulates one communication partner. This makes it very easy to capture only a snipped of the communication because the FPGA well knows the point in time when a command is sent. However, simulating one communication partner and not using two complete NFC-enabled devices is not valid for interoperability testing.
To get the specific snippet of a communication to analyze parameters such as described in [8,22,27] a measurement system with smart trigger functionality is essential.
High resolution but limited parameter. For some NFC communication relevant parameters, a standard oscilloscope without intelligent trigger functionality is sufficient. As shown in work [19], the HF-field of the DUT can easily be measured without analyzing a command or a response. Therefore, such a test system is suitable and can be used to find some interoperability issues if they are related to the HF-field strength.

5 Conclusion and Future Work

In this paper, we present an NFC interoperability sniffer tool providing smart trigger functionalities to reduce test time during manual debug sessions. Furthermore, we provide a solution to capture NFC communication frames with very high sample rates. Before elaborating on the design of the sniffer tool we point to the importance of interoperability testing and the need for an additional device supporting the test engineer to analyze the signal of interest. In Section 2 we present the used hardware and explain implementation details. We evaluated the proof of concept and show how the different trigger features work. Furthermore, we show how the sniffer tool including the smart trigger functionalities connected with an external oscilloscope can be used to capture the communication snippet of interest for further interoperability analysis. The parallel implementation of the different features additionally increases the usability of the whole system also for other measurement purposes. Using the proposed smart NFC sniffer for high-precision measurements significantly reduces the time required to set up an oscilloscope and to capture the correct communication snippet. Our goal is to improve interoperability testing for NFC-enabled devices to on the one hand increase the user acceptance and user experience. On the other hand, interoperability testing should be included in the product development life cycle and into the certification process.

This system can be improved by implementing a real-time protocol decoder on the FPGA to support triggering on a specific command and not on a command number. Furthermore, we would like to support more different data rates and applications. Within the ANITAS project [29] we will integrate this system to the automated NFC interoperability test system at CISC Semiconductor.. This will significantly reduce the time required during manual debug sessions.

References

1. Boada, L.: Near Field Communication devices: having interoperability issues (Feb 2016), https://blog.applus.com/nfc-devices-having-interoperability-issues/
2. CISC Semiconductor: CISC Semiconductor Interoperability Website (Nov 2018), https://www.cisc.at/services/nfc-interoperability-tests/, Accessed: 2020-05-06
3. Coskun, V., Ozdenizci, B., Ok, K.: The Survey on Near Field Communication. Sensors **15**(6), 13348–13405 (Jun 2015). https://doi.org/10.3390/s150613348
4. Couraud, B., Vauche, R., Deleruyelle, T., Kussener, E.: A very high bit rate test platform for ISO 14443 and interoperability tests. In: 2015 IEEE 16th International Conference on Communication Technology (ICCT). pp. 353–356 (Oct 2015). https://doi.org/10.1109/ICCT.2015.7399857
5. Erb, M., Steger, C., Preishuber-Pfluegl, J., Troyer, M.: A Novel Automated NFC Interoperability Test and Debug System. In: Smart SysTech 2019; European Conference on Smart Objects, Systems and Technologies. pp. 1–8 (Jun 2019)
6. Finkenzeller, K.: RFID Handbook: Fundamentals and Applications in Contactless Smart Cards, Radio Frequency Identification and Near-Field Communication. John Wiley & Sons (2010)
7. Fischer, J.: NFC in cell phones: The new paradigm for an interactive world [Near-Field Communications]. IEEE Communications Magazine **47**(6), 22–28 (Jun 2009). https://doi.org/10.1109/mcom.2009.5116794
8. Gebhart, M., Wienand, M., Bruckbauer, J., Birnstingl, S.: Automatic analysis of 13.56 mhz reader command modulation pulses. In: Eurasip RFID Workshop (2008)
9. Global Industry Analysts,Inc: Consumer Preference for Contacless Payment & Automated Ticketing Supported by Speed & Convenience Benefits Drives Demand for NFC Enabled Phones (oct 2018), https://www.strategyr.com/blog/blog-post.asp?bcode=MCP-7852
10. GSMA, NFC Forum, S.J.: Joint Position Paper on Interoperability of NFC Mobile Devices (Mar 2013)
11. Hawrylak, P.J., Ogirala, A., Cain, J.T., Mickle, M.H.: Automated Test System for ISO 18000-7 - Active RFID. In: 2008 IEEE International Conference on RFID. IEEE (Apr 2008). https://doi.org/10.1109/rfid.2008.4519355
12. Heinrich, C.: RFID and beyond - growing your business through real world awareness. Wiley (2005)
13. ISO/IEC: ISO 10373-6 Identification cards: Test methods (2016)
14. Kang, S.: Relating interoperability testing with conformance testing. In: IEEE GLOBECOM 1998 (Cat. NO. 98CH36250). vol. 6, pp. 3768–3773 vol.6 (Nov 1998). https://doi.org/10.1109/GLOCOM.1998.776013
15. Kindrick, J.D., Sauter, J.A., Matthews, R.S.: Improving conformance and interoperability testing. StandardView **4**(1), 61–68 (1996)
16. Kun, G., Yigang, H., Zhouguo, H., Bin, L., Kai, S., Yanqing, Z.: Design and Development of a Open Frame RFID System Unite Test Platform. In: Proceedings. The 2009 International Symposium on Information Processing (ISIP 2009). p. 205. Academy Publisher (2009)
17. Lacmanović, I., Radulović, B., Lacmanović, D.: Contactless payment systems based on RFID technology. In: The 33rd International Convention MIPRO. pp. 1114–1119 (May 2010)

18. Langer, J., Roland, M.: Anwendungen und Technik von Near Field Communication (NFC). Springer-Verlag (2010)
19. Langer, J., Saminger, C., Grunberger, S.: A comprehensive concept and system for measurement and testing Near Field Communication devices. In: IEEE EUROCON 2009. IEEE (May 2009). https://doi.org/10.1109/eurcon.2009.5167930
20. Lime Microsystems: Lime Microsystems Website (2019), `https://myriadrf.org/`, Accessed: 2020-05-06
21. Marinčić, A., Kerner, A., Šimunić, D.: Interoperability of iot wireless technologies in ambient assisted living environments. In: 2016 Wireless Telecommunications Symposium (WTS). pp. 1–6 (April 2016). https://doi.org/10.1109/WTS.2016.7482046
22. Muehlmann, U., Gebhart, M.: Automated analysis of iso/iec14443a interrogator command pulse shapes. In: SoftCOM 2009 - 17th International Conference on Software, Telecommunications Computer Networks. pp. 75–79 (Sep 2009)
23. NFC Forum: The Keys to Truly Interoperable Communications (Oct 2007), `http://nfc-forum.org/wp-content/uploads/2013/12/NFC-Forum-Marketing-White-Paper.pdf`
24. NFC Forum: ISO/IEC 14443 Analog Parameter Comparison and Alignment (Jan 2017)
25. Shobha, N.S.S., Aruna, K.S.P., Bhagyashree, M.D.P., Sarita, K.S.J.: NFC and NFC payments: A review. In: 2016 International Conference on ICT in Business Industry Government (ICTBIG). pp. 1–7 (Nov 2016). https://doi.org/10.1109/ICTBIG.2016.7892683
26. Song, E.Y., Lee, K.B.: An interoperability test system for ieee 1451.5–802.11 standard. In: 2010 IEEE Sensors Applications Symposium (SAS). pp. 183–188 (Feb 2010). https://doi.org/10.1109/SAS.2010.5439406
27. Stark, M., Gebhart, M.: How to guarantee phase-synchronicity in active load modulation for NFC and proximity. In: 2013 5th International Workshop on Near Field Communication (NFC). IEEE (Feb 2013). https://doi.org/10.1109/nfc.2013.6482449
28. StemLabs: RedPitaya Website (Nov 2018), `https://www.redpitaya.com`, Accessed: 2020-03-16
29. Österreichische Forschungsförderungsgesellschaft: Project homepage (Jun 2019), `https://projekte.ffg.at/projekt/2893228`, Accessed: 2020-05-05