

Xerox Corporation
Palo Alto Research Center
3333 Coyote Hill Road
Palo Alto, California 94304
(415) 494-4390

XEROX

Monday, October 19, 1981

Al Putoff, Technical Specialist
Ramtek Corporation
2211 Lawson Lane
Santa Clara CA 95050

Dear Mr. Putoff:

As per our conversation of last Friday, I am enclosing a recent revision of my list of difficulties I've had with interfacing the Ramtek 4100 ColorGraphic Printer. I believe this information may be useful to your engineering staff, and I would appreciate your forwarding this letter to the responsible engineers. I would be most appreciative of any comments or reply to any of the technical points raised herein. And I would certainly be happy to discuss any of these points further should your engineering staff wish to contact me directly.

Sincerely,

Richard C. Pasco
Member of Research Staff

Last Revised: October 19, 1981 2:30 PM
 Filed on: [Indigo]<DA>RamtekPlotter>RamtekProblems.Bravo

The following is a list of problems I've had interfacing our Ramtek 4100-01 Colorgraphic Printer, and my recommendations for changes (to the manuals, hardware, or firmware) to prevent others from experiencing similar problems.

*Richard Pasco
 Xerox Palo Alto Research Center
 3333 Coyote Hill Road
 Palo Alto CA 94304
 (415) 494-4390*

Documentation

I have received three documents with my 4100 printer: [1] A "Preliminary Operators Manual" dated February 3, 1981, [2] A "User Guide", publication Number 8000037-01, Rev. A, and [3] A "Hardware Reference Manual and Field Service Guide", Publication Number 8000037-01, Rev. A, dated September, 1981. Of these three, I have found [1] to be the most useful, because it provides a concise set of information about electrical interfaces to the printer and how to program it. This information is absent from [2], and in [3] it is obscured by much information about the internal workings of the printer. As the person who is programming the computer at the other end of the interface cable, I would most like to see a special section in current documentation concentrating detailed technical information about the electrical interface, data formats, and modes of operation. This information would include answers to questions raised in the remainder of this document, but would not address the circuitry implementing these operations within the printer.

Graphic mode

The statement "Graphic mode is entered by preceding each load of the buffer with the command byte 05H" ([1], page 7, Section 4.2) is not strong enough to convey the fact that after each nine lines of pixels the printer exits from graphics mode to text mode, and another 05H must be sent in order to continue graphics printing. This imposes on the host computer the necessity of keeping count of the lines sent to the printer. An early version of my graphics driver, without such a counter, sent a single 05H, then a succession of lines, each containing some number of graphic bytes followed by an End Line command. The first nine lines printed correctly, but subsequent data was ignored (as non-printing control characters) until a green pixel 05H happened along, which the 4100 interpreted as an Enter Graphic Mode command. To reduce this confusion, I would recommend changing the last sentence of the second paragraph to read, "At the end of the ninth line the printer starts the print operation, *and returns to text mode.*"

If it were possible without creating compatibility problems with other units in the field, I would recommend modifying the 4100 firmware so that after the Enter Graphic Mode command the printer would remain in graphic mode until an explicit Abort Graphic Mode command. This would relieve the host computer the necessity of counting by nines, and would permit future upgrades to printers with greater than nine wires per printhead, without modification to the host computer software.

Top-of-Forms.

If a FormFeed character is sent after power-up but before top-of-form is initialized using the manual procedure in the User's Manual, an incredible length of paper is spit out onto the floor. This disagrees with the in the manual's statement that such form feeds are ignored ([1], page 4, Section 3.1). This creates the situation where a naive user might power up the machine, start his printing, and walk away, unaware that the printer will waste paper. It also makes it kind of silly to have the power switch outside the lid if you still have to open the lid to set TOF. Recommendation: Modify the power-up initialization in the firmware to set Top-of-Form to "here" (wherever the paper happens to be) at power-up.

I have discovered that sending the character sequence to program forms length ([1], page 9, Section 4.3) effectively also sets top-of-form. That is, the sequence Escape Form-Feed 22_{decimal} (which explicitly sets the forms length to the default 11 inches) seems to have the same effect as the manual ‘RST’ switch in the offline mode. This useful feature should be documented in the User’s Manual.

The printer still has trouble keeping track of where the top-of-form is when graphics printing extends beyond a form boundary. We set top-of-form manually, print a line of two of text, enter graphics mode and print 13-14" of graphics, return to text mode and send a Form-Feed. One might expect that the printer would advance to the top of the first sheet after the one we were printing on. It does not. Instead, it moves some apparently random amount.

Wasted time and motion.

Send the printer an ‘Enter Graphics Mode’ command and begin sending graphics data. The print carriage immediately begins moving back and forth across the paper. It continues moving until until the buffer is full. If the host computer stops sending data before the buffer is full, the carriage will move indefinitely. Even if the buffer is quickly filled, time is wasted waiting for the carriage to return to the left edge of the page so it can begin to print.

Despite the parallel interface, the printer accepts graphics data barely fast enough to allow graphics printing of full lines at full speed.

Oscilloscope measurements indicate that the printer asserts ‘busy’ on a printing pass until the carriage reaches the right edge of the paper. Data transfer may then resume and, in order to print on the left-to-right pass, must be completed before the carriage reaches the left edge of the paper. Stopwatch measurements indicate that the printer carriage takes about 1.67 seconds to return. In order to transfer nine lines of graphics with 918 bytes per line (8262 bytes) during 1.67 seconds, each byte must be transferred in about 200 ms.

During data transfer, oscilloscope measurements indicate that beginning about 15 ms after each data strobe, the printer asserts ‘busy’ for approximately 160 ms, followed immediately by about 7 ms of ‘acknowledge’. This consumes 182 ms per byte, leaving just 18 ms for the host computer to supply the next byte.

The driver program in our host computer takes about 40 ms to supply the next byte. (Note that although this exceeds the 18 ms requirement derived above, it is still only one-fourth the time the printer takes to acknowledge each byte.) This brings the transfer time to 222 ms per byte, and it takes 1.84 seconds to fill the buffer. By this time, we’ve just missed the beginning of the next left-to-right pass, and, as described above, the carriage makes a complete round trip printing nothing. As a result, the effective printing rate is cut in half.

Recommendations: Apparently the present firmware keeps the carriage in motion whenever the graphics buffer is non-empty. Wasted time could be saved if the firmware were modified to put the carriage in motion only when the buffer is full and ready to print. It would also seem that a simple hardware FIFO at the parallel input to the printer could greatly ease the speed requirements on the host computer, by buffering data as it arrives from the host and reducing the time ‘busy’ is asserted.

There need to be more numbers on the Centronics Interface Timing Diagram ([1], Appendix A, and also [3], Figure 3-3). The following numbers were omitted entirely. We got them from oscilloscope measurements, but feel they should be provided in the manual:

Busy Delay:	15 ms, typical.
Busy (for normal data):	160 ms, typical.
Busy (for printing):	1.67 sec, minimum.
Ack Delay (for busy condition):	<< 1 ms. [Ack seems to happen immediately at end of Busy]
Ack:	7ms, typical.