Technical Correspondence

Conducted By Paul K. Pagel, N1FB
Senior Assistant Technical Editor

HF WEFAX FOR THE IBM® PC, PCjr AND C64

☐ Elmer Schwittek's WEFAX program just wouldn't work with my setup.[1] The printer behaved erratically and the line timing was off. After modifying the machine language code to the standard IBM dot-matrix format (ESC L 192 3, for a 960-dot line), the printer behaved, but the timing values still resulted in a smeared picture. I don't believe the timing difference is caused by computer clock frequency differences as Elmer speculates, but is caused by printer clock speed differences.

In his article, Keith Sueker points out the problems with printer timing differences and came up with the really unique idea of just stuffing the data bytes down the printer line at a very precise rate controlled by the computer's crystal-controlled clock.[2] Then, you can construct a delay loop to run out just as the next line of data is ready to begin. Schwittek used the standard printer routine that only sends a byte to the printer after it says it's ready—hence, introducing a delay not controlled by the computer's crystal-controlled clock. There are advantages to using this standard printer routine because it automatically adjusts the feed rate to exactly what any particular printer will accept, but as Sueker notes, it sure upsets the line-to-line timing. There is a straightforward way around this apparent dilemma: That's to do all the "data line" timing from the first of each data line to the start of the next data line, without any of the in-between operations entering into the timing at all.

Most computers have a separate timer chip used for a time-of-day clock, sound synthesizing, and so forth. This timer runs "off-line" without its count being affected by other code execution. The program just sets the count to zero at the beginning of each line. After executing all operations for that line, the program goes back to the counter and waits for it to reach the predetermined line-to-line delay (1.5 seconds for every-third-line WEFAX). Then the program resets the timer to zero and off it goes again. Instead of three interlocking adjustments that change with printers, you have just one—and it shouldn't need adjustment.

With the two very different computers I have (an IBM PCjr and a C64), the timer counts were set based on the published frequency of their clocks, and needed no adjustment. With this arrangement, you can change to other FAX formats (60 LPM or commercial pix) by changing only one number. On the IBM PCjr, I used the time-of-day clock. By resetting it (just during program

```
10 '              K3BC WEFAX.BAS
20 '
30 '
40 CLEAR,&HØØØ
50 KEY OFF:CLS
60 LOCATE 10,33:PRINT "Is printer ON? If so, please"
70 LOCATE 14,28:PRINT "press any key to continue...."
80 IF INKEY$ = "" THEN GOTO 80
90 CLS
100 LOCATE 13,24 : PRINT "Loading FAX program............."
110 LOCATE 15,24 : PRINT "Check to see if you're on USB..."
120 TP% = 124 'PIXEL SIZE (HORIZONTAL RESOLUTION)
130 TL% = 30  'TIME BETWEEN SUCCESSIVE LINES WITH 1 SEC. = 20. MUST
140           'BE MULTIPLES OF 10.   (VERTICAL RESOLUTION)
150 TY% = Ø   'YOUR VARIABLE FOR FURTHER DEVELOPMENT
160 DEF SEG = &H2FØØ
170 FOR F = Ø TO 463
180 READ G : POKE F,G
190 NEXT F
200 FOR F = 464 TO 2450 : POKE F,255 : NEXT F
210 CLS
220 LOCATE 6,22 : PRINT "If FAX audio signal is present, press"
230 LOCATE 7,22 : PRINT "'G' to start copying (to SYNC you"
240 LOCATE 8,22 : PRINT "must press 'G' during SYNC ticks)."
250 LOCATE 13,24 : PRINT "To stop FAX copy, press SPACEBAR"
260 LOCATE 18,26 : PRINT "To begin again, press 'RUN' [ F2 ]"
280 K = Ø
290 CALL K (TP%, TL%, TY%)
300 OUT 67,54 : OUT 64,Ø : OUT 64,Ø
310 LPRINT CHR$(27); "@"; : END
320 DATA &h55,&h8B,&hEC,&h8B,&h76,&hØA,&h8B,&hØ4
330 DATA &h5Ø,&h8B,&h76,&hØ8,&h8B,&hØ4,&h5Ø,&h8B
340 DATA &h76,&hØ6,&h8B,&hØ4,&h5Ø,&hB8,&h17,&h2F
350 DATA &h8E,&hD8,&h8F,&hØ6,&hØ9,&hØØ,&h8F,&hØ6
360 DATA &hØ7,&hØØ,&h8F,&hØ6,&hØ5,&hØØ,&hBA,&hØØ
370 DATA &hØØ,&hB4,&hØØ,&hBØ,&h1B,&hCD,&h17,&hB4
380 DATA &hØØ,&hBØ,&h41,&hCD,&h17,&hB4,&hØØ,&hBØ
390 DATA &hØ8,&hCD,&h17,&hB4,&hØØ,&hBØ,&h1B,&hCD
400 DATA &h17,&hB4,&hØØ,&hBØ,&h32,&hCD,&h17,&hB4
410 DATA &hØØ,&hCD,&h16,&h3C,&h67,&h74,&hØ6,&h3C
420 DATA &h47,&H74,&hØ2,&h75,&hF2,&hB9,&hFF,&hØØ
430 DATA &hBA,&hØ1,&hØ2,&hEC,&h3C,&h8Ø,&h76,&hF8
440 DATA &hE2,&hF6,&hbØ,&h36,&hE6,&h43,&hE9,&h5C
450 DATA &hØ1,&hBØ,&h1B,&hE8,&hF2,&hØØ,&hBØ,&h4C
460 DATA &hE8,&hED,&hØØ,&hBØ,&hCØ,&hE8,&hE8,&hØØ
470 DATA &hBØ,&hØ3,&hE8,&hE3,&hØØ,&hB4,&hØ1,&hB9
480 '
490 DATA &hØØ,&hØØ,&hBA,&hØØ,&hØØ,&hCD,&h1A,&hBØ
500 DATA &hØØ,&hA2,&hØØ,&hØØ,&hB9,&hCØ,&hØ3,&h8B
510 DATA &h3E,&hØ1,&hØØ,&hBA,&hØ1,&hØ2,&hEC,&hDØ
520 DATA &hDØ,&hDØ,&h55,&h6Ø,&h47,&h51,&h8B,&hØE
530 DATA &hØ5,&hØØ,&hE2,&hFE,&h59,&hE2,&hEC,&hE8
540 DATA &hA1,&hØØ,&hB9,&hCØ,&hØ3,&h8B,&h3E,&hØ1
550 DATA &hØØ,&hBA,&hØ1,&hØ2,&hEC,&hDØ,&hDØ,&hDØ
560 DATA &h55,&h6Ø,&h47,&h51,&h8B,&hØE,&hØ5,&hØØ
570 DATA &hE2,&hFE,&h59,&hE2,&hEC,&hB8,&hØØ,&hØØ
580 DATA &h8B,&h3E,&hØ3,&hØØ,&hAØ,&hØØ,&hØØ,&hB2
590 DATA &hFØ,&hF6,&hE2,&hØ1,&hC7,&hBA,&hØØ,&hØØ
600 DATA &hB9,&hFØ,&hØØ,&h8A,&h45,&h6Ø,&hE8,&h7D
610 DATA &hØØ,&h47,&hE2,&hF7,&hFE,&hØ6,&hØØ,&hØØ
620 DATA &h8Ø,&h3E,&hØØ,&hØØ,&hØ4,&h74,&hØ5,&hE8
630 DATA &h59,&hØØ,&hEB,&h98,&hBØ,&hØD,&hE8,&h67
640 DATA &hØØ,&hBØ,&hØB,&hE8,&h62,&hØØ,&hA1,&hØ1
650 '
660 DATA &hØØ,&h8B,&hØE,&hØ3,&hØØ,&hA3,&hØ3,&hØØ
670 DATA &h89,&hØE,&hØ1,&hØØ,&hB4,&hØ1,&hCD,&h16
680 DATA &h74,&hØC,&hBØ,&hØD,&hE8,&h49,&hØØ,&hBØ
690 DATA &hØA,&hE8,&h44,&hØØ,&hEB,&h1A,&hBØ,&h1B
700 DATA &hE8,&h3D,&hØØ,&hBØ,&h4C,&hE8,&h38,&hØØ
710 DATA &hBØ,&hCØ,&hE8,&h33,&hØØ,&hBØ,&hØ3,&hE8
720 DATA &h2E,&hØØ,&hE8,&h16,&hØØ,&hE9,&h4F,&hFF
730 DATA &hBØ,&hØE,&hE8,&h23,&hØØ,&hE8,&h3E,&hØØ
740 DATA &h9Ø,&h9Ø,&h9Ø,&h8C,&hDØ,&h8E,&hD8,&h5D
750 DATA &hCA,&hØ6,&hØØ,&hB4,&hØØ,&hCD,&h1A,&h3B
760 DATA &h16,&hØ7,&hØØ,&h75,&hF6,&hB4,&hØ1,&hBA
```

[1] E. Schwittek, "HF WEFAX On the IBM PC," Technical Correspondence, *QST*, Dec 1986, pp 46-47.
[2] K. Sueker, "Real-Time HF WEFAX Maps on a Dot-Matrix Printer," *QST*, Mar 1986, pp 15-20.

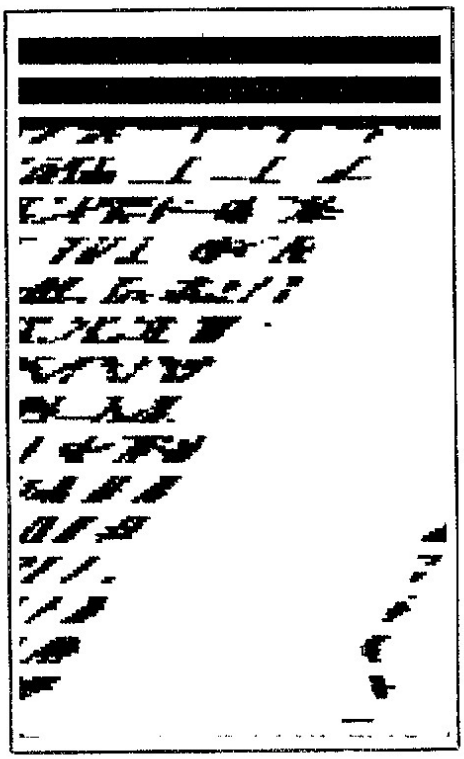
execution) to 20 ticks per second, you can use all the standard BIOS routines for the entire program. This helps the program work with other IBM PC-compatible computers, and with most any 8-dot-high dot-matrix printer that uses the CHR$(27); "L"; CHR$(n1); CHR$(n2) format for commanding 960-dot, graphics-mode printing. Of course, the printer must accept data fast enough to finish before the next data line is ready to start. If it doesn't, you can move the start of the next line out by another ½ second (10 ticks), or use Sueker's trick of force-feeding the printer to speed it up a little.

The program shown in Fig 1 is designed to be used with the receiver in the USB mode. Memory above the first 128-kbytes of RAM is used to store the machine-language code so it is compatible with the PC and PCjr. (The PCjr has about half the execution speed below this area as it does above.) If your machine has only 128 kbytes of RAM, change line 160 to DEF SEG = &h1F00, and change the last item in line 340 from &h2F to &h1F. Doing this with the PCjr will require the pixel count (line 120) to be changed to 60, and depending on your printer speed, may require a longer line timing in line 130, or a TP% of less than 60. This program has been checked out with two IBM-compatible printers, the Seikosha SP-1000I and Okidata's Okimate 20. The printer DIP switches are set to eliminate automatic line feeds.

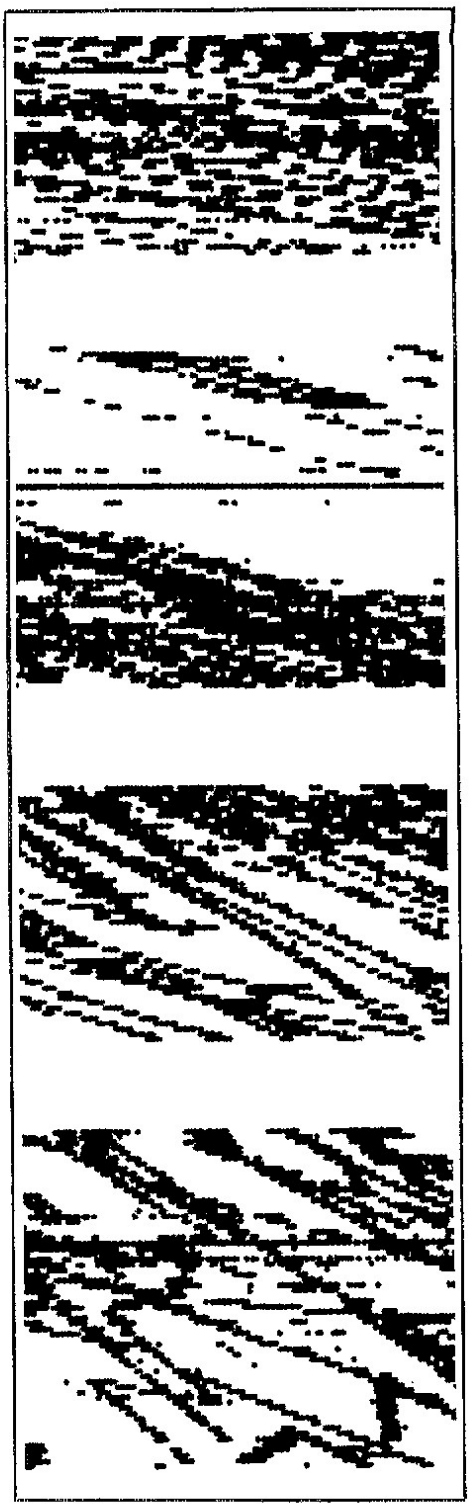
Some IBM PC-compatibles may require adjustment of the pixel dwell time, TP%. Also, the clock divisor value may need trimming. The divisor's LSB is the 7th data item (&h0B) in line 910, and its MSB is the third data item (&hE9) in line 920. That is, the divisor is &hE90B (59659 decimal). If the picture leans to the right, the divisor is too large—make it smaller. If the picture leans to the left, make the divisor larger. With an adjustment fineness of 1 part in approximately 60,000, you can get right on. Incidentally, they don't always get the pictures straight on the sending machine, so check for right angles on grid lines as a final check.

There are some programming tricks that are easy to do. You can change the pixel dwell time downward and go every other line (TP% = 60 to 85, TL% = 20) and get a blown-up view of part of the map with better resolution. In a practical sense, the usable resolution is often limited by propagation anomalies (multipath), however. Incidentally, you can connect to port &h201 of the computer by attaching the wires from the demodulator across the black push button Joystick B contacts on an IBM joystick, being careful to locate the ground connection. This relieves you from having to find a DB15 connector.

I also have Commodore 64™ WEFAX programs available for both 8-dot-high (480 and 960 dots across) and 7-dot-high (480 dots across) printers. These programs include disk storage modes, screen modes and real-time printing modes. The 8-dot modes are for the same IBM-compatible printers mentioned earlier; the printers are connected to the C64



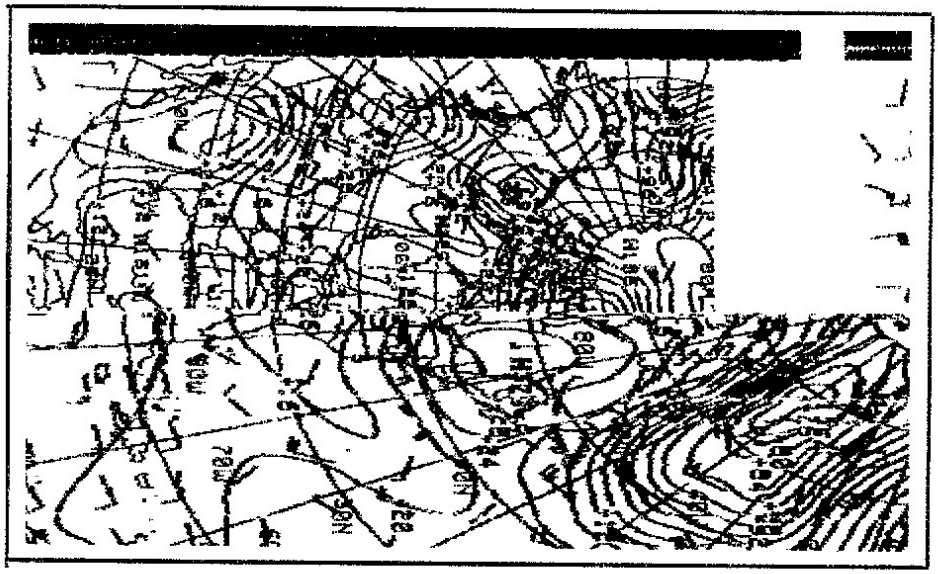
This picture section is leaning to the right, and unwanted line feeds were inserted during reception. The extraneous line feeds were caused by the presence of an unneeded—for the Epson FX-286—printer command; see text.



The top part of this photo shows the → picture section now leaning to the left and squashed because of a change in the MSB of the clock divisor. The next three picture slices depict how altering the LSB of the clock divisor begins to open up the picture.

```
770 DATA &h00,&h00,&hCD,&h1A,&hC3,&h90,&h34,&hFF
780 DATA &hB4,&h00,&hCD,&h17,&hC3,&h00,&h00,&h00
790 DATA &h00,&h00,&h00,&h00,&h00,&h00,&h00,&h00
800 DATA &h00,&h00,&h00,&hE8,&h03,&h00,&h00,&h00
810 DATA &h00,&h00,&h00,&h00,&h00,&h00,&hB0,&h20
820 '
830 DATA &hE8,&hDD,&hFF,&hB0,&h20,&hE8,&hD8,&hFF
840 DATA &hB0,&h4B,&hE8,&hD3,&hFF,&hB0,&h33,&hE8
850 DATA &hCE,&hFF,&hB0,&h42,&hE8,&hC9,&hFF,&hB0
860 DATA &h43,&hE8,&hC4,&hFF,&hB0,&h20,&hE8,&hBF
870 DATA &hFF,&hB0,&h57,&hE8,&hBA,&hFF,&hB0,&h45
880 DATA &hE8,&hB5,&hFF,&hB0,&h46,&hE8,&hB0,&hFF
890 DATA &hB0,&h41,&hE8,&hAB,&hFF,&hB0,&h58,&hE8
900 DATA &hA6,&hFF,&hB0,&h0D,&hE8,&hA1,&hFF,&hB0
910 DATA &h0A,&hE8,&h9C,&hFF,&hC3,&hB0,&h0B,&hE6
920 DATA &h40,&hB0,&hE9,&hE6,&h40,&hE9,&h99,&hFE
```

Fig 1—WFX, an HF WEFAX program for the IBM PC and PCjr. As shown, this program works with the Seikosha SP-1000I, Okidata Okimate 20 and IBM Proprinter printers. With some modifications (see text), the program also works with the Epson FX-286 operating in the Epson mode; other FX models should also work with the changes shown. Depending on the computer clock speed, one or two program byte values may have to be changed (see text) to obtain proper line timing. Similar programs, written in machine language and designed for use with the C64, are available from the author.



← The divisor is now correct, and the picture lines are vertical. However, this picture was received out of sync with the sending station. That accounts for the gap off to the right at the top of the picture, and the small section of picture information on the right-hand side. Also, the top part of this picture was captured *while using the* standard clock speed of the computer; turning on the turbo mode expands the picture horizontally as shown at the bottom. The terms "horizontal" and "vertical" might be confusing here. That's because many pictures (such as this one) are received rotated 90 degrees. If you want to save a particular picture received out of sync, you can "cheat." Use a sharp knife or razor blade to cut the picture, then paste it together on another sheet of paper.

Using WFX With the Epson FX-286 and IBM Proprinter

When Ben's program and accompanying printouts arrived, I was so impressed, I couldn't wait to try the program with my setup! Would it work? I have an Epson FX-286 printer and a PC clone, a Microproducts International XPC-XT turbo model. By setting three DIP switches, the FX-286 is capable of operating in two modes: Epson and IBM Proprinter. The computer's turbo mode allows it to run approximately 40% faster than a standard IBM PC.

Initial results proved only slightly disappointing. First of all, I was getting unwanted line feeds between the printed lines. I determined that was caused by the differences in the printer codes required by the Epson and IBM Proprinter. Switching to the Proprinter mode eliminated the extraneous line feeds. Once that was taken care of, I saw that although the copy was good, the entire picture was leaning to the right. (This indicated a general timing problem, one not related to the use of standard PC speed or turbo speed.) Changing the value of the seventh byte in line 910 (the low-order byte of the clock divisor) had virtually no effect in eliminating the skew. I then changed the value of the third byte in line 920 (the high-order byte of the clock divisor). I knew I'd hit the jackpot because the picture now slanted to the left! Following some empirical determination, I arrived at values of &hE8 for the high-order byte and &hEC for the low-order byte of the divisor. The picture lines are now perfectly vertical.

If your pictures are slanted one way or the other, you may need to use different byte values than those in the original program or those I'm using. (See the accompanying figures for sample printouts I obtained during experimentation with WFX.) First, try manipulating the value of the low-order byte. If that doesn't bring the desired results, alter the second byte value, then play with the value of the low-order byte until the vertical lines in the picture are straight.

The next step was getting the program to work with the FX-286 in the Epson mode. That was done by changing bytes 4 through 8 in line 390 and bytes 1 through 7 in line 400 to &h90. The &h90 is a NOP code (a do-nothing) that replaces the ESC 2 printer command included in the original program. With the Epson FX-286, the presence of this command causes extraneous line feeds (it's the 1/6-inch line feed command) to be generated even with the DIP switch set to allow carriage returns only. The modification described here should permit WFX to work with printers that operate similarly.

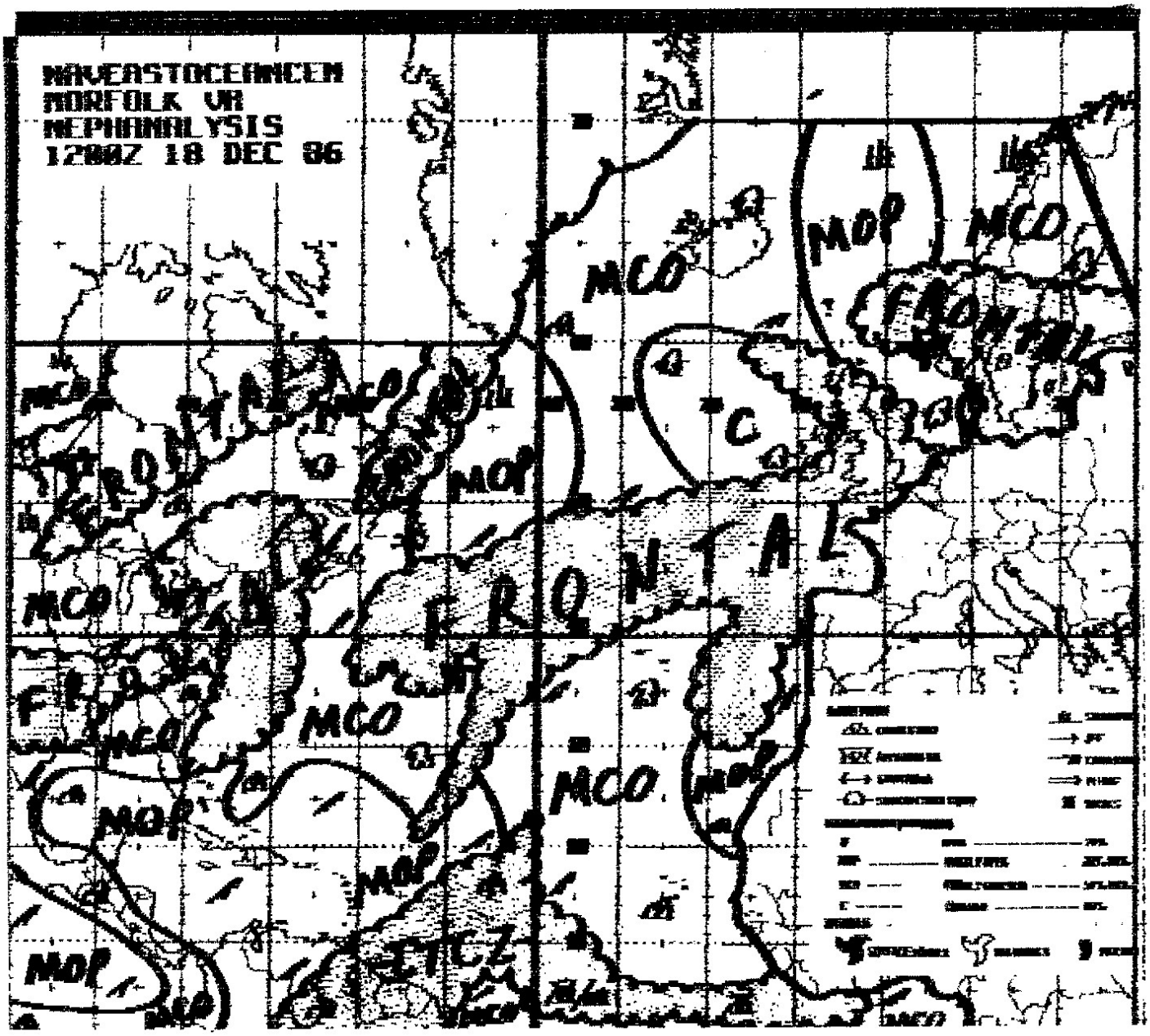
Ben supplied the program operating under DOS 2.1. I tried WFX using IBM PC DOS 2.1 and 3.1, BASIC and BASICA. All combinations worked without a hitch. It's *not* necessary to load GRAPHICS.COM before running WFX. Let me warn you that some TSR (terminate and stay resident) programs such as Borland's SideKick won't coexist peacefully with WFX in its unmodified form. During initial tests, you should boot with a "clean" AUTOEXEC.BAT file (don't load any TSR programs). Once you have WFX running properly, you can experiment to see which TSR programs will behave themselves with WFX. You can get WFX and SideKick to get along by changing line 160 to DEF SEG = &h1F00 and the last byte in line 340 to &h1F.

Kicking my PC clone into turbo mode simply expands the picture horizontally (or vertically, if the picture is rotated 90 degrees as many are). Such a procedure won't provide a correct aspect ratio, however, and circles will become ellipses. Using turbo mode is a "quick and dirty" way of getting some magnification, though, and doesn't require any program changes. With TP% = 84 and TL% = 20, a properly magnified section of the picture is obtained.

Make sure you have a good supply of paper and ribbons on hand. If you do a lot of WEFAXing, you'll need 'em! Because the pictures are created slowly, you'll need to set aside a period of free time to experiment with the program, if you're not immediately successful. That free time should be scheduled during periods of good signal reception. For instance, I've found the hours between 10 AM and 2 PM Eastern to be usually reliable for picture reception on 8080 kHz during the winter months (this is being written in January). But, I've occasionally had some good results during early morning and evening hours, too, on 3357 and 4271 kHz. If you have a transceiver with memories, store some WEFAX frequencies in a few of them for easy call-up at any time (see Sueker's article for a list of frequencies).

A few seconds may elapse between the time you press the G key and printing takes place, if your printer is equipped with a large buffer. Lastly, if you're lucky enough to have a four-character call sign, you can readily substitute it for Ben's. We'll leave that little exercise for you to work out.

The demodulator described by Sueker is available from A & A Engineering, 2521 W La Palma Ave, Unit K, Anaheim, CA 92801, tel 714-952-2114. A change has been made to the original circuit to allow it to be used with virtually all types of computers. Bare boards are $9.85; kits, $28.15; assembled and tested units, $39.95. If you need a power supply for this project or others, A & A also sells several models; the no. 137 ($32.95) is recommended by A & A for use with the demodulator. (See New Products, Feb 1987 *QST*, p 43). Please add $2.50 for shipping and handling charges.
—Paul K. Pagel, N1FB, ARRL HQ



Finally! One example of what we've been looking for!

through a Cardco (+G) interface. While I was unable to stick to standard printer routines in the real-time modes, thereby lowering the probability of the programs working with all printers, I do have real-time modes that have worked with the Gorilla Banana, TRP-100, DMT-105, Seikosha SP-1000I and Okimate 20 with various combinations of Cardco and Xetec printer interfaces. So far, getting the MPS-803 printer to work in real time has eluded me, but that printer does work in all other modes. Since it's unwieldy to publish all these programs, I offer them on disk for a $5 copying and mail fee, or your disk and a stamped, self-addressed disk mailer.

If you already have the Schwittek program on disk, it is probably easier to enter the IBM PC BASIC program listing of Fig 1 as a modification. I would appreciate any feedback on other computer and printer combinations that work with this program.—*Ben Vester, K3BC, 4921 Bonnie Branch Rd, Ellicott City, MD 21043*

Note: All correspondence addressed to this column should bear the name, call sign and complete address of the sender. Please include a daytime telephone number at which you may be reached if necessary.

Feedback

☐ Author Mike Masterson has forwarded some reader feedback on his article, "Three Fine Mice—MOuSeFET CW Transmitters," Dec 1986 *QST*. On p 20, the text in the second column, eighth line, should read "...voltages and acts with C15 to provide a..." Also, C15 of Fig 1 (pp 20-21) is the 0.1 μF capacitor in series with the gate lead of Q6, and is located between R3 and the 2.7 kΩ resistor. Lastly, Mike notes that he wound L1 (for 40 meters) with no. 36 enameled wire (not no. 28), and that it would be easier to use no. 36 enameled wire for both 80- and 40-meter L1 coils.

☐ Author Al Ward has pointed out an error in his article, "Monolithic Microwave Integrated Circuits, Part 2," Mar 1987 *QST*. On page 27, Fig 12, the gain curve labels are off by 10 dB. A corrected Fig 12 is shown here.