The Ottawa Packet Interface (PI): A Synchronous Serial PC Interface for Medium Speed **Packet** Radio

Dave Perry, VE3IFB

Packet Working Group
Ottawa Amateur Radio Club
P.O. Box 8873
Ottawa, Ontario
K1G 3J2

ABSTRACT

This paper describes the design and implementation of a synchronous serial interface card using DMA for IBM PC compatible computers for use in amateur packet radio applications.

1. Background

When I became involved with the local packet radio group a couple of years ago, state of the art was to use the Heatherington 56K modem [1] with a modified terminal node controller (TNC) for the PC interface. This arrangement was far from ideal. While the radio channel ran at 56 kbit/s, the TNC to computer link ran at 9.6 kbit/s. Phil Karn had written a special driver for the DRSI PCPA card to run at 56 kbit/s. [2] However, because the hardware required programmed I/O, the driver would lock up the PC each time a carrier was detected on the radio channel. Clearly what was needed was a new piece of hardware to perform the interface function. The requirements were identified:

- A low cost synchronous interface for the IBM PC and compatibles (even the 4.77 MHz variety).
- Targeted at the end user in a network (as opposed to linking or packet switch applications)
- Easy connection to the Heatherington modem
- Software support for the NOS TCP/IP package [2]

2. Design Alternatives

Two approaches were considered to produce an interface with improved throughput and to remove much of the processing burden from the host computer. The first was to use a co-processor to handle the synchronous data, with a dual ported RAM arrangement to communicate to the host machine. This would have been a scaled down version of the "Awesome I/O" interface proposed by Chepponis and Mans [3]. This was rejected as being too capable (and therefore costly) for what I was after - an inexpensive interface for the end user.

The selected method was to use direct memory access (DMA) to transfer the packets to and from the PC memory. This method drastically reduces the processor load. The host machine can set up a packet for transmission, start the DMA transfer, and go away to do other things while the transfer takes place. A similar arrangement exists for receive. The host processor is interrupted only after a packet has been transferred into memory. Of course, you never get something for nothing. The price paid in this case is that each DMA transfer steals a few bus cycles which could otherwise have been used by the host processor, but this is a very acceptable trade-off.

3. Half Duplex vs Full Duplex DMA

This interface was targeted at the end user in a local area network (LAN) configuration. There was therefore no requirement for the interface to be able to transmit and receive data at the same time, since there can be only one user of the LAN at any instant, DMA channels are a very limited resource in the PC architecture. There are only 3, and 1 or 2 of those are often used by other interface cards such as disk controllers. For these reasons, the card was designed to use half duplex DMA, which requires only one DMA channel. DMA is used for both transmit and receive, and the direction of transfer is reversed with each exchange.

4. Circuit Description

The PI card uses a 28530 dual port Serial Communications Controller (SCC). This device has been described as the Swiss army knife of serial chips, It does, however, require a bit of glue logic to interface it to the PC bus. U1 is a bidirectional bus buffer used to meet the loading and drive requirements of the bus. U4 and Ull-F provide buffering of additional bus signals. U9 and US-A and D are used to delay the leading edge of the buffered I/O write signal for the SCC. This is required to meet the timing requirements of the NMOS version of the chip. U3, a 74LS138, and the 16L8 PAL, U2, provide all address and chip select decoding.

U6-A and US-D provide a means of masking DMA requests independent of the internal registers of the SCC. This was needed because DMA and processor accesses to the SCC occur asynchronously, and it is possible to violate the minimum time between accesses required by the chip (6 times the clock period plus 200 ns). By disabling DMA accesses for a short time before and after a processor access, this constraint can be met.

The 8253 timer chip, U8, provides an independent timer for each serial channel. One timer in this chip is used as the prescaler for the remaining two. The timer outputs are connected to the CTS pin of each serial channel. In this way, the CTS interrupts can be used to time events.

U10 provides a divide by 32 function for baud rate generation for the B channel. This was included so that it will be possible to write a full duplex driver for the B channel. There is no divider for the A channel, because it is primarily intended to be externally clocked (by the Heatherington modem, for example), and because the DMA support for this channel is half duplex.

There is a transmit watchdog circuit for the A channel, consisting mainly of Ull-B, C2, and R4. If RTS is asserted for more than approximately 10 seconds, the circuit will disable transmit, Q1 gives an open collector output for keying a transverter. Q2 can also be used to give a separate open collector output for the RTS line. This is intended to drive the Heatherington modem, which has a pull-up resistor on this line. This will prevent the transverter from being keyed up when the host computer has been turned off. Ull-A, B and C are the clock oscillator and buffer for driving the SCC at 3.68 MHz. The timer chip is clocked at half this frequency by U6-B. Position U12 can either be jumpered across to provide a TTL level interface to the B channel, or it can be stuffed with a Motorola MC145406 RS232 transceiver chip.

5. Software

The PI card driver has now been integrated into NOS. The source code for it (as with the rest of NOS) is freely available.

6. Conclusions and Future Directions

File transfer rates of up to 5000 bytes/sec have been achieved under controlled conditions, while rates in the range of 3000 to 4000 bytes/sec are representative of real world performance on our network. While this is a great improvement over the rates of a few hundred bytes/sec which were previously attained, there is still room for improvement. The transmit delay time required by the modems is on the order of 15 ms and should be reduced, since it has now become the biggest factor limiting performance.

The interface has also been successfully tested at data rates up to 250 kbit/s over a wired connection with a special version of the driver.

As a result of volunteer efforts by members of the Packet Working Group of the Ottawa Amateur Radio Club, a number of PI cards are now in use around the world. Contact the club for information.

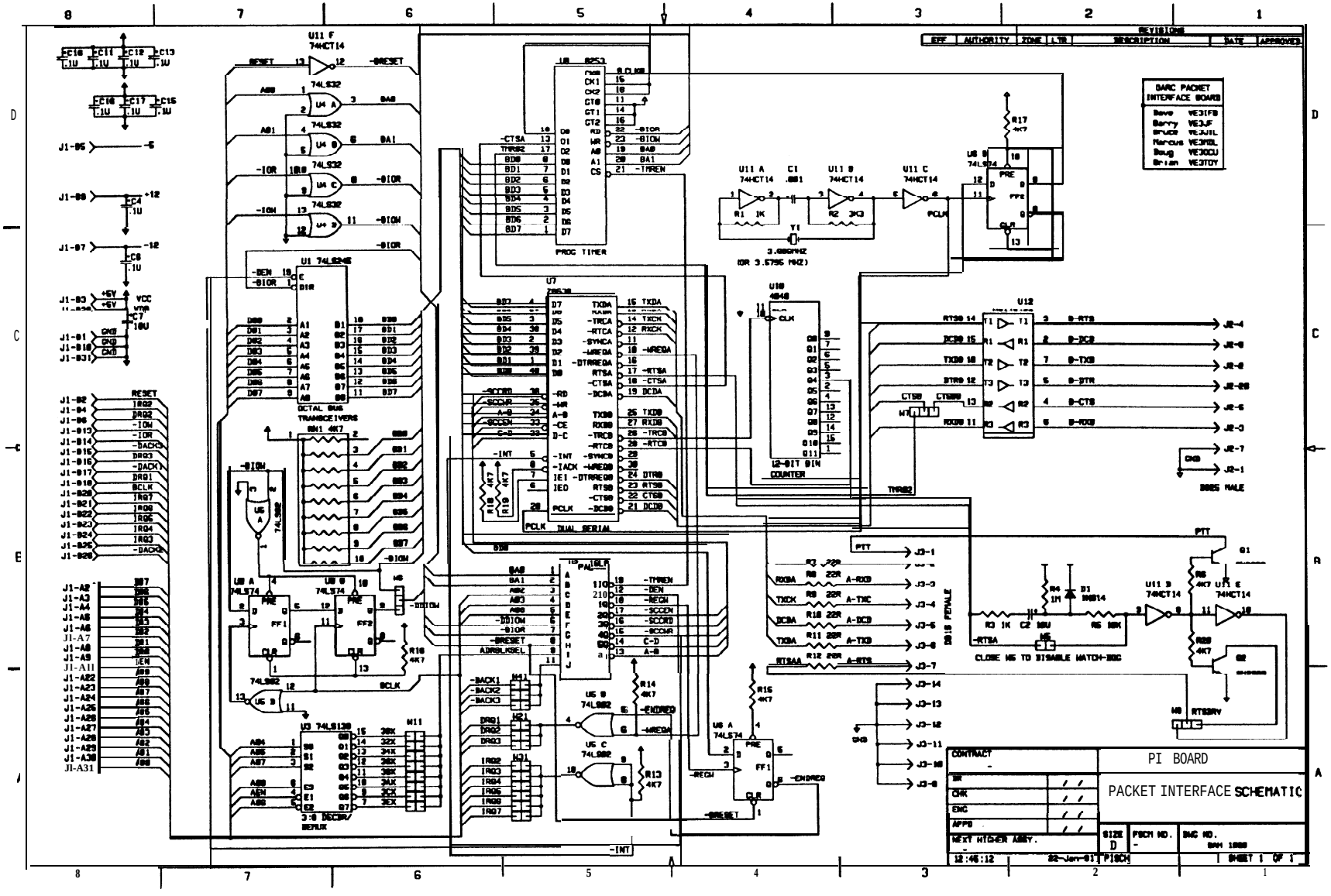
On the software side, there have been requests for a packet driver compliant version of the driver, and for G8BPQ front end support. I welcome correspondence with anyone who wishes to tackle these challenges.

7. References

1. Heatherington, Dale A., WA4DSY, "A 56 Kilobaud RF Modem", *ARRL Amateur Radio 6th Computer Networking Conference*, pp. 68-75, Redondo Beach, California, August 29, 1987

2. Karn, Phil, KA9Q, "Amateur TCP/IP in 1989", *ARRL Amateur Radio 8th Computer Networking Conference*,pp. 114-118, Colorado Springs, Colorado, October 7, 1989

3. Chepponis, Mike, K3MC and Mans, Bernie, AA4CG, "A Totally Awesome High-Speed Packet Radio I/O Interface for the IBM PC XT/AT/386 and Macintosh II Computers", *ARRL Amateur Radio 7th Computer Networking Conference*,pp. 36-40, Columbia, Maryland, October 1, 1988



example), and because the DMA support for this channel is half duplex.