Xerox 820-1 Compendium—Part 3

By David W. Borden, K8MMO
c/o AMRAD, PO Drawer 6148
McLean VA 22106-6148

Xerox 820-1 Again

Many AMRAD members purchased the Xerox 820-1 board in 1985 from the Xerox Outlet store in Texas (item now out of stock). Since that time, AMRAD has decided that the computer is another standard model capable of withstanding experimental assignments.

Using software written by Phil Karn, KA9Q, and modified by G. J. van der Grinten, PAØGRI, and myself, I have been transmitting and receiving packets with the Xerox 820-1 single board computer. The software required attention, but before discussing that, let's recall that the Xerox 820-1 board consists of a floppy disk controller and an SIO.

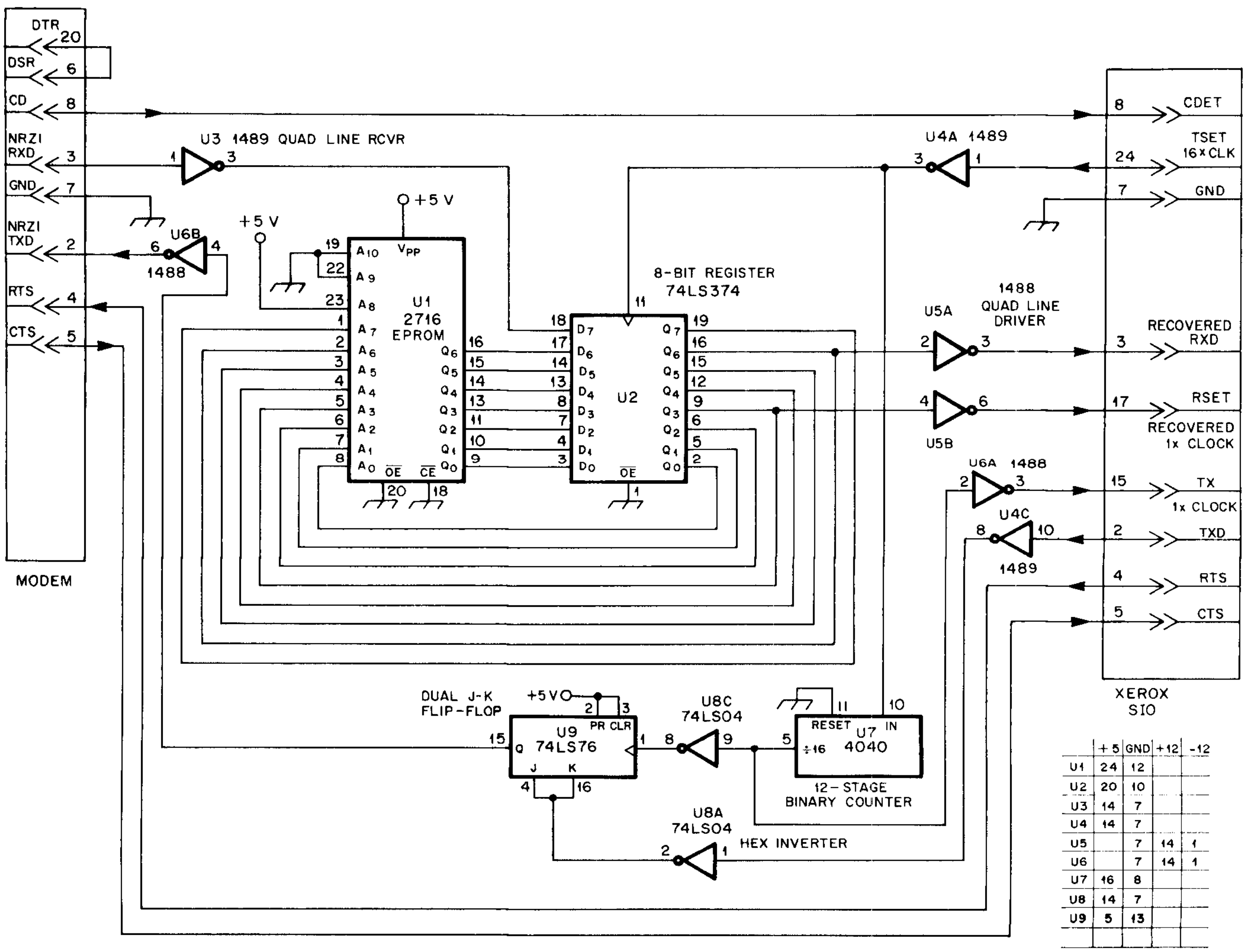
Once I received my package of hardware from the dealer, my immediate task was to find a parallel keyboard to connect to. I used an Apple II keyboard; they are inexpensive and easy to use. One drawback is that the Apple has no curly braces or brackets. This cramped my C coding style and I knew I'd have to find something else.

My disk drives are 5¼ inch Panasonic double-sided drives. They can be obtained from vendors advertising in the back of computer journals (or the *Computer Shopper*[26]). For packet operation, only a single-sided drive is necessary. With a fabricated connector cable, the drives should easily plug into the Xerox 820-1. A DB-37 male connector is available from Electronic Equipment Bank.[27]

Interface Board Receive

The packet device on the Xerox 820-1 is the Zilog SIO. It does not speak NRZI, nor listen to it, thus, an interface board that goes between the Xerox 820-1 and the modem is necessary. NRZI data must be transmitted to the modem and radio, and then received via the same route.

Notes appear on page 16.



	+5	GND	+12	-12
U1	24	12		
U2	20	10		
U3	14	7		
U4	14	7		
U5		7	14	1
U6		7	14	1
U7	16	8		
U8	14	7		
U9	5	13		

Fig 1—The Zilog SIO on the Xerox 820-1 does not understand NRZI code, thus it is necessary for this interface board to be connected between the Xerox board and modem.

Page 14, QEX, August 1986, published by The American Radio Relay League, Inc.

Also, it is necessary to recover clock from the incoming data in true synchronous fashion. In the old asynchronous days, the clocks had start and stop bits to synchronize on. In the synchronous world, there are no start and stop bits.

The interface board is shown in Fig 1 and has a varied history. Jon Bloom, KE3Z, built the first in March of 1983. At that time, AMRAD had purchased the ATT Blue/Green Box STD bus computers. They used an SIO, also. Jon built a repeater using these boxes and it is from this work that the receive side was developed—a state-machine decoder. The PROM and latch combination act as a programmed device to extract data clock and normal data from the incoming NRZI data stream.

Interface Board Transmit

A simple circuit for generating NRZI data was found in The FADCA's newsletter.[28] Designed by Sumner Hansen, WB6YMH, the circuit uses an inverter (added by Howard Goldstein, N2WX) and a JK flip-flop to generate the NRZI data from the normal data produced by the Xerox 820-1 SIO. I later added a 4040 divider to obtain 1x TX clock, which the SIO requires. The board uses nine chips that cost about $20. A slight modification of the Xerox board is required to obtain a 16x clock output and power for the board.

Adding Little Wires to Xerox

I wanted to use the SIO on the Xerox board with as little hassle as possible, so I added RS-232-C interface chips to the interface board. To do this, turn the Xerox board over and connect a wire from pin 29 of J9 to the DB-25 SIO connector, pin 24. This is the 16x clock that KE3Z points out as TSET on the schematic.

While you are working on the Xerox board, bring out the +12, −12 and 5-volt lines to the DB-25 SIO connector to power the interface board. This is not required, however. You can purchase an inexpensive board from Radio Shack and steal power from the Xerox board. This interface draws practically no power. Next, pull all jumper plugs on the Xerox board from J9 and place them as follows: 33-34 (1x RX Clock), 37-38 (1x TX Clock), 7-8 (TX Data), 11-12 (RX Data), 15-16 (RTS), 19-20 (CTS) and 27-28 (DCD).

The Prom

The heart of the KE3Z state machine is the PROM. To program your own 2716, burn in the code shown in Table 1.

Table 1
Code for Programming a 2716 PROM

41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	40
41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	40
41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	40
41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	40
41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	40
41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	40
41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	40
41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	40
41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	40
41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	40
41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	40
41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	40
41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	40
41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	40
41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	40
41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	40
01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	40
03	04	05	06	06	07	08	08	09	09	0A	0B	0B	0C	0D	0E
21	22	23	24	25	26	27	28	29	2A	2B	2C	2D	2E	2F	00
23	24	25	26	26	27	28	28	29	29	2A	2B	2B	2C	2D	2E
41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	40
43	44	45	46	46	47	48	48	49	49	4A	4B	4B	4C	4D	4E
61	62	63	64	65	66	67	68	69	6A	6B	6C	6D	6E	6F	00
63	64	65	66	66	67	68	68	69	69	6A	6B	6B	6C	6D	6E
13	14	15	16	16	17	18	18	19	19	1A	1B	1B	1C	1D	1E
11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F	30
33	34	35	36	36	37	38	38	39	39	3A	3B	3B	3C	3D	3E
31	32	33	34	35	36	37	38	39	3A	3B	3C	3D	3E	3F	70
53	54	55	56	56	57	58	58	59	59	5A	5B	5B	5C	5D	5E
51	52	53	54	55	56	57	58	59	5A	5B	5C	5D	5E	5F	30
73	74	75	76	76	77	78	78	79	79	7A	7B	7B	7C	7D	7E
71	72	73	74	75	76	77	78	79	7A	7B	7C	7D	7E	7F	70
FF	FF	FF	FF	FF	FF	FF	FF	FF	(the rest of the PROM is FF)						

Alternatives

There are other ways to design the interface board, but it might not be cost effective. The FADCA designed a small daughterboard containing the 8530, a super chip that is described in the next installment.

Why did I do all this? I want to go fast. Users will discover that fat computer generated packets do not compete well with skinny packets on the same channel. Two channels is not the answer. There are only so many repeaters to link the East Net chain. Bulletin boards are not the answer. The answer is the Level Three Network! That requires speed, at least 49 to 56 kilobaud. Hal Feinstein, N2WX, has done some work on the state machine and believes 44 kilobaud is possible. I intend to use it with spread spectrum on the radio. Alas, it is mostly software that makes the data world go round.

Xerox 820-1 Packet Software

The software runs on a Xerox 820-1 board that has been augmented by the hardware board. The packets are generated by the Zilog SIO residing on the Xerox 820-1. The add-on board generates the NRZI encoding on transmit and recovers clock and data on receive. This is not the only approach to Xerox 820-1 packets. TAPR sells a board which uses the more modern Zilog 8530 SCC and Terry Fox, WB4JFI, is developing software for it.[29] To install the TAPR board, remove the PIO from the 820-1 and plug the daughterboard in place of the PIO.

Here is a description of the code, written by Phil Karn, KA9Q:

"Description of AX.25 'C' Code

Overview

The code currently implements AX.25 Level 2 as defined in the Version 2.0 document (as released by the ARRL Ad Hoc Digital Communication Committee). This code contains a number of extensions to AX.25 which are new to Version 2.0 and, to my knowledge, is the first AX.25 implementation to do so. These include:

- Idle link keep-alive polls (timer T3)
- Acknowledgement delay (timer T1)
- "C" bits and correct P/F bit usage according to LAPB (link-access-protocol, balanced)

Item three is upward compatible in that it is not used whenever the software detects an "old" (ie, pre-Version 2.0) implementation of AX.25 (which includes all other AX.25 implementations). The keep-alive polls require proper use of the P/F bit and will not be sent to an old implementation.

This code also supports multiple simultaneous logical links. This is an implementation rather than a protocol issue;

it involves keeping individual "link descriptors" for each station, and uses a hash lookup algorithm to find the appropriate link descriptor whenever a frame is received. Support is also provided for multiple physical HDLC ports so that more than one channel can be used.

These extensions required me to separate and classify the commands and parameters (from a unit like the TAPR TNC) into groups that are applied on a per-link, -line or -terminal basis as appropriate.

Digipeating works, but repeats a frame only on the physical link on which it was received. With relatively little work, it would be possible to do cross-channel digipeating, although with the work underway to define and implement higher level protocols, it is not clear this is a "clean" thing to do.

A number of things in the TAPR TNC are not here. Some of them, such as CW identification and the Vancouver protocol haven't been implemented."

Software Description

General

All I/O is interrupt driven. Full use is made of the 820-1's interrupt capability; queues are emptied or filled with characters going to and from the terminal and SIO. A top-level loop in main.c executes higher-level functions which operate on the "other ends" of the interrupt queues. The CTC timer ticks at a 10-Hz rate and provides all real-time functions. A halt instruction at the end of the main loop causes the CPU to wait until an interrupt occurs; this might help reduce RFI and/or power consumption on a CMOS version of the Z80. It also makes possible the use of a "CPU speedometer" which monitors the HALT line from the CPU and indicates the amount of real time being used by the software.

Review of Software

Phil's description, only partly reproduced above, also speaks of each routine and how it works. In the software trade, we call this a three-banana program: 1) the coding is finished, 2) the program works and 3) it is documented. I find the program a joy to use. At first, I experienced trouble with bugs. I received a finished version of the program, but not before it was tested. I found that it would not send full packets. The last three characters were trashed as the RTS line was lowered and output ceased. Phil later visited AMRAD and brought a fixed version that worked properly.

Vancouver Code

I ran a wire across the room and placed Bell 202 modems at both ends. A Xerox 820-1 was connected at one end and a Vancouver board (with an AMRAD daughterboard attached) was at the other end. What a test! In a quick summation, the Vancouver board does not handle the new version 2.0 AX.25 protocol SSID field or P/F bit. KA9Q added two quick fixes to allow for communication with the Vancouver boards. I then moved the clip leads from the Xerox 820-1 modem to the radio interface and fired up on 145.01 MHz.

On-The-Air Testing

I connected with several operators on 145.01 MHz. Phil's code worked swell. The software allowed a full trace of what was happening on the channel. Both outgoing and incoming frames were traced (as desired) and each frame type was announced along with desired bits broken out so you can see if the other station was sending the right stuff.

Multiple Connections

Multiple connections is required for networking and it would be neat for network servers to use them too (allowing data base access or download to multiple users). I connected with two packet users at one time and tried to carry on two conversations. The software worked well. I went into the command mode and ordered the software to "convers WB4JFI-2," then I typed data to Terry. Later, I typed a control-C followed by a carriage return to go back to the command mode. I entered "convers AJ9X-0" and transmitted information to Mike. The software enabled me to type to Mike and faithfully receive frames from Terry, simultaneously. When I got back to Terry, the frames flooded the screen.

The Zilog has two channels and Phil's code addresses both. I said we could have multiple connects on channel A, well, channel B is there also. How about receiving a file in the transparent mode on channel B while discussing the file on channel A with the file server, when you are also connected to another user on channel A? How much can your brain handle? There is more. How many modems and radios do you have?

There is something else you could do with channel B that is addressed in the description above. Take in all the packets from 147.21 MHz and put them out on 145.01 MHz, and vice versa. That is a packet switch! As Phil points out, this is cheating. We need AX.25 Level 3 or another protocol. However, remember both channel A and B could have multiple connections. The mind boggles.

[Dave Borden, K8MMO, continues to share his enthusiasm about the Xerox 820-1 board in Part 4 of the Compendium. K8MMO presents an ICOM IC-2AT connection and Terry Fox, WB4JFI, talks about a FAD and a modem.]

Notes

[26]*Computer Shopper Magazine*, 407 S Washington Ave, Titusville, FL 32781.

[27]Electronic Equipment Bank, 516 Mill St, NE, Vienna, VA 22180, tel 1-800-368-3270, VA residents only 703-938-3350.

[28]*The Beacon*, Mar 1984, The Florida Amateur Digital Communications Association, Gwen Reedy, 812 Childers Loop, Brandon, FL 32511.

[29]TAPR, PO Box 22888, Tucson, AZ 85734.

Bits

FCC Cracks Down on Illegal Microcomputer Operations

If you have recently purchased a clone of your favorite computer, or are planning to, there is one thing you should know: Federal law prohibits computers and other electronic devices from emitting transient radio waves that interfere with radio and television reception. This is a four-year old ruling that many manufacturers choose to ignore!

If your purchase does not exhibit an FCC label, you have the right to return the computer to your dealer and ask for a replacement. It is not only illegal to manufacture and sell these interference-causing appliances, but the user can be fined or imprisoned if an interference complaint is involved.

Thus far, the FCC has been shaking illegal manufacturers out of the trees. Many small-based operations, including mail-order firms, can't afford to pay the price once they're found out—afford FCC approved parts or $5,000 and the two months it takes to test the machine for FCC approval.

Cautious buyers should check the physical appearance of the computer before purchase. Are the mating surfaces flush, tight and clean of paint? Is the case metal or metal shielded with no holes? The cables should be shielded and the cable connectors flush, tight and shielded. The internal electrical connections should be clean, tight and paint free. The microprocessor should also be shielded. These are only some of the steps manufacturers may take to eliminate microcomputer radio interference.

You can fool some of the people all of the time, and all of the people some of the time. By exposing sales schemes such as these to potential buyers, we can only hope to revise the sentence to read. . . you cannot fool all of the people all of the time!—*Maureen Thompson, KA1DYZ*

[Some of this information was excerpted from J. Forbes, "FCC Cracks Down on Micro Emissions," *InfoWorld*, May 12, 1986, Vol 8, Issue 19, p 1, and P. Chabal, "FCC Warns Owners of Illegal Micros," *InfoWorld*, May 19, 1986, Vol 8, Issue 20, p 5.]