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

mainframe	timing	cursor	memory A	memory B	Pin	mainframe	timing	cursor	memory A	memory B	Pin
+	●	●	●	●	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	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 d1 1	0	+	0	●	0	45. Output load
0	+	0	●	●	16. Memory clock d1 2	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

*-- 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.