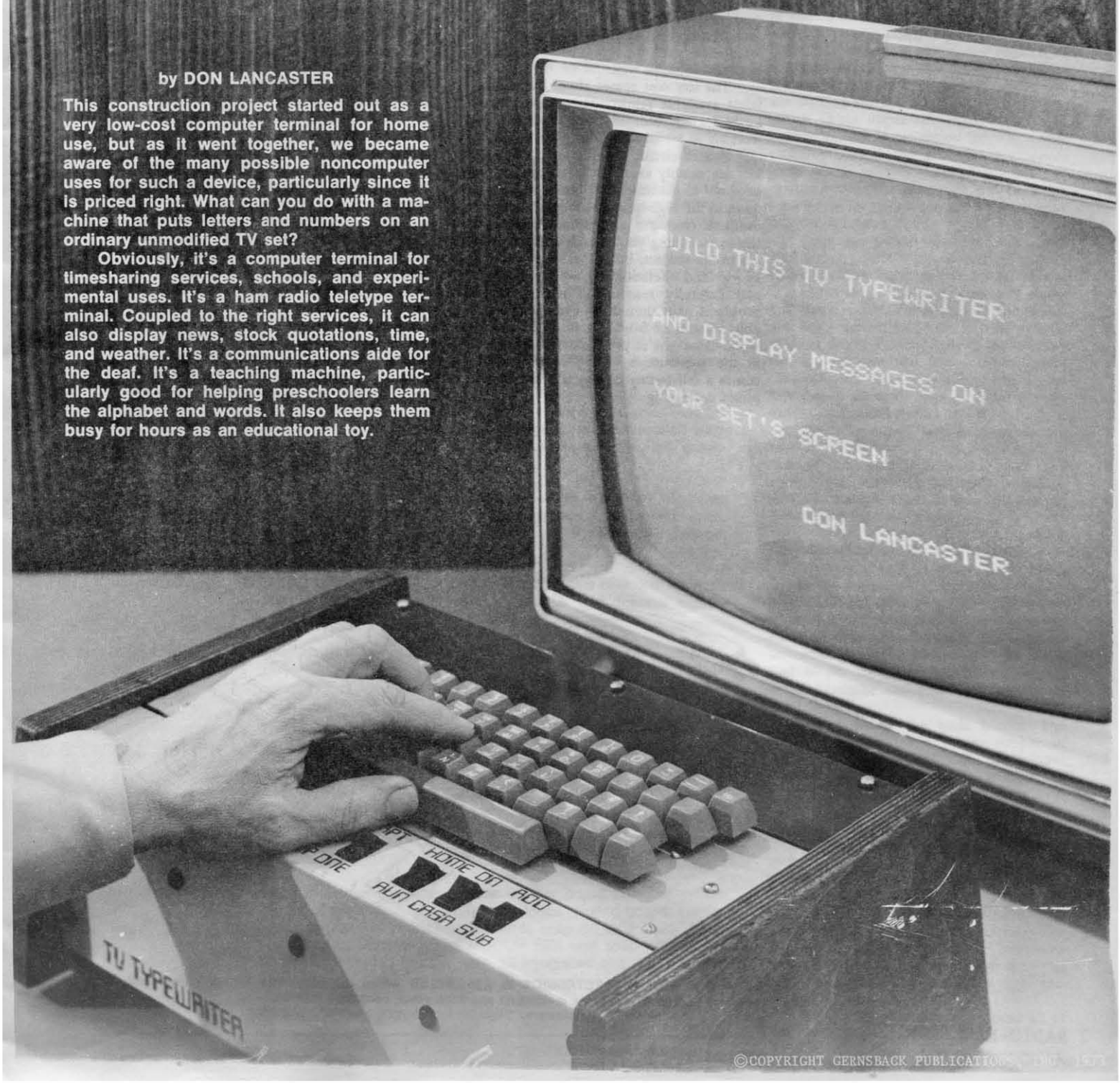
TV TYPEWRITER

This character generator and your TV set form a computer terminal, educational toy or display device. Basic details are here. Full and complete data is yours when you need it.

by DON LANCASTER

This construction project started out as a very low-cost computer terminal for home use, but as it went together, we became aware of the many possible noncomputer uses for such a device, particularly since it is priced right. What can you do with a machine that puts letters and numbers on an ordinary unmodified TV set?

Obviously, it's a computer terminal for timesharing services, schools, and experimental uses. It's a ham radio teletype terminal. Coupled to the right services, it can also display news, stock quotations, time, and weather. It's a communications aide for the deaf. It's a teaching machine, particularly good for helping preschoolers learn the alphabet and words. It also keeps them busy for hours as an educational toy.



It's a super sales promoter, either locally or on a store-wide basis. It's easily converted to a title machine for a video recorder. It's a message generator or "answer back" unit for advanced two-way cable TV systems. Tied to a cassette recorder, it's an electronic notebook and study aid, or a custom catalog. It's an annunciator for plant, schools, and hospitals that tells not only that someone is needed, but why and where.

And, if all that isn't enough, it's easy to convert into a 12- or 16-place electronic calculator. You can also make a clock out of it, and, with extensive modification, you can even make a 32-register, 16-place serial digital computer out of the beast!

Cost of the project? Around $120 for the basic unit. This is slightly under two month's normal *rental* of commercial units that don't do nearly as much, and less than 1/10 the cost of anything commercial you could buy to do the same job. And we feel that this cost is finally low enough that a lot of new uses are now not only possible, but reasonable as well.

The low cost comes about by using the latest available semiconductors, leaving the keyboard and case as flexible options, and working in kit form.

Printed-circuit boards and complete kits are readily available as are any special or hard-to-get-normally parts. A limited quantity of high-quality keyboards are also available from the same source. This is not the sort of thing you'd want to try as a first electronic project, but if you are willing to slowly and methodically work things out and carefully reason out any debugging problems, you shouldn't have an unreasonable amount of trouble getting the thing to work. Once you're past a certain stage early in the construction, the TV set itself becomes a self-testing display that greatly simplifies debugging.

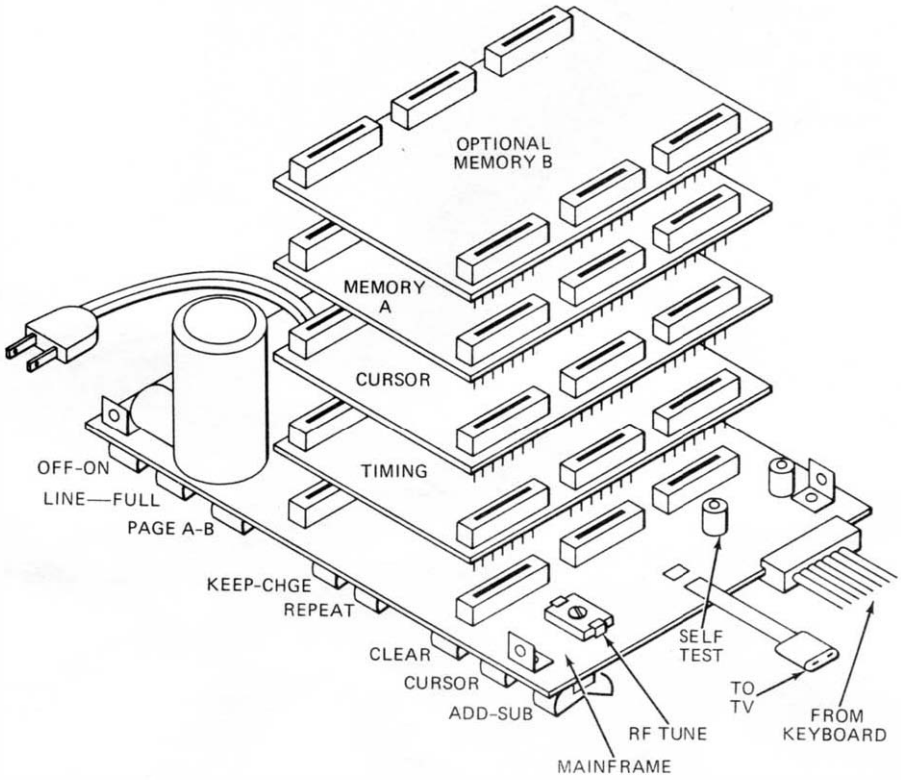
To make things easier, you can get a complete copy of the *entire* story that includes additional design information, how it works, PC patterns, construction details, etc. *DO NOT ATTEMPT CONSTRUCTION WITHOUT THIS ADDITIONAL INFORMATION!*

Construction is done in stages. Once each stage is tested, it is safe to go into the next, progressively working up to a complete unit. The basic machine we'll show you works from a keyboard or a set of six switches and a pushbutton. Thanks to the plug-in construction, low-cost add-on circuit boards can let you talk to a computer or a cassette recorder, or adapt the unit for 12-place calculation. These add-ons will be picked up later if enough readers seem interested. They're not needed for most of the possible applications of this TV Typewriter.

Specs of the unit

Complete specs appear in Table 1. The basic device generates and stores 512 characters, arranged as sixteen lines of 32 characters each. A second page of characters is easily added internally to bring the total up to 1024 characters. For more storage, a C90 cassette can store well over a hundred pages, so the total capability is quite large. The characters available are standard ASCII ones that include the capital letters, numerals, and most punctuation.

The TV Typewriter is self-powered and contains its own miniature TV transmitter which simply clips onto the antenna terminals of an unmodified TV tuned to an unused channel. Several TV's may be driven simultaneously, and a direct video output is also available for industrial and experimental uses. While any TV can be "borrowed" and used with the typewriter, small, high-quality portables give the nicest presentation, and slight size and position



HOW TYPEWRITER ELECTRONICS IS ASSEMBLED within the case. The timing, cursor and memory boards plug into the mainframe and each other, cordwood fashion.

A SURPLUS KEYBOARD is used in this version of the TV typewriter. Unit on cover has a homemade keyboard based on the article in the February issue.

TABLE I

COMPLETE SPECIFICATIONS—TV TYPEWRITER

STORAGE:	1024 Characters arranged as 2 pages of 16 lines of 32 characters each.
OUTPUTS:	Rf Output tuneable from channel 2 through 5; clips directly to the antenna terminals of one or more unmodified television sets. Optional positive-white video output.
INPUTS:	Parallel, TTL compatible, ASCII character code (Table II) is input with positive logic on six lines; a seventh *keypressed* line is suddenly brought to ground to input character, Internal debouncing. The full 8-bit ASCII code may also be used as an input. If done, *any* CTRL input will be interpreted as a combined CARRIAGE RETURN and LINE FEED, CTRL output available for code extension.
FORMAT:	Begins in upper left HOME position and proceeds as in normal typing. Carriage return and linefeed automatic at end of line. At bottom of screen, jumps to upper left HOME position and rewrites over old text.
EDITING:	Winking cursor indicates next character position. Cursor may be blanked and may be independently moved in any direction with or without changing text. One or more letters may be easily changed at any time.
TIMEBASE:	Internal, crystal controlled TTL divider. Basic video clocking rate = 4.562 MHz. 15,840 kHz noninterlaced horizontal scan rate; 60-Hz vertical scan rate. Easily converted to full interlace for Video Recorder titling applications.
MEMORY:	512 word by 6 bit MOS dynamic storage, bus oriented for easy page conversion, optional memory output, and optional extension to calculator, computer, and other functions.
CONTROLS:	Internal: Rf frequency (trimmer capacitor) Position—(Jumpers—4 horiz; 3 vert for 12 possible locations.) EXTERNAL: ON-OFF PAGE OR LINE SCAN KEEP-CHANGE memory protect A or B page select REPEAT or SINGLE character HOME or RUN cursor location CURSOR ON-OFF ADD-SUBTRACT cursor direction
CONSTRUCTION:	Modular mother-daughter boards. Mother board contains power supply, rf modulator, and control switches. Timing board, cursor board and one or two memory boards snap on as a stack. Add-ons such as calculator and MODEM FSK unit snap onto same stack; not included in basic unit. 33 integrated circuits, of which 8 are MOS LS1.
SIZE:	7"x8½"x3", not including keyboard or case.

adjustments can further help the appearance, although they are not needed.

The characters are added one at a time and normally go on the screen just like you were typing. This is done by providing the proper ASCII character code on seven input lines and tripping an eighth "key pressed" line to enter the character. A winking cursor tells you where the next character is to go, but you can turn this off if you want to. Should the screen get filled, the machine starts over again on the top, rewriting over the old message.

Besides the normal operation you have a complete editing capability. You can move the cursor either direction anywhere you want and then change only the characters or words you wish to, thus editing something you already have on the screen. This nicely handles mistakes without having to start over again. A REPEAT key is available for putting down a group of identical characters or getting to a given position in a hurry. There's a KEEP-CHANGE switch to protect what you have written while you are moving around, and you can home the cursor to the upper left either by itself or erasing the whole picture on the way. Other switches control the direction the cursor goes, which page is being displayed, and optionally whether the mode will be a full screen one for typewriter use or a line scan one for calculator use.

Computer people would call this a parallel input system with off-line editing. A single machine command is available; this is the LINE FEED. Thus, any CTRL key moves you down a line. Other remote commands are easily added, but were left off to hold the cost down. The contents of the memory can be retransmitted with simple circuit modifications, and the whole system is bus-oriented to allow all sorts of add-ons without major circuit rework.

Character input rate is asynchronous and up to 30 characters per second, thus making the beast three times faster and compatible with the industry standard ASR-33 teletype. Hard copy is via cassette recorder or Poloroid[r] photos.

Organization of the Instrument

To keep things as simple as possible, the circuit is arranged like a set of snap-together blocks. This way, the only interconnect wiring consists of the line cord and the 300-ohm twinlead output. Since the interconnect wiring is locked into the board and 60-pin connector system, the biggest single headache and potential error source is eliminated.

Fig. 1 shows the basic blocks. The MAINFRAME contains a power supply of +12, +5, -5, and -12 volts; the control switches; the rf modulator; the internal test system; and connectors for both the keyboard and the other boards in the stack.

There are three other essential boards. The MEMORY board is the most important and the most complex. It contains a dynamic MOS (Metal Oxide Semiconductor) shift register that stores 512 words of 6 bits each. It also holds a single-line memory; a character generator; and an output video register. We'll see later that the single-line memory is needed to get each character back eight times in sequence for eight successive TV scans.

For a page-A memory, you need all of

this board. The additional page-B memory does not need a new single-line memory, character generator, and output video register, as it can borrow the one in the page-A memory when the second page is in use. This is called *bus organization.* The character generator will respond to anything that is enabled on the bus, be it page-A memory, page-B memory, a calculator add-on, or whatever. Of course, we have to be careful to only enable *one* possible source of characters at a time, but this is easy. We can also use the bus optionally to *output* characters to the outside world.

The output of the memory board also contains a video combiner that assembles the character video, sync signals, and the internal test signal into one composite video output. This output may either be used directly or routed to the rf modulator for clip-on operation of an unmodified TV. It can be optionally flashed or blinked.

The TIMING board contains a crystal divider and TTL (Transistor Transistor Logic) countdown chain that generates all the needed signals to run the typewriter in proper sequence. It does not normally use interlace, but the timing chain is split so that the somewhat more complex TV full-interlace system can be added if you need this sort of thing for video titling. There are two principal areas to the timing board, the MAIN timing, and the DERIVED timing. The main timing is the continuous waveforms obtained off the crystal divider, while the derived outputs combine portions of the main timing signals into properly coded signals needed to run the rest of the typewriter. Two examples are the composite sync signal and a blinker used for flashing, cursor winking, and repeat functions.

The third essential board is a CURSOR board. Anyone who ever tried to design and debug a simple one of these will easily understand why it is called a cursor board. Anyway, the cursor keeps track of where the next character is to go; runs the winking line that shows the character position; controls entry of the character; and optionally sets up characters for output. It also contains an input conditioner, and debouncer and a detector for CTRL commands that tells the typewriter to carriage return rather than enter a character.

Many cursor systems are extremely complex. This one is relatively simple in that it uses a *phase-shift counting technique.* The cursor has a *continuously running* counter just like the main timing chain does. Its output drops suddenly in some *relative* position, indicating where the next character is to go.

To back the cursor up, we throw in another count pulse. To run it forward, we hold back one normal count pulse. Thus, the *relative* position or phase of the cursor counter advances or backs up with respect to the system timing. Actually, to go forward, we hold back *two* normal system timing pulses and throw in a new one. This buys us a simplification of circuitry, but still ends up with the same result.

An ADD-SUBTRACT switch on the mainframe controls the cursor direction for editing. LINEFEED is handled by adding and removing the proper number of counts in the proper position in the cursor counter so that the new position is reached. Just like most typewriters, the linefeed always returns to the lefthand side.

In normal operation, each character entry moves the cursor over one character. When it gets to the end of the line, it starts again on the next line. When it gets to the bottom of the page, it starts again at the top. A CLEAR or HOME override also moves the cursor to the upper lefthand position.

And, this is about all you need for a normal parallel entry type of TV typewriter. One possible optional board is a MODEM or frequency-shift-keying interface. This would use a MOS chip and some TTL to convert to or from a serial tone input, suitable for computer or telephone line communication. A cassette recorder will work just as well with the modem for electronic notebook use.

Another possible add-on makes the typewriter into a calculator. This is done by converting the scan from a complete frame to a single line of numerals and would use a surplus calculator chip to provide the familiar calculator functions. If you already have or need the TV typewriter for something else, this add-on is far cheaper than a conventional calculator would be, and its display would be obviously larger and more readible.

Or, you can add most anything else you want onto the machine by tieing into the bus-oriented lines (b1 through b6). For instance you can think of the memory as sixteen registers of 32 numbers each, and those numbers are decimal numbers plus, not bits! With six bits per word, you can store 10 possible numerals and 54 machine commands in any word! Or, you can split the registers into 32 registers of 16 decimal numbers each, building your own computer or programmable calculator.

Of course, this computer-add-on is very much an advanced experimenter project, but it really doesn't take much more than a double handful of TTL to pull it off. While such a computer will be relatively slow (around a 33-ms cycle time), it does provide an extremely accurate and very low cost computer approach, particularly when you are working directly with BCD numbers instead of binary.

Some basics

Before we turn to the actual circuitry, some basics of what a character is and how it can get on a TV screen is in order. Lets start with the characters:

If we had six possible binary bits of either 1 or 0, we would have sixty-four different possible combinations ranging from 000000, 000001, 000010, . . . through to 111110 and finally 111111. These 64 different states can represent all the capital letters, all the numbers, a blank, and most punctuation, following the standard ASCII code. In the TV Typewriter, all of the six bits of this code must be presented at once or in *parallel* form, and this is the *only* code the circuitry shown can use. Other codes can be converted to ASCII before going into the TV Typewriter. A seventh bit is used to separate characters from internal commands.

Our final presented character consists of an array of 5 × 7 dots. Since it only takes 6 bits to store a coded character and at least 35 bits to store a generated one, its obviously much better and cheaper to generate the characters *after* they are stored, rather than before.

For other keyboards and encoders, the

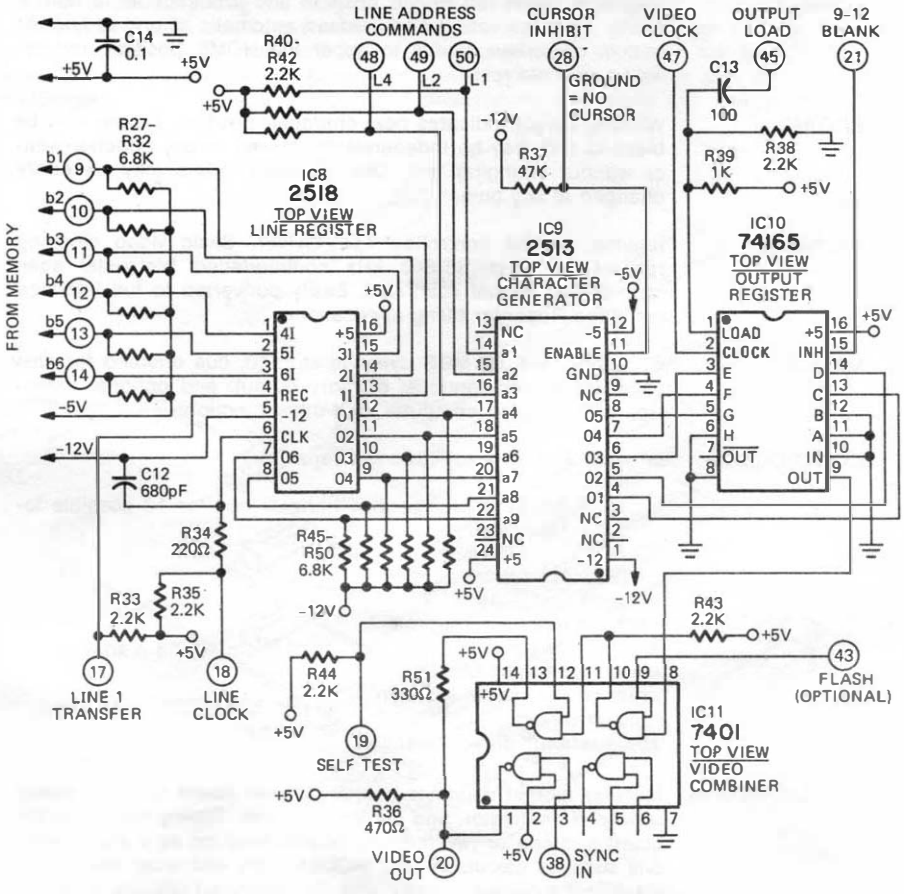
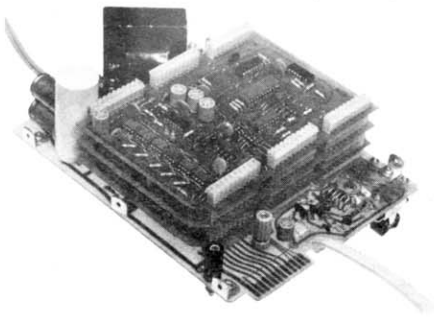


FIG. 2—SCHEMATIC OF THE CHARACTER GENERATOR. the use of integrated circuits greatly reduces circuit complexity and makes possible a very compact device.



INSIDE VIEW SHOWS STACKED BOARDS containing the electronic circuits in the TV typewriter.

typewriter gives you lots of +5 and a choice of +12 or -12 volts at relatively low current. The original **Radio-Electronics** ASCII encoder needs the +12; a mechanically encoded keyboard will only need +5, while a MOS encoded keyboard probably will need +5 and -12 volts.

A limited quantity of suitable keyboards is presently available from the kit source. Other sources of input material include computers, calculators, the phone line, or a cassette recorder. Many of these signals will be in serial or one-bit-at-a-time form and have to be changed to the parallel ASCII code. This takes an add-on board using a MOS terminal-receiver chip. Regardless of the source, your input must be in parallel form when presented to the typewriter, with a "1" near but not exceeding the internal +5 and a "0" near but not going below ground. Internal debouncing is provided on the cursor board for manual keyboard entry.

Interfacing with the TV

To use an unmodified TV, we have to build a miniature transmitter and arrange the signals so they compare as closely as possible to a normal broadcast set of scanning standards. This way, any TV can be driven by the typewriter simply by clipping onto the antenna terminals and tuning to an unused low channel.

A TV starts in the upper lefthand corner and sweeps a dot rapidly to the right and slowly downward, taking around 62 μs to get across the screen and 33 ms to get to the bottom. It then repeats the process again and again, presenting a series of dots that assemble into a series of still pictures that the tube phosphor and your eye integrate to get the effect of a complete and usually moving picture.

Brightness is changed on each dot by controlling the picture tube's cathode current, which in turn follows the input rf signal seen at the antenna terminals. Very low signal is seen as white. The stronger the signal, the blacker the picture. The sync signals are the strongest of all, or "blacker than black", and are used to synchronize the scanning of the television set to the transmitted signal.

To provide sync, we need one horizontal sync pulse at the beginning of each scan line, and one vertical sync pulse for the beginning of each frame. To keep things simple, we make all the frames identical in the TV Typewriter instead of using interlace. Interlace has no advantage on a stationary message presentation and simply adds parts. If you have to you can easily add it for video titling or other places where you must superimpose the TV Typewriter's output onto an existing program.

We assign 48 possible character positions across the TV screen, but we only use 32 of them. The remaining 16 allow for retrace and the extreme overscan used on economy TV's. We assign 12 scan lines per row of characters. The uppermost scan line is blank except possibly for a winking cursor that appears as a bar above the next character to get input. The next 7 lines form the character as a series of dots 5 dots wide by 7 dots high, and the final 4 lines are blank. These allow for the space between character lines.

We likewise assign 22 possible character lines but only use 16 of them, this time saving 7 for vertical retrace and overscan. By picking the right timing frequencies, we obtain a horizontal rate thats so close to the normal rate the TV doesn't know the difference, and a vertical rate of exactly 60 Hz. The latter is especially important to keep hum bars out of the display. Since each frame is stationary and ends with a bunch of blanks, Equalizing pulses are neither needed nor used.

Each character takes six bits of storage, arranged as one parallel 6-bit word. The storage is used to hold the character from time of entry until it is no longer needed, which can range from seconds to days. For a single-page memory of 512 characters, we use six 512-bit recirculating MOS shift registers. These go around once each TV frame. The timing and cursor boards together decide *where* in each of the six registers a new character is to go. Once in memory, the character stays in the same relative slot until it is cleared or replaced. The memory is violatile, meaning that you lose the message if power drops for more than a half a second or so.

Note that an ASCII blank is 100000. All 1's is a "?" and all 0's is an "@". This is helpful when troubleshooting, for its rather difficult to get a totally blank screen by accident. On the other hand, this means we have to be careful when we clear our memory to erase the screen. Here, we purposely have to set up the 100000 code. One way to do this, is to hold down the keyboard's space bar during the clearing process. A better way is to remove the keyboard encoder power (via the UNCLEAR or not-clear line from HOME switch S6) from the encoder to get all 0's out. Then a "1" can be force-fed to the a6 input line to set up the proper code. This gives us a one-button clearing operation. Other schemes can easily be worked out, but the essential thing is that the 100000 code gets set up during the time you want to erase what you have.

We normally put a character into memory and leave it there for a relatively long time. The memory usually is in a *recirculate* mode where its own output is connected internally back to its input. The memory is a *serial* device—the bits take turns coming out one at a time, and 512 pairs of clock pulses are needed to turn the memory over exactly once.

The memory is normally in the recirculate mode. We keep it there if we are using the other page or if we have our KEEP-CHANGE switch in the KEEP position. We also force it to recirculate if a CARRIAGE RETURN command (a6 *and* a7 = 0) arrives, for, just like a typewriter, we don't want to enter anything while the carriage goes back to the beginning of a line.

The only time we actually enter a character is once when the cursor board tells us it is the right time, then only if we are working on this page, want to change the character, and are not trying to return the carriage.

About keyboards

So where do we get our characters? The simplest and far cheapest source is from six switches and a pushbutton, arranged to get us +5 for a "1" and ground for a "0" and set up so the pushbutton gives you a sudden +5 to ground "Key-pressed" output. This is handy (almost essential) for testing, and can be used for message generation, although it takes quite a bit of practice to get any speed.

A keyboard is the next best bet, such as the low-cost keyboard in the February 1973 issue of **Radio-Electronics,** and its low-cost companion ASCII encoder (April 1973). Many of the surplus keyboards offered in the back pages of **Radio-Electronics** can also be either used directly or readily adapted.

The input code *must* be in ASCII. Any keyboard that consists of one or two make contacts per key *must* be converted in a suitable encoder, again such as the low-cost ASCII encoder described in the March 1973 issue of **Radio-Electronics.** Further, the ASCII output *must never exceed* the internal +5 volt supply of the typewriter, nor should it go below ground, even by a small amount.

Jumpers on the timing band allow for selection of 12 possible display positions so that internal TV adjustments need not be changed.

Character generation

We use raster-scan dot-matrix characters providing an array of 5 dots wide by 7 dots high for each character with one "undot" between characters for spacing. Seven passes of the TV raster are needed to generate each line of characters. This says we must borrow one line's worth of characters (32) from the memory and put it into a new *line register* memory, use it over again at least seven times, and then later on, go get a new line of characters. To do this, we need a line memory, a single IC consisting of six 32-bit recirculating shift registers.

Let's go through a typical scan and see what happens. Most of the circuitry is shown in Fig. 2.

Suppose we just retraced to the upper left hand corner. We're now on line 1 of the top of the characters. On line No. 1, our line register is connected to the memory and it samples the next 32 characters to be presented. The main memory thus fills the line register. For the next twelve scans, the memory is idle, but the line register brings the same characters back over and over again. On scans No. 2 through No. 8, the character is actually generated. The line register drives a character generator.

The character generator is also connected to logic that tells it which part of each character it is working on. The output of the character generator goes to an output register that converts the characters into actual video.

Since line No. 1 is supposed to be all blanks, the character generator is told this and we get all blanks, except possibly for a brief cursor winking bar.

On the second scan line, we again clock the line register 32 times, letting it go once around. The main memory just waits. This time, the character generator is told to work on line No. 2 and please put down the *top row* of dots on each character. For instance, if a "T" comes up, we get five "1's" in a row. An "S" would be 01110 and so on. As the TV scans across, each top row of dots for each following character is put down.

On the next pass, we again clock the line register 32 times. This time, the *second* row of dots gets output, with a "T" being a 00100 and an "S" being 10001, and so on. Lines No. 4, 5, 6, 7, and 8 are handled the same way, with the character generator working on the line it is told to and the line register going once around for each line. By the end of the eighth line we have put down all the dots we need for a line of 32 complete dot-matrix characters. The circuitry is blanked for the next four scan lines, providing us with a space between character lines.

On line No. 13 (a new line "1"), our main character memory is once again clocked 32 times and the line register is simultaneously clocked. This fills up the line register with a new set of 32 characters. The same operation repeats for each of the sixteen rows of characters that we want to put down.

Notice that the timing runs in bursts and is not continuous. Thus, the line register runs for 32 counts and waits 16 for retrace and so on. The memory does the same thing, but only on every twelfth line during the active scan. Carefully established internal timing delays take care of settling times between memory, line register, character generator, and the final video generating output register. The output register converts the five parallel outputs of the character generator into serial, high speed video.

About the memory bus

So far, we've assumed that we were using the page-A memory with the page-A character generator. Thanks to the memory bus (b1 through b6) we can connect anything we like to the character generator, including the page-A memory, the page-B memory, or anything else we want to hang on these lines.

To run page-B, we simply disable the page-A memory and enable the page-B memory's output. The handy thing about bus organization is that no complex switching is involved. Whatever is enabled gets connected to the character generator; other things tacked on just sit there. The only restriction is that *we have to enable only one character source at a time.* We can also use the same memory bus optionally to output characters to a computer, a cassette recorder, or a phone line.

This way, with suitable add-ons we have a choice. We can send one character at a time directly from the keyboard, or we can send an entire page at a time from the memory. The latter is faster and more complex but has the advantages that you can fix all the mistakes first and don't tie up nearly as much outside equipment. **R-E**

PARTS LIST

MAINFRAME

C1—5000-μF 10-V electrolytic
C2, C3—1000-μF 25-V electrolytic
C4—100-μF 6-V electrolytic
C5, C7—470-pF disc for vhf bypassing
C6—3–30 pF trimmer
C8—27-pF mica
C9—47-pF mica
C10—Gimmick attenuator made of twin lead
D1 to D6—1N491 or equal 1N4001
D7—12-V, 1-W Zener, 1N4742 or equal
D8—6.8-V, 1-W Zener, 1N4736 or equal
D9—5.1-V, 1-W Zener, 1N4734 or equal
D10 to D14—1N914 or equal
F1—1-A fuse and fuseholder
IC1—7805 regulator (Fairchild or Motorola)
J1, J2—Binding posts, one yellow, one black (5 way)
L1—Coil made from 4" of No. 14 solid wire
Q1—2N918 transistor in metal can, **do not substitute!**
R1, R2—47 ohms, ½-W
R3—22 ohms, ¼-W
R4, R9, R10—2200 ohms, ¼-W
R7—4700 ohms, ¼-W
R8—470 ohms, ¼-W
S1, S2, S3, S4, S7, S8—dpdt rocker switch
S5 to S6—dpdt rocker switch, momentary spring return
SO1 to SO6—connector, Molex 09-52-3103
SO7—TV lead-in connector
T1—Power transformer, dual 12-V center tapped secondaries, 1.5-A. Signal 24-1A or equal
MISC:—PC Board, 8¾ x 6¾; mounting brackets and hardware (6); switch mounting hardware (8 sets); line cord and cable clamp; hardware for T1; vertical heat sink for IC1; No. 24 jumper wire; sleeving; No. 14 wire for L1; fuse clips and hardware, 300-ohm twinlead, 18"; PC terminals, optional-2; solder.

MEMORY BOARD

Note: Each system needs one "Memory A" board. Memory "B" boards are optional. These parts are needed for *either* a Page A or a Page B memory:
C1, C3, C5, C7—100-μF 15-V electrolytic
C2, C4, C6, C8, C9, C10, C11—0.1-μF disc ceramic
D1, D4, D5, D6—1N914 or equal
D2, D3, D7—1N4001 or equal
IC1 to IC 6—2524 MOS 512-bit recirculating shift register (Signetics)
IC7—7406 hex driver, TTL
P1 to P60—Connector pins to fit Molex 09-52-3103 connectors
Q1, Q2—2N5139
All resistors ¼-W carbon
R1 to R6, R25—2200 ohms
R7, R8, R15—2.7 ohms
R9, R23—10,000 ohms
R10, R12—22 ohms
R11, R13—4700 ohms
R14, R18, R19, R20—150 ohms
R16, R17—100 ohms
R21, R22—1000 ohms
R24—330 ohms
R26—470 ohms
These parts are needed ONLY for a page A memory:
C12—680-pF mica
C13—100-pF mica
C14—0.1-μF disc ceramic
IC8—2518 MOS hex 32-bit recirculating register (Signetics)
IC9—2513 character generator, MOS (Signetics)
IC10—74165 TTL 8-bit PISO register
IC11—7401 TTL open collector NAND gate
R34—220 ohms
R33, R35, R38, R40 to R42, R43, R44—2200 ohms
R27 to R32, R45 to R50—6800 ohms
R36—470 ohms
R37—47,000 ohms
R51—330 ohms
MISC: PC Board, 4½" x 6½"; #24 wire jumpers; Sleeving; PC Terminals (Optional-2), Solder.

TIMING BOARD

C1 to C4—01-μF 10-V disc ceramic
C5, C6—160-pF mica
C7—0.001-μF disc ceramic
C8—100-μF 6-V electrolytic
C9—33-μF 6-V electrolytic
IC1—MC4024 dual astable (Motorola)
IC2, IC3,IC5—8288 divide by 12 (Signetics)
IC4—7473 dual JK, TTL
IC6–8288
IC7, IC8—7432 quad OR gate, TTL
IC9, IC12—7402 quad NOR gate, TTL
IC10, IC11–7410
P1 to P60—Pins to fit Molex 09-52-3103 connector
R1—330 ohms ¼-W
R2, R3—220 ohms, ¼-W
R4—2200 ohms, ¼-W
SO1 to SO6—Molex 09-52-3103 socket
XTAL 1—4561, 920-kHz series-resonant crystal
MISC: PC Board, 4½" x 6½"; #24 solid wire jumpers; Sleeving; PC Terminals (optional-21); solder.

CURSOR

C1—1200-pF mica
C2—4300-pF mica
C3—620-pF mica
C4—6200-pF mica
C5—1000-pF mica
C6 to C9, C12 to C15—0.1-μF disc ceramic
C10—100-μF 6-V electrolytic
C11, 16—10-μF 6-V electrolytic
IC1—7408 quad AND gate, TTL
IC2, IC4—74197 or 74177 or 8281 or 8291 divide by 16 TTL
IC3—7473 dual JK TTL
IC5, IC6—7402 quad NOR gate TTL
IC7—7474 dual D flip-flop, TTL
IC8—7400 quad NAND gate, TTL
IC9—555 timer, Signetics
P1 to P60—pins to fit Molex 09-52-3103 connector
Q1—2N5129
R1, R5, R8, R13, R16, R21—1000 ohms, ¼-W
R2, R3, R4, R6, R7, R9, R11, R14, R17, R18, R22—2200 ohms, ¼-W
R10, R19—330 ohms, ¼-W
R12—100 ohms, ¼-W
R15—100,000 ohms, ¼-W
R20—150 ohms, ¼-W
SO1 to SO6—Molex 09-52-3103 connector
MISC: PC Board, 4½" x 6½"; No 24 wire jumpers; sleeving; PC terminals (optional-8); solder.

The following items are avilable from Southwest Technical Products, 219 West Rhapsody, San Antonio, Texas, 78216.

All circuit boards are etched and drilled

Mainframe board:	**No. TVT-1, $9.75**
Timing board:	**No. TVT-2, $5.75**
Cursor board:	**No. TVT-3, $5.75**
Page A or B board	**No. TVT-4, $5.75**
High-quality keyboard, custom remanufactured for TV typewriter use (less-encoder)	**No. TVT-5, $18.75**

A complete or nearly complete kit of parts will also be offered, but pricing depends on semiconductor availability at time of publication. Write for a complete list of available parts and prices for assembled units.

TABLE II

ASCII CHARACTER CODE USED IN T.V. TYPEWRITER

Char	A6	A5	A4	A3	A2	A1	Char	A6	A5	A4	A3	A2	A1
@	0	0	0	0	0	0	blank	1	0	0	0	0	0
A	0	0	0	0	0	1	!	1	0	0	0	0	1
B	0	0	0	0	1	0	"	1	0	0	0	1	0
C	0	0	0	0	1	1	#	1	0	0	0	1	1
D	0	0	0	1	0	0	$	1	0	0	1	0	0
E	0	0	0	1	0	1	%	1	0	0	1	0	1
F	0	0	0	1	1	0	&	1	0	0	1	1	0
G	0	0	0	1	1	1	'	1	0	0	1	1	1
H	0	0	1	0	0	0	(	1	0	1	0	0	0
I	0	0	1	0	0	1	)	1	0	1	0	0	1
J	0	0	1	0	1	0	*	1	0	1	0	1	0
K	0	0	1	0	1	1	+	1	0	1	0	1	1
L	0	0	1	1	0	0	comma	1	0	1	1	0	0
M	0	0	1	1	0	1	-	1	0	1	1	0	1
N	0	0	1	1	1	0	.	1	0	1	1	1	0
O	0	0	1	1	1	1	/	1	0	1	1	1	1
P	0	1	0	0	0	0	0	1	1	0	0	0	0
Q	0	1	0	0	0	1	1	1	1	0	0	0	1
R	0	1	0	0	1	0	2	1	1	0	0	1	0
S	0	1	0	0	1	1	3	1	1	0	0	1	1
T	0	1	0	1	0	0	4	1	1	0	1	0	0
U	0	1	0	1	0	1	5	1	1	0	1	0	1
V	0	1	0	1	1	0	6	1	1	0	1	1	0
W	0	1	0	1	1	1	7	1	1	0	1	1	1
X	0	1	1	0	0	0	8	1	1	1	0	0	0
Y	0	1	1	0	0	1	9	1	1	1	0	0	1
Z	0	1	1	0	1	0	:	1	1	1	0	1	0
[	0	1	1	0	1	1	;	1	1	1	0	1	1
/	0	1	1	1	0	0	<	1	1	1	1	0	0
]	0	1	1	1	0	1	=	1	1	1	1	0	1
^	0	1	1	1	1	0	>	1	1	1	1	1	0
und.	0	1	1	1	1	1	?	1	1	1	1	1	1

To input a character, the proper code above is set up and the Keypressed input is suddenly brought to ground. A "0" is any voltage from 0 to +0.8 volts. A "1" is any voltage from +3 to +5 volts.

If you are using a computer, a modem, or a commercial keyboard with the full **eight** bit code, the eighth or parity bit is ignored. If bits 6 and 7 together are NOT "0", the characters above get input. IF bits 6 and 7 together are "0", regardless of the code, you get a simultaneous line feed and carriage return, and a "CTRL" output.

TABLE III-A

WHAT DO THE CONNECTOR PINS DO?

1. **Ground** – all boards
2-8. **Input ASCII code from keyboard.** Sent to memory A and memory B as needed. A6 and A7 also go to cursor for CTRL detection. True TTL logic.
9-14. **Memory buss.** Memory A, Memory B, or outside world act as sources. Memory A character generator or outside world act as loads. P channel, silicon gate MOS compatible. Only one source should be enabled at a time.
15,16 **Memory clocks.** Normally +5V. Each goes low briefly 512 times per frame to clock main memory. 32 times on line No. 1; 32 times on line No. 13; 32 times on line No. 25, etc. Negative TTL logic.
17. **Goes low on lines 1, 13, 25.** normally high. Used to change line register from recirculate to update each time a new line of characters is needed.
18. **Line Register clock.** 32 clock pulses delivered every line. normally high. TTL negative logic.
19. **Self test.** Superimposes "1" = white on video picture when connected to any TTL point in system.
20. **Video output.** Composite video for outside world of rf modulator. Sync = ground. White = maximum positive.
21. **9-12 Blank.** Goes high on lines 9-12, 21-24, 33-36, etc. to prevent register from clocking video when blanks are wanted. TTL positive logic.
22. **Keypressed.** Goes to ground when key is pressed. Signal is conditioned by cursor and then used for update.
23. **Blinker.** Source of 4-Hz signals used for cursor winking, repeat, or outside world special applications. TTL.
24,25 **Clear and UNclear.** 24 goes to +5V on clear, is otherwise grounded. 25 goes to ground on clear, is otherwise +5V.
26,29. **Up-down direction control.** If 26 is grounded, cursor moves up a line on linefeed. If 26 is connected to P clock on 29, cursor moves down a line on linefeed.
27,30. **Right left controls.** If these pins are open, the character cursor backs up a space. If they are shorted, they go forward a space on command. NOT directly TTL compatible.
28. **Cursor ON.** Ground turns cursor OFF. Blinking update gives brief positive signal to turn cursor ON above indicated character. Not directly TTL compatible.
31. **Protect A.** Prevents character entry if grounded.
32. **Calculator control.** Optional signal. Goes to ground when line scan rather than full frame scan is selected.
33. **Protect B.** Prevents character entry if grounded.
34. **Enable A.** Enables Memory A and connects it to output bus if +5; disables it if grounded.
35. **Enable B.** Prevents character entry if grounded.
36,37 **Line/Frame scan select.** Connects to clock on 37 for special scans, is open otherwise. P clock connected to pin 37 may be optionally combined with logic for the display of one character through 12 whole lines of characters. Characters are continuously input or output in line mode. Normal operation is in frame scan, and this clock is not used.
38. **Sync.** Composite V and H sync. Normally positive TTL negative logic. Position controlled by jumpers on timing board.
39-42, 44, 46 **Spare pins for add-ons.**
43. **Flash Display.** Optional. Ground puts out display for emphasis of one word, one line, or full frame. Useful with external timing to flash information, particularly negative numbers in a calculation.
45. **Output load command.** Loads characters into output register a suitable time delay after each character slot. TTL. Falling edge provides load command.
47. **Video Clock.** "A" clock of 4.561 MHz. Marches characters out of output register as serial video if not inhibited by pin 21 9-12 blank.
48-50 **"What character line is it?"** commands used to tell character generator which row of dots to work on. TTL positive logic. 000 = line 1, 001 = line 2 (put down top row of dots.) 002 = line 3 9 put down next row of dots), etc. . .
51. **Update.** Normally low TTL. goes high in proper slot for character update. Generated by cursor. Used by memory. Inhibited at memory if protected or CTRL being received. Overridden at memory during clear.
52. **CTRL** goes to ground if inputs A6 and A7 are grounded. Indicates a transparent or machine command has been received. Interpreted as a carriage return and line feed in basic unit.
53. **Horizontal out.** Optional 15,840 (with crystal specified) or 15,750 Hz (with external phase lock loop) output useful for interlace and video titling.
54. **Interlace Reset.** Optional. Holds entire vertical counter at maximum count when grounded. First horizontal clock edge following starts new frame.
55. **V output.** 60 Hz used by cursor for synchronizing. Handy for scope sync. Optionally useful for interlace.
56. **-12-V** **25-mA available**
57. **-5-V** **25-mA available**

58,59 +5-V **250-mA available**
60. Ground. Use both grounds on all PC boards.

TABLE IV

KEYBOARD CONNECTOR PINOUTS

A. +12 volts for keyboard if needed (or -12 with jumper)
B. +5 volts for keyboard via CLEAR (HOME) switch
C. KEYPRESSED Normally +. Grounding enters
D. INPUT A1
E. INPUT A2
F. INPUT A3
G. INPUT A4
H. INPUT A5
Inputs A1 thru A5 Must be Grounded during CLEAR (HOME) cycle.
I. INPUT A6
J. INPUT A7
K. SPARE
L. GROUND

All Inputs TTL Compatible. Fits any PC edge connector with 0.156" contact centers or may be directly soldered to flat cable or harness assembly.

WARNING: DO NOT ALLOW ANY INPUT VOLTAGE TO EXCEED PIN B VOLTAGE (Internal +5) OR GO BELOW GROUND.

TABLE V

WHAT DO THE ROCKER SWITCHES DO?

OFF-ON (S1) When this switch is forward, 110 Vac power reaches the transformer. Power is removed with the switch backwards.

LINE-FULL (S2) This switch decides whether a normal full scan is to be displayed or whether a single line or a group of lines is to be shown. In the FULL SCAN (normal operation); pin No. 36 is left open. In the LINE SCAN operation, pin No. 36 is connected to a selected optional clock of timing that forces reset of the one-and-only-one IC7 on the cursor. Extra timing is needed for line scan and varies with the application. A ground is also optionally placed on line No. 32 when in the LINE scan position. For normal use, the switch should be left in the ~~line~~ full position.

A/B (S3) This switch decides which page is to be displayed and which page is to allow character entry. Half the switch controls the output enable or bus access of a memory and half of the switch works with the page protect to allow character entry. Program jumpers on memory A and memory B boards set up which board does what. You normally enable and load the *same* board, except for electronic notebook use. +5 on the enable lines connects you to the output bus. A ground on the PROTECT lines prevents character entry.

KEEP-CHANGE (S4) This is the memory protect switch and overrides S3 by forcing a ground on both the A enable and B enable lines in the PROTECT position.

REPEAT (S5) (momentary) When forward, the blinker clock is applied in place of the conditioned keypressed input, adding characters or moving the cursor at 4 places per second. In its normal position the TV typewriter advances one character for each character entered.

HOME (S6) (momentary) This circuit provides one line thats normally +5V and goes ground and one that does the opposite. To clear the TV typewriter, three things happen: (1) power is removed from the key board, forcing all inputs to the 0 state; (2) a "1" is applied to input A6 via memory diode D4; (3) Flip flop IC3 in the cursor is clamped and held till after the clear is released and system timing indicates a new frame is to begin. If you are in the KEEP position, homing simply resets the cursor and keeps the message intact. If you are in the CHANGE position when you home, the entire message is erased and replaced with spaces or a 100000 code.

CURSOR ON (S7) Moving this switch forward grounds pin 28. This keeps the winking cursor from appearing on the screen.

ADD-SUBTRACT (S8) This controls the cursor and entry direction. In the ADD position, you move forward or down a line. Forward because pins 27 and 30 are shorted to provide a big capacitor and Down because the down clock (P) is connected to pin 29, the line direction pin. In the SUBTRACT position, you move backward or up a line. Backward because only a small capacitor remains between pins 27 and 30 and up since only a brief clock pulse appears when pin 29 is grounded.

PARTS LIST IMPROVED ASCII ENCODER

C1 0.1 - μF disc ceramic Mount *flat.*
D1 to D 4 1N914 or equal silicon computer diode
IC1 HDO165 encoder (Harris)
IC2 7402 TTL quad NOR gate
IC3 MC789AP hex inverter, RTL, **do not substitute**
IC4, IC5 7400 TTL quad NAND gate
Q1, Q2 2N5139, silicon pnp
R1, R2 Varies with keyboard, 1000 ohms for mechanical contacts and +5V supply; 3300 ohms for elastomeric high resistance contacts and +12V supply.
R3, R4 1000 ohms, 1/4-W carbon resistor
MISC: PC Board, Solder; No. 24 soldereeze wire, 20 feet for keyboard wiring, sleeving, No, 24 solid wire jumpers.

NOTE: The following are available from Southwest Technical Products, 219 West Rhapsody, San Antonio, Texas, 78216

Mainframe board: No. TVT-1, $9.75
Timing board: No. TVT-2, $5.75
Cursor board: No. TVT-3, $5.75
Page A or B board No. TVT-4, $5.75
High-quality keyboard, custom remanufactured for TV typewriter use (less-encoder) No. TV-5, $18.75

Mainframe circuitry

The mainframe is shown in Fig. 3 with its PC and component guides in Fig. 10 and 11.

The power supply has its +5 volts regulated by IC1 which must be heatsunk to a vertical radiator. The -5, -12 and optional +12V supplies are Zener regulated. Note that the -5 is derived from the -12 by a series string of a 6.8 and a 5.1 = V Zener. Also, note that some transformers have split secondaries. If yours does, be sure to connect the secondaries aiding rather than bucking. The +5V supply is short circuit proof and good for an amp or more. Long term shorts on the other supplies can damage R1 or R2 by overheating.

The control switches are mounted as shown and their operation is summarized in table V. REPEAT and HOME are momentary switches.

The rf circuitry uses Q1 as an oscillator and D10 as a modulator. It may be disabled for video-only by omitting it entirely or by removing R3. Coil L1 is made up of 6 turns of No. 14 wire wound on a 3/8" form and spaced out to 3/4". Tuning range should be 55 to 80 MHz. Note the tap at exactly one turn.

The current through R7 determines the high frequency rf resistance of the diode and thus the amount of carrier to be amplitude modulated and sent to R8. No current is the white level. A moderate current is black, and a high current is sync, or blacker-than-black. Sync pulses should exceed the black level by 25 to 35% in rf amplitude.

The signal across R8 is our output, but it is a bit strong to directly drive a TV. If we attempt to attenuate it here, coupling from the coil could hurt the TV white level. So, we add 8" of twin lead, and overlap another piece of twin lead 2" from the far end, securing it with plastic hardware or tape. This cuts our signal down to size with capacitive coupling and still doesn't give us any excessively long output radiators.

This, of course, is a TV transmitter and the only way it is legal is if it doesn't radiate much, and above all doesn't interfere with anything. Because of this, the TV typewriter should be housed in a metal case and attenuator C10 should not be eliminated. Be sure to remove all antennas from the TV when operating. Frequency is adjusted with C6. Be sure to tune to an *unused* low channel.

Memory board schematics

There are two types of memory board, A page-A memory includes the line register, character generator, and video output register. One Page-A memory is essential for the TV typewriter. A Page-B memory contains only storage and may be used as a 512 character add-on. Additional page-B memories may also be tacked on to extend the character storage is they are properly enabled, but this gets expensive, and a cassette storage system should be considered above 1024 characters.

Fig. 4 shows the circuitry common to both the Page-A and Page-B memories, while Fig. 8 shows *only* the circuitry needed extra on Page-A. This is followed by the parts lists, PC layouts (Figs. 12 and 13), and mechanical and component details.

Each memory board stores 512 characters in the form of a six = bit ASCII coded character. The memory consists of IC1 through IC6 which are 512-bit recirculating MOS shift registers, driven by the two phase clock driver IC7 and Q1, Q2. These transistors translate the TTL level clock pulses into MOS levels, giving us a swing from +5 to -12 on the clock lines. IC7 increases the power level so that the clock line capacitance can be driven without degrading the clock waveforms.

Note that IC7 has unusual supply connections. It has its most negative pin 7 connected to -12V and its most positive pin 14 connected to -5V and operates on a 7 volt differential. The clock lines may be viewed at the test points with an oscilloscope. Any short, however brief *directly* on IC7's output can damage the device.

Recirculation of the memory is controlled at pin No. 5 of IC1-6. If pin No. 5 is grounded, the memory recirculates. If pin No. 5 is positive, new data gets entered. The UPDATE command normally allows entry via R26, provided that a CTRL (carriage return) command is NOT being received and that the page in use is not being protected.

Memory output is controlled at pin No. 3 of IC1-6. If pin No. 3 is grounded, you get no output. Make pin No. 3 positive to get connected to the bus lines B1 through B6. Jumpers set up the proper Page-A and Page-B connections for enable and protect; normally you protect the page you are *not* working on *or* looking at.

The diodes provide reverse polarity protection, for any of the MOS devices can be damaged if the supplies get mixed up. Even with the diodes, be careful.

Fig. 4 is all you need for a page-B memory. For a Page-A memory, you also need the character generator of Fig. 5. IC7 is the line register that samples the memory on every line No. 1 and brings the characters back eight times in succession for generation. Pulldown resistors R27 to R32 serve for anything that's hung on the bus input B1 to B6.

IC8 does the actual character generation, receiving the ASCII character code at the right time from IC7 and getting a "which-line-is-it?" command on the line address inputs L1, L2, and L4. Enable pin No. 11 gives you an all-white output if it is positive and normal operation if grounded. When you want a cursor, you briefly make this line positive in the proper location on line No. 1.

IC8 transfers its characters to the output register IC9, where they are converted into serial video. An output load command transfers the information when the character generator has been settled and the data is valid. Grounds are provided on the unused parallel and serial register inputs. On the last character of the line, "0"s are marched out for the next 16 character slots, effectively blanking the rest of the line during retrace.

Additional blanking is provided on lines 9 to 12 and vertical retrace by inhibiting the output clock with this input.

IC9's output consists of raw video, black where you want nothing and white where you want a dot. While you can get opposite polarity video on pin No. 7, it generally looks awful and is not recommended for use.

Video is combined with the self-test and sync in the video combiner, getting the sync from the timing board. The output is composite video.

For special use, you can logically connect the blinker to the flash terminal to flash or blink either the entire message or a logically selected portion of the display.

The self-test superimposes itself onto the video with positive-black. It may be connected for test purposes on any TTL circuit in the TV typewriter, and you can use it as a pattern detector or a logic probe.

Composite video output appears on terminal No. 20 and is based on sync-ground, black = 2 volt; white = 5 volts level. An external line driver is needed if you want to use the video directly with long cables, but the circuit impedance is low enough to drive a few feet of unterminated coax if you have to.

The timing board

The timing board may be thought of as two types of circuits: the *main* timing generated as a counter chain from a crystal oscillator, and the *derived* timing that makes up the proper combinations of main timing signals to be useful. The main timing schematic is shown in Fig. 6, while the derived timing appears in Fig. 7, with PC and component layouts in Fig. 14 and 15.

We'll start with the main timing. The circuit is nothing but a string of "by two" and "by six" dividers that take the 4,561-MHz crystal reference and divide it down to generate all the needed, locked together reference frequencies, finally ending up with our 60-Hz vertical output frequency.

IC1 is a dual oscillator. Half of it is a 4-Hz blinker. This is used to wink the cursor, and to provide the REPEAT key action where you want to put down a bunch of identical characters or rapidly move to a new location. It can also be used, with or without logic to flash the display or a portion of the message. This is handy to drive home an important message or change, and is a very simple way to show minus numbers in a calculation.

The other half of IC1 is the 4561.920-kHz oscillator. It is the system reference frequency and the rate at which we clock the output register IC10 on memory board A. Thus, it sets the dot rate for our video put-down. For very fancy applications (video titling, recording, etc..) you can replace the crystal with a capacitor and lift pin 2 from ground. Pin 2 now becomes a VCO input that lets you phase-lock the TV typewriter timing to external circuits or systems. Normally, you use the crystal.

We next go through a divide-by-six in IC2, dropping us to 760.32kHz. This is our basic character rate, and intermediate values

of this divider are used by the derived timing for register clocking pulses. 760.32kHz is also the rate at which the characters are loaded into the output register IC10 on the Memory A board.

From this point, we go into a divide-by-48, made up of IC3 and parts of IC2 and IC4. This divider generates the 48 possible character positions across the line, of which 32 are actually used and 16 are reserved for retrace and overscan. The output of this divider is our horizontal rate, or 15,840 Hz. Note that this is negligibly faster than the usual interlaced 15,750 Hz.

A divide-by-12 in IC4 and IC5 counts horizontal lines for us, directly giving us the "what line is it?" commands for the character generator. It also gives us inputs for derived timing involving the line I transfer.

The output at this point is identified as an "0" clock on the internal test points on the timing board is 1320 Hz. Finally, IC6 and IC5 do a divide by 22 to give us the 22 possible character rows on the screen. Feedback via IC10 of the derived timing ("T" Clock) shortens what would normally be a divide-by 12 to a divide by 11. We use 16 of the 22 lines for characters and save 7 for retrace and overscan. Our final output frequency is the 60-Hz vertical rate.

All of the possible main timing chain signals appear on internal test points A through U, with U being the slowest and A being the fastest. The compliment of G, or G is also brought to a test point since it is useful in the derived timing.

Derived timing

The derived timing is on the same board as the main timing chain. It combines the continuous main timing chain waveforms into suitable "by bursts" signals needed for TV typewriter sequencing and control. Half of IC7 AND,s (Negative Logic) the N and S clocks to give us a 9-12 BLANK signal, an output that is high on each line 9-12, or counting from the top of the tv picture, on horizontal lines 9-12, 21-24, 33-36, 45-48, and so on. The waveform is used to generate the vertical space between characters as well as blanking the characters for the vertical retrace and overscan time. It works by inhibiting (stopping) clock pulses (A Clock) from marching video out of the output register IC10 on the page A memory board.

The same IC also ANDs (negative logic) the K, L, M, and N clocks to give us an output that is low only on each horizontal line 1. (Lines I, 13, 25, 37. . . . . etc. .) The output is used directly as a line-1 transfer command that connects the line register to the memory only during lines 1, 13, etc.. so a new line of characters can be transferred from the memory to the line register. It is also used by IC10 to allow clocking of the memory only on lines chosen.

IC8 generates our line clock by negative logic ANDing the J and D clocks. This gives us 32 clock pulses per line, used to march the characters through the line register. IC9 provides a suitable time delay *after* each register clocking and then provides an output load command to the output register. The time delay is essential, for after you clock the line register, its output takes a brief amount of time to change. This changes the input on the character generator, which also takes a while to get its output correct. Only after we have the right output do we want to transfer the valid new character information into the output register.

Clocks for the main memory are called $\phi 1$ and $\phi 2$. We get these from IC10, which suitable combines some high frequency clocks (B and C) with the Line-1 output, the 32 pulses per line logic, and some other signals. The net result is a pair of 32 pulses per line, present only on lines 1, 13, 25,. . . and of the proper width to drive the clock driver circuitry on the memory boards. Note the clock levels are only TTL here; the transistors and inverters on the memory boards convert them to the full swing MOS levels at very low impedance to drive the capacitance of the memory clock lines.

The final third of IC10 does the shortening of the vertical interval for us, converting IC6 from a divide by 12 to a divide by 11.

IC11 and most of IC12 generate our horizontal and vertical sync pulses and combine them into a composite sync signal. The horizontal pulses are around 5.2 microseconds wide and happen once each horizontal line. The vertical pulses are around 1.5 milliseconds long and happen once each vertical frame. The position of the pulses is adjustable by changing the jumpers shown on the memory board, giving you four possible horizontal positions and three possible vertical positions for a total of 12 potential locations on the TV screen. In cases where a TV badly overscans or when you have a color set or something else you don't want to make any internal position adjustments on, a simple changing of these jumpers will center the picture for you. These could be made continuously adjustable, but the extra complexity of four monostables and two controls didn't seem worth the benefits.

About interlace

Interlace is not normally used, nor is it desirable on a stationary, words-only presentation. You can pick it up if you have to by using the INTERLACE RESET input, which when grounded, resets the entire vertical counting chain to its *maximum* count. When the reset is released, the next whole line of horizontal timing restarts the new frame. To synchronize the internal horizontal with an external system such as a video recorder, you can either lower the crystal slightly to get exactly 15,750 and use this as system timing, or else you can replace the crystal with a capacitor and voltage control IC1 by applying a +3 to +5 volt control signal onto pin 2 of the oscillator. A very simple phase lock loop system then can lock the typewriter terminal to the video recorder or whatever you may be interfacing.

The important point is that you don't need or want interlace for the majority of applications, and the only time you have to have it is when you must *superimpose* your message on top of some existing program material not under your control.

The cursor circuitry

The cursor board decides where and when a new character is to be entered. It also conditions the keyboard inputs, and optionally lets us use a line scan instead of a frame scan, and optionally controls the winking cursor.

It's easiest to look at this board in two parts. The input conditioning and sequencing is shown in Fig. 8, while the actual character position counter is shown in Fig. 9. Both circuits are on the same board and internally connected via tests points A through D.

Since virtually any keyboard or encoder could be used with the TV typewriter, a relatively elaborate conditioning circuit is provided. The input conditioning eliminates contact bounce. It also *waits* after a contact is made for the encoder in the keyboard to catch up and put out valid data. After that, it delivers an update command that lasts *exactly* one frame.

Sometime during the next frame, the character position counter decides where the new character gets put. If it gets entered at all depends on whether the pages are protected or not, and whether there is a CTRL or carriage return command present.

Keypressed signals consist of the KP input (22) going to ground. This input is filtered by R21, C16 to eliminate the worst of the keybounce and noise, particularly any noise on *break* or key release when you are using an elastomeric keyboard or something else without a snap action. Q1 unloads the filter and drives a Schmitt trigger in IC8 that gives us a clean, snap action, and adds noise immunity to the input.

The output of the Schmitt trigger trips a monostable IC9 that gives us around 10 milliseconds of delay. This makes sure the keyboard code is valid and everything is settled before we try entering any data. The output of the delay monostable is converted to a short pulse by C12.

The output pulse from IC8 trips a one-and-only-one synchronizer IC7. This consists of a set-reset flip flop driving a synchronous type D flip flop, and provides an output that lasts for one whole vertical interval, and only one whole vertical interval.

The one frame output goes directly to the update gate back-forth direction control in IC1 and is inverted to handle the update and cursor gates in IC5. How this is done will become more obvious when we talk about the character position counter later. Finally, the one-and-only-one output goes through a CTRL detector in IC8 that decides if a line feed, carriage return or CTRL command is being received. If it is, a CTRL output is generated that prevents the character from being entered. At the same time, a "move up a

line" or a "move down a line" command is delivered to IC5.

We can also optionally "force feed" the one-and-only-one in IC7 with the LINESCAN input, which updates and moves us one character per frame. This option is handy for clocks and calculators but is not used for normal typewriter use. Also, we have set the circuit up so that *any* CTRL command gives us a carriage feed in order to save parts. If you want to you can add extra decoding and logic to independently bring out as many machine commands as you want to.

A keypressed command is random with respect to the frame by frame system timing. So, something between a very small amount of time and an almost full frame has to go by before the one-and-only-one can start with the next frame. The set-reset flip flop in IC7 absorbs this time difference. Up to the entire next frame may be needed for character entry, depending on where the character is. Thus, it takes *two* whole frames worst case to enter a character via the keypressed input. One to synchronize and one to actually enter. This gives us a 33 millisecond fastest possible update rate or about 30 characters per second. The normal computer teletypes run about 10 characters per second maximum; thus the TV typewriter can easily handle their data rates.

Notice that, in interests of economy, the character information lines A1-A7 are unconditioned. This means that the selected character must be valid when the keypressed delay in IC9 ends and must *stay* valid for at least 33 milliseconds after that. For the vast majority of manually operated keyboards, this is no problem at all. For some special or faster systems, you might like to add latches to the input to store the valid data for the length of time it is needed. While you can, in theory update all 512 characters in a single frame, this takes a bunch of more complicated circuitry, and if you can, run your system at less than 30 characters a second (CPS).

This rate will take you about 17 seconds to fill the screen at 30 CPS and around 51 seconds at the 10 CPS typewriter rate. If you have two pages, you could fill one while using the other by changing the update and protect jumpers around.

Character position counter

The character position counter circuit is shown in Fig. 9. Many cursors use an add-subtract or up-down counter that's static and a big comparator to find out when the next character is to go in its proper place. While this works, it's big, expensive, and takes a lot of fancy parts. It's also a bear to debug. We use a much simpler system here that gets the same job done without the need for up-down counters or comparators. It's called a *phase shift counter.* All we do is have a divide-by-512 counter that goes around just like the system timing does, for it is driven by the 512 clock pulses that run the memory. The counter runs *continuously* in bursts just like the memory does. Once each frame, the output suddenly drops, indicating that this is the place to put a new character and that the cursor should also be shown at this time. If we don't tamper with the inputs, the counter always drops on the same place in each frame.

Now, the trick is to back the counter up or move it forward *with respect to the system clock pulses.* Add an extra pulse one frame, and the output drops one count *earlier,* backing us up one character. Hold back one pulse per 512 and the counter goes ahead one character, *changing its relative phase or character position with respect to the system timing.* To make things a little bit easier, we either throw in a short extra count to back up, or a very long extra count that's so long it overrides *two* system clock pulses to go forward. One extra minus two held back is the same as holding one back and a lot easier to do.

So much for the normal character-to-character operation. To get a carriage return, you break the divide-by-512 counter into a divide-by-32 and a divide-by-16, the former for characters across a line and the latter for lines. For carriage return, you reset the character counter to its highest count and hold back or add one line count pulse from the line counter. This returns you to the left and up or down a line at the beginning of the next frame. We either add a brief pulse to move us up a line or a long pulse that overlaps two normal ones to move us down a line.

Finally, to home or get to the upper lefthand corner, we reset both counters to their highest state, and release the counters immediately at the beginning of a frame. This starts everything off on the right foot for a new sequence.

The output of the character position counter drops immediately before the desired character position. It loads a winking cursor command into the proper slot on line 1 of this particular character group. And, if we are in an update cycle, and if we are on an unprotected page, it allows entry of the character by switching the memory from recirculate to update just this one character position.

Turning to the actual circuitry of Fig. 9., the input clock is ANDed with an add character command in IC1. IC2 and half of IC3 form the character counter, while IC4 counts character lines. IC4's output is controlled and distributed by the cursor and memory updates by IC5, under control of the update command from the one-and-only-one and the cursor off-on switch. At the *end* of an update, test point C suddenly drops pulses one of the AND gates in IC1. If pins 27 and 30 are open, this pulse is so brief that it gets added to the normal $\phi 1$ clock pulses and we get an *extra* count pulse, backing us up one character. Short pins 27 and 30, and the pulse is so long that it starts before the first normal clock pulse and lasts till *after* the second normal one goes away. Here, we add one pulse but block two, leaving us in the hole by one pulse. The character counter moves forward one character.

Linefeed is controlled by the flip flop in IC5. To go up a line, the flip flop produces a very brief pulse. To go down a line, the flip flop is set and held exactly long enough to block two normal count pulses. Again, we add one and block two, moving us down a line. Meanwhile the character counter is reset to its maximum count, so that at the beginning of the next frame, we start at the lefthand side.

The final flip flop in IC3 is set on a clear command and released on the beginning of the next field. This holds the counters in the upperleft position until the system clear is released and a new frame begins. The counters are always loaded or cleared to their *maximum* count. This way, the first system clock pulse at the beginning of the frame sets us to zero, instead of one, making sure everything ends up where it belongs.

Construction

Because of the complexity of this project the construction MUST be done in progressive stages and should not be started until the complete story is on hand and thoroughly understood. If one step seems to present problems, DO NOT GO BEYOND ANY OPERATION THAT DOES NOT SEEM CORRECT. STOP AND FIND OUT WHY! Here, very briefly is the suggested building and debugging sequence.

1. Mount the stack connectors. First, very carefully inspect the PC boards for any possible problems. Minute shorts will be extremely hard to find later. Note the connectors are more or less alternated so that the stack fits together one and only one way. Be sure everything on each board follows the same pattern. Be sure the notch on each board goes the same way. After the stack neatly snaps together, add all the jumpers and all the bypass capacitors on all the boards as well as the protection diodes on the memory boards.

2. Build the power supply. If the transformer has two secondaries, be sure to connect them aiding rather than bucking. The 5-V supply uses the 6-V transformer outputs, while the + and -12-V supplies use the 12-V taps. Be sure to watch the proper polarity on everything particularly the Zeners. Use a proper heatsink for IC1, All the switches can be mounted, being very careful not to short anything underneath the switches with the switch pads when they are bent and soldered in place. Check out the supply, looking for +5 on pins 58 and 59, -5 on 57 -12 on 56 and +12 at the optional keyboard power point. Mount the binding posts for ground and self-test. Do not proceed till all the voltages are correct.

3. Build and check the rf modulator. Wind the coil first, 6 turns on a 3/8" mandril, spaced out to 1" long. Vertical mounting leads are then attached, making sure the tap is at precisely one turn. The tab on Q1 is between the emitter and extra case connection. Eight inches of twinlead are attached to the output, and a new piece of twinlead is taped or plastic bolted to that with a 2" overlap, forming as output attentuator and ending up with a suitable connector. To test the rf modulator, temporarily short the video output pin 20. This tells the modulator to put out *maximum* signal. Apply the output to a suitable TV, preferably a high-quality, small-screen

black and white receiver. When power is applied, you should be able to tune the trimmer capacitor to channels 2,3,4, or 5. You should get a completely blank screen and complete audio quieting with proper tuning. There is no sync yet. Do not go beyond this point till you are sure the modulator and oscillator are working properly.

4. Build the timing board and the video output. Add all parts to the timing board, picking three position jumpers at random. Add only IC10 and IC11 and their related parts to the memory A board. Apply power and check for any shorts on +5V. If all is well, you should have 16 rows of 32 white boxes on the screen, and the position jumpers should move you into 3 possible vertical and 4 possible horizontal locations. Pick the best one for the TV you are using. If you have any problems, check first to make sure the main timing chain is ending up with 60 Hz (not nothing or 54 Hz!). Then look for a composite sync output on pin 38, followed by composite video on pin 20. If you have a raster that is blank, look for output load 45, video clock 47, or IC10 problems. The boxes come about since you are loading open-circuit "1"'s onto IC10's inputs. Don't forget to remove the pin 20 ground from step 3!

This is the first plateau of the construction. Beyond this point, things are more or less self-checking. At this time it is a good idea to go through and check *every* terminal in the system with the self-check input, carefully noting everything, learning as much as you can about the timing, and looking for potential problems. (The boxes may be erased by temporarily shorting two output registers to ground. If you can't get to this point, an oscilloscope with a triggered sweep is almost essential for servicing. Beyond this point, the circuit more or less services itself.

DO NOT GO BEYOND THIS POINT UNLESS YOU HAVE 512 WHITE BOXES STABLY AND CLEARLY DISPLAYED WHERE YOU HAVE SELECTED ON THE TV SCREEN.

5. Add the character generator. First check the place where IC9 is to go for -12V on pin 1, +5V on pin 24, and -5V on pin 12. Add the pullup and pulldown resistors and solder IC9 in place. As with all the MOS IC's in this project, leave them in their protective foil or foam and quickly solder them in place with a small soldering *iron.* Always be sure all related circuitry is in place *before* adding any MOS IC. Very briefly apply power. You should get a screen full of @'s. IF YOU DON'T GET A SCREEN FULL OF @'s STOP IMMEDIATELY AND FIND OUT WHAT IS WRONG'

Once you have the screen full of at's, make up a jumper consisting of a 330-ohm resistor connected to +5 and briefly connect this one at a time to the signal end of R45 through R50. The character should change from "at" to A to D to H to P to blank and back again. If you get these characters, you'll probably have the rest of them, as we'll find out in the next step. Again, do not go beyond this point until you can display at your command "at", A's, D's, H's, P's, and blanks. You might like to use your self-test again with a blank screen to look at waveforms. Many TV's are reluctant to present an all-white screen with only tiny portions black, so positive logic signals (those that spend most of their time at ground) will generally look bad. These are in the minority. If you want to see them, get a good scope or a better TV. If you have any misgivings about anything on the project at this point, STOP and find out what the problems are. The sharpness of the characters will depend on the accuracy of the tuning and the quality of the TV, but with any reasonable TV, you should get a reasonable size display of good sharpness. Ghost images may mean tuning inaccuracy or transmission line reflections. Run your hand up and down the lead-in to see if this is a problem. Add a couple of 300-ohm resistors if this is a problem.

6. Add the Line Register. Check with the self-test for a line one transfer output (one white line—eleven black ones, repeated 16 times), and a line clock (32 narrow vertical white stripes). Add the remaining Page-A memory parts, ending up with IC8, following the usual MOS precautions. Once again, you should get a screen full of "ats", and running up and down the bus B1 through B6 (pins 9-14) should generate the now familiar at A-D-H-P-blank sequence. The only difference now is that we are loading and storing characters for eight lines at a time instead of just displaying them.

The load and store cyclic sequence can be checked in the following way. Temporarily jumper the timing board R clock to bus B1. This should give you half at's and half blanks on the screen. The Q clock should give you two stripes of at's, four rows per stripe. The P clock should give you four stripes of at's, alternating two rows per stripe. And the 0 clock should alternate rows of at's and blanks, one row each. Now for the key "is it storing?" test. The J, K, L, M, and N clocks should NOT change the pattern from one of all "at's", but the I clock should turn the right half of the screen blank, giving you a vertical stripe. Similarly, the H clock should give you two vertical stripes, the G four, the F eight, the E sixteen. D, C, B, and A clocks should do nothing. If these tests are OK, the line register is probably working properly. If not, double check for line clock, line-one transfer, proper connections, etc...

Now, we can have some fun. Reconnect the B6 line to the 0 clock. Connect B5 to the I clock, B4 to the H clock, B3 to the G clock, B2 to the F clock and B1 to the E clock. You should get the entire alphabet, numbers, and punctuation spelled out, repeating every two lines.

DO NOT GO BEYOND THIS POINT UNLESS YOU ARE COMPLETELY AND STABLY DISPLAYING THE ENTIRE ALPHABET, REPEATED EIGHT TIMES.

7. Debug the Cursor. Add all components to the cursor board. Set the TV typewriter up for a screen full of at's and add the board. The display should not change, except for the possible appearance of the cursor. Check first the Cursor on-off switch and then the Clear to home the cursor. Now switch to repeat and check out the back forth and up-down cursor motion. Pins 2 and 3 on the connector must be grounded for up-down operation.

Next, go to a keyboard or a set of switches that has at least a working spacebar and carriage return key, and check out the cursor for one-at-a-time operation. If you have any troubles, use the internal test points, starting with H. H is a squared and delayed keypressed input. G is a brief pulse 10 milliseconds later. C is a gate that lasts one frame on keypressed command. B is its inverse, and A is identical to B but is present only when a CTRL is present, e.g. pins 2 and 3 on the connector or input bits A6 and A7 are grounded.

Finally, F shows the input to the character position counter, while E is the mid point in the counter. At F, you should have 512 pulses per normal frame. On a frame where an update is to take place or the cursor is to be moved, you should get an *additional* pulse in the upper left. A very brief one for left or upwards motion, and a longer one that overlaps exactly two normal clock pulses for right or downwards motion.

DO NOT GO ON TILL THE CURSOR IS WORKING TO YOUR SATISFACTION.

8. Add the Memory Clock Driver. Add all the remaining Memory A board parts *except* IC1 through IC6. Check at the two clock test terminals for two groups of 512 dots, using the self-test or preferably a good scope. If you have a scope, look at these points for a normally +5V waveform that drops to -12V for 200 ns sharply and repeating 512 times per frame. A short on the output of IC7. however brief, can damage the device. This is why R10 and R12 are needed.

DO NOT CONTINUE UNLESS YOU ARE REASONABLY CERTAIN THE MEMORY CLOCKS ARE PRESENT AND WORKING AT THE RIGHT AMPLITUDE.

9. Add one memory. Now add your keyboard or test switch group and IC5. As with the other MOS chips, leave these in their protective foam until immediately before use, then quickly solder them in place with a small soldering iron. When the protect and page select switches are properly set, you should be able to selectively store at's and P's wherever you want to on the screen. Note that holding the CLEAR down forces a load command onto the memory. This is useful for test, although you have to derive the encoder +5V temporarily from sonething besides the NOT CLEAR source.

10. Add another memory. IC6 this time. Now, you should get at's, zeros, blanks and P's wherever you want, and when the keyboard is properly set up, you should be able to clear the screen on command, again with the protect and page select switches properly set. Note that the keyboard encoder must put out either 100000 or a 000000 code during clearing. This easiest done be deriving the keyboard power from the unclear portion of the clear or home switch. DO NOT GO ON UNTIL YOU CAN ERASE

AND SELECTIVELY ENTER THE FOUR CHARACTERS WHEN AND WHERE YOU WANT TO.

11. Finish the Memory. Add the remaining four IC's ONE AT A TIME and test them individually. Each new IC doubles the number of characters you can handle, to eight, sixteen, thirty two, and finally the full sixty four characters. At this point, the characters should correspond to the typewriter keys and you should have a fully working unit. If you are also using a page-B, after you are sure everything is working correctly, repeat steps 8 through 11 above to complete your TV typewriter.

A note on debugging

This entire circuit was designed and debugged through several revisions with nothing but a beat up old TV set and a VOM! Thus, by using the self-test and reasoning things out carefully, exotic test equipment should not be necessary to get yoiur unit to work– BUT CAREFUL WORK AND REASONED, LOGICAL TESTING, WILL BE ABSOLUTELY ESSENTIAL. If you have a scope, by all means use it. If you don't, it won't hurt any. The important thing is to not let a batch of exotic equipment lull you away from a careful debugging.

On any project this complicated, it is virtually impossible to eliminate all the little bugs in the circuit until many units have been assembled by many people. Thus, if you are one of the early builders of this circuit – BE CAREFUL AND EXPECT PROBLEMS! Above all, tell us what you found out and how you fixed them, so we can make suitable corrections. We have made every effort to be as accurate as possible.

As a final debugging note, one that should be obvious–the PC boards can go on the stack in any order. Thus, if you seem to have a problem board, put it on top, where the self test can be best used to full advantage.

Picking a keyboard

Any keyboard will work with the TV typewriter–provided it is connected and encoded in such a way that you get seven bits worth of TTL compatible ASCII code out and a keypressed output that is normally high and drops to ground when a key is pressed. The trouble that you may run into if you do use just any keyboard is that the conversion and encoding process may get tricky or expensive, or perhaps some keytops will have to be rearranged and relabeled. Let's look at some popular alternatives:

1. Six switches and a pushbutton. This always works and is the cheapest possible route. This is also almost essential for initial debugging. The trouble, of course, is that its hard to learn the code and a pain to gain any speed this way.

2. Kit Keyboards. While the supply lasts or continues to be available, high quality, genuine *Microswitch* sealed reed operated keyboards are offered from the kit source with proper markings and encodings.

3. Radio-Electronics Low Cost Keyboard and Encoder: These articles appeared in the February and April 1973 issues and showed how you could build your own custom keyboard for under 25¢ a key. Some fairly fancy mechanical work is involved.

4. Any single contact new or surplus electrical keyboard and the New Radio-Electronics Encoder. This encoder (scheduled for Nov. 1973) is an improved version of the original that uses far fewer parts. The new schematic appears here, and it is available in kit form. It will convert any keyboard that consists of spst normally open key contacts, but the keyboard cannot have a common bus for one side of the key contacts. This works equally well with mechanical or resistive, elastomeric contacts. One thing the encoder must have is ASCII pairings on the keys. Thus a "capital comma" has to be a >, a "Capital 2", and so on. The needed pairings are shown in table II and Figure 21. Some keytops may need remarking if they aren't standard ASCII to begin with.

5. Teletype Computer KSR Units – these are mechanical keyboards that occasionally crop up surplus. They are self encoding and need very little to interface them with the TV typewriter. The code is usually upside down, so it has to be inverted with hex inverters or something similar. Some types have a solenoid recocking mechanism that needs reset after a character is entered. These are rather bulky and not too modern looking, but are a good route if you can find one.

So much for the good guys. Now, on to the bad scenes:

6. Old Ham and Western Union Teletypes. These use an essentially obsolete 5-bit Baudot code, besides being kludgey and noisy. You need an elaborate code converting read-only-memory to use them, as well as a flip flop to keep track of shifts. A pain, and if you happen to have one and can get a ROM programmed cheap enough, it may work.

7. Old Friden Flexowriters. Also a bad scene. They are a special code and need ROM conversion.

8. IBM Selectric Keyboards. These are a special SELECTRIC code and are purely mechanical, besides having 2 bunch of the keypairings wrong. They can be used by adding seven keyswitches, changing the keytops and adding a Selectric to ASCII read-only memory. The latter is a stock but expensive integrated circuit, or you can cut your own.

9. Other electric typewriters – forget it. All normal electric typewriters are mechanical, not electric. All the electric does is turn the crank for you. They are purely mechanical beasts without contacts that are totally unreasonable to encode to ASCII.

10. New, commercial terminal keyboards – these are ideal, beautiful, and perfect. Let us know if you find any for under $250 in single quantities!

TABLE III

MAIN CONNECTOR PINOUTS

+ ● ● ● ●	1. Ground	+ 0 0 ● 0	31. Protect A
+ 0 ● ● ●	2. Input A7	+ 0 0 0 0	32. Calculator Control*
+ 0 ● ● ●	3. Input A6	+ 0 0 0 ●	33. Protect B
+ 0 0 ● ●	4. Input A5	+ 0 0 ● 0	34. Enable A
+ 0 0 ● ●	5. Input A4	+ 0 0 0 ●	35. Enable B
+ 0 0 ● ●	6. Input A3	+ 0 ● 0 0	36. Line/Frame scan sel.*
+ 0 0 ● ●	7. Input A2	● + 0 0 0	37. Line Scan Clock*
+ 0 0 ● ●	8. Input A1	0 + 0 ● 0	38. Sync In
0 0 0 + +	9. Output B1*	0 0 0 0	39. (spare)
0 0 0 + +	10. Output B2*	0 0 0 0 0	40. (spare)
0 0 0 + +	11. Output B3*	0 0 0 0 0	41. (spare)
0 0 0 + +	12. Output B4*	0 0 0 0 0	42. (spare)
0 0 0 + +	13. Output B5*	0 0 0 ● 0	43. Flash Display*
0 0 0 + +	14. Output B6*	0 0 0 0 0	44. (spare)
0 + 0 ● ●	15. Memory clock ϕ1	0 + 0 ● 0	45. Output load
0 + 0 ● ●	16. Memory clock ϕ2	0 0 0 0 0	46. (spare)
0 + 0 ● 0	17. Line 1,13,25. .Transfer	0 + 0 ● 0	47. Video Clock
0 + 0 ● 0	18. Line Register Clock	0 + 0 ● 0	48. L4 command
0 + 0 ● 0	19. Self Test	0 + 0 ● 0	49. L2 command
● 0 0 + 0	20. Video Out	0 + 0 ● 0	50. L1 command
0 + 0 ● 0	21. Blank 9-12,21-24,etc.	0 0 + ● ●	51. Update
+ 0 ● 0 0	22. Keypressed	0 0 + ● ●	52. CTRL output
● + ● 0 0	23. Blinker	0 + 0 0 0	53. Horiz. Out*
+ 0 0 ● ●	24. Clear(+5 on Clear)	0 ● 0 0 0	54. Interlace Rst*
+ 0 ● 0 0	25. Unclear (Gnd on Clear)	0 + ● 0 0	55. V output
+ ● ● 0 0	26. up-down direct. cntrl.	+ 0 0 ● ●	56. -12 V
+ 0 ● 0 0	27. right-left control	+ 0 0 ● ●	57. -5V
+ 0 ● ● 0	28. Cursor ON	+ ● ● ● ●	58. 5V
+ 0 ● 0 0	29. down direction clock	+ ● ● ● ●	59. 5V
+ 0 ● 0 0	30. right-left control	+ ● ● ● ●	60. Ground

mainframe — timing — cursor — memory A — memory B

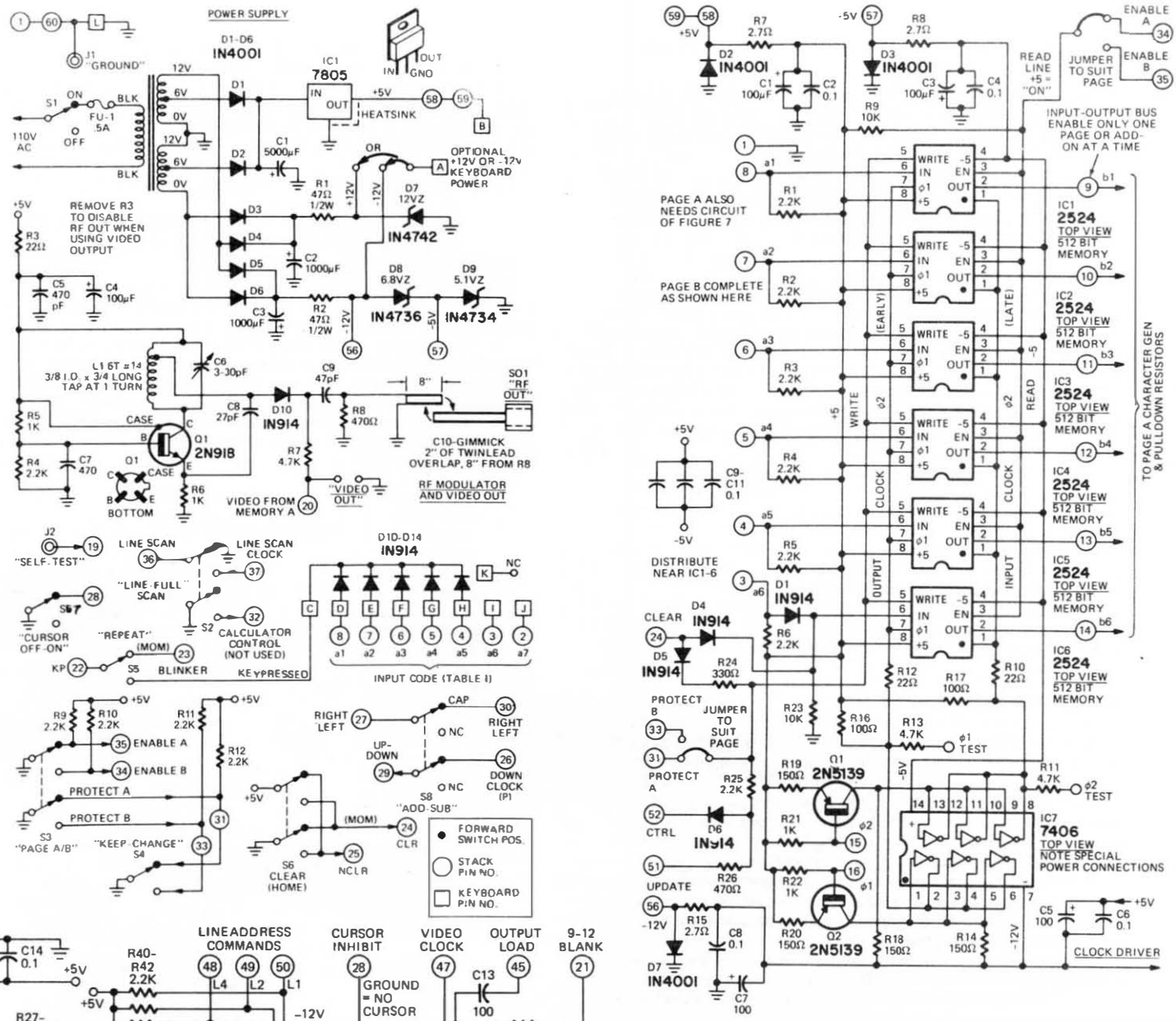
*-- optional pin not used in basic typewriter circuit.

+ - Signal source

● - Signal used

0 - Signal not used.

While a far simpler connector system could have been used, this system lets the modules snap together in any manner, so that a module to be tested ends up on top of the stack. It also eliminates any interconnection wiring, the biggest potential source of problems on a system of this type.



TOP LEFT

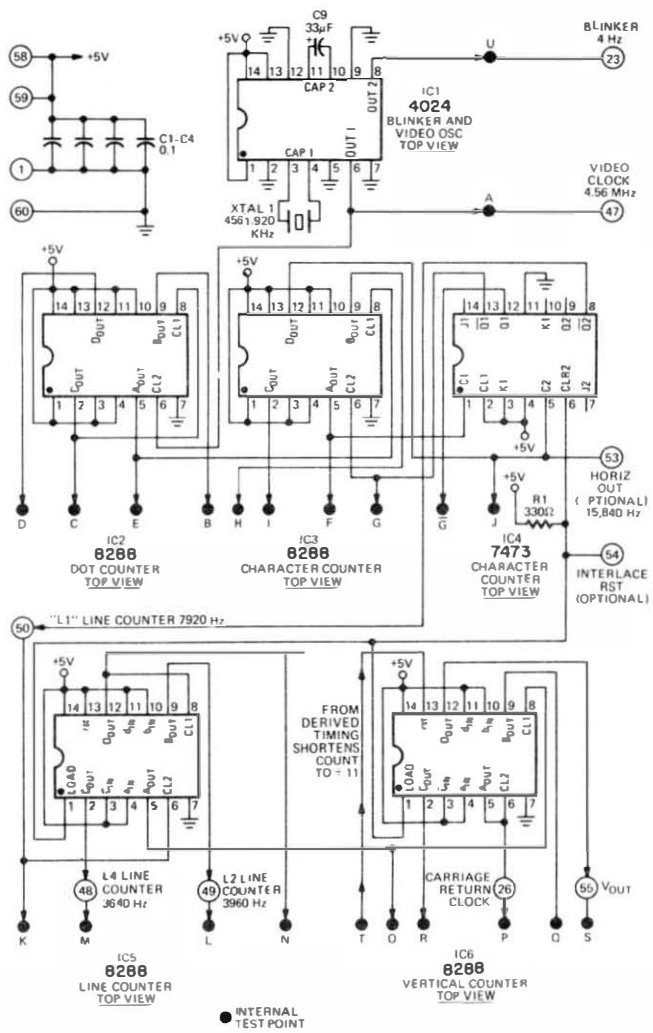
FIG. 3 -- MAINFRAME SCHEMATIC of TV Typewriter, includes power supply.

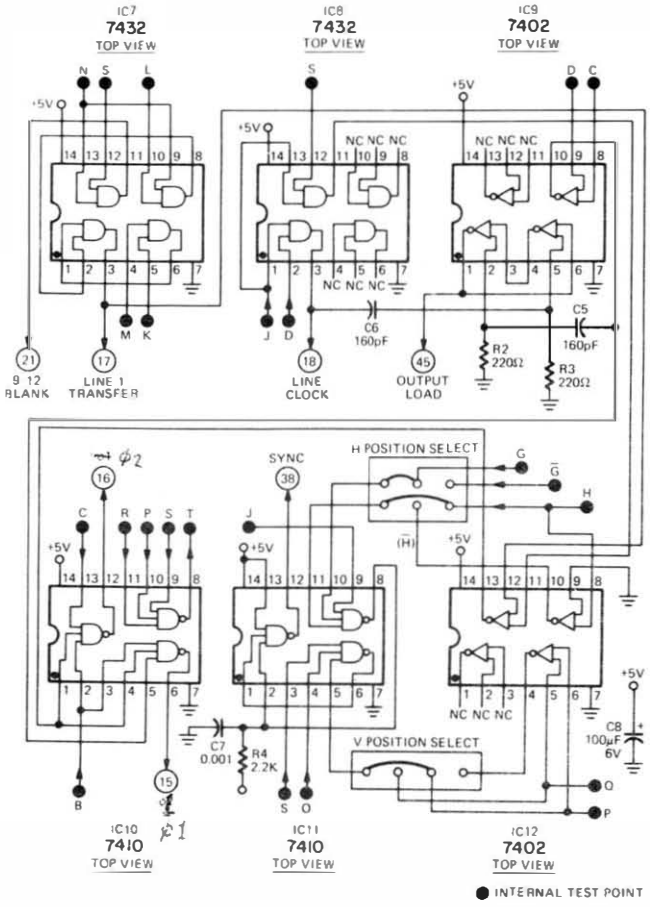
ABOVE

FIG. 4 -- PAGE "A" OR PAGE "B" memory circuitry.

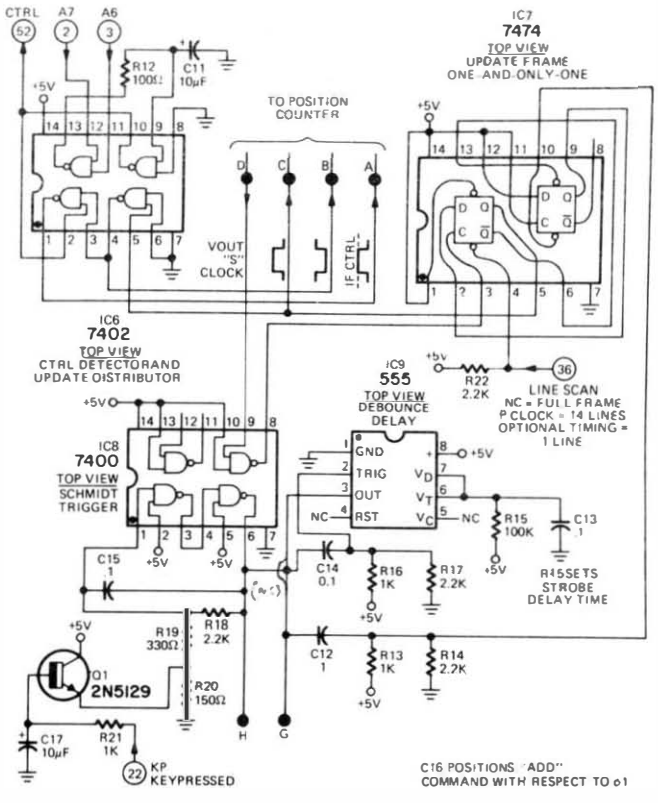
LEFT

FIG. 5 -- PAGE "A" character generator circuitry is needed only on "A" page circuit board.





TOP LEFT
FIG. 6 -- MAIN TIMING CHAIN schematic.



ABOVE
FIG. 7 -- DERIVED TIMING schematic.

LEFT
FIG. 8 -- CURSOR INPUT conditioning and sequencer circuit.

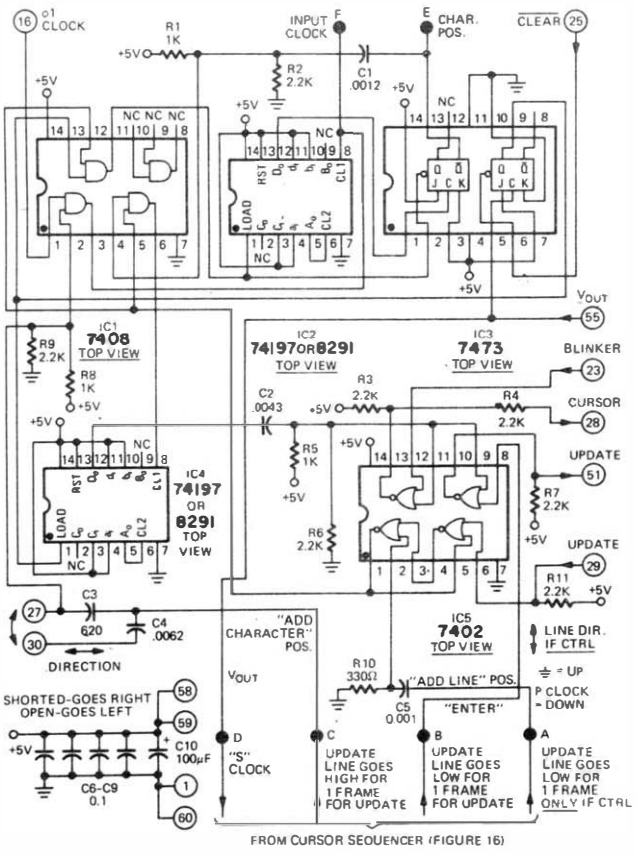
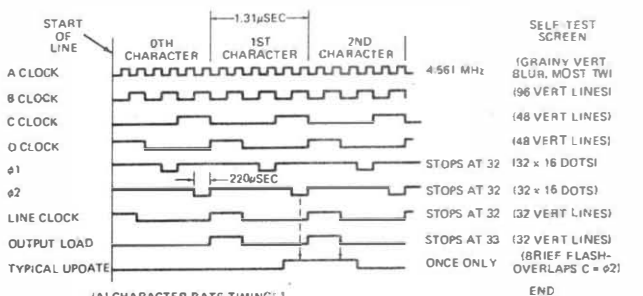
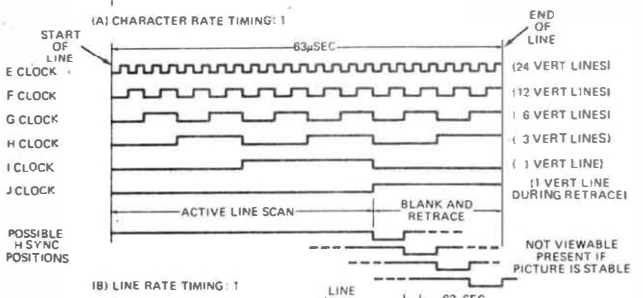


FIG. 9 -- CURSOR BOARD -- character position, counter schematic.

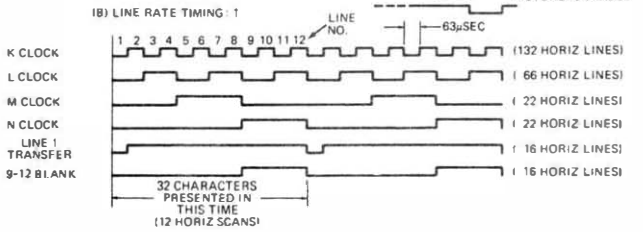
KEY WAVEFORMS at various points of TV Typewriter circuit.



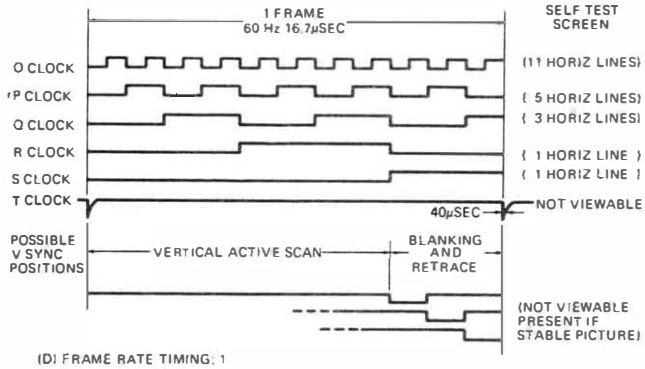
(A) CHARACTER RATE TIMING: 1



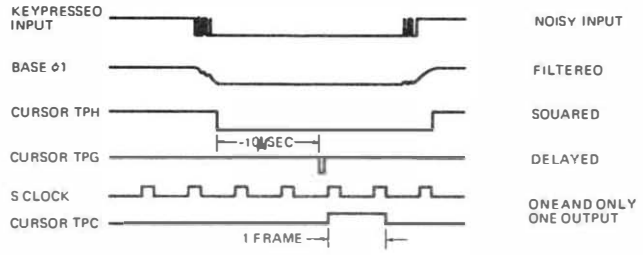
(B) LINE RATE TIMING: 1



(C) CHARACTER ROW TIMING: 1



(D) FRAME RATE TIMING: 1



(E) INPUT CONDITIONING: 1

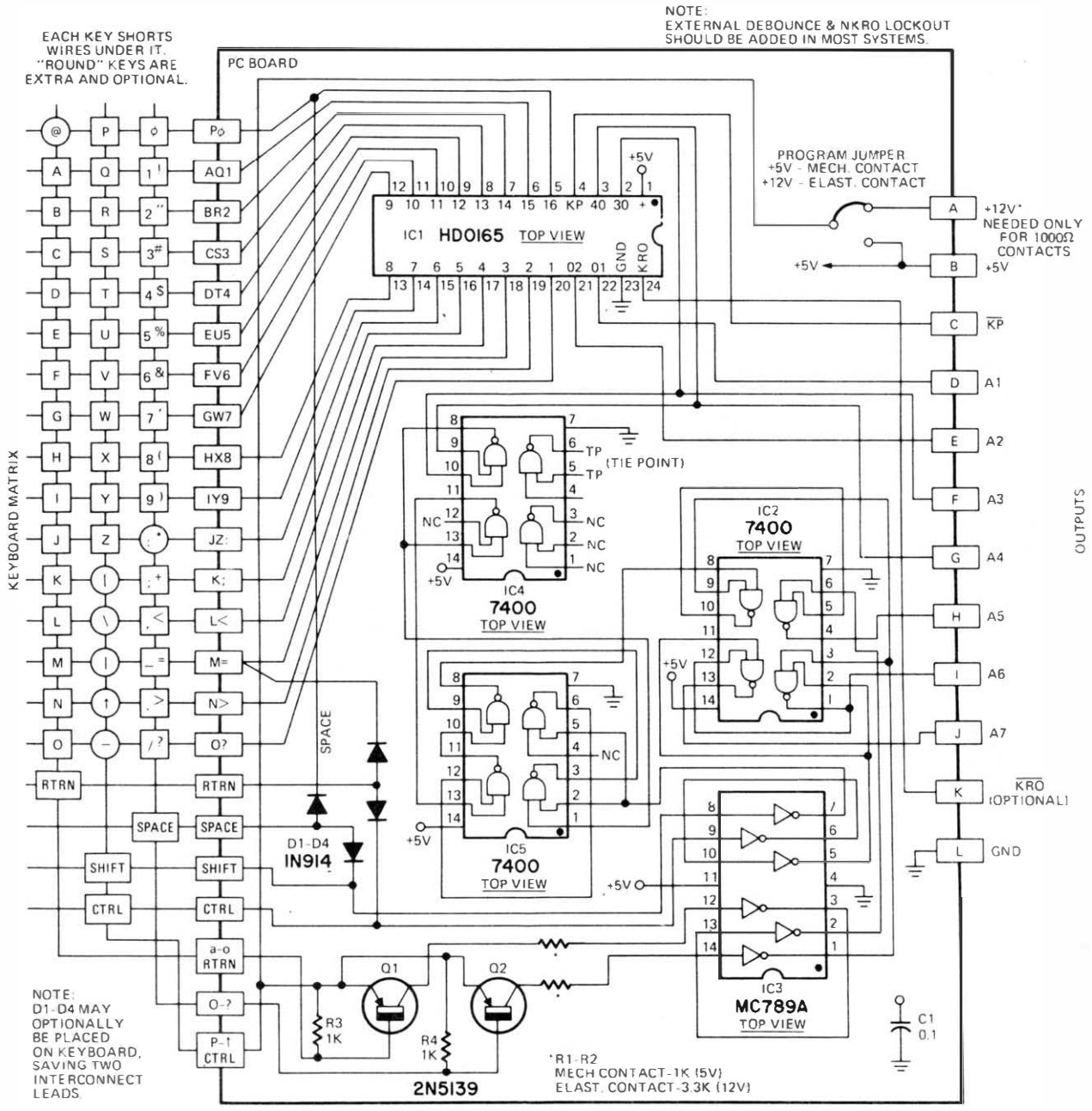
GO FORWARD ONE SPACE:
WAVEFORM STARTS BEFORE FIRST φ1; LAST THRU SECOND φ1, BUT STOPS BEFORE THIRD φ1 CLOCK ON FRAME FOLLOWING UPDATE

GO BACK ONE SPACE:
BRIEF PULSE BEFORE FIRST φ1 ON FRAME FOLLOWING UPDATE

GO DOWN ONE LINE:
P CLOCK RESETS CURSOR ICS AFTER TWO LINE CARRIES HAVE PASSED BUT BEFORE THIRD ONE ON FRAME FOLLOWING UPDATE

GO UP ONE LINE:
BRIEF PULSE BEFORE FIRST LINE CARRY ON FRAME FOLLOWING UPDATE

(F) CURSOR MOTION: 1



IMPROVED ASCII ENCODER schematic.

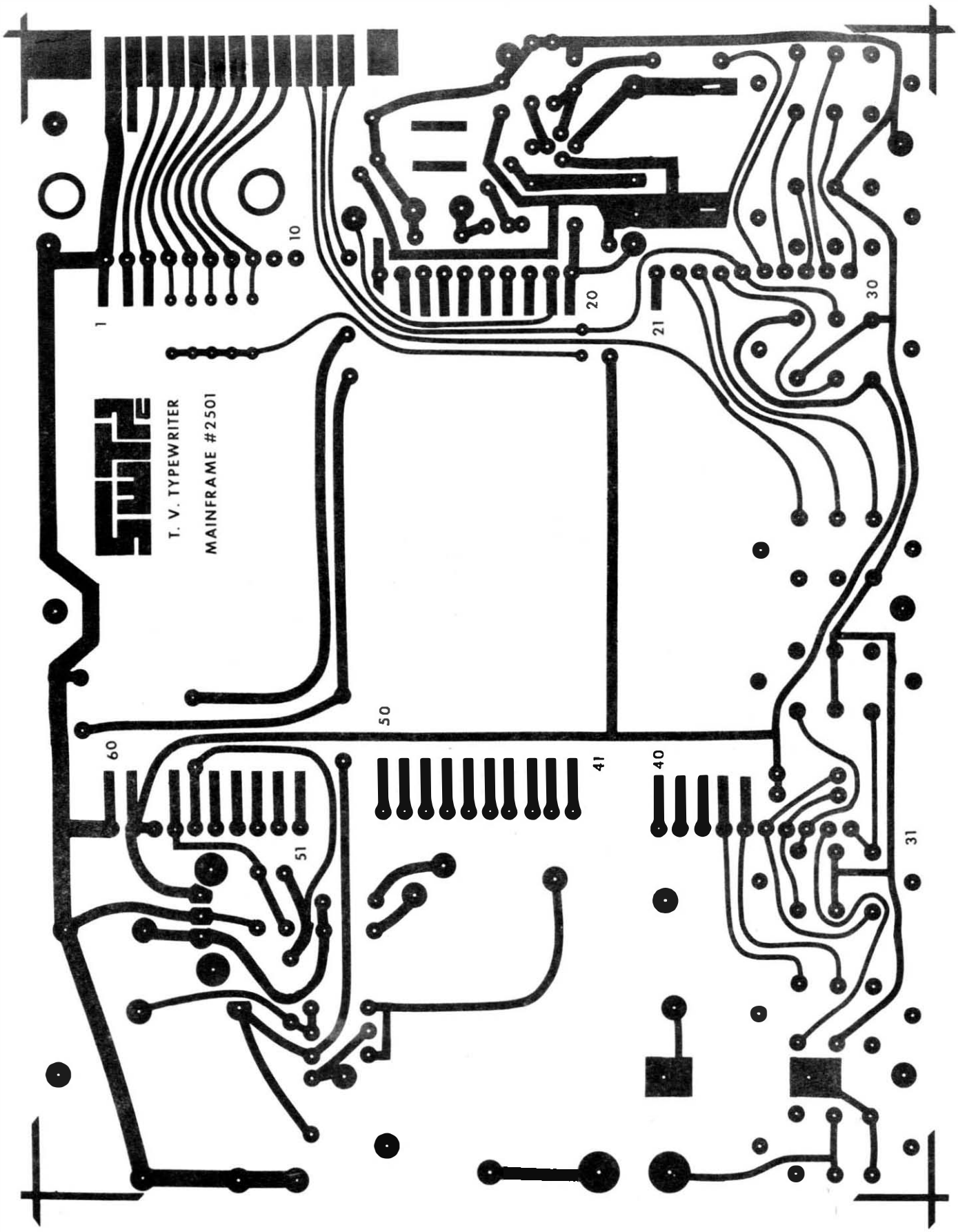


FIG. 10 -- FULL SIZE FOIL PATTERN
for mainframe.

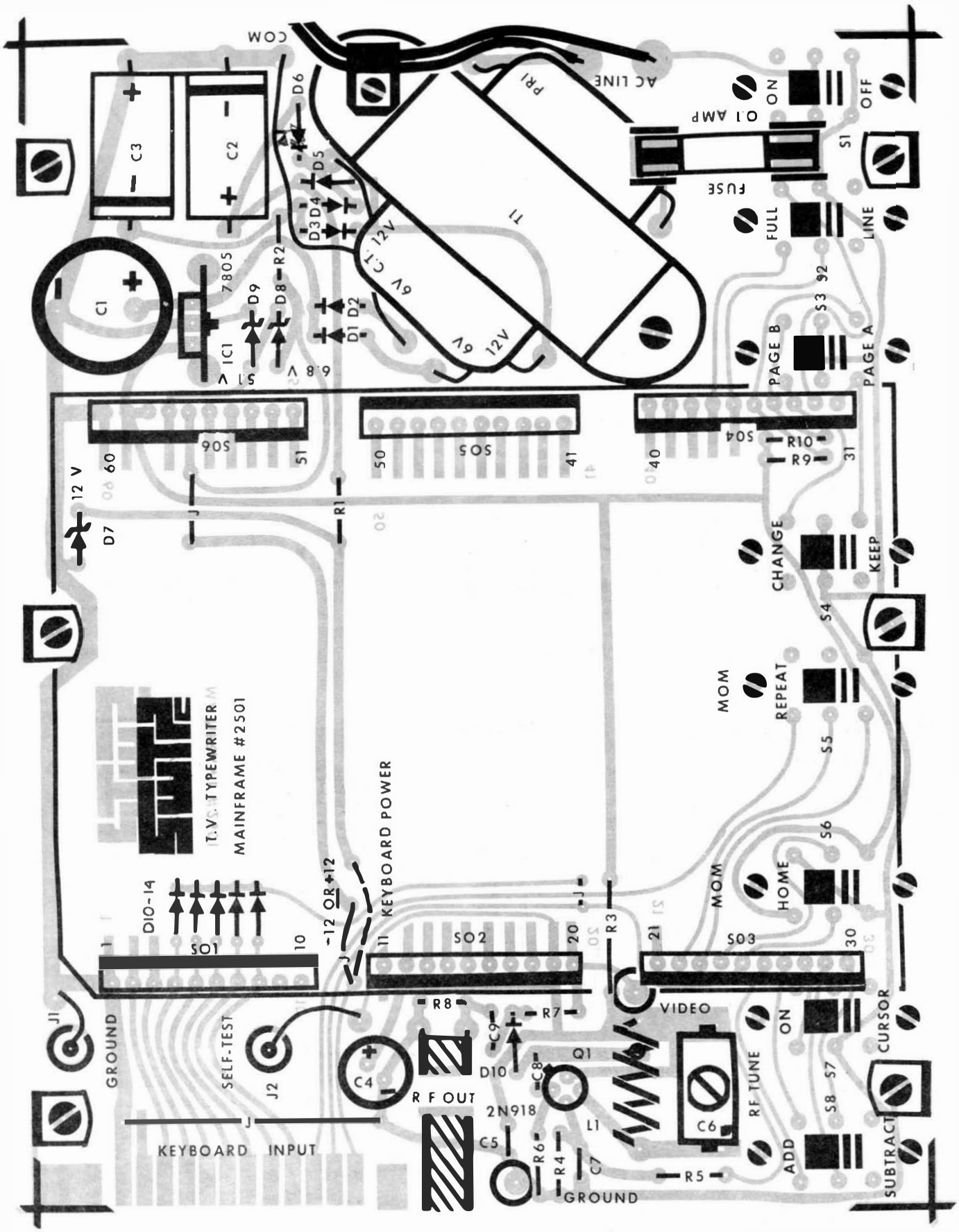


FIG. 11 -- PARTS PLACEMENT on main-frame circuit board.

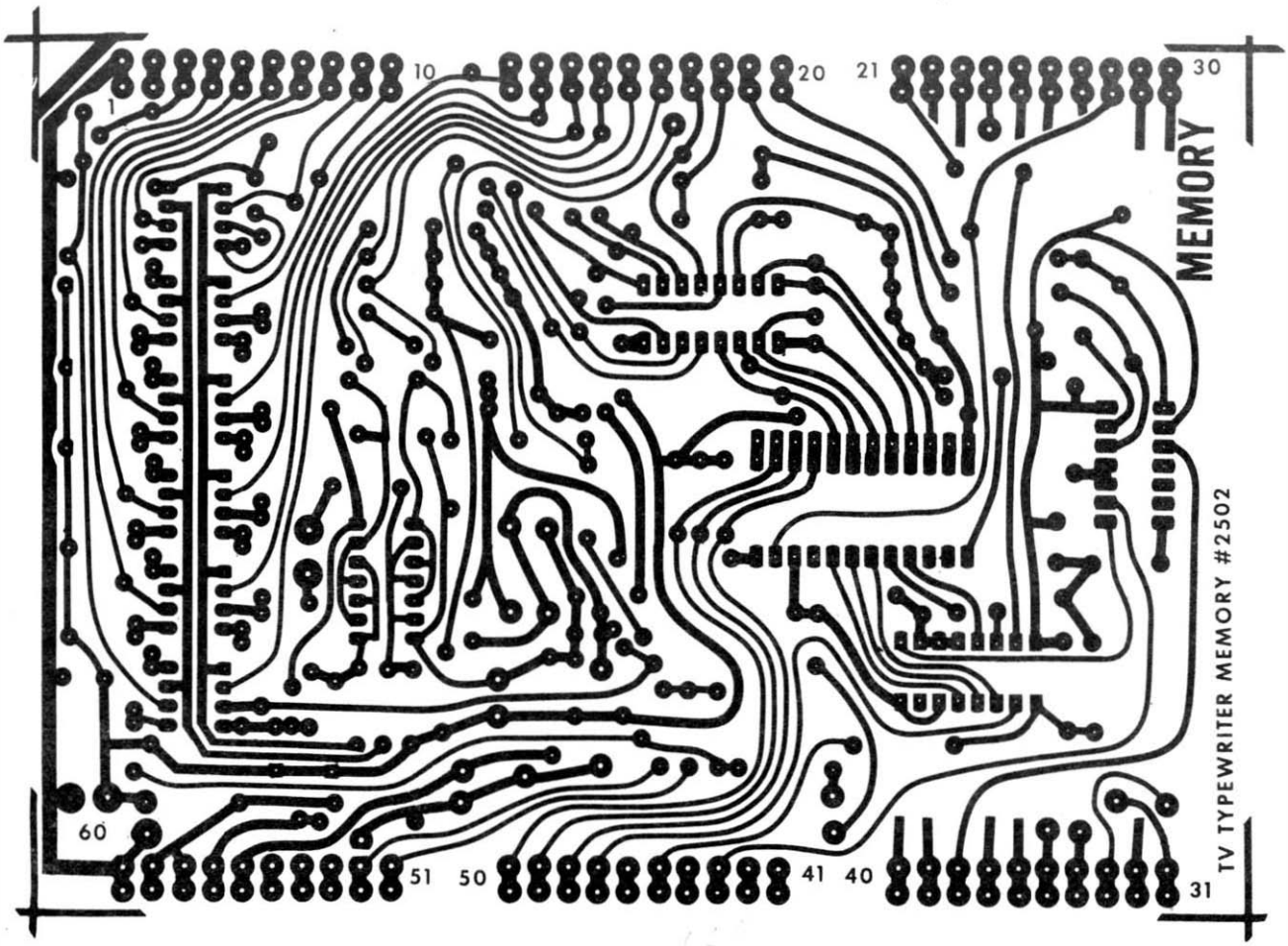


FIG. 12 -- FULL SIZE FOIL PATTERN for memory board.

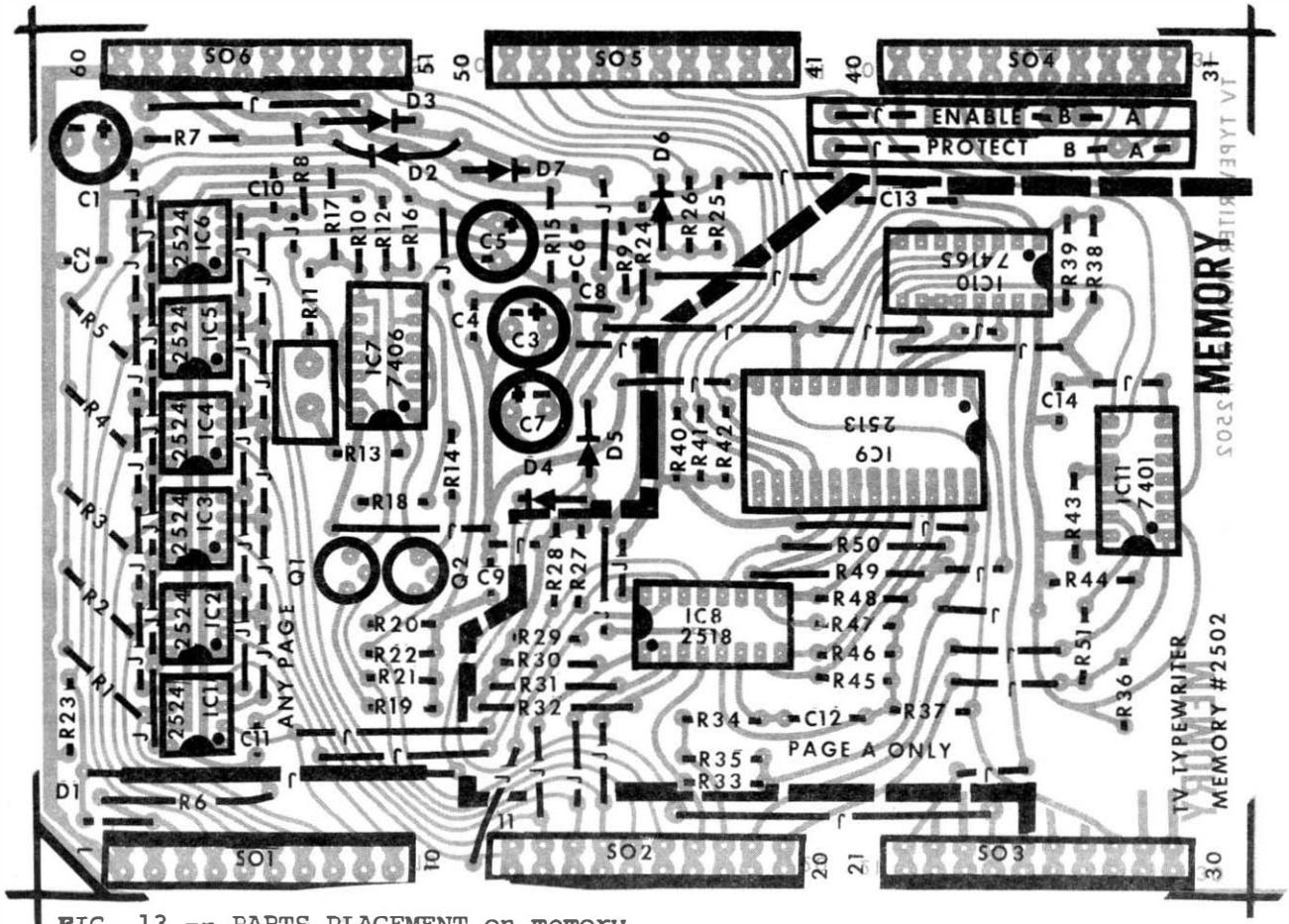


FIG. 13 -- PARTS PLACEMENT on memory circuit board.

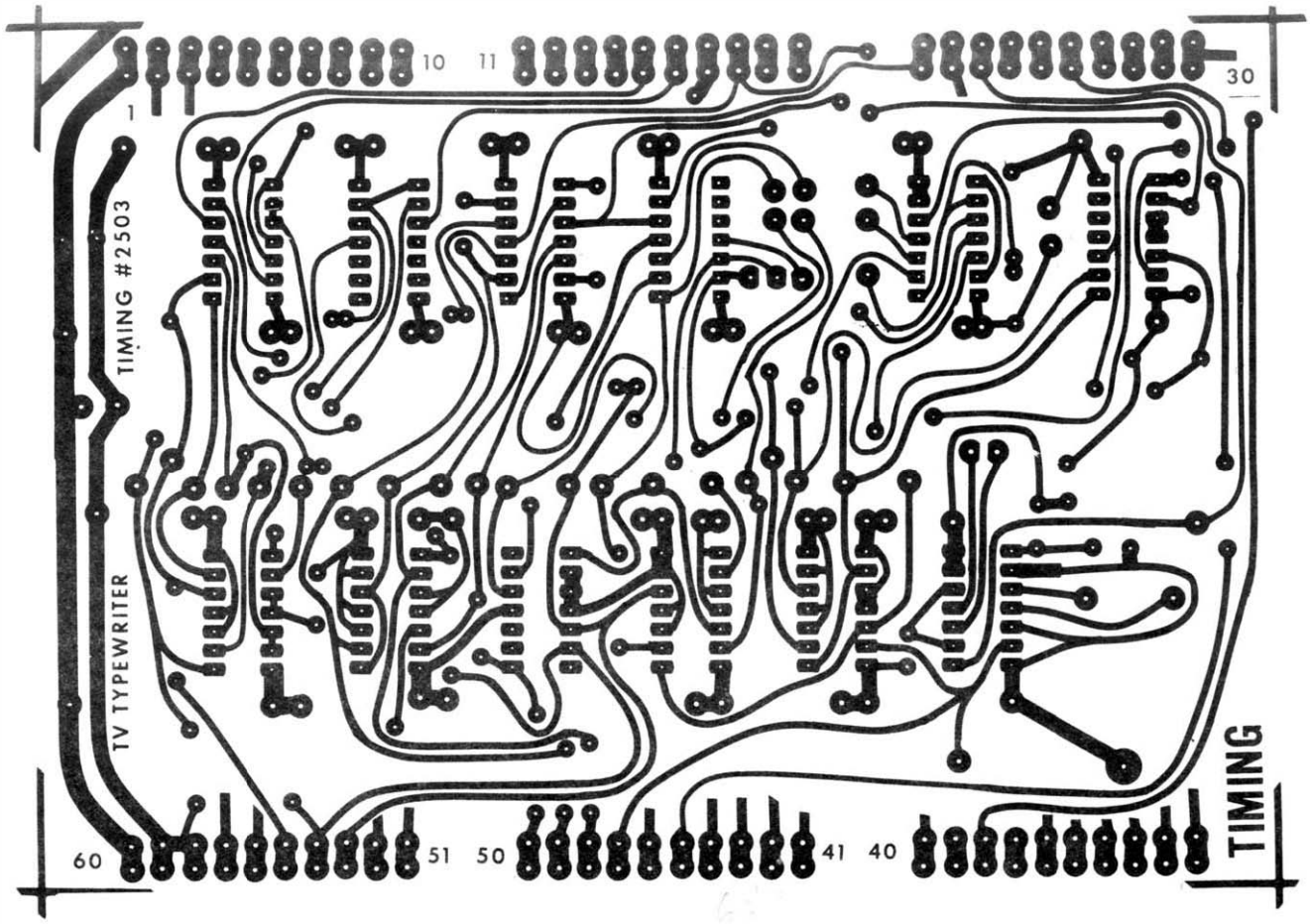


FIG. 14 -- FULL SIZE FOIL PATTERN
for timing board.

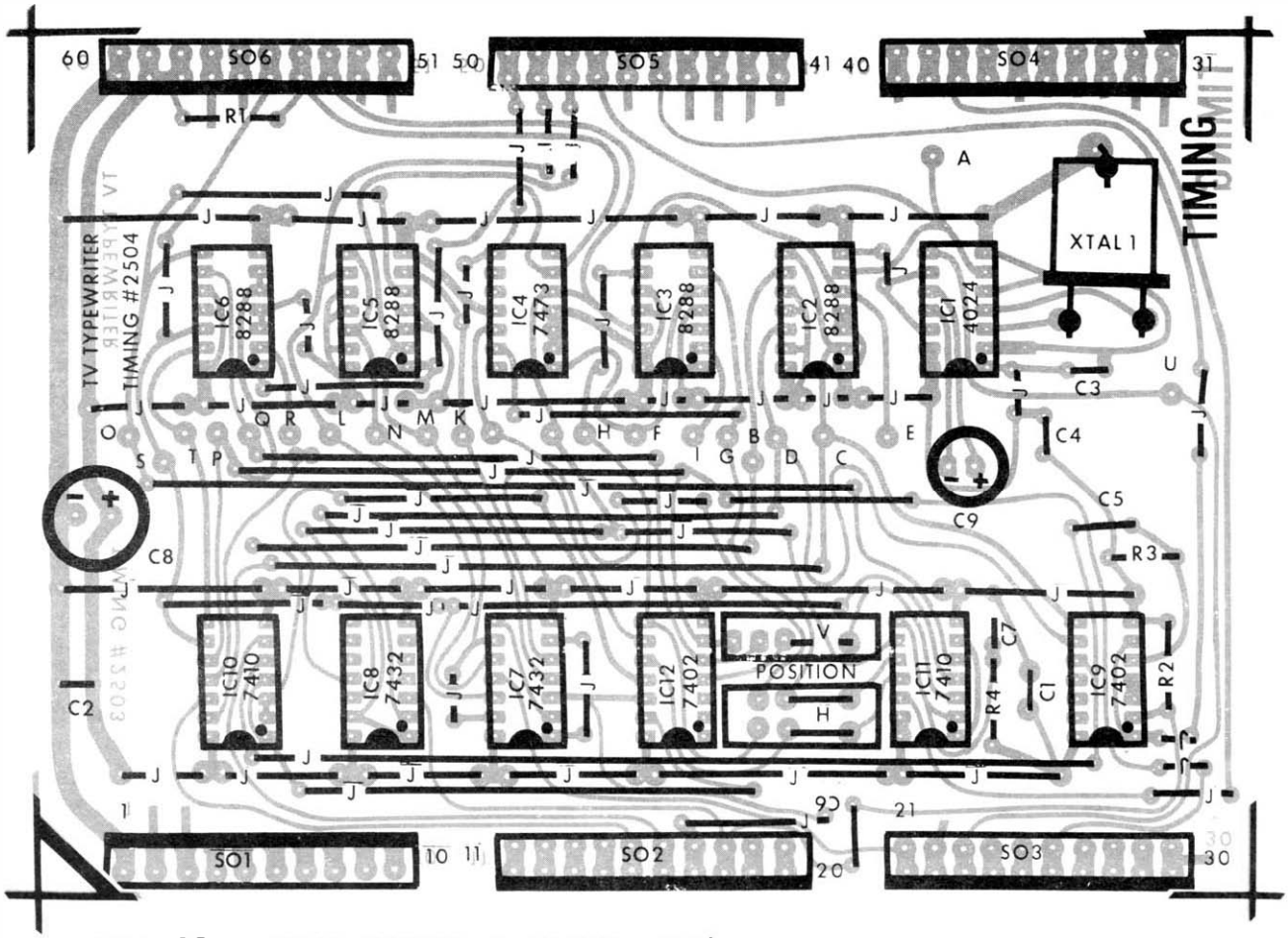


FIG. 15 -- PARTS LAYOUT on timing board.

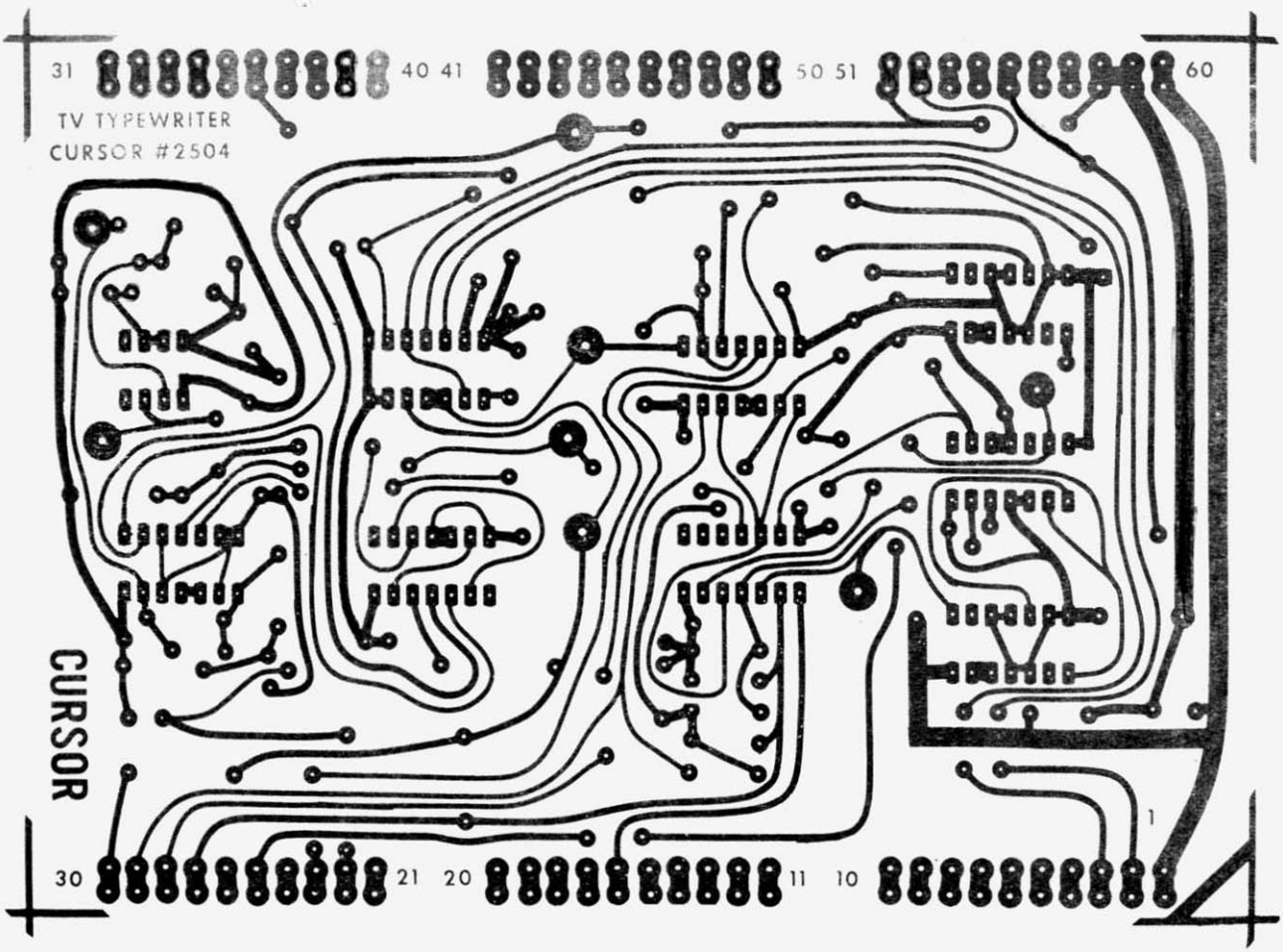


FIG. 16 -- FULL SIZE FOIL PATTERN for cursor board.

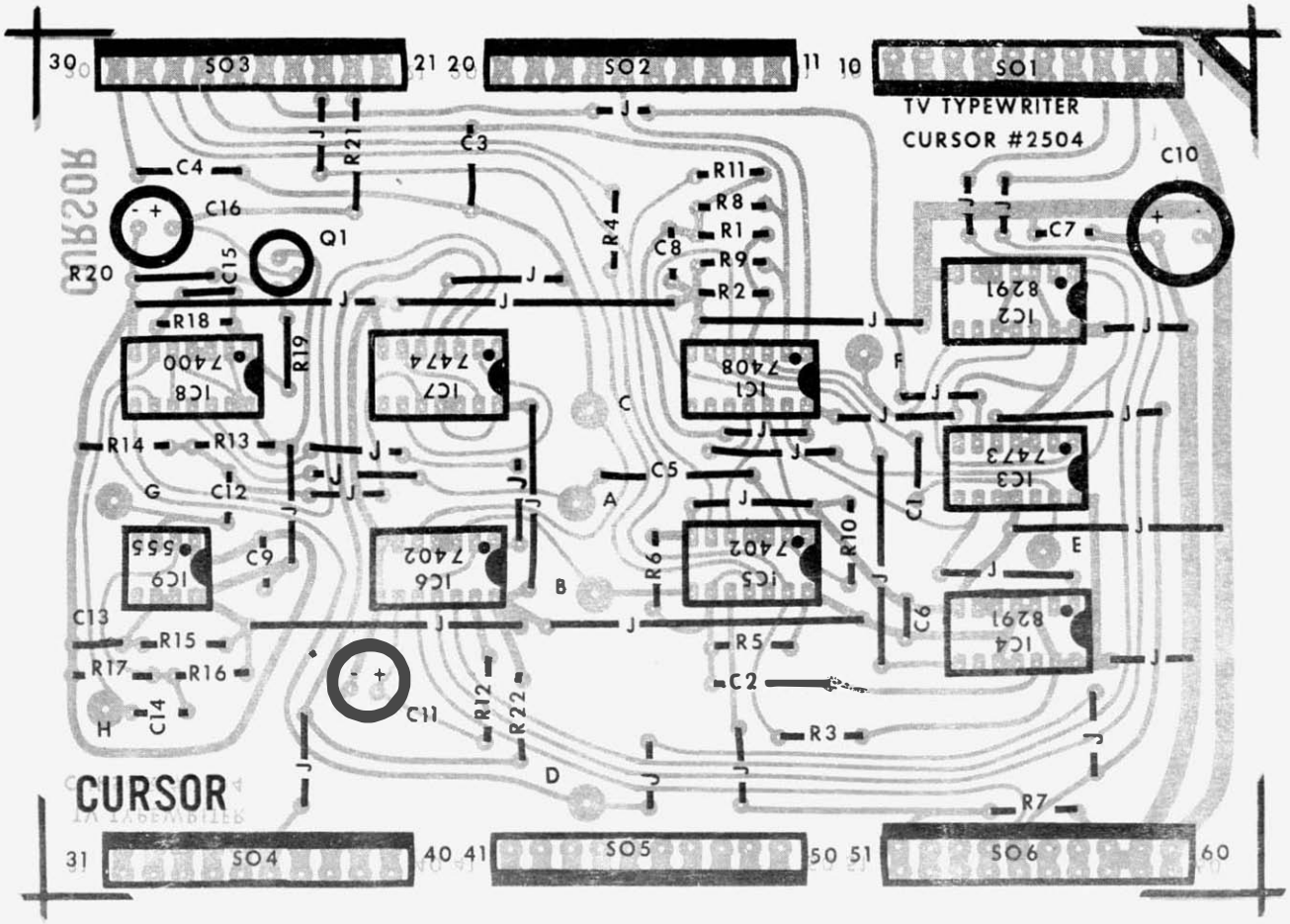


FIG. 17 -- PARTS PLACEMENT on cursor

Memory board parts list, page 6; D2, D3, D7 should be 1N4001.

The power transformer is available from Signal Transformer Company, One Junius Street, Brooklyn, N.Y. 11212, at $6.00 plus postage.

Molex sockets are available from Force Electronics, 343 Hindry Avenue, Inglewood, Calif. 90301. The sockets (09-52-3103) are 34¢ each. The matching pins (09-64-1101) are 39¢ per 10-pin assembly. Minimum order $10.00 plus 50¢ postage.

Here are some additional comments from Don Lancaster which may help answer other reader questions:

A color TV set has a video bandwidth of only 2.5 MHz; a black and white set has slightly more. This limits the number of characters per line to 32 or possibly 40, if an unmodified stock TV is to be used, particularly an economy one. Commercial terminal systems of 72 or 80 characters per line use special video systems with bandwidths of almost 10 MHz.

More memory can be added, but since the memory is the most expensive part, it very much ups the cost. Considering the limitations on video bandwidth and overscan on an unmodified TV, it would be difficult to do more than 512 characters per page. Of course, if you want to modify the TV, you can get denser displays.

If you want an all-the-time fixed memory, use a read-only-memory or a data selector instead of the shift registers used. There is no memory device I know of that is cheap enough to use with the TV typewriter, can be written into very rapidly and simply, and still keeps its information when power is disconnected. Mag core comes close, but is complex and hard to use on a small system. So do erasable ROM's, but it takes a while to program them. Next year, we'll probably see better devices; right now, I don't know of any. Most terminal applications don't need memory through power-down times anyway, and those that do can run on standby power.

At least 1/3 and preferably 1/2 the scan in each direction must be saved for retrace and blanking, particularly on economy TV's.

To obtain full interlace (the only time you either need or want full interlace is when you must superimpose your message on top of an existing, uncontrollable program source), you get horizontal and vertical signals from the system you are going to superimpose the characters on. These must be separate and not combined as EIA sync. They also must be TTL compatible. You compare the two horizontal outputs with a phase detector such as the Motorola MC4044 and derive an error signal to correct the 4.56 MHz oscillator phase lock loop style. The crystal is removed and replaced with a capacitor and the voltage control input is driven by the error output of the phase detector. Cost of this modification is under $10, but custom engineering is involved for each application.

Baudot and EBDIC and SELECTRIC codes are generated on the keyboard simply by redefining the key matrix, and possibly adding a flip flop or two. At the TV TYPEWRITER end, you have to add a read-only-memory such as the Harris PROM 0512 and another flip flop to convert to ASCII, or you can sometimes use commercial code converters.

We are trying to work up add-on's, but I am swamped with work right now, and they won't be immediately available. Custom engineering at this time, even at our incredibly exorbitant rates, simply isn't available. My thanks to the incredible number of readers responding to this project.

Here's a few corrections to the TV Typewriter supplement:

1. Diode D6 is backwards on the power supply overlay, and the negative supply diodes are shown backwards on the schematic.

2. On figure 3 schematic, NCLR pin 25 should also go to keyboard input B and the diodes D10-14. The connection between diodes D10-14 and "C" should be deleted. The PC board is correct.

3. Callouts are missing on the keyboard edge connector. "A" is nearest the RF twinlead; "L" is nearest J1.

4. Delete R11 and R12 from the mainframe figure 3 schematic. Add R5, R6 to the mainframe parts list, 1K, 1/4 watt carbon.

5. There are several printing problems on the supplement overlays. The overlays on the kit PC boards are correct and complete.

6. On the improved ASCII encoder schematic, IC2 should be 7402. TP tie points go to pins 4 and 5 of IC4.

7. Table V. For normal use, the switch should be left in the FULL position.

8. Timing E, cursor TPH should be 10 milliseconds, not microseconds.

9. An additional 0.05 ufd disc capacitor with minimum lead lengths might be needed across the TOP of cursor IC1 (7408) fron pin 7 to 14. Counter IC substitutions in the cursor might require slight shifts in pulse widths and positions. You can tell by a careful study of test point F on the cursor board. In the SUBTRACT position, one extra dot should appear before the 512 timing pulses every keypressed. in the ADD position, one short line should eliminate two of the normal 512 timing pulses.

10. An inverter formed from pins 11 and 12 of IC8, cursor board must be placed between IC6 pin 1 and "A" on the cursor board. This is shown correctly on the foil pattern (figure 16) but shoudl be added to the schematic of figure 8.

11. The dot to the left of C14 on figure 8 cursor should be a no connection. Once again, the foil is correct.

12. Connector stack pins 15 and 16 are correct as shown on the foil patterns. ø1 and ø2 notation only are apparently backwards in figure 7 and Table III.

13. On the main timing chain schematic, figure six, the LEFT end of C5 should go to R3. The RIGHT end of C6 should go to R2. The foil pattern is correct.

14. In figure 3, mainframe schematic, CURSOR OFF-ON should be S7, not S5.

In general, so far, we have found no errors on the foil patterns. Unless things change with more corrections, always assume the foil pattern and the printed overlay (with the exception of power supply diode D6 overlay) are correct.

Errors are almost inevitable on a project this complicated. My thanks to the readers who have sent in the corrections. Please keep sending them in so we can keep others up to date.