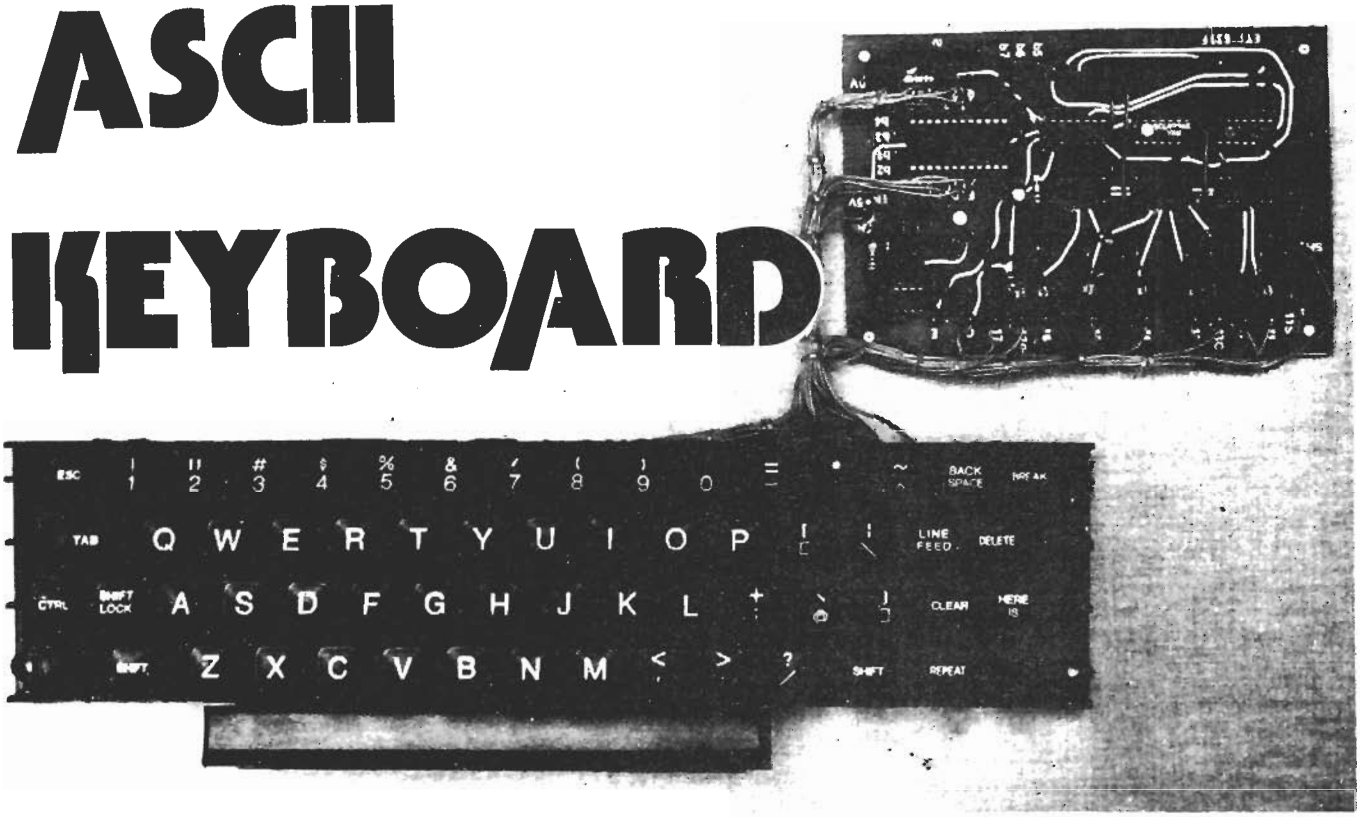
ASCII KEYBOARD



THIS PROJECT, NUMBER 631, FORMS THE FIRST PART OF OUR MPU HOME SYSTEM, THE ENCODER BOARD PROVIDES FULL ASCII OUTPUT, AND CAN BE USED WITH MOST COMMERCIAL KEYBOARD UNITS.

TO COMMUNICATE WITH A computer you need some sort of input device and some sort of output reader. The input unit can be a series of switches on which you set up the required code and press a button to enter each character. While this is economical in parts it is not economical in time.

This encoder project is designed to allow very easy access to the computer whilst being reasonably economical. It is very flexible and allows for almost any keyboard to be used. Control functions can be activated by a single key if desired and lower case letters can be eliminated at the flick of a switch.

The output from the keyboard is in the form of a parallel bus and the data has to be serialised to provide a universal input which will then communicate with any computer designed to work with a teletype.

DESIGN FEATURES

When we first looked at a keyboard encoder we intended to use a single chip device to simplify design. However, looking at the devices available and their limitations (and cost and availability) it was decided to compromise and use the HD0165 keyboard encoder. This IC has been available for many years and we use it to decode the first 4 lines. For the other three lines we decided to use discrete components. The eighth line is not used at this stage (it is used for a parity check after serialisation).

Initially the use of a 16 x 3 matrix was contemplated. Then we would use the shift and control keys to get the other outputs. However, not all keys with the same three-line code (b5, 6, 7) are upper case (or lower case). On our keyboard 0 1 ... : ; are lower case, and = ? are upper case; yet all have an output code 3. The same applies to other rows and the matrix has thus expanded to 16 x 7. To get the control functions a control and the function key have to be pressed simultaneously, which is inconvenient for commonly-used functions (such as space or line feed).

Consequently an additional three lines are used and this allows any of the control functions to be activated by a single key.

Most VDUs or microcomputer operating systems cannot handle lower-case letters and therefore outputs are provided which can be linked to ensure that a shift command is given automatically when any key from A-Z is pressed.

When connecting to the keyboard we had to decide how to wire the contacts. The easiest and neatest way is to use a double-sided pc board with plated-through holes. Using such a board it is hard to solder the other side when it is against the keyboard!

The alternative, and the approach we chose, is to link the underside of the keyboard using "solderable" enamelled wire and normal hookup wire to the control card. This takes a little time to wire but is much cheaper and is universal. Although we used a

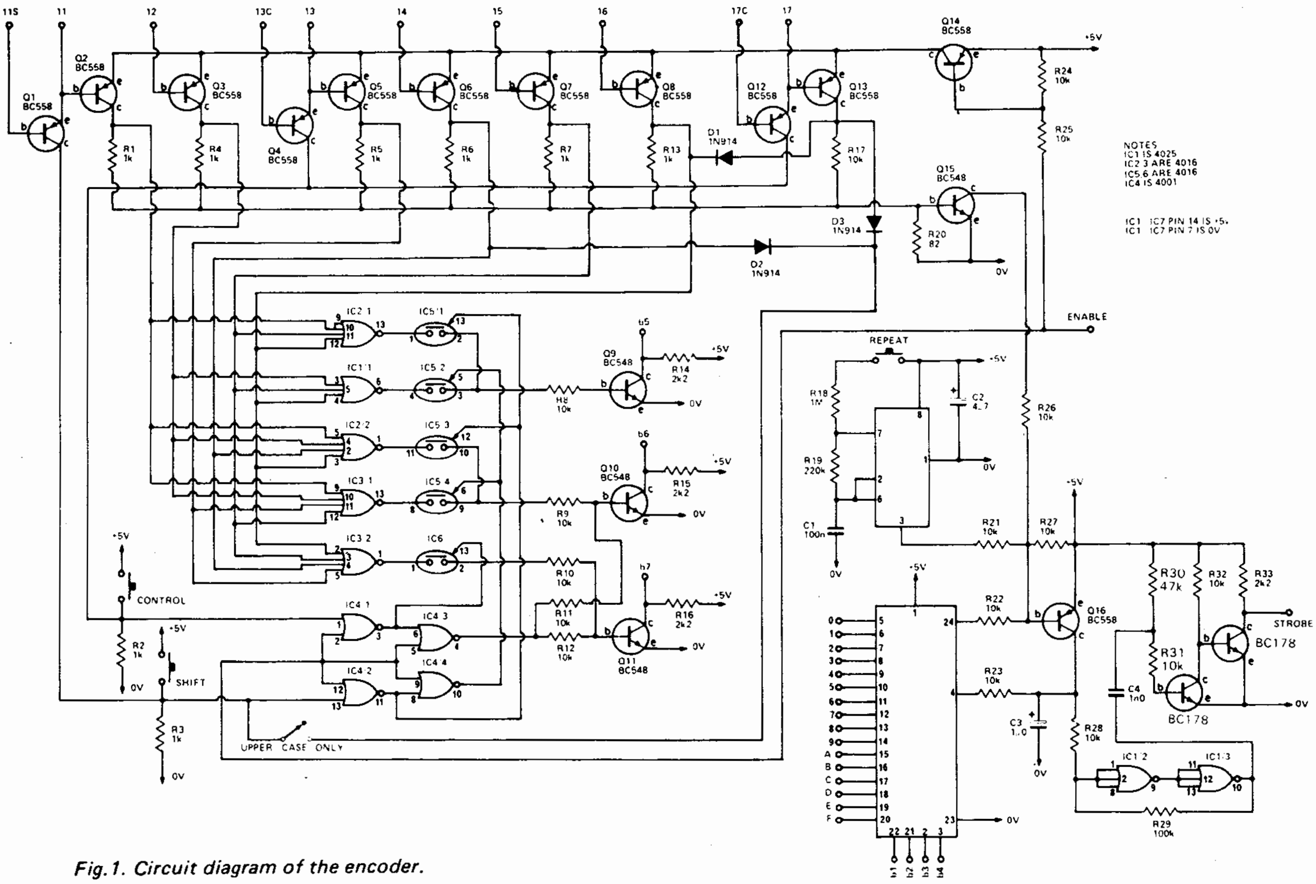


Fig. 1. Circuit diagram of the encoder.

How it works

The Harris HDO165 IC is a 16-line keyboard encoder; if any one of its 16 inputs is taken high (+5V) an output code appears on the four output lines. At the same time another output (pin 4) goes low and another output (pin 24) goes low to indicate this.

In this project we use this IC to generate the least significant four bits (b1, b2, b3, and b4) of the seven bits we need to represent the complete character.

To decode the other three bits we used discrete transistors and CMOS gates. Each key joins one of the inputs of the HDO165 to one of the points 11-17. If the enable line is low (i.e. 0V) Q14 will be hard on and we will have 5V (less a little) on the emitter of Q2,3,5,6,7,8 and 13. The input of the HDO165 appears as a resistor of about 500-600 ohms, to 0V. Therefore connecting (say) point 14 to point 3, we turn on Q6 giving +5V at its collector and also the HDO165 gives an output corresponding to three (0011).

The high output from Q6 gives a high on the inputs of IC 2/2 and IC3/2 causing the outputs of these gates to be low. The other gates, IC1/1, IC2/1 and IC3/1 have high outputs. If the control or shift key is not pressed, we have a '0' at the input of IC4/1 and IC4/2 giving a high output from these gates and hence a low output from IC4/3 and IC4/4. This enables IC5/1, IC5/3 and IC6. These ICs are simply electronic switches with a resistance of either 300 ohms (on) or infinity (off).

Therefore Q9 will be on as IC2/1 is high, Q10 will be off as IC2/2 is low and Q12 will also be off as IC3/2 is low. This gives a total output of 110 0011 which represents 63 (hex) or lower case c.

We will leave you to work out the other combinations. If the shift key is pressed, IC5/2, IC5/4 and IC6 are enabled selecting a different code (upper case C is 43 hex) and if the control key is pressed, Q10 and 11 are turned on by IC4/3 and Q9 is controlled by IC1/1 and IC2/1 ('control C' is 03 (hex), representing ETX).

When a key is pressed the output (pin 4) of the HDO165 goes low and C3 is discharged via R23. After about 10ms. the gates IC1/2, 3, which are connected as a schmitt trigger, operate and the output (IC1/3) goes low. This is coupled via C4. Q17/18 act as a monostable giving a negative-going pulse of about 200µs wide. When one key is pressed about 0.4V is developed across R20, not quite enough to turn on Q15. If a second key is pressed in a different row, the additional current in R20 will forward-bias Q15 which will then turn on Q16. This holds C3 charged, independent of the HDO165. If two keys are pressed in the same row on output (pin 24), the HDO165 detects this and goes low and Q16 is again turned on disabling the strobe pulse.

If the repeat button is pressed IC7 oscillates at about 10Hz and the pulsing alternately turns Q16 on and off generating strobe pulses at about 10 per sec.

The output of Q6 (A-O) and Q17 (P-Z) are diode ORed and if the 'upper case only' link or switch is closed it automatically gives a shift command. For the control functions additional inputs are used in Q1, Q4 and Q12. If the input to one of these transistors is connected to one of the HDO165 inputs it still turns on the transistor associated with it and also lifts either the control or shift inputs as required.

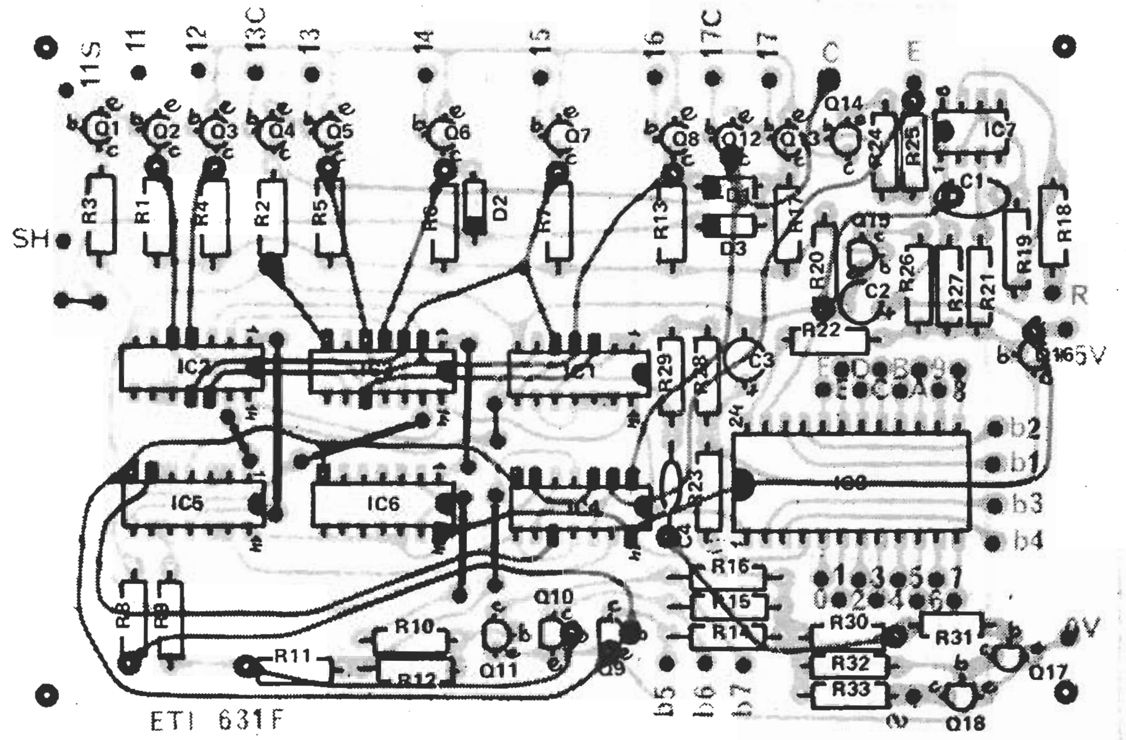


Fig.2. Component overlay of the encoder.

double-sided board for the control logic we don't require plated-through holes, as both sides can be easily soldered.

CONSTRUCTION

Assemble the PCB board with the aid of the overlay in Fig 2. When soldering the components use a small iron and make sure *all* connections on the component side are soldered as well as those on the copper side. The links on the component side must be insulated where they cross copper tracks, to prevent shorting.

Because you have to solder on both sides of the PCB you cannot use ICs sockets (unless they are wire-wrap types). The exception here is the HD0165 where all connections are on the copper side. Note also that the HD0165 is not CMOS or MOS and requires no special handling.

To make wiring easier mark the keys on the underside of the keyboard, to indicate what functions they represent. Now using "solderable" enamelled wire join the points as given in Table 1. The connection from the control board is also given and this should be made

CONTROL CARD	KEYBOARD SEQUENCE
0	0 P @ SP
1	1 Q A
2	2 R B
3	3 S C
4	4 T D
5	5 E U
6	6 F V
7	7 G W
8	8 H X BS
9	9 I Y TAB
A	: J Z LF
B	[; K ESC
C	\ L ,
D	— M] CR
E	Λ . N
F	DEL 0 /

CONTROL CARD	KEYBOARD SEQUENCE
11	: ; 0 9 8 7 6 5 4 3 2 1
12	— / . ,
13	@
14	L O I K J M N H B G F C D E A
15	Λ \ [ ]
16	DEL
17	P U Y T R V X Z S W Q
11S	SP
13C	BS TAB LF CR
17C	ESC

TABLE 1 ETI 631

How to wire up the keyboard

Parts List

Resistors all ½ W 5%	
R1-R7	1 k
R8-R12	10 k
R13	1 k
R14-R16	2k2
R17	10 k
R18	1 M
R19	220 k
R20	82R
R21-R28	10 k
R29	100 k
R30	220 k
R31	100 k
R32	10 k
R33	2k2
Capacitors	
C1	100 n polyester
C2	4µ7 25 V
C3	1 µ 25 V
C4	1n0 polyester
Semiconductors	
D1-D3	1N914
Q1-Q8	BC558 or BC108
Q9-Q11	BC548 or BC178
Q12-Q14	BC558 or BC108
Q15	BC548 or BC178
Q16	BC558 or BC108
Q17, 18	BC178
Integrated Circuits	
IC1	4025 (CMOS)
IC2, 3	4002 (CMOS)
IC4	4001 (CMOS)
IC5, 6	4016 (CMOS)
IC7	NE555
IC8	HD0165

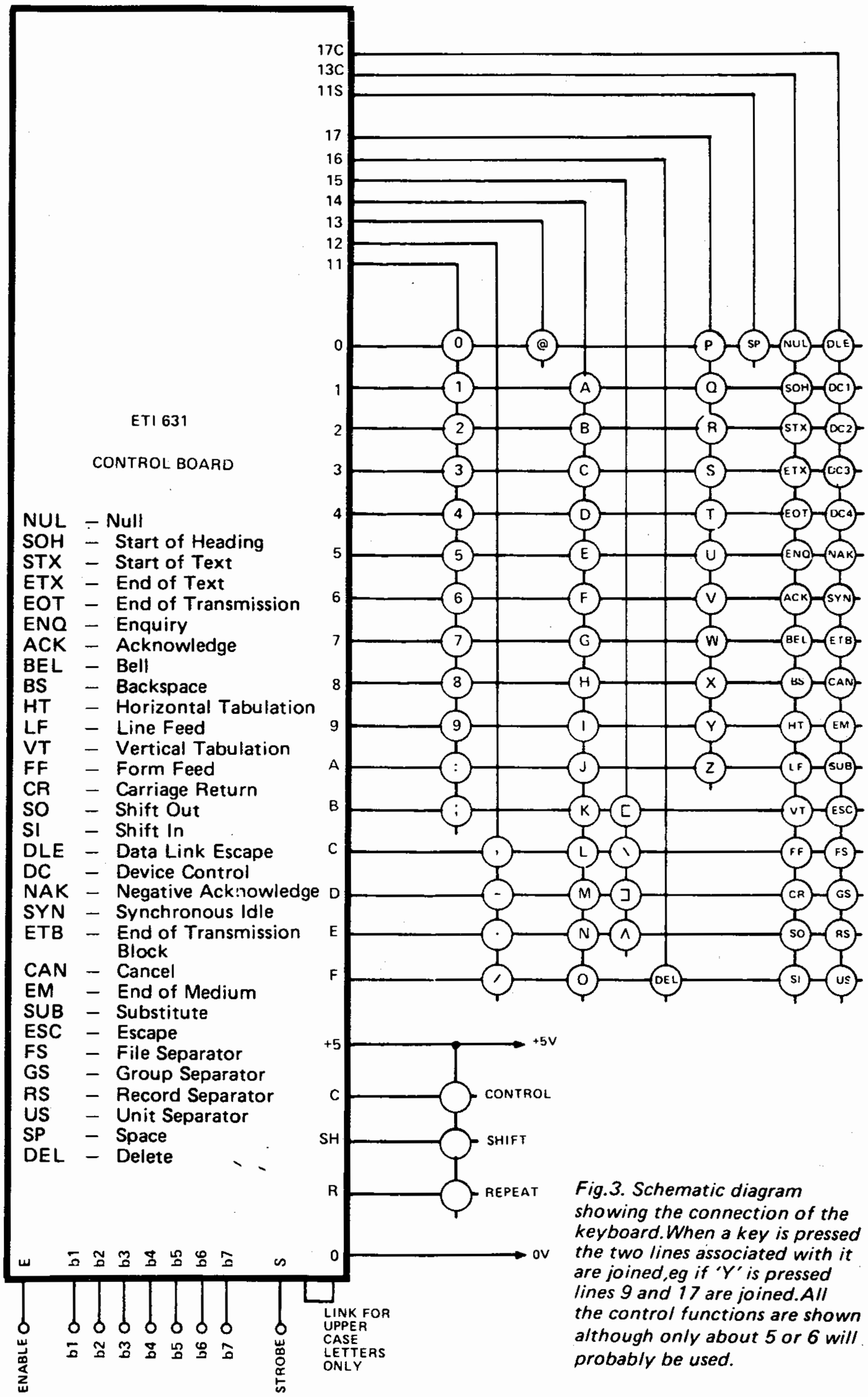


Fig.3. Schematic diagram showing the connection of the keyboard. When a key is pressed the two lines associated with it are joined, eg if 'Y' is pressed lines 9 and 17 are joined. All the control functions are shown although only about 5 or 6 will probably be used.

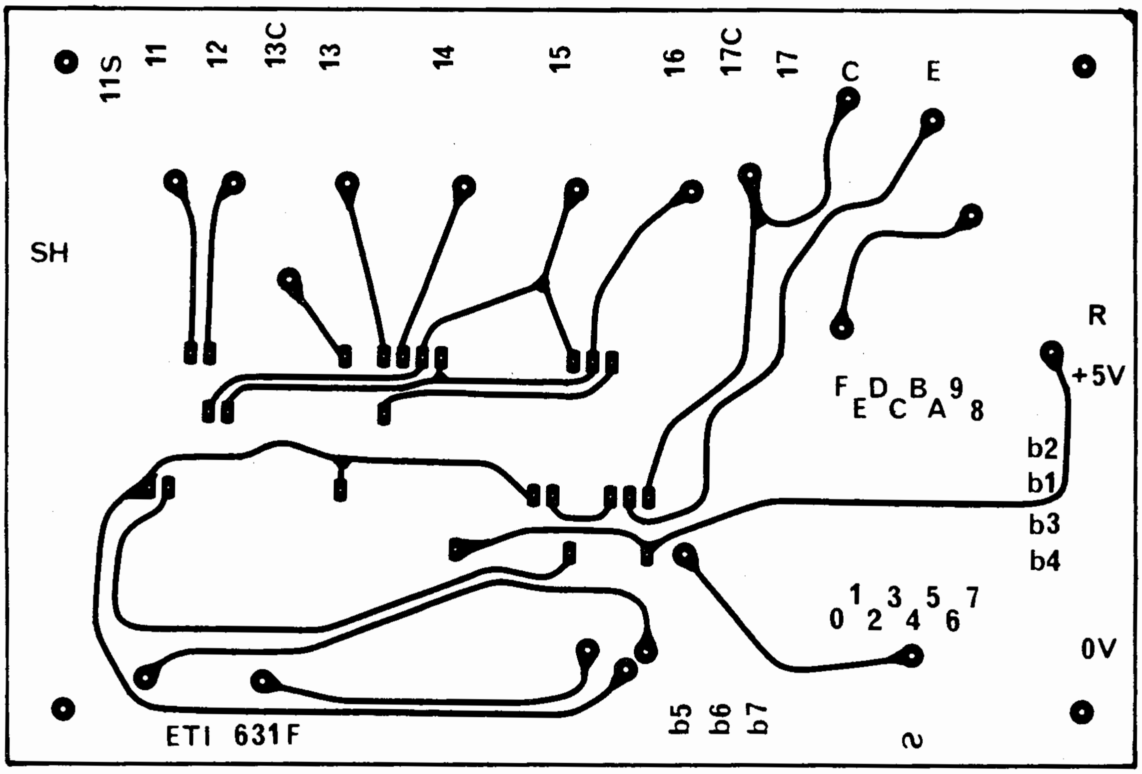
using normal hookup wire. The control functions can be wired between the points given either by taking two wires back to the control board or finding the same wire, if previously used, on the keyboard and linking across.

We have not described a housing for the unit as it will probably be mounted along with the VDU and UART (possibly under a TV set).

However, the control card can mount under the keyboard by spacing it up slightly. It may be necessary to have a piece of metal (Bacofoil, etc) under the keyboard/control card, connected to 0V. (To prevent 50Hz pickup into the wiring to the keyboard.) The effect of this is unwanted outputs from the strobe or non-operation of the strobe output.

To supply the unit 5V at 50mA is needed. To enable the keyboard a low (0V) is needed on that input. The data output are positive logic (ie, "1" is +5V) and the strobe output is active low.

Connecting the keyboard to a hex display gives an easy check that all wiring is correct. The list in Table 1 gives the character the access, the ASCII code, and the hex code. Alternatively 7 LEDs can be connected (cathode to 0V) across the outputs



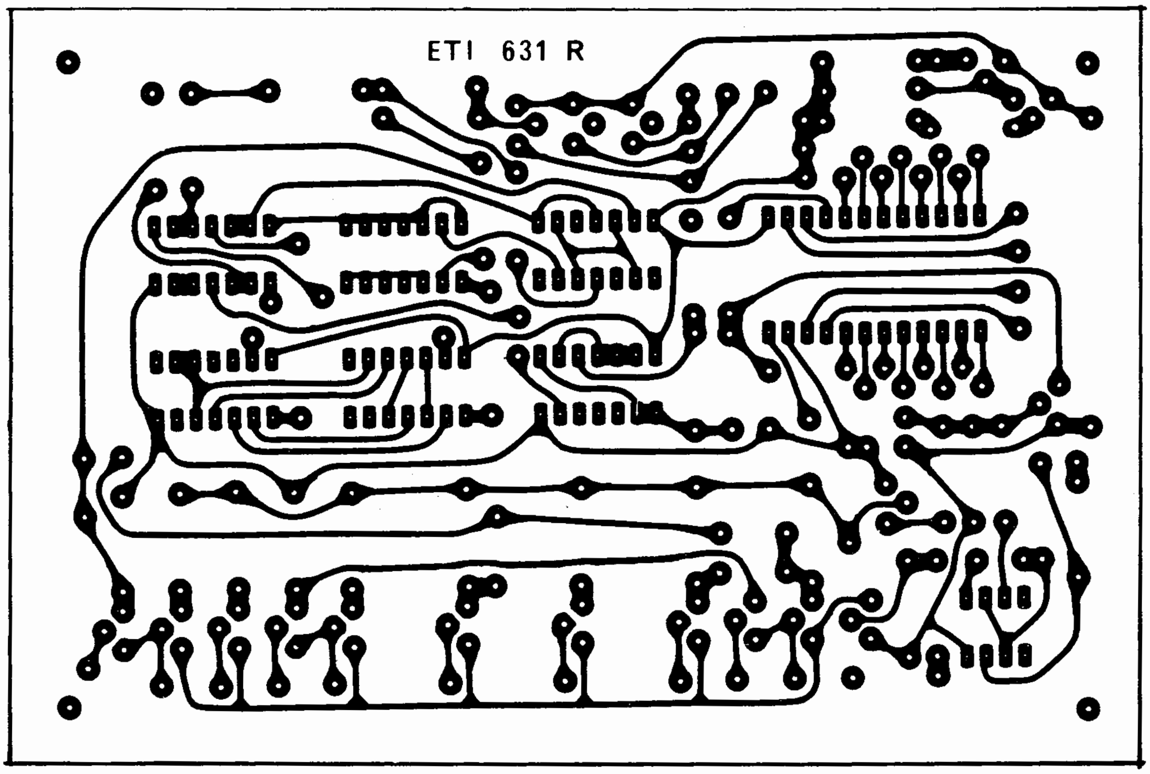


Fig.4. Printed circuit layout (both sides) Full size 150 x 100 mm.

TABLE 2

FUNCTION	ACCESS	b7	b6	b5	b4	b3	b2	b1	HEX CODE
NUL	CTRL @	0	0	0	0	0	0	0	0 0
SOH	CTRL A	0	0	0	0	0	0	1	0 1
STX	CTRL B	0	0	0	0	0	1	0	0 2
ETX	CTRL C	0	0	0	0	0	1	1	0 3
EOT	CTRL D	0	0	0	0	1	0	0	0 4
ENQ	CTRL E	0	0	0	0	1	0	1	0 5
ACK	CTRL F	0	0	0	0	1	1	0	0 6
BEL	CTRL G	0	0	0	0	1	1	1	0 7
BS	CTRL H	0	0	0	1	0	0	0	0 8
HT	CTRL I	0	0	0	1	0	0	1	0 9
LF	CTRL J	0	0	0	1	0	1	0	0 A
VT	CTRL K	0	0	0	1	0	1	1	0 B
FF	CTRL L	0	0	0	1	1	0	0	0 C
CR	CTRL M	0	0	0	1	1	0	1	0 D
SO	CTRL N	0	0	0	1	1	1	0	0 E
SI	CTRL O	0	0	0	1	1	1	1	0 F
DLE	CTRL P	0	0	1	0	0	0	0	1 0
DC1	CTRL Q	0	0	1	0	0	0	1	1 1
DC2	CTRL R	0	0	1	0	0	1	0	1 2
DC3	CTRL S	0	0	1	0	0	1	1	1 3
DC4	CTRL T	0	0	1	0	1	0	0	1 4
NAK	CTRL U	0	0	1	0	1	0	1	1 5
SYN	CTRL V	0	0	1	0	1	1	0	1 6
ETB	CTRL W	0	0	1	0	1	1	1	1 7
CAN	CTRL X	0	0	1	1	0	0	0	1 8
EM	CTRL Y	0	0	1	1	0	0	1	1 9
SUB	CTRL Z	0	0	1	1	0	1	0	1 A
ESC	CTRL [	0	0	1	1	0	1	1	1 B
FS	CTRL \	0	0	1	1	1	0	0	1 C
GS	CTRL]	0	0	1	1	1	0	1	1 D
RS	CTRL ^	0	0	1	1	1	1	0	1 E
US	CTRL DEL	0	0	1	1	1	1	1	1 F
SP	SHIFT 0	0	1	0	0	0	0	0	2 0
!	SHIFT 1	0	1	0	0	0	0	1	2 1
"	SHIFT 2	0	1	0	0	0	1	0	2 2
#	SHIFT 3	0	1	0	0	0	1	1	2 3
$	SHIFT 4	0	1	0	0	1	0	0	2 4
%	SHIFT 5	0	1	0	0	1	0	1	2 5
&	SHIFT 6	0	1	0	0	1	1	0	2 6
'	SHIFT 7	0	1	0	0	1	1	1	2 7
(	SHIFT 8	0	1	0	1	0	0	0	2 8
)	SHIFT 9	0	1	0	1	0	0	1	2 9
*	SHIFT :	0	1	0	1	0	1	0	2 A
+	SHIFT ;	0	1	0	1	0	1	1	2 B
,	,	0	1	0	1	1	0	0	2 C
-	-	0	1	0	1	1	0	1	2 D
.	.	0	1	0	1	1	1	0	2 E
/	/	0	1	0	1	1	1	1	2 F
0	0	0	1	1	0	0	0	0	3 0
1	1	0	1	1	0	0	0	1	3 1
2	2	0	1	1	0	0	1	0	3 2
3	3	0	1	1	0	0	1	1	3 3
4	4	0	1	1	0	1	0	0	3 4
5	5	0	1	1	0	1	0	1	3 5
6	6	0	1	1	0	1	1	0	3 6
7	7	0	1	1	0	1	1	1	3 7
8	8	0	1	1	1	0	0	0	3 8
9	9	0	1	1	1	0	0	1	3 9
:	:	0	1	1	1	0	1	0	3 A
;	;	0	1	1	1	0	1	1	3 B
<	SHIFT ,	0	1	1	1	1	0	0	3 C
=	SHIFT -	0	1	1	1	1	0	1	3 D
>	SHIFT .	0	1	1	1	1	1	0	3 E
?	SHIFT /	0	1	1	1	1	1	1	3 F

FUNCTION	ACCESS	b7	b6	b5	b4	b3	b2	b1	HEX CODE
@	@	1	0	0	0	0	0	0	4 0
A	SHIFT A	1	0	0	0	0	0	1	4 1
B	SHIFT B	1	0	0	0	0	1	0	4 2
C	SHIFT C	1	0	0	0	0	1	1	4 3
D	SHIF D	1	0	0	0	1	0	0	4 4
E	SHIFT E	1	0	0	0	1	0	1	4 5
F	SHIFT F	1	0	0	0	1	1	0	4 6
G	SHIFT G	1	0	0	0	1	1	1	4 7
H	SHIFT H	1	0	0	1	0	0	0	4 8
I	SHIFT I	1	0	0	1	0	0	1	4 9
J	SHIFT J	1	0	0	1	0	1	0	4 A
K	SHIFT K	1	0	0	1	0	1	1	4 B
L	SHIFT L	1	0	0	1	1	0	0	4 C
M	SHIFT M	1	0	0	1	1	0	1	4 D
N	SHIFT N	1	0	0	1	1	1	0	4 E
O	SHIFT O	1	0	0	1	1	1	1	4 F
P	SHIFT P	1	0	1	0	0	0	0	5 0
Q	SHIFT Q	1	0	1	0	0	0	1	5 1
R	SHIFT R	1	0	1	0	0	1	0	5 2
S	SHIFT S	1	0	1	0	0	1	1	5 3
T	SHIFT T	1	0	1	0	1	0	0	5 4
U	SHIFT U	1	0	1	0	1	0	1	5 5
V	SHIFT V	1	0	1	0	1	1	0	5 6
W	SHIFT W	1	0	1	0	1	1	1	5 7
X	SHIFT X	1	0	1	1	0	0	0	5 8
Y	SHIFT Y	1	0	1	1	0	0	1	5 9
Z	SHIFT Z	1	0	1	1	0	1	0	5 A
[	[	1	0	1	1	0	1	1	5 B
\	\	1	0	1	1	1	0	0	5 C
]	]	1	0	1	1	1	0	1	5 D
^	^	1	0	1	1	1	1	0	5 E
–	SHIFT DEL	1	0	1	1	1	1	1	5 F
	SHIFT @	1	1	0	0	0	0	0	6 0
a	A	1	1	0	0	0	0	1	6 1
b	B	1	1	0	0	0	1	0	6 2
c	C	1	1	0	0	0	1	1	6 3
d	D	1	1	0	0	1	0	0	6 4
e	E	1	1	0	0	1	0	1	6 5
f	F	1	1	0	0	1	1	0	6 6
g	G	1	1	0	0	1	1	1	6 7
h	H	1	1	0	1	0	0	0	6 8
i	I	1	1	0	1	0	0	1	6 9
j	J	1	1	0	1	0	1	0	6 A
k	K	1	1	0	1	0	1	1	6 B
l	L	1	1	0	1	1	0	0	6 C
m	M	1	1	0	1	1	0	1	6 D
n	N	1	1	0	1	1	1	0	6 E
o	O	1	1	0	1	1	1	1	6 F
p	P	1	1	1	0	0	0	0	7 0
q	Q	1	1	1	0	0	0	1	7 1
r	R	1	1	1	0	0	1	0	7 2
s	S	1	1	1	0	0	1	1	7 3
t	T	1	1	1	0	1	0	0	7 4
u	U	1	1	1	0	1	0	1	7 5
v	V	1	1	1	0	1	1	0	7 6
w	W	1	1	1	0	1	1	1	7 7
x	X	1	1	1	1	0	0	0	7 8
y	Y	1	1	1	1	0	0	1	7 9
z	Z	1	1	1	1	0	1	0	7 A
{	SHIFT [	1	1	1	1	0	1	1	7 B
¦	SHIFT \	1	1	1	1	1	0	0	7 C
}	SHIFT]	1	1	1	1	1	0	1	7 D
~	SHIFT ^	1	1	1	1	1	1	0	7 E
DEL	DELETE	1	1	1	1	1	1	1	7 F

KEYBOARDS WOULD APPEAR TO BE DIFFICULT TO LAY HANDS ON THESE DAYS. CHILTMEAD, WHO ADVERTISE ELSEWHERE IN THIS ISSUE HAVE A LIMITED SUPPLY OF SUITABLE UNITS. RING TO CHECK BEFORE ORDERING. ANOTHER NAME TO TRY IS ELECTRONIC BROKERS. A THIRD ALTERNATIVE IS TO BUY THE SWITCH UNITS FROM R.S. AND BUILD YOUR OWN. EXPENSIVE THIS (THERE IS NO REASON WHY A PRE-ENCODED ASCII KEYBOARD CANNOT BE USED WITH SYSTEM 68.)

To be continued