

1. Die DSP-Treiber

1.1. Die Motivation

Was spricht dafür, Packet-Radio Modems mit “general purpose”, d.h. frei programmierbarer, nicht auf einen bestimmten Zweck festgelegter digitaler Signalverarbeitungshardware zu realisieren? Man bedenke, daß auch das G3RUH/DF9IC einige Konzepte der digitalen Signalverarbeitung verwendet.

- Die Flexibilität. Dieselbe Hardware kann für grundverschiedene Betriebsarten verwendet werden, so z.B. 1200 Baud AFSK Packet, 9600 Baud FSK Packet, RTTY, Amtor, Pactor.
- Der Preis. Da diese Hardware für einen anderen Zweck hergestellt wird, sind viel größere Stückzahlen und damit kleinere Preise möglich.
- Experimente mit andern Modulationsarten, anderer Codierung usw. sind möglich ohne aufwendige Hardware-Basteleien.

Die Nachteile:

- Mit dedizierter Signalverarbeitungshardware (z.B. DF9IC-Modem) können höhere Signalisierungsraten (Baudraten) erreicht werden. Die Baudrate muß nun aber nicht identisch mit der Bitrate sein, man kann ja mehrere Bits pro Symbol (pro Baud) übertragen. Dies kommt dem programmierbaren DSP entgegen.
- Der Stromverbrauch kann größer sein, obwohl DSPs im Vergleich mit andern Prozessoren viel mehr “MIPS pro Watt” liefern.

1.2. PSADRV R

Dieser Treiber unterstützt PC-Soundkarten, die das PSA-Chipset von Analog Devices enthalten. Unter anderen sind dies folgende Soundkarten: Echo Speech Corp. Personal Sound System, Cardinal DSP16, Orchid Soundwave 32, Wearness Beethoven und Adaptec AMM-1570 [1]. Diese Soundkarten enthalten einen frei programmierbaren digitalen Signalprozessor. Dies ist die beste Lösung, da sie den Host-Prozessor kaum belastet. Der Host-Rechner muss nur die empfangenen Pakete im grossen Zwischenspeicher (8kByte) abholen, und die Senderpakete in den Zwischenspeicher schreiben. Der Rest erledigt der DSP, auf Wunsch auch beides gleichzeitig (Fullduplex).

Die Programme `TPAT1200.EXE` für 1200 Baud AFSK und `TPAT9600.EXE` für 9600 Baud FSK, gemäß G3RUH, erlauben die Einstellung aller Audio-Parameter. Hier kann man die Eingangs- und Ausgangssignalpegel einstellen, aber auch das Augendiagramm darstellen und Bitfehler zählen. Beim Verlassen der Programme wird ein File (`FIRM1200.DSP` und `FIRM9600.DSP`) geschrieben, das die DSP-Firmware und alle Audio-Parameter enthält. Sollte einmal ein DSP-Firmware-Upgrade nötig werden, muß ich nur die Programme `TPAT*.EXE` zu verteilen, die man dann einmal starten muß, und schon ist die Sache erledigt.

Beim Treiberstart muß man nun die DSP-Firmware-Files angeben:

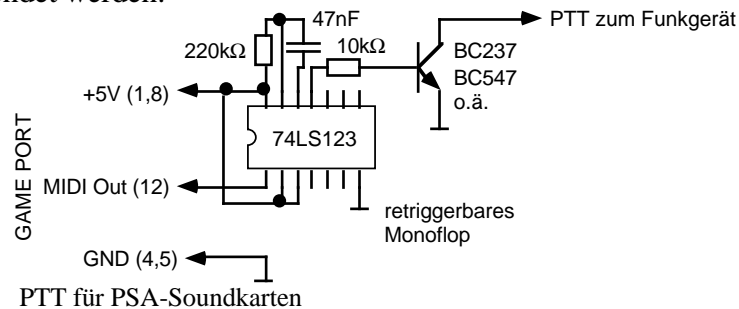
```
PSADRV R FIRM1200.DSP FIRM9600.DSP
```

Damit kann man in FlexNet beide Betriebsarten einschalten. Braucht man eine Betriebsart nicht, kann man das entsprechende Firmware-File weglassen, um Speicher zu sparen.

Das Einzige, was hier nicht ganz so elegant ist, ist die Ausgabe des PTT-Signals. Die Soundkarte hat leider keinen DC-gekoppelten Ausgang, an dem man ein digitales Signal ausgeben könnte. Hier hat man drei Möglichkeiten:

- Der `FLAG_OUT` Ausgang des DSP's wird in der Soundkarte nicht benötigt. Man kann den, gepuffert mit einem MOSFET (BS170), als PTT Signal verwenden. Das Gate des MOSFET's kommt an den `FLAG_OUT`-Ausgang des DSPs ADSP2115, beim PLCC-Gehäuse ist das Pin 52, und beim PQFP/TQFP-Gehäuse ist das Pin 31. Die Source kommt an Masse, und der Drain an das Funkgerät.

- Wem die IC-Gehäuse zu klein sind oder nicht an der Soundkarte herumlöten möchte, kann den MIDI-Ausgang der Soundkarte als PTT-Signal verwenden. Da der MIDI-Ausgang aber an einer seriellen Schnittstelle angeschlossen ist, kann er auch keinen konstanten Pegel ausgeben. Daher muß man den MIDI-Ausgang auf ein retriggerbares Monoflop mit etwa 5ms Pulsdauer geben, um ein konstantes Signal zu erzeugen. Dazu kann z.B. folgende Schaltung verwendet werden:



- Für 1200 Baud geht auch eine einfache VOX-Schaltung, zumal beide Ausgänge (links und rechts) auf verschiedene Pegel programmiert werden können. Man kann somit den einen Ausgang auf den Pegel des Funkgerätes, den anderen auf das Maximum einstellen.

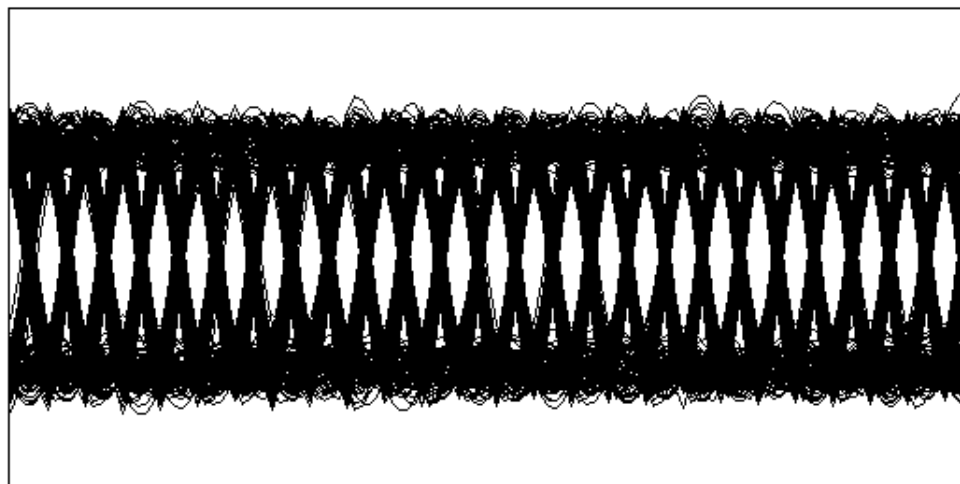
9600 Baud Test Pattern Generator - (C) 1994 by Tom Sailer HB9JNX

Tx Pattern: Random (0=Off 1=Low 2=High 3=Diddle 4=Flags 5=Random)
Left Output: -24dB (F1+ F2-) Input: 16dB (F5+ F6-) Source: Aux 1 (F9)
Right Output: -24dB (F3+ F4-) Input: 16dB (F7+ F8-) Source: Aux 1 (F10)

TxDelay: 150ms (E- R+) **TxTail: 30ms (T- Z+)** **Slottime: 100ms (U- I+)**
Half duplex (X) **P-Persistence: 40 (O- P+)**
Filter (ALT-F): Default filter (linear phase lowpass)

DCD: * **Eye diagram trigger: continuous (ALT-D)**
BER: 0

ALT-X ESC
Quit
ALT-C C c
Clear Eye
ALT-B B b
Clear BER



Das Programm TPAT9600.EXE

1.3. DSK und DG1SCR

Diese beiden Treiber sind fast identisch. Der eine ist für das DSP Starter's Kit von Texas Instruments mit dem 320C26-Prozessor (dasjenige mit dem 320C50 kann nicht verwendet werden!), der andere für die DSP-Karte von DG1SCR. Das DSP-Board wird an eine (normale) serielle Schnittstelle des PC's angeschlossen. RxD und TxD dienen der Kommunikation zwischen Host und DSP, während DTR bei der Initialisierung benötigt wird. An RTS erscheint das PTT-Signal, daher darf RTS nicht durchverbunden werden!

Die Treiber können ebenfalls 1200 Baud AFSK und 9600 Baud FSK nach G3RUH, wobei man bei 9600 Baud darauf achten muß, daß dem Empfängersignal keine DC-Komponente überlagert ist.

Diese Treiber belasten den Host-Recher deutlich mehr als der PSADRV. Der DSP übernimmt hier lediglich die Modulation und Demodulation des Signales und schickt einen seriellen Bitstrom zum PC, der daraus die Pakete fischen muß (HDLC-Codierung/Decodierung). Weiter hat die Rechenleistung nur für ein rudimentäres Sende-Filter bei 9600 Baud gereicht, man sollte daher eine analoge Nachfilterung verwenden.

1.4. WSS, WSS_FAST und WSS_9K6

Diese Treiber erlauben es, mit einer ganz normalen, WindowsSoundSystem-kompatiblen Soundkarte Packet-Radio zu machen! Mittlerweile sind fast alle Soundkarten WindowsSoundSystem-kompatibel [1], außer diejenigen von Creative Labs (SoundBlaster...). Bei diesen Treibern muß der Host-Prozessor die ganzen DSP-Aufgaben übernehmen, was ihn stark belastet. Es muß daher mindestens ein 486er sein, SX oder DX spielt keine Rolle, da der Treiber keine Fließkommaarithmetik benötigt. Der Treiber kann nur Halbduplex – er verwendet zum Datentransport einen DMA-Kanal, und der kann die Daten nur in eine Richtung schieben. Es soll zwar bei den WSS-Karten die Möglichkeit geben, zwei DMA-Kanäle zu verwenden, doch leider konnte ich, trotz intensiver Suche, keine Infos darüber finden. IRQ-Nummer und DMA-Kanal sind bei diesen Karten softwarekonfigurierbar, der Treiber stellt automatisch die angegebenen Werte ein. Man muß sich jedoch selbst vergewissern, daß die angegebenen Ressourcen nicht bereits belegt sind! Der Treiber darf nicht hochgeladen werden, da das sonst Probleme mit den DMA-Buffern gibt.

Um den Rechenaufwand in Grenzen zu halten, werden nur 8Bit-Samples verarbeitet, der Dynamikbereich des Treibers ist also nicht allzu gross.

- WSS: Die normale Version für 1200 Baud AFSK.
- WSS_FAST: Diese Version braucht weniger Rechenleistung, dafür mehr Speicher als WSS. Es wurden die meisten Multiplikationen, die es zu Hauf braucht und die auf den Intel-Prozessoren langsam sind, durch Tabellen ersetzt. Damit die Tabellen nicht ins Unendliche wachsen, wurde die Dynamik weiter reduziert. Man sollte diesen Treiber also satt aussteuern, aber keinesfalls übersteuern. Ich kann leider keine Pegel angeben, da das von der eingesetzten Karte abhängt.
- WSS_9K6: Die Version für 9600 Baud FSK. Es machte hier wenig Sinn, einen einzigen Treiber für 1200 und 9600 Baud zu schreiben, da die Tabellen, die viel Speicher brauchen, für beide Betriebsarten keine Gemeinsamkeiten aufweisen.

Zum Schluß der Treibereien noch was ganz anderes:

1.5. IPPD

Dieser Treiber wurde entwickelt, um Derivate von Phil Karn's NET (Amateur-TCP/IP-Software) über Ethernet mit PC/FlexNet zu verbinden. Der Treiber setzt auf einen Packet Driver [2] auf, den es für alle gängigen Ethernetkarten gibt. Er verwendet dazu AXIP-Encapsulation, d.h. den Transport eines AX25-Paketes in einem IP-Paket. Dadurch sind diese Pakete auch

Routing-fähig. Dem Einsatz dieses Treibers im Weitverkehr sind allerdings Grenzen gesetzt. Er unterstützt keine IP-Fragmentation (Aufteilen eines IP-Paketes in mehrere kleinere Fragmente, wenn ein Netzwerk so lange Pakete nicht transportieren kann), und versteht auch keine ICMP-Meldungen.

And now for something completely different

2. Der Hostmode-Emulator

TFEMU erlaubt es, Programme, die das TFPC- oder das DRSI-Interface unterstützen, mit PC/FlexNet zu betreiben. TFEMU ist also ein Protokollkonverter, der die Firmware-Befehle in FlexNet-Aufrufe umsetzt. TFEMU unterstützt auch die in TFPCX eingeführte Erweiterung des TFPC-Interfaces [3].

TFEMU unterstützt die meisten der von TheFirmware [4] bekannten Befehle. Einige TF-Befehle ergeben bei PC/FlexNet allerdings keinen Sinn, so z.B. kann man die Kanalzugriffsparameter wie Persistence, Frack usw. nicht einstellen, da dies PC/FlexNet automatisch tut.

TFEMU kann auch mehrmals geladen werden. Die verschiedenen Instanzen von TFEMU sprechen sich dann untereinander ab, sodaß man sie unter demselben Rufzeichen betreiben kann. Einkommende Connects füllen zuerst das erstgeladene TFEMU bis zum höchsten erlaubten Stream auf (ESC Y Befehl, natürlich nur, wenn Rufzeichen und SSID übereinstimmen). Dies ist unter anderem bei der PacketCluster-Software nützlich.

Die Existenz von TFEMU soll aber die Applikationsentwickler nicht davon abhalten, das FlexNet-Applikationsinterface direkt zu unterstützen, da die erzielbare Geschwindigkeit um Größenordnungen besser ist. Dank dem Kit von Gunter, DK7WJ, ist die Verwendung des FlexNet-Interfaces auch viel einfacher als die Handhabung des Hostmodes.

3. Literatur

- [1] c't, Magazin für Computertechnik, Februar 1995
- [2] FTP Software Inc., Russell Nelson, PC/TCP Version 1.09 Packet Driver Specification
- [3] René Stange, DG0FT, TFPCX.DOC zur Version 2.10 von TFPCX
- [4] NORD><LINK, Beschreibung der TheFirmware 2.7 Befehle