

Using a PC and a Soundcard for Popular Amateur Digital Modes

Thomas M. Sailer, HB9JNX/AE4WA
Weinbergstrasse 76
CH-8408 Winterthur
Switzerland

Abstract

Recently, standard personal computers (PCs) have become powerful enough to do serious digital signal processing (DSP) without the need for a specialized DSP coprocessor. A standard PC soundcard serves as the interface between the analog world of the radio and the digital world of the PC processor. This equipment together with an appropriate software package allows the ham to operate many popular digital modes without a TNC.

Historical Perspective

Processing digital signals with a general purpose digital computer is not new at all. But only recently personal computers (PCs) have become fast enough to do serious realtime digital signal processing. A **highend** 486 or Pentium class machine is required for these applications. Analog interfaces (soundcards) with good quality are also relatively new; early models suffered from low resolution, low sampling rates and high prices.

Amateur modes having no timing relationship between receiving and transmitting, such as RTTY, FAX, SSTV or POCSAG, can be implemented quite easily. Standard operating system drivers for the soundcard may be used, and neither huge buffering nor long latencies do hurt. Several programs handling these modes popped up recently on the internet.

Modes requiring fast switching between reception and transmission, low latencies and an exact timing, such as packet radio and the synchronous shortwave data modes like Amtor and **Pactor**, are more difficult to implement. This paper reviews the problems associated with these modes and presents possible solutions.

After experimenting with digital signal processors, I started implementing a packet radio modem for PC's running DOS in late 1994. The development platform at that time was a 486DX2/66. A first version was presented at the Darmstadt, Germany, meeting in 1995 [Sai95]. The DOS version was continually developed further, got modularized, that is, the soundcard drivers and modem code was separated. The code was ported to Linux in early 1996 [Sai96], and back to Windows95 in late 1996 [Sai97].

The soundcard packet radio software is now available free of charge for amateur radio applications. The DOS and Windows95 versions are available from the **PC/FlexNet homepage** (see below), while the Linux version is included in the standard version 2.1 .x kernel sources. Version 2.0.x users need to get the patch from the zone FTP site (see below).

Motivation

The software modem provides several advantages over the conventional approach using a TNC.

- Since there is no dedicated TNC hardware, this solution is cheap. Terminals are not widely used anymore, a PC is common in typical HAM shacks, and you will need it anyway to run those fancy graphical terminal programs. Today's PCs are shipped with soundcards already installed, furthermore soundcards go for as little as \$30. Therefore, almost all the required hardware is already available at the typical ham shack.
- Easy diagnostics. No more fiddling with oscilloscopes, just run the diagnostic application to plot the input signal, eye diagrams etc.
- Software configurable. It is easy to change the operating mode without a hardware change with only a few keystrokes. With a conventional TNC, you have to buy and install another modem.
- Compared to specialized DSP processors, PC development tools are usually much more mature; the modem code may be single-stepped, data may be logged to disk, or even multiple modems connected by a software radio channel simulator may run simultaneously on a single machine, which greatly simplifies debugging and offers new development possibilities.
- Mobile operations: a contemporary laptop computer with builtin sound hardware and a handheld transceiver is all you need to operate on the road.

Of course, the software solution also has a few disadvantages.

- It requires host CPU processing power. The requirements are quite moderate, however. A typical host CPU load is 10% of a Pentium75 (1200 Baud AFSK). The value scales nicely with CPU clock frequency. MMX is neither required nor used, since its moderate benefits hardly justify the pain to develop MMX aware applications.
- The modem software ties up the soundcard. It is however possible to operate two soundcards in a single PC.
- Audio quality of the soundcards is widely varying. The general trend seems to be the audio quality being inversely proportional to price. Cheap cards with just a single chip from Analog Devices or Crystal Semiconductors usually do fine, while the rather popular models from Creative Labs offer less quality. Their mixer chip's bass and treble controls introduce phase distortion. Because of the widely varying signal impedances, levels and connectors it is impossible to draw a generic wiring diagram to the radio.
- It is still not completely free of dedicated hardware. Soundcards usually do not have DC coupled outputs. Therefore, a simple interface circuit is required to connect the radio's PTT line to another PC port. The simplest circuit consists of a resistor and a transistor, plus the required connectors. Circuit diagrams of example circuits may be found on the [FlexNet homepage](#).

Packet Radio

Packet radio requires a fairly fast switching time between receive and transmit, in the order of milliseconds. If this requirement is not met, the channel access algorithm, namely CSMA or eventually DAMA, rapidly shows performance degradation, leading to excessive collisions on the channel [Wel97].

Unfortunately, neither standard Windows95 sound drivers nor the Linux sound driver offers this performance. Actually, the Linux driver is „almost there“. An experimental user-mode implementation of the packet radio modem using the Linux sound driver performs well on some computers, but suboptimum on others, depending on the CPU speed and network activity, etc. DOS does not provide a sound driver at all anyway.

Therefore, the packet radio software has to provide its own soundcard driver. This limits the supported soundcards to the vast majority of the Soundblaster or WSS' compatible cards. Since the driver requires direct access to the hardware, it has to be implemented as a kernel module under Linux, or as the Windows equivalent, a VXD.

Both the standard operating system sound driver and the packet radio modem driver want exclusive access to the hardware. This complicates installation. Details on the installation procedure can be found on the **FlexNet homepage** for the DOS/Windows95 case or in the **AX25-HOWTO** in the Linux case.

The DOS/Windows95 driver uses **PC/FlexNet** [Jos95a, Jos95b] as its AX.25 stack. This means that every application supported by **FlexNet** or its compatibility modules can be used. This is the vast majority of the existing packet radio programs. There is no need for special terminal programs.

Under Linux, an AX.25 stack is incorporated into the kernel networking. Therefore, the natural choice was to implement the soundcard modem driver as a standard Linux networking interface. While slightly more complex to install, other AX.25 stacks such as *NOS can be used as well.

Popular HF Protocols

Popular HF protocols, such as **AMTOR (SITOR)** [CCI86] or **Pactor 1** [HS90], pose a few additional problems that have to be addressed.

HF protocols have very stringent timing requirements. CCIR recommends less than 20ppm clock deviation for **SITOR**! There are a few different possibilities to derive a clock signal in the PC:

1. the soundcard sample clock
2. the CPU clock feeding the cycle counter
3. the system timer
4. peripheral clock generator, such as the baud rate generator of a serial interface

¹ Windows Sound System; a hardware standard initially designed by Microsoft; it has nothing to do with the -availability of Windows drivers

While 1 is the natural choice for full duplex soundcards, it cannot be used on the half duplex variety, due to the discontinuity during switching between receive and transmit. 2 is a convenient source because it can be read easily and provides fine granularity, but is unfortunately only available on Pentium class machines. 3 might be inconvenient because of the games the operating system plays with the system timer. 4 ties up additional hardware resources otherwise not needed.

Neither of these sources fulfill the 20ppm requirement. The problem is not the stability, at least not if the PC is operated indoors, but the initial frequency. After all, a 100MHz P5 system may as well run with 100.1MHz or 99.9MHz. Therefore, a method to calibrate these clock sources is required. Now the only method to measure the frequency of a signal is to compare it to a signal with known frequency. Such a reference signal has to fit into the **passband** of the soundcard. I have experimented with the following signals:

- Longwave timecode transmitters. DCF77 can be received easily throughout central europe at 77.5kHz, and its pseudo noise phase code allows accurate measurements within a few minutes.
- TV broadcast horizontal line sync. VCRs with baseband video output are readily available and some TV stations have horizontal sync frequency derived from an atomic clock, such as the second **german** state network (ZDF).

GPS could also be used, but its outputs, the second clock and eventually the 10MHz reference output, do not fit into the **passband** of a soundcard, therefore some additional circuitry would be required.

Additionally, the „search space“ where to look for signals is much bigger than in the typical packet radio case. While in a typical packet radio mode the only unknown is the starting time of a transmitted packet, in an HF environment a station may be calling in any one of several major modes (such as **Amtor** or **Pactor**), each of which may consist of several variations (such as inverted or not in **Pactor**), and with an unknown offset from the receiving station's carrier frequency.

This requires that the standby station uses many demodulators in parallel to dig for possible signals, which leads to the somewhat paradoxical situation that the standby operating mode requires much more CPU power than an ongoing circuit. The user may however limit the search space in exchange for CPU load, such as by limiting the maximum allowable frequency offset. The development of such a HF engine running under the Linux operating system is in progress. Most needed now is a decent terminal program; anyone wanting to volunteer is asked to get in touch with the author.

Outlook

I am planning to continue developing the available software by adding new modes and adapting it to new hardware when it gets widespread, such as AC97.

I am also planning to bring the benefits of the software modem approach to packet radio nodes. The target platform is likely to be the next generation **RMNC/FlexNet** hardware.

Web Resources

Analog Devices, Inc.

<http://www.analog.com>

Crystal Semiconductors	http://www.crystal.com
Creative Labs	http://www.creaf.com
PC/FlexNet	http://home.pages.de/~flexnet
Linux AX.25 utilities	ftp://zone.pspt.fi/pub/linux
Author's Ham page	http://www.ife.ee.ethz.ch/~sailer/ham/ham.html

Bibliography

- [CCI86] CCIR recommendation 476-4, Direct Printing Telegraph Equipment in the Maritime Mobile Service, 1970- 1986
- [HS90] H.-P. Helfert, U. Strate, **PACTOR - Einführende Protokollbeschreibung (Level 1)**, 5. November 1990
- [Jos95a] Gunter Jost, An Introduction to **FlexNet**, 14th ARRL digital communications conference, Arlington, TX, 1995
- [Jos95b] Gunter Jost, **PC/FlexNet - Die neue Plattform für PR-Anwendungen**, 11. Internationale Packet-Radio-Tagung, Darmstadt, 1995
- [Sai95] Thomas Sailer, **FlexNet-Workshop: Die DSP-Treiber**, 11. Internationale **Packet-Radio-**Tagung, Darmstadt, 1995
- [Sai96] Thomas Sailer, „**cheaper packet**“ mit Linux, 12. Internationale Packet-Radio-Tagung, Darmstadt, 1996
- [Sai97] Thomas Sailer, „**PacketBlaster 97**“ - Soundkarten-PR mit aktuellen Betriebssystemen, 13. Internationale Packet-Radio-Tagung, Darmstadt, 1997
- [Tuc84] Tucson Amateur Packet Radio Corp., AX.25 Amateur Packet-Radio Link-Layer Protocol Version 2, **TAPR**, October 1984
- [We197] Matthias Welwarsky, Kanalzugriffsverfahren im Packet Radio, 13. Packet-Radio-Tagung, Darmstadt, 1997