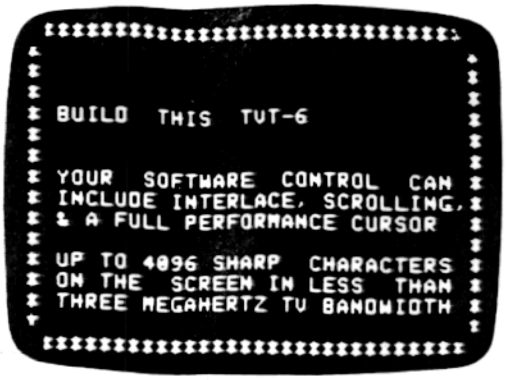


BUILD THE TVT-6 Part II

System debugging, software, and how to interface to other processors.

BY DON LANCASTER

LAST MONTH, we discussed construction of the TVT-6 TV typewriter and explained how it works and how it is connected to a KIM-1 microcomputer. We also started a discussion of the operating secrets of the TVT-6. Here, we complete the "secrets" discussion and go on to system debugging, some useful programs, and tell you how to interface the TVT-6 with other microprocessors.

Software. Four examples of tested, annotated, and workable KIM-1 software are given in the tables in this article. Table II contains a 16 × 32 scan program with full interlace. It automatically generates almost all the timing required by the TVT-6 and its companion TV monitor for this display format. The program is run by jumping to memory location 17Ad. The display is stopped by interrupting with the operating system, the cursor, or other program.

Table III is an optional full-performance cursor for the 16 × 32 system and includes scrolling, full cursor motion, and erase-to-end-of-screen capabilities. It is run by allowing the keypress signal from the keyboard to interrupt the scan program (any of the three Tables) via the IRQ interrupt line. Note that the cursor program is totally independent of the SCAN program. The only things the two programs share in common are the same pages of display memory. The screen-read-to-cassette can be performed using the existing KIM-1 operating system programs. You can also load from cassette to display, using the automatic search firmware.

Table IV is a 16-line/64-character scan program that requires only 64 words to be written into memory for the entire program. This program can be used to display the entire 1k of minimum KIM-1 memory for use as a super front-panel display if desired. For display-only applications, 1k of contiguous memory

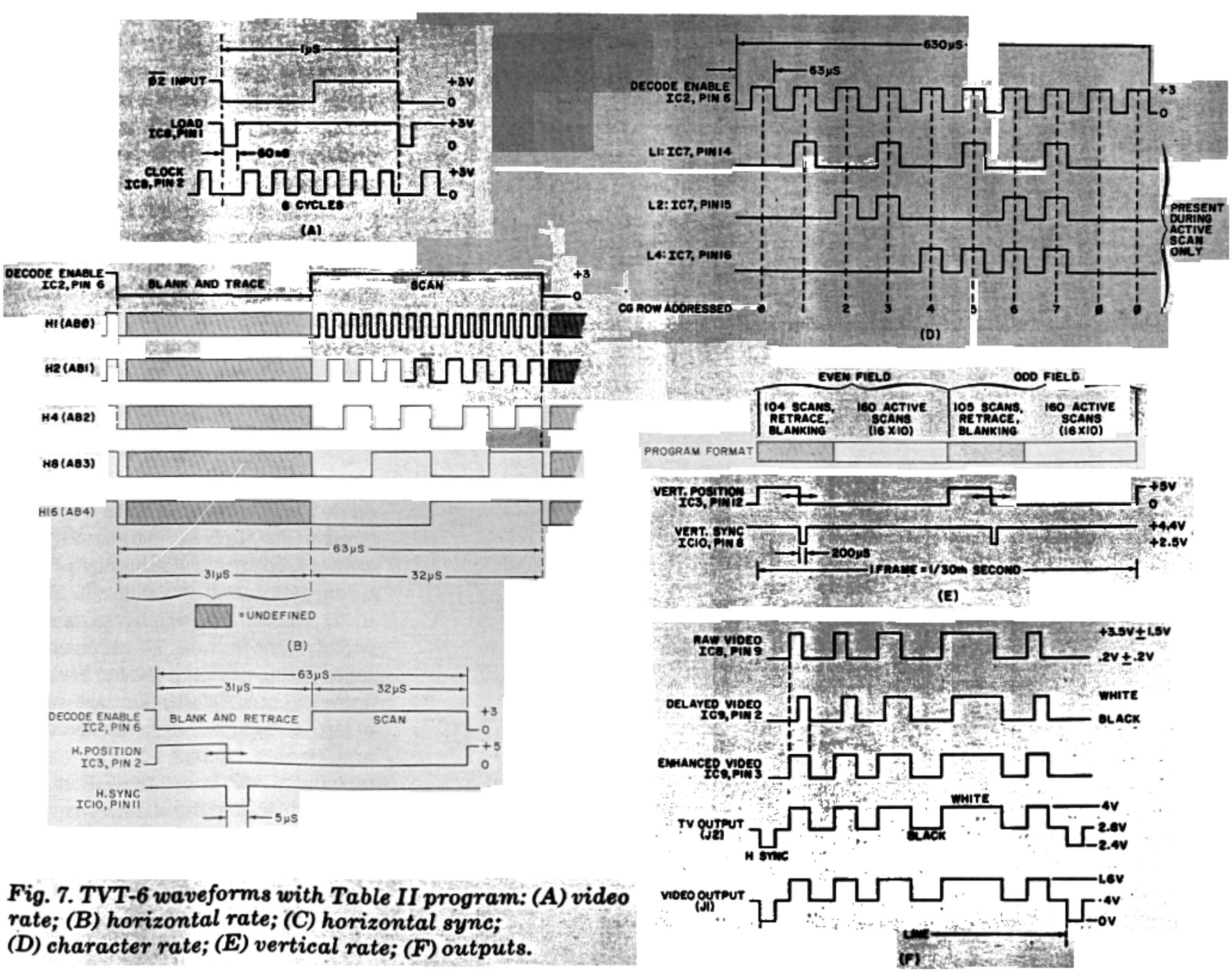


Fig. 7. TVT-6 waveforms with Table II program: (A) video rate; (B) horizontal rate; (C) horizontal sync; (D) character rate; (E) vertical rate; (F) outputs.

TABLE II

16 line X 32 character per line Interlaced
TVT6 Raster Scan:

```
µP - 6502          Start - JMP 17Ad       Displayed 0200-03FF
System - KIM-1     End - Interrupt        Program Space 1780-17E2

|HS|VS|L4|L2|L1|0|1|V8|   |V4|V2|V1|H16|H8|H4|H2|H1|
     Upper Address               Lower Address

        1780  NOP  EA                  Equalize 2 cycles
        1781  STA  8d  (8A)   (17)     Store upper address
        1784  PHA  48                  Equalize 10 cycles
        1785  PLA  68                  Continued

        1786  BNE  d0  00              Continued
        1788  JSR  20  00     80       ////Character Scans 1-8////
        178b  ADC  69  08              Increment Character Scan Counter
        178d  CMP  C9  C0              Is VS = 1?

        178F  BCC  90  F0*             No, do next character scan
        1791  JSR  20  (E0)   (17)     Equalize 15 cycles via sub
        1794  JSR  20  00     80       ////Character Scan 9////
        1797  TAX  AA                  Save Upper Address

        1798  LDA  Ad  (89)   (17)     Get Lower Address
        179b  ADC  69  1F              Increment L; Set C on V4 overflow
        179d  STA  8d  (89)   (17)     Restore lower word; save carry
        17A0  TXA  8A                  Get Lower Word

        17A1  BNE  d0  00              Equalize 5 cycles
        17A3  NOP  EA                  continued
        17A4  ADC  69  C0              Add carry; Reset VS
        17A6  JSR  20  00     80       ////Character Scan 10////

        17A9  CMP  C9  84              Is it line "17"?
        17Ab  BCC  90  d3*             No, continue character scans
-START->17Ad  LDA  Ad  (dF)   (17)     Get Interlace word
        17b0  EOR  49  80              Change field

        17b2  BMI  30  05*             Jump if even field
        17b4  STA  8d  (dF)   (57)     Odd Field V Sync; Restore Interlace word
        17b7  LDX  A2  66              Load short number of VB scans
        17b9  JSR  20  (E0)   (17)     Equalize 15 cycles via sub

        17bC  JSR  20  (E0)   (17)     Equalize 15 cycles via sub again
        17bF  BPL  10  05*             Jump if odd field
        17C1  STA  8d  (dF)   (57)     Even Field V Sync; restore interlace
        17C4  LDX  A2  67              Load long number of V Blank scans

        17C6  JSR  20  1E     80       ////1st V blanking scan ////
        17C9  CLD  d8                  Equalize 9 cycles
        17CA  PHA  48                  Continued
        17Cb  PLA  68                  Continued

        17CC  LDA  A9  00              Initialize lower address
        17CE  STA  8d  (89)   (17)     Continued
        17d1  LDA  A9  82              Initialize upper address
        17d3  STA  8d  (8A)   (17)     Continued

        17d6  JSR  20  00     80       ///Remaining V Blanking scans////
        17d9  CLC  18                  Initialize carry
        17dA  DEX  CA                  One less scan

        17db  BMI  30  A4*             Start Character scan
        17dd  BPL  10  Ed*             Repeat Vertical blanking scan
        17dF       80                  Interlace word storage
        17E0  BCS  b0  00              ///Equalize 15 SUBROUTINE ////

        17E2  RTS  60                  Continued.
```

NOTES: TVT6 must be connected and scan microprogram PROM (IC1) must be in circuit for program to run.

Both 17b4 and 17C1 require that page 17 be enabled when page 57 is addressed. This is done automatically with KIM-1 circuitry.

Step 1788 goes to where the upper address stored in 178A and the lower address stored in 1789 tells it to. Values in these slots continuously change throughout the program.

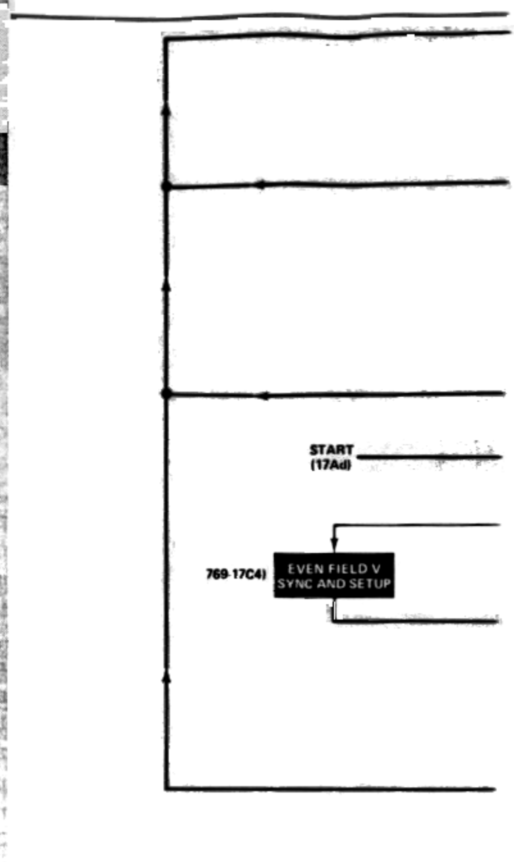
For a 525-line system, use 17b8 64 and 17C5 65 and a KIM-1 crystal of 992.250 kHz. This is only needed for video superposition and titling applications.

Normal program horizontal frequency 15,873.015 Hz; Vertical frequency 60.0114 Hz. 63 us per line; 264.5 lines.

* Denotes a relative branch that is program length sensitive.

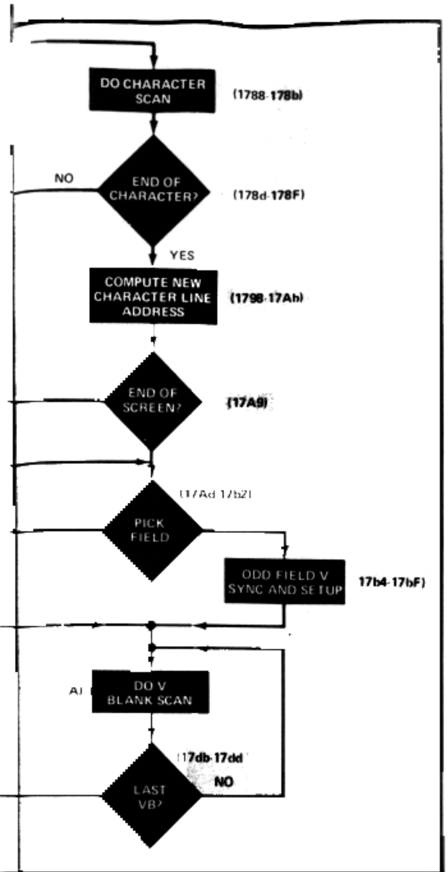
( ) Denotes an absolute address that is program location sensitive.

TVT6 length jumper must be in "32" position.



is required. Keep in mind that the KIM-1 has some operating system slots in the top of page zero and the stack at the top of page one. Unless you actually want to display the stack and operating system parameters, do not use these slots.

The 64-character line makes the TV receiver's horizontal frequency run considerably lower than normal. This will require a readjustment of the horizontal-hold control or some extra capacitance across the existing horizontal-hold capacitor. The width of the raster may also have to be reduced; this is most easily accomplished by adding a low-value inductor in series with the yoke. These changes are best made in a small-screen, *transformer-powered* monochrome TV receiver. The tradeoff of a lowered horizontal frequency produces a long character line but still allows 1 µs/character. This will not tax the bandwidth restrictions of TV receivers or r-f modulators. (Editor's Note: The small-screen Sears TV receiver we used required adjustment of horizontal size and linearity, a 0.033-µF Mylar capacitor in parallel with the 0.068-µF capacitor used for *C408* in the receiver, and an inductor consisting of 60 turns of No. 24 enameled wire on a ½" Nylon form in series with the red yoke lead in the receiver. In addition, it was necessary to disconnect one side of *C201* in the receiver



to defeat the sound trap. *Never attempt to modify a TV receiver that is powered directly from the ac line without an isolating transformer.*)

Table V contains a program that we call "Cruncher the Bear." This program produces 64 fully interlaced characters in each of 32 rows, for a total of 2048 sharp ASCII characters on-screen at one time within the 3-MHz bandwidth. You can add a hex-to-ASCII converter that slowly sequences high- and low-order machine code characters in the same slot and end up with 4096 hex characters displayed in only 3 MHz of bandwidth.

Table V requires a contiguous 2k of memory with a common upstream tap and separate chip enables. However, it is easily incorporated if you really want or need to display as many characters as the program allows.

Other software is easily written and developed for the TVT-6. For example, you may wish to have a 32 × 44 or a 32 × 48 character display and still use normal, or nearly normal, horizontal scanning rates. This allows for video titling and superimposition, oversize characters, color graphics, lower-case characters, and game displays. There is no lower limit to the number of character rows or characters per line you can use. If you have limited memory available,

Table III

16 X 32 Full-performance Cursor:

```
µP -- 6502        Start -- IRQ    Displayed 0200-03FF
System -- KIM-1   End   --- RTI   Program Space 0100-01dF

Input to Parallel Word A  | Ø | A7 | A6 | A5 | A4 | A3 | A2 | A1 |   IRQ
                                                            -->| |<-- 10 µs

Clear - CAN (18)                  Cursor Home - SOH (ØA)
Carriage Return - CR (Ød)         Scroll Up - DC1 (11)
Cursor Up - VT (Øb)               Erase to End - DC2 (12)
Cursor Down - LF (ØA)             Spare Hook - DC3 (13)

Cursor Left - BS (Ø8)             Enter -- All characters
Cursor Right - HT (Ø9)            Ignore -- All other CTRL

Enter via
  IRQ
   └──> 0100  PHA  48                    Save A
        0101  LDY  A0  00                Reset Y Index
        0103  LDA  A5  (EE)              Get Cursor and test for range
        0105  CMP  C9  03                Is cursor on page 3?

        0107  BEQ  FO  04*               Yes, OK to continue
        0109  CMP  C9  02                Is cursor on page 2?
        010b  BNE  d0  3A*               No, Home cursor
        010d  LDA  b1  (Ed)              Get old cursored character
0147
        010F  AND  29  7F                Erase old cursor
        0111  STA  91  (Ed)              Replace character without cursor
        0113  LDA  Ad  00      17        Get new character from A parallel Int.
        0116  CMP  C9  20                Is it a character to be entered?

0142 <- 0118  BCS  b0  28*               Yes, go and enter character
        011A  CMP  C9  18                Clear Screen?
015E <- 011C  BEQ  FO  40*               Yes, clear screen
        011E  CMP  C9  0d                Return Carriage?

0152 <- 0120  BEQ  FO  30*               Yes, Return carriage
        0122  CMP  C9  0b                Cursor Up?
0194 <- 0124  BEQ  FO  6E*               Yes, Up Cursor
        0126  CMP  C9  0A                Cursor Down?

0166 <- 0128  BEQ  FO  3C*               Yes, Down Cursor
        012A  CMP  C9  09                Cursor Right?
0158 <- 012C  BEQ  FO  2A*               Yes, Right Cursor
        012E  CMP  C9  08                Cursor Left?

01A7 <- 0130  BEQ  FO  75*               Yes, Left Cursor
        0132  CMP  C9  01                Cursor Home?
0147 <- 0134  BEQ  FO  11*               Yes, Home Cursor
        0136  CMP  C9  11                Scroll Up?

0175 <- 0138  BEQ  FO  3b*               Yes, Scroll Up
        013A  CMP  C9  12                Spare Hook?
014A <- 013C  BEQ  FO  0C*               Ignore--Restore Cursor
        013E  CMP  C9  13                Erase to EOS?

01b1 <- 0140  BEQ  FO  6F*               Yes, Erase to EOS
        0142  JSR  20  (D3)    (01)      /////Enter Character/////
        0145  BNE  d0  03*               End of Screen?
        0147  JSR  20  (C2)    (01)      Yes, Home Cursor

        014A  LDA  b1  (Ed)              ////Restore Cursor/////
        014C  ORA  09  80                Add cursor to cursored character
        014E  STA  91  (Ed)              Replace cursored character
        0150  PLA  68                    Get A

<-Out-- 0151  RTI  40                    Return to Scan
        0152  LDA  A5  (Ed)              ////Carriage Return/////
        0154  ORA  09  1F                Move Cursor to Right End
        0156  STA  85  (Ed)              Restore Cursor

        0158  JSR  20  (d5)    (01)      Increment Cursor
        015b  JMP  4C  (45)    (01)      Finish
        015E  JSR  20  (C2)    (01)      ////Clear////Home Cursor
        0161  JSR  20  (Cb)    (01)      Clear Screen

0147 <- 0164  BEQ  FO  E1*               Finish
        0166  LDA  A5  (Ed)              ////Cursor Down//// Get Cursor
        0168  CLC  18                    Clear Carry
        0169  ADC  69  20                Move Cursor Down

        016b  STA  85  (Ed)              Restore Cursor
        016d  BCC  90  03*               Overflow of page?
        016F  JSR  20  (d9)    (01)      Yes, increment upper page
        0172  JMP  4C  (45)    (01)      Finish

        0175  JSR  20  (C2)    (01)      ////Scroll Up//// Home Cursor
        0178  LDY  A0  20                Add Offset to Index
        017A  LDA  b1  (Ed)              Get Offset Indexed Character
        017C  LDY  A0  00                Remove Offset from Index

        017E  JSR  20  (d3)    (01)      Enter Moved Character and Increment
        0181  BNE  d0  F5*               Repeat?
        0183  CLC  18                    Clear Carry
        0184  LDA  A9  01                Set A to page 3

        0186  STA  85  (EE)              Set Cursor to Page 3
        0188  LDA  A9  E0                Set A to start of last line
        018A  STA  85  (Ed)              Set Cursor to Start of last line
014A <- 018C  BCS  b0  bC*               Finish if carry set
```

(Continued on next page.)

Table III (Continued)

```
         018E  JSR  20 (Cb)  (01)  Clear last line
         0191  SEC  38             Set Carry
         0192  BCS  b0  F0*        Restore Cursor to start of last line
         0194  LDA  A5 (Ed)        ////Cursor Up/////Get Cursor

         0196  SEC  38             Set Carry
         0197  SBC  E9  20         Move up one line
         0199  STA  85 (Ed)        Restore Cursor
014A <-- 019b  BCS  b0  Ad*        Underflow of page?

         019d  DEC  C6 (EE)        Yes, decrement page
         019F  LDA  A9  01         Set A to Page 1
         01A1  CMP  C5 (EE)        Did screen underflow?
014A <-- 01A3  BNE  d0  A5*        No, Finish

0147 <-- 01A5  BEQ  F0  A0*        Yes, Home Cursor
         01A7  DEC  C6 (Ed)        ///Cursor Left////Decrement Cursor
         01A9  LDA  A9  FF         Set A to page underflow
         01Ab  CMP  C5 (Ed)        Test for page underflow

019d <-- 01Ad  BEQ  F0  EE*        Change Page if off Page
014A <-- 01Af  BNE  d0  99*        Finish if on page
         01b1  LDA  A5 (EE)        ////Erase to EOS///Get Cursor
         01b3  PHA  48             Save Upper cursor location

         01b4  LDA  A5 (Ed)        Get lower cursor location
         01b6  PHA  48             Save lower cursor location
         01b7  JSR  20 (Cb)  (01)  Clear to End of Screen
         01bA  PLA  68             Get lower cursor location

         01bb  STA  85 (Ed)        Restore lower cursor
         01bd  PLA  68             Get upper cursor location
         01bE  STA  85 (EE)        Restore upper cursor
014A <-- 01C0  BNE  d0  88*        Finish


         01C2  LDA  A9  00         ///SUB//Home Cursor///
         01C4  STA  85 (Ed)        Set lower cursor to zero
         01C6  LDA  A9  02         Put page 2 in A
         01C8  STA  85 (EE)        Set upper cursor to 0200

         01CA  RTS  60             Return to main program
     ┌-> 01Cb  LDA  A9  20         ///SUB//Enter Space///
     │   01Cd  JSR  20 (d3)  (01)  Enter space via Sub
     └-- 01d0  BNE  d0  F9*        Repeat if not to end

         01d2  RTS  60             Return to main program
         01d3  STA  91 (Ed)        ////SUB//Enter,Increment// store
         01d5  INC  E6 (Ed)        Increment Cursor
         01d7  BNE  d0  06*        Overflow?

     ┌-- 01d9  INC  E6 (EE)        Yes, Increment cursor page to 03
     │   01db  LDA  A9  04         Load A with page 4
     │   01dd  CMP  C5 (EE)        Test for Overflow
     └-> 01df  RTS  60             Return to main program
```

NOTES: $\overline{\text{IRQ}}$ vector must be stored in 17FE 00 and 17FF 01.

Total available stack length is 32 words. Approximately 16 are used by operating system, cursor, and scan program. Stack must be initialized to 01FF as is done in KIM-1 operating system. For 30 additional stack locations, relocate subroutines starting at 01C2 elsewhere.

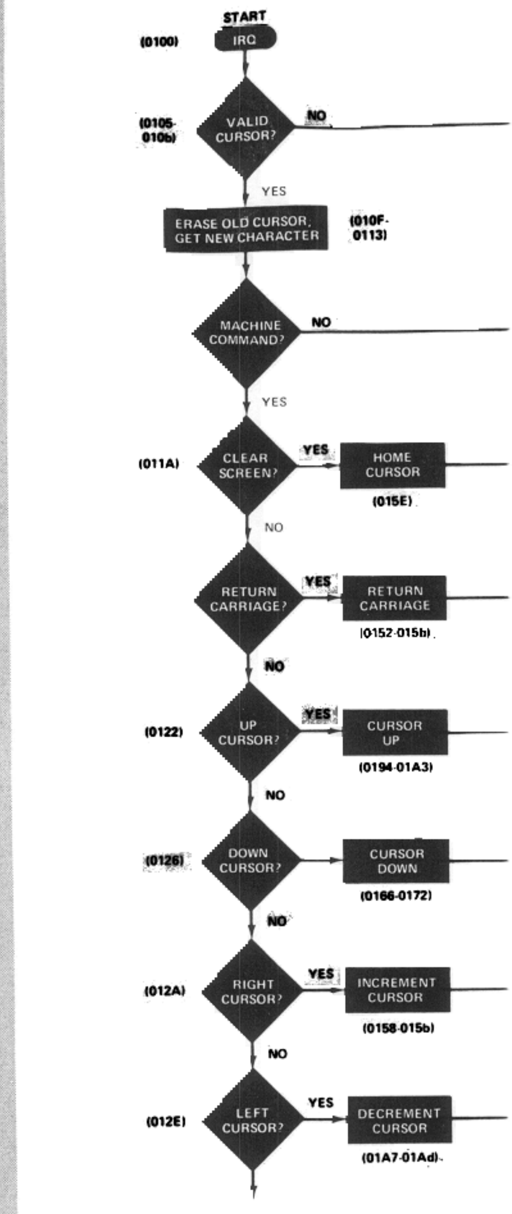
To protect page, load 00F3 04. To enable entry load 00F3 00.

Cursor address is stored at 00Ed low and 00EE high on page zero.

To display cursor load 014d 80. To not display cursor load 014d 00.

* Denotes a relative branch that is program length sensitive.

( ) Denotes an absolute address that is program location sensitive.



you can run 8 × 32, 4 × 64, 1 × 64, or even 1 × 8 character formats. All this takes is software changes, and the circuitry of the TVT-6 remains the same.

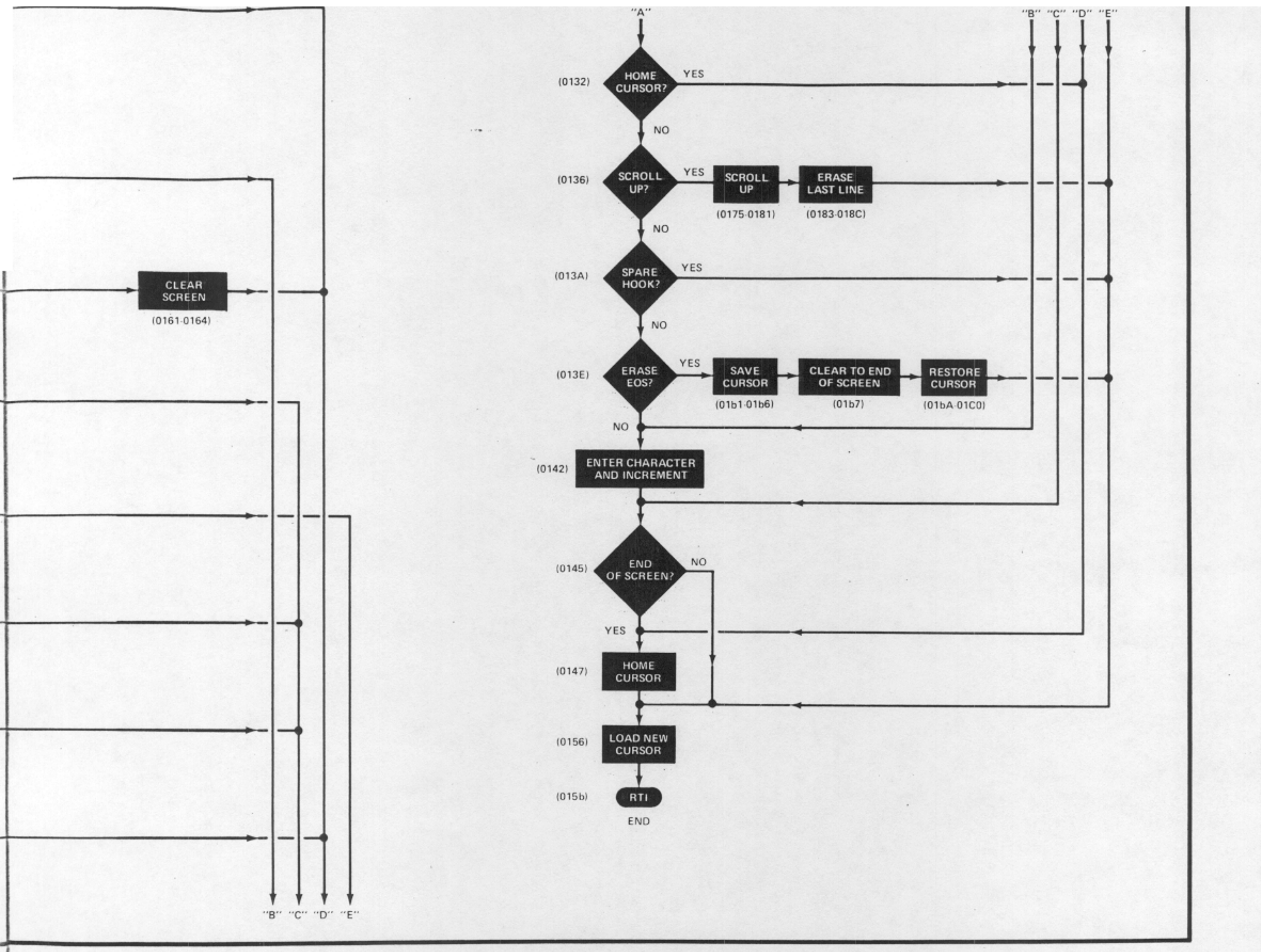
Initial Debugging. At this point, there should be no IC's in the sockets of the TVT-6 board assembly. Start by connecting the LENGTH jumper to 32 and the CURSOR jumper to YES on the TVT-6 board. (Note: These points are pads located at the center of the circuit board, not the edge-connector contacts.) Temporarily insert a jumper wire between pins 3 and 14 on the *IC5* socket. Center the two position control potentiometers and install *IC1*, *IC2*, and *IC6* in their respective sockets.

Connect your video monitor to the TVT-6 board and power up the system. Check for the presence of the SCAN instructions (see PROM Truth Table in Fig. 1 of Part 1) at hex locations 8000 through 8020. Write a simple program that jumps to a subroutine at location 8000 and then loops. Single-step through this program to verify proper operation of the SCAN instruction. Do not

USING THE TVT-6 WITH OTHER POPULAR MICROPROCESSORS

Both parts of this article have used the TVT-6 with the 6502 microprocessor-based KIM-1 microcomputer. Here is how to use the TVT-6 in μC's that use other popular microprocessors.

6800. The 6800 μP is very similar to the 6502 and, therefore, is easiest to convert. The SCAN microprogram can be LDAB(C6) for words 0 through 30 and RTS(39) for word 31. A literal translation of the tightest part of the SCAN program (1D;1782 through 178C) is: STA(B7); JSR(BD); ADDA(8B); CMPA(81); BCC(24). This routine requires 25 μs to cycle through as compared to the 21 μs required for the 6502.



8080. A stock 8080 μP can normally change its program counter once every 2 μs, but it can be "tricked" into doubling its speed *during a SCAN microprogram* by driving the usual address line A9 of the display memory from SYNC. The SCAN microprogram is then NOP(00) for words 0 through 30 and RET(A9) for word 31. A tighter than literal translation of the SCAN program (1D;1782 through 178C) is: STAXB(02); CALL(AD); ADD(82); CMP(BB); JNC(DB), which requires 24 μs to cycle through. Here, the TVT-6 address lines A5 through A1 must be relabelled A4 through AØ, respectively.

Z80. The Z80 μP can use 8080-developed software with speed-doubling scans, or it can simply be run faster, allowing the program counter to change once every microsecond. Use a literal translation of the program for the 6502.

12 Address Line μP's. The four upper address lines of 12 address line μP's can be decoded to allow normal operation, 8 to 12 lines of scan, a vertical sync pulse, an operating return system, and an optional "page-change" command. This leaves a 256-character page on the bottom eight bits, and the "page-change" command can be latched to change to any number of additional pages, as required.

General Hints. Horizontal scan should last at least 62, 63.5, or 64 μs for conventional horizontal-frequency operation. The microprogram scan must *end* exactly this number of microseconds later for each horizontal line in the total scan program. The total number of lines must produce a vertical frequency between 59.9 and 60.1 Hz per field. Note that a portion of the RTS time will be spent during the active (microprogram) scan time. Horizontal scans that last longer than 85 μs may make it difficult to obtain TV interface.

You can shorten a *blank* microprogram active scan by an *even* number simply by jumping ahead when you call your subroutine. For example, a JSR 8000 may produce a 32-character scan, while a JSR 8002 can produce a 30-character scan. This approach can come in handy when there is a need for equalizing scan lengths between character rows and during vertical retrace.

TABLE IV

16 line X 64 character per line TVT6 Raster Scan:

µP - 6502 System - KIM-1	Start - JMP 17AA End - Interrupt	Displayed 0000 - 03FF Program Space 1780-17bE

HS	VS	L4	L2	L1	0	V8	V4	V2	V1	H32	H16	H8	H4	H2	H1

Upper Address — Lower Address

```
        1780  LDA  A9  80            Initialize Upper Address
        1782  STA  8d  (87)  (17)    Store Upper Address
        1785  JSR  20  00    80      ////Character Scans 1-8////
        1788  ADC  69  08            Increment character scan counter

        178A  CMP  C9  C0            Is VS = 1?
        178C  BCC  90  F4*           No, Do next character scan
        178E  TAX  AA                Save Upper Address
        178F  LDA  Ad  (86)  (17)    Get lower address

        1792  BCS  b0  00            Equalize 3 cycles
        1794  JSR  20  04    80      ////Character Scan 9////
        1797  BCS  b0  00            Equalize 3 cycles
        1799  ADC  69  3F            Increment Lower; Set C on V2 overflow

        179b  STA  8d  (86)  (17)    Restore Lower Address; save carry
        179E  TXA  8A                Get upper address
        179F  JSR  20  00    80      ////Character Scan 10////
        17A2  ADC  69  C0            Add Carry; Reset VS

        17A4  CMP  C9  84            It is "Line 17"?
        17A6  BCC  90  dA*           No, continue character scans
        17A8  BCS  b0  00            Yes, Go to vertical blanking scans
-START→ 17AA  CLD  d8                Equalize 2 cycles

        17Ab  JSR  20  00    C0      ////Vertical Sync Scan////
        17AE  LDX  A2  22            Load #V Blank Scans -2
        17b0  LDA  A9  00            Initialize Lower Address
        17b2  STA  8d  (86)  (17)    Continued

        17b5  CLC  18                Equalize 2 cycles
        17b6  BCS  b0  00            Equalize 2 cycles again
        17b8  JSR  20  00    80      ////Vertical Blanking Scans////
        17bb  DEX  CA                One less scan

        17bC  BMI  30  C2*           Start Character Scan
        17bE  BPL  10  F5*           Repeat Vertical Blanking scans
```

NOTES: TVT6 must be connected and scan microprogram PROM (IC1) must be in circuit for program to run

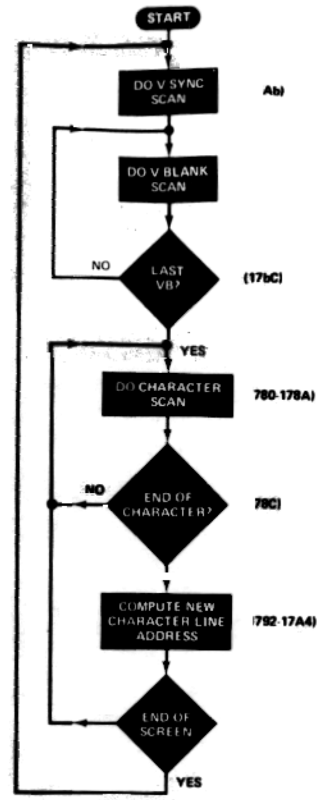
Step 1785 goes to where the upper address stored in 1787 and the lower address stored in 1786 tells it to. Values in these slots continuously change throughout the program.

Normal program horizontal frequency is 11,764.705 Hz. Vertical Frequency is 60,024 Hz. 85 us per line; 196 lines. Character time 1 us. 160 active lines, 36 retrace. Needs TV set adjustment and possible modification (hold and width).

* Denotes a relative branch that is program length sensitive.

( ) Denotes an absolute address that is program location sensitive.

TVT6 length jumper must be in "64" position.



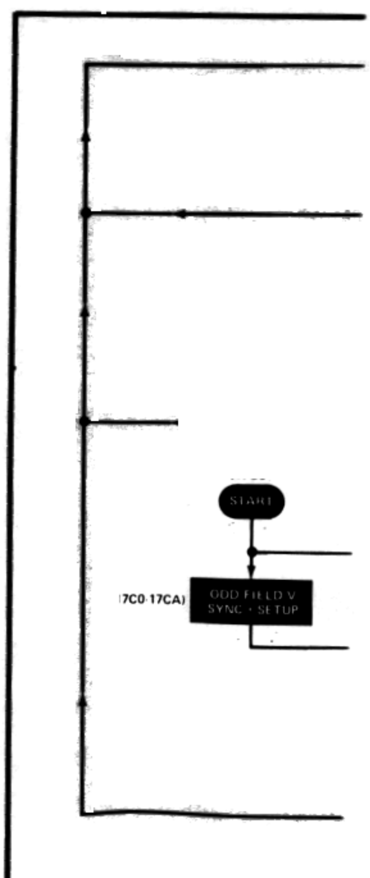
16 x 64 RASTER SCAN

proceed beyond this point until you are certain that the SCAN subroutine is operating properly. (Critical waveforms to be observed with an oscilloscope are illustrated in Fig. 7 using the program listed in Table II.)

Insert *IC3* into its socket and load the program given in Table II. (*Never* install an IC in a powered circuit; always turn off the power, install the IC, and power up again.) Set the address to 17Ad and depress GO. Using an oscilloscope, check at test point VR for the presence of a 60-Hz pulse. Switch the scope to line-sync and observe that the pulse remains fixed or drifts very slowly across the screen. Again, do not proceed until you are certain that the SCAN program is operating properly.

Install all remaining IC's, except *IC5*, in their respective sockets on the TVT-6 board. At this point, the screen should be filled with a stable display of 512 cursor boxes. Viewed up close, the boxes should appear to be "hiding" characters. Do not proceed until you have the indicated display.

Checking with Fig. 7, particularly with respect to the LOAD and CLOCK on *IC8* (Fig. 7A) verify whether or not the appropriate waveforms are present. If they are, remove the jumper wire from the *IC5* socket and install *IC5*. Now, the screen of the monitor should have displayed on it a full array of characters with about half of them winking cursor blocks. Load the following hex numbers into memory, starting at location 0200:



20, 20, 20, 50, 4F, 50, 55, 4C, 41, 52, 20, 20, 45, 4C, 45, 43, 54, 52, 4F, 4E, 49, 43, 53, 20, 20, 54, 56, 54, 2D, 36, 20, 20. Return to address 17Ad and depress GO. The top display line should now read "POPULAR ELECTRONICS TVT-6" and be indented three spaces. If all is well to this point, you can begin feeding in your cursor programs, add external keyboard and/or cassette loads and dumps, etc.

Should you encounter problems with your TVT-6, always begin debugging by using the 16 × 32 format on a KIM-1, even if you plan on using longer line lengths or plan to translate the code into another coding system. Note that the translation *must* be at the machine-language level because the SCAN program must provide the exact number of machine cycles as well as the proper sequencing. The 64-character lines will require some adjustments to be made in the monitor TV receiver's horizontal circuit as detailed earlier.

Closing Remarks. We have presented here full construction and operating details for a very versatile and inexpensive TV typewriter for use with the KIM-1 microcomputer. If you have a computer that uses a microprocessor other than the 6502 used in the KIM-1, we refer you to the box for use details. ◇

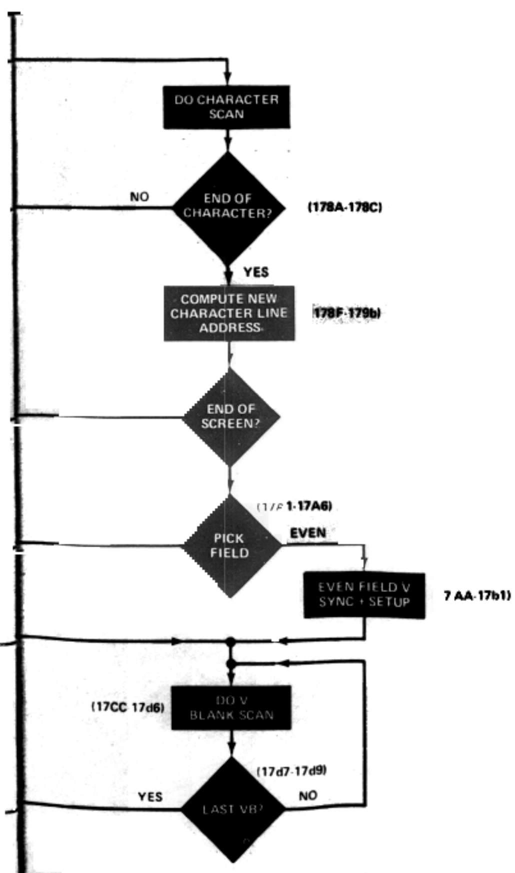


TABLE V

CRUNCHER THE BEAR Program for a 32 line X 64 character per line TVT6 raster scan.

µP - 6502 Start - JMP 17C0 Displayed 0000-07FF
System - KIM-1 End - Interrupt Program Space 1780-17dA

HS	VS	L4	L2	L1	V16	V8	V4	V2	V1	H32	H16	H8	H4	H2	H1

Upper Address — Lower Address

1780	LDA	A9	80			Initialize Upper Address
1782	STA	8d	(87)	(17)		Store Upper Address
1785	JSR	20	00	80		////Character Scans 0-7////
1788	ADC	69	10			Increment Character Gen by 2
178A	CMP	C9	C0			Is VS = 1?
178C	BCC	90	F4*			No, Do next character scan
178E	PHA	48				Save Upper Address
178F	LDA	Ad	(86)	(17)		Get Lower address
1792	ADC	69	3F			Increment L; Set Carry on V2 overflow
1794	STA	8d	(86)	(17)		Restore L; Save carry
1797	PLA	68				Get Upper Word
1798	JSR	20	0C	80		////Character Scans 8,9 ////
179b	ADC	69	C0			Add Carry; Reset Upper Address
179d	CMP	C9	88			Is it "Line 33"?
179F	BCC	90	E1*			No, repeat Character Scans
17A1	LDA	Ad	(81)	(17)		Get Interlace word
17A4	ADC	69	78			Set Carry if Odd Field finished
17A6	BCC	90	0C*			Start Even Field if Carry Clear
17A8	LDX	A2	22			Load Even number of V Scans -2
17AA	LDA	A9	80			Load Even Field Upper Start
17AC	STA	8d	(81)	(57)		Even Field V Sync + Restore Interlace
17AF	LDA	A9	88			Even Field Line 33 CMP Value
17b1	STA	8d	(9E)	(17)		Store Even 33 CMP Value
17b4	LDA	A9	00			Clear Accumulator
17b6	STA	8d	(86)	(17)		Initialize Lower Address
17b9	LDY	A0	06			Equalize 31 cycles
17bb	DEY	88				continued
17bC	BPL	10	Fd			continued
17bE	BCS	b0	0C*			Jump if even field
START → 17C0	LDA	A9	88			Load Odd Field Upper Start
17C2	STA	8d	(81)	(57)		Odd Field V Sync + Restore Interlace
17C5	LDA	A9	90			Odd Field line 33 CMP Value
17C7	STA	8d	(9E)	(17)		Store Odd 33 CMP Value
17CA	LDX	A2	23			Load Odd number of V Scans
17CC	JSR	20	3F	80		//// 1st V Blanking Scan ////
17Cf	PHA	48				Equalize 7
17d0	PLA	68				continued
17d1	CLD	d8				Equalize 4
17d2	CLC	18				continued
17d3	JSR	20	00	80		//// Other V Blanking Scans ////
17d6	DEX	CA				One Less scan
17d7	BMI	30	A7*			Start Character Scans
17d9	BPL	10	F6*			Repeat V Blanking Scan

NOTES: TVT6 must be connected and scan microprogram PROM IC1 must be in circuit for program to run. TVT6 length jumper must be in "64" position.

Step 1785 goes to where the upper address stored in 1787 and the lower address stored in 1789 tells it to. Values in these slots continuously change throughout the program.

Step 1781 is 80 for even fields and 88 for odd fields. Step 179E is 88 for even fields and 90 for odd fields.

Both 17AC and 17C2 require that page 17 be enabled when page 57 is addressed. This is done automatically with KIM-1 circuitry.

Note that 2K worth of contiguous memory from 0000 to 07FF is needed. This takes a KIM-1 modification. Both sets of 1k words must share a common upstream tap but be separately enabled.

Normal program horizontal frequency is 11,764.705 Hz. Vertical Frequency is 59.8712 Hz. For 60 Hz vertical use 1.002150 MHz crystal. 85 us per line; 196.5 interlaced lines per field; two fields per frame. One us character time, 160 active lines per field. Needs TV set adjustment and possible modification (hold and width).

* Denotes a relative branch that is program length sensitive.

( ) Denotes an absolute address that is program location sensitive.

Letters

USE THE OLD APPROXIMATIONS

I am sure you provided a service for many readers with the discussion in "Accurate Milliammeters On a Budget" (June 1977). As an old Ham, I wonder why you did not give the old approximations for shunt calculations and for determining the internal resistance of a meter movement. They yield results whose scalar accuracy is better than that of the meters themselves.—*D. Conover, WA6MVZ, La Mesa, CA.*

The ones presented are more accurate, though both provide results more accurate than meters themselves.

SHORTWAVE-LISTENING BOOSTER

Your articles on shortwave listening and reports on SW receivers are excellent. I am just getting started as an SWL'er, and POPULAR ELECTRONICS is helping me a great deal in my new hobby. Please keep Harry L. Helms's articles, the DX Listening column, and Shortwave Broadcasts Charts coming. —*Paul Semenza, Tarrytown, NY.*

TRANSPOSING BITS

In the "Pixie Graphics Display" article (July 1977), if the data pins on the 1861 IC are transposed, the bits will be displayed with the LSB first and the MSB last. This arrangement will be a little easier to use when calculating a display from software or an A/D converter. Just transpose D7 and DØ , D6 and D1, D5 and D2, and D4 and D3.—*Richard DeLombard, Huron, OH.*

TVT-6 DISPLAY UNCROWDING

We built a "TVT-6 Video Display" unit (July 1977) and interfaced it with a KIM microcomputer. While following your published debugging instructions, we noted that our video monitor was displaying letters that were not complete because they were crowded together. Signal tracing revealed that the LOAD signal was okay but the CLOCK signal presented only 3 cycles/μs instead of the specified 6 cycles/μs. I tried replacing *C5* with a smaller value of capacitance, with the result that the display was greatly improved. After some cut-and-try experimenting, we ended up with a 390-pF value and a perfect display. Anyone who runs into a similar problem with one of these video-display units might want to take note of our experience. —*David A. Byrd, Memphis, TN.*

ENLARGER REGULATOR PRECAUTION

Since your enlarger voltage-regulator project in the November 1977 issue is specifically aimed at the *color* darkroom worker, it would be well to point out that this regulator cannot be used with some enlarger color heads that have built-in filtration. Such heads usually have low-voltage, high-intensity lamps and transformer power supplies. Use of a dc supply, like that shown for the regulator in the November issue, can result in damage to the transformer. —*Bennett Evans, New York, NY.*

Out of Tune

In "How to Convert a 'Four Banger' for Stopwatch Functions" (August 1977), the *IC2* and *IC3* designations are shown transposed in Fig. 2. The Fig. 1 schematic diagram is correct.

In the Parts List in "Build a Digital Camera Shutter Timer" (August 1977), *DIS1* through *DIS5* are described as common-anode displays; they are actually common-cathode displays.