

Ein Universalmodem für den USB Bus

Thomas Sailer, HB2JNX/HB9JNX/AE4WA,
und Johannes Kneip, DG3RBU

14. März 2000

1 Einleitung

Vor etwa fünf Jahren entstanden konkrete Pläne und Baugruppen [5] für schnelle Userinstiege mit 76.8kBit/s. Damit entstand auch das Bedürfnis nach günstigen und schnelleren Modems als die bisher üblichen. Damals war der erweiterte Parallelport die einzige Schnittstelle, die weit verbreitet war, genügend Bandbreite ermöglichte und ausserdem einfach anzusteuern war. Dies führte zur Entwicklung des Modemadapters für den Enhanced Parallel Port (EPP) [11]. Doch diese Lösung benötigte ein externes Modem (z.B. [10, 4]) und war deshalb immer noch aufwändiger als gewünscht. Ein Redesign [8, 7] integrierte das Modem und erzielte dadurch eine Kostenersparnis, ausserdem gewann das Design durch die Verwendung moderner Bausteine (FPGA) an Flexibilität. Damit beherrscht das Design neben FSK auch AFSK [9].

Seither hat sich der Universelle Serielle Bus (USB) als schnelle Universalschnittstelle des PCs etabliert. Er bietet einige Vorteile gegenüber dem EPP-Port:

- Schnelle Übertragung, 12MBit/s
- Serielle Datenübertragung, d.h. dünnere und billigere Kabel
- Mittels Hubs können sich bis zu 127 Geräte einen Port teilen
- Exakte Spezifikation, daher weniger Kompatibilitätsprobleme

Ausserdem enthalten die wichtigsten Betriebssysteme wie Windows98, Linux, FreeBSD und NetBSD entweder bereits USB-Treiber, oder dann befinden sie sich in einem fortgeschrittenen Entwicklungsstadium. Es steht daher nichts mehr im Wege, diese Schnittstelle für ein Amateurfunkmodem zu benutzen.

2 Entwicklungsziele

Das Ziel der Entwicklung war es, ein möglichst vielseitiges Gerät zu entwickeln, ohne jedoch die Kosten aus den Augen zu verlieren.

Das Design unterstützt folgende Betriebsarten:

1. FSK (G3RUH kompatibel), 9.6 kBit/s bis etwa 300 kBit/s
2. 1200 Baud AFSK
3. Externes Modem mittels des standardisierten Modem Disconnect interfaces [10]
4. Audio IO

Punkt 1 deckt dieselben Applikationen wie das EPP Modem [7, 8] ab. Punkt 2 macht diesen Adapter für 1200 Baud User attraktiv, da er bei einem späteren Umstieg auf höhere Geschwindigkeiten keine verlorene Investition darstellt. Punkt 3 erlaubt es, Modems für spezielle Modulationsarten zu verwenden, und Punkt 4 ermöglicht es, einen FM-Transceiver per Software fernzusteuern.

Wird der Modem Disconnect Stecker nicht benutzt, können seine Pins als freiprogrammierbare I/O-Ports benutzt werden.

Die Schaltung enthält auch einen Steuerport für den T7F [3] transceiver. Der T7F ist ein beliebter 9600 Baud FSK tauglicher Synthesizertransceiver für das 70cm Band. Ein A/D-Wandler misst das RSSI-Signal und macht damit die Signalfeldstärke dem PC zugänglich, vier digitale Signale ersetzen den Kanalwahlschalter, ein digitales Signal wählt zwischen 12.5kHz und 25kHz Kanalaraster und eine 1200 Baud asynchrone UART ermöglicht die Programmierung des Kanalspeichers des T7F.

3 Die Schaltung

Abbildung 1 zeigt das Schema des Prototypen. Es werden sich aber sehr wahrscheinlich noch Änderungen in der endgültigen Version ergeben, sodaß die endgültigen Treiberversionen unter Umständen mit dieser Schaltung nicht mehr zusammenarbeiten werden.

Das Design basiert zum einem grossen Teil auf dem bekannten EPP-Design [8, 9, 7]. Die Schemas beider Modems sehen daher auf den ersten Blick sehr ähnlich aus, es wurden jedoch einige Detailverbesserungen vorgenommen.

3.1 Auswahl eines Microcontrollers

Die USB Spezifikation verlangt von den Geräten, daß sie in der Lage sind, relativ komplizierte Abfragen abzu- arbeiten. Es wäre zwar grundsätzlich möglich, auch dies mit einem FPGA¹ zu erledigen, dies würde jedoch ein grosses und damit teures FPGA erfordern, sodaßes günstiger erscheint, einen Mikrocontroller mit eingebautem USB-Interface dafür zu verwenden. Im April 1999 haben wir uns daher auf dem Markt nach geeigneten Bausteinen mit full speed (12 MBit/s) USB interface umgesehen (Tabelle 1).

Hersteller	Baustein	CPU Kern	Features	Probleme
AMD	Am186CC	8086	HDLC	\$
AnchorChips	AN2131	8051	8kB SRAM, benötigt keinen nichtflüchtigen Speicher	langsamer Mikrocontrollerkern
Cypress	7C64213, 7C64313	proprietär	8kB EPROM 256B SRAM, 1kB FIFO SRAM, master DMA ins FIFO SRAM	noch nicht erhältlich, einfacher Microcontrollerkern
Infineon	SAB- C541U	8051	8kB ROM, 256B SRAM	sehr langsamer Microcontrollerkern, zu wenig SRAM
Motorola	MPC823	PowerPC	HDLC, ethernet	\$\$, BGA Gehäuse

Tabelle 1: Marktübersicht Mikrocontroller mit USB Interface, April 1999

Auf den ersten Blick erscheint der AMD 186CC als ideal für diesen Zweck, er ist jedoch recht teuer und weil er keinen On-Chip Speicher enthält, benötigt er noch externen FLASH und SRAM Speicher, was die Schaltung noch zusätzlich verteuert. Für einen kleinen Knotenrechner aber mit drei Funkports und einem USB-Port (z.B. als Mailbox Frontend Digi) ist er sehr geeignet, es existiert auch schon ein Prototypendesign dafür.

¹“Field Programmable Gate Array”: programmierbare Logik [2]

3.1 Auswahl eines Microcontrollers

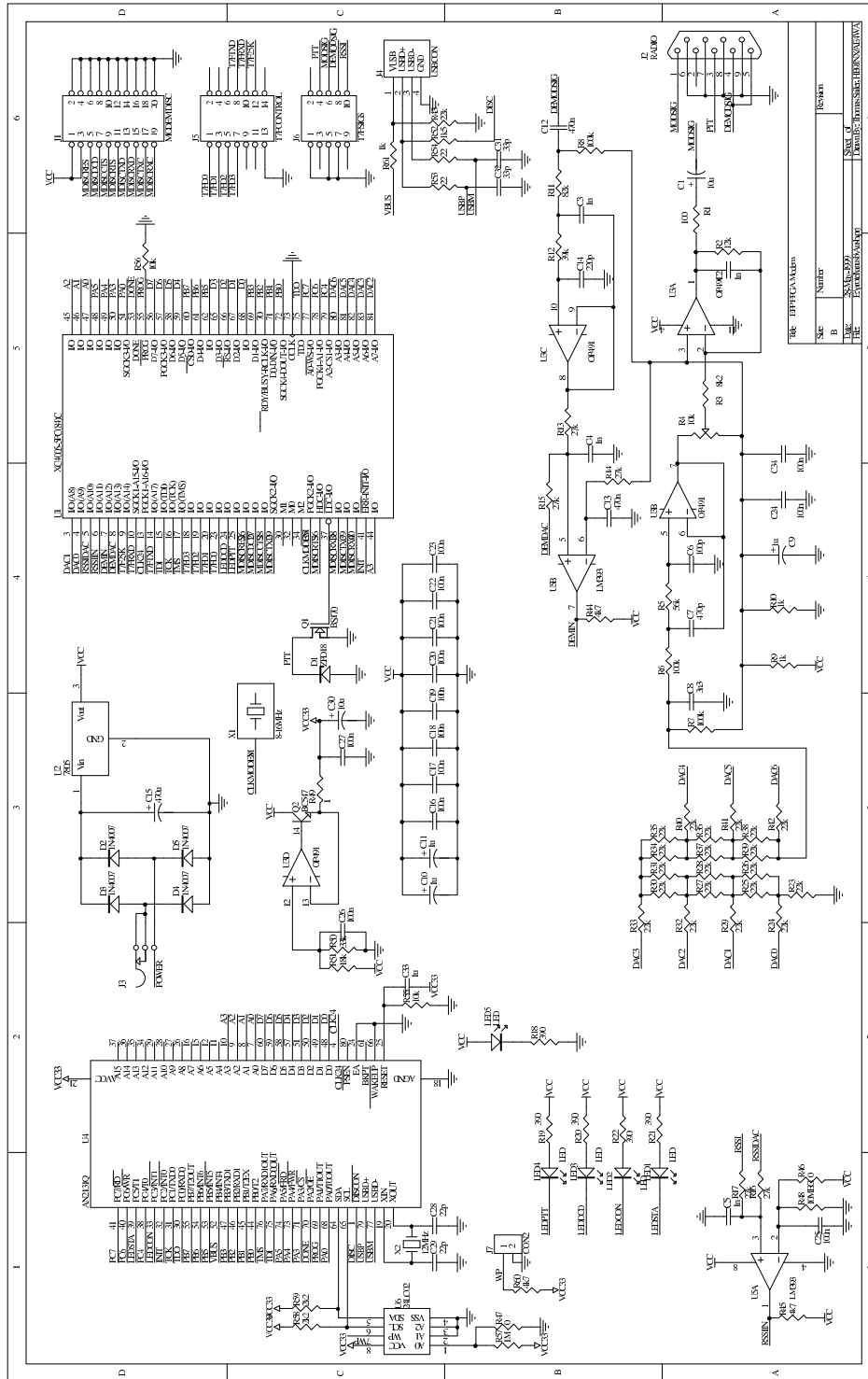


Abbildung 1: Schema des Prototypen – siehe Text

Der AnchorChips² Controller besitzt 8k SRAM on-chip. Sein herausstechendes Merkmal hingegen ist, daß die komplette Firmware für die CPU über den USB-Bus heruntergeladen werden kann. Daher braucht die CPU keinen externen nichtflüchtigen Speicher wie z.B. FLASH. Der Nachteil dieses Controllers ist aber der recht langsame CPU Kern. Er ist zwar deutlich schneller als ein Original-8051, aber externe Zugriffe sowie Zugriffe auf den USB-Controller sind sehr ineffizient in der 8051-Architektur. Alles in allem aber ist der Controller doch eine gute Wahl für diesen Zweck.

Weil der AnchorChips-Controller bereits SRAM on-chip hat und der USB-Bus recht niedrige Latenzzeiten zulässt, kann ein externes SRAM, wie es noch beim EPP-Adapter nötig war, eingespart werden.

3.2 Spannungsversorgung

Der USB-Bus kann externe Geräte mit 5V versorgen. Die Spezifikation erlaubt jedoch sehr grosse Toleranzen auf dieser Spannung, die durch resistive Verluste in den Kabeln auftreten können. Es ist daher unmöglich, 5V IC's direkt (ohne DC/DC Konverter) aus dieser Spannung zu versorgen. Es ist jedoch möglich, 3.3V IC's mit einem low drop Spannungsregler zu versorgen.

Die meisten USB-Controller sind daher als 3.3V IC's ausgelegt, so auch der AnchorChips Controller. Seine IO Pins sind jedoch 5V tolerant.

Würde der Analogteil des Modems mit nur 3.3V versorgt, dann wäre der erzielbare Spannungshub am Ausgang für die meisten Transceiver zu klein. Daher wurde entschieden, den Analogteil mit 5V zu versorgen. Um das Interface zwischen Digitalteil und Analogteil so einfach wie möglich zu halten, wurde ein 5V FPGA verwendet. Der Analogteil ist daher identisch mit demjenigen des EPP Modems [8, 9, 7].

Ein USB-Port kann Geräte mit maximal 500mA³ versorgen, dies ergibt etwa 2.5 Watt. Typische 9600 Baud Transceiver liefern aber eine HF-Ausgangsleistung von 7 Watt, benötigen daher etwa das Doppelte an Versorgungsleistung. Es ist daher unmöglich, den Transceiver aus dem USB-Bus zu speisen, man braucht also auf jeden Fall ein externes Netzgerät, welches typischerweise etwa 12V liefert. Es macht daher keinen Sinn, einen teuren DC/DC-Wandler einzusetzen, um das Modem aus dem USB-Bus speisen zu können. Es wurde daher auch für das Modem eine externe 12V-Versorgung vorgesehen.

3.3 Firmware Download

Um die eingangs erwähnte Flexibilität zu erreichen, wurden zwei programmierbare Bausteine verwendet, das FPGA und der USB-Mikrocontroller. Ausser einem kleinen I²C-EEPROM gibt es keinen nichtflüchtigen Speicher im Modem, das EEPROM speichert nur den Hersteller- und den Produktcode, damit das Design von anderen Geräten mit dem AnchorChips-Controller unterscheidbar wird.

Damit die Schaltung betriebsbereit wird, muss erst mal die Firmware für beide Bausteine heruntergeladen werden. Dies geschieht in zwei Schritten. Zuerst wird der Mikrocontroller mit Code gefüttert, dessen einziger Zweck es ist, Daten vom USB-Bus entgegenzunehmen und in den FPGA zu laden. Sobald der FPGA konfiguriert ist, wird der Controller wieder gestoppt und die eigentliche Firmware für den gewählten Betriebsmodus heruntergeladen.

Danach simuliert der Controller ein Aus- und wieder Einsteckzyklus am USB-Bus. Dies veranlasst das PC-Betriebssystem, die USB Gerätetabellen wieder einzulesen. Dem Controller erlaubt dies, einen anderen Produkt ID zurückzumelden, was das Betriebssystem wiederum veranlasst, andere Treiber zu laden.

3.4 Der FSK-Modus

Anders als beim EPP Modem, wo der HDLC-Decoder in der Treibersoftware auf dem PC realisiert wurde, tauscht das USB Modem mit dem PC bereits fertig decodierte, "rohe" Pakete aus. Der HDLC Codier- und

²mittlerweile von Cypress aufgekauft

³nur Ports von Self Powered Hubs

3.4 Der FSK-Modus

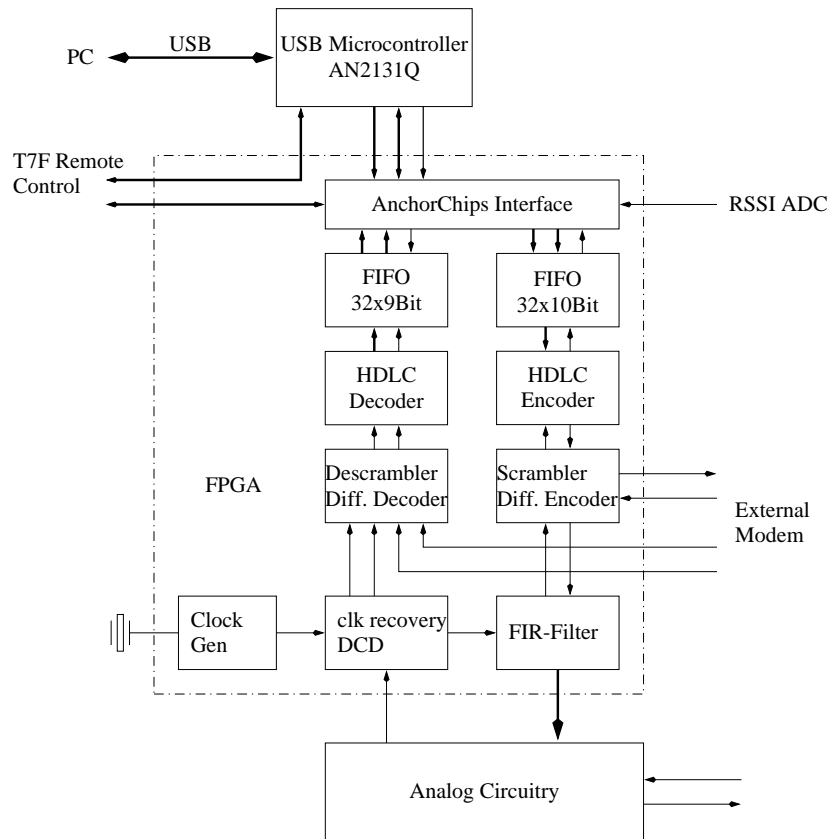


Abbildung 2: Blockdiagramm des FSK Modus

Decodierprozess findet daher auf dem Modem statt.

Die HDLC-Verarbeitungsaufgaben können in zwei Klassen eingeteilt werden: Bitverarbeitende Aufgaben wie (un)stuffing, CRC Berechnung und Flagerkennung, und Speicherverwaltungsaufgaben, wie das Speichern des gerade empfangenen Paketes und die Entscheidung am Ende bei Erhalt der CRC, ob es behalten oder verworfen werden soll. Bitweise Verarbeitung lässt sich effizient im FPGA realisieren, für die Speicherverwaltung ist der FPGA ziemlich ungeeignet.

Der Mikrocontroller kommuniziert mit dem FPGA über ein Registerinterface, welches aus 16 8bit-Registern besteht und an seinen externen (XDATA) Adress- und Datenbus angeschlossen ist.

Kleine FIFO's im FPGA entkoppeln den HDLC Encoder/Decoder und den Mikrocontroller. Die FIFO's speichern 8 Datenbits und 2 Tagbits, welche angeben, ob es sich beim zugehörigen Byte um ein Paketdatenbyte oder ein "Meta"-Byte handelt.

Das Sende-FIFO kann an vier Adressen angesprochen werden, die niederwertigsten Adressbits steuern die beiden Tagbits bei (siehe Tabelle 2).

Im Sendefall existiert nur ein Tagbit. Der HDLC-Decoder fügt immer dann ein Metabyte ein, wenn ein HDLC Flag oder Abort empfangen wurde. Das Byte enthält dann ein Bitfeld, welches angibt, ob die CRC korrekt war und wieviele "residual bits" empfangen wurden. Weil der Empfang eines HDLC-Flags für die Speicherverwaltung eine aufwändige Operation ist (das alte Paket muss entweder verworfen oder gespeichert werden, und Speicherplatz für ein neues Paket muss bereitgestellt werden), fasst der HDLC-Decoder im FPGA mehrere aufeinanderfolgende Flags zusammen.

Tag (Adresse)	Zweck
00 FSKTXDATA	Sende Datenbyte. Das Byte wird durch Stuffing und CRC Generator geschleust.
01 FSKTXCRC	Sende 8 CRC Bits. Das Datenbyte ist nicht von Bedeutung.
10 FSKTXRAW	Sende 8 "raw" Bits, d.h. ohne sie durch Stuffing und den CRC Generator zu schleusen. Damit können Flags gesendet werden.
11 FSKTXRAWCLR	Wie 10, initialisiert aber zusätzlich den CRC-Generator (alles Einsen).

Tabelle 2: HDLC encoder tags

Der Rest des FSK-Modems entspricht demjenigen im EPP Modem Adapter, welches in [8, 7] beschrieben wurde.

3.5 Der AFSK-Modus

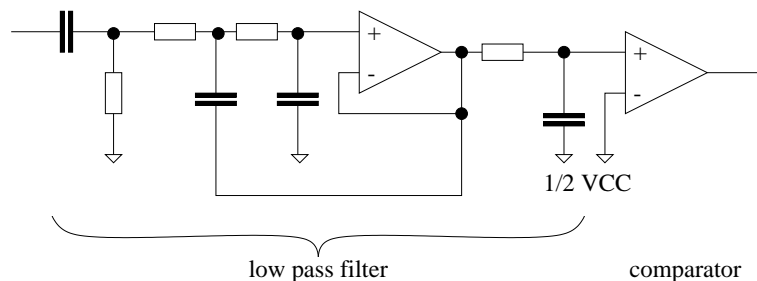


Abbildung 3: Der Analoge Schaltungsteil eines Standard FSK Modems

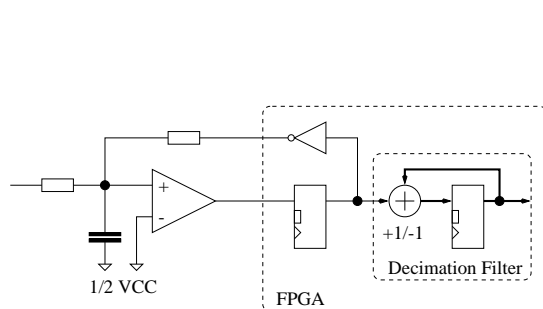
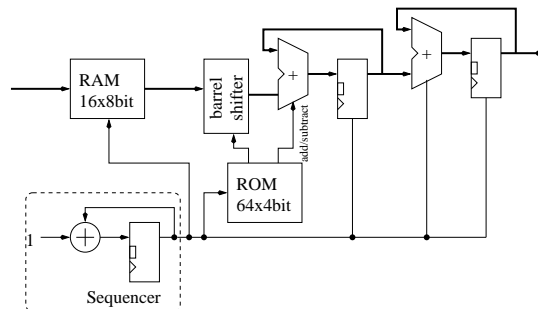
Abbildung 4: $\Delta\Sigma$ -A/D-Wandler

Abbildung 5: AFSK Demodulator

FSK-Modems, welche FPGA's enthalten, sind schon seit einiger Zeit recht populär, daher existieren auch einige "Konkurrenzdesigns", zum Beispiel das YAM [6]. Die Flexibilität der FPGA's hat nun zum Versuch geführt, mit den FSK-Modems auch AFSK demodulieren zu können. Für YAM gibt es seit etwa anderthalb Jahren eine AFSK-Firmware.

Leider wurde der Analogteil (Abbildung 3) dieser Modems aber für FSK entwickelt und ist ungeeignet für die nachfolgende AFSK-Demodulation, sodaß die Performance zu wünschen übrig lässt. Im AFSK-Modus ar-

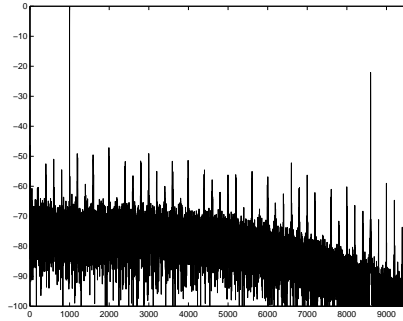


Abbildung 6: "Loopback spectrum"

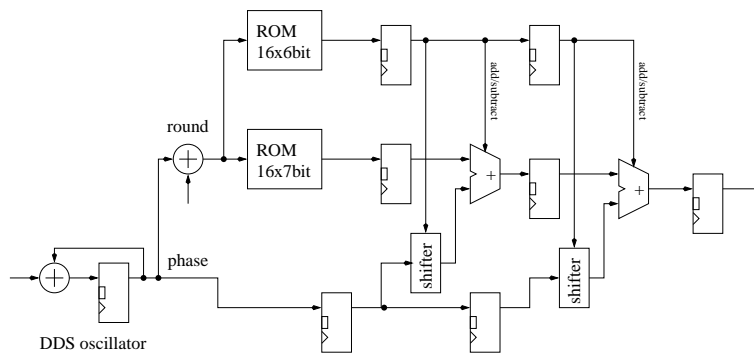


Abbildung 7: AFSK Modulator

beitet der im FSK-Modus als Slicer dienende Komparator nämlich als 1Bit-A/D-Wandler, was für eine robuste AFSK-Demodulation einfach eine zu kleine Auflösung ist.

Es zeigt sich aber, daß mit nur einem zusätzlichen Widerstand und ein wenig Logik im FPGA ein $\Delta\Sigma$ -A/D-Wandler realisiert werden kann, welcher zwar nicht High-End-Audio-Ansprüche erfüllt (dazu ist der Komparator viel zu langsam), aber dennoch bei weitem ausreicht für eine robuste AFSK-Demodulation (Abbildung 4).

Dieses Verfahren wurde bereits mit Erfolg auf dem EPP-Modemadapter implementiert und an der letztjährigen Konferenz detaillierter vorgestellt [9].

Weil die Logik des AFSK-Modems deutlich grösser ist als diejenige des FSK-Modems, andererseits aber die Datenrate sehr viel kleiner, wurde der HDLC-Decoder in Software auf dem Mikrocontroller implementiert. Der FPGA enthält nur ein doppelt gepuffertes Schieberegister, welches die rohen Bits speichert.

4 Der Analog-IO-Modus

Manchmal ist es nützlich, den Datentransceiver als ganz normalen voice FM Transceiver zu benutzen. Es ist daher möglich, das Modem einfach als A/D und D/A-Wandler zu betreiben und die Samples an den PC weiterzugeben. Auf dem PC läuft dann ein Programm, welches die Samples an eine installierte Soundkarte weiterleitet und dabei die Samplerate anpasst.

Das USB-Modem verhält sich dabei wie eine USB-"Soundkarte", es kann der mit dem Betriebssystem mitgelieferte USB Audio Klassentreiber verwendet werden. Der USB-Bus unterstützt diese Betriebsart mittels Isochroner Transfers, welche eine vereinbarte Bandbreite und Latenz garantieren, aber keine Übertragung wiederholen, falls Übertragungsfehler auf dem USB-Bus auftreten.

5 Das T7F-Interface

Die sieben Digitalsignale des T7F-Interfaces werden direkt vom FPGA gespiesen, damit bleibt die Flexibilität erhalten, dieses Interface an zukünftige Transceiver anzupassen.

Der A/D-Wandler für das RSSI-Signal funktioniert ebenfalls nach dem $\Delta\Sigma$ -Prinzip. Mittels der Widerstände am Komparator kann der Wertebereich eingestellt werden und festgelegt werden, ob der Wandler die Spannung oder den Strom (wie üblich bei den beliebten FM-Receiver-IC's) gemessen werden soll.

6 Status des Projekts

Es existiert ein funktionsfähiger Prototyp seit etwa einem halben Jahr. Der Prototyp ist jedoch in Eile entstanden, um möglichst schnell mit der Software/FPGA Entwicklung zu beginnen zu können. Die Schaltung wird daher zur Zeit gerade überarbeitet und sollte bald bei Baycom [1] bezogen werden können.

Die Treiber werden zu gegebener Zeit auf der Baycom-Webseite veröffentlicht.

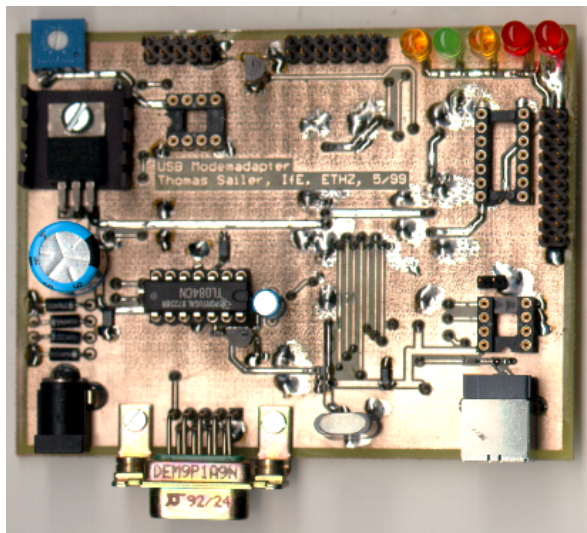


Abbildung 8: Oberseite der Prototypenplatine

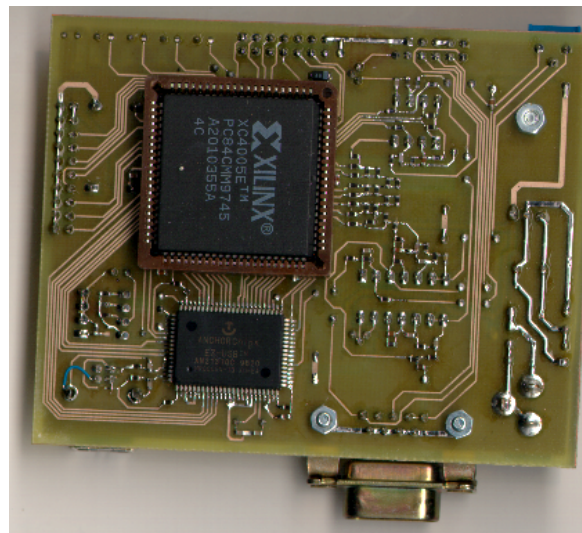


Abbildung 9: Unterseite der Prototypenplatine

7 Zusammenfassung

Dieser Artikel stellte ein vielseitig verwendbares Modem für den Amateurfunk vor, welches an den mittlerweile sehr verbreiteten USB-Bus angeschlossen werden kann.

Ein Prototyp dieses Modems wurde in am Ham Fest in Friedrichshafen ausgestellt und stieß auf einiges Publikumsinteresse.

8 Ausblick

Das Interface zur Fernsteuerung eines Transceivers stellt erst den ersten Schritt dar. Wir werden die Integration von Transceiver und Modem weiter untersuchen.

LITERATUR

Eine Schwachstelle gängiger Amateur-FSK-Transceiver ist der Modulator. Quarztransceiver verwenden oft eine Kapazitätsdiode, um den Quarz zu "ziehen", und diese Diode verursacht durch ihre Kennlinie erhebliche Nichtlinearitäten. Synthesizer-Transceiver wiederum verwenden meistens eine Zweipunktmodulation, welche die Dämpfung der tieffrequenten Anteile durch das PLL-Loopfilter ausgleichen soll. Das Modulationssignal wird durch Filter in die beiden Zweige aufgeteilt. Dieses Filter muss nun aufwändig abgeglichen werden und verursacht auch bei perfektem Abgleich noch sichtbare Verzerrungen im Augendiagramm.

Das Verwenden einer breitbandigen PLL und das Aufbringen der Modulation durch Modulieren des Teilverhältnisses des Teilers in der PLL Schleife mittels $\Delta\Sigma$ Techniken scheint hier eine einfache und vielversprechende Alternative zu sein, um die Qualität des modulierten Signales zu verbessern.

Literatur

- [1] Baycom \diamond Hard- und Software GmbH, Bert-Brecht-Weg 28, D-30890 Barsinghausen, Tel. ++49 (5105) 585 050, Fax ++49 (5105) 585 060, <http://www.baycom.org/>, Email: baycom@baycom.org.
- [2] Xilinx Corporation. The Programmable Logic Data Book. <http://www.xilinx.com>.
- [3] Holger Eckardt, DF2FQ. Handbuch zum 70cm-FM/FSK-Transceiver T7F, 1998.
- [4] Johannes Kneip, DG3RBU. Das FSK+-Modem mit Echoduplex. *Adacom Magazin 10*, 1997.
- [5] Martin Liebeck, DL2ZBN und Alexander Kurpiers, DL8AAU. Hochgeschwindigkeits-Packet-Radio – Baugruppen für das 70cm Band. *Adacom Magazin 10*, 1997.
- [6] Nico Palermo, IV3NWV. Yet Another 9k6 Modem. <http://jupiter.web-hosting.com/~nicopal/yam/>.
- [7] Thomas Sailer, HB9JNX/AE4WA and Johannes Kneip, DG3RBU. An Inexpensive PC-Modem for 76.8kBit/s User Access. In *1998 ARRL and TAPR Digital Communications Conference, Chicago, Illinois*, 1998.
- [8] Thomas Sailer, HB9JNX/AE4WA und Johannes Kneip, DG3RBU. Alternative Implementierung des EPP Adapters. In *17. Internationale Packet Radio Tagung, Darmstadt*, 1998.
- [9] Thomas Sailer, HB9JNX/AE4WA, und Johannes Kneip, DG3RBU. EPPFLEX – ein universeller Modemadapter für den EPP-Port. In *18. Internationale Packet Radio Tagung, Darmstadt*, 1999.
- [10] Wolf-Henning Rech, DF9IC. Modernes FSK-Modem – kompatibel zum Standard nach G3RUH. *Adacom Magazin 2*, 1991.
- [11] Wolf-Henning Rech, DF9IC, Johannes Kneip, DG3RBU, Gunter Jost, DK7WJ und Thomas Sailer, HB9JNX. Ein Modemadapter für den EPP. In *41. Weinheimer UKW-Tagung, Skriptum*, 1996.