

Soundmodem NG – Aktuelle Entwicklungen

Thomas Sailer, HB2JNX/HB9JNX/AE4WA

1 Einleitung

Um es gleich vorwegzuschicken, der in diesem Artikel beschriebenen Code existiert zwar grösstenteils, es ist aber noch eine Menge Feinarbeit vonnöten, bevor an eine Verteilung an Endanwender zu denken ist.

Soundmodem feiert dieses Jahr das fünfjährige Jubiläum [5]. Zu dieser Zeit war mein Entwicklungsrechner ein mit 66 MHz getakteter 80486, PCI war kaum geboren. Um die CPU-Belastung trotz Prozessoren mit langsamer Multiplikation in erträglichem Rahmen zu halten, mussten damals einige Designentscheidungen zu Lasten der Robustheit gemacht werden. Heute ist kaum mehr ein PC-Prozessor unter 500 MHz erhältlich, heutige Prozessoren sind superskalar, verfügen über schnelle Multiplier, Floating Point Execution Units, und Multimediainstruktionen. DOS hat seither zugunsten von Windows an Bedeutung verloren. Zeit also, das damalige Design neu zu überdenken.

2 Audio Ein- und Ausgabe

Unter DOS gibt es keine standardisierte Schnittstelle zu Soundkarten. Auch unter Windows gab es vor fünf Jahren noch nichts für diesen Zweck brauchbares. Andererseits waren bis auf wenige seltenere Ausnahmen alle Soundkarten registerkompatibel zum SoundBlaster oder WSS-Standard. Daher enthielt SoundModem einen eigenen Soundkartentreiber. Heutzutage sind PCI-Soundkarten Stand der Technik. Eine einheitliche Registerschnittstelle gibt es nicht mehr. Soundkartentreiber moderner Betriebssysteme bieten niedrige Latenz und sind somit für Soundmodem geeignet. Um nun nicht für jede Karte einen eigenen Treiber schreiben zu müssen, muss der Betriebssystemtreiber benutzt werden.

Die bisherige Lösung musste im Kernel-Modus arbeiten (VXD unter Windows9x, Kernel-Modul unter Linux), da sie Zugriff auf die Soundkartenhardware benötigte. Wird nun der Soundtreiber des Betriebssystems benutzt, ist dies nicht mehr nötig, es reicht ein Userspace-Programm, welches einfacher zu programmieren und debuggen ist, und welches FPU und Multimedia-Einheiten des Prozessors ohne Einschränkungen benutzen kann.

2.1 Linux

Die Linux-Soundtreiberschnittstelle bietet schon seit langem kleine Latenzzeiten, `read` liefert alle bis zum Aufrufzeitpunkt eingetroffenen Samples, und das Aufweckintervall lässt sich über die Fragmentgrösse steuern.

2.2 Windows

Mit DirectX 7 gibt es auch unter Windows eine entsprechende Schnittstelle. Während ein DirectSound Ausgabetreiber heute zum Pflichtlieferumfang jeder Soundkarte zählt, da dieser für aktuelle Spiele unerlässlich ist, sieht es im Falle von DirectSoundInput düsterer aus. Viele Soundkarten liefern noch keinen DirectSoundInput-Treiber mit und benutzen den Microsoft Legacy Treiber, der das DirectSoundInput-API auf das betagte MMSYSTEM-API abbildet. Latenzen von bis gegen 100ms [4] sind die Folge.

3 Modems

Es wäre wünschenswert, Demodulatoren für verschiedene Betriebsarten an denselben Audiostrom ankoppeln zu können, z.B. für Multimode-Zugänge. Die bisherigen Demodulatoren waren aber für eine bestimmte Samplingrate konstruiert.

Die neuen Demodulatoren lassen sich frei parametrieren. Sie berechnen die nötigen Filterparameter usw. beim Start. Jeder Demodulator berechnet die minimal nötige Samplingrate, gewählt wird dann die nächst grössere, durch die Soundkartenhardware unterstützte, Rate.

Die minimale Samplingrate des AFSK-Modems beträgt derzeit das Achtfache der Bitrate, das FSK-Modem das Anderthalbfache der Bitrate, womit mit einer 44.1kHz-fähigen Soundkarte maximal 28800Bit/s decodiert werden können.

Jeder Demodulator besitzt seinen eigenen Thread, während sich die Modulatoren eines Kanals einen Thread teilen, da sie sich ja gegenseitig ausschliessen.

Es sind auch neue Modems hinzugekommen.

3.1 Robuster FSK-Demodulator

Neben einem konventionellen FSK-Demodulator gibt es auch einen robusteren, rechenaufwändigeren Demodulator. Dieser Demodulator modelliert den Kanal als FIR-Filter mit 3 Taps plus DC-Offset. Ein Decision-Directed (DD) Estimator liefert Schätzwerte für diese Parameter, und ein Maximum Likelihood Sequence Detector (MLSD), implementiert als Viterbi Algorithmus (VA), decodiert die Datenbits. Der Gewinn beträgt, je nach Wahrscheinlichkeitsdichte des Rauschens, 1–2dB.

3.2 6400Bit/s Modem für Handies

Die Übertragungsfunktion eines Handys fällt unterhalb von etwa 500 Hz deutlich ab. Es muss daher ein Modulationsverfahren gewählt werden, welches keine spektralen Anteile mehr um DC herum hat. Oberhalb von 3kHz ist der Abfall ebenfalls deutlich.

Das Modem verwendet deshalb 8PSK mit einer Symbolrate von 2400 Baud und einer Trägerfrequenz von 1800 Hz. Um die Symbole zuverlässig decodieren zu können ist es unerlässlich, den Übertragungskanal (im Wesentlichen die Audiofilter der Handies) auszumessen und zu korrigieren.

Der Demodulator modelliert den Kanal als FIR-Filter mit 3 Taps. Weil der Kanal abhängig vom Handymodell ist, und auf einem Userzugang die Durchgänge einzelner Funkgeräte sehr kurz sein können, ist es wichtig, dass die Schätzung der Kanalparameter (der 3 FIR Taps) sehr schnell von Statten geht.

Während [3] sich um die Schätzung der Kanalparameter drückt und vom User einen Abgleich verlangt, verwendet [1] ein Decision-Directed (DD) Verfahren. Dieses verwendet zwar alle Symbole zur Übertragung von Nutzdaten, hat aber unter Umständen lange, bis Daten übertragen werden können. Dies spielt aber bei einem Vollduplex-Link eine untergeordnete Rolle.

Dieses Modem teilt die Sendedaten in Blöcke zu 128 Symbolen (entsprechen 384 Bits oder 48 Bytes) auf. Zwischen zwei Blöcken wird ein Trainingssymbolblock mit 16 Trainingssymbolen gesendet. Durch das Einbetten von dem Empfänger bekannten Datensymbolen reduziert sich zwar die Nutzdatenrate, die Kanalschätzung wird jedoch wesentlich vereinfacht und beschleunigt, da die Anzahl der Unbekannten drastisch reduziert wird. Der Empfänger braucht nun nicht mehr Daten und Kanalparameter gleichzeitig zu schätzen, sondern nur noch die Kanalparameter (Data-Aided, DA). Anhang A leitet einen Maximum Likelihood (ML) Schätzer für die Kanalparameter eines AWGN Kanales her [2]. Die Trainingssymbole werden ebenfalls zur Zeitsynchronisation und zur Schätzung der Samplingrate des Senders benutzt, welche die Baudrate und die Trägerfrequenz bestimmt.

Auch dieses Modem verwendet den Viterbi Algorithmus (VA) als Maximum Likelihood Sequence Detector (MLSD). Der VA besteht hier aber aus 64 Knoten, und insgesamt 512 Branches müssen pro Datensym-

bol berechnet werden. Dieser enorme Rechenaufwand lässt sich glücklicherweise gut mit MMX¹ oder VIS² bewältigen. In Falle des Cyrix 6x86MX liess sich sogar ein Speedup von 10 mit der MMX-Implementation gegenüber der reinen Integer-Implementation erreichen, obwohl MMX nur zwei Branches parallel berechnen kann. Ein Grund für diesen überproportionalen Speedup liegt in der Elimination eines unvorhersagbaren bedingten Sprunges bei der Auswahl des überlebenden Branches. MMX implementiert diese Operation mittels AND-Masken, d.h. ohne Änderung des Kontrollflusses. Damit wird diese Modulationsart auf 166 MHz-Prozessoren wie dem UltraSparc I praktikabel.

Derzeit verwendet das Modem noch keine FEC, sondern sendet direkt einen HDLC-Bitstrom.

Das Modem wurde mit zwei Standard Handies, einem C550 und einem C701 "in der Luft" getestet. Dies bietet zwar noch keine Garantie, dass es auch mit anderen Handies funktioniert, die Chancen dazu sind aber gut. Eine Untersuchung von DF91C [6] hat ergeben, dass Standard-Handies den Frequenzgang am stärksten beschränken. Die von Henning gemessenen Frequenzgänge decken sich gut mit meinen Messungen an den beiden Handies.

Ein Ärgernis der Standard-Handies ist, dass sie das eingebaute Mikrofon nicht abschalten, auch wenn eine externe Signalquelle angeschlossen wird. Damit sendet das Handy Umgebungsgeräusche mit. Dies ist schon bei 1200 Baud AFSK ein Problem, wird aber durch die schnellere Übertragung noch verschärft.

4 Diagnoseapplikation

Eine Diagnoseapplikation wird es nicht mehr geben. Die Darstellung z.B. eines Augendiagrammes bedeutet einen erheblichen zusätzlichen Rechenaufwand. Jedoch sollen einige Kennwerte des Demodulators ausgegeben werden, wie z.B. der Mean Square Error ($S/(N+D)$) oder die geschätzten Kanalparameter.

5 Zusammenfassung und Ausblick

SoundModem wird modernisiert und an die aktuelle Situation angepasst. Leistungsstärkere Prozessoren erlauben robustere Modems und neue Modulationsarten, das Aufsetzen auf Betriebssystem-Soundkartentreiber macht SoundModem auf allen neuen Soundkarten, welche vom Betriebssystem unterstützt werden, lauffähig, und Modems werden parametrierbar.

A Data Aided (DA) Maximum Likelihood (ML) Estimator für Kanal und SNR

$$\begin{aligned}
 h &= [h_1, h_2, \dots, h_k]^T && k \times 1 && \text{Kanalfilterkoeffizienten} \\
 r &= [r_1, r_2, \dots, r_n]^T && n \times 1 && \text{Empfangene Symbole} \\
 \mathbf{A} &= \begin{pmatrix} a_1 & a_2 & \cdots & a_k \\ a_2 & a_3 & \cdots & a_{k+1} \\ \vdots & \vdots & \ddots & \vdots \\ a_n & a_{n+1} & \cdots & a_{n+k-1} \end{pmatrix} && n \times k && \text{Matrix der Trainingssymbole} \\
 \Lambda &&& n \times n && \text{Kovarianzmatrix des Rauschens}
 \end{aligned}$$

¹MultiMedia eXtension des IA32-Instruktionssatzes

²Visual Instruction Set der UltraSparc-Prozessoren von Sun Microsystems

$$r = \mathbf{A}h + n \quad (1)$$

$$\Lambda = \sigma^2 I_n \quad \Lambda^{-1} = \frac{1}{\sigma^2} I_n \quad (2)$$

$$p(r|\sigma, h) = (2\pi)^{-\frac{n}{2}} \det(\Lambda)^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(r - \mathbf{A}h)^H \Lambda^{-1} (r - \mathbf{A}h)\right) \quad (3)$$

$$(r - \mathbf{A}h)^H (r - \mathbf{A}h) = r^H r - h^H \mathbf{A}^H r - r^H \mathbf{A} h + h^H \mathbf{A}^H \mathbf{A} h \quad (4)$$

$$p(r|\sigma, h) = (2\pi)^{-\frac{n}{2}} \sigma^{-n} \exp\left(-\frac{1}{2\sigma^2}(r^H r - h^H \mathbf{A}^H r - r^H \mathbf{A} h + h^H \mathbf{A}^H \mathbf{A} h)\right) \quad (5)$$

$$\ln p(r|\sigma, h) = -\frac{n}{2} \ln(2\pi) - n \ln \sigma - \frac{1}{2\sigma^2}(r^H r - h^H \mathbf{A}^H r - r^H \mathbf{A} h + h^H \mathbf{A}^H \mathbf{A} h) \quad (6)$$

$$\frac{\partial}{\partial h} \ln p(r|\sigma, h) = -r^H \mathbf{A} + h^H \mathbf{A}^H \mathbf{A} = 0 \quad (7)$$

$$\mathbf{A}^H \mathbf{A} h = \mathbf{A}^H r \quad (8)$$

$$h = (\mathbf{A}^H \mathbf{A})^{-1} \mathbf{A}^H r \quad (9)$$

$$\frac{\partial}{\partial \sigma} \ln p(r|\sigma, h) = -\frac{n}{\sigma} + \frac{1}{\sigma^3}(r - \mathbf{A}h)^H (r - \mathbf{A}h) = 0 \quad (10)$$

$$\sigma^2 = \frac{1}{n}(r - \mathbf{A}h)^H (r - \mathbf{A}h) \quad (11)$$

Literatur

- [1] Alexander Kurpiers, DL8AAU. 4-FSK für Vollduplex-Linkstrecken – Realisierung mit EZKIT lite. *Adacom Magazin*, (12):pp. 12–22, 1998.
- [2] Stefan A. Fechtel Heinrich Meyr, Marc Moeneclaey. *Digital Communication Receivers, Synchronization, Channel Estimation and Signal Processing*. John Wiley & Sons, Inc, 1997.
- [3] Marten Güttner, DL8MDV. 4-FSK – eine Alternative zu 2-FSK? *Adacom Magazin*, (11):pp. 29–33, 1998.
- [4] Men Muheim. persönliche Kommunikation, 2000.
- [5] Thomas Sailer, HB9JNX. FlexNet-Workshop. 1995.
- [6] Wolf-Henning Rech, DF9IC. Equalizer für 1200 Baud. *Adacom Magazin*, (7):pp. 71–73, 1994.