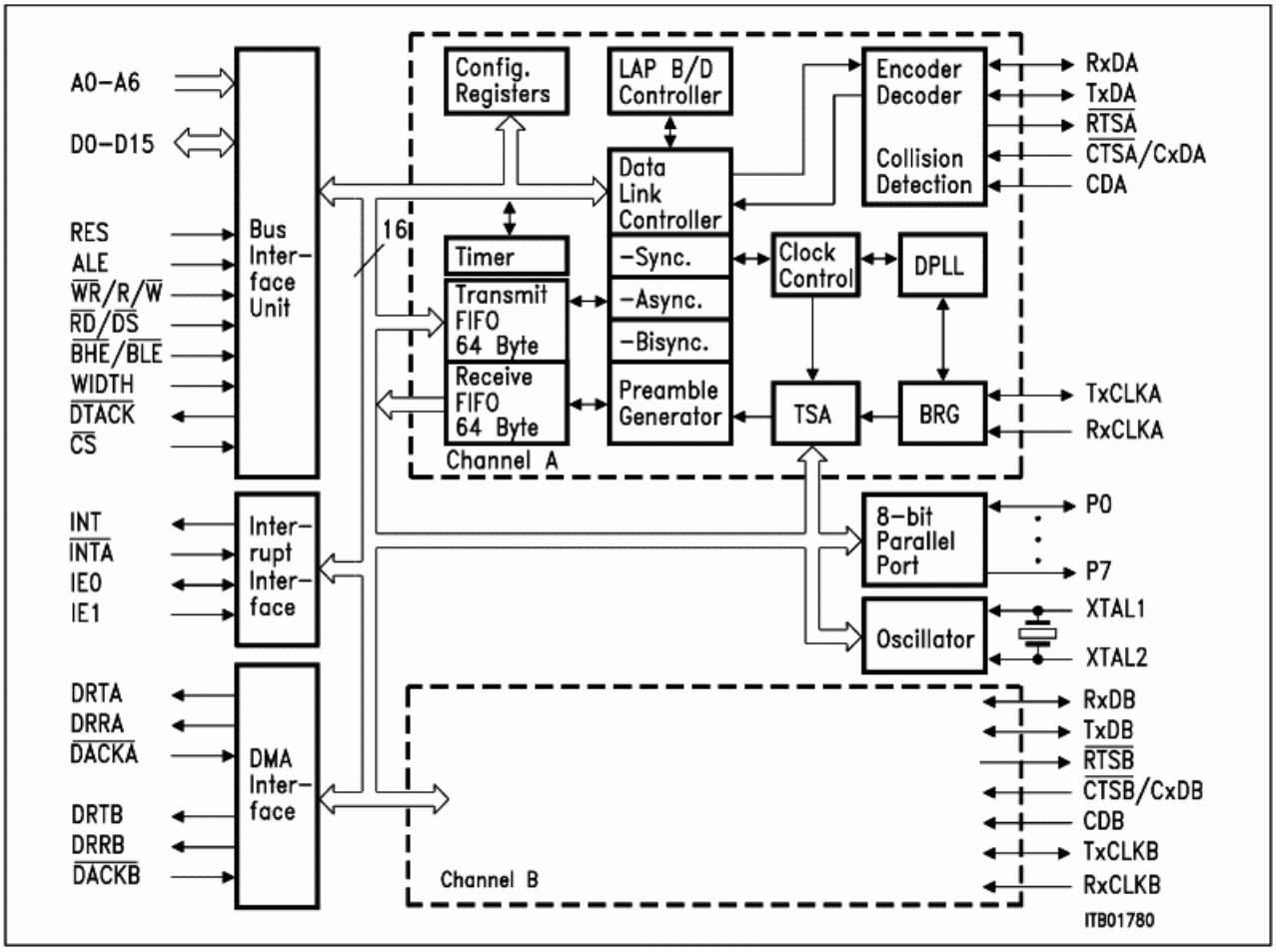
A KISS/SLIP TNC for 10Mbps

Matjaž Vidmar, S53MV

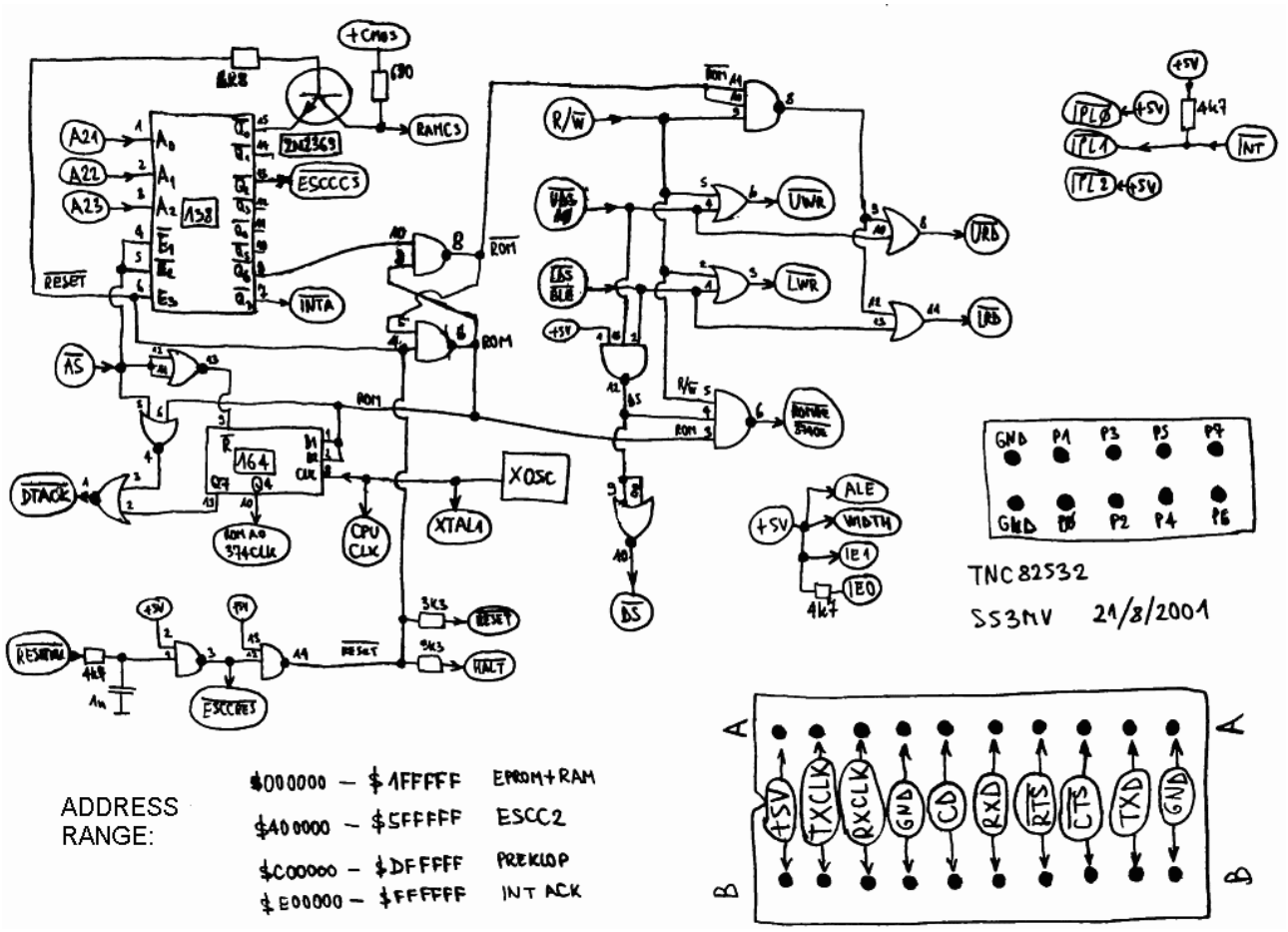
Designing a high-speed packet-radio TNC is not a trivial task. The radio channel is significantly different from wires. Therefore the usual DMA approach for high-speed data transfer with standard HDLC controllers does not work very well on a radio channel. Radio noise generates random short packets. Although the CRC of most of these packets is bad, the CPU is loaded heavily to reprogram the DMA and HDLC controllers.

A different solution is to avoid using DMA at all if there is enough storage capability in the HDLC controller itself. The Infineon ESCC chip SAB82532N10V3.2 has 64 bytes of FIFO storage on each receive and each transmit channel. This makes interrupt requests to the CPU much less frequent and allows an efficient data transfer by the CPU itself, without using any DMA controllers.

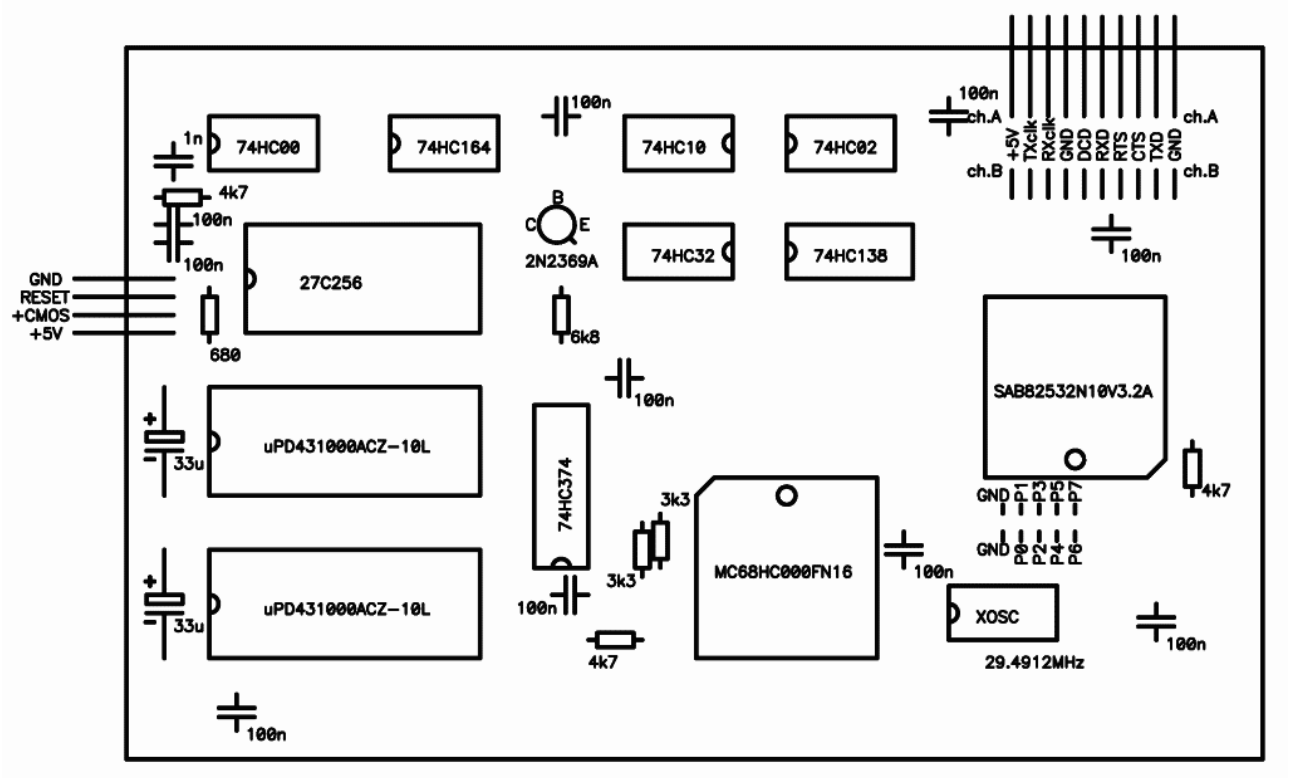
This is the block diagram of the SAB82532N10V3.2:



The TNC itself was designed around the MC68HC000 microprocessor in a similar way as my old KISS/SLIP TNC design (Z8530 HDLC operating at 1.2Mbps). Besides the CPU and HDLC chips there are two RAM chips and an 27C256 EPROM. The content of the 8-bit wide EPROM is copied into the 16-bit wide RAM at power up using some simple TTL logic:

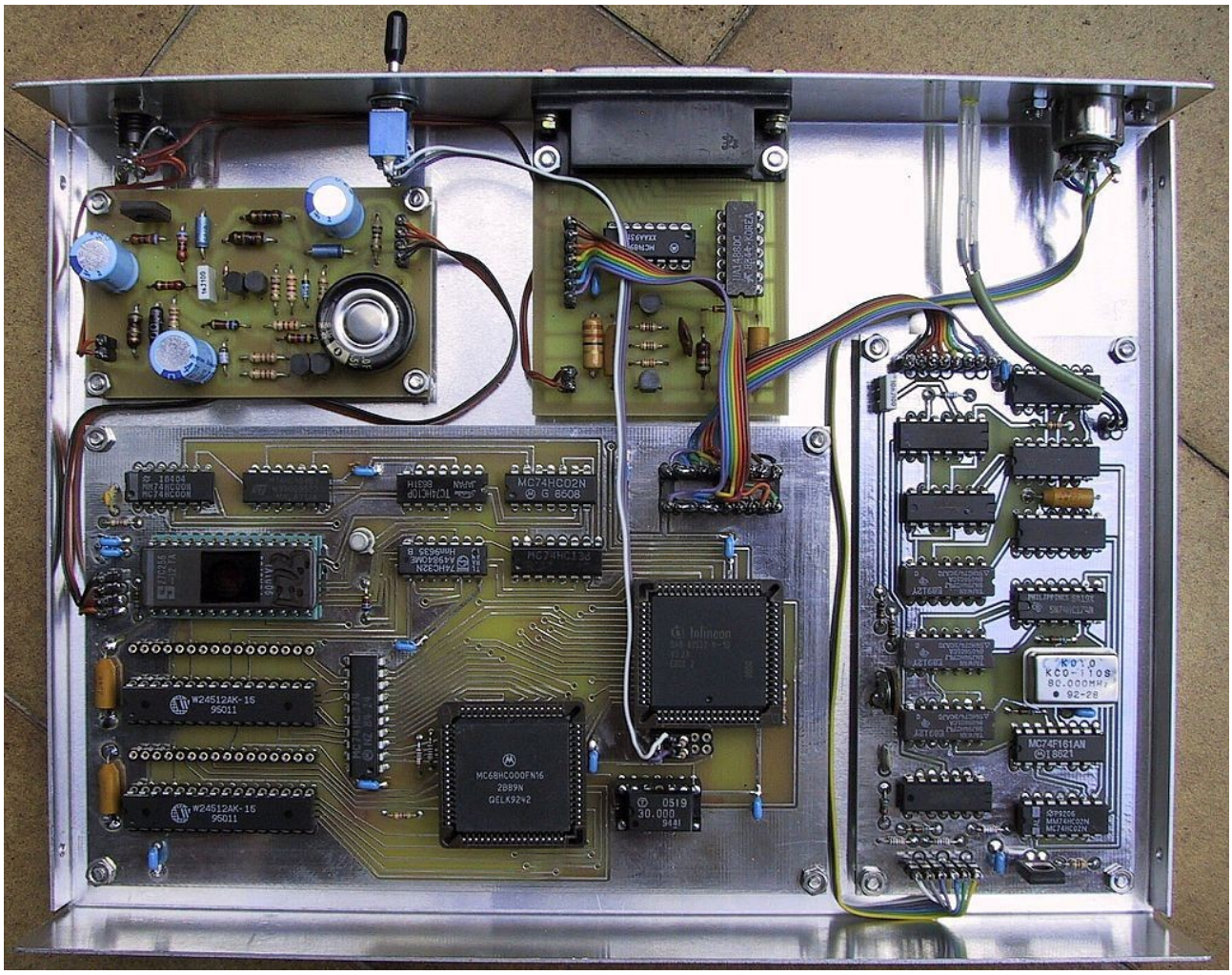


The data-bus width is 16 bits except for the EPROM and the latter is only used at power-up. The components of the 10Mbps TNC are installed on a 100mmX160mm double-sided PCB:

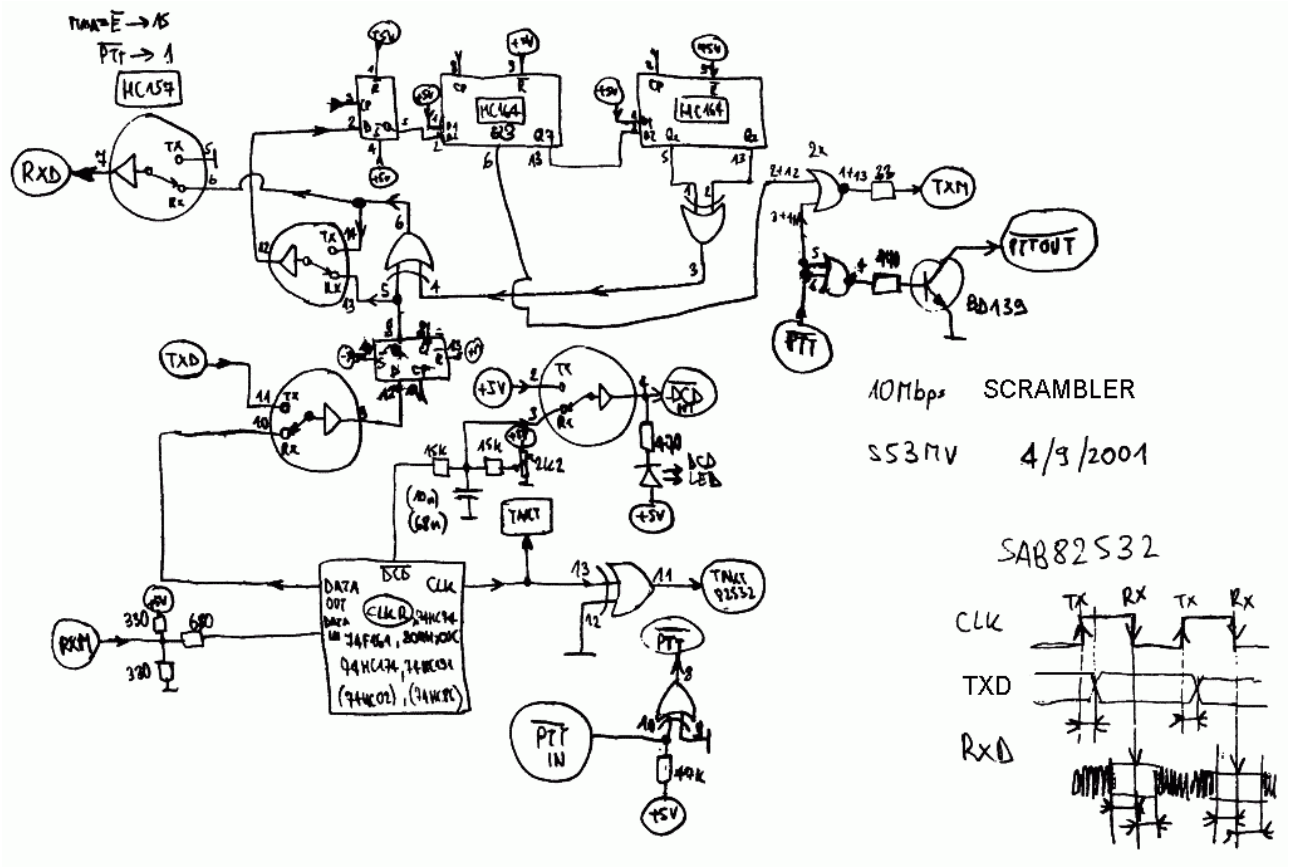


The two serial channels of the SAB82532N10V3.2 can be used in different ways. In the described TNC, the channel "A" is used for high-speed HDLC to the 10Mbps radio, while the channel "B" is used for conventional RS-232 asynchronous communication up to 921.6kbps. The parallel port is used to select the RS-232 speed and for circuit checkout.

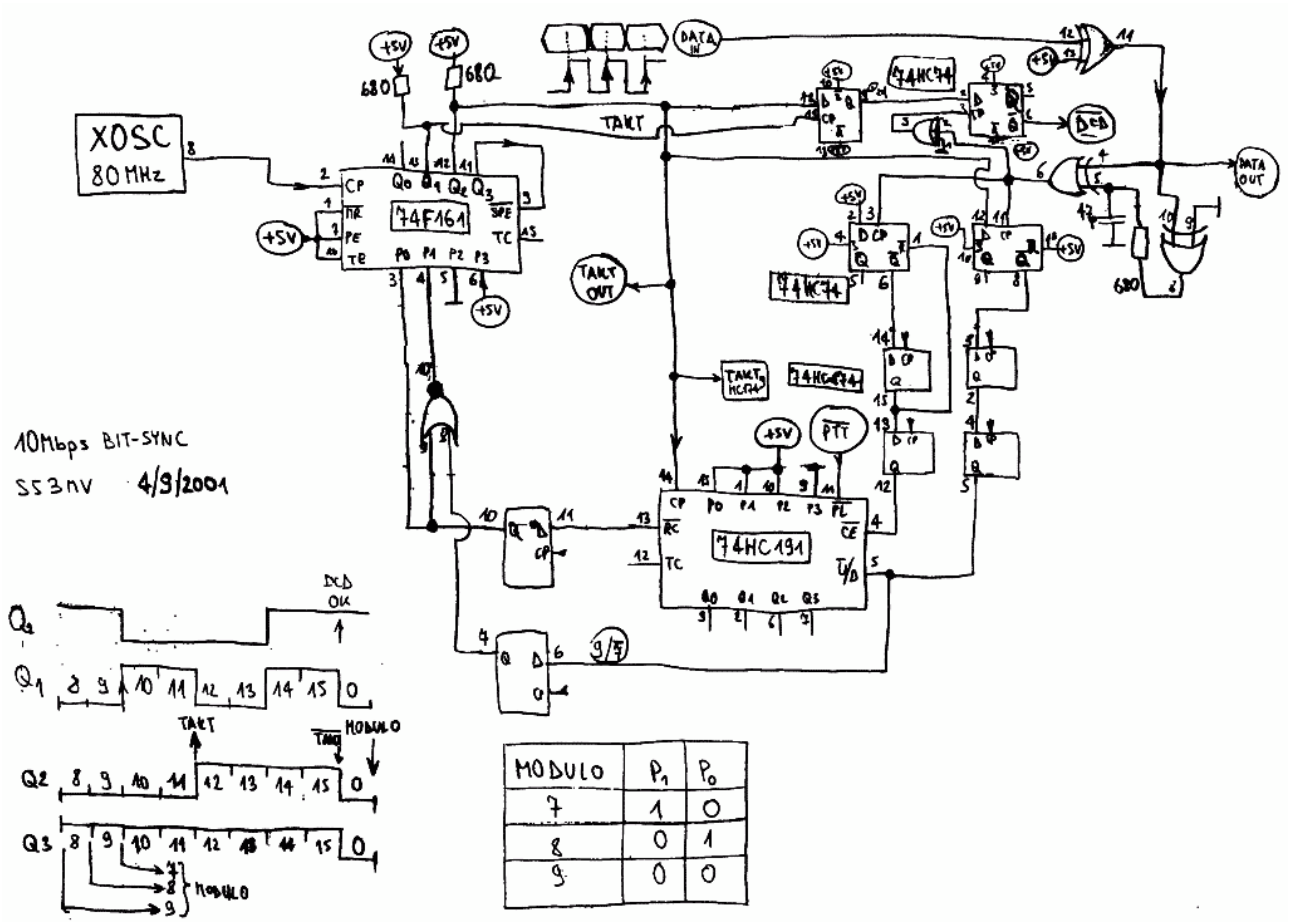
The whole TNC includes a power supply with a battery (capacitor) RAM backup and RESET generator, a RS-232 1488/1489 interface and a 10Mbps scrambler/clock-recovery circuit:



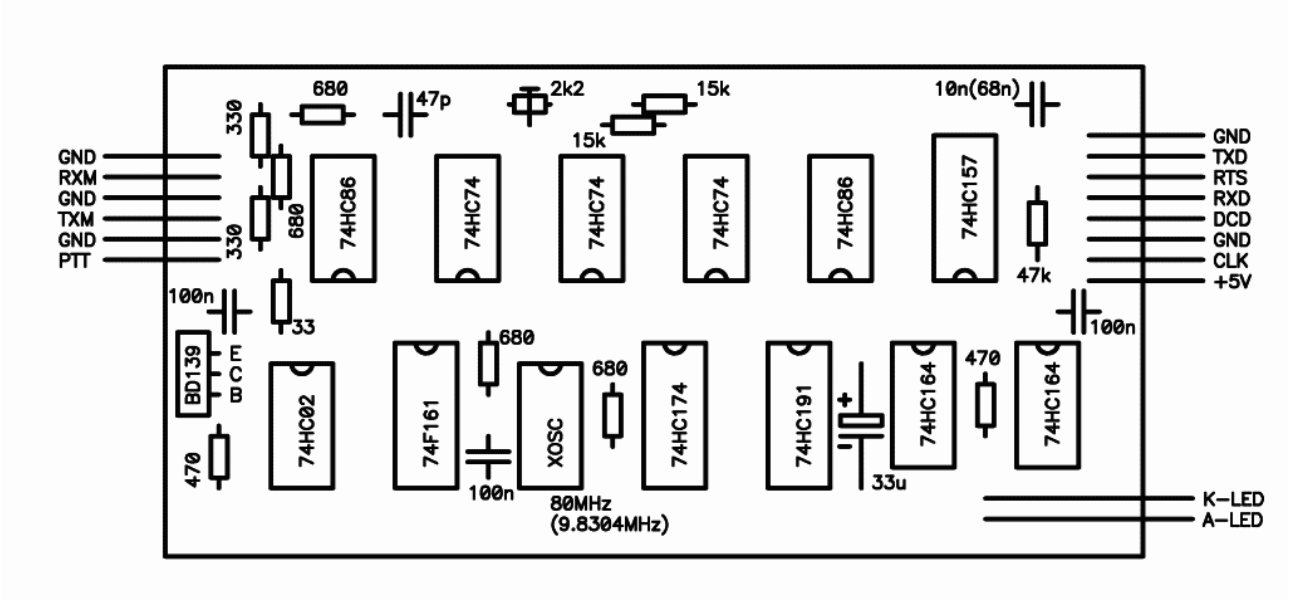
The power supply and the RS-232 interface are the same as in my old 1.2Mbps TNC design. The 10Mbps scrambler is designed with conventional 74HC... logic:



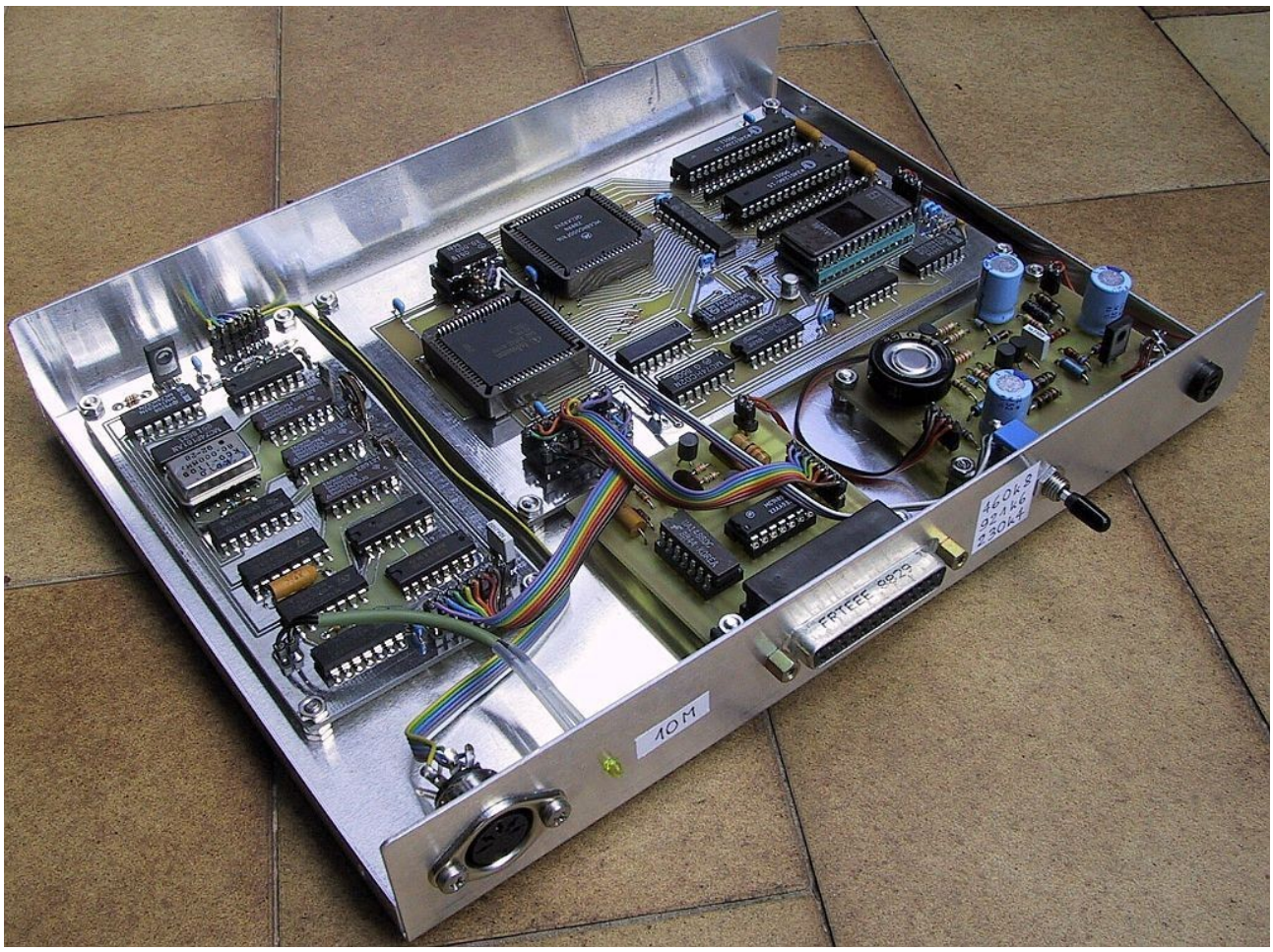
The clock recovery requires a fast DPLL. The latter works at a clock frequency of 80MHz and is designed with a single fast component (74F161 counter), while all other logic is from the 74HC... family:



The 10Mbps scrambler/clock-recovery circuit is built on a 60mmX120mm double-sided PCB (values in brackets for 1.2288Mbps operation):



The front panel includes (from left to right): the radio connector, the DCD LED, the RS-232 connector, the RS-232 speed-selection switch and the 12V-power-supply connector:



The SAB82532N10V3.2 allows a system clock up to 33MHz. The MC68HC000 in the PLCC-68 package can usually be overclocked reliably up to at least 40MHz. A full-duplex operation with short packets that fit in the SAB82532's FIFOs is possible up to 33Mbps. Reliable full-duplex operation with longer data frames is possible up to 12Mbps.

Simplex transmission experiments were made at 11.6Mbps due to the available 92.9405MHz clock oscillators, using both an artificial transmission line and simple UWBFM radios at 5.7GHz. Reliable operation can be therefore expected at 10Mbps.

The current EPROM firmware is E24 or E25. E24 supports conventional KISS and SLIP operation on the RS-232 port. At a 29.4912MHz (30MHz) CPU/ESCC clock frequency, the TTL input levels on the parallel ports P0 and P1 select 115k2, 230k4, 460k8 or 921k6 operation. The ports P2 and P3 are held high for simple pullup resistors to P0 and P1.

The ports P4, P5, P5 and P7 are connected to an internal counter so that the operation of the TNC CPU can be verified by a blinking LED. The version E25 includes a power-on delay required by some RS-232 to USB converters. All firmware automatically adjusts the RAM size to 64kb, 128kb or 256kb by checking the installed RAM chips.

* * * * *