Design und Evaluierung eines mobilen ADS-B Empfängers

Markus Gräb

Bachelorarbeit

Design und Evaluierung eines mobilen ADS-B Empfängers

vorgelegt von

Markus Gräb

26. Mai 2014

Technische Universität Kaiserslautern
Fachbereich Informatik
AG Distributed Computer Systems

Betreuer: M.Sc. Schäfer, Matthias
Prüfer: Prof. Dr. Schmitt, Jens

Eidesstattliche Erklärung

Hiermit versichere ich, die vorliegende Bachelor-Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet zu haben. Alle wörtlich oder sinngemäß übernommenen Zitate sind als solche kenntlich gemacht worden. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Kaiserslautern, den 26. Mai 2014

Markus Gräb

Ich danke Matthias Schäfer für die gute Betreuung der Bachelor Arbeit.

Zusammenfassung

ADS-B (Automatic dependent surveillance-broadcast) Kommunikation ist ein wichtiger Schritt bei der Modernisierung des Flugverkehrsmanagements. Dabei sind viele Eigenschaften des Protokolls noch nicht ausreichend untersucht worden. Diese Arbeit befasst sich mit dem Vergleich von mehreren Amateur ADS-B Empfängern. Die Empfänger werden über verschiedene Faktoren verglichen und deren Probleme identifiziert. Dafür wurde ein SDR (Software Defined Radio) Sender zur Generierung von ADS-B Nachrichten genutzt. Um die Sensitivität der Empfänger zu messen, wurden die Signalstärke und Signalrate der Signalquelle bei den durchgeführten Experimenten variiert. Des Weiteren ist ein portabler ADS-B Empfänger entwickelt worden, welcher auf mobilen Endgeräten mit Android läuft und eine transportable Plattform bietet, um ADS-B Verkehr mitzuschneiden.

Abstract

The ADS-B (Automatic dependent surveillance-broadcast) protocol is a crucial component of the next generation air traffic management system. Many characteristics of the protocol are not investigated sufficiently. This thesis compares multiple low cost ADS-B Receiver. Thereby multiple factors were used for comparison and to identify the specific problems of the receivers. A SDR (Software-defined radio) platform was used to generate ADS-B packets for the tests. In order to measure the sensitivity of the receivers the signal strength and message rate was varied for the experiments. In addition a portable ADS-B receiver was developed, which runs on mobile devices with Android and provides a transportable platform for sniffing ADS-B traffic.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1 Einleitung

1.1 Motivation

Zur Unterstützung der Flugzeugnavigation wird seit einiger Zeit das Verfahren des Sekundärradars eingesetzt. Dabei ist eine Bodenstation vorhanden, welche periodisch Signale zum Flugzeug sendet. Auf dieses Signal antwortet ein Transponder im Flugzeug mit einer Antwort. Dieses Verhalten unterscheidet sich vom Primärradar, bei dem nur die Reflektion vom Flugzeug zurückgesendet und ausgewertet wird. Dadurch ist es möglich, weitere Statusinformationen über das Radar anzufragen, wie zum Beispiel die aktuelle Flughöhe, Geschwindigkeit oder eine Identifikation des Flugzeuges. Diese Informationen sorgen für eine höhere Sicherheit im Flugverkehr, da sonst nur die Entfernung zwischen den beiden Positionen Radarschüssel und Flugzeug bekannt wäre.

Der zurzeit international verwendete Standard der Kommunikation und Lokalisierung mittels Sekundärradar heißt MODE-S. Es gibt verschiedene Generationen dieses Protokolls. MODE-A sendet nur eine ID zurück, wobei nur 4.096 verschiedene Identitäten zur Verfügungen stehen. Bei MODE-C erhält man eine zusätzliche Höheninformation, was auch zu einer genaueren Positionsbestimmung führt. MODE-S fügt weitere Antworttypen hinzu.

Das Prinzip des Sekundärradars behält viele Probleme des Primärradars. Es steht nur eine beschränkte Genauigkeit der Position und eine geringe Aktualisierungsrate ($\sim 12\,\mathrm{s}$, die Umdrehungsdauer der Radarschüssel) zur Verfügung. Weiter können sich Flugzeuge gegenseitig überdecken [1].

Das grundlegende Vorgehen eines Sekundärradars ist (Abbildung 1.1), dass kontinuierlich Interrogation Requests mit einer Richtantenne ausgesendet werden. Diese Requests werden von einem Transponder auf dem Flugzeug empfangen und es wird nach einer festen Wartezeit mit einem Interrogation Reply geantwortet, welche zusätzliche Informationen enthalten kann. Diese Nachricht wird von der Radarstation empfangen und die Round-Trip-Time gemessen. Aus der Verzögerung und der Stellung der Radarschüssel kann eine ungefähre Position des Flugzeuges geschätzt werden.

Heutzutage wird MODE-S weitreichend eingesetzt, um die Flugverkehrskontrolle mittels Sekundärradars zu unterstützen. Dafür haben die meisten Länder ein großflächiges Netzwerk aus mehreren Radarstationen aufgebaut. Beispielsweise hat Deutschland ein Netzwerk aus 32 Radarstationen zur Überwachung des Flugraums und zusätzliche in größeren

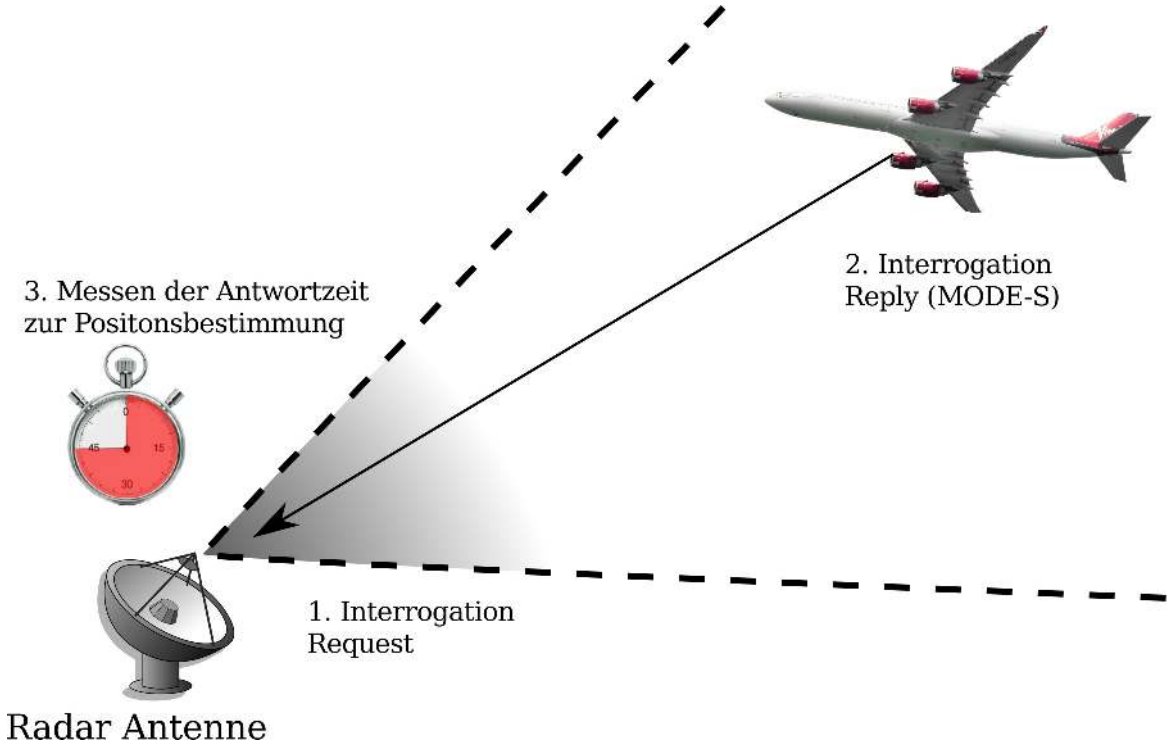


Abbildung 1.1: Darstellung eines Sekundärradars. Zuerst wird ein Interrogation Request von dem Radar ausgesendet. Diese Nachricht wird vom Transponder empfangen und mit einem Interrogation Reply beantwortet. Danach wird die ungefähre Position des Flugzeuges durch die Verzögerung bestimmt.

Flughäfen [2]. Diese überwachen den Luftraum und deren Daten werden zentral zusammengeführt. Dieses Verfahren ist durch ICAO[1] und EUROCONTROL standardisiert.

1090ES ist eine Erweiterung der bestehenden MODE-S Spezifikation, es sind mehrere neue Pakettypen vorhanden, die unter anderem **Automatic Dependent Surveillance - Broadcast** (ADS-B) Daten beinhalten können. Diese Daten werden nicht mehr auf Nachfrage, sondern in einem festen Intervall als Broadcast versendet. Die Aktualisierungsrate ist bei Broadcasts viel höher als bei der Interrogation mittels Sekundärradar.

Automatisch gesendete Broadcast Pakete stellen einen Paradigmenwechsel dar. Vorher in MODE-S/A/C wurden nur auf Nachfrage (Interrogation) Informationen des Flugzeuges versendet, jetzt versenden die Sender in einem festen Intervall die Statusinformationen. Die Daten sollen außer der Flugverkehrskontrolle auch den anderen Flugzeugen zur Verfügung stehen. Außerdem ist auch vorgesehen, Bodenfahrzeuge mit ADS-B OUT auszustatten, um ein besseres Situationsbewusstsein zu erreichen.

Es wird unterschieden zwischen Teilnehmern, die nur Daten senden können, diese sind mit einem ADS-B OUT ausgestattet und Teilnehmern des Flugverkehrs, welche Daten empfangen und auswerten können, sie verfügen zusätzlich über einen ADS-B IN.

[1] International Civil Aviation Organization

Der heutige Stand ist, dass die meisten großen kommerziellen Luftverkehrsflotten mit ADS-B OUT versehen sind. Durch Observationen wurde festgestellt, dass bereits 2/3 der Flugzeuge ADS-B Nachrichten versenden [3]. ADS-B IN ist bei kommerziellen Flugzeugen nicht so weit verbreitet [3], da die Kosten zur Ausstattung noch höher sind. Zum Einbau von ADS-B OUT muss nur der bereits installierte MODE-S Transponder erweitert werden.

Heutzutage wird ADS-B bereits an solchen Orten alleinig zur Flugverkehrskontrolle genutzt, an denen keine Überwachung per Radar verfügbar ist, wie beispielsweise Alaska [4], da der Aufbau einer ADS-B Flugverkehrskontrolle preiswerter ist als der einer Radarüberwachung.

In Zukunft soll das Sekundärradar vollständig durch ADS-B zur Flugsicherung ersetzt werden. Im EU Luftraum müssen ab 2015 alle neuen Flugzeuge mit einem ADS-B Sender ausgestattet sein und ab 2017 müssen alle Flugzeuge einen Sender besitzen. Man verspricht sich davon, eine effizientere Nutzung des Flugraums, da eine höhere Dichte an Flugzeugen möglich wird.

Es werden selbständig die Position, Geschwindigkeit und andere Statusinformationen der Flugzeuge versendet. Dadurch erhofft man sich eine höhere Genauigkeit der Lokalisierung und Planung von Flugzeugrouten. Zusätzlich können Flugzeuge, welche mit ADS-B IN ausgestattet sind, selbständig die gesendeten Daten der anderen Teilnehmer auswerten und ohne Unterstützung der Flugsicherung sicherer navigieren.

Die Position des Flugzeuges wird bestimmt mithilfe von GPS, EGNOS oder anderen satellitengestützten Verfahren zur Positionsbestimmung. Teilweise wird die Position relativ zum Start durch Inertialsensoren geschätzt, dabei können im Laufe der Zeit große Abweichungen auftreten.

Diese Positionsdaten werden von den Flugzeugen eigenständig versendet. Das ist ein bedeutender Unterschied zu den bisherigen Verfahren, da beim (Sekundär-)Radar die Lokalisierung durch die Radarstationen durchgeführt wurde, jetzt muss der ausgesendeten Position vertraut werden. Dafür ist die Positionbestimmung in den meisten Fällen genauer.

Es gibt bereits Systeme, welche ADS-B Daten auf großer Basis überwachen und zusammentragen. Eines dieser Service ist Flightradar24, in Abbildung 1.2 ist deren Kartenansicht abgebildet. Dafür wurde ein weltweites Netz aus ADS-B Empfängern aufgebaut, die es möglich machen, Flugbewegungen zu beobachten und zu überwachen.

Flightradar24 ist aber nicht für die wissenschaftliche Forschung geeignet, die das Verhalten und andere Aspekte des ADS-B Protokolls untersuchen will. Denn die Daten sind nur als ausgewerteter Datenstrom verfügbar und nicht die eigentliche ADS-B Kommunikation. Es gehen viele Informationen verloren, beispielsweise von welchen Stationen die Pakete empfangen werden, wie oft dasselbe Paket wiederholt ausgesendet wird, genaue Zeitstempel, Signalstärke der empfangenen Pakete und mehr.

Deswegen wurde das OpenSky Projekt gegründet, in Abbildung 1.3 sind die zurzeit im Einsatz befindlichen Empfangsstationen zu sehen. Diese Knoten sind über mehrere Länder

[2] `http://flightradar24.com`

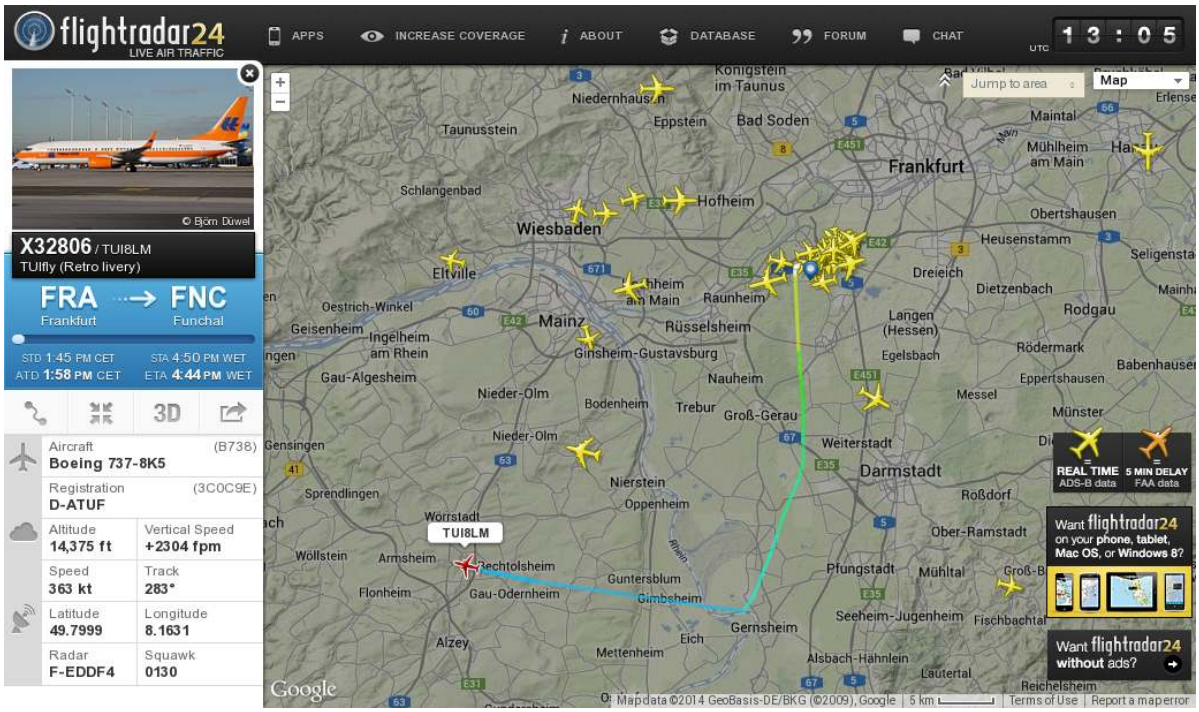


Abbildung 1.2: Abbildung von Flightradar24[2]. Zu sehen ist die aufgezeichnete Flugbahn eines Fluges mit weiteren Zusatzinformation.

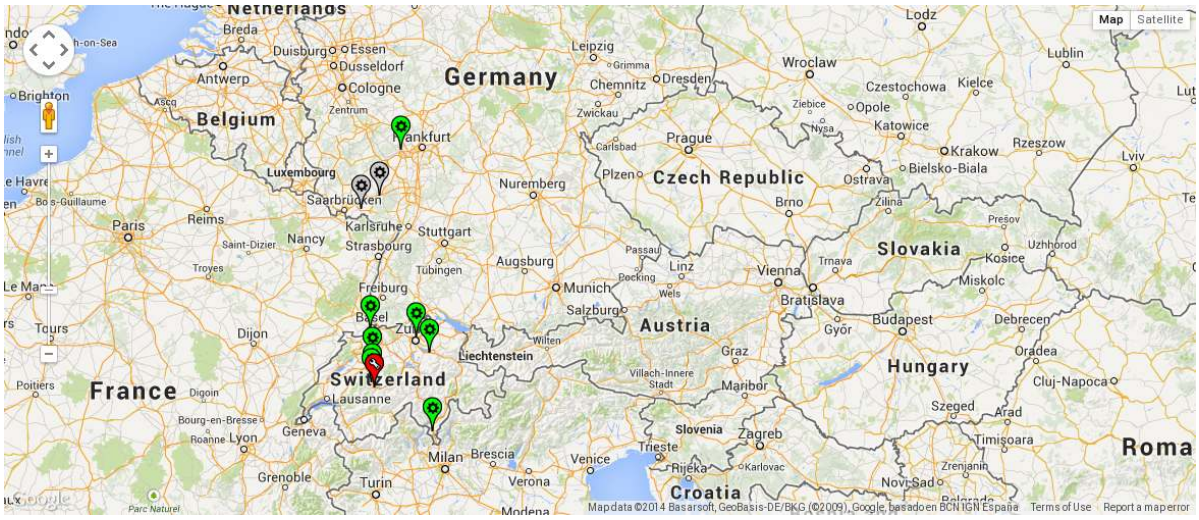


Abbildung 1.3: Der aktuelle Stand des Opensky Netzwerkes (6. April 2014). Zu sehen sind die zuzeit eingesetzten Empfänger.

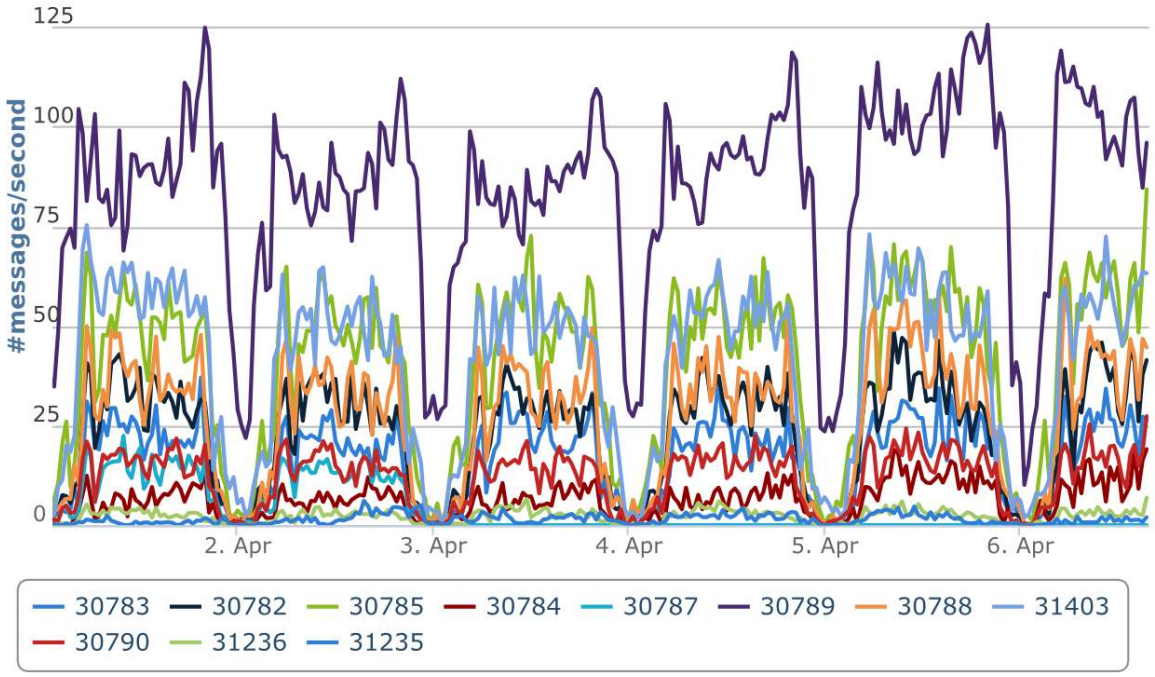


Abbildung 1.4: Darstellung des empfangenen ADS-B Verkehrs des Opensky Netzwerkes. Jede Datenreihe stellt einen Empfangsknoten dar.

verteilt und bieten dadurch eine großflächige Überwachung der ADS-B Kommunikation. Die aufgenommenen Daten werden, zur Auswertung, zentral in einer Datenbank gespeichert. Das System ist genauer in den Veröffentlichungen [3] und [5] beschrieben.

Es wird versucht, auf großer Basis ADS-B Pakete mitzuschneiden und Wissenschaftlern diesen Datensatz zur Verfügung zu stellen. Dabei wird einmal die Beobachtung der Daten in Echtzeit als auch Betrachtung über eine langen Zeitraum unterstützt. Beispielsweise kann darauf Multilateration der Flugzeuge durchgeführt werden

In Abbildung 1.4 wird der zurzeit empfangene ADS-B Verkehr pro Knoten dargestellt. Die Datensätze stammen jeweils von den Empfängern, welche in Zentraleuropa stationiert sind.

Das Projekt selbst wurde besonders auf niedrige Kosten ausgelegt. Die einzelnen Knoten werden teilweise von Privatpersonen betrieben. Dadurch soll mit einem kleinen Budget ein großes Netzwerk mit vielen Teilnehmern aufgebaut werden. Zurzeit werden nur Empfänger der Marke Kinetic SBS-3 eingesetzt, die mit einem Preis von über 565€ vergleichsweise teure Empfänger sind.

Aktuelle Projekte nutzen OpenSky für die:

- Erkennung von Implementierungsfehlern und fehlerhaften Sendern.
- Untersuchen der Performance von ADS-B.
- Validierung von gesendeten Daten.
- Durchführung von Multilateration.

Abbildung 1.5: Einsatz des ADS-B Rekorders in den Schweizer Alpen.

Es gibt verschiedene andere Amateur ADS-B Empfänger und Software Plattformen zum Empfang von ADS-B Nachrichten. Diese werden in den Kapiteln Versuchsaufbau und Auswertung durch umfangreiche Messungen einzeln untersucht und untereinander verglichen.

Die Motivation zur Entwicklung eines ADS-B Rekorders für Android ist die Standorterschließung. Mithilfe eines mobilen Empfängers kann die Software in Situationen eingesetzt werden, bei denen es sonst nur unter großen Umständen möglich wäre, Messungen anzufertigen. In Abbildung 1.5 kann man einen solchen Einsatz in den Schweizer Alpen sehen.

Eine weiterer Anwendungsbereich ist der mobile Einsatz, wenn der Empfänger nicht stationär montiert ist. Ein Beispiel wäre die Aufzeichnung eines vollständigen Fluges indem man den ADS-B Empfänger innerhalb des Flugzeuges platziert. Mit mehreren stationären Empfängern lässt sich nicht der vollständige Flug aufzeichnen. Besonders in Verbindung mit dem GPS Empfänger des Smartphones lassen sich die gesendeten GPS Koordinaten verifizieren. Zusätzlich ist die Kommunikation während dem Start- und Landevorgang interessant, welche noch nicht untersucht wurde.

1.2 Das ADS-B Protokoll

1.2.1 Medium

Es gibt zwei Kommunikationsmedien, welche für die ADS-B Kommunikation spezifiziert sind.

Eines davon ist UAT (Universal Access Receiver), wobei dieses Medium nur in den USA für den kommerziellen Flugverkehr eingesetzt wird. Es arbeitet auf 978 MHz. UAT wurde

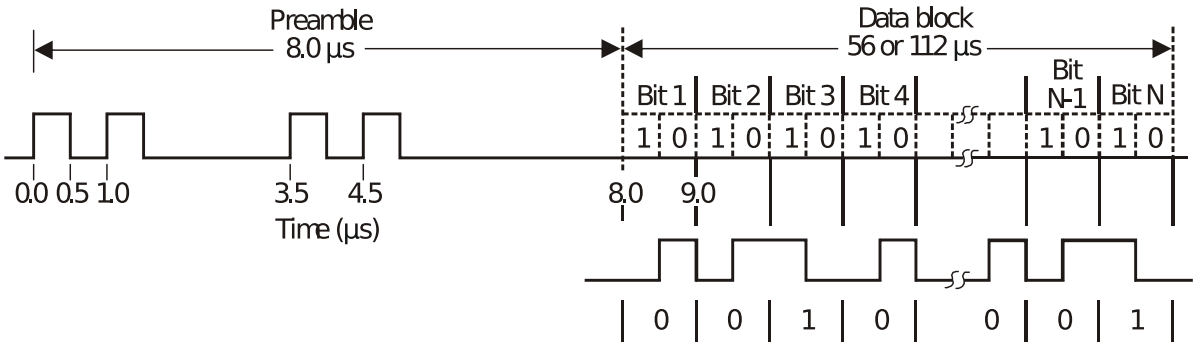


Abbildung 1.6: Kodierung eines MODE-S Paketes, wobei der Datenblock für ADS-B immer 112 Bit lang ist [6].

speziell für den ADS-B Einsatz entwickelt und bietet zusätzlich Unterstützung für FIS-B (Flight Information Service - Broadcast) und TIS-B (Traffic Information Service - Broadcast). Diese Services bieten weitere Datenquellen zur Sicherung des Flugraums. UAT wird in Europa für den nicht kommerziellen Flugverkehr eingesetzt.

In Europa ist nur das 1090ES (1090 MHz Extended Squitter) Protokoll als Medium für den kommerziellen Flugverkehr zertifiziert. Dieses Kommunikationsverfahren wird bereits für die MODE-S und MODE-C Kommunikation eingesetzt. Der verwendete Broadcast Kanal liegt auf 1090 MHz. Der Kommunikations-Standard ist in [6] spezifiziert. Auf 1030 MHz ist ein Rückkanal vorhanden, um Interrogation Requests (Anfragen) an Flugzeuge zu senden.

Die Entscheidung 1090ES zu nutzen, wurde aus Kostengründen und zur Unterstützung der schnellen Ausbreitung von ADS-B getroffen. Die Ausrüstung von Flugzeugen mit ADS-B Transpondern ist viel preiswerter, da existierende MODE-S Transponder nur modifiziert werden müssen, um ADS-B auszusenden. Aber es wird kein FIS-B unterstützt, welches den Piloten mit weiteren unterstützenden Informationen versorgt, wie beispielsweise Wettervorhersagen.

MODE-S wurde in den 80er Jahren entwickelt, dadurch gibt es keinerlei Sicherheitsmaßnahmen, wie beispielsweise Authentifizierung der Absender. Durch den Einsatz als Protokoll für Sekundärradar gibt es auch keinerlei Maßnahmen zur Vermeidung von Nachrichtenkollisionen.

Es wird nur die Kommunikation über den 1090ES Kanal betrachtet, da es zurzeit weitreichender eingesetzt wird. Viele Probleme lassen sich auch auf das UAT Medium übertragen.

Die MODE-S Kommunikation benutzt eine einfache Puls-Positions-Modulation (PPM), im speziellen ist es die Manchester Kodierung. Bei 0 wird eine steigende Signalflanke gesendet, bei 1 eine sinkende Flanke. In Abbildung 1.6 ist die Kodierung einer einzelnen Nachricht zu sehen. Es wird eine Präambel ausgesendet, die 4 Pulse aussendet und danach folgt die eigentliche Nachricht. Damit hat eine Nachricht mit 112 Bit Daten eine Länge von 120 μs.

ADS-B OUT Sender übertragen periodisch die Pakete im Broadcast. Dabei wird ein Random Backoff eingesetzt, der genaue Wert hängt vom Inhalt der Nachricht ab. Beispielsweise wird

Format (5 Bit)	(3 Bit)	ICAO24 (24 Bit)	ME (56 Bit)	CRC (24 Bit)
17	CA (Capability)		ADS-B Nachricht	
18	CF (Control Field)		ADS-B Nachricht	

Tabelle 1.1: Struktur eines MODE-S Datepaketes mit Länge 112 Bit und Typ 17 und 18.

für die ADS-B Positions Nachricht vorgeschrieben, dass die Nachrichten in einem Intervall von 0,4 bis 0,6 s ausgesendet werden sollen[7]. Dieser Random Backoff ist gleichverteilt. Das Fehlen von einem Clear Channel Assessment führt zu einer sehr hohen Rate an Paketverlusten und niedriger Effizienz des Protokolls.

Die Nachrichten werden mit einer Leistung von bis zu 500 W versendet, was dafür sorgt, dass die Reichweite der Nachrichten sehr hoch ist. Die Reichweite der Signale ist bei einer solchen Sendeleistung nur noch durch die Line of Sight (Sichtverbindung) und die Erdkrümmung beschränkt.

1.2.2 Paketspezifikation

Für die ADS-B Kommunikation gibt es in 1090ES einen speziellen Formatcode, der diese Pakete markiert. Jedes Paket ist 112 Bit lang und in Tabelle 1.1 kann man die Struktur sehen.

Die ICAO24 Adresse ist eine eindeutige Identifikationsnummer, die jedem Transponder zugewiesen wird, diese ID hat eine Länge von 3 Byte. Die Adressen sind in Blöcken den einzelnen Ländern zugewiesen. Durch den Wertebereich kann das Ursprungsland des Flugzeuges herausgefunden werden.

Das Format Feld für die ADS-B Kommunikation hat den Wert 17 oder 18. Das Format 17 steht führ *Extended Squitter* und 18 für *Extended Squitter/non transponder*.

Extended Squitter (Format 17) wird von Flugzeugen zum Broadcast ihrer Statusinformationen eingesetzt. Format 18 wird für TIS-B und Transponder eingesetzt, die kein MODE-S besitzen.

Der Mehraufwand pro Paket ist sehr hoch, die Hälfte der 1090ES Nachricht sind ADS-B Nutzdaten. Der maximale Durchsatz ist $\frac{112\,\text{bit}}{120\,\mu s} = 0,9\,\text{Mb/s}$, wobei die Werte in der Praxis viel niedriger sind, da bei mehreren Sendern sehr viele Nachrichtenkollisionen auftreten.

Zur Berechnung der Checksumme wird die zyklische Redundanzprüfung (Cyclic Redundancy Check) eingesetzt, wobei die Checksumme von der vollständigen Nachricht berechnet wird. Das Polynom zur Berechnung der Checksumme lautet:

$$G(x) = 1 + x^3 + x^{10} + x^{12} + x^{13} + x^{14} + x^{15} + x^{16} + x^{17} + x^{18} + x^{19} + x^{20} + x^{21} + x^{22} + x^{23} + x^{24}$$

2 Versuchsaufbau

In diesem Kapitel wird der Versuchaufbau zum Untersuchen der Performance von mehreren ADS-B Empfängern erklärt. Die Performance soll über verschiedene Faktoren beurteilt und verglichen werden.

2.1 Vorstellung der Empfänger

Es wurden mehrere frei verfügbare Amateur Empfänger ausgewählt. Dazu sind 2 Empfänger gewählt, welche auf den Mode-S und ADS-B Empfang spezialisiert sind: Kinetic SBS-3 und GNS 5890. Außerdem werden 2 SDR Empfänger getestet: DVB-T Stick mit RTL2832-U Chip und eine Ettus USRP2 und als dazugehörige Empfangssoftware: dump1090 und gr-air-modes.

Tabelle 2.1 beschreibt die grundlegenden Eigenschaften der verschiedenen Empfänger.

Ein Software Defined Radio (SDR) ist eine allgemeine Empfangshardware. In herkömmlichen Empfängern ist die vollständige Signalverarbeitung von der Einspeisung des Antennensignals bis zu dem dekodierten Datenstrom in Hardware implementiert. Heutzutage ist es mit steigender Rechenleistung möglich, einen großen Teil dieser Schritte in Software durchzuführen.

Beispielsweise soll ein ADS-B Signal auf 1090 MHz empfangen werden. Nach dem Nyquist-Shannon-Theorem muss ein Signal mit $2 \cdot f_{\text{grenz}} \leq f_{\text{abtast}}$ abgetastet werden, um es verlustfrei rekonstruieren zu können. Die Variable f_{grenz} ist die maximale Frequenz des Eingangssignals, in diesem Fall muss $f_{\text{grenz}} > 1090\,\text{MHz}$ sein. Daraus folgt, dass die benötigte Abtastrate des AD-Wandlers über 2 GHz sein muss.

Statt dem Tiefpass wird eine Bandbegrenzung durchgeführt, für diese Unterabtastung gilt $2 \cdot (f_{\text{max}} - f_{\text{min}}) \leq f_{\text{abtast}}$. Die Variable f_{min} ist die untere Frequenzschranke und f_{max} die obere. Damit ist es möglich, mit einer niedrigeren Abtastrate das bandbegrenzte Signal zu digitalisieren und in Software weiter zu verarbeiten. Dieses ADS-B Signal kann mit einer Abtastrate von nur 2 MSamples erfolgreich empfangen werden.

Das Softwarepaket GNURadio und verschiedene Hardwareempfänger haben in den letzten Jahren dafür gesorgt, dass diese Technik erschwinglich wurde und auch außerhalb des militärischen Umfelds eingesetzt wird.

[1] Quelle: `http://www.flightstore.co.uk/pilot-supplies-c1/kinetic-avionics-sbs-3-virtual-radar-p761`

(a) Kinetic SBS-3[1]

(b) GNS-5890 Empfänger

(c) Die Platine des GNS-5890 Empfängers, zu sehen ist der PIC-Mikrocontroller.

(d) Software Defined Radio: Ettus Research USRP2

(e) Software Defined Radio: DVB-T Empfänger mit RTL2832U Chipsatz.

Abbildung 2.1: Die verschiedenen verwendeten Empfänger

		Schnittstellen		Metadaten			
Empfänger	Preis/Lizenz	Anbindung	Antenne	Zeitstempel	weitere	Verbreitung	Dokumentation
Kinetic SBS-3	565€	Ethernet (autonom)[2]	SMA	✓40 MHz		+++	+++
GNS-5890	150€	USB	MCX	✓12 MHz		+	
USRP2	> 1000€	GBit-Ethernet	SMA	—		+	+
RTL2832U	7€	USB	MCX	—		o	+
gr-air-modes	GPLv3	SBS1-Server, stdout	—	✗	RSSI		o
dump1090	BSD	SBS1-Server, stdout	—	✗	Infos über CRC-Korrektur		+

Tabelle 2.1: Vergleichstabelle der Eigenschaften.

[2]Kann die Daten ohne direkt angeschlossenen PC auswerten. Bei Feldern mit — kann keine Aussage über diese Eigenschaft getroffen werden.

2.1.1 Kinetic SBS-3

Kinetic SBS-3 ist ein leistungsfähiger ADS-B und MODE-S Empfänger (Abbildung 2.1(a)). Er kostet etwa 565€. Die Anbindung erfolgt über 100-Mbit/s-Ethernet und bietet dadurch den Vorteil, dass der Empfänger entfernt von einem Rechner montiert werden kann. Es wird optional ein Client Modus unterstützt, sodass der Empfänger selbstständig alle Nachrichten an einen entfernten Server schickt.

Der Empfänger bietet die Option, dass jedes Paket mit einem 24 Bit Zeitstempel annotiert wird, dieser wird mit 20 MHz inkrementiert.

Als zusätzliche Funktionalität sind 2 vollständige SDR Empfänger integriert, die den Funkbereich von 26 MHz bis 980 MHz unterstützen und jeweils eine Bandbreite von 8 MHz besitzen. Diese können beispielsweise eingesetzt werden, um den Sprachflugfunk zu überwachen. Das Signal kann Onboard auf einem FPGA dekodiert werden und über einen Audioausgang ausgegeben werden. Es ist auch ein Mixer vorhanden, sodass die beiden Tuner gleichzeitig abgehört werden können.

Als weitere Funktionalität kann der Empfänger AIS (Automatic Identification System) und ACARS dekodieren. AIS ist ein System, welches zum Austausch von Positions- und anderen Daten von Schiffen genutzt wird (Frequenzen 161,975 und 162,025 MHz). ACARS ist ein Protokoll zum Austausch von Textnachrichten zwischen Flugzeugen (Frequenzen um 130 MHz).

Als weitere Schnittstelle ist ein USB-Anschluss vorhanden, welcher hauptsächlich zur Stromversorgung und zum Aktualisieren der Firmware dient. Es gibt zugängliche I^2C und serielle Schnittstellen. Darüber ist es möglich, GPS-Empfänger anzuschließen. Es war vorgesehen, einen dezidierten Touchscreen-Controller für die SDR-Tuner anzubieten, dieser ist aber zum aktuellen Zeitpunkt nicht erhältlich. Für diese beiden Schnittstellen gibt es keine öffentliche Dokumentation.

Der Empfänger bietet 2 getrennte Antenneneingänge (SMA Buchse), einen für den ADS-B Empfang und einen für die SDR-Empfänger. Somit können jeweils passende Antennen genutzt werden.

Zur Aufzeichnung wurde folgender Befehl eingesetzt:

```
echo "\x10\x02\x17\xa2\x74\xa0\xce\x96\x65\xf8\x4f\xb1\x10\↵
   x03\x7f\xde" | nc 192.168.1.170 10001 | ./reader.py
```

Die TCP-Verbindung zum SBS-3 Empfänger wird mithilfe von `netcat` aufgebaut. Am Anfang wird eine Login Nachricht zum Empfänger geschickt. Das Skript `./reader.py` liest den Datenstrom aus der Standardeingabe ein und decodiert ihn anhand der verfügbaren API Spezifikation[3]. Die decodierten Nachrichten werden mit einen zusätzlichen UNIX-Zeitstempel auf der Standardausgabe ausgegeben.

[3] `http://kinetic-avionics.com/api.php`

#	Inhalt	Bedeutung
1	10	DLE
2	02	STX
3	17	TYPE LOGIN_REQUEST
4	a2	
5	74	
6	a0	
7	ce	
8	96	MESSAGE_TAG
9	65	
10	f8	
11	4f	
12	b1	
13	10	DLE
14	03	ETX
15	7f	CRC high byte
16	de	CRC low byte

Tabelle 2.2: Die gesendete Login Nachricht.

Das Datenpaket der Login-Nachricht ist in Tabelle 2.2 entschlüsselt. Der MESSAGE_TAG ist eine von der Software des Herstellers gesendete Nachricht und sorgt dafür, dass die Pakete ohne Verzögerung ankommen. Ohne diesen Login werden alle Nachrichten um 5 Minuten verzögert. Dieser wurde durch Mitschneiden der Kommunikation bestimmt.

Diese Verzögerung wurde aus Sicherheitsgründen implementiert. Ähnliche Einschränkungen hat die US-Amerikanische Flugbehörde (FAA) eingeführt, diese stellen zwar Zugriff auf ihre Flugdaten zur Verfügung, aber nur mit einer Verzögerung von 5 Minuten. Damit wird versucht sicherheitskritische Informationen, wie aktuelle Position oder andere Statusmeldungen nur verzögert bekannt zu geben, sodass Beobachtern nur ein beschränktes Bild des aktuellen Flugverkehrs geboten wird.

2.1.2 Global Navigation Systems GNS-5890

Der GNS-5890 ist ein MODE-S Empfänger in USB-Stick Format und wird über USB angeschlossen (Abbildung 2.1(b)). Mit einem Preis von 150€ ist er preiswerter als der Kinetic Empfänger.

Der USB-Stick meldet sich als Gerät der Klasse *Communikations Device Class*, welches verschiedene Geräteklassen für Kommunikationsgeräte wie Netzwerkkarten, ISDN-Karten und Modems definiert. Genauer gesagt meldet es sich als Device des *Abstract Control Models*

(CDC-ACM). Die Treiber für solche Geräte werden in den meisten Betriebssystemen mitgeliefert und der Standard ist frei verfügbar [8]. Das Protokoll öffnet eine einfache serielle Schnittstelle zur Kommunikation mit dem Empfänger.

Das serielle Kommunikationsprotokoll des Empfängers basiert auf dem adsbPIC Projekt [9]. Der GNS-5890 unterstützt nur eine Teilmenge der Optionen. Die verfügbaren Optionen sind im Kapitel 4 aufgezählt.

Es ist möglich, alle Pakete mit einem 12 bit Zeitstempel zu annotieren, wobei die Uhr mit 12 MHz inkrementiert wird.

Als Besonderheit gibt es eine Anwendung für Android, welche den Empfänger nutzt. Diese bietet eine Anzeige der zurzeit empfangenen Flugzeuge in einer Radaransicht.

Der Empfänger wurde in Verbindung mit der entwickelten Android Anwendung getestet. Als Plattform für die Tests wurde ein Android-Gerät mit Intel Atom N270 CPU (1,6 GHz) benutzt, die Android Version stammt vom Android-X86 Projekt und hat die Version 4.0-r1. Als Einstellung für den Test wurde gewählt, dass nur ADS-B Nachrichten empfangen und die Zeitstempel aktiviert werden.

Zur Verwendung unter Linux muss dem USB-Kernel Modul eine zusätzliche Option übergeben werden. Der USB-Treiber ist sonst nicht für dieses Gerät aktiviert.

```
modprobe usbserial vendor=0x04d8 product=0xf8e8
```

2.1.3 Ettus Research USRP2

Der USRP2 von Ettus Research ist ein vollwertiges Software Defined Radio (kurz SDR). Die Einheit kann als Empfänger und Sender eingesetzt werden, in Abbildung 2.1(d) ist die Hardware zu sehen.

Das SDR wird über eine Gigabit-Ethernet Schnittstelle angebunden und unterstützt eine Abtastrate von bis zu 25 $^{\text{MSample}}/_{\text{s}}$ bei einer Genauigkeit von 16 bit, was einer Datenrate von 800 $^{\text{Mb}}/_{\text{s}}$ entspricht.

Die USRP Geräte besitzen eine modulare Architektur. Die Hauptplatine besteht aus einem FPGA zur Signalverarbeitung, den AD- und DA-Wandlern und der Uhr. Das Radio-Frontend kann modular je nach Einsatzbereich gewählt werden, diese Platine ist das Daughterboard.

Das verwendete Daughterboard ist das SBX Rev. 3.[4] Es kann im Frequenzband von 400 MHz bis 4 GHz arbeiten und bietet eine maximale Bandbreite von 40 MHz. Durch diese Eigenschaften eignet sich es zum Senden und Empfangen von ADS-B Nachrichten. Die maximale Sendeleistung ist 100 mW = 20 dBm. Damit ist es auch möglich, starke Signalquellen zu simulieren.

[4] https://www.ettus.com/product/details/SBX

Die Eingangsverstärkung und Ausgangsverstärkung des Daughterboards heißt Gain. Der Gain Wert ist jeweils von 0–32 wählbar, wobei die Einheit dB ist. Zum Empfang wurde er auf dem Default-Wert belassen, welcher die Hälfte des maximalen Wertes beträgt (16 dB).

Der Preis der Hardware ist verhältnismäßig hoch, das eingesetzte Paket aus Ettus USRP2 und SBX Daughterboard kostet über 1000€. Aber das Einsatzgebiet ist im Vergleich zu den beiden spezialisierten ADS-B Empfängern viel größer.

Für die Versuche wurde eine Samplingrate von 10 $^{\text{MSample}}/_{\text{s}}$ zum Senden und Empfangen gewählt. Mit der gewählten Rate ist es immer noch möglich, 2 USRPs an einem Rechner zu betreiben. Die Samplingrate ist hoch genug, um ADS-B Nachrichten zu empfangen.

2.1.4 RTL2832U

Ein weiterer SDR Empfänger, der betrachtet wird, basiert auf dem RTL2832U Chipsatz. Diese Empfänger werden meist als DVB-T oder DAB Empfänger verkauft, sind aber vollständige SDR-Empfänger. Sie zeichnen sich durch niedrige Preise aus und sind für unter 7€ erhältlich. Die eingesetzten Treiber wurden von Osmocom im rtl-sdr Projekt entwickelt [10].

Je nach Hersteller des Sticks sind verschiedene Tuner verbaut. Der Tuner übernimmt die Aufgabe der analogen Signalverarbeitung. Der nutzbare Frequenzbereich ist ungefähr 24 – 1700 MHz, dieser unterscheidet sich je nach verbautem Tuner.

Die theoretisch maximal erreichbare Samplerate beträgt 3,2 $^{\text{MSample}}/_{\text{s}}$, aber in der Praxis sind Werte bis zu 2,5 $^{\text{MSample}}/_{\text{s}}$ möglich. Die verfügbare Bandbreite ist 2 – 3 MHz. Ein besonderes Feature ist eine automatische Gainkontrolle, welche die Eingangsverstärkung automatisch der Signalstärke anpasst.

Getestet wurde ein Noname DVB-T Stick, welcher diesen Chipsatz besitzt. Als Tuner ist der Rafael Micro R820T Chip verbaut, welcher einen Frequenzbereich von 24 – 1766 MHz unterstützt [10]. Als Samplerate wurde 2 $^{\text{MSample}}/_{\text{s}}$ gewählt, da es mit dieser Rate möglich war ohne Datenverluste aufzunehmen.

Der verwendete Befehl zur Aufnahme lautet:

```
rtl_sdr -f 1090000000 -s 2000000 out.bin
```

Der Gain des Tuners ist dadurch auf automatische Kontrolle gestellt.

2.1.5 Decodersoftware

Für die SDR-Empfänger werden zwei verschiedene Programme getestet.

Die Programme bekommen von dem SDR Empfänger IQ-Daten übergeben. IQ-Daten sind eine Repräsentation des empfangenen Signals in der komplexen Ebene. Das IQ-Signal

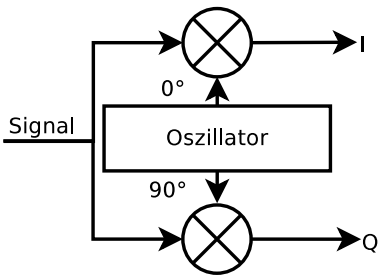


Abbildung 2.2: Generierung der IQ-Signale. Der Oszillator schwingt mit der Trägerfrequenz des Signals. Der I Anteil entsteht durch Multiplikation des Oszillatorsignals mit dem ursprünglichen Signal. Bei der Multiplikation mit Q hat der Oszillator eine Phasendrehung von 90 Grad.

entsteht durch Multiplikation des Eingangs mit der Trägerfrequenz, in Abbildung 2.2 ist das Vorgehen zu sehen. Der I Anteil wird durch die Multiplikation eines Trägersignals mit dem Eingangssignal gebildet. Der Q Anteil entsteht durch die Multiplikation des um 90 Grad phasengedrehten Trägersignals.

Der I-Wert ist der reelle Signalanteil und Q der imaginäre Anteil. Die Amplitude ist mit $\sqrt{I^2+Q^2}$ berechenbar und der Phasenwinkel mit $\tan(\phi) = {}^{I}/_{Q}$.

Durch Aufzeichnung dieser Daten können verschiedene Empfänger mit dem gleichen Datensatz verglichen werden. Dafür wurden die Rohdaten des RTL2832U Empfängers gespeichert und jeweils den Decoderprogrrammen übergeben.

Zur Verwendung der Aufnahmen mit gr-air-modes wurde das Datenformat der Aufnahmen von 8 Bit unsigned Int auf 32 Bit Float konvertiert, welches das übliche Datenformat von gnuradio ist. Der folgende Pseudocode wurde dafür eingesetzt.

```
(((float32) input) - 127) * 8e-3
```

dump1090

Die dump1090 Software[5] wurde speziell für den Einsatz mit RTL2832U Empfängern entwickelt. Das Programm ist in C geschrieben und bietet nach dem Entwickler eine bessere Empfangsleistung als andere Dekoder. Sie bietet Unterstützung, den Paketinhalt von vielen Formatnummern zu dekodieren, und überzeugt durch eine große Anzahl an verschiedenen Schnittstellen zum Weiterverarbeiten und zur Ausgabe der Daten.

Es ist ein HTTP-Webserver implementiert, welcher die Flugzeuge in einer Kartenansicht darstellen kann. Die Daten können im SBS-1 Datenformat weitergeben werden. Es gibt auch eine Option zur Weitergabe der rohen ADS-B Nachrichten. Zur Aggregation von verschiedenen Empfängern ist es möglich einen Server Modus zu nutzen, welcher die Daten von mehreren Empfängern kombinieren kann.

Ein Alleinstellungsmerkmal ist, dass die Software bei fehlerhaften Nachrichten versucht, eine CRC-Korrektur durchzuführen. Manche Pakettypen enthalten statt der Checksumme ein Feld, welches die Checksumme XOR der ICAO24 ist, dafür wird eine Liste mit den bisher gesehenen ICO24 Nummern gespeichert.

[5] `https://github.com/antirez/dump1090`, Commit `53cca39`

Außerdem gibt es eine Option zur aggressiveren Decodierung, welche erheblich mehr Rechenleistung benötigt. Dabei wird versucht, mehr Bitfehler zu korrigieren, und bis zu 2 Demodulationsfehler werden toleriert. Mit Demodulationsfehler wird gemeint, dass beim Decodieren die 2 Samples eines Symbols den gleichen Wert haben, obwohl bei dem Manchester-Code die gesendeten Symbole `[HIGH, LOW]` oder `[LOW, HIGH]` sind.

Die verwendete Option zum Decodieren der aufgenommenen IQ-Daten lautet, wobei im Aggressiven Modus noch --aggressive angehängt wird:

```
dump1090 --ifile FILENAME
```

gr-air-modes

Die Software gr-air-modes[6] basiert auf dem GNU-Radio Projekt. Sie wurde in einer Kombination von C++ und Python geschrieben, wobei die performancekritische Signalverarbeitung in C++ und die Weiterverarbeitung der Nachrichten in Python durchgeführt wird. Die Wahl des SDR-Empfängers ist nicht beschränkt.

Zur Ausgabe der Nachrichten gibt es eine GUI, welche eine Kartenansicht bietet. Wie die dump1090 Software unterstützt die gr-air-modes auch die Ausgabe der ADS-B Nachrichten mittels Server und dem SBS-1 Datenformat. Als Besonderheit kann die Software die Flugpositionen zum FlightGear-Flugsimulator weitergeben, sodass der Simulator reelle Daten des Flugverkehrs hat. Es ist auch wieder ein Client Modus vorhanden, welcher Daten von anderen gr-air-modes Instanzen entgegen nehmen kann.

Es werden die meisten spezifierten Pakettypen decodiert.

Die Software benutzt einen Threshold beim Decodieren der Nachrichten. Der Threshold filtert alle Nachrichten aus, deren Signalstärke (RSSI) kleiner als der gewählte Wert ist. Für die Experimente wurde der Defaultwert von 7 dB benutzt. Der RSSI Wert wird folgend berechnet (Ausschnitt aus `lib/slicer_impl.cc`) , wobei `rx_packet.reference_level` der RSSI ist:

```
rx_packet.reference_level = (in[i]
                           + in[i+2]
                           + in[i+7]
                           + in[i+9]) / 4.0;
```

Das Array `in` besteht aus dem bereits synchronisierten Signal, jede Zelle im Array entspricht einem Zeitabschnitt $0,5\,\mu s$ und ist das mittlere Signallevel. Die Positionen 0, 2, 7 und 9 sind die Pulse in der gesendeten Präambel. Der Wert kann mittels $\text{level}_{db} = 20 \cdot \log_{10}(\text{level})$ nach dB umgerechnet werden.

[6] `https://github.com/bistromath/gr-air-modes`, Commit `42bf16f`

Abbildung 2.3: Die eingesetzten Dämpfungsglieder.

Zum Speichern der rohen MODE-S Nachrichten wurde der Funktion `make_parser` aus der Datei `python/parse.py` eine Ausgabe der Nachricht, die aktuelle Uhrzeit und RSSI hinzugefügt, da sonst nur der decodierte Nachrichteninhalt ausgegeben werden kann.

Die verwendeten Einstellungen zum Empfang sind:

```
modes_rx  -p --rate 10e6 --args addr=192.168.10.4 -A "TX/RX"
# -p Use pulse matched filtering
# -r RATE   set sample rate
# -A ANTENNA
```

2.2 Experimente

Mit einem Programm werden ADS-B Nachrichten generiert, welche folgende Form haben:

Format (5 Bit)	CA (3 Bit)	ICAO24 (5 Bit)	ME (56 Bit)	CRC (24 Bit)
17_{10}	0	$FFFFFF_{16}$	$F0000000000000_{16}$	

Das ME Feld wird als Zähler eingesetzt, bei jedem Experiment wird es mit 0 initialisiert und dann pro Paket um 1 inkrementiert. Dadurch lässt sich klar erkennen welche Nachrichten verloren gegangen sind und ob Paketverlust in größeren Blöcken auftritt. Außerdem können andere Fehler wie eine falsche Reihenfolge der Pakete erkannt werden.

Der Typ des Extended Squitter Subpaketes ist 30, was einem reservierten Typ entspricht. Dieser hat keine spezifizierte Funktion und wird in der Praxis nicht eingesetzt [7].

Diese Nachrichten werden mittels PPM moduliert und über die Gigabit-Ethernet Schnittstelle zur USRP2 mit einer Samplerate von 10 $^{\text{MSample}}/_{\text{s}}$ gesendet.

Als variable Faktoren wurden die Nachrichtenrate und die Stärke des gesendeten Signals gewählt. Durch variieren dieser beide Werte können die wichtigsten Eigenschaften der verschiedenen Empfänger gemessen und sinnvoll verglichen werden.

Signalstärke

Die Signalstärke wurde in 2 dB Schritten von einer Dämpfung mit -20 dB bis 130 dB variiert. Sie wurden jeweils durch eine Kombination von Dämpfungsliedern in der Signalleitung

Spacing [μs]	Nachrichtenrate (gerundet) [$^1/_s$]	Anzahl Nachrichten pro Versuch
9880	100	18000
4880	200	36000
3213	300	54005
880	1000	180000
0	8333	1500000

Tabelle 2.3: Die verwendeten Abstände zwischen den Paketen und die dazugehörige Paketrate.

und variieren der Ausgangsleistung der USRP2 erreicht. Als stärkstes Signal wurde eine Kombination von einer Dämpfung um 10 dB und einem maximalen Gain von 30 dB gewählt. In Abbildung 2.3 sind die Dämpfungsglieder abgebildet.

Wie bereits erwähnt haben ADS-B Sender in Flugzeugen eine sehr große Ausgangsleistung. Durch den direkten Anschluss der Empfänger an den USRP-Sender wurde versucht, diese Signalstärke zu simulieren.

Bei dieser Sendeleistung kann es auf der Empfängerseite zu Problemen kommen. Es kann zu Verzerrungen beispielsweise durch Sättigung kommen. Dieses Phänomen wird auch als Donut Effekt bezeichnet [3]. Deswegen müssen sehr starke Signale getestet werden. Das USRP2 mit SBX 400 Daughterboard hat eine maximale Ausgangsleistung von 100 mW, was in Verbindung mit der minimalen gewählten 10 dB Dämpfung ungefähr einer Empfangsstärke von 10 mW entspricht.

Der Gainwert des USRP Senders entspricht der Verstärkung des Signals dB.[7] Diese Aussage wurde nicht überprüft und die Ausgangsleistung des USRP Daughterboards ist nicht kalibriert. Die Verifizierung liegt außerhalb des Rahmens dieser Arbeit und muss bei einer Weiterführung bestätigt werden.

Die Daten wurden in den Blöcken $[-10\,\text{dB}, 20\,\text{dB}]$, $[20\,\text{dB}, 50\,\text{dB}]$, $[50\,\text{dB}, 80\,\text{dB}]$, $[80\,\text{dB}, 110\,\text{dB}]$, $[110\,\text{dB}, 130\,\text{dB}]$ erfasst, wobei am Anfang der Gain des Senders auf 0 steht und in Schritten mit Größe 2 bis auf 30 erhöht wird.

Nachrichtenrate

Die Nachrichtenrate ist ein weiterer wichtiger Faktor, der betrachtet wird, da durch den automatischen Broadcast der Nachrichten ein hoher Nachrichtenverkehr herrscht. In Figur 2.3 sind die gewählten Nachrichtenwerte zu sehen. Besonders von Interesse ist das Verhalten bei hohen Nachrichtenraten, wie sie oft in der Praxis auftreten.

[7] http://gnuradio.org/doc/doxygen/classgr_1_1uhd_1_1usrp__sink.html#af8cc5c7ef194bd19c5db8fde88f51c11

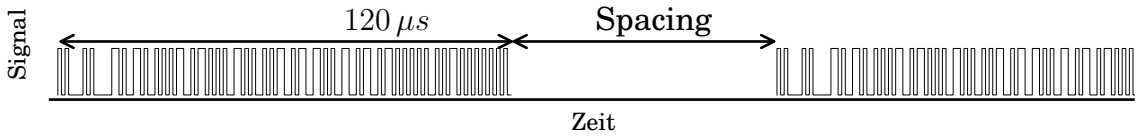


Abbildung 2.4: Darstellung des Spacings zwischen zwei 1090ES Paketen. Die Zeit zwischen dem Ende des ersten Paketes und dem Beginn der Präambel des zweiten Paketes ist das Spacing. Es ist auch die Paketlänge von 120 μs eingezeichnet.

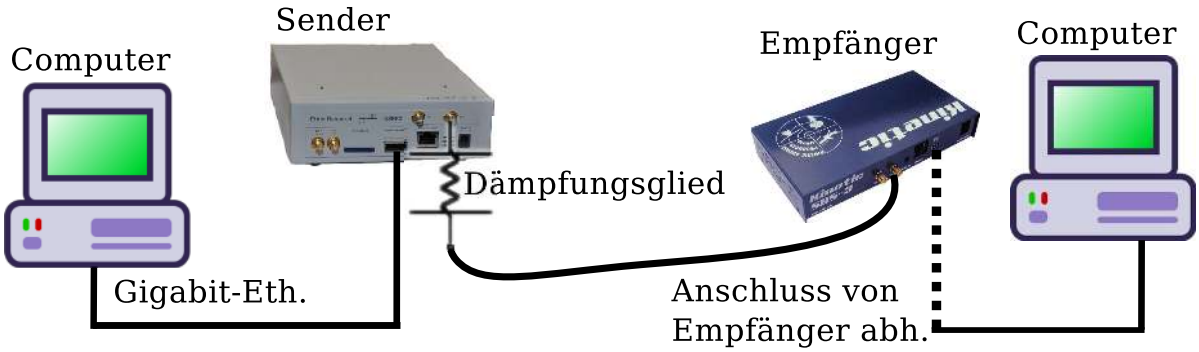


Abbildung 2.5: Darstellung des Versuchsaufbaus. Auf der linken Seite ist der PC, welcher die Signaldaten zum Sender schickt. Von dem USRP2 Sender führt die Signalleitung zum ADS-B Empfänger. Dieser ist an einen zweiten Computer angeschlossen, welcher die Messergebnisse protokolliert.

In Abbildung 2.4 ist das Spacing zwischen 2 Paketen dargestellt. Zu sehen ist die Dauer eines Paketes von 120 μs. Die Zeitdauer zwischen dem Ende des ersten Paketes und dem Beginn der Präambel des zweiten Paketes ist das Spacing der Nachrichten.

Die Performance des Empfängers bei einem Spacing von 0 zwischen den Paketen zeigt auch die Robustheit der Decodierung. Durch den geringen Abstand der einzelnen Nachrichten, können fälschlicherweise Präambeln detektiert oder übersehen werden oder auch andere Probleme der Decodierung auftreten.

Pro Aufnahme werden $^{180\,s}/_{\text{Spacing}+120\,\mu s}$ Nachrichten versendet, wobei Spacing die Wartedauer zwischen zwei Nachrichten ist. Für die Dauer jedes Experiments wurde 180 Sekunden gewählt.

Es wurden keine Versuche mit realen ADS-B Daten durchgeführt. Für eine umfassende Untersuchung der Performance eines Empfängers ist es nötig, dass die Robustheit in einer realen Umgebung untersucht wird. Die generierten Signale unterscheiden sich von echtem Verkehr in verschiedenen Aspekten, beispielsweise treten in der Praxis viele Kollisionenn auf. Diese Versuche sind außerhalb des Umfangs dieser Arbeit und müssen für eine Folgearbeit weiter untersucht werden.

In Abbildung 2.5 ist der Versuchsaufbau zu sehen. Links ist der Computer, welcher die ADS-B

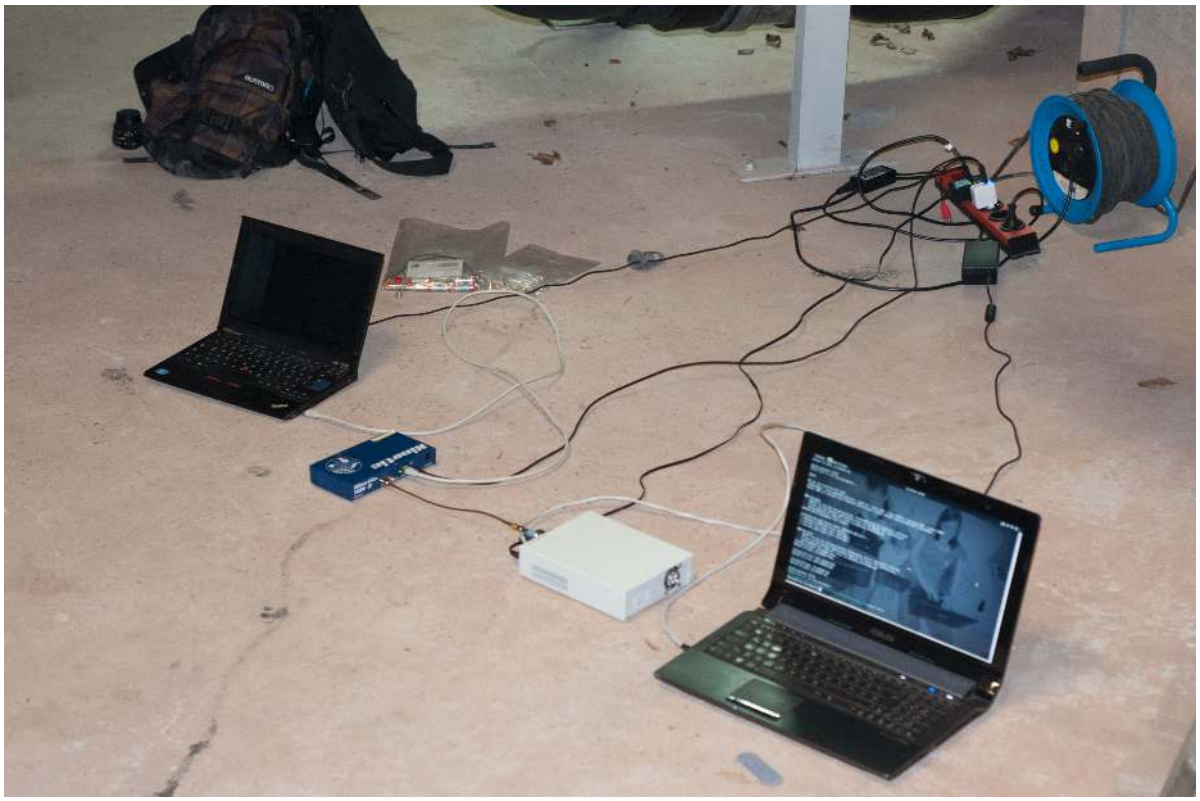


Abbildung 2.6: Foto des Versuchsaufbau im Keller der Universität.

Nachrichten erzeugt und zum USRP2 Sender weiterschickt. Ein zweiter PC protokolliert die Messergebnisse und ist entweder über USB oder Ethernet an den ADS-B/SDR Empfänger angeschlossen. An den Signalausgang des Senders ist eine Kombination von Dämpfungsgliedern direkt angeschlossen. Über ein SMA-Kabel inklusive passendem Adapter werden die einzelnen Empfänger verbunden.

Die Versuche wurden im Keller durchgeführt (Abbildung 2.6), um sicherzustellen , dass keinerlei Signale nach außen dringen und die Messungen selber nicht durch Interferenz gestört werden.

3 Auswertung

Die Versuche wurden jeweils mit den dazugehörigen Tools unter Linux oder im Fall des GNS-5890 Empfängers auf Android aufgezeichnet. Die Datenauswertung wurde in Python und NumPy geschrieben.

Die Daten wurden immer in den beschriebenen 30 dB Blöcken aufgenommen. Zu jedem ADS-B Paket und für den Anfang jedes Experimentes wurde jeweils der Zeitstempel gespeichert.

Zur Auswertung müssen die einzelnen Experimente aus der Gesamtaufnahme zuverlässig getrennt werden. In Abbildung 3.1 ist das verwendete Verfahren dargestellt. Während der Verarbeitung der ADS-B Nachrichten entstehen Latenzen, wie beispielsweise durch Interrupts, Scheduling des Prozesses, Dauer die aktuelle Uhrzeit abzufragen und andere Faktoren. Deswegen kann diese Trennung nicht nur anhand der verfügbaren Zeitstempel durchgeführt werden, stattdessen wird für die Trennung ein Intervall um den Beginn des Aufnahmestarts betrachtet. Durch den Zähler in den einzelnen ADS-B Nachrichten können die Aufnahmen zuverlässig getrennt werden, im Diagramm i ist dieser Zählerwert aus dem ME Feld als y-Achse dargestellt.

Insgesamt wurden fast 452 Millionen ADS-B Nachrichten versendet und 239 Millionen Nachrichten[1] aufgezeichnet und ausgewertet.

3.1 Probleme des Versuchsaufbaus

Während des Auswertens der Aufnahmen sind verschiedene Probleme mit dem Messaufbau aufgefallen, diese werden in diesem Abschnitt erläutert.

3.1.1 Schlechte Performance bei hohen Gainwerten

In den Messungen hat sich gezeigt, dass bei hohen Gainwerten die Signalqualität stark sinkt. Besonders kann man diesen Effekt an den Überlappungsgrenzen zwischen den verschiedenen Aufnahmeblöcken sehen, bei denen der Gain auf 30 steht. Dabei fällt auf, dass die Stärke des Effektes stark je nach Empfänger schwankt.

[1] Zahl ist nicht in Relation mit den gesendeten Nachrichten, da der RTL-SDR Empfänger 3 mal ausgewertet wurde.

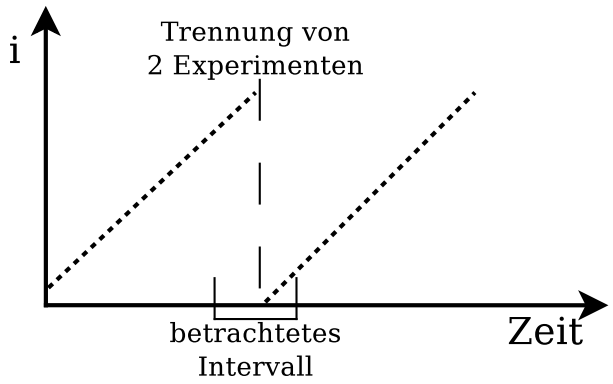


Abbildung 3.1: Darstellung des Verfahrens zur Trennung der einzelnen Experimente. Der Wert i ist der Zähler aus den empfangenen ADS-B Nachrichten. Der ungefähre Start eines Experiments ist bekannt. Es wird ein Intervall um diesen Punkt betrachtet und nach den minimalen empfangenen Zählerwert gesucht, um die Experimente danach zu trennen.

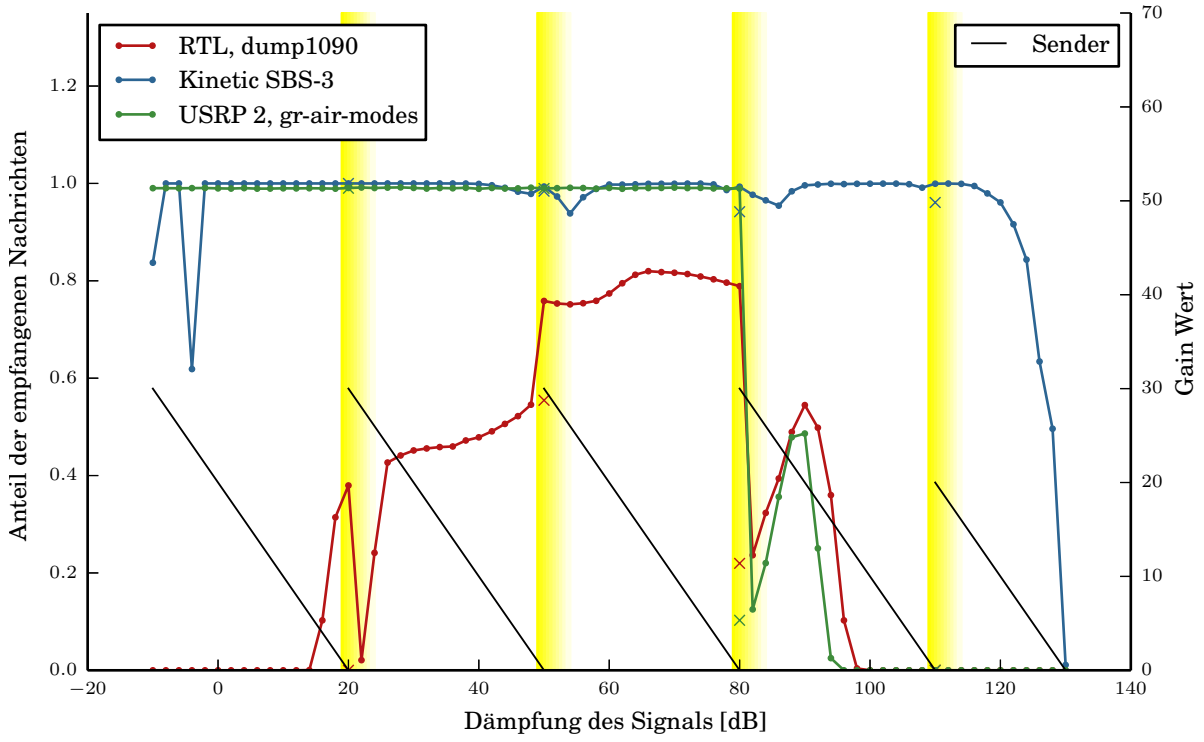


Abbildung 3.2: Darstellung der Clipping-Probleme des Senders. Senderate ist 100 $^{msg}/_{s}$. Gezeigt ist eine Auswahl von Empfängern. Die zusätzlichen Werte sind mit einem x markiert. Die Stellen mit Gain = 30 sind zusätzlich gelb unterlegt. Es sind deutliche Einbrüche bei einem Gain von über 20 zu erkennen.

Evtl. Könntest du die Stellen an denen der Effekt auftritt noch einmal sichtlich markieren

In Abbildung 3.2 ist der Verlauf der empfangenen Nachrichten bei variierender Dämpfung dargestellt, es wurde eine niedrige Senderate von 100 $^{msgs}/_{s}$ gewählt. Für die Dämpfungswerte 20 dB, 50 dB, 80 dB und 110 dB gibt es 2 Messwerte, der zweite Messwert wird durch ein Kreuz in der Farbe der Messreihe gekennzeichnet. Der als Kreuz gekennzeichnete Messwert wurde mit einem Gain von 30 dB und einer zusätzlichen Dämpfung von 30 dB in der Signalleitung gemessen, die andere Messung wurde bei einen Gain von 0 dB durchgeführt.

Es kann erkannt werden, dass bei einem Gain von 30 dB die Empfangsrate bei allen Empfängern einbricht. Besonders stark bricht die Rate bei dem RTL-SDR Empfänger ein. Im Bereich zwischen 80 und 100 dB ist zu sehen, dass die Einbrüche schon bei Gainwerten unter 30 dB anfangen.

Es ist wahrscheinlich, dass der USRP2 SDR Sender das Ausgangssignal bei hohen Gainwerten verzerrt. Wenn man Abbildung 3.6 betrachtet, ist es offensichtlich, dass ab ungefähr einem Gain von 20 dB die Signalstärke des Senders nicht mehr steigt.

Jedoch kann die genaue Fehlerquelle durch die durchgeführten Messungen nicht identifiziert werden. Um in den folgenden Betrachtungen dieses Problem des Senders sinnvoll einzubringen, werden die Experimente mit Gain 30 immer mit Kreuzen in den Plots markiert.

3.1.2 Störungen durch die Signalleitung

Beim Auswerten der Daten des ersten Durchlaufs wurde festgestellt, dass trotz einer extrem hohen Dämpfung (130 dB) immer noch Nachrichten empfangen wurden. Nach weiteren Untersuchungen stellte sich die Abstrahlung des Senders und des Übertragungskabels als Ursache des Signallecks heraus. Bei den Messungen wurde zuerst der Fehler gemacht, dass die Dämpfungsglieder auf der Seite des Empfängers montiert wurden. Dabei hat das Signal so stark gestreut, dass Empfänger die meisten Nachrichten empfangen konnten.

Als Maßnahme wurde der Bereich von $110 - 130$ dB neu vermessen. Dabei wurden die Dämpfungsglieder auf Senderseite montiert, sodass die Streuung minimiert wird. Zusätzlich wurde der USRP2 Sender weiter abgeschirmt, indem er in antistatische Folie eingepackt wurde. Die antistatische Folie wirkt als Faradayscher Käfig und schirmt alle hochfrequenten Signale nach außen ab. In Abbildung 3.3 ist der Vergleich zwischen den zwei Messaufbauten zus sehen. Bei dem korrigierten Messaufbau ist die Rate der empfangenen Nachrichten niedriger, da weniger Streuungen vorhanden sind, welche die Experimente stören.

In den folgenden Ergebnissen wurden die Messungen aus dem fehlerhaften Versuchsaufbau nicht berücksichtigt.

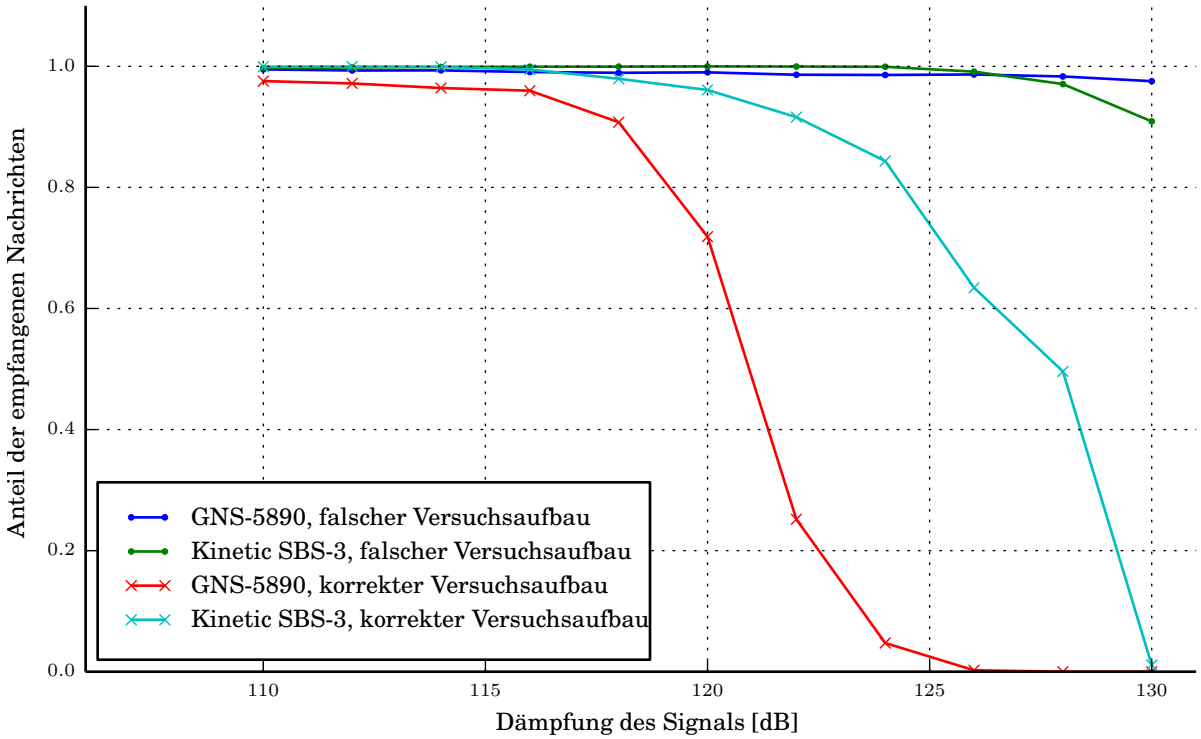


Abbildung 3.3: Performance des Kinetic SBS-3 und GNS-5890 Empfängers bei einer erneuten Messung. Bei der verbesserten Abschirmung des USRPs und Vermeidung von Signallecks fallen die Nachrichtenraten früher ab.

3.2 Vergleich der Empfänger

Signalstärke

In Abbildung 3.4(a) kann man den Anteil der empfangenen Nachrichten bei variierender Signalstärke sehen. Als Nachrichtenrate wurde 100 $^{msg}/_{s}$ gewählt, wodurch Nachrichtenverluste wegen zu hohem Durchsatz ausgeschlossen sind (vergleiche Abbildung 3.4(b)).

Der Kinetic SBS-3 Empfänger hat durchweg eine sehr gute Performance.

Der GNS-5890 bietet fast die gleiche Performance, wobei der Donut Effekt stärker ausgeprägt ist. Der Bereich, in welchem alle Nachrichten empfangen werden, ist kleiner als bei dem Kinetic Empfänger.

Die USRP Empfänger bietet bei hohen Signalpegeln eine sehr gute Empfangsleistung, ab 80 dB Dämpfung bricht der Anteil an empfangenen Nachrichten stark ein. Es ist bei niedrigen Signalstärken möglich, durch Wahl einer höheren Eingangsverstärkung Nachrichten zu empfangen.

Der RTL2832U Empfänger bietet eine längst nicht so gute Performance. Der optimale Bereich ist sehr klein und selbst bei optimaler Empfangsstärke werden nur 82 % der Nachrichten empfangen (dump1090).

Nachrichtenrate

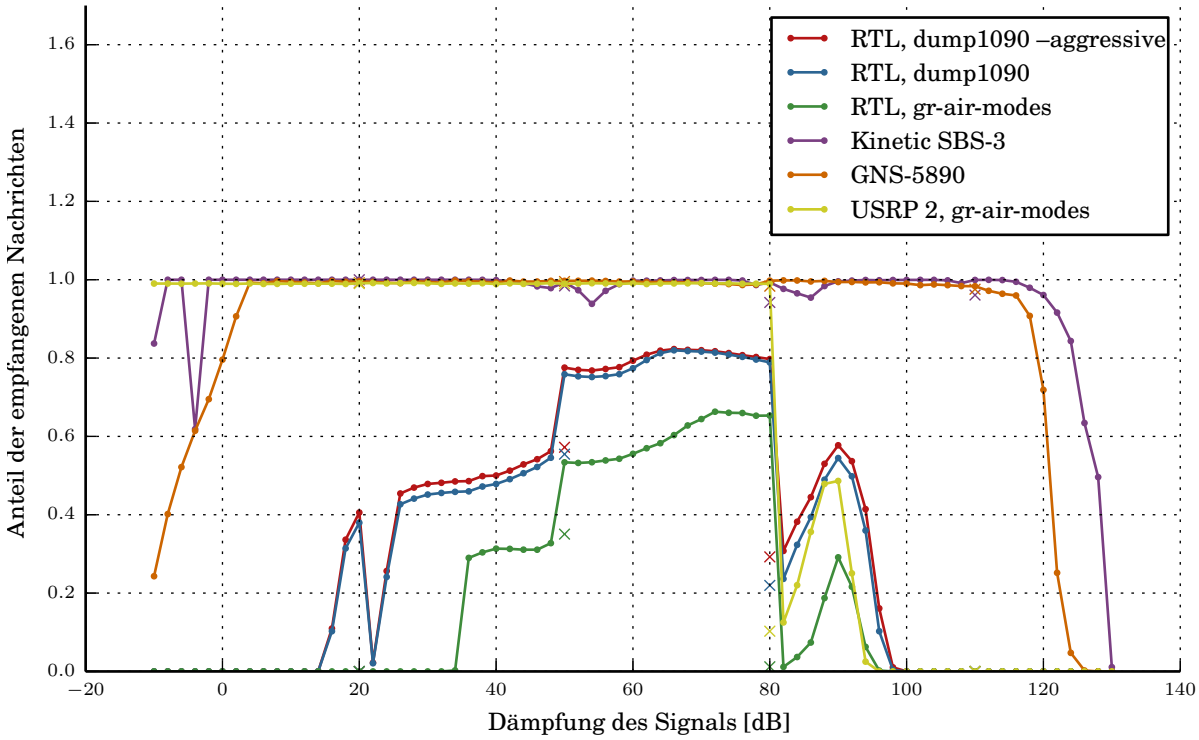
In Abbildung 3.4(b) ist die Empfangsrate bei variierender Senderate dargestellt. Die Dämpfung wurde mit 60 dB so gewählt, dass ein optimaler Empfang herrscht.

Es ist sichtbar, dass keiner der Empfänger Probleme mit Nachrichtenraten von bis zu 300 $^{msgs}/_{s}$ hat. Durch Performanceprobleme bricht beim GNS-5890 Empfänger bereits sehr früh der Anteil der empfangenen Nachrichten ein.

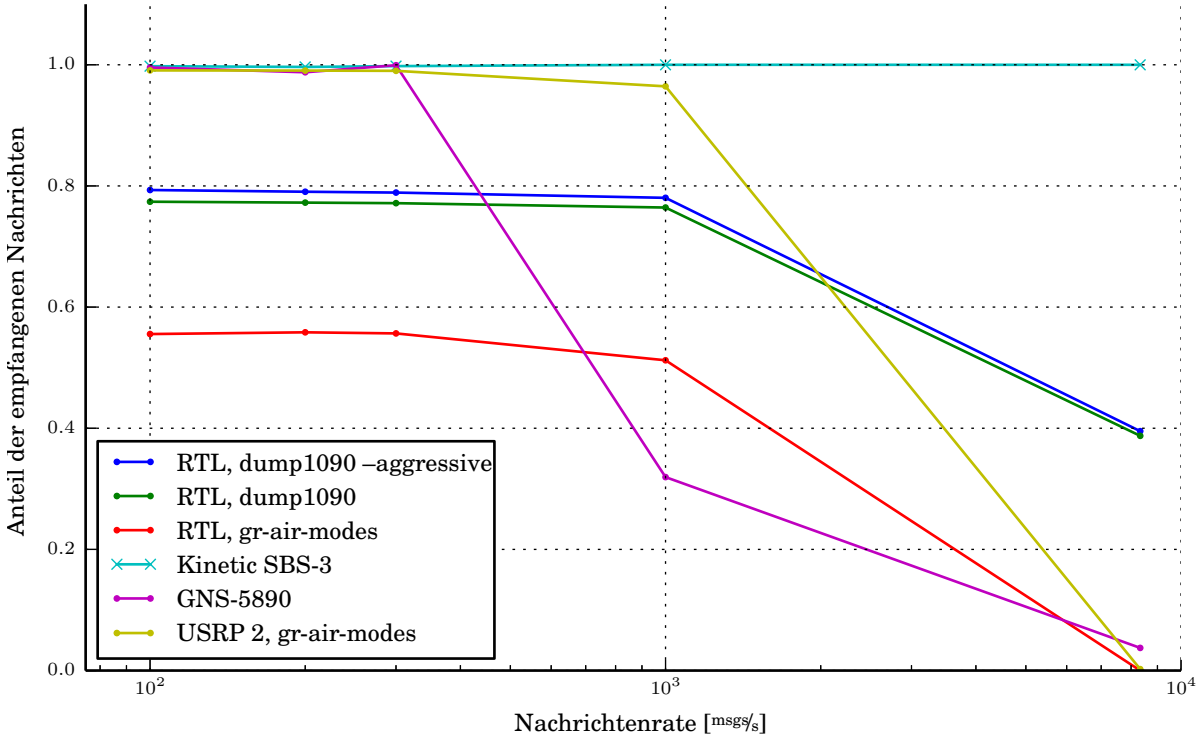
Der USRP Empfänger mit gr-air-modes decodiert bei der maximalen Nachrichtenrate nur noch wenige richtige Pakete. Es liegt wahrscheinlich an einem Softwareproblem des ADS-B Decoders, da die Software durch die enge Aneinanderreihung nicht mehr die korrekten Präambeln erkennt.

Beeindruckend ist wieder die Performance des Kinetic SBS-3 Empfängers, der gleichbleibende Leistung unabhängig von der Senderate erbringt. Sogar die maximale Rate wird ohne Probleme empfangen.

Bei der Nachrichtenrate gibt es verschiedene Faktoren, welche die Messergebnisse beeinflussen. Dazu zählen der Dekoder, die Schnittstelle zum PC und die Datenspeicherung. Der Dekoder ist der wichtigste Faktor. Bei dump1090 und gr-air-modes kann man ausschließen, dass die beiden anderen Faktoren Einfluss haben, da sie vollständig auf dem



(a) Darstellung der Rate der empfangenen Nachrichten bei variabler Dämpfung des Signals. Die Rate der Nachrichten ist 100 $^{msg}/_{s}$. Die Kreuze stellen die Experimente mit Gain 30 dar.



(b) Diagramm des Anteils der empfangenen Nachrichten bei variabler Nachrichtenrate.

Abbildung 3.4: Vergleich aller Empfänger.

PC ausgeführt werden. Fehlende Rechenleistung und Engpässe bei der Schnittstelle würden sich durch Nachrichtenverluste zwischen SDR-Empfänger und PC zeigen. Diese sind bei den Messungen nicht aufgetreten. Besonders die Implementierung der gr-air-modes Software hat Probleme mit der höchsten Nachrichtenrate. Die beiden Hardware ADS-B Empfänger haben anscheinend einen robusteren Decoder implementiert. Der GNS-5890 ist wahrscheinlich durch die USB-Schnittstelle beschränkt, da dort nur eine beschränkte Bandbreite zur Verfügung steht. Aus der leistungsfähigen Ethernet Schnittstelle des Kinetic SBS-3 Empfängers lässt sich dessen gute Performance erklären.

In Tabelle 3.1 ist ein Gesamtvergleich der verschiedenen Empfänger.

			gr-air-modes		dump1090[2]
	Kinetic SBS-3	GNS-5890	USRP2	RTL2832U	RTL2832U
Empfang (Dämpfung 60 dB) 100 msgs/s	100 %	100 %	99 %	56 %	77 %
Empfang (Dämpfung 60 dB) 8333 msgs/s	100 %	4 %	0 %	0 %	39 %
Empfang (Geschwindigkeit 100 msgs/s)	96 %	89 %	66 %	18 %	32 %
Empfang (Geschwindigkeit 300 msgs/s)	96 %	89 %	66 %	17 %	31 %
Empfang (Geschwindigkeit 1000 msgs/s)	96 %	30 %	65 %	16 %	31 %
Anzahl inkorrekter Nachrichten	4786	224	2668	0	58

Tabelle 3.1: Gesamtvergleich der Empfänger. Die Anzahl inkorrekter Nachrichten ist die Summer aller Experimente.

[2]Normaler Modus

Senderate [$^{msgs}/_{s}$]	Anteil an inkorrekten Paketen
100	0,197 %
200	0 %
300	0 %
1000	0 %
8333	$2,66 \cdot 10^{-6}$ %

Tabelle 3.2: USRP2: Prozentualer Anteil an inkorrekt empfangenen Paketen.

3.3 Diskussion der Ergebnisse

3.3.1 Kinetic SBS-3

Der Kinetic SBS-3 hat durchweg eine sehr gute Performance. Von allen Empfängern kann er den größten Bereich an Signalstärken decodieren. Er bewältigt auch sehr hohe Nachrichtenraten und als einziger Empfänger die maximalen Rate fast verlustfrei.

In Abbildung 3.5 ist der Anteil an empfangenen Nachrichten dargestellt und die gleichbleibend gute Performance zu sehen.

3.3.2 USRP2 mit gr-air-modes

Die SDR Software gr-air-modes kann auch das Signallevel (RSSI) des empfangenen Signales bestimmen. In Abbildung 3.6 wird dieser dargestellt. Am Anfang des Kapitels wurde bereits erwähnt, dass bei hohen Gainwerten des Senders der Anteil der empfangenen Nachrichten sinkt. Mit diesem Graphen wird die Vermutung bestätigt, dass die Ausgangsverstärkung ab einem gewissen Gainwert gesättigt ist.

Es wäre zu erwarten, dass die RSSI linear sinkt, aber an den Punkten mit Gain 30 (Dämpfung 20, 50, 80 dB) bricht der RSSI-Wert ein. Diese Beobachtung stärkt die Vermutung, dass bei hohen Gainwerten das Signal zusätzlich verzerrt wird.

Der Gainwert verhält im Bereich von $0-18$ ungefähr linear in Relation zum RSSI. Als Folgerung sollten für zukünftige Messungen nur Gainwerte von $0-18$ für das SBX Daugtherboard verwendet werden. Bei höheren Gainwerten sinkt die Signalqualität und die Signalstärke stimmt nicht mit dem Gainwert überein.

Es folgt eine Betrachtung der inkorrekt empfangenen Nachrichten. Mit inkorrekter Nachricht ist gemeint, dass eine Nachricht empfangen wurde, die nicht von dem Experiment gesendet worden ist. In Tabelle 3.2 wird der Anteil der inkorrekt empfangenen Nachrichten gegenüber der Senderate betrachtet.

Insgesamt sind nur wenige der empfangenen Pakete inkorrekt. Die Ausnahme ist, dass bei der Nachrichtenrate von 100 $^{msgs}/_{s}$, der Anteil bei 0,197 % liegt und damit verhältnismäßig

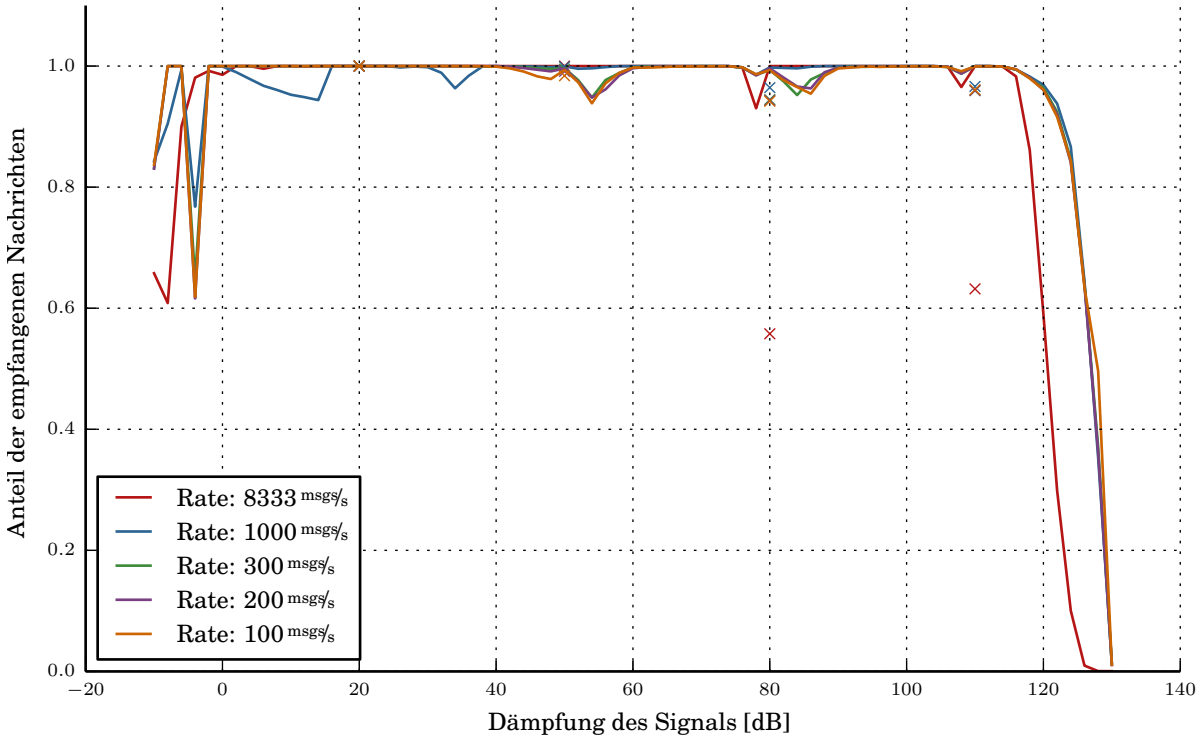


Abbildung 3.5: Darstellung der Rate des Kinetic SBS-3 Empfängers. Die Kreuze stellen die Experimente mit Gain 30 dar.

hoch ist. Die hohe Fehlerrate tritt nur bei der niedrigen Senderate auf, da dort die Zeitdauer, bei der kein gesendetes Signal auf der Leitung liegt besonders lange ist. Dadurch werden zufällige Nachrichten aus dem Rauschen herausgelesen.

Diese Annahme wird bestätigt, wenn das Signalevel der Nachrichten betrachtet wird. Bei korrekten Nachrichten ist das Signallevel im Schnitt $-5,3\,\mathrm{dB}$ und bei inkorrekten Nachrichten im Schnitt $-23\,\mathrm{dB}$. Hier ist zu sehen, dass die inkorrekten Nachrichten aus dem Leitungsrauschen stammen.

Diese Fehler können verschiedene Ursprünge haben. Die Software benutzt einen gleitenden Mittelwert des empfangenen Signals, um das Grundrauschen zu ermiteln. Bei hoher Aktivität auf dem Kanal steigt dieser an und bei niedriger Aktivität sinkt er, sodass der Empfänger sensitiver wird. Ein weiterer Faktor ist der Threshold für das Signallevel eines Paketes, dieser filtert alle Pakete aus, bei denen gilt: RSSI Paket $<$ RSSI Noisefloor $+$ Threshold. Als Threshold wurde 7 dB gewählt, bei einem höheren Wert empfängt er signalschwache Pakete nicht mehr.

Ein weiterer Faktor ergibt sich durch den Versuchsaufbaus. Durch den abgeschirmten Versuchsaubau ist das Rauschen auf der Leitung besonders niedrig, sodass manche Störungen stärker auftreten können.

Ein weiteres Problem von gr-air-modes ist, dass bei der höchsten Senderate die Anzahl an empfangenen Nachrichten sehr niedrig ist. Insgesamt wurden bei allen durchgeführten

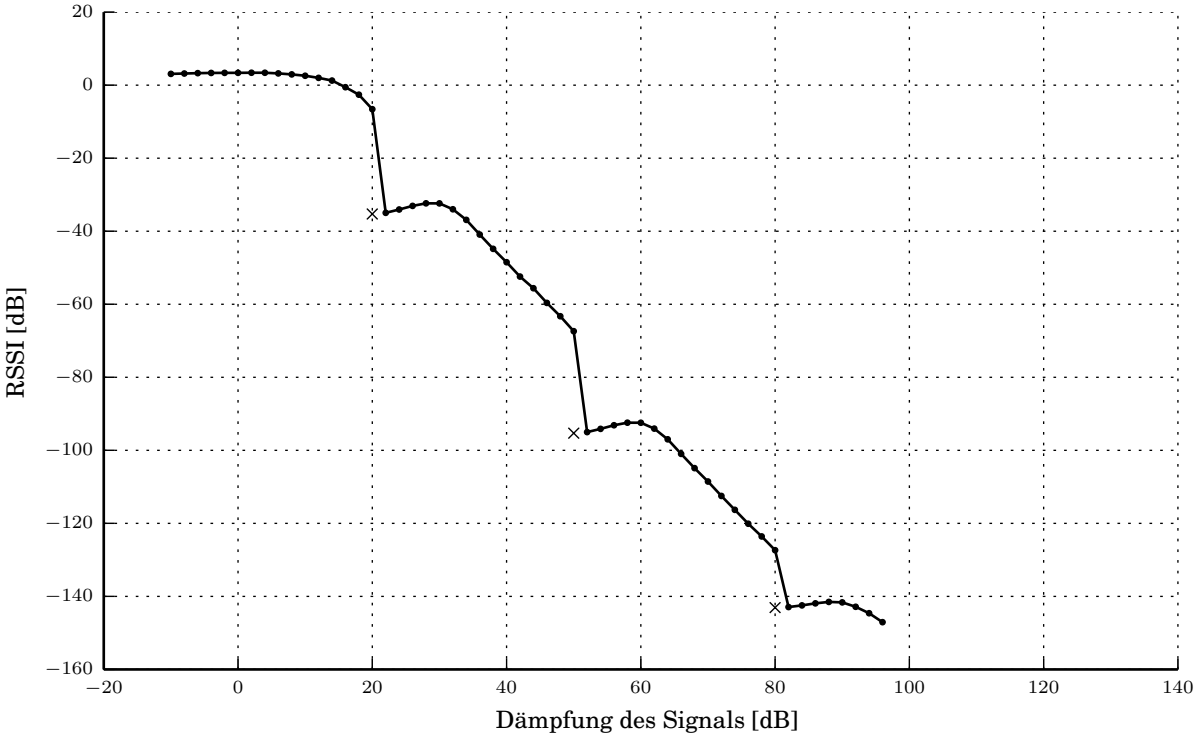


Abbildung 3.6: USRP2: RSSI der aufgezeichneten Pakete. Es ist zu sehen, dass die Signalstärke ab einem Gainwert von 20 des Senders nicht mehr ansteigt. Außerdem ist zu erkennen, dass bei einem Gain von 30, die RSSI sinkt. Die Kreuze stellen die Experimente mit Gain 30 dar.

Experimenten mit einem Spacing von 0 nur 2,27 Prozent der Nachrichten empfangen.

Wenn diese Experimente mit einem Spacing von 0 genauer betrachtet werden, ist auffällig, dass immer ein kurzer Block fast perfekt empfangen wird und danach keine Nachrichten mehr empfangen werden. In Abbildung 3.7 ist solch eine Aufnahmen abgebildet, die Dämpfung ist 20 dB. Es wird der zeitliche Verlauf des Anteils der empfangenen Nachrichten dargestellt. Dafür wird ein gleitender Mittelwert eingesetzt und jedes Fenster hat eine Größe von 5000 Nachrichten. Um den Zeitpunkt $t = 20\,\mathrm{s}$ wurden insgesamt 4288 Nachrichten empfangen. Für den Rest der Aufnahme wurden keine Nachrichten mehr empfangen.

Dieses Problem muss weiter untersucht werden. Wahrscheinlich ist ein Implementierungsfehler in gr-air-modes vorhanden.

3.3.3 RTL2832U

In Abbildung 3.8 kann erkannt werden, dass die automatische Gain Kontrolle vom Tuner funktioniert. Der RSSI ist über große Strecken konstant. Ab der Dämpfung von 70 dB fällt die Signalstärke linear ab.

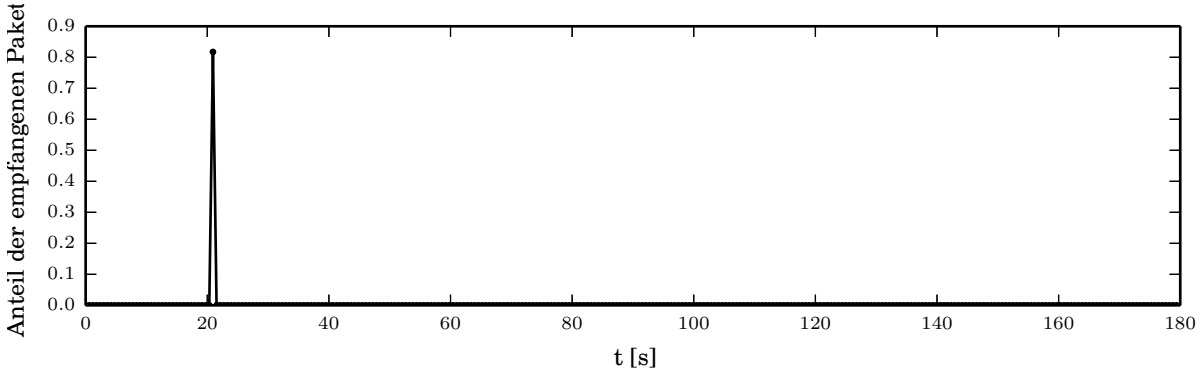


Abbildung 3.7: USRP2: Darstellung einer Aufnahme mit Dämpfung 20 dB und Spacing 0 μs. Zu sehen ist der Anteil der empfangen Nachrichten gegenüber der Zeit. Während den ganzen Experinents wird nur bei $t = 20$ s ein Block Nachrichten empfangen.

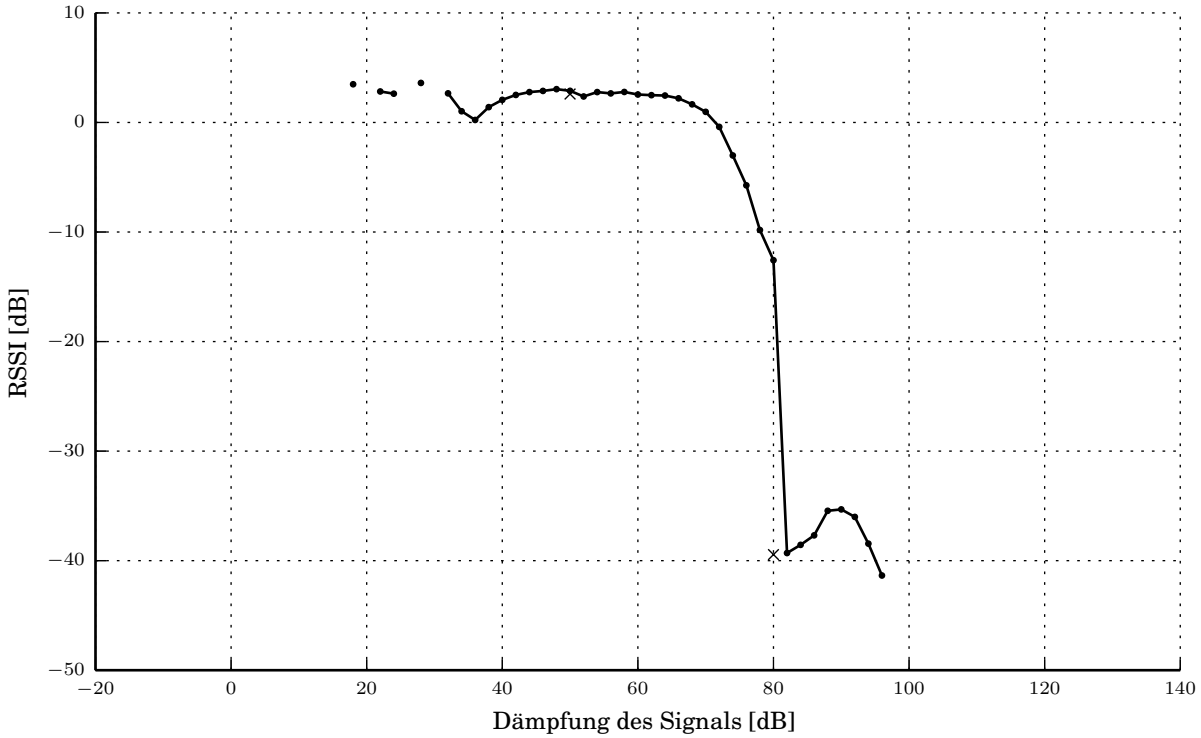


Abbildung 3.8: RTL2832U, gr-air-modes: RSSI der aufgezeichneten Pakete. Der RSSI ist durch die automatische Gainkontrolle des Empfängers über große Strecken konstant. Die Kreuze stellen die Experimente mit Gain 30 dar.

	korrekt	inkorrekt
nicht repariert	23,523,565	0
repariert	3,788,220	58

a normaler Modus

	korrekt	inkorrekt
nicht repariert	23,062,997	0
repariert	5,570,702	4106

b aggressiver Modus

Tabelle 3.3: dump1090: Betrachtung der decodierten Nachrichten aller Experimente. Inkorrekt: Es wurde eine Nachricht empfangen, welche nicht gesendet wurde. Repariert: dump1090 hat erfolgreich CRC-Korrektur durchgeführt.

Vergleich dump1090 und gr-air-modes

Zum Vergleich der dump1090 und der gr-air-modes Software wurden die gleichen IQ-Signalaufzeichnungen des RTL2832U Empfängers verwendet. In Abbildung 3.9 sind die Empfangsraten der beiden Programme zu sehen. Im direkten Vergleich der beiden Decoderprogramme bietet dump1090 ein besseres Empfangsverhalten, der Anteil an empfangenen Nachrichten ist durchweg höher. Außerdem werden bei niedrigen Signalstärken noch Pakete empfangen, bei denen gr-air-modes keine mehr empfängt. Ein Problem der gr-air-modes Software ist, dass keine Nachrichten bei der höchsten Senderate empfangen werden, dieses Problem wird später genauer betrachtet.

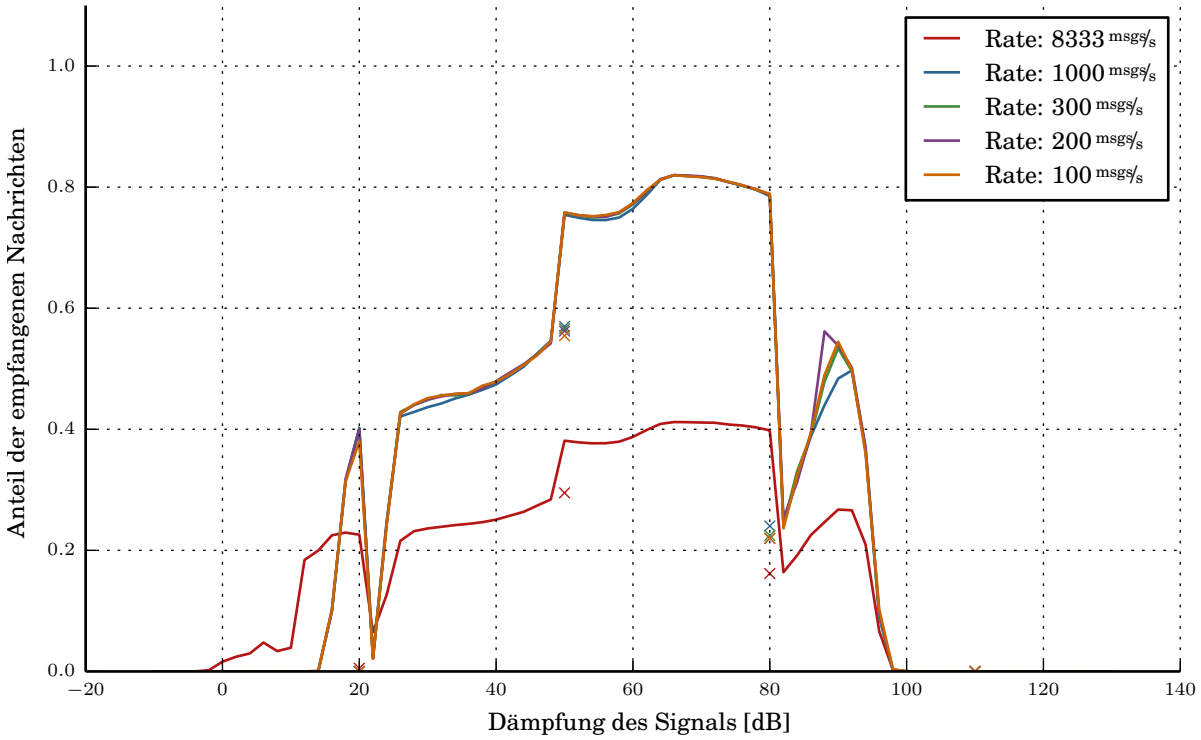
Dump1090: Vergleich Aggressive und Normal Modus

Ein Satz IQ-Daten der Experminente benötigt mit der --aggressive Option 6239 s Rechenzeit, wogegen der Normale Modus nur 502 s gerechnet hat. Der zusätzliche Rechenaufwand kommt durch 2 Faktoren zustande. Durch die Korrektur von bis zu 2 Bitfehlern ist der Rechenaufwand bei erkannter Präambel und falscher CRC viel höher. Gleichzeitig werden auch mehr Nachrichten auf dem Medium erkannt, da die Demodulation fehlertoleranter arbeitet und bis zu 2 Demodulationsfehler erlaubt.

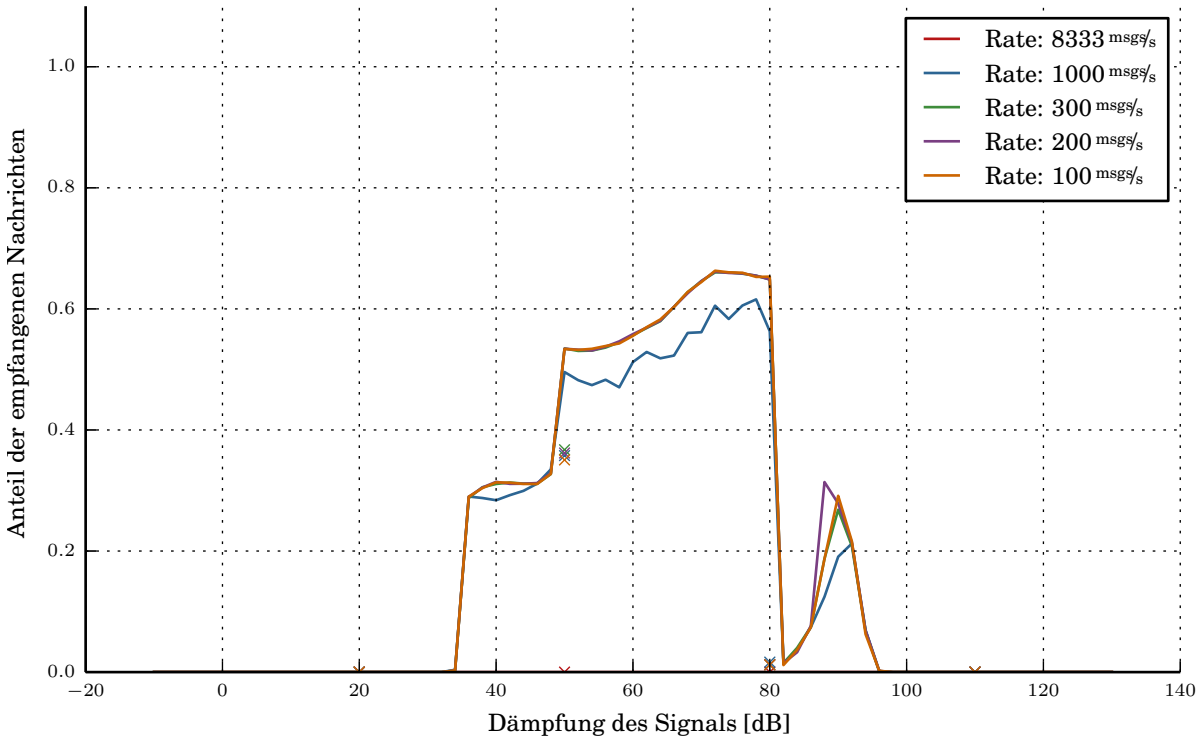
Die Korrektur der Bitfehler ist zusätzlich ineffizient implementiert. Die Bits werden nacheinander negiert und bei jeder Iteration wird die Checksumme vollständig neu berechnet. Stattdessen könnte über eine CRC-Tabelle diese Operation um eine Vielfaches beschleunigt werden, dann müsste pro Iteration nur eine XOR-Operation durchgeführt werden.

In Tabelle 3.3 werden die empfangenen Pakete betrachtet. Repariert bedeutet, dass die Software beim Decodieren einen CRC-Fehler festgestellt hat und dieses Paket reparieren konnte. Der aggressive Modus erzeugt mehr falsche Pakete mit richtiger Checksumme. Überraschend ist, dass der normale Modus 2 % mehr Nachrichten mit korrekter Checksumme empfängt, was mit dem Problem zu tun hat, dass im aggressiven Modus mehr Präambeln detektiert und dadurch nicht die eigentlichen Präambeln gefunden werden.

Im Anhang ist eine genauere Auflistung zu finden (Tabelle 6.1). Es werden 1 % weniger Präambeln erkannt, aber der aggressive Modus kann 47 % mehr Checksummen korrigieren.



(a) Einsatz von dump1090. Konstante Performance über alle Geschwindigkeiten, nur bei der höchsten Packetrate halbiert sich etwa die Rate der empfangenen Nachrichten.



(b) Einsatz von gr-air-modes. Im Vergleich zu (a) werden immer weniger Nachrichten empfangen. Der Donut-Effekt ist stärker ausgeprägt. Bei der höchsten Signalrate wird nichts mehr empfangen.

Abbildung 3.9: RTL2832U: Vergleich der Empfangsrate der verschiedenen Software-Decoder bei variierender Dämpfung und Senderate. Die Kreuze stellen die Experimente mit Gain 30 dar.

```
1304 void detectModeS(uint16_t *m, uint32_t mlen){
1334   for (j = 0; j < mlen - MODES_FULL_LEN*2; j++) {
1471       if (errors == 0 || (Modes.aggressive && errors < 3)) {
1588           /* Skip this message if we are sure it's fine. */
1589           if (mm.crcok) {
1590               j += (MODES_PREAMBLE_US+(msglen*8))*2;
1591           }
1592       } else {
1492       }
1518   }
1533 }
```

Abbildung 3.10: Ausschnitt aus `dump1090.c`. Falls ein Paket erfolgreich empfangen wurde (Zeile 1589), wird der Zähler `j` zuerst um die Länge des empfangenen MODE-S Paketes erhöht und danach im Schleifenkopf um 1. Dadurch werden $0,5\,\mu s$ zu viel im Datenstrom übersprungen. Die Folge ist, dass zwei direkt aufeinander folgende Pakete nicht empfangen werden.

Dort wird bestätigt, dass der aggressive Modus weniger Nachrichten fehlerfrei empfängt, aber durch die Korrektur in der Summe mehr Nachrichten empfangen kann.

Bei der höchsten Nachrichtenrate werden 38,7 % (normal) bzw. 37,8 % (aggressiv) der Nachrichten empfangen, dieser Wert ist etwa halb so groß wie bei den anderen Senderaten. Aus dieser Beobachtung kann gefolgert werden, dass der Decoder Probleme mit dieser hohen Senderate hat.

In Abbildung 3.10 ist ein Ausschnitt der Funktion dargestellt, welche die MODE-S Nachrichten dekodiert. Die Variable `m` ist der bereits synchronisierte Datenstrom, eine Zelle sind $0,5\,\mu s$ des Signals. Falls eine Nachricht erfolgreich dekodiert wurde und die Checksumme korrekt ist, wird die Nachricht als erkannt markiert und j wird erhöht. Dabei wird der Zähler zuerst in Zeile 1590 um `MODES_PREAMBLE_US+(msglen*8)*2` erhöht, was der Zeitdauer einer MODE-S Nachricht von 120 oder $56\,\mu s$ entspricht. Im Schleifenkopf wird `j` nochmals um $1 \sim 0,5\,\mu s$ erhöht. Dadurch wird der nächste Datenwert hinter der empfangenen Nachricht übersprungen. Dieses hat zur Folge, dass die Präambel einer direkt nachfolgenden Nachricht übersprungen wird.

Zum Test wurde Zeile 1590 abgeändert, sodass `MODES_PREAMBLE_US+(msglen*8)*2-1` auf j addiert wird. Danach wurde das Programmm mit einer Aufnahme mit Spacing 0 (Dämpfung 60 dB) getestet. Mit der ursprünglichen Programmversion wurden 3.800.543 Pakete empfangen., mit der korrigierten Version 7.149.844 Pakete. Mit diesem Bugfix wurden bei diesem Testfall 88 % mehr Pakete empfangen. Da das Ziel ist die im Internet verbreitete Variante zu untersuchen, beziehen sich alle Messungen im Folgenden auf die

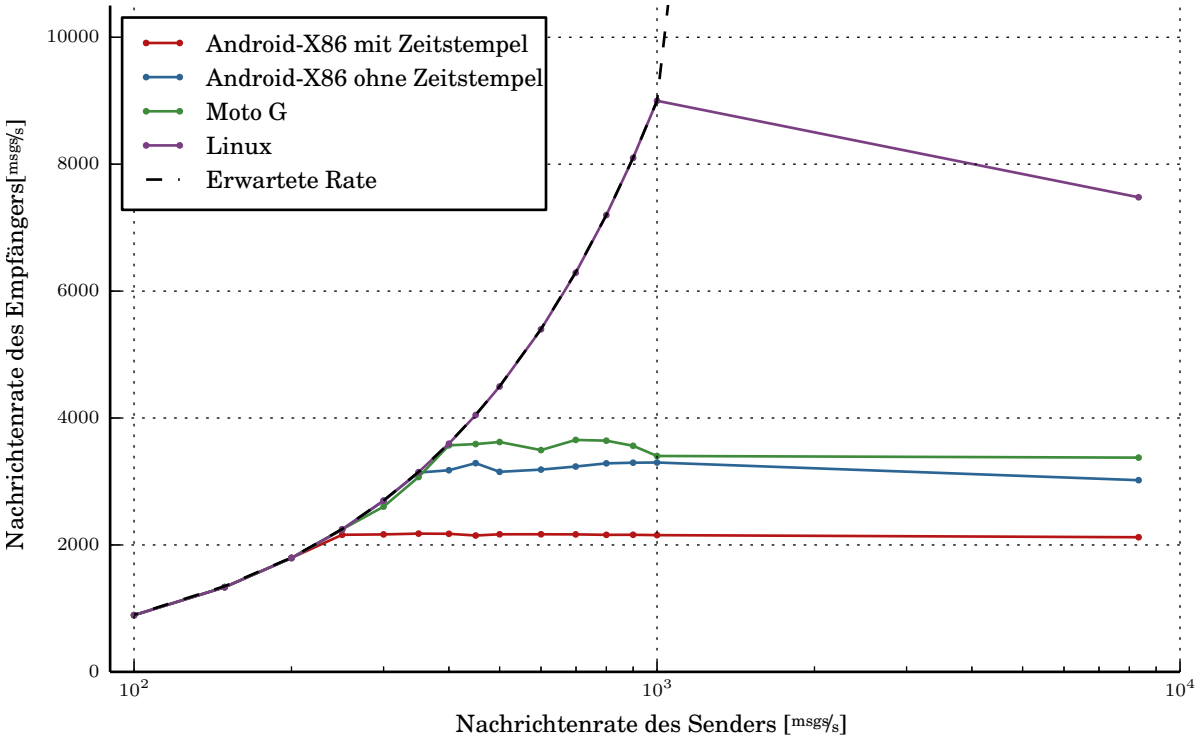


Abbildung 3.11: GNS-5890: Betrachtung der maximalen Nachrichtenrate (Dämpfung 60 dB). Maximale Empfangsrate je nach Hardware unterschiedlich.

originale, fehlerhafte Version.

3.3.4 GNS-5890

In Abbildung 3.11 ist die maximale Nachrichtenrate dargestellt, als Dämpfung ist 60 dB gewählt. Die graue Datenreihe stellt das optimale Verhalten dar, wenn alle gesendeten Nachrichten empfangen werden. Die unteren drei Datenreihen wurden mittels der Android Anwendung empfangen, welche im Kapitel 4 vorgestellt wird. Die obere Reihe wurde unter Linux aufgenommen. Dort ist der Rechenaufwand zum Speichern niedriger und die Plattform verfügt über mehr Rechenleistung, zusätzlich puffert das CDC-ACM Modul die Ausgabe des Empfängers.

Als Testgerät wurde einmal das Notebook mit Intel Atom CPU genutzt (Android-X86), welches auch für die anderen Tests eingesetzt wurde. Das andere Testgerät ist ein Motorola Moto-G, welches mit einem 4 Kerne Prozessor ausgestattet ist, jeder Kern ist mit 1,2 GHz getaktet.

Es kann erkannt werden, ab welchem Punkt das Maximum der empfangenen Nachrichtenrate erreicht ist. Je nach Rechenleistung des Testsystems ist die maximale Rate unterschiedlich. In Tabelle 3.4 sind nochmals die maximalen gemessenen Raten notiert.

Plattform	maximale gemessene Nachrichtenrate [$^{msgs}/_{s}$]
Linux	999,9
Motorola Moto G	406,1
Android-X86 ohne Zeitstempel	366,6
Android-X86 mit Zeitstempel	242,1

Tabelle 3.4: GNS-5890: Die maximal gemessenen Empfangsraten.

Durch die gute Performance des GNS-5890 Empfängers und dessen kleine Bauform. Wurde dieser Empfänger gewählt, um ihn unter Android zu evaluieren.

4 Entwicklung eines ADS-B Rekorders

Als Empfänger wurde der GNS-5890 USB-Stick eingesetzt. Durch die kleine Bauform und die USB-Schnittstelle eignet er sich besonders als portabler Empfänger. Es gibt zwar bereits eine Android-Anwendung vom Hersteller für diesen Empfänger, aber diese bietet nur eine Anzeige der Flugpositionen.

4.1 Design Entscheidungen

4.1.1 Backend

Als Mindestanforderung für die Android Plattform wurde die Android Version 4.0 gewählt. Wie man in Tabelle 4.1 sehen kann, liegt der Anteil der Geräte mit Android Version > 4.0 bei ungefähr 78,6 %, damit ist gewährleistet, dass die Anwendung auf den meisten Endgeräten lauffähig ist. Diese Statistik beruht auf den Daten von Geräten, welche am 4. Februar 2014 den Android Play Store besuchten. Wahrscheinlich ist der Anteil der Geräte mit neueren Versionen (>4.0) etwas überzeichnet, da vor allem Anwender mit neueren und schnelleren Geräten häufiger den Android Play Store betreten.

Zur Speicherung der ADS-B Pakete wurde eine SQLite Datenbank gewählt. Gegenüber der Speicherung in einem einfachen Textformat bietet das den Vorteil, dass sowohl der benötigte Speicherplatz, als auch die Latenz beim Einfügen neuer Daten geringer ist.

Version	Codename	API	Verbreitung
2.2	Froyo	8	1,3 %
2.3.3- 2.3.7	Gingerbread	10	20,0 %
3.2	Honeycomb	13	0,1 %
4.0.3- 4.0.4	Ice Cream Sandwich	15	16,1 %
4.1.x	Jelly Bean	16	35,5 %
4.2.x		17	16,3 %
4.3		18	8,9 %
4.4	KitKat	19	1,8 %

Tabelle 4.1: Verbreitung der einzelnen Android Versionen [11].

Gerät	Speicherart	Dauer	Dateigröße
Samsung Galaxy S3	JSON	8935 ms	2100 kb
	SQLite	8480 ms	1204 kb
Emulator	JSON	25836 ms	2100 kb
	SQLite	7761 ms	1204 kb

Tabelle 4.2: Vergleich der Speichermethoden. Die SQLite Datenbank ist auf beiden getesteten Plattformen schneller und benötigt weniger Speicherplatz.

Es wurde ein einfacher Vergleich der Geschwindigkeiten der beiden Datenformate durchgeführt. Als Textformat wurde JSON (JavaScript Object Notation) gewählt, da dieses Format wenig Overhead hat, weit verbreitet und einfach per Software parsebar ist. Zur Speicherung wurde die Android eigene Klasse `android.util.JsonWriter` eingesetzt. Als Datenbank wurde SQLite eingesetzt, da sie ressourcenarm und portabel ist. Die Speicherung in SQLite wurde mittels einer Transaktion implementiert, dadurch werden alle Einträge in einer atomaren Operation angelegt. Dieses Vorgehen bietet eine bessere Performance, als alle Einträge in der Datenbank einzeln anzulegen. Das Testprogramm legte 6000 Einträge an.

Beide Verfahren waren auf dem Smartphone (Samsung Galaxy S3) ungefähr gleich schnell, aber die Dateigröße der SQLite-Datenbank war erheblich kleiner, siehe Tabelle 4.2. Deswegen ist die Wahl der Datenspeicherung auf eine SQLite-Datenbank gefallen.

4.1.2 Oberfläche

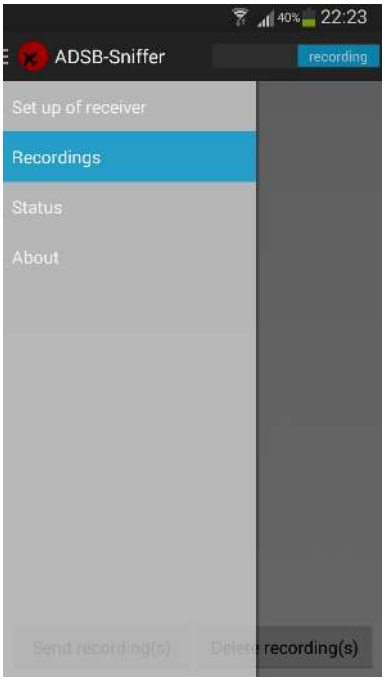
Die Oberfläche besteht aus mehreren Ansichten. Zum Navigieren der Ansichten wurde ein *Navigation Drawer* genommen (Abbildung 4.1(a)).

Einstellungen

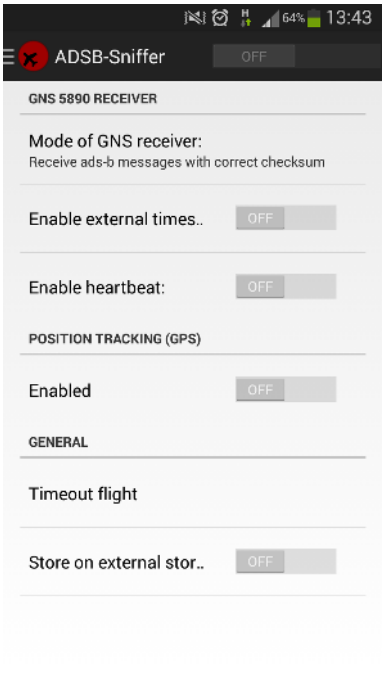
Die erste Ansicht, welche auch bei dem Start angezeigt wird, ist der Einstellungsdialog (Abbildung 4.1(b)), er bietet Möglichkeiten zum Einstellen aller Optionen des Empfängers. Es gibt 4 unterschiedliche Betriebsmodi des Empfängers:

- Empfänger aus
- Empfange alle Nachrichten
- Empfange alle ADS-B Nachrichten
- Empfange alle ADS-B Nachrichten mit korrekter Checksumme

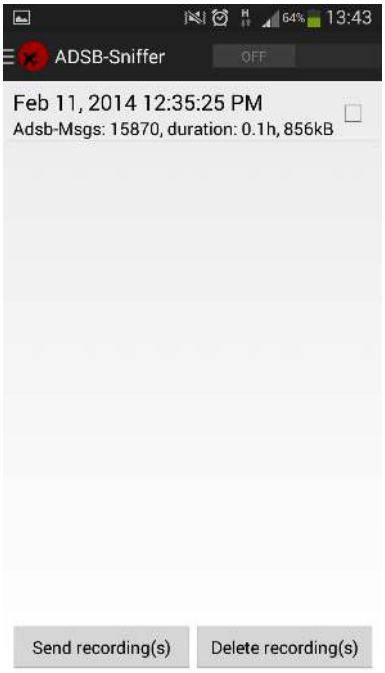
Es kann eingestellt werden, ob die empfangenen Nachricht mit einem Zeitstempel vom Empfänger versehen werden.



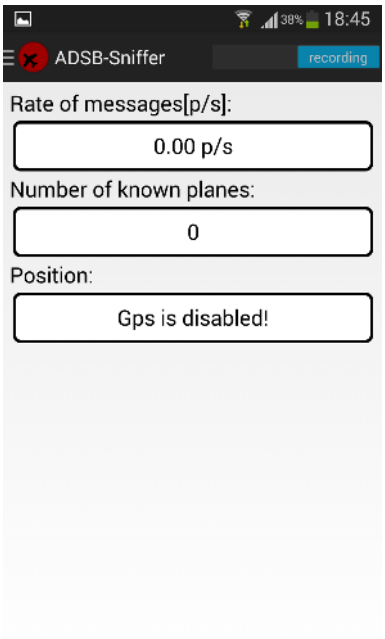
(a) Das Navigations Schubfach.



(b) Der Einstellungsdialog



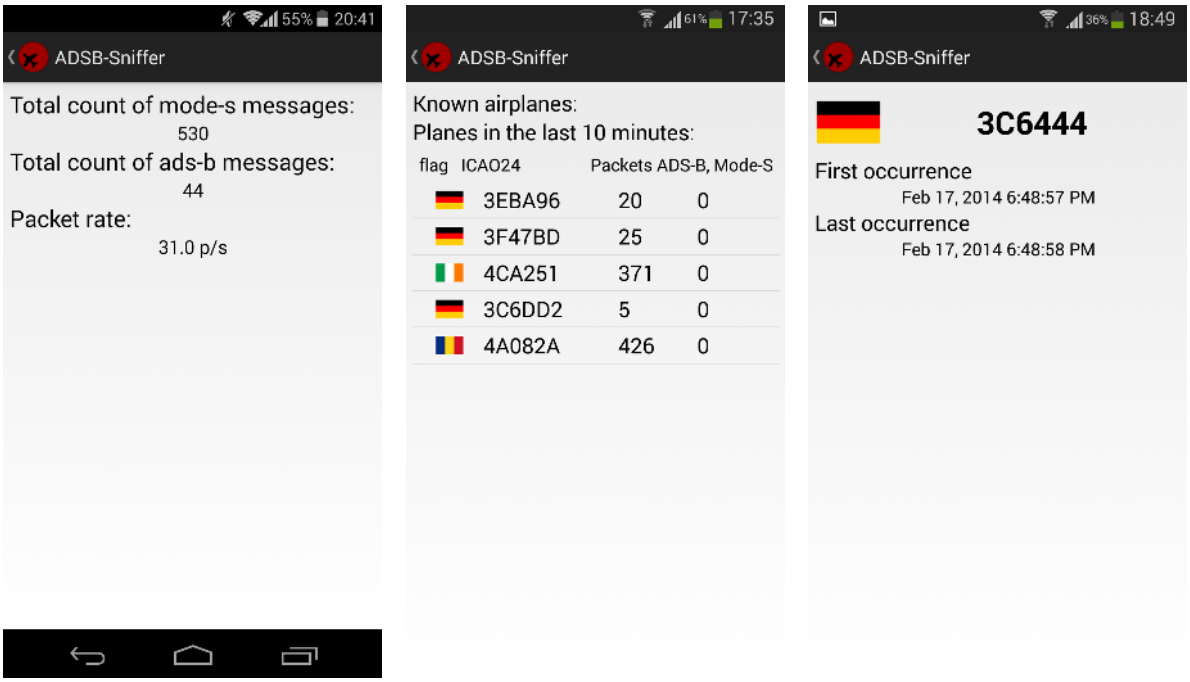
(c) Die Liste der zurzeit gespeicherten Aufnahmen



(d) Der Statusbildschirm

(e) Die About Ansicht

Abbildung 4.1: Die Hauptansichten der Android Anwendung.



(a) Detailansicht der empfangenen Nachrichten.

(b) Detailansicht für alle Flugzeuge, welche zurzeit empfangen werden.

(c) Die Detailansicht für ein einzelnes Flugzeug.

Abbildung 4.2: Die Unterbildschirme.

Das GPS Tracking kann aktiviert werden, dann wird etwa jede Sekunde die aktuelle Position gespeichert.

Die Anwendung versucht ADS-B Nachrichten einzelnen Flügen zuzuordnen. Wenn innerhalb des Timeout Zeitraums keine weiteren Nachrichten dieses Flugzeuges empfangen werden, gilt der Flug als beendet. Wenn nach diesem Zeitraum wieder Nachrichten dieses Flugzeuges empfangen werden, wird angenommen, dass ein neuer Flug angefangen wurde, beispielsweise da das Flugzeug in einem Flughafen zwischengelandet ist. Als Timeout ist 10 min eingestellt.

Es ist eine Option zur Wahl des Speicherortes der Aufnahmen vorhanden. Je nach Länge des Fluges können große Datenmengen anfallen, deswegen ist es sinnvoll, diese Option anzubieten.

Recordingliste

Die nächste Ansicht ist die Darstellung der zurzeit gespeicherten Aufnahmen (Abbildung 4.1(c)). Pro Aufnahme wird entweder der Name oder das Anfangsdatum der Aufnahme angezeigt. Durch Klicken auf die Aufzeichnung kann ein Name vergeben werden. Es werden noch weitere Informationen zu jeder Aufnahme angezeigt.

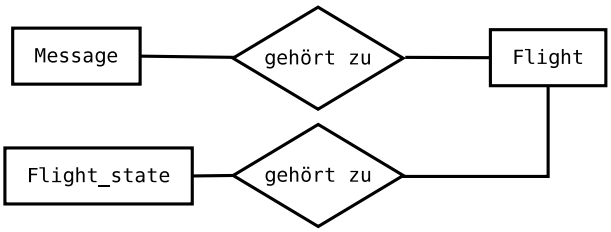


Abbildung 4.3: ER Relation der Datenbank.

Status

Der Statusbildschirm (Abbildung 4.1(d)) bietet grundlegende Statusinformationen zu den zurzeit empfangenen Messwerten. Es gibt Unteransichten, diese wurden nach Schichten unterteilt. Eine Ansicht, um den Verkehr auf dem 1090ES Medium zu betrachten, ist Abbildung \ref{fig_android_sub_modesref{fig_android_sub_modes}t die aktuelle Nachrichtenrate und die Anzahl der empfangenen Nachrichten.

Die andere Ansicht zeigt die zurzeit beobachteten Flüge (Abbildung 4.2(b)). Anhand der ICAO24 wird die Nationalität des Flugzeuges bestimmt und als Flagge angezeigt. Falls der MODE-S Empfang aktiviert ist, wird deren Anzahl an empfangenen MODE-S Paketen angezeigt. Die Detailansicht zu den einzelnen Flugzeugen (Abbildung 4.2(c)) zeigt zusätzlich noch den Zeitpunkt des ersten und letzten empfangenen Paketes an.

4.2 Datenbank

Das Datenbankschema wurde so gewählt, dass die Datenbank möglichst performant beim Einfügen von einzelnen Nachrichten ist. Bei ADS-B können sehr hohe Empfangsraten von über 300 ADS-B Nachrichten pro Sekunde auftreten.

Um die einzelnen Einfüge-Operationen zu beschleunigen, werden sie als Transaktionen von 100 Nachrichten gebündelt. Die Transaktion kann durch Bündelung und Sperrung der Datenbank auf Thread-Ebene, eine erhebliche Beschleunigung erreichen.

In der Tabelle `flights` werden alle beobachteten Flüge abgespeichert.

In Tabelle `flight_state` werden pro Sekunde Statusinformationen (Anzahl an empfangenen ADS-B und Mode-S Nachrichten) zu den aktuellen Flügen gespeichert. Damit lassen sich Nachrichtenraten ausrechnen und es kann überprüft werden, welche der beobachteten Flugzeuge ADS-B Nachrichten aussenden.

In Tabelle `position` werden die aktuellen Positionsdaten des Gerätes gespeichert. Diese Positionsinformationen werden ungefähr jede 1,5 s gespeichert, wobei die Aktualisierungsrate vom benutzten Smartphone und der GPS-Empfangsqualität abhängt.

Zusätzlich gibt es noch eine Tabelle `metadata`. Falls der Aufnahme ein Namen gegeben wurde, wird er dort in der Spalte `name` gespeichert.

Im Anhang ist das vollständige SQL-Schema zu finden (Abbildung 6.2).

4.3 Geschwindigkeitsprobleme

Im vorherigen Kapitel wurden bereits kurz die Performanceprobleme des GNS-5890 Empfängers angesprochen. Hier werden die spezifischen Probleme der Android Anwendung erklärt.

Die Anwendung läuft in 2 Threads: dem UI Thread, welcher alle Benutzerinteraktionen und Aktualisierungen der Oberfläche behandelt, und dem Hintergrund Thread. Letzterer liest die Daten vom USB Bus, parst die empfangenen MODE-S Nachrichten, wertet sie aus und fügt sie, falls es sich um ADS-B Nachrichten handelt, in die Datenbank ein. Die Einfügeoperation wird bereits durch Benutztung von Transaktionen beschleunigt[1], aber es kommt immer noch zu Performancengepässen.

Während der Verarbeitungszeit kann vom USB-Device nicht gelesen werden. Da der GNS-5890 Empfänger nicht genug Nachrichten zwischenspeichert kann es zu Paketverlusten kommen. Dazu trägt auch die verwendete Hardware bei, der Empfänger benutzt als Mikrocontroller einen PIC18F2550[2]. Dieser besitzt 2 kByte RAM und damit keine Ressourcen für einen Nachrichtenbuffer.

[1] Eine Transaktion ist zwar langsamer als das Einfügen einer einzelnen Zeile in die Datenbank. Der Gesamtaufwand wird durch Transaktionen niedriger.

[2] `https://www.microchip.com/wwwproducts/Devices.aspx?dDocName=en010280`

5 Zusammenfassung

Es wurden mehrere übliche ADS-B Empfänger verglichen und deren individuelle Probleme untersucht. Es hat sich herausgestellt, dass der Kinetic SBS-3 Empfänger durchweg eine sehr gute Empfangscharakteristik hat und mehr Nachrichten als die anderen Empfänger empfangen hat. Der GNS-5890 hat auch eine sehr gute Performance und ist für einen Bruchteil des Preises erhältlich.

Die SDR Empfänger bieten eine längst nicht so überzeugende Performance, sind dafür vielseitiger einsetzbar. Teile der Probleme sind auf Softwareprobleme zurückführbar. Mit besseren Decodern ist eine verbesserte Rate zu erwarten. Alle gefundenen Probleme der Implementierungen wurden den Entwicklern mitgeteilt.

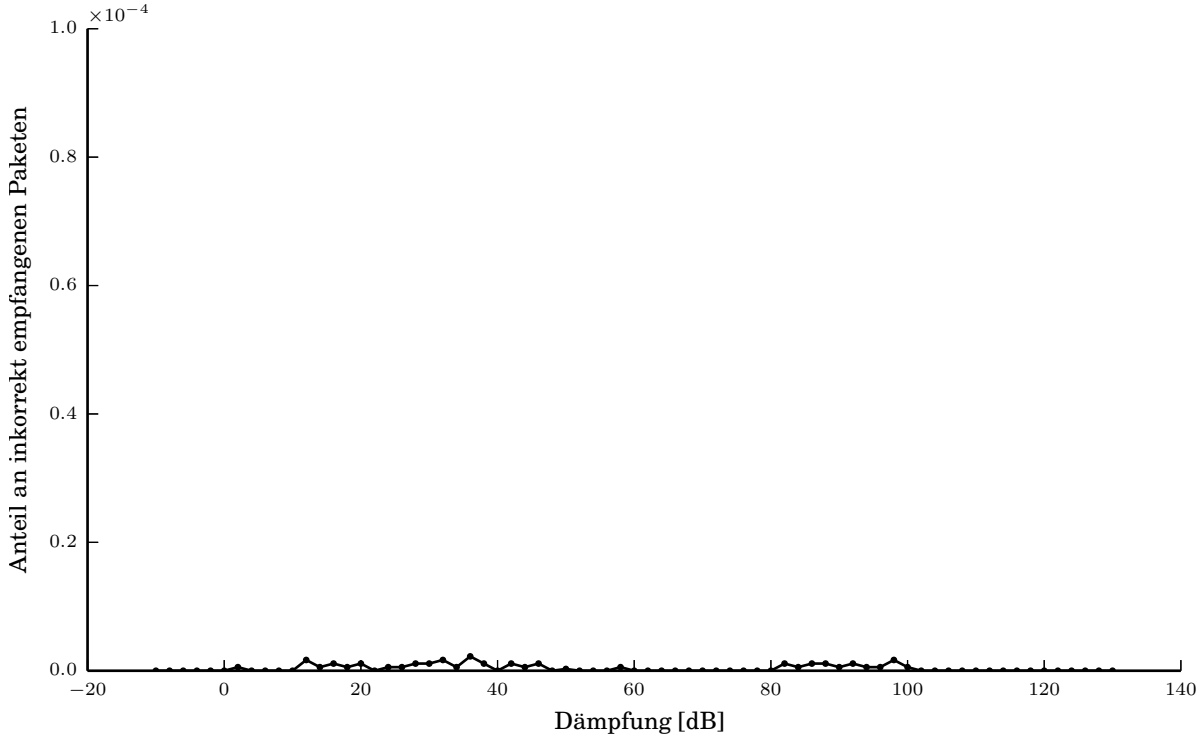
Bei den Experimenten wurde kein realer ADS-B Verkehr gemessen, die Betrachtung wäre interessant. In der Praxis treten oft Nachrichtenkollisionen auf, diese wurden bei den Tests nicht betrachtet, beispielsweise in [12] wurde ein solches System beschrieben und getestet. Es wurde nicht die Genauigkeit der Zeitstempel untersucht, um die Eignung zum Einsatz für Multilateration zu testen. In [5] wurde Multilateration mit dem Kinetic SBS-3 Empfänger durchgeführt, aber die Genauigkeit nur begrenzt betrachtet.

Es wurde ein ADS-B Rekorder für Android entwickelt, der auf Basis des GNS-5890 Empfängers arbeitet. Dieser kann als portables Empfangssystem benutzt werden. Dabei wurden die Limitierungen der Implementierung untersucht.

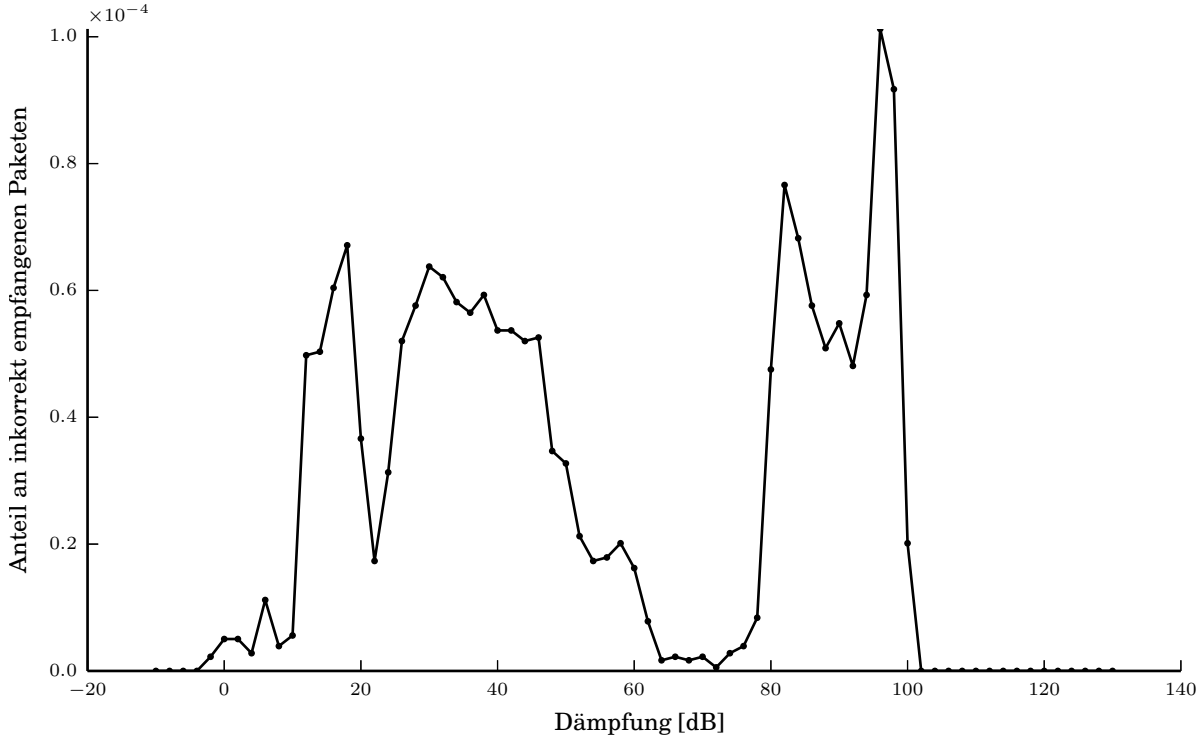
6 Anhang

Eigenschaft	Normal	Aggressiv	Differenz (Aggressiv - Normal)
valid preambles	80489192	79773030	-716162
demodulated again after phase correction	15994416	13808295	-2186121
demodulated with zero errors	41354502	40666714	-687788
with good crc	23523565	23062999	-460566
with bad crc	17830937	17659239	-171698
errors corrected	3788278	5574817	1786539
single bit errors	3788278	3343681	-444597
two bits errors	0	2231136	2231136
total usable messages	27311843	28637816	1325973

Tabelle 6.1: Darstellung der verschiedenen Modi von dump1090. Die Daten wurden mit der `--stats` Option generiert, es wurden alle Aufnahmen aufsummiert.



(a) Fehlerrate im normalen Modus



(b) Fehlerrate im aggressiven Modus.

Abbildung 6.1: Betrachtung der verschiedenen Fehlerraten der dump1090 Software.

```
1  CREATE TABLE IF NOT EXISTS messages(
2           id        INTEGER NOT NULL PRIMARY KEY,
3           flight    INTEGER NOT NULL,
4           timestamp INTEGER,
5           time      INTEGER NOT NULL,
6           icao24    TEXT NOT NULL,
7           format    INTEGER NOT NULL,
8           message   TEXT NOT NULL,
9           checksum  INTEGER NOT NULL,
10          FOREIGN KEY(flight) REFERENCES flights(id)
11          );
12
13 CREATE TABLE IF NOT EXISTS flight_state(
14          id        INTEGER NOT NULL PRIMARY KEY,
15          flight    INTEGER NOT NULL,
16          time      INTEGER NOT NULL,
17          adsb_cnt  INTEGER NOT NULL,
18          smode_cnt INTEGER NOT NULL,
19          FOREIGN KEY(flight) REFERENCES flights(id)
20          );
21
22 CREATE TABLE IF NOT EXISTS flights (
23          id     INTEGER NOT NULL PRIMARY KEY,
24          first  INTEGER NOT NULL,
25          last   INTEGER,
26          icao24 TEXT NOT NULL
27          );
28
29  CREATE TABLE IF NOT EXISTS position(
30          time      INTEGER NOT NULL PRIMARY KEY,
31          latitude  REAL NOT NULL,
32          longitude REAL NOT NULL,
33          altitude  REAL,
34          speed     REAL,
35          direction REAL
36          );
```

Abbildung 6.2: SQLite Schema der Datenbank

Literatur

[1] Donald L Mccallie. „Exploring potential ads-b vulnerabilities in faa's nextgen air transportation system". Magisterarb.

[2] *Wikipedia - Radarstationen (Revision vom 25. 4. 2014 01:46)*. URL: http://de.wikipedia.org/w/index.php?title=Radarstation&oldid=129811235.

[3] Martin Strohmeier, Matthias Schäfer, Vincent Lenders, Ivan Martinovinic und Jens B. Schmitt. „Realities and Challenges of NextGen Air Traffic Management: The Case of ADS-B". In: *IEEE Communications Magazine* (2014), S. 1–7.

[4] *Faa faces significant risks in implementing the automatic dependent surveillance – broadcast program and realizing benefits*. Techn. Ber. Federal Aviation Administration, 2011.

[5] Matthias Schäfer, Martin Strohmeier, Vincent Lenders, Ivan Martinovic und Matthias Wilhelm. „Bringing up OpenSky: A Large-scale ADS-B Sensor Network for Research". In: *ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)* (2014).

[6] *Aeronautical Telecommunications - IV Surveillance and Collision Avoidance Systems*. Fourth Edition. 2007.

[7] „Minimum Operation Performance Standards (MOPS) for 1090 MHz Extended Squitter Automatic Dependent Surveillance - Broadcast (ADS-B) and Traffic Information Services - Broadcast (TIS-B)". In: 1 (2011).

[8] „Universal Serial Bus Class Definitions for Communication Devices". In: (1999).

[9] *ADS-B PIC Receivers Projektseite*. URL: http://sprut.de/electronic/pic/projekte/adsb/adsb.htm.

[10] *RTL-SDR Projektseite*. URL: http://sdr.osmocom.org/trac/wiki/rtl-sdr.

[11] *Android Dashboard Stand 4. Februar 2014*. URL: http://developer.android.com/about/dashboards/.

[12] Jian Chen, Shengli Zhang, Hui Wang und Xiufeng Zhang. „Practicing a Record-and-replay System on USRP". In: *Proceedings of the Second Workshop on Software Radio Implementation Forum*. SRIF '13. Hong Kong, China: ACM, 2013, S. 61–64. ISBN: 978-1-4503-2181-5. DOI: 10.1145/2491246.2491257. URL: http://doi.acm.org/10.1145/2491246.2491257.