

Diplomarbeit

# Interneteinbindung über Mikrowellenrichtfunk

Ausgeführt zum Zwecke der Erlangung des akademischen Grades eines  
Diplom-Ingenieurs unter Leitung von

Arpad L. Scholtz

E389

Institut für Nachrichtentechnik und Hochfrequenztechnik

eingereicht an der Technischen Universität Wien  
Fakultät für Elektrotechnik und Informationstechnik

von

**Christoph DIEMBERGER**

9526244

Gärtnergasse 15/2, 2230 Gänserndorf

Wien, im Juni 2003

# Zusammenfassung

Die folgende Diplomarbeit wurde am Institut für Nachrichtentechnik und Hochfrequenztechnik ausgearbeitet. Die Aufgabe war, auf einer gegebenen Richtfunkstrecke zwischen der TU-Wien und einem Gebäude in Wien Hietzing eine Datenfunkverbindung mit einer Datenrate von 1,2 Mbit/s in Betrieb zu nehmen. Zu diesem Zwecke sollten vier Funkmodems und deren Anbindung an das Betriebssystem Linux entwickelt und aufgebaut werden. Als Endprodukt war eine TCP/IP Internetverbindung über die Datenfunkstrecke gefordert.

Das Gesamtsystem arbeitet im 23 cm Amateurfunkband und verwendet den Datenfunkstandard *Packet Radio*. Nach erfolgreicher Inbetriebnahme der Modems im Labor wurden in einem ersten Funkfeldversuch zwei Antennen am Dach der TU-Wien für weitere Messungen verwendet. Die erzielten Ergebnisse und der Vergleich mit einer Streckendämpfungsmessung an der tatsächlichen Funkstrecke nach Hietzing lieferten schlussendlich eine Schätzung für die Systemreserve von ca. 6 bis 8 dB.

Im abschließenden Versuch zwischen der TU-Wien und dem Endpunkt in Hietzing ergaben sich tageszeitabhängige Paketverluste zwischen 0% und 3%. Diese Ergebnisse lassen auf Störungen durch *Man Made Noise* schließen. Eine Verbesserung der Übertragungsqualität wäre durch eine Erhöhung der Sendeleistung oder durch eine Erhöhung der Antennenstandorte möglich.

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>4</b>
1.1	Vorarbeiten . . . . .	4
1.2	Anforderungen und Ziele . . . . .	5
<b>2</b>	<b>Funkstrecke</b>	<b>6</b>
2.1	Allgemeine Beschreibung . . . . .	6
2.2	Berechnung LOS . . . . .	7
2.3	Charakterisierung NLOS . . . . .	7
<b>3</b>	<b>Systemkonzept</b>	<b>9</b>
3.1	Gesamtsystem . . . . .	9
3.2	Signalbeschreibung . . . . .	10
3.2.1	Datensignale . . . . .	10
3.2.2	Hochfrequenzsignal . . . . .	10
3.3	Protokollstack . . . . .	11
3.4	Schnittstellenkarte . . . . .	12
3.4.1	Blockschaltbild . . . . .	12
3.4.2	Scrambler/Descrambler . . . . .	13
3.4.3	Symboltaktückgewinnung . . . . .	13
<b>4</b>	<b>Funkmodem</b>	<b>15</b>
4.1	Blockschaltbild . . . . .	15
4.2	Fertiges Modem . . . . .	16
4.3	Signaltheoretische Beschreibung . . . . .	17
4.3.1	Sendepfad . . . . .	17
4.3.2	Empfangspfad . . . . .	19
4.4	Schaltungstechnische Beschreibung . . . . .	22
4.4.1	Vorverstärker und AGC . . . . .	22
4.4.2	Costas Loop . . . . .	23
<b>5</b>	<b>PC Software</b>	<b>27</b>
5.1	Systemvoraussetzungen . . . . .	27
5.2	Eingesetzte Softwarekomponenten . . . . .	27
5.3	Konfiguration der Schnittstellenkarte . . . . .	28
5.3.1	Hardware Konfiguration . . . . .	28

<i>INHALTSVERZEICHNIS</i>	3
5.3.2 Inbetriebnahme des Netzwerkinterface . . . . .	29
5.4 Gatewaykonfiguration . . . . .	29
5.5 Weitere Konfigurationsoptionen . . . . .	31
5.5.1 Erweiterung auf eine Verschlüsselungstechnologie . . . . .	31
5.5.2 Erweiterung auf ein Mehrbenutzerszenario . . . . .	32
5.5.3 Repeaterbetrieb . . . . .	33
<b>6 Messungen</b>	<b>34</b>
6.1 Allgemeines . . . . .	34
6.2 Labormessungen . . . . .	34
6.3 Systemtest . . . . .	36
6.4 Messungen an der Richtfunkstrecke . . . . .	38
<b>7 Schlußfolgerungen</b>	<b>43</b>
<b>A Schaltungen und Stücklisten</b>	<b>44</b>
A.1 Stückliste Vorverstärkerplatine . . . . .	44
A.2 Stückliste Costas Loop . . . . .	45
A.3 Vorverstärkerschaltung . . . . .	46
A.4 Costas Loop Analogteil . . . . .	47
A.5 Costas Loop Digitalteil . . . . .	48
<b>B Schaltungstechnische Modifikationen</b>	<b>49</b>
<b>C Anleitung zur Bitfehlermessung</b>	<b>50</b>
<b>D Installation von Linux</b>	<b>52</b>
<b>E Aktualisierung der Linux Installation</b>	<b>54</b>
<b>F Konfigurationsscript</b>	<b>57</b>
<b>G Timingparameter</b>	<b>59</b>
<b>H Konfiguration der Schnittstellenkarte</b>	<b>60</b>
<b>I Abkürzungen</b>	<b>61</b>
<b>Literaturverzeichnis</b>	<b>62</b>

# Kapitel 1

## Einleitung

### 1.1 Vorarbeiten

Am Institut für Nachrichtentechnik an der Technischen Universität Wien haben in den letzten Jahren ermutigende Projekte und Experimente zum Thema Datenfunk stattgefunden. Gleichzeitig hat sich der kommerzielle Markt für Funknetzwerke sehr schnell entwickelt. Auf freigegebenen Frequenzbändern (z.B: ISM Band) existieren seit einigen wenigen Jahren fertige Systeme, die preislich auch für Privatpersonen und KMUs leistbar sind und außerdem direkt von der Stange gekauft werden können. Das hier vorgestellte Datenfunkprojekt unterscheidet sich von diesen Systemen aber in einigen Punkten grundlegend:

- Aufbau mittels handelsüblichen diskreten Bauelementen statt hochintegrierter „System On A Chip“ Lösungen
- Festlegung auf ein standardkonformes System basierend auf Packet Radio AX.25 unter Benützung eines Amateurfunkbandes
- Überbrückung einer durch Hindernisse stark beeinträchtigten Funkstrecke mit relativ geringer Sendeleistung

Das hier vorgestellte Projekt fand seine Anfänge im Jahr 1999 in Form zweier Diplomarbeiten von Walter Kunz [1] und Christian Zacherl [2]. Diese Vorarbeiten versetzten mich in die Lage, fertige Teile der damaligen Hardware zu übernehmen, speziell im Hochfrequenzteil, und an meine Anforderungen anzupassen. Ich werde im Laufe der Arbeit des öfteren auf diese beiden Vorarbeiten Bezug nehmen. Im Signalverarbeitungsteil war ein komplettes Redesign unabdingbar. Weiters hat im Jahr 2001 Werner Keim [3] im Rahmen seiner Diplomarbeit auf derselben Funkstrecke eine Datenübertragung mit 56 kbit/s implementiert. Nach diesem Erfolg reifte am Institut die Idee, diese Strecke auf eine höhere Datenrate zu erweitern.

## 1.2 Anforderungen und Ziele

Bezug nehmend auf die Vorarbeiten soll auf der bereits bekannten Strecke zwischen der TU-Wien und einem Hochhaus in Wien Hietzing (Luftlinie ca. 6.9 km) eine 1,2 Mbit/s Richtfunkverbindung aufgebaut werden. Dafür ist es notwendig, auch in Hinblick auf weiterführende Experimente und Projekte, vier identische Funkmodems zu konstruieren und in Betrieb zu nehmen. Zur Anbindung an den Computer existieren Schnittstellenkarten [9] und dazu passende Treiber für das Betriebssystem Linux [6].

Die Richtfunkstrecke soll im 23 cm Band betrieben werden, wobei bei der Auswahl der Sendefrequenz die bereits existierende Bandauslastung berücksichtigt werden muss.

Ein inhärentes Design Kriterium ist die Mehrbenutzerfähigkeit. Eine Erweiterung der fertigen Funkstrecke auf mehr als zwei Teilnehmer muß bei der Adressierung und beim Kanalzugriff vorgesehen sein.

Als Basis für weitere Experimente und für den anschaulichen Betrieb ist eine Interneteinbindung über TCP/IP gefordert.

Das 56kbit/s Projekt [3] dient als Maßstab für die Leistungsfähigkeit in Hinblick auf Datendurchsatz und Stabilität. Im Rahmen dieses Projektes wurde die Funkstrecke genau vermessen. Außerdem können die bereits errichteten, sehr leistungsfähigen Antennen für das neue Projekt verwendet werden. Dadurch ist ein direkter Vergleich der beiden Systeme gegeben.

Zusätzlich zu den in der Einleitung geforderten Eigenschaften der Funkmodems sei noch erwähnt, daß auf einen möglichst geringen Abgleichaufwand Wert gelegt wird. Weiters sollen die Hochfrequenzkomponenten komplett geschirmt sein, da angenommen werden muss, daß das Modem in der Nähe von anderen Amateurfunkgeräten in Dauerbetrieb gehen wird.

# Kapitel 2

## Funkstrecke

### 2.1 Allgemeine Beschreibung

Richtfunkstrecken sind aus der modernen Nachrichtentechnik nicht mehr wegzudenken. Auch hier haben digitale Modulationsverfahren analoge Systeme weitgehend abgelöst. Die Ausbreitungsverluste elektromagnetischer Wellen steigen mit dem Quadrat der Entfernung wobei wir die Absorptionsverluste der Erdatmosphäre bei den von uns betrachteten Strecken und Frequenzen vernachlässigen können. Bei einer klassischen Richtfunkverbindung bemüht man sich, eine optische Sichtverbindung zwischen den beiden Stationen sicher zu stellen. Elektromagnetische Wellen hoher Frequenzen, z.B. Mikrowellen, verhalten sich *quasioptisch*. Reflexionen und Abschattungseffekte, die aufgrund der geringen Wellenlänge von 23 cm auch schon bei relativ kleinen Hindernissen starken Einfluss haben, spielen eine maßgebende Rolle.

Auf Richtfunkstrecken versucht man durch Bündelung der Strahlungsleistung mittels eines möglichst geringen Öffnungswinkels ein Maximum an abgestrahlter Energie in Richtung des Empfängers zu senden. Die Richtwirkung der Antennen, der sogenannte Gewinn, ist demnach ein wesentliches Kriterium.

Bei der hier vorliegenden Übertragungsstrecke handelt es sich um eine *Non Line of Sight* - Verbindung, kurz *NLOS*. Es existieren zwei massive Hindernisse, zwei Hochhäuser, die direkt in die Sichtlinie der Strecke hineinragen und dadurch die Übertragungseigenschaften des Kanals stark beeinflussen. Zusätzlich ragen mehrere Einzelobjekte in die sogenannte 1. Fresnelzone. Innerhalb dieses Ellipsoides kommt es zu Interferenzen zwischen verschiedenen Ausbreitungs-Pfaden, die ihrerseits durch Reflexionen entstehen. Die Gesamtentfernung beträgt 6,9 km. Eine Skizze zeigt Abbildung 2.1

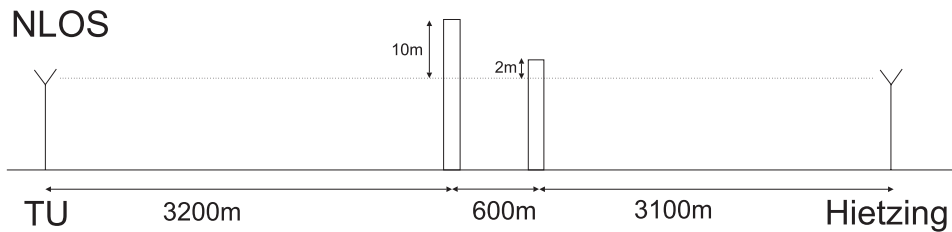


Abbildung 2.1: Funkstrecke von der TU-Wien nach Hietzing

## 2.2 Berechnung LOS

Zuerst wird eine Berechnung unter Annahme *Line Of Sight* durchgeführt und dann mit den Ergebnissen von [3] verglichen.

Die Freiraumdämpfung, ausgedrückt in Dezibel ergibt sich zu:

$$L_{LOS} = 10 \lg \left( \frac{4\pi d}{\lambda} \right)^2. \quad (2.1)$$

Durch Einsetzen der Wellenlänge von 23 cm und der Entfernung von 6,9 km

$$L_{LOS} = 10 \lg \left( \frac{4\pi 6900m}{0,23m} \right)^2 \quad (2.2)$$

erhalten wir demnach eine Streckendämpfung von  $L_{LOS} = 111,5dB$ . Die zusätzlichen Verluste, die durch Beugung an den Hindernissen entstehen, wurden in [3] abgeschätzt und anschließend gemessen.

## 2.3 Charakterisierung NLOS

Zur Charakterisierung der tatsächlichen NLOS Strecke setzen wir:

$$L_{NLOS} = L_{LOS} + L_{Beugung}. \quad (2.3)$$

Zu diesem Zweck wurde eine Messung mit einem Sinussignal durchgeführt. Diese Messung ist für das Gesamtsystem schon deshalb recht aussagekräftig, weil Antennen und Antennenzuleitungen der tatsächlichen Übertragungsstrecke gleich mitvermessen wurden. Das Sendesignal wurde durch ein Handfunkgerät (Ausgangsleistung 1 Watt) erzeugt und auf der anderen Seite der Strecke mit einem Spektralanalysator dargestellt und gemessen. Mithilfe des Empfangspegels konnte ein Pegelplan für die angepeilte Modemausgangsleistung von 27 dBm erstellt werden.



Sendeleistung TU-Wien:	+	27,0	dBm
Antennenzuleitung TX:	-	6,0	dB
Antenne TX:	+	15,0	dB
Streckendämpfung:	-	136,0	dB
Antennengruppe RX:	+	21,0	dB
Antennenzuleitung RX:	-	2,0	dB
<hr/>			
Empfangspegel Hietzing:	-	81,0	dBm

Anhand des Pegelplanes lassen sich nun die zusätzlichen Beugungsverluste sofort ablesen, und zwar mit  $L_{\text{Beugung}} = 24,5\text{dB}$ . Eine einfache Berechnung anhand von Gleichung 6.1 zeigt, daß diese NLOS Funkstrecke einer equivalenten LOS Funkstrecke über eine Distanz von 114 km entspricht.

# Kapitel 3

## Systemkonzept

### 3.1 Gesamtsystem

Beide Stationen bestehen aus identischen Komponenten. Nach Überprüfung der Belegung des 23 cm Amateurfunkbandes anhand von Tabellen wurde als Betriebsfrequenz 1276 MHz und als Betriebsart Semi-Duplex gewählt. In dieser Betriebsart ist immer nur eine Funkstelle im Sendebetriebe. Ein *Data Carrier Detect Signal*, kurz DCD, zeigt an, ob der Kanal frei ist und die Station in den Sendebetriebe wechseln kann. Kollisionen am Funkkanal werden somit weitgehend vermieden. Falls Pakete gestört werden, muß der Protokollstack dafür sorgen, daß die Übertragung wiederholt wird. Eine Datensicherung in Form einer FEC (*Forward Error Correction*) ist nicht vorgesehen, die einzelnen Pakete enthalten aber eine Prüfsumme, wodurch Übertragungsfehler erkannt werden können.

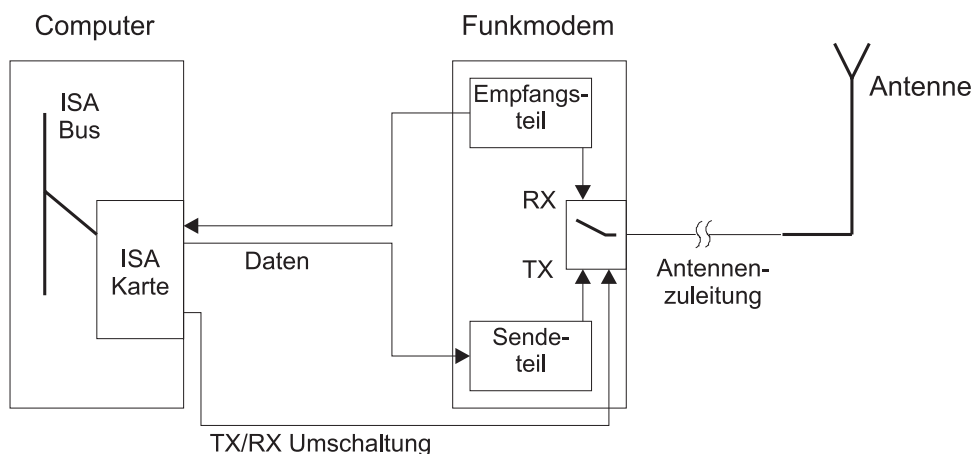


Abbildung 3.1: Systemkonzept einer kompletten Funkstation

## 3.2 Signalbeschreibung

### 3.2.1 Datensignale

Die drei Datensignale zwischen Modem und Computer in Abbildung 3.1 liegen auf TTL Signalpegel und bestehen aus Sendedaten TxD, Empfangsdaten RxD und einem Steuersignal zur Sende/Empfangsumschaltung /PTT. Die Signale RxD und TxD sind *Non return to zero inverted*, kurz NRZI, kodiert. Bei dieser Kodierung steht eine logische „1“ für einen Zustandswechsel und eine logische „0“ für die Beibehaltung des momentanen Zustandes (siehe Abbildung 3.2). Das PTT (*Push To Talk*) Signal ist Low Active, d.h. eine logische Null steht für den Sendemodus.

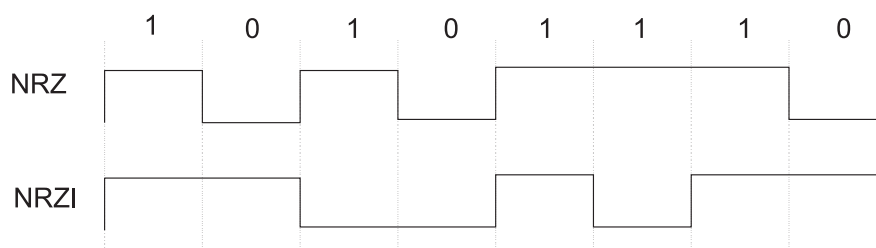


Abbildung 3.2: Vergleich NRZ und NRZI Kodierung

Das TxD Signal besteht aus einem synchronen 1,2 Mbit/s Bitstrom, der von der Schnittstellenkarte erzeugt wird. Auf der Schnittstellenkarte befindet sich ein 18-Stage Bitscrambler (siehe Abschnitt 3.4). Die Sendesymbolstatistik ist demnach praktisch gleichverteilt.

Das RxD Signal ist ebenfalls ein synchroner Bitstrom gleicher Bitrate. Der Symboltakt muß auf der Schnittstellenkarte mithilfe einer digitalen PLL regeneriert werden. ( siehe Abschnitt 3.4 ).

Das Zuleitungskabel ist ein geschirmtes Flachbandkabel.

### 3.2.2 Hochfrequenzsignal

Die angepaßte Antennenspeiseleitung ist durch ein H2000 Koaxialkabel mit Bezugswiderstand 50 Ohm realisiert. Das Kabel hat bei der Trägerfrequenz laut Spezifikation eine Dämpfung von 0,16 dB pro Meter.

Als Modulationsverfahren kommt BPSK (*Binary Phase Shift Keying*) zum Einsatz. Dieses Modulationsverfahren ist einfach zu erzeugen und zeichnet sich auch bei höheren Bitraten durch seine Störsicherheit aus. Die Signalpunktconstellation

ist in Abbildung 3.3 dargestellt.

Dazu ist zu sagen, daß das Funksignal im Empfänger inkohärent demoduliert wird. Die Signalpunktconstellation rotiert demnach mit der Differenzfrequenz der beiden Oszillatoren im Sender und im Empfänger (Siehe Abschnitt 4.3).

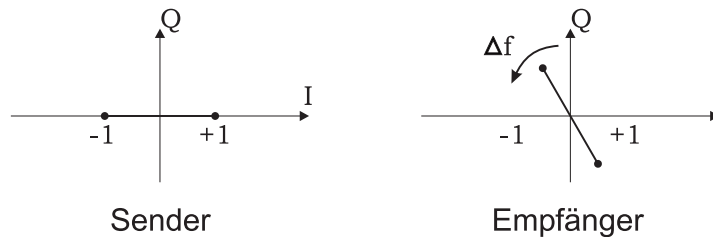


Abbildung 3.3: Signalpunktconstellation

### 3.3 Protokollstack

Wie in der Einleitung erwähnt, basiert das System auf dem Packet Radio Standard. Packet Radio ist eine Amateurfunkbetriebsart für digitale Datenkommunikation. Als Protokoll wird AX.25 eingesetzt. AX.25 ist vereinfacht gesagt ein für den Amateurfunk modifiziertes X.25 Protokoll. Konkret wurde im Protokoll-Header das Adressfeld dermaßen erweitert, sodaß es die bis zu 7 Byte langen Amateurfunkrufzeichen aufnehmen kann. AX.25 ist primär ein verbindungsorientiertes Protokoll, unterstützt aber auch einen Datagram Modus, bei dem keine Paketbestätigungen stattfinden, und die einzelnen Pakete unnummeriert gesendet werden (*Unnumbered Information Frames*). Dieser Modus ist bestens geeignet zum Transport von TCP/IP Paketen, wobei die Flußkontrolle dann komplett von TCP übernommen wird. Einige weitere wichtige Eigenschaften von AX.25 sind folgende:

- flexible Paketgröße
- Source Routing durch Angabe einer Empfängeradressfolge im Header möglich
- mehrbenutzertauglich
- 16 Bit FCS (*Frame Check Sequence*) in jedem Paket
- eindeutiges 8 Bit Start/End Flag der Form „01111110“

Für den Betrieb wurde eine MTU (*Maximum Transfer Unit*) von 1500 Byte für das Protokoll IP definiert. Dieser Wert erwies sich in weiterer Folge als guter

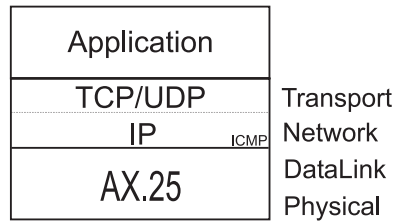


Abbildung 3.4: Protokollstack

Kompromiß der Paketgröße zwischen Datendurchsatz und Antwortzeit.

Dem Amateurfunk wurde von der RIPE - Das ist die Organisation, die für die IP Adressvergabe zuständig ist - ein eigenes Klasse A IP Netz zugewiesen, und zwar der Adressbereich 44.0.0.0/8. Österreichische Amateurfunke benutzen das Klasse B Netz 44.143.0.0/16. Zuständig für die Vergabe ist OE1KDA Krzysztof Dabrowski.

### 3.4 Schnittstellenkarte

#### 3.4.1 Blockschaltbild

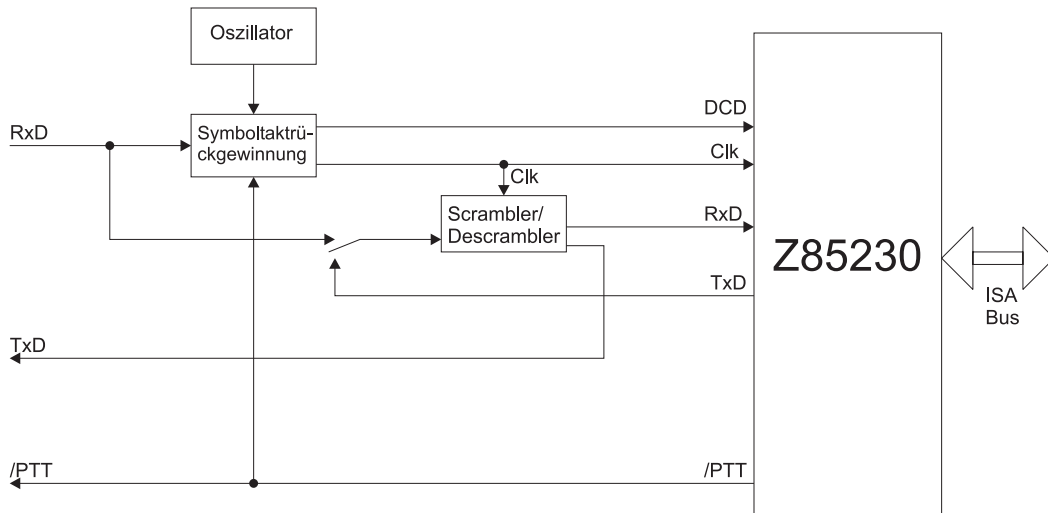


Abbildung 3.5: Blockschaltbild Schnittstellenkarte

Die Anbindung an den ISA Bus erfolgt über einen DMA (*Direct Memory Ac-*

cess) Kanal. Damit sind hohe Übertragungsraten bis zu einigen Mbit/s direkt in den Hauptspeicher ohne zutun der CPU möglich. Herzstück der Karte ist ein sogenannter Serial Communications Controller, und zwar ein Z85230 von Zilog. Dieser Baustein erledigt einige Aufgaben, die sonst der Software-Treiber erledigen müßte, wie CRC Berechnung und Generierung des Start/End Flags. Der Baustein besitzt zwei Datenkanäle - Kanal A und B. Bei diesem Projekt wird nur Kanal A verwendet der über den Scrambler an die 9-polige Sub-D Buchse *K3* geführt wird.

### 3.4.2 Scrambler/Descrambler

Datenpakete die vom AX.25 Protokollstack generiert werden, werden auf Bitebene verwürfelt. Das ist notwendig, weil das Gesamtsystem nur dann fehlerfrei arbeitet, wenn innerhalb des Bitstroms die logische „0“ und die logische „1“ gleich wahrscheinlich sind. Der Scrambler besteht aus einem 18-stufigem Schieberegister, das zu einem Infinite Impuls Response Filter zusammenschaltet ist. Dessen Z-transformierte Impulsantwort kann folgendermaßen geschrieben werden:

$$H_{Scrambler}(z) = z^{-17} \frac{1}{1 \oplus z^{-12} \oplus z^{-17}}. \quad (3.1)$$

Der Operator  $\oplus$  steht für die *modulo2* Operation. Der Faktor  $z^{-17}$  im Zähler bewirkt eine Gesamtverzögerung von 17 Taktimpulsen. Die Impulsantwort dieses Systems ist eine Pseudozufallsfolge maximaler Länge mit der Periode  $2^{17} - 1 = 131071$  Taktzyklen.

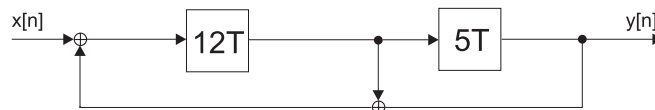


Abbildung 3.6: Scrambler

Die Scramblerschaltung wird im Empfangsfall mit Hilfe von Logikgattern so umkonfiguriert, daß ein Transversalfilter mit der inversen Impuls Antwort entsteht, allerdings ohne die Verzögerung  $z^{-17}$ .

$$H_{Descrambler}(z) = \frac{1}{H_{Scrambler}(z)} = 1 \oplus z^{-12} \oplus z^{-17} \quad (3.2)$$

### 3.4.3 Symboltaktrückgewinnung

Die Daten werden vom Modem in einem synchronen NRZI Kodierten Bitstrom geliefert. Dadurch ist im Empfänger eine direkte Zuordnung zwischen logischer

„0“ und logischer „1“ zwar möglich, ohne Regeneration des Symboltaktes können aufeinanderfolgende Symbole aber nicht korrekt getrennt werden. Die Schaltung für die Symboltaktrückgewinnung besteht im Wesentlichen aus einer digitalen Phasenregelschleife, deren Funktion bereits bei [1] sehr ausführlich dargestellt wurde. Sowohl die Empfangsdaten als auch das regenerierte Taktsignal haben nach der Regelschleife einen konstanten Phasenjitter von +/- 50 ns. Während der Sendephase arbeitet die Schaltung als 8-fach Frequenzteiler und liefert demnach bei dem eingesetzten 9,8304 MHz Quarzoszillator einen stabilen Sendetakt von  $9,8304/8 = 1,2288 MHz$ . Wenn die PLL der Symboltaktrückgewinnung eingerastet ist, wird das Signal DCD aktiv und dient als Indikator, daß der Kanal belegt ist und Daten empfangen werden. Der Schnittstellenkartentreiber sorgt dafür, daß bei aktiviertem DCD Signal das Modem nicht in den Sendebetrieb wechselt.

# Kapitel 4

## Funkmodem

Im nun folgenden Kapitel wird auf die einzelnen Baugruppen des Funkmodems näher eingegangen. Dabei beschränkt sich die Beschreibung im Wesentlichen auf den Demodulator und den vorgeschalteten Verstärker mit integrierter AGC (*Automatic Gain Control*). Diese Teile mußten einem kompletten Redesign unterworfen werden. Der HF Teil wurde bis auf einige wenige Modifikationen (angeführt in Anhang B) von [1] übernommen und ist in dessen schriftlicher Ausarbeitung sehr anschaulich erklärt.

### 4.1 Blockschaltbild

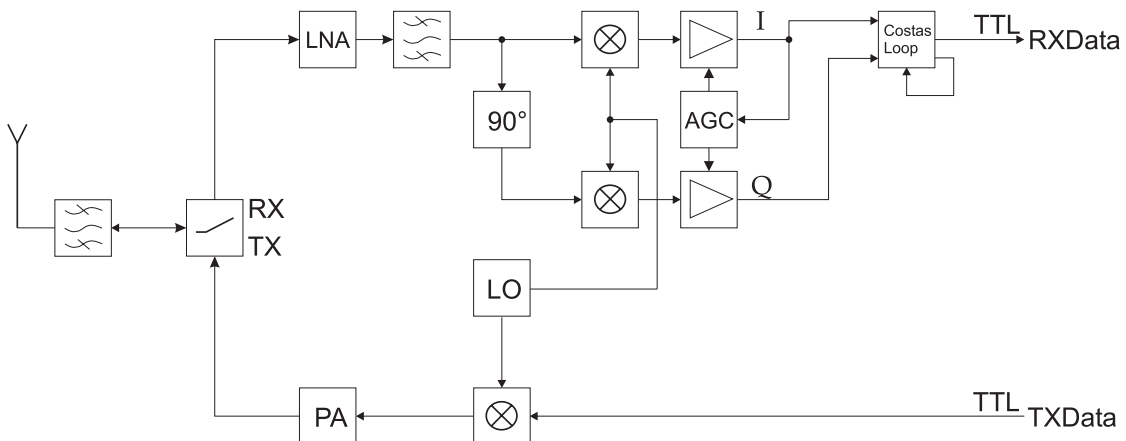


Abbildung 4.1: Funkmodem Blockschaltbild

In weiterer Folge werden nun die Pfade der Sende- und der Empfangsdaten und danach die schaltungstechnische Lösung des Signalverarbeitungsteils genauer betrachtet.



## 4.2 Fertiges Modem

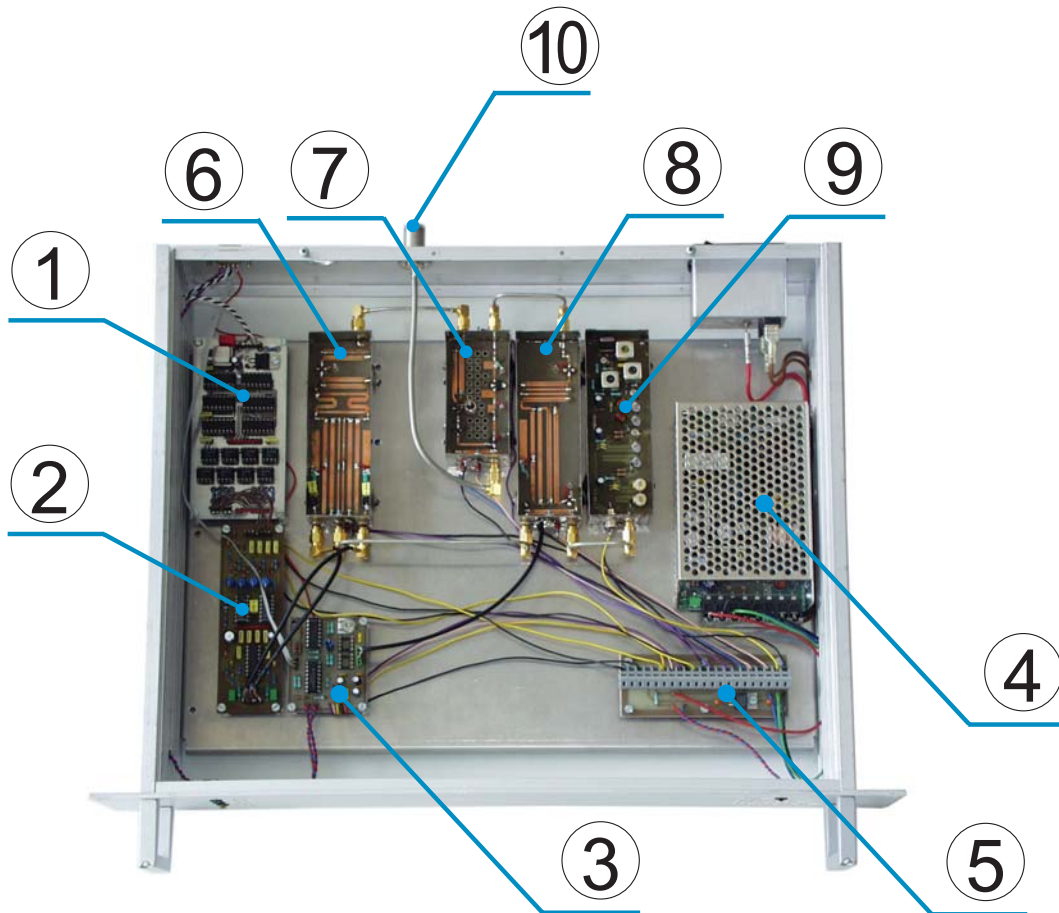


Abbildung 4.2: Funkmodem

- 1 Demodulatorplatine (Costas Loop)
- 2 Vorverstärker und AGC
- 3 Steuerplatine für LEDs und Sende/Empfangsumschaltung
- 4 Netzteil
- 5 Spannungsverteilungsplatine
- 6 I/Q Mischer
- 7 HF Verstärker
- 8 Sendemischer
- 9 Oszillator
- 10 Antennenbuchse

## 4.3 Signaltheoretische Beschreibung

### 4.3.1 Sendepfad

Die Sendedaten, die vom Computer geliefert werden, liegen auf einem Pegel von +5 Volt für die logische „1“, und 0 Volt für die logische „0“. Der daraus resultierende Gleichspannungsanteil wird durch einen Koppelkondensator entfernt. Über ein Trimpotentiometer kann der tatsächliche Pegel, der in den Modulator geleitet wird, modifiziert werden. An dieser Stelle befindet sich auch noch eine Drossel der Größe  $1\mu H$ , mit der man die hochfrequenten Anteile der Rechteckschwingung unterdrückt. In Abbildung 4.3 ist eine NRZI kodierte „1111“ Folge dargestellt. Diese Bitkombination stellt gleichzeitig die höchste mögliche Signalfrequenz dar. Die Bitdauer (= Symboltakt  $T_s$ ) ergibt sich zu  $1/1,2288MHz = 817ns$ .

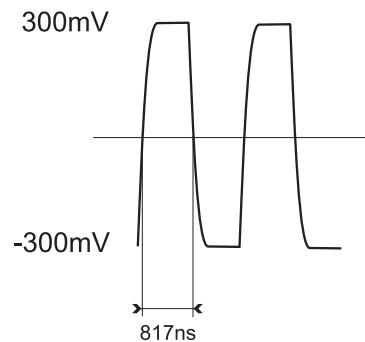


Abbildung 4.3: Datensignal am Eingang des Sendemischers

Das Sendesignal  $g(t)$  kann durch ein Rechtecksignal  $rect(t - \frac{T_s}{2}; \frac{T_s}{2})$  modelliert werden, das über ein Tiefpaßfilter mit der Grenzfrequenz  $\omega_g = 11,5 \cdot 10^6 s^{-1}$  gefiltert wird. Im Frequenzbereich führt das auf einen Sendepuls von

$$G(j\omega) = sinc(\omega T_s) \frac{1}{1 + j \frac{\omega}{\omega_g}}. \quad (4.1)$$

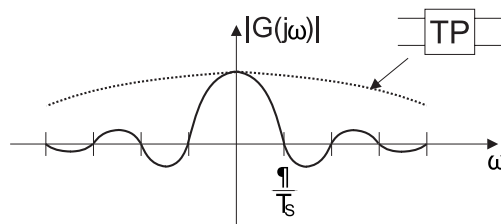


Abbildung 4.4: Datensignal am Eingang des Sendemischers (Frequenzbereich)

Der Tiefpaß führt zu einer Absenkung der Nebenminima, und damit zu einer Reduktion der tatsächlich benötigten Übertragungsbandbreite. Wie aus Abbildung 4.4 ersichtlich ist, hat der Tiefpaß auf die Form des Signalspektrums wenig Einfluß. Eine exakte Berechnung des Sendespektrums anhand der hier aufgestellten Gleichungen wird im Kapitel 6 durchgeführt.

Die Sendesymbole sind mittelwertfrei und gleichverteilt:  $E(A[k]) = 0$ ,  $p_A(A[k]) = 1/2$ . Die mittlere Leistung der Symbole ergibt sich zu:  $E(A^2[k]) = 1$ . Der Operator  $E()$  steht für den Erwartungswert. Daraus kann man das Leistungsdichtespektrum des Signales im Basisband  $S_{LP}(j\omega)$  sofort anschreiben:

$$S_{LP}(j\omega) = \frac{1}{T_s} |G(j\omega)|^2. \quad (4.2)$$

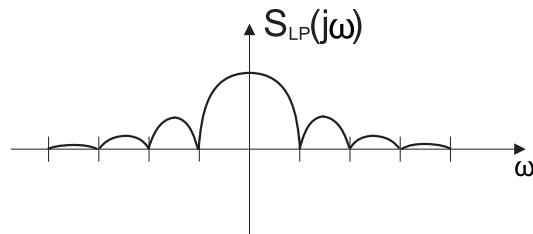


Abbildung 4.5: Leistungsdichtespektrum am Eingang des Sendemischers

Der Modulator arbeitet als sub-harmonischer Mischer. Das eingespeiste Oszillator Signal ist eine Sinusschwingung auf der halben Trägerfrequenz  $f_{LO} = 638\text{MHz}$ . An einem Diodenpaar mit nichtlinearer Kennlinie wird das Datensignal mit dem Oszillatorsignal gemischt. Danach wird das gewünscht Ausgangssignal  $f_{HF} = 2f_{LO} + f_{ZF}$  über einen Bandpaß ausgekoppelt. Die Zwischenfrequenz  $f_{ZF}$  ist in unserem Fall 0. Das Basisbandsignal wird daher auf die doppelte LO-Frequenz „aufmoduliert“.

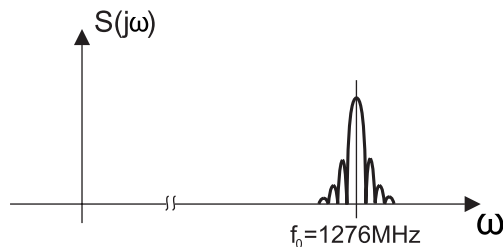


Abbildung 4.6: Leistungsdichtespektrum des BPSK Hochfrequenzsignals

Die minimale Bandbreite von BPSK wäre laut Theorie gleich groß wie die Symbolrate  $B_{min} = 1/T_s$ . Da aber unser Sendeimpuls bezüglich seiner Bandbegrenzung nicht optimal ist, rechnen wir mit einer geringfügig höheren effektiven Bandbreite, die später auch gemessen wird. Für zukünftige Erweiterungen könnte man durch den Einsatz eines Sendefilters die Bandbreite des Systems reduzieren oder durch Einsatz einer höherwertigen Modulation bei gleichbleibender Bandbreite die Datenrate erhöhen.

### 4.3.2 Empfangspfad

Der Empfangspfad arbeitet nach dem Prinzip der *direct conversion*. Nach der Vorfilterung und einem möglichst rauscharmen Hochfrequenzverstärker (LNA = Low Noise Amplifier) wird das Empfangssignal dem Empfangsmischer zugeführt. Dieser besteht aus zwei voneinander unabhängigen Mischstufen. Durch den Einsatz eines  $90^\circ$  Hybrid wird erreicht, daß man mit nur einem Oszillatorsignal Inphase- und Quadraturkomponente unabhängig voneinander aus dem Hochfrequenzsignal gewinnen kann. Der Mischvorgang ist inkohärent, da weder eine fixe Phasen- noch eine fixe Frequenzbeziehung zwischen Sender und Empfänger besteht. Das führt zu der bereits in Abbildung 3.3 dargestellten Eigenschaft, daß die Signalpunkt-konstellation mit der Frequenz  $\Delta f$  rotiert.  $\Delta f$  ist die momentane Differenzfrequenz zwischen den beiden Oszillatoren im Sende- und im Empfangsmodem. Diese Rotation läßt sich nun auch direkt in der I/Q Ebene abbilden:

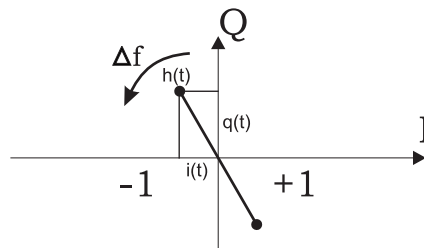


Abbildung 4.7: Darstellung der beiden Signalkomponenten

Durch Projektion auf I und Q Achse ergibt sich:

$$i(t) = h(t) \cos 2\pi\Delta ft \quad (4.3)$$

$$q(t) = h(t) \sin 2\pi\Delta ft \quad (4.4)$$

Das Verhältnis der beiden Komponenten  $i(t)$  und  $q(t)$  liefert direkt den Tangens des momentanen Phasenfehlers zwischen Sende- und Empfangsoszillator  $\phi$ . Um nun eine Demodulation mit maximalem Störabstand zu erreichen, genügt es nicht, beide Signalkomponenten ohne weitere Maßnahmen einem Entscheider zuzuführen. Im ungünstigsten Fall einer um  $45^\circ$  gedrehten Signalpunkt-konstellation hätten beide Signale nur 70 Prozent ihres Pegels, und dadurch einen Verlust im

Rausch-Störabstand von 3 dB.

Zur Lösung dieses Problems bietet sich eine sogenannte „Costas Loop“ an. Die Idee dahinter, dargestellt für die Signalkomponente  $i(t)$ , ist der Abbildung 4.8 zu entnehmen.

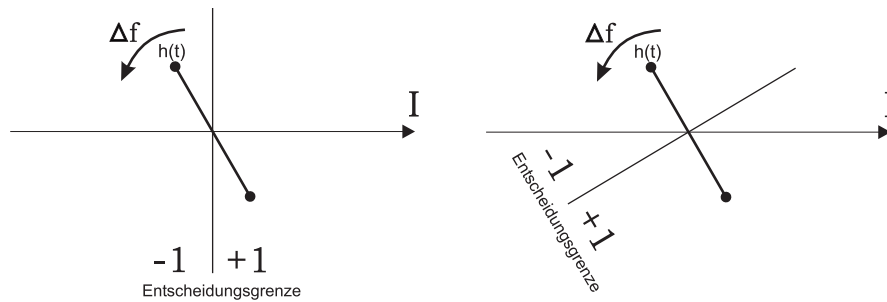


Abbildung 4.8: Entscheidungsgrenze rotiert mit Signalkonstellation

Die Entscheidungsgrenze trennt die beiden Halbräume in denen auf das Symbol +1 oder -1 entschieden wird. Falls es gelingt, die Entscheidungsgrenze mit derselben Frequenz  $\Delta f$  mitrotieren zu lassen, dann steht die Entscheidungsgrenze immer normal zum Symbolwechsel, und dadurch liegt am Eingang des Entscheiders immer der maximale Signalpegel an.

Die in diesem Projekt eingesetzte „Costas Loop“ besteht aus einem Widerstandsnetzwerk, mehreren Komparatoren und einer digitalen Regelschleife. Die Rotation der Entscheidungsgrenze wird so gelöst, indem das Inphase-Signal über ein Widerstandsnetzwerk an 16 Entscheidungskomparatoren herangeführt wird. Jeder Entscheidungskomparator stellt einen konstanten Drehwinkel in der Signalkonstellation dar. Die eigentliche Rotation wird erst danach durch eine Logikschaltung bewerkstelligt, indem aus den 16 Ausgangssignalen der Komparatoren immer nur eines als eigentliches Datensignal ausgewählt wird. Die Schaltungstechnische Realisierung wird in Abschnitt 4.4 behandelt.

Wenn die Regelschleife eingerastet ist, liegt die aktive Entscheidungsgrenze maximal  $\Delta\phi = 22,5^\circ$  neben dem Optimum. Das ist also der Winkel, in dem die Schleife gerade noch nicht in den nächsten Bereich umschaltet. Daraus resultiert ein SNR-Verlust von

$$20 \lg \frac{1}{\cos 22,5^\circ} = 0,69 \text{ dB}. \quad (4.5)$$

Auch der Bitscrambler kann die Leistungsfähigkeit des Systems negativ beeinflussen. Wenn ein einzelner Symbolfehler (=Bitfehler) auf der Strecke auftritt, so bildet der Descrambler diesen auf drei Bitfehler ab. Der Fehler kommt also mit

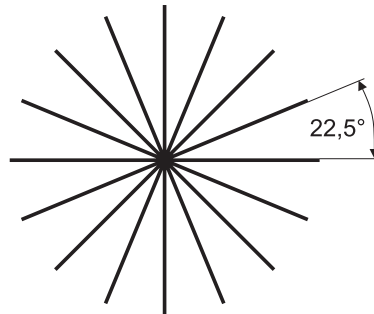


Abbildung 4.9: 16 fixe Entscheidungsgrenzen

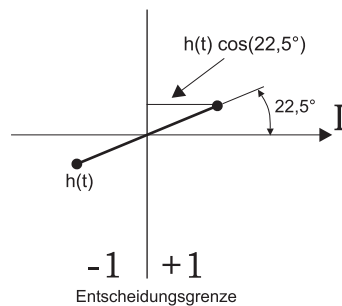


Abbildung 4.10: Maximaler Pegelverlust

einem „Echo“ im Abstand von 12 Taktzyklen und noch einmal im Abstand von 17 Taktzyklen (Vergleiche Abbildung 3.6). Falls innerhalb der Scrambler-Länge (in diesem Fall 17) ein weiterer Bitfehler auftritt, so kann dieser auf vier oder sogar auf sechs Bitfehler abgebildet werden, abhängig vom Abstand zum ersten Fehler. Die Entscheidung, ob ein Paket verworfen wird oder nicht ist aber unabhängig von der exakten Anzahl der Symbolfehler. Bereits bei einem falschen Symbol stimmt die Prüfsumme nicht mehr mit dem Paketinhalt überein, und der AX.25 Protokollstack verwirft das Paket.

Weiters stellt sich die Frage, welche maximale Differenzfrequenz  $\Delta f$  die Regelschleife ausgleichen kann. Aus dem Schaltbild in Anhang A läßt sich ablesen, daß zwei kaskadierte 4 Bit Zähler als Frequenzteiler eingesetzt werden. Die Taktfrequenz wird von dem Oszillator erzeugt und beträgt 14,3 MHz. Der Fangbereich der „Costas Loop“ ist daher:

$$\frac{14,3MHz}{2^8} = \Delta f_{max} = \pm 55,9kHz \quad (4.6)$$

## 4.4 Schaltungstechnische Beschreibung

### 4.4.1 Vorverstärker und AGC

Diese Schaltung (siehe Anhang A) besteht im Wesentlichen aus drei Komponenten: Eingangsfilter, Verstärkerbaustein und einer Regelschleife für die Automatic Gain Control. Die Abwärtsmischung liefert Inphase- und Quadraturkomponente als zwei getrennte Signale über eine kurze, geschirmte Verbindung. Beide Signale werden komplett getrennt voneinander durch die Verstärkerschaltung geführt. Die Regelspannung für die AGC Regelung wird aus der Kombination der beiden Signale abgeleitet. Die beiden  $1k\Omega$  Widerstände am Eingang der Schaltung versorgen den Vorverstärker im Abwärtsmischer.

Das Eingangsfilter ist ein LC-Tiefpaß 2. Ordnung und in Abbildung 4.11 isoliert dargestellt. Die 3 dB Grenzfrequenz liegt bei  $f_g = 750kHz$ . Bei der Messung hat sich gezeigt, daß aufgrund der Bauteilverluste kaum eine Resonanzüberhöhung auftritt. Die Annahme einer konstanten Signalamplitude ist daher gerechtfertigt.

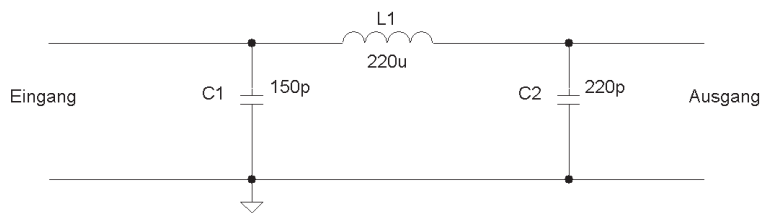


Abbildung 4.11: Eingangsfilter

Inphase- und Quadraturkomponente werden nach dem Eingangsfilter getrennt voneinander verstärkt. Der Verstärkerbaustein NE592 besitzt zwei differentielle Ausgänge und würde normalerweise eine symmetrische Versorgungsspannung benötigen. Durch das vorgeschaltete Widerstandsnetzwerk wird zu den Eingangssignalen eine Gleichspannung von 5,5 Volt addiert. Nun wird der Verstärker mit +12 Volt versorgt und kann daher zwischen ca. 0,5 Volt und 11,5 Volt aussteuern. Der Gleichspannungsanteil wird anschließend über Koppelkondensatoren entfernt. Der NE592 Verstärker zeichnet sich durch seine rauscharmut und seine hohe Bandbreite von 40 MHz aus.

Die Regelschleife besteht aus einem Stellglied (CMOS Inverter Gatter CD4049), einem Schleifenfilter (Abbildung 4.12) und der Eingangsgröße. Als Eingangsgröße werden über ein Diodengleichrichternetzwerk die I und Q Signalkomponenten zugeführt. Das Schleifenfilter fügt eine Zeitkonstante hinzu, die einerseits auf Pegelsprünge, z.B. am Anfang eines Datenpakets, schnell reagieren muß, andererseits aber die Schwankungen der gleichgerichteten Datensignale glättet. Aus diesen Anforderungen ergibt sich ein RC-Tiefpaß mit einer Grenzfrequenz von

6,6 kHz. Als Stellglied dient der PMOS Transistor eines CMOS Inverter Gatters, der bei steigender Spannung am Gate-Kontakt seinen Drain-Source Widerstand verringert, und so die Signalspannung reduziert.

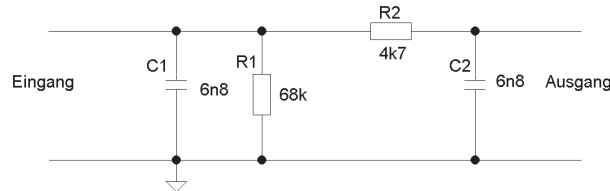


Abbildung 4.12: AGC Schleifenfilter

Die Schaltung verfügt weiters über einen Monitor Ausgang, mit der man die Regelspannung der AGC hochohmig abgreifen kann. Im Laufe des Projekts wurde dieser Ausgang für eine Pegelanzeige, bestehend aus einem Voltmeter IC und einem LED-Balken, benutzt. Die Spannung an diesem Ausgang liefert Werte zwischen 0,6 (kein Empfangssignal) und ca. 4 Volt (Übersteuerung des Empfangsteils) und damit nur eine qualitative Beurteilung des Empfangspegels.

Die beiden verstärkten Signalkomponenten sowie deren spannungsinvertierten Signale werden ohne Gleichspannungsanteil an die Demodulatorschaltung weitergeleitet.

#### 4.4.2 Costas Loop

Die Schaltung (siehe Anhang A) hat einen analogen und einen digitalen Teil. Ein vereinfachtes Blockschaltbild zeigt Abbildung 4.13 .

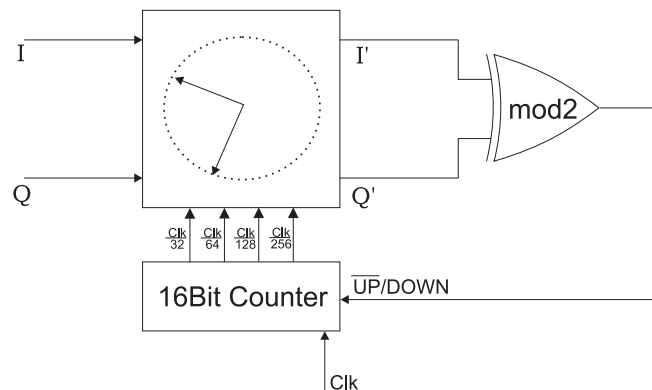


Abbildung 4.13: Costas Loop Blockschaltbild

Der 16Bit Zähler wird durch zwei kaskadierte 74HC191 (4-Bit Zähler) Bausteine realisiert und wählt je nach dem Stand der höchsten vier Bit einen bestimmten Phasenzustand aus. Die Auswahl erfolgt durch die Multiplexer Bausteine



74HC151. Das Widerstandsnetzwerk ist in Abbildung 4.14 anschaulicher dargestellt.

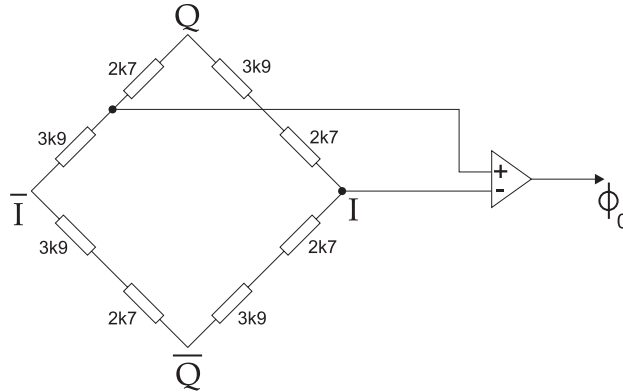


Abbildung 4.14: Widerstandsnetzwerk

Um die Entscheidungsgrenze des Signals  $\Phi_0$  auszurechnen, zeichnen wir obige Schaltung noch einmal und lösen das lineare Spannungsteiler Netzwerk folgendermaßen auf:

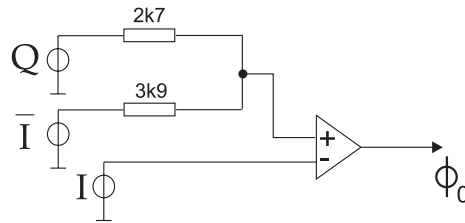


Abbildung 4.15: Spannungsteiler

$$I = Q \frac{3k9}{6k6} - I \frac{2k7}{6k6} \tag{4.7}$$

$$I = 0,59Q - 0,4I \tag{4.8}$$

$$Q = 2,37I \tag{4.9}$$

Die Entscheidungsgrenze dieses Komparators ist demnach eine um  $67,2^\circ$  geneigte Gerade durch den Ursprung der I/Q - Signalkonstellation. Die anderen Entscheidungsgrenzen ergeben sich analog ( Abbildung 4.17).

Genaugenommen ist diese Beschreibung nicht vollständig, da die Costas Loop eigentlich 16 Zustände haben sollte. Die restlichen acht sind aber exakt invers zu den hier errechneten, und können im darauf folgenden Logikteil leicht ergänzt werden. Für die weitere Beschreibung sei nun angenommen, daß die Signalkonstellation stationär um  $\phi = 10^\circ$  gedreht und das empfangene Symbol  $S=+1$

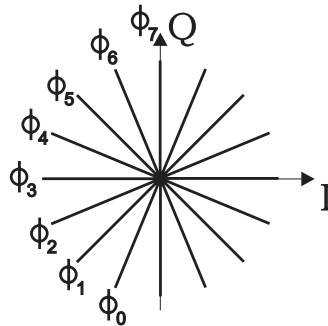


Abbildung 4.16: Entscheidungsgrenzen

ist. Als Anfangszustand wird  $\Phi_7$  angenommen, was einem Zählerstand von 1110 entspricht (LSB steht an erster Stelle). Beim Zählerstand 1110 wird nun der Kom-

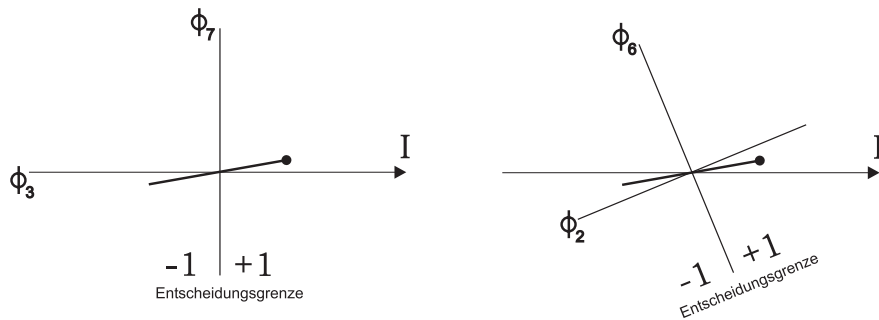


Abbildung 4.17: Symbol +1 bei 10° Phasendrehung

parator für den Zustand  $\Phi_7$  (OPV1, siehe Schaltung) auf logisch „1“ entscheiden und der Komparator des um 90° gedrehten Zustandes  $\Phi_3$  (OPV5) ebenfalls auf logisch „1“. Das Ergebnis der drei XOR Verknüpfungen ergibt nun ein Ausgangssignal RxD von logisch „1“ und eine Drehrichtung von logisch „1“, demnach „down“. Nach 16 Clockimpulsen wird sich daher der Zählerstand auf 0110 ändern. Das entspricht wiederum der Entscheidungsgrenze  $\Phi_6$ . Das Eingangssignal bleibe gleich bei  $S=+1$ . Der Komparator für den Zustand  $\Phi_6$  (OPV2) entscheidet auf logisch „1“, der für den Zustand  $\Phi_2$  (OPV6) auf logisch „0“. Das Ergebnis ist wiederum logisch „1“ auf der RxD Datenleitung, aber diesmal das Signal „up“ auf der Steuerleitung zum Zähler. Nach weiteren 16 Clockimpulsen wechselt daher der Zähler wieder in den Grundzustand 1110 und der Vorgang beginnt erneut. Aus den beiden zusätzlichen XOR Gattern kann man sehr anschaulich die Trennung der beiden Halbebenen bzw. die Unterscheidung der vier Quadranten in der Signalpunktkonstellation ableiten. Falls die Rotationsfrequenz kleiner als die bereits errechneten 56 kHz ist, ändert sich die Phasendrehung zwischen zwei Zählerzuständen um maximal einen Schritt. Die Schleife kann also folgen. Falls die Schleife in der unteren Halbebene einrastet, dann sind die empfangenen Daten

invertiert. Das Gesamtsystem entscheidet aber aufgrund der eingesetzten NRZI Kodierung letztlich trotzdem korrekt.

Die eingezeichneten D-FlipFlops dienen dazu, auftretende Glitches im Logikteil zu unterdrücken. Das 100k Widerstandsnetzwerk am Eingang der Demodulatorschaltung hebt den Pegel, sodaß keine negativen Spannungen auftreten. Der Oszillator ist ein TTL Baustein mit integriertem Schwingquarz.

# Kapitel 5

## PC Software

### 5.1 Systemvoraussetzungen

Das hier vorgestellte Projekt benötigt einen Personal Computer nach dem Industrie Standard, kurz PC genannt. Der Computer muß folgende Hardware-Grundvoraussetzungen erfüllen:

- Intel oder 100 Prozent kompatible CPU ab Generation Pentium 1
- Ein freier 16 Bit ISA Slot
- Ein freier Hardwareinterrupt und ein freier 8-Bit DMA Kanal
- 64 MByte RAM
- 2 GByte Festplatte
- CD-ROM Laufwerk (empfohlen)
- Ethernet Netzwerkkarte (empfohlen)
- VGA Grafikkarte

### 5.2 Eingesetzte Softwarekomponenten

Als Betriebssystemplattform kommt Linux zum Einsatz. Ein AX.25 Treiber für die Schnittstellenkarte ist im Betriebssystem enthalten [6]. Das Projekt ist nicht auf eine bestimmte Distribution beschränkt. In diversen Vorarbeiten hat sich die Distribution von *RedHat* bewährt. Der folgende Teil bezieht sich auf die zum Zeitpunkt der Diplomarbeit verfügbare Version 8.0.

Hardware wird in Linux direkt vom System-Kern (*Kernel*) angesprochen. Die Software Treiber werden entweder beim Starten des Betriebssystems geladen oder können während des Betriebs als *loadable module* dynamisch geladen werden. In beiden Fällen muß der Kernel allerdings darauf vorbereitet sein. Linux Distributionen werden meistens mit einem bereits fertig kompilierten Kernel ausgeliefert.

Das für die Schnittstelle benötigte Treibermodul *dmascc.c* ist dabei im allgemeinen in diesem ausgelieferten Kernel nicht enthalten. Bei *RedHat* wird das Treibermodul aber zumindest als C Source Code mitgeliefert. Eine Neukompilierung des Kernels ist daher zwingend erforderlich. Im Zuge dessen empfiehlt es sich, von <http://www.kernel.org> oder einem Mirror Server die momentan aktuelle Kernelversion zu beziehen. Der Installationsvorgang von *RedHat 8.0* und der Umstieg auf eine aktuelle Kernelversion ist in Anhang D bzw. Anhang E beschrieben.

Die hier vorgestellte Konfiguration entspricht der minimalen Ausstattung, die notwendig ist, um zwei Computer (Funkgateways) über die Funkschnittstelle zu verbinden. Weiters soll es möglich sein, eine Kommunikation zwischen den beiden drahtgebundenen Netzwerken, in die die Funkgateways eingebunden sind, herzustellen (Routerbetrieb).

Das Funkgateway besteht demnach aus folgenden Softwarekomponenten:

- Linux Kernel ab Version 2.2 (Version 2.4 empfohlen)
- Treibermodul für eingebaute Ethernetkarte
- Treibermodul für die Schnittstellenkarte
- TCP/IP Protokoll
- AX.25 Protokoll
- IP Masquerading
- OpenSSH
- GCC (GNU C Compiler)
- Kernel development Kit
- *dmascc.cfg* Konfigurationsscript

## 5.3 Konfiguration der Schnittstellenkarte

### 5.3.1 Hardware Konfiguration

Die Schnittstellenkarte wird über den ISA Bus des Computers angesteuert. Die Kommunikation mit dem ISA Bus erfolgt Byteweise. Der Input/Output Adressbereich auf dem ISA Bus liegt zwischen der Adresse 0000h und FFFFh. Die Adressbereiche verschiedener Peripheriegeräte innerhalb dieses Adressraumes dürfen sich nicht überlappen. Des weiteren müssen noch ein Interrupt und ein DMA Kanal gewählt werden. Die entsprechende Jumper-Belegung ist Anhang H zu entnehmen.

### 5.3.2 Inbetriebnahme des Netzwerkinterface

Das Kommunikationsprotokoll AX.25 ist in den Linux Distributionen im allgemeinen enthalten. Die Schnittstellenkarte gliedert sich in das System als Netzwerkinterface mit dem Namen `DMASCC0` ein. In der Konfigurationsdatei `/etc/modules.conf` muß folgender Eintrag ergänzt werden:

```
options dmascc io=0x230
```

Die I/O Adresse muß mit dem auf der Schnittstellenkarte eingestellten Adressbereich übereinstimmen. Durch den Befehl `modprobe dmascc` wird nun das Treibermodul geladen. Sollte es hier zu einer Fehlermeldung kommen, empfiehlt sich ein Blick in das Fehlerprotokoll mit dem Befehl `dmesg`. Ist in dem Computer eine zweite Schnittstellenkarte aktiv, so würde man diese über den Namen `DMASCC2` ansprechen. Das Treibermodul von Klaus Kudielka muß nach dem Einladen noch durch das Konfigurationsscript `dmascc_cfg` aktiviert werden (siehe Anhang G). Danach kann man dem Netzwerkinterface mit dem Linux eigenen Konfigurationstool `ifconfig` eine IP Adresse zuweisen. Als Beispiel sei hier angeführt:

```
ifconfig dmascc0 44.143.22.12 hw ax25 0E3CDB-1 netmask 255.255.255.0
```

Die beiden Parameter `hw ax25` teilen dem Kernel mit, daß dieses Interface über AX.25 angesprochen werden soll. `0E3CDB-1` ist ein Amateurfunkrufzeichen und steht für die MAC-Adresse (*Media Access Control*) des Interface - Diese muß innerhalb des Funkfeldes eindeutig sein. Damit ist die Funkschnittstelle für TCP/IP bereits fertig konfiguriert. Vorläufig kann man den beiden Stationen beliebige IP-Adressen zuordnen. Für einen einfachen Übertragungsversuch bietet sich das Tool `ping` an, mit dem man ICMP Kommandos an die jeweils andere Station schickt. Die gerufene Station antwortet auf das `ping` Paket, indem es seinen Inhalt übernimmt und an die rufende Station zurückschickt. Die rufende Station mißt dabei die Laufzeit (*round Trip Time*), bis das Paket wieder vollständig zurückgekommen ist. Bitfehler auf der Übertragungstrecke werden von AX.25 durch die integrierte Prüfsumme CRC (*Cycle redundancy Check*) sofort erkannt. Das Paket wird dann verworfen und eine entsprechende Meldung innerhalb des Protokollstacks an IP übergeben. Der `ping` Befehl zählt die verlorenen Pakete und gibt sie unter dem Begriff *packet loss* am Bildschirm aus.

## 5.4 Gatewaykonfiguration

Sobald die Kommunikation über TCP/IP funktioniert, kann man die beiden Computer nun für den Gatewaybetrieb konfigurieren. Ein mögliches Szenario wäre demnach ein privates Netzwerk, das man über die Funkschnittstelle mit dem Internet verbinden will. Da man im privaten Netz typischerweise private IP Adressen, z.B. im Adressraum 192.168.0.0/24, einsetzt, muß man eine IP-

Adressumsetzung einsetzen. Im Folgendem wird nun eine Lösung beschrieben, bei der man sowohl beim Übergang privates LAN  $\Rightarrow$  Funkfeld als auch beim Übergang Funkfeld  $\Rightarrow$  Internet auf den beiden beteiligten Gateway Computern eine Adressumsetzung implementiert (Abbildung 5.1).

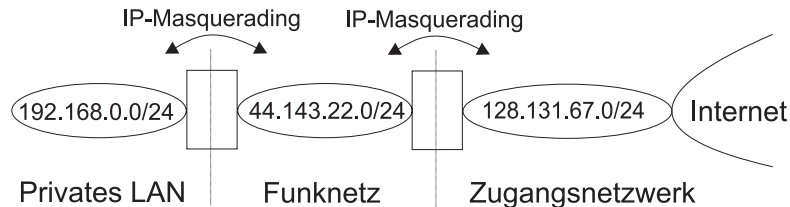


Abbildung 5.1: Adressumsetzung (*IP-Masquerading*)

Die Weiterleitung von IP-Paketen zwischen zwei physikalisch getrennten Netzwerkschnittstellen wird über die Konfigurationsdatei `ip_forward` folgendermaßen aktiviert.

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

Eine einfache Weiterleitung ohne zusätzliche Maßnahmen erfüllt aber in diesem Fall nicht die Anforderungen, da zumindest eines der beiden Netzwerkinterfaces eine private IP-Adresse besitzt, die im Internet nicht weitergeleitet wird. Zusätzlich muß man daher über den Befehl `iptables` die Netzwerkadressumsetzung aktivieren. Zur Illustration der nun folgenden Konfigurationsschritte sei an dieser Stelle die Netzwerkstruktur des beschriebenen Beispiels detailliert dargestellt (Abbildung 5.2).

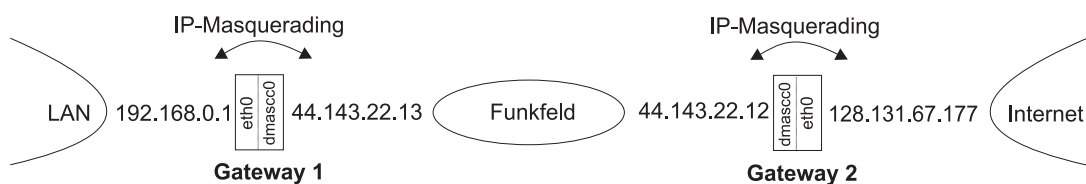


Abbildung 5.2: Netzwerkstruktur

Eine Adressumsetzung auf dem Computer „Gateway 1“ wird durch den folgenden Befehl eingerichtet:

```
iptables -t nat -A POSTROUTING -o dmascc0 -j MASQUERADE
```

Dadurch wird die Absenderadresse von Paketen manipuliert, die den Systemkern in Richtung der Schnittstelle `DMASCC0` verlassen. Auf dem Computer „Gateway 2“ lautet der Befehl daher entsprechend:

```
iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

Wichtig ist ebenfalls, daß man auf beiden Computern in den Netzwerkeinstellungen ein sogenanntes *Default Gateway* einrichtet. Pakete, deren Empfängeradresse außerhalb der drei oben gezeigten Netze liegt, werden an die dort eingetragene Adresse weitergeleitet. Auf „Gateway 1“ muß dieser Eintrag folgendermaßen aussehen: `default via 44.143.22.12 dev eth0`. Der dafür notwendige Befehl lautet:

```
route add default gw 44.143.22.12
```

Für den Computer „Gateway 2“ lautet der Eintrag in diesem Beispiel: `default via 128.131.67.1 dev eth0`. Diese Adresse entspricht dem Router, der das Netzwerk mit dem Internet verbindet.

Der Befehl `iptables` wird auch dann eingesetzt, wenn es darum geht, bestimmte Filterregeln für ein- und ausgehende Pakete zu erstellen. Speziell bei „Gateway 2“ sollte man zumindest einige grundlegende Paketfilter definieren, da dieser Computer permanent mit dem Internet verbunden ist und daher vor möglichen Angriffen aus dem weltweiten Netz geschützt werden muß.

Das Beispielskript in Anhang F `iprulez` ist in der hier abgedruckten Form für den Computer „Gateway 2“ voll funktionsfähig. Durch die Definition von Variablenamen wie zum Beispiel `INET_DEV` kann es sehr schnell an ähnliche Umgebungen angepasst werden. Aufgerufen wird es über die beiden Parameter `start` und `stop`, also z.B.

```
./iprulez start
```

um die komplette `iptables` Gatewaykonfiguration zu aktivieren. Die Firewallregeln im zweiten Abschnitt des Beispielskripts schränken den Zugang auf den Computer von außen rigoros ein. Danach ist nur noch ein Zugriff über das Protokoll *SSH* möglich, das einen Fernwartungszugriff auf das System erlaubt.

## 5.5 Weitere Konfigurationsoptionen

### 5.5.1 Erweiterung auf eine Verschlüsselungstechnologie

Bei der Verwendung von Funknetzen ist es außerdem üblich, eine Verschlüsselung der Daten einzusetzen. Prinzipiell ist das Funkfeld ja nicht auf den Bereich zwischen den beiden Antennen beschränkt, und die Datensignale können unter Umständen durch Reflexionen auch an anderen Orten empfangen werden. Ein einfaches Erraten der Scrambling-Sequenz genügt demnach für das Abhören des Datenstromes. Im Hinblick auf den experimentellen Charakter des Projekts und aufgrund der Tatsache, daß es sich nicht um sensible Daten handelt, haben wir von einer Verschlüsselung abgesehen. Eine nachträgliche Implementierung auf Basis des Standards *IPSec* ist aber unter Linux mithilfe des Software Paketes



*FreeS/Wan* einfach durchzuführen und wurde auch kurzzeitig ausprobiert. Diese Art der Verschlüsselung wirkt direkt auf die einzelnen IP-Pakete und ist für die höher liegenden Anwendungsschichten vollkommen transparent. Durch den Einsatz von digitalen Zertifikaten und einer Verschlüsselung nach dem Algorithmus *3DES* erreicht man eine Schlüssellänge von 168 Bit, die nach heutigen Maßstäben als sehr sicher bewertet wird. *IPSec* beinhaltet bereits eine Adressumsetzung, so daß in diesem Fall die hier angeführten Weiterleitungsoptionen angepaßt werden müssen. Verschlüsselungsprotokolle auf höheren Ebenen, wie z.B. SSL (*Secure Sockets Layer*), können selbstverständlich auch in der derzeitigen Betriebsart verwendet werden, um einzelne Übertragungen gezielt zu schützen.

### 5.5.2 Erweiterung auf ein Mehrbenutzerszenario

Auf die in der Einleitung hingewiesene Mehrbenutzeroption soll an dieser Stelle auch noch kurz eingegangen werden. Der Betrieb von mehreren Teilnehmern auf einem *Shared Medium* erfordert prinzipiell folgende Eigenschaften:

- gezielte Adressierung der einzelnen Teilnehmer
- Mechanismus für den Kanalzugriff
- Kollisionserkennung

Die Adressierung im Funkfeld basiert auf den Amateurfunk Rufzeichen. Der Mechanismus für die Zuordnung von Rufzeichen auf IP-Adressen und umgekehrt heißt ARP (*Address Resolution Protocol*). Jede Teilnehmerstation betreibt eine Tabelle, in der jeder IP Adresse, mit der bereits erfolgreich kommuniziert wurde, ein Rufzeichen (*MAC-Adresse*) zugeordnet ist. Vor einem Verbindungsaufbau zu einer bestimmten IP Adresse wird in dieser Tabelle die betreffende MAC-Adresse gesucht, und das AX.25 Paket praktisch sofort mit der korrekten Empfänger Adresse bestückt. Im Funkfeld ist die Station dann eindeutig adressiert, nimmt das Paket entgegen und leitet es an seine IP Schicht weiter. Die gerufene Station kann nun auch sofort direkt antworten, da die anrufende Station ihre MAC-Adresse im Paket ebenfalls mitgeliefert hat. Falls in der ARP-Tabelle keine Zuordnung möglich ist, muß die Sendestation einen Rundruf (*Broadcast*) im Funkfeld durchführen. Das Broadcast Paket benützt dafür das Rufzeichen *QST-0* als Zieladresse und schickt die nachzufragende IP-Adresse im Datenfeld mit. Jede Station leitet dieses Paket an seine IP Schicht weiter und vergleicht die nachgefragte IP Adresse mit ihrer eigenen. Diejenige, die nun tatsächlich gesucht wird, antwortet mit ihrer MAC-Adresse und die rufende Station nimmt diese in ihre ARP-Tabelle auf. Dieser Mechanismus skaliert sehr gut mit der Anzahl der Teilnehmer und wird auch auf Ethernet Systemen erfolgreich eingesetzt.

Sind mehr als zwei Teilnehmer in einem Funkfeld aktiv, muß es einen weiteren Mechanismus geben, der die vorhandene Kanalkapazität auf alle Teilnehmer

aufteilt, und zwar unter den beiden Gesichtspunkten maximaler Datendurchsatz einerseits und akzeptable Antwortzeiten andererseits. AX.25 benützt hierfür das Verfahren CSMA/CA (*Carrier Sense Multiple Access/Collision Avoidance*). Im Unterschied zu drahtgebundenen Netzen ist es bei Funknetzen im allgemeinen für eine gerade sendende Station nicht möglich, festzustellen, ob eine andere Station zum selben Zeitpunkt sendet und damit die Übertragung stört. Zerstörte Pakete reduzieren den Datendurchsatz von TCP Verbindungen erheblich und sollen weitgehend vermieden werden. Vor jedem Sendepaket fragt der Treiber daher den Status des Kanales ab. Das DCD Signal der Schnittstellenkarte dient als Indikator für die Zustände „Kanal frei“ bzw. „Kanal besetzt“. Auch bei einem vermeintlich freien Kanal kann es zu Kollisionen kommen, nämlich dann, wenn zwei Teilnehmer unabhängig voneinander zum selben Zeitpunkt ein Sendesignal absetzen. Unter Einbeziehung der beiden Treiber Parameter *Persist* und *Slottime* (siehe Anhang G) wartet eine Station, die gerade Sendedaten in ihrem Sendepuffer hat, nach jedem Zustandswechsel von „Kanal besetzt“ zu „Kanal frei“ eine zufällige Zeitspanne ab, bevor sie den Sendevorgang startet. Innerhalb dieser Zeitspanne läuft der Kanal leer, was einem Verlust an Übertragungskapazität gleichkommt. Durch eine gute Wahl der beiden Treiberparameter können Kollisionen minimiert und damit der systembedingte Verlust an Übertragungskapazität mehr als wettgemacht werden. Die Optimierung der betreffenden Treiberparameter ist in dem hier vorliegenden Mehrwegekanal allerdings eine anspruchsvolle Aufgabe und sicher eine interessante Option für etwaige Nachfolgeprojekte.

### 5.5.3 Repeaterbetrieb

Das AX.25 Adressfeld besteht nicht nur aus Sende- und Empfangsadresse, sondern kann auch die Adressen von eventuell beteiligten Zwischenstationen (*Repeatern*) aufnehmen. Anders als bei TCP/IP wird aber nicht im Repeater entschieden, welchen Weg das Paket weiterverfolgen soll, sondern die gesamte Route mit allen Zwischenstationen muß bereits in der Sendestation bekannt und in das AX.25 Paket mit übergeben werden. Große dynamische Netze, wie zum Beispiel das Internet, lassen sich auf diese Weise daher nicht realisieren. Für kleinere, mehrfach verzweigte Netze kann man diese Eigenschaft des Protokolls aber dafür einsetzen, um entweder die Reichweite des Netzwerks zu vergrößern, oder um eine Kommunikation zwischen Stationen aufzubauen, die ansonsten nicht direkt miteinander kommunizieren könnten (Zum Beispiel aufgrund mangelnder Sichtverbindung). Auch eine Tunnelfunktion, bei der man zwei unabhängige Funknetze über ein leitungsgebundenes Netzwerk verbindet, ist denkbar. Unabhängig von AX.25 hat man aber mit TCP/IP ein sehr leistungsstarkes, dynamisch routbares Protokoll, auf das man für diese Zwecke in jedem Fall zurückgreifen kann.

# Kapitel 6

## Messungen

### 6.1 Allgemeines

Die durchgeführten Messungen gliedern sich in drei Gruppen:

- Messungen im Labor mit Koaxialkabelverbindung zwischen den Funkmodems
- Messungen im Labor mit Verbindung über zwei Dachantennen
- Messungen an der tatsächlichen Funkstrecke TU-Wien - Hietzing

Die erste Inbetriebnahme der Funkmodems fand praktisch ausschliesslich im Labor statt. Die beteiligten Modems wurden über ein Koaxialkabel verbunden und eine Eichleitung mit variabler Dämpfung dazwischen geschaltet. Die Modems befanden sich im selben Raum und hatten einen Abstand von wenigen Metern zueinander. Trotz der Schirmung der einzelnen Modems kann man in dieser Konfiguration eine Kopplung zwischen den beiden Gehäusen nicht ausschliessen, speziell bei einer hohen Dämpfung der Eichleitung. Die Ergebnisse der ersten Messungen sind daher unter diesem Gesichtspunkt zu betrachten. Im zweiten Schritt wurde zwischen zwei Antennen am Dach des Instituts eine Testfunkstrecke aufgebaut. In diesem Fall stehen nun die Modems in unterschiedlichen Stockwerken, so daß eine direkte Kopplung auszuschließen ist. Durch das Einfügen einer zusätzlichen variablen Dämpfung vor dem Empfangsmodem und einer vorangegangenen Messung der Dämpfung dieser Dachfunkstrecke, war es nun möglich, die tatsächliche Funkstrecke nach Hietzing zu simulieren. Diese Ergebnisse werden schlußendlich mit den Ergebnissen aus den letzten Experimenten verglichen, die an der eigentlichen Richtfunkstrecke nach Hietzing durchgeführt worden sind.

### 6.2 Labormessungen

Nach dem Zusammenbau und dem ersten Funktionstest der Modems wurde in einem ersten Schritt die Ausgangsleistung der Geräte gemessen. An dieser Stelle seien nun zum Zwecke der Unterscheidung der einzelnen Modems untereinander

deren „Namen“ eingeführt. Die Ausgangsleistung wurde mit einem Wattmeter der Firma Bird im 1 Watt Meßbereich durchgeführt.

Newton	0,55W	27,4 dBm
Faraday	0,68W	28,3 dBm
Maxwell	0,72W	28,6 dBm
Marconi	0,76W	28,8 dBm

Die Schwankungen lassen sich einfach durch Bauteil- und Fertigungstoleranzen, speziell im Bereich der Mikrostreifenleiter, erklären.

Eine weitere Frage war nun, ab welchem Empfangspegel die ersten Bitfehler auftreten. In dieser Hinsicht unterscheiden sich die Modems praktisch nicht. Bei der Messung über das Koaxialkabel war dabei eine scharfe Grenze zu beobachten. Bis zu einem Empfangspegel von -88 dBm trat auch bei längerer Beobachtungszeit kein Bitfehler auf. An dieser Stelle sei aber hier nochmals auf das Problem der Kopplung hingewiesen. Eine exakte Bitfehlermessung mit räumlich getrennten Modems war demnach notwendig und wird im nächsten Abschnitt beschrieben.

Der Abgleich der lokalen Oszillatoren der einzelnen Modems erfolgte auf einer Frequenz von 638,452 MHz. Die Stabilität im Dauerbetrieb blieb deutlich unter  $10^{-6}$  und somit für unsere Anforderungen ausreichend, jedoch war eine recht starke Temperaturabhängigkeit zu beobachten. Die LO Ausgangsleistung geht direkt in die effektive Ausgangsleistung des Gesamtsystems ein und erreichte im Zuge des Abgleichs bei jedem Modem ca. 11 dBm.

In Abbildung 6.1 ist nun der Bereich um die Mittenfrequenz des gemessenen Ausgangsspektrums dargestellt. Die darauffolgende Abbildung 6.2 ist das Ergebnis einer Berechnung, die nach der Formel für das Signalspektrum, Gleichung 4.1 und Gleichung 4.2 aus Kapitel 4, erstellt wurde. Die Nullstellen in der berechneten Grafik wurden im Bild unterdrückt - Sie treten im Realsystem nicht auf. Die Berechnung erfolgte im Basisband und beinhaltet nur positive Frequenzen. Aus Abbildung 6.1 läßt sich nun die effektive 3 dB Übertragungsbandbreite mit ca. 1,5 MHz ablesen. Die Breite des Hauptmaximums beträgt 2,4 MHz. Abbildung 6.3 zeigt die Kombination aus Messung und Berechnung in einem Diagramm.

Abbildung 6.4 stellt praktisch den gesamten Frequenzbereich des Ausgangsspektrums dar. Hierbei erkennt man neben den gewünschten Spektralanteilen bei 1276 MHz auch unerwünschte Komponenten. Bei der halben Trägerfrequenz sieht man etwa, daß auch Spektralanteile des lokalen Oszillators, der ja auf 638 MHz schwingt, durch den Demodulator hindurch auf den Antennenausgang gelangen. Ebenso existieren Spektralanteile auf der doppelten und dreifachen Trägerfrequenz. Die Unterdrückung dieser Komponenten ist leicht abzulesen und beträgt 40 bzw. 50 dB. Schlußendlich ist in Abbildung 6.5 ein Frequenzbereich zu sehen, der das komplette 23 cm Band umfaßt.

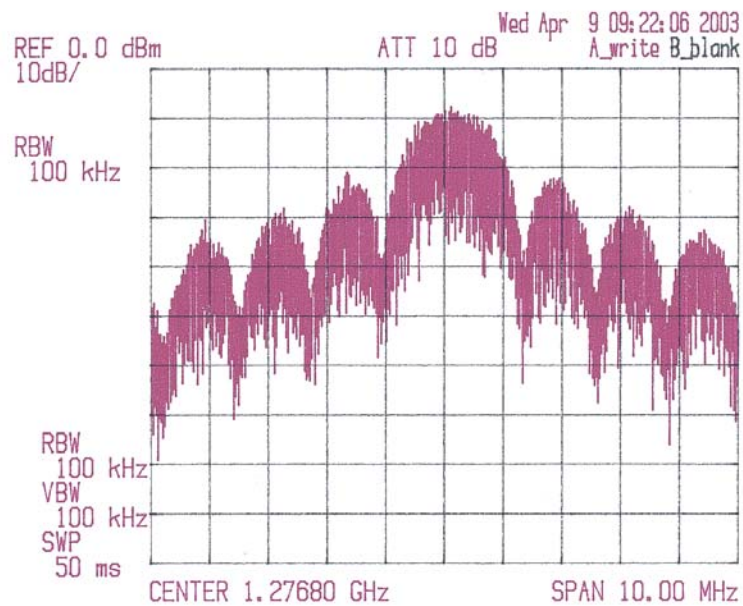


Abbildung 6.1: Gemessenes Ausgangsspektrum

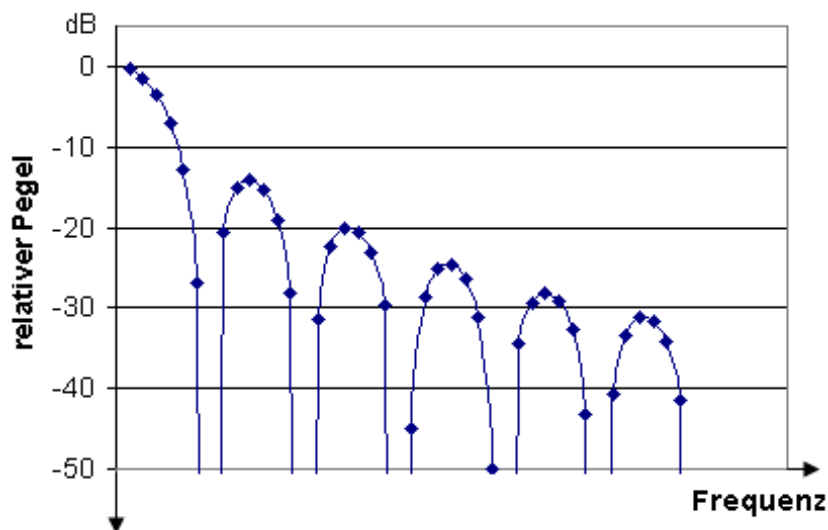


Abbildung 6.2: Berechnetes Ausgangsspektrum

### 6.3 Systemtest

Der Systemtest bestand, wie oben beschrieben, aus einer Funkstrecke am Dach des Institutsgebäudes. Mithilfe eines 1 Watt Handfunkgerätes und eines Spektrumanalysators konnte die effektive Streckendämpfung inklusive Zuleitungskabel und Antennengewinn mit 66 dB gemessen werden. Anhand des Pegelplans aus

Kapitel 2 kann nun sofort die Zusatzdämpfung errechnet werden, die für eine Simulation der tatsächlichen Funkstrecke nach Hietzing notwendig ist.

$$L_{\text{zusatz}} = +27\text{dBm} - 137\text{dB} + 66\text{dB} = -44\text{dB} \quad (6.1)$$

Wichtig ist dabei, daß diese Zusatzdämpfung vor dem Empfangsmodem eingefügt wird. Würde man den Abschwächer senderseitig einfügen, so wäre das Signal auf der Funkstrecke sehr schwach und etwaige Interferenzen würden sich stark auswirken. Der Empfangspegel an der Antennenbuchse des Empfangsmodems beträgt nun, wie gewünscht, -81 dBm.

Die Inbetriebnahme mit dem zusätzlichen 44 dB Abschwächer funktionierte auf Anhieb und, wie erwartet, auch ohne Paketverluste. An dieser Stelle konnte nun mit dem Linux Tool *iperf* die effektive TCP Netto-Datenrate ermittelt werden. Mit den im Anhang G beschriebenen Timingparametern ergab sich bei fehlerfreier Übertragung eine konstante Datenrate von 690 kbit/s. Die Diskrepanz zwischen Bruttodatenrate (1,22 Mbit/s) und diesem Wert erklärt sich einerseits aus dem Protokolloverhead von AX.25 und TCP/IP, aber auch durch die bescheiden gewählten Timingparameter, durch die speziell bei der Umschaltung zwischen Sende- und Empfangsmodus in Form von Einschwingvorgängen Zeit verloren geht. Die gemessene Roundtriptime des *ping* Befehles von ca. 34 ms pro 64 Byte Paket bestätigt diese Behauptung.

Um nun die tatsächliche Empfindlichkeit des Modulators festzustellen, ist eine Bitfehlermessung notwendig. Wie bereits in Kapitel 4 kurz angedeutet, würde der Scrambler das Ergebnis verfälschen und wird daher überbrückt. Für die Funktion des Gesamtsystems ist eine annähernd gleichwahrscheinliche Symbolverteilung notwendig. Der professionelle Bitfehlermeßplatz der Firma *Anritsu*, bestehend aus Sender und Empfänger, erfüllt diese Bedingung durch den Einsatz einer Pseudozufallsfolge mit der Periode 32765. Sender und Empfänger können unabhängig voneinander betrieben werden. Die Symboltaktückgewinnung auf der Schnittstellenkarte regeneriert den Systemtakt, dadurch kann sich der Empfänger auf die empfangene Bitfolge synchronisieren und erkennt falsch detektierte Bits. Der Sender liefert die Daten im NRZ Format. Durch Einspeisen eines externen Taktsignales ist es möglich, die Bitrate auf die erforderlichen 1,2288Mbit/s einzustellen. Am Display des Empfängers kann man direkt die BER (*Bit error rate*) ablesen. Für hinreichend genaue Messungen empfiehlt sich eine Meßzeit (*gating time*) von 120 Sekunden. Die exakte Bedienung des Gerätes ist [4] zu entnehmen. Das Ergebnis der Messung ist nun die Bitfehlerwahrscheinlichkeit in Abhängigkeit des Empfangspegels.

Pegel	BER
-89 dBm	1,00E-07
-90 dBm	9,00E-07
-91 dBm	9,40E-06
-92 dBm	2,40E-04
-93 dBm	7,60E-04
-94 dBm	2,10E-03
-95 dBm	4,80E-03
-96 dBm	1,00E-02

Eine grafische Auswertung zeigt Abbildung 6.4 . Dort ist nicht nur die gemessene Bit Error Rate eingezeichnet, sondern auch eine Vergleichskurve, die eine ideale BPSK Demodulation symbolisieren soll. Der SNR-Verlust von 4 dB basiert auf einer Schätzung und resultiert aus den SNR-Verlusten im Eingangsfiler, den diskreten Entscheidungsgrenzen und anderen Faktoren, deren Ursache im Bereich von Fertigungs- und Bauteiltoleranzen zu suchen ist.

Die ideale BPSK Demodulationskurve folgt der Q-Funktion. Eine qualitative Ähnlichkeit mit der realen Meßkurve ist klar ersichtlich. In Verbindung mit dem Pegelplan aus Kapitel 2 läßt sich aus der Grafik eine Systemreserve von ca. 8 dB ablesen. Dieses Ergebnis ist aber unter dem Gesichtspunkt zu betrachten, daß bis jetzt praktisch nur idealisierte Versuche im Labor stattgefunden haben. Auf der tatsächlichen Funkstrecke muß mit weiteren negativen Einflüssen gerechnet werden.

## 6.4 Messungen an der Richtfunkstrecke

Den Abschluß der Versuchsreihe bildet nun die Inbetriebnahme der Funkstrecke zwischen der TU-Wien und dem Endpunkt in Hietzing. Bereits im Zuge der ersten Kontaktaufnahme zwischen den beiden Stationen kam es zu vereinzelt Paketfehlern, deren Verteilung uns zu einem Experiment veranlaßte, welches die Paketfehler über einen längeren Zeitraum beobachten sollte. Im Zuge dieses Experiments wurden jeweils 1000 *ping* Pakete mit einer Länge von 64 Byte am Standort TU-Wien generiert. Falls das Paket über die Funkstrecke und wieder zurück ohne Bitfehler übertragen wurde, wird es als gültig gezählt. Gestörte Pakete werden verworfen und gehen somit verloren. Der Versuch erstreckt sich über 24 Stunden. Zu Beginn der vollen Stunde werden 1000 Pakete im Abstand von einer Sekunde erzeugt. Die grafische Auswertung zeigt Abbildung 6.7 .

Das Histogramm zeigt eine Tageszeitabhängigkeit der Paketfehlerrate zwischen 0% und 3%. Dieses Ergebnis deutet auf eine Störung im Funkkanal durch *Man Made Noise* hin. Ideen für etwaige Verbesserungen sind im nächsten Kapitel angeführt. Da im Gesamtsystem keine Fehlerkorrektur vorgesehen ist, und auch die AX.25 Übertragung verbindungslos und ohne Paketbestätigungen abläuft, wirken

sich verworfene Datenpakete, die letztlich erst von TCP neu angefordert werden, in der Nettodatenrate relativ stark aus. Die oben angeführte Datenrate von 690 kbit/s wird nur im fehlerfreien Fall erreicht. Bei einer Paketfehlerrate von 3% bricht die Nettodatenrate auf ca. 300 kbit/s ein. Ein zweiter Versuch demonstriert die Stabilität der Verbindung trotz vereinzelter Paketfehler:

Datenübertragung	Zeit	durchschnittliche Datenrate
675.315.712 Byte	4h 04m 05s	46112 Byte/s = 368,898 kbit/s

Eine genaue Analyse der Kanaleigenschaften wird aber erst nach einer exakten Bitfehlermessung nach obigem Vorbild möglich sein.



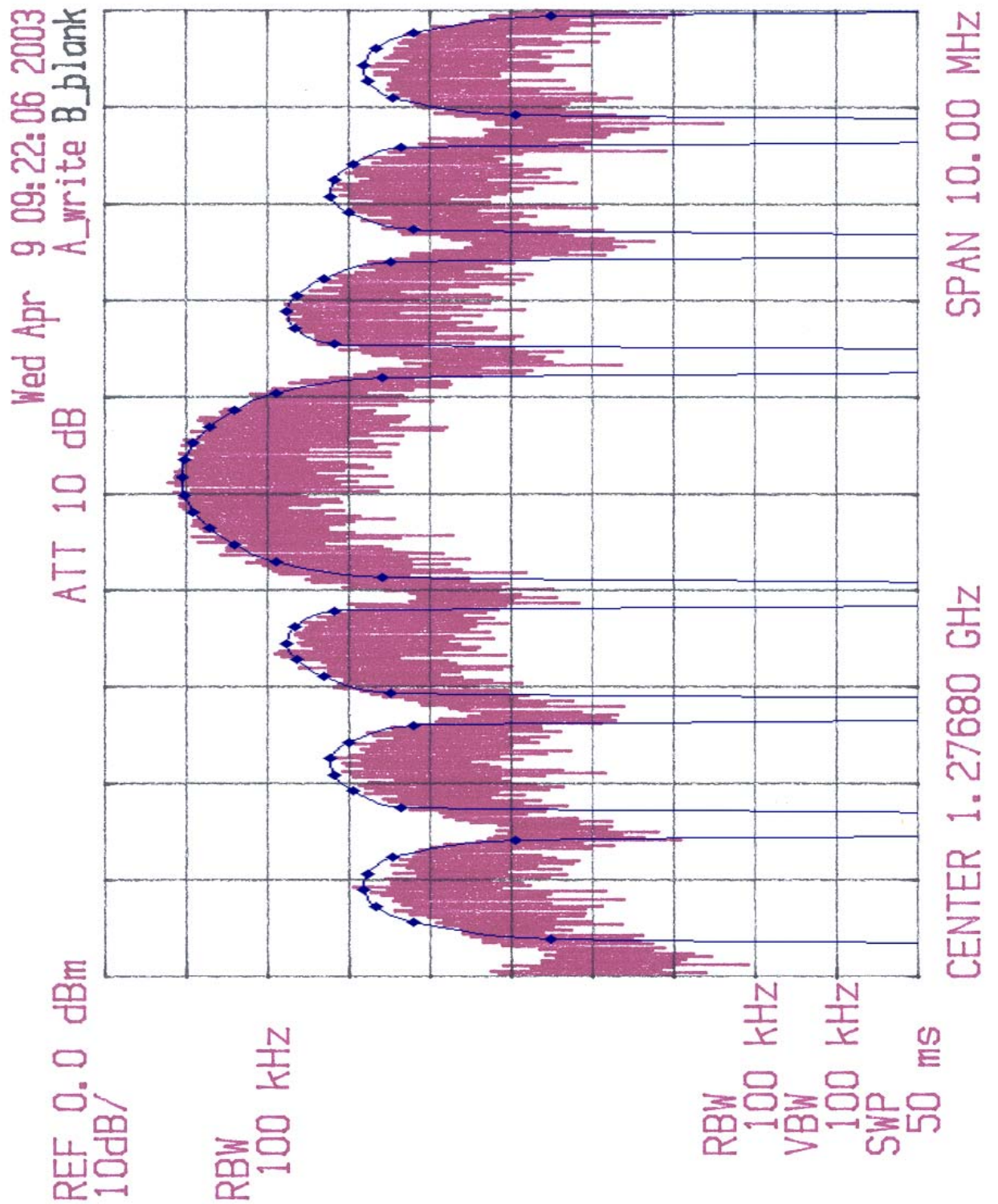


Abbildung 6.3: Gemessenes Ausgangsspektrum verglichen mit Berechnung

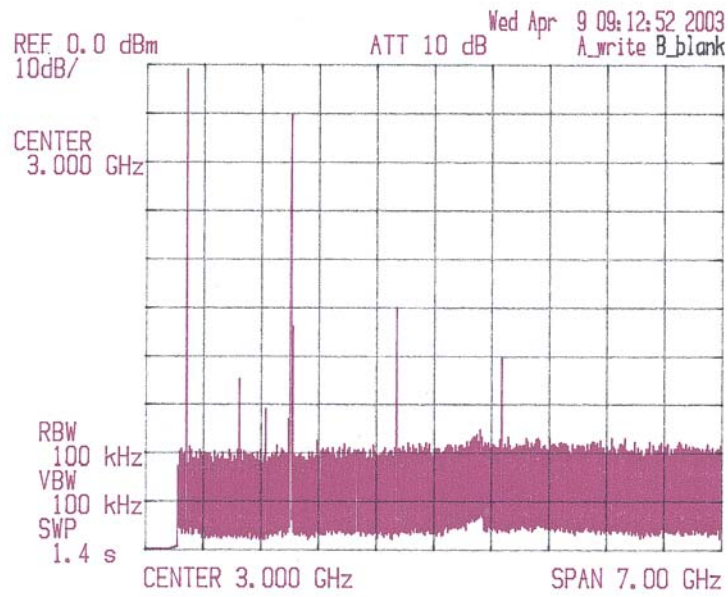


Abbildung 6.4: Gemessenes Gesamtausgangsspektrum

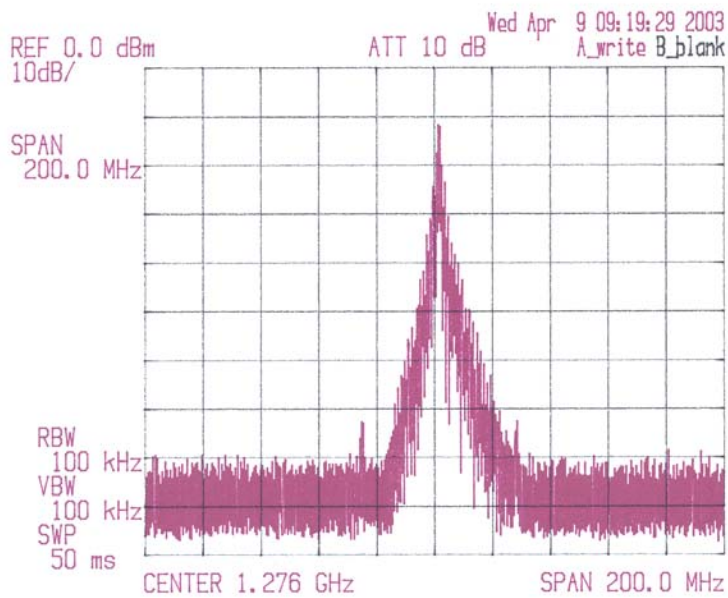


Abbildung 6.5: Ausgangsspektrum unter Betrachtung des gesamten 23 cm Bandes

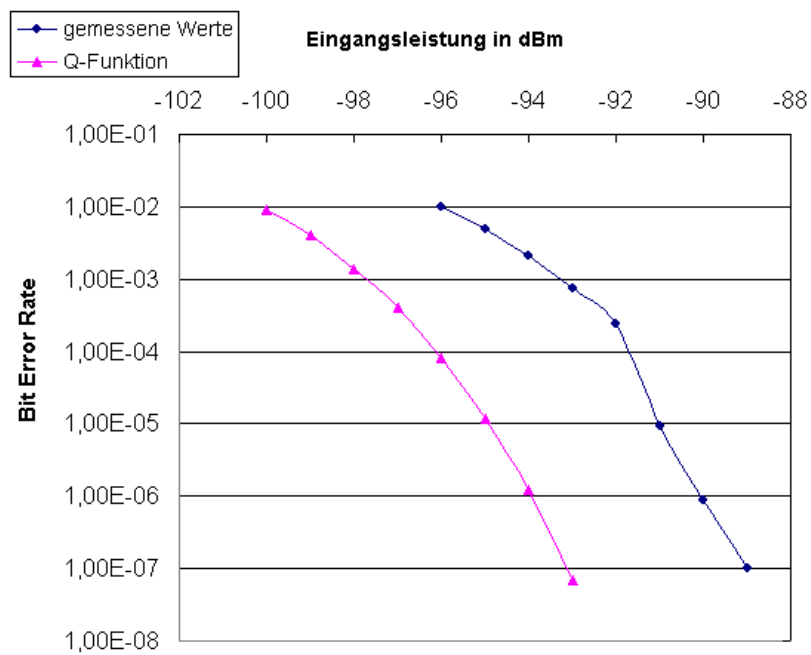


Abbildung 6.6: Gemessene Bitfehlerrate

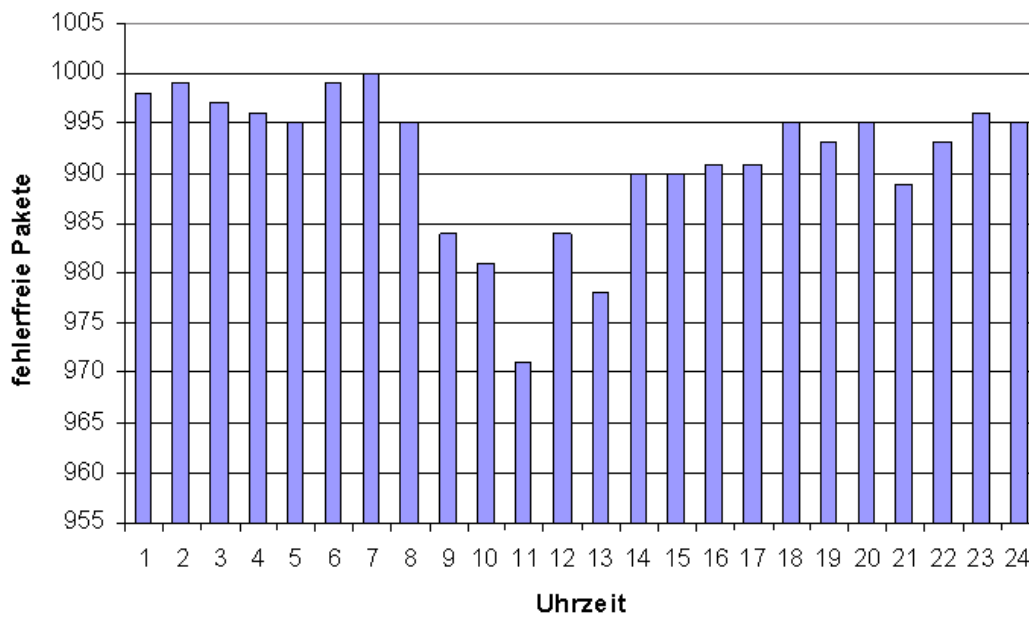


Abbildung 6.7: 24 Stunden Paketfehlerexperiment

# Kapitel 7

## Schlußfolgerungen

Im Zuge der Diplomarbeit wurden alle vier Modems zur Funktion gebracht. Alle vier unterscheiden sich in ihren wesentlichen elektrischen Eigenschaften praktisch nicht. Angesichts der großen Streckendämpfung und der relativ geringen Sendeleistung rechnete ich bereits im Vorfeld damit, daß wir nur eine geringe Systemreserve für den Betrieb der Funkstrecke haben würden. Stabile Datenübertragung mit durchaus beeindruckenden Geschwindigkeiten, auch von größeren Datenmengen, wurde mehrfach demonstriert. Verbesserungen sind natürlich immer möglich. Einige davon seien hier, ohne Anspruch auf Vollständigkeit, aufgezählt:

- Antennenstandorte erhöhen und dadurch die Streckendämpfung verringern
- Sendeleistung erhöhen
- Kabellängen minimieren, z.B. indem man das Modem outdoor direkt neben der Antenne betreibt
- Genaue Analyse der Störungen und möglicherweise Ausweichen in einen anderen Bereich des 23 cm Bandes
- Fehlerkorrektur auf Bitebene zwischen Schnittstellenkarte und Modem
- Bandbreitenreduktion durch Sendefilter

# Anhang A

## Schaltungen und Stücklisten

### A.1 Stückliste Vorverstärkerplatine

Bezeichnung	Wert/Typ	Anzahl
Widerstand	1k5	2
Widerstand	2k7	6
Widerstand	470	5
Widerstand	4k7	1
Widerstand	68k	1
Widerstand	10k	1
Widerstand	22E	1
Drossel	220 $\mu$	2
SIP Widerstandsleiste	8x10k	1
Kondensator	150p	2
Kondensator	220p	2
Kondensator	470 $\mu$	1
Kondensator	100n	6
Kondensator	6n8	2
Kondensator	470 $\mu$	1
Kondensator	22 $\mu$	1
Kondensator	1 $\mu$	1
Kondensator	1n8	4
Kondensator	220n	4
Diode	1N4151	10
IC	NE592	2
IC	CD4049	1
Stiftleiste	5pol	1
Stiftleiste	7pol	1

## A.2 Stückliste Costas Loop

Bezeichnung	Wert/Typ	Anzahl
IC	LM311	8
IC	74HC191	2
IC	74HC151	2
IC	74HC86	1
IC	74HC174	1
IC	LM7805	1
IC	Quarzoszillator 14,3MHz	1
Widerstand	2k7	4
Widerstand	3k9	4
SIP Widerstandsleiste	8x1k	1
SIP Widerstandsleiste	8x100k	1
Widerstand	22E	1
Kondensator	100n	5
Kondensator	100 $\mu$	1
Stiftleiste	5pol	1
Stiftleiste	6pol	1

### A.3 Vorverstärkerschaltung



## A.4 Costas Loop Analogteil





## A.5 Costas Loop Digitalteil



# Anhang B

## Schaltungstechnische Modifikationen

In der Schaltung des I/Q - Mischers, zu finden in [1] , Absatz A.5, werden vier Bauteile modifiziert. Die Tabelle zeigt die Bauteilnummern und deren neue Werte:

Bezeichnung	Wert
Drossel L19	$47\mu$
Drossel L20	$47\mu$
Kondensator C3	$1\mu$
Kondensator C4	$1\mu$

Auf der Schnittstellenkarte wurden ebenfalls zwei Bauteilwerte verändert:

Bezeichnung	Wert
Kondensator C33	22n
Kondensator C22	10n

# Anhang C

## Anleitung zur Bitfehlermessung

Der Bitfehlermessplatz *Anritsu ME520B* besteht aus einem Sendermodul und einem Empfängermodul. Für die Bitfehlermessung sind folgende Einstellungen am Sendegerät wesentlich (vergleiche auch [4] Seite 3-66):

Bitrate	EXT
Format	NRZ
Pattern	$2^{15} - 1$
Error Addition	BIT - SINGLE
Clock Output	Level TTL
NRZ Output	Level TTL
Freq. Offset	OFF

Am Empfängergerät:

Gating Period	TIME
Error	Mode BIT - Sync AUTO - Display RATE
Bit Rate	EXT
Format	NRZ
Pattern	$2^{15} - 1$
Clock Input	AUTO
NRZ Input	MAN

Auf der Rückseite des Sendergeräts befindet sich der Eingang EXT Clock. Hier koppelt man den Sendetakt ein, den man aus der kleinen Platine gewinnt, die über das Flachbandkabel mit der Scramblerplatine verbunden ist. Der NRZ Ausgang des Senders liefert nun das Sendesignal. Dieses wird direkt an Pin Nummer 5 des 9-poligen Sub-D Steckers am Funkmodem angeschlossen. Damit wird der Scrambler auf der Scramblerplatine umgangen.

Empfangsseitig schließt man Pin Nummer 3 des 9-poligen Sub-D Steckers am Funkmodem an den NRZ Eingang des Empfängergeräts. Zuletzt verbindet man noch den regenerierten Takt, den man von der blauen Bitfehlermeßplatine (gelbes Verbindungskabel) bezieht, mit dem CLOCK Eingang des Empfängergeräts.

Für die ersten Versuche sollte man mit der minimalen *Gating Time* (1 Sekunde) beginnen. Durch Betätigen der Taste START/STOP beginnt dann die eigentliche Messung.

Während der Messung ist wichtig, daß die beiden rechten LEDs SYNC LOSS und NO SIGNAL am Empfängergerät nicht aktiv sind. Falls doch, wird empfohlen, durch Änderung an der Eichleitung ein neuerliches Einrasten der Costas Loop zu erzwingen.

# Anhang D

## Installation von Linux

Dieser Abschnitt beschränkt sich auf die Linux Distribution *Redhat 8.0* und geht ausschließlich auf die für die Funktionalität als Funkgateway benötigten Komponenten ein. Für eine exakte Installationsanleitung verweise ich auf [8]. Die erste CD-ROM des Paketes ist bootfähig. Durch Eingabe von `linux text` startet man die Installation im Textmodus. Aus Gründen der Vergleichbarkeit mit englischsprachigen Texten empfehle ich die englischsprachige Installation. Beim nächsten Dialog sollte man dann aber doch den deutschen Tastatortreiber `de-latin1` auswählen. Bei der Auswahl des Installationsumfangs wird `Custom` gewählt. Für die Partitionsaufteilung sei an dieser Stelle empfohlen:

Device	Size	Filesystem	Mount Point
<code>/dev/sda1</code>	80M	ext3	<code>/boot</code>
<code>/dev/sda2</code>	128M	swap	
<code>/dev/sda3</code>	1750M	ext3	<code>/</code>

Im nächsten Dialog wählt man den Bootmanager aus. Die hier beschriebene Installation basiert auf dem Bootmanager LIL0, der in den *MBR* (Master Boot Record) installiert wird. Danach wird automatisch ein Eintrag für das Bootimage `/dev/sda3` erstellt. Im Zuge der Installation sollte man noch keinerlei Firewall Konfiguration aktivieren, und daher im nächsten Dialog die Option `[*] no Firewall` auswählen. Der nächste Installationsschritt besteht aus der Abfrage des *ROOT* Passwortes. Das ist das Passwort des *Superusers* und sollte so gewählt werden, daß es einem Angriff aus dem Internet durch gezieltes Raten standhält. Zumindest einer der beiden Funkgateways wird permanent mit dem Internet in Verbindung stehen, und ist demnach potentiellen Angreifern ausgesetzt. Da das *Superuser*-Benutzerkonto nur für administrative Tätigkeiten am System benutzt werden sollte, empfiehlt es sich, am darauffolgenden Bildschirm einen Arbeitsuser anzulegen.

Als nächstes werden die Optionen für die Authentifizierung (Authentication Configuration) abgefragt. Die Vorschläge am Bildschirm gilt es zu übernehmen.

- [\*] Use Shadow Passwords
- [\*] Enable MD5 Passwords

Linux Distributionen werden üblicherweise im Umfang von mehreren CD-ROMs ausgeliefert. Zuletzt wählt man daher die Funktionen aus, die man für diesen Computer tatsächlich benötigt. Hinter jedem Eintrag verbirgt sich eine Fülle von Einzelpaketen, die über die Option `[*]select individual packages` auch gezielt an- bzw. abgewählt werden können. Für die Grundkonfiguration des Funkgateways genügt es aber, wenn man unter `Package Group Selection` folgende Auswahl trifft:

- [\*] Administration Tools
- [\*] Graphical Internet
- [\*] KDE Desktop Environment
- [\*] Kernel Development
- [\*] Server Configuration Tools
- [\*] System Tools
- [\*] Text-based Internet
- [\*] X-Window System

- [ ] Select Individual packages

Der nun folgende Installationsvorgang ist begleitet von dem ein oder anderen Wechsel des Datenträgers und sollte nach rund einer Stunde abgeschlossen sein.

Der erste Startvorgang des Linux Systems kann, je nach eingesetzten Hardware Komponenten, einen manuellen Eingriff erfordern. Die Plug and Play Funktionalität des Betriebssystems findet beispielsweise viele Netzwerk- und Soundkarten automatisch und bindet sie korrekt ein. Falls ein Netzwerkadapter beim Bootvorgang erkannt wird, öffnet sich sogleich ein Dialogfeld, wo man die TCP/IP Konfiguration des Adapters durchführen kann. Je nachdem, ob es sich um das eigentliche Gateway ins Internet, oder um den Verbindungscomputer zwischen privatem Netzwerk und Funkschnittstelle handelt, wird man an dieser Stelle die geeigneten IP-Adressen angeben. Wie bereits in Kapitel 5 erwähnt, ist es für die Integration der Funkschnittstellenkarte notwendig, den Systemkern neu zu übersetzen. Diesem Vorgang widmet sich der folgende Abschnitt.

## Anhang E

# Aktualisierung der Linux Installation

Wie im Abschnitt 4.4 bereits angesprochen, wird an dieser Stelle dringend empfohlen, immer die aktuellste verfügbare Kernelversion als Basis für das Betriebssystem einzusetzen. In Österreich sei als Bezugsquelle genannt: <http://gd.tuwien.ac.at>. Der Kernel liegt üblicherweise als gepacktes *.tar* Archiv vor. Wichtig ist nun, daß man dieses Archiv in das Verzeichnis */usr/src/* entpackt. Dazu wechselt man in dieses Verzeichnis und startet den Vorgang durch folgenden Befehl:

```
tar xvfz <dateiname>.tar.gz
```

Dieses Kommando erzeugt ein Verzeichnis namens */usr/src/<dateiname>*. Falls es notwendig ist, den hier vorliegenden Kernel nachträglich zu modifizieren (*Kernelpatch*), dann würde man jetzt in das neu angelegte Verzeichnis */usr/src/<dateiname>* wechseln, und einen mitgelieferten *Patch* über

```
patch -p1 < <dateiname>.tar.gz
```

einspielen. Der Systemkern und alle integrierten Module liegen nun im Quelltext vor. Als nächsten Schritt gilt es, eine Konfigurationsdatei zu erstellen, anhand derer der Systemkern dann vollautomatisch in Maschinsprache übersetzt wird. Diese Datei heißt *.config* und befindet sich im neu erstellten Quellcode Verzeichnis. Um sicherzustellen, daß sich die komplette Arbeitsumgebung in einem definierten Ausgangszustand befindet, wechselt man in das Quellcode Verzeichnis und setzt anschließend folgenden Befehl ab:

```
make mrproper
```

Falls die Installation des Linuxgesamtpaketes nach Anhang D fehlerfrei verlaufen ist, dann befindet sich der Quellcode des zur Zeit aktiven Systemkerns im Verzeichnis */usr/src/linux-2.4*. Durch den Befehl

```
cp /usr/src/linux-2.4/.config .
```

kann man sich die Konfigurationsdatei des aktiven Kernels, deren fehlerfreie Funktion durch den Installationsvorgang sicher gestellt wurde, in das neue Quellcode Verzeichnis holen. Anschließend modifiziert man die übernommene Datei mit

```
make menuconfig
```

Für die Funktion als Funkgateway findet man in dem menugeführten Programm unter der Option *Amateur Radio Support* die betreffenden Einträge. Die korrekte Konfiguration muß foldermaßen aussehen:

```
[*] Amateur Radio support
--- Packet Radio protocols
<*>  Amateur Radio AX.25 Level 2 protocol
[*]   AX.25 DAMA Slave support
< >   Amateur Radio NET/ROM protocol
< >   Amateur Radio X.25 PLP (Rose)
```

Die Auswahl mittels `< * >` führt zu einer permanenten Aktivierung des AX.25 Protokolls, im Gegensatz zur Auswahl mittels `< M >`, was einer Vorbereitung des Kernels mit einer anschließend notwendiger Aktivierung durch den Befehl `modprobe` entspräche. Unter dem Menüpunkt *AX.25 network device drivers* findet sich nun der Schnittstellentreiber, der mit der Option

```
<M> High-speed (DMA) SCC driver for AX.25
```

modular eingebunden wird. Für die übrigen Konfigurationsoptionen, den Systemkern betreffend, gilt: Eine Beibehaltung aller derzeit aktiven Optionen ist möglich, kostet jedoch Ressourcen und kann den Vorgang des Übersetzens überaus stark verlängern. An dieser Stelle sei daher empfohlen, daß man Hardwaremodule, die man weder jetzt noch in naher Zukunft auf diesem Gerät brauchen wird, großzügig entfernt. Als Beispiele dafür seien genannt: WLAN Systeme, SCSI Adapter, USB Treiber, Multimedia- und Soundadapter, etc. Nun gilt es aus dem Konfigurationsprogramm auszusteigen, wobei die Kernelkonfiguration in der zuvor genannten Datei `.config` abgespeichert wird. Der Ablauf für das Übersetzen des Systemkerns lautet:

```
make dep
make clean
make bzImage
make modules
make modules_install
make install
```

Das letzte Kommando kopiert einige neu erstellte Binär-Dateien in das Verzeichnis `/boot` und kann eventuell in einer anderen Linux Distribution fehlen oder anders benannt sein. Ersatzweise kann man die Platzierung der fertigen Binär-Dateien auch per Hand erledigen. Dazu verweise ich auf [7]. Zuletzt kontrolliert man die Einträge in der Konfigurationsdatei für den Bootmanager `/etc/lilo.conf` und ergänzt sie, falls notwendig, durch folgende Zeilen:



```
image=/boot/vmlinuz-2.4.20
    label=linux2.4.20
    initrd=/boot/initrd-2.4.20.img
    read-only
    append="root=LABEL=/"
```

Die exakten Dateinamen, sowie der Name des Bootimages, müssen angepaßt werden. Jede Änderung dieser Konfigurationsdatei muß mit dem Befehl

```
lilo
```

in den *MBR* übernommen werden. Bei dem gerade eben beschriebenen Vorgang geht der alte, gerade aktive Systemkern, nicht verloren, sondern kann beim Hochfahren des Systems über den Bootmanager weiterhin angewählt werden. Diese Eigenschaft erweist sich als vorteilhaft, falls der neue Systemkern nicht auf Anhieb funktionieren sollte.

Im Folgenden ist der komplette Vorgang, der zum Einspielen eines neuen Systemkerns notwendig ist, noch einmal kurz zusammengefaßt:

- Besorgen der aktuellen Kernelversion
- Entpacken in */usr/src*
- Definierten Ausgangszustand über `make mrproper` herstellen
- Alte Kernelkonfiguration in neues Verzeichnis kopieren (*.config*)
- Neuen Kernel über `make menuconfig` konfigurieren
- Kernel übersetzen
- Bootimage erstellen
- System neu starten

Achtung! Ab der nächsten Kernelgeneration (Version 2.6) könnte ein Übernehmen der alten Kernelkonfiguration nicht mehr möglich sein. In diesem Fall muß man die Konfigurationsdatei *.config* komplett neu erstellen.

# Anhang F

## Konfigurationsscript

Das Script `iprulez` befindet sich im Verzeichnis `/usr/bin` und initialisiert die Funkschnittstelle. Im zweiten Teil werden verschiedene Paketfilter definiert, die schlußendlich eine Firewallfunktion für den Computer darstellen.

```
IPTABLES=/sbin/iptables

*****
# .Internet device
INET_DEV=eth0
# interface ip-address
INET_IP=128.131.67.177
# subnet
INET=0.0.0.0/0
*****

*****
# .LAN device
LNET_DEV=dmascc0
# local interface ip-address
LNET_IP=44.143.22.12
# local interface net-address
LNET=44.143.22.0/22
*****

case "$1" in
    start)                                # Parameter start given?

    echo "IPTABLES Rulez active!"
    route add default gw 128.131.67.1
    echo 1 > /proc/sys/net/ipv4/ip_forward
    $IPTABLES -t nat -A POSTROUTING -o eth0 -j MASQUERADE

    # First flush all old rules...
    $IPTABLES -F INPUT
    $IPTABLES -F FORWARD
    $IPTABLES -F OUTPUT

    # ...then set to D/A/A
    $IPTABLES -P INPUT DROP
    $IPTABLES -P FORWARD ACCEPT
    $IPTABLES -P OUTPUT ACCEPT

    # full access for locals
    $IPTABLES -A INPUT -i ${LNET_DEV} -s ${LNET} -d ${INET} -j ACCEPT
    $IPTABLES -A INPUT -i ${LNET_DEV} -s ${LNET} -d ${LNET} -j ACCEPT
    $IPTABLES -A INPUT -i ${INET_DEV} -s 127.0.0.1/32 -d 127.0.0.1/32 -j ACCEPT
```

```

$IPTABLES -A INPUT -i ${LNET_DEV} -s 127.0.0.1/32 -d 127.0.0.1/32 -j ACCEPT
$IPTABLES -A INPUT -i lo -j ACCEPT
$IPTABLES -A OUTPUT -o lo -j ACCEPT

# deny access for ip spoofers and log packet
$IPTABLES -A INPUT -i ${INET_DEV} -s ${LNET} -d ${INET} -j DROP

# open Input Ports for Services SSH, IPSec

$IPTABLES -A INPUT -i ${INET_DEV} -p tcp --dport 22 -s ${INET} -d ${INET_IP}/32 -j ACCEPT
$IPTABLES -A INPUT -i ${INET_DEV} -p udp --dport 500 -s ${INET} -d ${INET_IP}/32 -j ACCEPT
$IPTABLES -A INPUT -i ${INET_DEV} -p udp --dport 50 -s ${INET} -d ${INET_IP}/32 -j ACCEPT
$IPTABLES -A INPUT -i ${INET_DEV} -p tcp --dport 50 -s ${INET} -d ${INET_IP}/32 -j ACCEPT

# answers for requests to ports above 1023
$IPTABLES -A INPUT -i ${INET_DEV} -p tcp --dport 1024:65535 -s ${INET} -d ${INET_IP}/32 -j ACCEPT
$IPTABLES -A INPUT -i ${INET_DEV} -p udp --dport 1024:65535 -s ${INET} -d ${INET_IP}/32 -j ACCEPT

# allow icmp answers
$IPTABLES -A INPUT -i ${INET_DEV} -p icmp -s ${INET} -d ${INET_IP}/32 -j ACCEPT

# allow full local interface
$IPTABLES -A INPUT -i ${LNET_DEV} -j ACCEPT

;;
stop)

# Remove all Gateway and Firewall settings

$IPTABLES -F INPUT
$IPTABLES -F OUTPUT
$IPTABLES -F FORWARD
$IPTABLES -P INPUT ACCEPT
$IPTABLES -P OUTPUT ACCEPT
$IPTABLES -P FORWARD ACCEPT
;;
*)
echo "Usage: $0 {start|stop}"
exit 1
esac
exit 0

```

# Anhang G

## Timingparameter

Das Treibermodul DMASCC ist direkt nach dem Laden noch nicht aktiv. Ein eigenes Konfigurationstool namens `dmascfg` muß ausgeführt werden, bevor die Funktionsschnittstelle benutzt werden kann. `dmascfg` wird dabei über mehrere Konfigurationsparameter aufgerufen, welche neben allgemeinen Einstellungen auch einen wesentlichen Einfluß auf das Timing der Sende/Empfangsumschaltung haben. Der Aufruf folgt dabei folgendem Schema:

```
dmascfg <Interface> --<Parameter> <Wert>
```

`dmascfg` befindet sich im Verzeichnis `/usr/sbin`. Eine komplette Beschreibung der Funktionalität von `dmascfg` ist [6] zu entnehmen. Im dort verfügbaren Paket `dmascfg-1.27.tar.gz` findet sich eine sehr ausführliche Beschreibung aller Parameter in der Datei `README`. Die verwendete Konfiguration sieht folgendermaßen aus:

```
dmascfg dmascfg0 --frequency 0 --nrzi 1 --clocks 0x08 --txdelay 4
--txpause 0.5 --txtimeout 150 --txtail 0.1 --rtsoff 7 --dcdon 2
--dcdoff 7 --slottime 2 --persist 256 --waittime 2 --dma 1
```

Diese Konfiguration hat sich im Hinblick auf ihre Stabilität im Laufe der Zeit bewährt. Falls man aber die Richtfunkstrecke entgeltlich einrichtet, kann eine Optimierung in Richtung höherer Datenrate durchaus wünschenswert sein. In einem Mehrbenutzerszenario muß dem Parameter `--persist` besondere Aufmerksamkeit geschenkt werden.

# Anhang H

## Konfiguration der Schnittstellenkarte

Auswahl des Adressbereichs über folgende Jumper:

SW1-x	6	5	4	3	2	1
230H	off	on	on	on	off	off
240H	off	on	on	off	on	on
300H	off	off	on	on	on	on
310H	off	off	on	on	on	off
320H	off	off	on	on	off	on
330H	off	off	on	on	off	off
380H	off	off	off	on	on	on
3F0H	off	off	off	off	off	off

Des weiteren muß ein freier Hardware-Interrupt und ein DMA Kanal ausgewählt werden, der dann auf der Schnittstellenkarte über folgende Jumper eingestellt wird:

IRQ5	J18 1 - 2 on
IRQ7	J18 2 - 3 on
IRQ10	J14 on
IRQ11	J15 on
IRQ12	J16 on
IRQ15	J17 on
DMA0	J2, J5 on
DMA1	J3, J4 on
DMA3	J1, J6 on

# Anhang I

## Abkürzungen

AGC	Automatic Gain Control
ARP	Address Resolution Protocol
BER	Bit Error Rate
BPSK	Binary Phase Shift Keying
CRC	Cycle Redundancy Check
CSMA/CA	Carrier Sense Multiple Access / Collision Avoidance
DCD	Data Carrier Detect
DMA	Direct Memory Access
FCS	Frame Check Sequence
FEC	Forward Error Correction
ICMP	Internet Control Message Protocol
IP	Internet Protocol
ISA	Industry Standard Architecture
ISM	Industrial-Science-Medical
LNA	Low Noise Amplifier
LO	Lokaler Oszillator
LOS	Line Of Sight
MAC	Medium Access Control
MBR	Master Boot Record
MTU	Maximum Transfer Unit
NLOS	Non Line Of Sight
NRZ	Non Return to Zero
NRZI	Non Return to Zero Inverted
PLL	Phase Locked Loop
PTT	Push To Talk
RxD	Receive Data
SCC	Serial Communication Controller
SSH	Secure Shell
SSL	Secure Sockets Layer
TxD	Transmit Data

# Literaturverzeichnis

- [1] Walter Kunz: *Urban Area Data Network I*, Diplomarbeit November 1999
- [2] Christian Zacherl: *Urban Area Data Network II*, Diplomarbeit Juni 2000
- [3] Werner Keim: *Mikrowellendatenfunkstrecke mit Netzwerkanbindung*, Diplomarbeit September 2001
- [4] Anritsu Corp.: *Digital Transmission Analyzer ME520B Operation Manual*, 1989
- [5] Franz Hlawatsch: *Analog and Digital Communication Techniques*, Oktober 2000
- [6] Klaus Kudielka: <http://cacophonix.nt.tuwien.ac.at/~oe1kib/Linux/>
- [7] Alavoor Vasudevan: *The Linux Kernel HOWTO*, v5.4, 11 April 2003
- [8] RedHat Inc.: *RedHat Linux 8.0 Manual*, 2002
- [9] Marko Kovacevic: <ftp://lea.hamradio.si/pub/s5scc/>
- [10] Matjaz Vidmar: <http://lea.hamradio.si/~s53mv/bpskdem/bpskdem.html>
- [11] Roland Best: *Theorie und Anwendungen des Phase-locked Loops*, 1993
- [12] Ernst Bonek, Gottfried Magerl: *Wellenausbreitung 2*, März 1999
- [13] Hans Weinrichter, Franz Hlawatsch: *Stochastische Grundlagen Nachrichtentechnischer Signale*, 1991