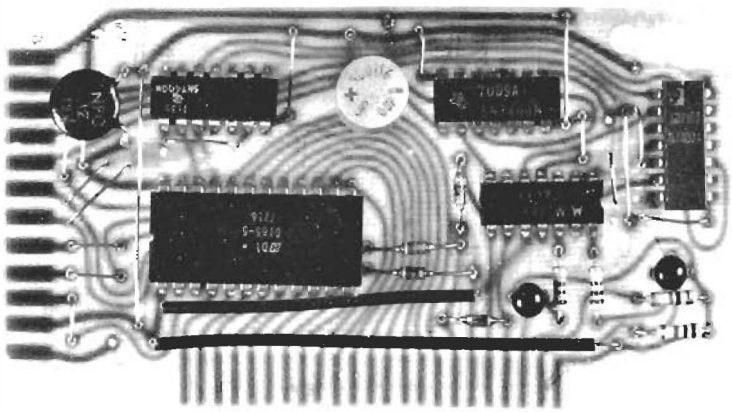


BUILD Improved ASCII Encoder

by DON LANCASTER

Simplified circuit lets you convert virtually any surplus keyboard into the proper code to talk to a computer or the Radio-Electronics TV Typewriter

THE ORIGINAL RADIO-ELECTRONICS ASCII keyboard encoder (**Radio-Electronics,** April, 1973) was designed to convert the single-make contacts of the Low Cost Keyboard (**Radio-Electronics,** February, 1973) into the proper ACSII computer code for talking to either a computer or the TV Typewriter (**Radio-Electronics,** September 73). Here's a greatly improved version of the same circuit.

It's much smaller, uses far fewer parts, has true TTL compatible outputs, provides an optional "there's two keys down" output and is designed to exactly fit one end of the currently popular "keypunch" type of surplus keyboards. Costs are about the same as the original but no kits are sold. It works with any keyboard that has one isolated pair of spst contacts that are normally open per key. The contacts may be mechanical, reed switches, contacts, or resistive elastomeric pads. For low impedance contacts, you need a single +5-volt supply. For resistive contacts up to 1000 ohms, you need a second +12-volt supply at low current.

In its present form, it will encode all the upper case alphabet, most punctuation, all the numbers, a spacebar, a carriage return, a shift key to shift from numbers to punctuation, and a control key for transparent or machine commands. Additional keys are easily added if needed. A slight modification of the pre-encoder matrix on the **Radio-Electronics** low-cost keyboard (February 1973) is needed to use the new unit.

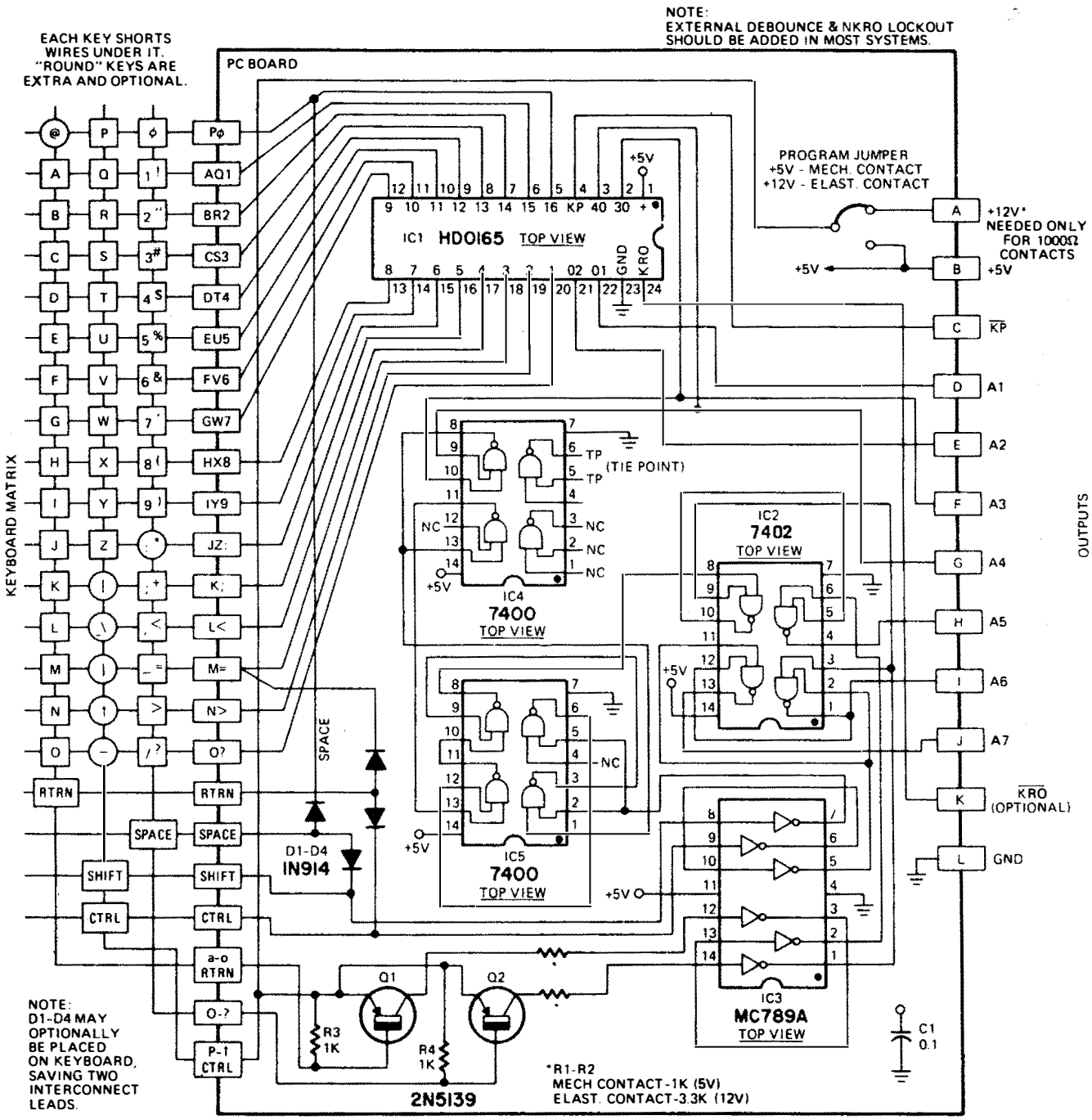
More on ASCII

The standard computer code in use today is called ASCII, short for American Standard Code for Information Interchange. The complete ASCII code is shown in Table I. While there are 128 possible entries to the code, we are only interested in the first six columns of these. Columns 0 and 1 are the transparent or machine commands. Things that don't end up in print or as part of a message, but instead return carriages, start and stop equipment, indicate beginnings and endings, and so on. Column 2 is the most popular punctuation, while column 3 is the numbers and more punctuation. In this encoder, we get from column 2 to column 3 by using a *shift* key. Thus, a capital "1" becomes a !, and so on. Finally Columns 4 and 5 are the alphabet and some lesser used punctuation.

We can get by with a basic assembly of 48 keys less a few if we're willing to do away with {,/,},↑ and ___. With a Control or CTRL key, we can shift any column 2 or 4

TABLE I
THE INDUSTRY STANDARD ASC11 COMPUTER CODE

b7 → b6 → b5 → / Bits b4 b3 b2 b1	Column Row	0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
		0	1	2	3	4	5	6	7
0 0 0 0	0	NUL	DLE	SP	0	@	P	`	p
0 0 0 1	1	SOH	DC1	!	1	A	Q	a	q
0 0 1 0	2	STX	DC2	"	2	B	R	b	r
0 0 1 1	3	ETX	DC3	#	3	C	S	c	s
0 1 0 0	4	EOT	DC4	$	4	D	T	d	t
0 1 0 1	5	ENQ	NAK	%	5	E	U	e	u
0 1 1 0	6	ACK	SYN	&	6	F	V	f	v
0 1 1 1	7	BEL	ETB	'	7	G	W	g	w
1 0 0 0	8	BS	CAN	(	8	H	X	h	x
1 0 0 1	9	HT	EM	)	9	I	Y	i	y
1 0 1 0	10	LF	SUB	*	:	J	Z	j	z
1 0 1 1	11	VT	ESC	+	;	K	[	k	{
1 1 0 0	12	FF	FS	,	<	L	\	l	/
1 1 0 1	13	CR	GS	—	=	M	]	m	}
1 1 1 0	14	SO	RS	.	>	N	^	n	~
1 1 1 1	15	SI	US	/	?	O	—	o	DEL



1PARTS LIST

C1—0.1-μF disc ceramic, Mount *flat*.
D1, D2, D3, D4—1N914 or equivalent silicon computer diode
IC1—HD0165 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 +5 supply; 3300 ohms for elastomeric high resistance contacts and +12 supply.
R3, R4—1000 ohms, ¼-watt carbon

MISC: PC Board, Solder; No.24 Soldereeze wire, 20 feet for keyboard wiring, sleeving, No.24 solid wire jumpers.

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

PC Board, etched and drilled: $5.75.

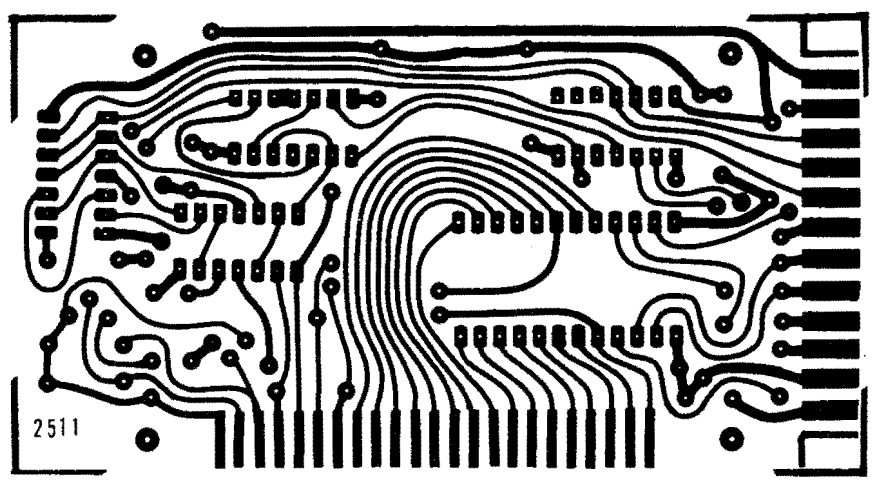
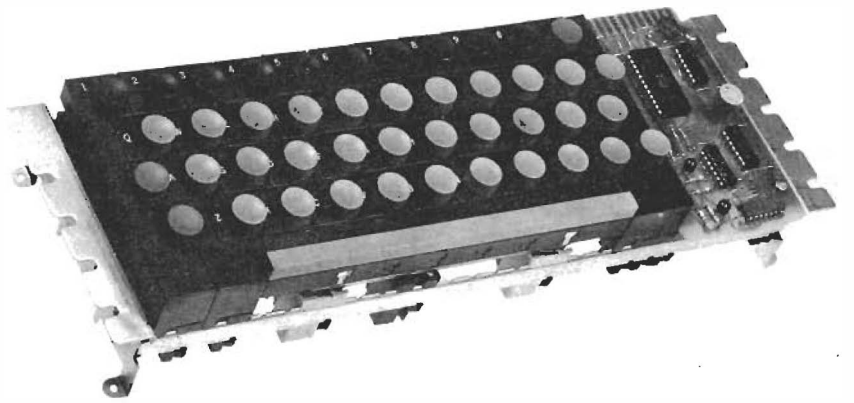


FIG. 1—ASCII ENCODER CIRCUIT (top) is easy to build. FULL SIZE FOIL PATTERN (above) is for the circuit board. PARTS LAYOUT (right) shows where to mount the components on the circuit board.

code into a column 0 code, and any column 3 or 5 code into a column 1 code. Thus, we need no new keys for the control commands, unless we are really going to use that command often. CARRIAGE RETURN is often used, so, it's handy to have a special key that *simultaneously* gives us a CONTROL and a M command. Similarly, we can get a spacebar by simultaneously giving a SHIFT and a 0 command. Other special functions (DELETE, ESCAPE, ALT MODE, etc. . . .) are easily added in the same way.

To decide when a code is sent, a keypressed command is given when a key is pressed, telling things on the other end that something new is happening. We *do NOT deliver* a keypressed command for the shift or control key, for they are always used in conjunction with another key. And, in our circuit, we get a free "there's two keys pressed!" output that can be used to tell whatever is on the other end that the typist is running too fast or just made a mistake and please ignore what just arrived. One final, and slightly messy detail involves the > = < and ? keys. Normally, we like to type commas, dashes, periods, and slashes *without* shifting, and save the question, equals, greater than, and less than for *shifted* commands. This is clearly backwards from the standard code. So if we are going to go along with the standard code (often we are forced to because of the keytops on the keyboard we're going to use), we have to arrange the shift key so that it operates *backwards* on these four keys. All this takes are two 21¢ IC's, but this is a complex and painful little detail to resolve.

The output of the code consists of seven bits in *parallel*, or all-at-once form. An eighth *parity* bit can optionally be added for error detection, or the seventh bit can optionally be dropped to get the 10-bit code that has only alphanumerics to run a character generator. Should we want to talk to a computer or a phone line, we have to convert this code to a *serial* form, easily done with either the circuit shown in the original article or with a new MOS terminal transmitter/receiver chip. Depending on the type of keyboard and the debouncing in the rest of the system, we may have to add a contact conditioning and debouncing system as well.



TYPICAL KEYBOARD WITH ENCODER. The small encoder board is mounted at the right end of the keyboard.

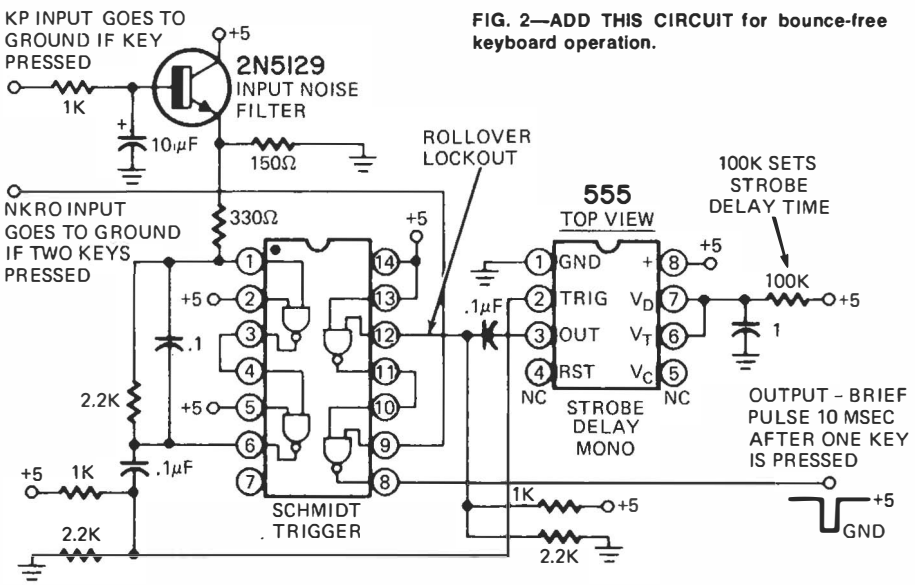
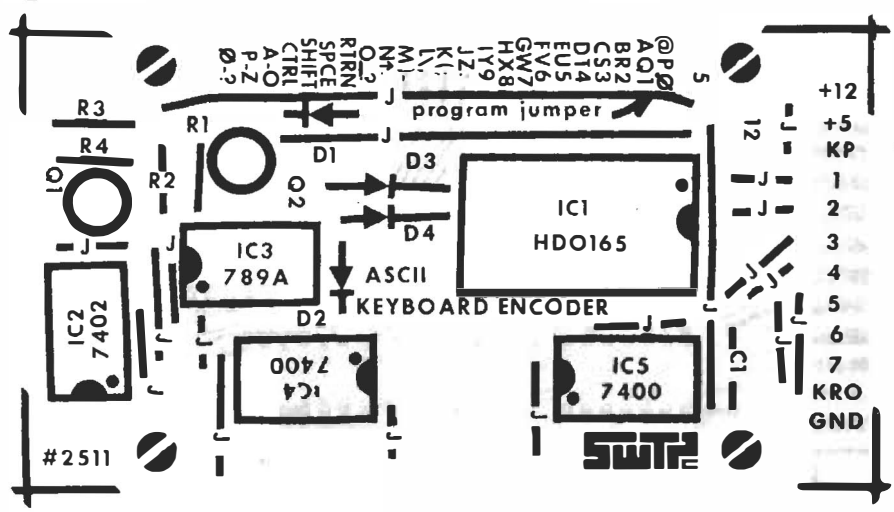


FIG. 2—ADD THIS CIRCUIT for bounce-free keyboard operation.



About the new circuit

The new circuit is shown in Fig. 1. Except for IC1 (presently around $7.50), all the remaining parts are nickel and dime stuff, and there are only 19 components in all. Just like the code of Table I, we can split the problem into two parts, for the lower four bits couldn't care less what the upper three are doing, so long as everything ends up right. Thus a lower four bits 1101 code could be a carriage return, a group seperator (a very rare machine command), a dash or minus, an equals, a M, or a large unbracket. IC1 singlehandedly takes care of the lower four bits for us. It has sixteen input lines and four output lines. If you make any *one* (only one!) input line positive, it gives the binary equivalent to that code. Thus the third line generates a 0011, the eighth line a 1000, and so on.

The inputs are RTL style and simply need an impedance path to +5 or +12 to serve as an input command. Whatever *else* the input current flows through on the way to set up the upper three bits is of no concern to IC1, so long as the current gets there when it is needed. IC1 also generates a keypressed output that's high if all the inputs are low and goes low if any key is pressed. It also produces an optional output that goes low if two keys or more are simultaneously pressed. This is called a NKRO output, short for N-key-rollover.

It only takes about +3.5 volts to turn on an IC1 input. Since the input is current operated, we can either get our current from a low impedance (mechanical or reed) contact and a +5 supply, or from a higher impedance (elastomeric or foam) contact and a +12 supply. Around two milliamperes are needed, but it can handle much more than that safely. Thus, we can use virtually any kind of keyboard contact simply by picking one optional low current supply voltage.

So much for the lower four bits. The upper three bits are generated by responding to *what* the IC1 input current is routed through on the way down from the positive supply. If it goes through nothing, we set up P-Z. If it goes through Q1, we set up A-O, and if it goes through Q2, we set up zero through 9 and the related punctuation. The

(continued on page 92)

IMPROVED ASCII ENCODER

(continued from page 61)

sensing is translated down to ground by IC3 and acted on by IC2 to get the proper three bit code. For instance, with no current through Q1 or Q2, the output code 101-XXXX is sent, corresponding to a character P through Z. Current through either other transistor sets up the numbers or lower alphabet code. And this gives us our basic 48 key encoding scheme.

Now, we add some refinements. The shift key acts on IC2 to change from punctuation to numbers or backwards. If a code from 0000 through 1011 is sent, this is recognized by IC5 as a number, a colon, or a semicolon. IC5 then tells IC4 to leave the shift command the way it was, and a *unshifted* key gets you a number, and a *shifted* key gets you the equivalent punctuation. On the other hand, if codes 1100 through 1111 are sent, IC5 detects this, and through the exclusive-or gate in IC4, compliments the shift line. Now, the shift key works *backwards*, putting the four problem punctuation keys back the way they belong. This is admittedly a bootstraps operation, but the time for the output to tell the input to shift is only a few nanoseconds, an utterly negligible time compared to most contact condiing and compliment-the-shift-if-12-or-greater circuit *only* affects the number keys. Any alphabet or punctuation key in columns IV or V are *not* affected by this circuit since Q2 is not sensing a "numbers" key.

Finally, the control key forces outputs 6 and 7 to ground regardless of anything else, automatically shifting from a alphanumeric command to a machine command. The spacebar is diode encoded to simultaneously provide a shift and a 0, or a "capital Ø", while the carriage return key uses two diodes to simultaneously give you a "control" and a "M" command. Other special keys are easily added, often by using two more diodes per key. If you need a DEL or delete key, the simplest way to do this is to break output line 6 with a SPDT key and route it to +5. The normal output of the encoder is 101-1111 when no key is in use. This changes it to 111-1111 or DEL. If you must have the lower case alphabet (make sure the other end of the system can handle or recognize it), you can leave the DEL key down to generate it.

Building it

A PC board is shown along with a parts list and notes on kit availability. We cannot tell you, now, what keyboards are available. Check the ads in the back of this issue for a guide. Before construction, you have to decide whether you are going to use mechanical contacts or resistive ones and place the program jumper accordingly, +5 for mechanical contacts or +12 for elastomeric ones. A 12-volt battery may also be used, as no input current is drawn unless a key is pressed, or you can tack a 6-volt battery on top of the +5-volt line for 11 volts. Resistors R1 and R2 also have to be changed, 1K for the +5-volt supply, or 3.3K for the +12-volt supply. If the encoder is to be used on a keypunch style keyboard, it easily mounts on one end of the unit by using four spacer blocks. A slight amount of filing may be

(continued on page 94)

(continued from page 92)

necessary on these blocks, depending on your soldering on the circuit board to get things to lie flat. The bypass capacitor should be mounted *component* side, but with slightly long leads and bent flat. Otherwise interference between C1 and the metal case or rails holding the keyboard in your system may result.

There are 23 input leads. These are directly wired to the keys following the wiring matrix of figure one. (If you're using the **Radio-Electronics** Low-Cost Keyboard, rewire the jumper matrix to suit.) If you have to run through a connector or otherwise want to minimize the interconnect leads, the diodes D1 through D4 can be placed on the keyboard end, reducing you to 21 connections. Normally you mount the encoder integrally with your keyboard and this doesn't matter.

Note that whatever keyboard you use, the keytops must be ASCII ones. This means that the "capital 2" has to be a #, the capital colon a *, the capital semicolon a dash or minus, and so on. Otherwise, you have to remark the keyboards.

Using a keypunch keyboard

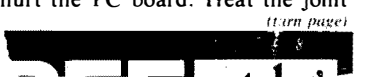
These are easily recognized by their bright blue keys and are now available from several surplus sources. More can possibly be expected since keypunch equipment is more or less headed to obsolescence. An unmodified one of these has the keys and keypairings all wrong, since they originally

were EBCDIC coded, a non-standard and older code. Fortunately the callouts go where you can't reach them with a finger, and attractive stick-ons are easy to do, (see parts lists) as are instant transfer letters, or white ink fro the stationery store.

Various solvents (lacquer thinner, etc.) easily remove the old ink, or it may be carefully scraped off with an Xacto knife or a razor blade. Don't soak the keys in solvent as it may attack the plastic. Keyboards available from some sources are completely rebuilt for ASCII use. If you are doing your own instead, rearrange the parts so you have a rectangular block of keys with the black keys on top and the blue ones in the middle. Again, if you are rebuilding your own, you probably will end up short by at least two or three blue keys. These may be repainted as needed. ASCII modified keyboards come with everything the right color. Extra mounting blocks on one end support the encoder. The encoder may be mounted above or below the rails, although the lower position is much easier to wire. Output may be directly soldered, or connected with a standard 12-pin PC edge connector.

One way to simplify the wiring with this pe of keyboard is to use *solderable* magnet re—stuff that you can solder right through insulation. Beldsol and Soldereeze are types. This way, you loop the wire from inal to terminal and then solder it in e without stripping. Stripping or at least nning is still recommended at the PC d end. It takes extra heat to solder gh the insulation, but don't use so big or gun that you melt the key assembhurt the PC board. Heat the joint

(turn page)



(continued from page 95)

first. After the insulation smokes, add a *minimum* amount of solder.

Testing the encoder

The obvious test is an acid test—connect it to the TV Typewriter or a terminal and see if it sends the right letters. A way that ties up less fancy equipment is to use a batch of IC driveable test lamps or LED's to *simultaneously* monitor *all* the outputs.

If some stickiness or reluctance is experienced with the spacebar and carriage return keys, raise the supply voltage slightly or replace D1 and D2 (the diodes that go to IC1) with metal barrier low-threshold diodes such as a MBD101 or something else with a 0.3 volt forward drop or less at 6-mA current. Of several IC1's tested, operation was satisfactory down to 4.7 volts. *Maximum* permissible voltage applied to the +5 terminal is 6.8 volts. A tightly regulated 5.1 or 5.2-volt supply should work with practically all units.

As with any keyboard system, some form of debouncing and noise elimination is *essential*. Normally, as in the TV typewriter, this is provided internally. If not, one possible circuit is shown in Fig. 2 that handles most any keyboard. An optional parity generator and parallel to serial converter were shown in the original article. These may be used as add-ons, or newer MOS asynchronous receiver-transmitter integrated circuits can be used for the same task. If enough readers need this sort of thing, we'll work up a project on it. **R-E**