nr. 44
december 1978
U.K. 50p.
U.S.A.|Can. $1.50

# elektor

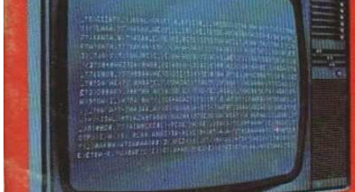
up-to-date electronics for lab and leisure

## ASCII keyboard

## plus

## elekterminal

## equals



## TV typewriter

| | | | | | |
|---|---|---|---|---|---|
| Austria | S. 33 | France | F. 7 | Norway | Kr. 9 |
| Belgium | F. 55 | Germany | DM. 3.80 | Sweden | Kr. 9 |
| Denmark | Kr. 9 | Netherlands | DFL. 3.25 | Switzerland | F. 4.40 |

# elekterminal

## Low cost video terminal for µP/TV typewriter applications

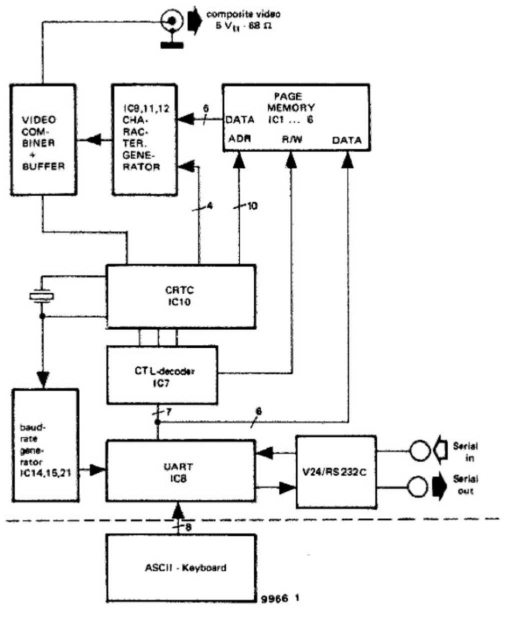
**There is no doubt that by far the most convenient and elegant method of displaying data from a microprocessor is on a Visual Display Unit (VDU). When used in conjunction with an ASCII keyboard (such as that described in last month's issue of Elektor) the video interface circuit described here forms a complete video data terminal which can be used with the Elektor SC/MP system or any microprocessor system possessing a serial input/output facility.**

The video terminal described in this article is of the serial (i.e. non memory-mapped) type, in which the video RAM used to store the characters to be generated on the screen is not shared by the microprocessor. There are several advantages of this type of system: firstly the terminal can be used independently (i.e. is not tied to a microprocessor) as a 'TV typewriter'. Secondly, the unit is TTY compatible, and in conjunction with a MODEM, can be employed to transmit/receive data over the telephone line. Thirdly, since most microprocessor systems already possess serial input/output routines, it means that the terminal can be used with the vast majority of different µP's and that the necessary device driver software is for the most part already present.

The Elekterminal uses one of the new single-chip CRT controllers, the SF.F 96364 from Thomson-CSF (Sescosem). Due in part to the large number of functions assumed by this one chip, the complete video interface for the terminal uses only 21 ICs, is accommodated on a board little larger than Eurocard format, yet offers the following comprehensive features:

- 1024 characters per page, formatted as 16 lines x 64 characters
- plug-in option allows character memory to be expanded to 16 pages
- choice of six different Baud rates: 75, 110, 150, 300, 600 or 1200
- programmable serial interface characteristics: i.e. choice of 6- or 7-bit ASCII code, even, odd or no-parity, 1- or 2 stop bits generation.
- choice of TTL or RS232C voltage levels

1



- normal (white on grey) or inverted (black on grey) video signal
- sophisticated cursor control and screen scrolling functions provided in hardware

## Block diagram

The block diagram of the Elekterminal is shown in figure 1. A detailed description of the ASCII keyboard was contained in last month's issue, thus the remainder of this article is devoted to the video interface card proper.

The ASCII output of the keyboard is fed directly to the UART. UART stands for Universal Asynchronous Receiver/Transmitter. This is an LSI IC which accepts data, whether in serial or parallel form, from a peripheral device (keyboard, modem) and transmits this data with appropriate serial-parallel or parallel-serial conversion to the CPU or video interface. Basically the UART allows the keyboard, VDU and CPU to communicate with one another. A more detailed description of this important IC is contained later in the article.

To be able to transmit data at different speeds, a programmable Baud rate generator is required. As was explained in the article on the cassette interface (Elektor 36), the Baud rate is defined as the total number of bits including control bits such as stop and parity bits which are transmitted in one second. The programmable Baud rate generator generates a number of frequencies which are 16 x the desired Baud rate. The output frequencies are derived by dividing down a clock input obtained from the crystal oscillator of the CRTC.

The CRTC is without doubt the most important component in the entire circuit. CRTC stands for Cathode Ray Tube Controller, however it might be more accurate to describe the chip as a 'video processor'. However one calls it, the device is another LSI IC which performs a wide variety of control functions with a minimum of peripheral hardware. In the past, video interface cards required a veritable mountain of discrete logic ICs to perform the tasks which are now assumed by this single chip. Among other things the CRTC generates the line and field sync pulses of the video signal, is responsible for the addressing of page memory, and controls the character generator. The chip also provides cursor control and screen scrolling in hardware. Like the UART, a more detailed discussion of the CRTC is contained later in the article.

The page memory, which holds the data to be displayed on the screen, is formed by a number of static RAMs. The entire memory is scanned once every frame (20 ms). The ASCII code which is stored in page memory is converted into a form suitable for display with the aid of a character generator; after parallel-serial conversion it is then added to the horizontal and vertical sync pulses in the video combiner. The latter provides a 5 $V_{pp}$ signal which can be fed direct to a video monitor or else via a VHF/UHF modulator (such as that contained in the October issue of Elektor this year) to the aerial input of a domestic TV receiver.

The only section of the circuit which remains to be discussed is the CTL decoder. This is basically a ROM which decodes the ASCII character transmitted by the UART and informs the CRTC whether it is a control signal (non-printing code) or a character to be displayed on the screen.

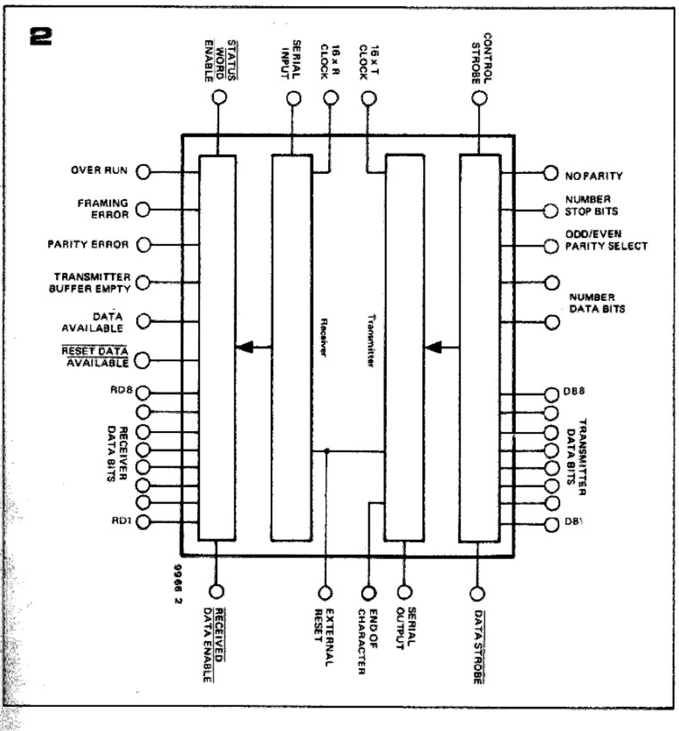
## UART, character generator and CRTC

Since many readers may be unfamiliar with these important devices, it is worth while taking a closer look at just exactly how they operate.

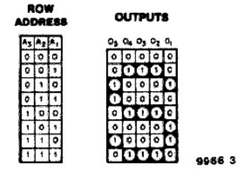
### UART

The block diagram of the AY-5-1013 UART which is used in the Elekterminal is shown in figure 2. In fact the UART can be thought of as two ICs (a transmitter and a receiver) which are housed in the same package and which combine certain functions in order to economise on the number of terminal pins. The UART is basically a device which provides asynchronous control of data communications, that is to say it is capable of both receiving and transmitting data at different rates, as well as performing parallel-serial and serial-parallel conversions, adding or deleting the necessary control and error detecting bits as required.

2



3



| $A_3$ | $A_2$ | $A_1$ | $O_5$ | $O_4$ | $O_3$ | $O_2$ | $O_1$ |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 |

**Figure 1. Block diagram of the Elekterminal.**

**Figure 2. Simplified internal block diagram of the UART.**

**Figure 3. Characters are generated by means of an 8 x 5 dot matrix. The top row of dots remains permanently extinguished.**

The transmitter and receiver have independent reference clock inputs, which determine the rate at which the data flow occurs, the UART can thus be used for both code- and speed conversion. The Baud rates are determined by the output frequency of a programmable Baud rate generator contained on-chip. The output frequency of the Baud Generator is 16 x the Baud rate.

Data are fed to the transmitter section of the UART (e.g. the ASCII output of the keyboard) in parallel form. The UART converts the parallel code into a serial data stream, adding the necessary start-, stop- and (if desired) parity bits. The system user can program the format of the transmitted or received serial character to suit his own requirements. That is to say that he has the choice of one or two stop bits, odd, even or no parity bit, and of selecting a 5, 6, 7 or 8-bit data word. The receiver section of the IC does exactly the opposite of the transmitter, i.e. it deletes the start and stop bits from the received serial character, checks for parity errors (which are flagged by setting the parity-error output), and presents the data in parallel form at the data outputs. When using the UART as a Baud rate- or code converter the data at the receiver outputs are fed to the data inputs of the transmitter; code conversion is performed by means of a ROM decoder connected between the two sections.

### Character generator

Less complex than the UART, but just as important is the character generator. This IC is responsible for translating the ASCII code stored in the video RAM into a format which can be used to generate the equivalent alphanumeric characters on the screen. In general the characters are formed by means of a dot matrix, the most common types of which are 5 x 7 and 7 x 9. Both types have their advantages and disadvantages. Because of the greater resolution available the 7 x 9 matrix produces characters which are more attractive and have greater detail. However the larger number of dots in the matrix requires a corresponding increase in the bandwidth of the video signal. With 64 characters per line this bandwidth is several Megahertz too large for conventional TV receivers, and results in poor picture definition. For this reason the 7 x 9 matrix is generally reserved for use with video monitors.

Although the characters generated by a 5 x 7 matrix are somewhat simpler, it is still possible to obtain excellent definition using a normal domestic TV receiver which has a video input. Even with the unavoidable picture degradation caused by a VHF/UHF modulator, the legibility of the resulting display is still perfectly satisfactory.

The format of the matrix produced by the character generator is illustrated in figure 3. The information contained in each row is stored in a ROM which is addressed by the 6-bit ASCII code (stored in the video RAM) together with a 3-bit row address which is supplied by control logic in the IC.

Figure 4 shows the simplified internal block diagram of the character generator. The total of nine address bits allows up to 512 different 5-bit rows to be selected; since 8 are required to form one complete character, the total number of characters available is 64. Depending upon the ASCII code, the correct data word for each row address is put on the five data outputs. With the aid of the output inhibit pin the data outputs can be placed in the high impedance state (tri-state mode), thereby allowing two or more character

4

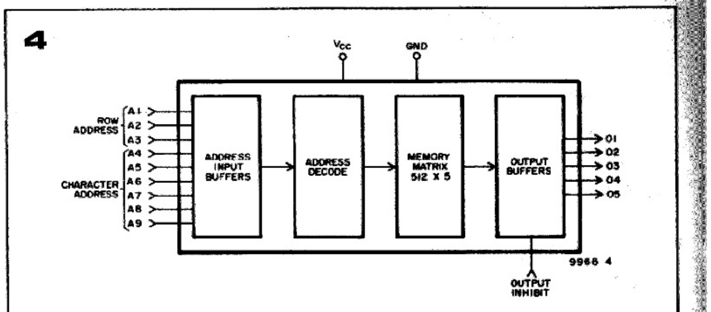


Figure 4. The character generator is basically nothing more than a specially programmed ROM. The only difference between it and a normal ROM is the shorter word length of 5 bits.

5

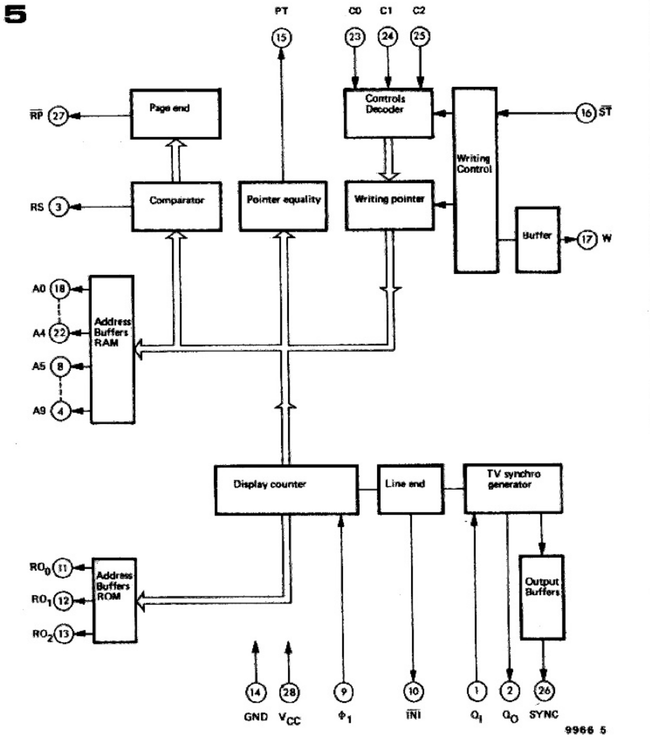


Figure 5. This simplified block diagram of the CRTC illustrates the large number of functions performed by the one IC.

Table 1. Depending upon the state of the control inputs, C0, C1 and C2, the SF.F 96364 will execute the following control functions.

Table 2. An overview of the division factors required to obtain the various Baud rate clock frequencies. Rounding up the figures listed in table 2a it is possible to obtain a low-cost Baud rate generator which is still accurate to within 1% (table 2b).

Table 1.

| | $C_2$ | $C_1$ | $C_0$ | Execution time ms |
|---|---|---|---|---|
| Page erase and cursor home (top-left) | 0 | 0 | 0 | 132 |
| End of line erase and cursor return (at left) | 0 | 0 | 1 | 8.3 |
| Line feed (cursor down) | 0 | 1 | 0 | 8.3 |
| Inhibition of the character sent | 0 | 1 | 1 | 8.3 |
| Cursor left (one position) | 1 | 0 | 0 | 8.3 |
| Erasure of cursor-line | 1 | 0 | 1 | 8.3 |
| Cursor up (one position) | 1 | 1 | 0 | 8.3 |
| Normal character | 1 | 1 | 1 | 8.3 |

Table 2a

| Baud rate | f UART | division factor 1000 MHz | division factor 1008 MHz |
|---|---|---|---|
| 75 | 1200 Hz | 833.33 | 840 |
| 110 | 1760 Hz | 568.18 | 572.73 |
| 150 | 2400 Hz | 416.67 | 420 |
| 300 | 4800 Hz | 208.33 | 210 |
| 600 | 9600 Hz | 104.17 | 105 |
| 1200 | 19200 Hz | 52.08 | 52.50 |

Table 2b

**Comparison of division factors**

| Baud rate | 1 MHz | 1008 MHz |
|---|---|---|
| 75 | 64 x 13 | 64 x 13 (+8) |
| 110 | 44 x 13 (−4) | 44 x 13 |
| 150 | 32 x 13 | 32 x 13 (+4) |
| 300 | 16 x 13 | 16 x 13 (+2) |
| 600 | 8 x 13 | 8 x 13 (+1) |
| 1200 | 4 x 13 | 4 x 13 |

generators to be connected in parallel so as to provide the remaining 64 ASCII characters, lower case letters and special symbols.

### CRTC

Almost all the major microprocessor manufacturers have already brought out a CRTC or are in the process of doing so. Most CRT Controllers are designed to be used in conjunction with a microprocessor, and some are even tied to a particular family of processor.

The device used here, however, is an exception to this rule, and the video interface card, of which it is the heart, can be used to form an independent VDU/TV typewriter as well as an output terminal for any microprocessor which has a serial output. The device in question is the SF.F 96364 from Thomson-CSF, which, as we shall see, provides all the control and timing signals required for screen display, as well as providing a number of sophisticated screen management facilities (cursor control and screen scrolling etc.) A simplified block diagram of the SF.F 96364 is shown in figure 5.

One of the most important tasks of the CRTC is to generate the sync pulses required to display a video signal. With the aid of a simple on-chip crystal oscillator the SF.F 96364 provides a close approximation to the CCIR standard sync signal. Line and field sync pulses are both combined in the one sync waveform. The sync generator also drives the display counter which is responsible for the addressing of the character generator (i.e. providing the correct row address) and of the video RAM (page memory). In addition, the display counter provides information for the cursor- and page-end comparators. The cursor comparator supplies a signal which ensures that the cursor appears at the correct point on the screen. The page-end comparator allows the amount of addressable video RAM to be extended to include extra pages, since the RS output is used to enable the $\overline{\text{VDU}}$ to 'turn a page' in mid-screen. The $\overline{\text{RP}}$ output clocks the counter used to address the additional pages of character memory.

A detailed explanation of how the video RAM is extended will be contained in a subsequent article describing an add-on circuit which will permit the use of up to 16 pages of memory.

The above mentioned functions of the CRTC are of course indispensable, however the most important features of the device are almost certainly the screen formatting control functions which are available in hardware. Many less sophisticated video interface cards rely upon software routines to provide control of cursor and screen scrolling, which means that they are necessarily tied to a microprocessor. The SF.F 96364, however, can provide fairly sophisticated screen control options on-chip, allowing it to function independently.

As was stated in the description of the block diagram in figure 1, the CTL decoder provides a 3-bit instruction code which informs the CRTC that the ASCII character transmitted by the UART is in fact a non-printable control character. Depending upon the code applied to pins $C_0$, $C_1$ and $C_2$, the CRTC will perform one of the cursor control functions listed in table 1. Certain control instructions take a comparatively long time to execute, since they have to be carried out during blanking intervals so as to minimise display distortion.

As will be explained later, the number of control functions can be extended by manipulating the W-(Write-)signal; this facility is exploited in the Elekterminal.

## Circuit

The 21 ICs and associated components which are shown in figure 6 form the complete circuit of the Elekterminal. All that is required to render the terminal operational is the addition of an ASCII encoded keyboard. Page memory, which holds the ASCII version of the characters to be displayed on the screen, is 6 bits wide and is formed by six 2102A4 1 K x 1 RAMs. The '4' in the type number indicates the access time of the device, which in this case is 450 ns. If character memory is to be expanded to several pages, it may well be worth investing in low power

6

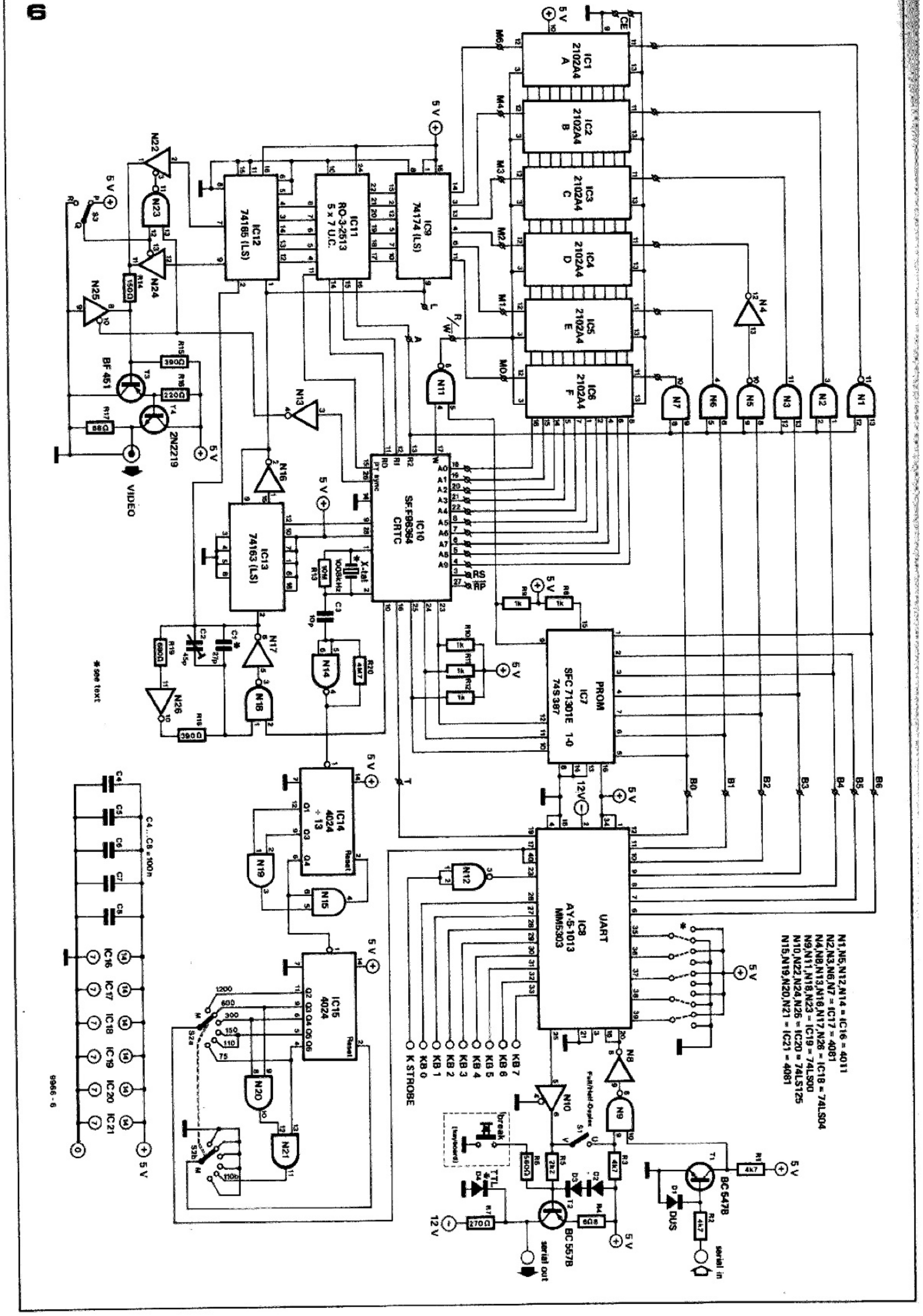

IC1 2102A4 A
IC2 2102A4 B
IC3 2102A4 C
IC4 2102A4 D
IC5 2102A4 E
IC6 2102A4 F
CE
M0
M1
M2
M3
M4
M5
R/W
IC9 74174 (LS)
IC11 RO-3-2513 5 x 7 U.C.
IC12 74165 (LS)
IC10 SFF96364 CRTC
IC13 74163 (LS)
PROM IC7 SFC 71301E 1-0 74S387
UART IC8 AY-5-1013 MM5303
IC14 4024 ÷ 13
IC15 4024
BF 451
2N2219
VIDEO
X-tal 1008kHz
* see text
C4 ... C8 = 100n
B0
B1
B2
B3
B4
B5
B6
KB 0
KB 1
KB 2
KB 3
KB 4
KB 5
KB 6
KB 7
K STROBE
Full/Half-Duplex
break
(keyboard)
TTL
BC547B
BC557B
DUS
serial in
serial out
1200
600
300
150
110
75
S2a
S2b
S3
S1
5 V
12 V
0
IC16
IC17
IC18
IC19
IC20
IC21
N1,N5,N12,N14 = IC16 = 4011
N2,N3,N6,N7 = IC17 = 4081
N4,N8,N13,N16,N17,N26 = IC18 = 74LS04
N9,N11,N18,N23 = IC19 = 74LS00
N10,N22,N24,N25 = IC20 = 74LS125
N15,N19,N20,N21 = IC21 = 4081
9966 - 6

Table 3.

| PIN | LEVEL | TRANSMITTED OR RECEIVED FORMAT |
|---|---|---|
| 35 | 1 | No parity bit |
| | 0 | Transmitted parity bit |
| 39 | 1 | Even parity |
| | 0 | Odd parity |
| 36 | 1 | 2 stop bits |
| | 0 | 1 stop bit |
| 37<br>38 | 0<br>0 | 5 bits/character |
| 37<br>38 | 0<br>1 | 6 bits/character |
| 37<br>38 | 1<br>0 | 7 bits/character |
| 37<br>38 | 1<br>1 | 8 bits/character |

Table 4.

| Function | Key | corresponds to |
|---|---|---|
| line-feed | LF | CTL J |
| carriage-return + erase to end of line | CR | CTL M |
| cursor up | VT | CTL K |
| cursor down | LF | CTL J |
| cursor left | BS | CTL H |
| cursor right | HT | CTL I |
| home cursor | FS | ——— |
| home cursor + page erase | FF | CTL L |
| scroll up (cursor down) | ESC | CTL [ |
| carriage return (no erasure) | - - (GS) | CTL ] |
| erase current line | - - (SUB) | CTL Z |

memories (2102AL4), since they can lead to a saving in current consumption of up to roughly 30%.

Since each character is formed by 8 rows of 5 bits, the ASCII code stored in the page memory is read out 8 x every frame. With 64 characters per line, the memory is scanned in blocks of 64 words. IC10, i.e. the CRTC, ensures that the same block is scanned 8 times in succession, and that the character generator is provided with the correct row addresses. The outputs of the memory are not connected directly to the character generator, but to an intermediate latch (IC9). The memory address can thus remain one step ahead of the position on-screen, which means it has ample time to set up the following ASCII code on its outputs.

The 5-bit parallel 'row' data from the character generator is fed to a shift register (IC12), where it is converted into serial form and thus becomes suitable for video display. This shift register is driven by a 'dot-clock' with a frequency of approx. 11 MHz. The dot-clock generator is formed by N17, N18 and N26. Since all eight rows of a character must be positioned directly beneath one another, the dot-clock generator is synchronised by the CRTC. This is achieved with the aid of the $\overline{\text{INI}}$ line (see figure 5) which goes low after the 64th character, thereby stopping the dot-clock generator until the following line sync pulse.

All memory addressing is clocked by the dot-clock, since the 'character-clock', which, via the $\Phi$1-input, drives the address counter in the CRTC, is derived from the dot-clock signal via a divide-by-eight counter (IC13).

The frequency of the dot-clock, which can be varied by means of C2, determines the width of a character: the lower the frequency the wider the character. The minimum usable frequency is determined by the available space on the TV screen. If too low a frequency is chosen, the characters will run off the edge of the screen. On the other hand, if the frequency is too high, the characters will be compressed into one portion of the screen, with a consequent loss of definition and legibility. Thus it is important that C2 be adjusted so as to obtain an optimal picture on-screen.

The spacing between character lines is regulated by the CRTC, which blanks the video signal for the period of four line scans. Between successive character lines there is therefore a space of half a line. Spacing between the actual characters is provided by the shift register, IC12. Since this is an eight-bit shift register and the character generator output is only five bits wide, each character can be preceded by two unmodulated dot columns and followed by one such column. Between each character there is thus a space three dots wide. The fact that, including spaces, each character is in fact eight bits wide explains why the character clock is derived from the dot-clock via a divide-by-eight counter.

The serial data stream output by the shift register is also available in its inverted form. One can therefore choose between a positive (white on grey) or negative (black on grey) video signal.

The video combiner is built round N22 . . . N25. Depending upon the position of S3, N22 . . . N24 provide a video signal of appropriate polarity. N23 not only inverts the signal at S3, but also inhibits the inverted video signal during the sync period. This causes the current through N25 during the sync pulses to be limited to an acceptable value.

The voltage divider network formed by R14 and R15 determines the ratio between the amplitude of the video signal and that of the sync signals. With the values shown the black level is around 30%.

Table 5.

Positive logic

| Address | $O_3$ | $O_2$ | $O_1$ | $O_0$ |
|---|---|---|---|---|
| 0 to 127 | 1 | 0 | 0 | 0 |
| 128 to 135 | 0 | 0 | 1 | 1 |
| 136 | 0 | 1 | 0 | 0 |
| 137 | 0 | 1 | 1 | 1 |
| 138 | 1 | 0 | 1 | 0 |
| 139 | 0 | 1 | 1 | 0 |
| 140 | 1 | 0 | 0 | 0 |
| 141 | 1 | 0 | 0 | 1 |
| 142 to 153 | 0 | 0 | 1 | 1 |
| 154 | 1 | 1 | 0 | 1 |
| 155 | 0 | 0 | 1 | 0 |
| 156 | 0 | 0 | 0 | 0 |
| 157 | 0 | 0 | 0 | 1 |
| 158, 159 | 0 | 0 | 1 | 1 |
| 160 to 254 | 1 | 1 | 1 | 1 |
| 255 | 0 | 0 | 1 | 1 |

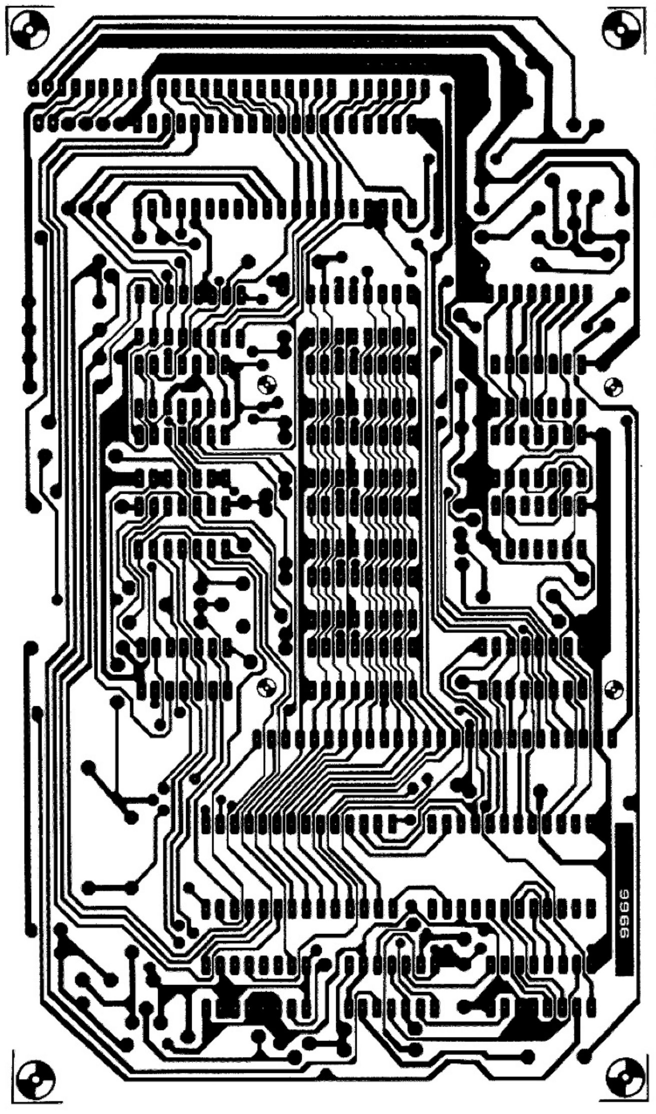
**Figure 6. Complete circuit diagram of the video interface for the Elekterminal. With the addition of an ASCII keyboard, which is connected to the K strobe- and KB0 . . . KB7 lines, the circuit forms a complete video data terminal.**

**Table 3. The programmable serial interface characteristics of the UART. This table applies both to the AY-5-1013 and to the MM5303. A recommended bit format is shown shaded; this format corresponds to the arrangement of wire links illustrated in figures 6 and 8.**

**Table 4. In addition to the control functions listed in table 1, the 4-bit PROM decoder offers several extra possibilities. These extra control functions can be generated either by special keys or with the aid of the control key and one of the data keys.**

**Table 5. The program for the PROM decoder IC7.**

7



The video combiner is followed by a buffer stage which has an output impedance of 68 Ω, and to which one can directly connect a length of coaxial cable. Assuming that the cable is terminated with the correct impedance it is possible to use lengths of over 10 m without any problems. The buffer stage has the effect of increasing the black level to around 35%; this can of course be corrected by altering the values of R14 and R15, however the increase in sync signal level does no harm and the adjustment is not really worth the bother.

So much for the circuitry which is responsible for actually generating the video signals; there remains the circuit which allows the unit to communicate with peripheral devices such as the CPU and/or keyboard. The most important interface element is of course the UART, the basic operation of which has already been described. As was stated then, the rate at which data is transmitted by the UART is determined by a clock signal, whose frequency is 16 x the desired Baud rate. Normally a monolithic Baud rate generator is employed to provide the clock signal, however these ICs are still fairly expensive and require a 1 MHz crystal to supply the basic clock frequency from which the 'x 16' clock signals are obtained by means of frequency division.

An obvious alternative is to make use of the crystal oscillator for the CRTC to provide the necessary clock frequencies. This can be done quite simply by first of all amplifying the signal at the output of the oscillator (with the aid of N14), and then feeding it to a programmable divider (IC14 and IC15). The accuracy of the clock frequencies obtained thereby is better than 1%. Generally speaking, the methods adopted for the transmission of data allow a reasonable tolerance in the accuracy of signals, so that a deviation in the region of 1% remains a perfectly acceptable figure. Table 2 lists the relationship between the UART frequencies and the (theoretical) division factors (2a) obtained with crystal frequencies of 1 MHz and 1008 kHz. The manufacturer of the SF.F 96364 recommends a crystal frequency of 1008 kHz in order to avoid interference from the mains frequency. In practice, however, a 1 MHz crystal is perfectly satisfactory.

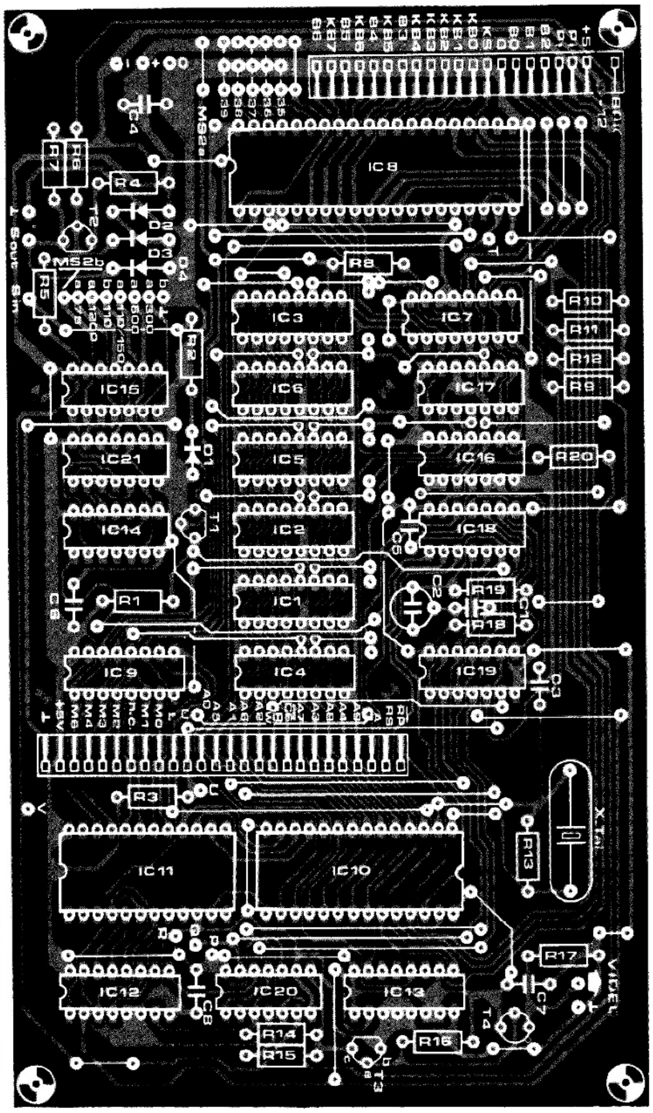
As can be seen from table 2, keeping to an accuracy of 1% and rounding up to the nearest whole figure produces identical division ratios for both crystal frequencies. The result is a Baud rate generator at roughly 20% the price of a monolithic type. Results obtained in practice have failed to indicate that this approach has any untoward effect upon the performance of the circuit.

Once provided with suitable clock frequencies, the UART can receive and transmit data at any one of six different Baud rates, which are selected by means of switch S2. An important point about most UARTs is that the output logic levels are often not TTL-compatible. The most common voltage levels used are the so-called RS232C and V24 norms. These two norms, which are virtually identical and more or less interchangeable, have the advantage of minimum signal levels of +5 V (logic '1') and –5 V (logic '0') and maximum levels of + and –15 V respectively. Thus they clearly have a much greater noise immunity than TTL logic levels.

The circuit described here attempts to reach a compromise solution by employing a discrete interface design which is compatible with both RS232C/V24 and TTL voltage levels. If the output signal is required to drive TTL loads, then D4 should be included. This diode limits the output voltage to –0.6 V. Without the diode the output voltage can swing between +5 V and –12 V. The output impedance is deliberately held low in order to facilitate matching with the cable.

As was mentioned earlier, the format of

8



**Parts list to figure 6:**

Resistors:
R1 . . . R3 = 4k7
R4 = 6Ω8
R5 = 2k2
R6 = 560 Ω
R7 = 270 Ω
R8 . . . R12 = 1 k
R13 = 10 M
R14 = 150 Ω
R15 = 390 Ω
R16 = 220 Ω
R17 = 68 Ω
R18 = 390 Ω
R19 = 680 Ω
R20 = 4M7

Capacitors:
C1 = 27 pF (see text)
C2 = 45 pF trimmer
C3 = 10 pF
C4 . . . C8 = 100 n

Semiconductors:
D1 . . . D4 = DUS
T1 = BC 107B, BC 547B or equ.
T2 = BC 177B, BC 557B or equ.
T3 = BF 451
T4 = 2N2219

ICs:
IC1 . . . IC6 = 2102-1,2102A4, 2102AL4
IC7 = SFC 71301E 1-0 (pre-programmed) or equivalent eg 74S387 (to be programmed as shown in table 5)
IC8 = AY-5-1013,MM5303
IC9 = 74LS174
IC10 = SF.F 96364 (Sescosem)
IC11 = RO-3-2513
IC12 = 74LS165
IC13 = 74LS163
IC14,IC15 = 4024
IC16 = 4011
IC17,IC21 = 4081
IC18 = 74LS04
IC19 = 74LS00
IC20 = 74LS125

Miscellaneous:
S1 = SPST switch
S2 = 2 pole 6 way switch
S3 = SP changeover switch
P.c.b. connectors (female): type ITT-Cannon G09
1 x 22 way (keyboard)
1 x 26 way (extension card)
For keyboard cable (male): type ITT-Cannon G09
1 x 22 way
X1 = 1008 kHz or 1000 kHz crystal

**Figures 7 and 8. Track pattern and component overlay of the p.c.b. for the Elekterminal (EPS 9966).**

the serial output/input signal can be programmed by the system user. The number of stop/start bits, choice of parity bit and of data word length can be selected by making the appropriate connections to pins 35 . . . 39. The details are listed in table 3, where a preferred format (7-bit code with even parity) is shown shaded. If desired the parity bit can be omitted (no parity) since although the UART checks for parity errors in the received signal, the parity-error output is not brought out externally. Thus the parity bit is only of use to the device receiving a character *transmitted* by the UART.

The Elekterminal is capable of being operated in both the full-duplex and half-duplex mode. In a full-duplex system, where the terminal is linked to a μP, the CPU and terminal communicate in both directions simultaneously. That is to say that the computer is programmed to echo what is transmitted (from the keyboard via the UART) back to the terminal display. In half-duplex systems the terminal is normally wired up so that the screen responds directly to the keyboard. Switching between half- and full-duplex is accomplished with the aid of S1, which is included between the serial output- and input pins of the UART.

The UART relays the ASCII code from the keyboard (or CPU) onto the data bus lines B0 . . . B6, where it is picked off by the CRTC and character memory. Before the data reaches the RAMs, however, it is converted from 7- to 6-bit ASCII code; bit 5 is ignored and bit 6 inverted. In addition, gates N1 . . . N7 offer the possibility of forcing the 'space' code (100000) onto the data inputs of the RAMs, so that, if the appropriate control code is applied to the CRTC, an entire line or the complete screen can be erased. These are in fact only two of the many control functions which the Elekterminal possesses.

The 7-bit ASCII control codes are detected and decoded by a 256 x 4 ROM (IC7), which forms the CTL decoder of figure 1. The ASCII code is placed on the address inputs of this ROM and the code which appears at the data outputs is fed to the C0, C1 and C2 control lines of the CRTC.

Table 1 has already listed a number of the control functions offered by the CRTC. However by utilising the read/write line of the RAMs it is possible to extend these. Table 4 provides an overview of all the various cursor control and screen scrolling functions which are provided by the Elekterminal. The majority of these functions can be selected by an individual key on the Elektor ASCII keyboard which was published last month. However, with the exception of 'home cursor', all the above functions can also be obtained using the control key and the appropriate data key, which means that the video interface is compatible with keyboards other than the Elektor model.

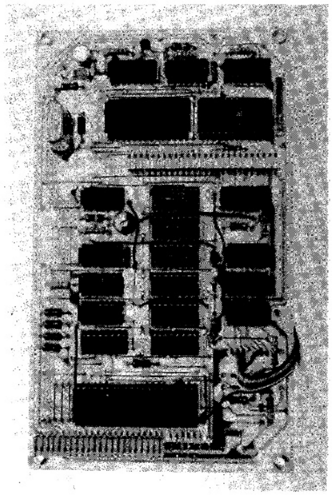
The PROM decoder for the CRTC is programmed as shown in figure 5. This device is available from a number of manufucaturers under the type number 74S387. Since only 128 combinations are possible with a 7-bit code, only half of the PROM is used.

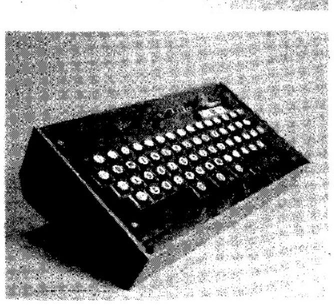
## Printed circuit board

The printed circuit board for the video interface card (see figures 7 and 8) is not much larger than Eurocard format, but nonetheless is single-sided. Because of this there are a considerable number of through connections to be made (approx. 60 in all), however the extra effort required is more than compensated for by the lower cost of a single-sided board. The p.c.b. has been specially designed to accomodate an extension board to increase the number of memory pages to 16. The latter (which will appear shortly) simply plugs into the video interface card with tne aid of edge connectors. At the left hand end of the video interface board is a simple connector to accept the ribbon cable from the ASCII keyboard. All the connections to the keyboard, i.e. including supply lines, can be made via the connector socket. This connector is also used to bring out the data lines from the UART. These connections will be required when incorporating the extension board.

The second p.c.b. connector provides access to all the address and data lines of the character memory and to two lines which are used to enable the RAMs to be addressed properly. This connector is designed to accomodate the memory extension board. It should be emphasised that the Elekterminal as presented here represents a complete output peripheral, which can be extended to accomodate more memory by plugging in one or more additional cards. The only on-board modification required is the removal of one through connection.

The UART (IC8) is programmed by means of wire links. The connections indicated on the component overlay correspond to the recommended format listed in table 3.





## Connection to a TV

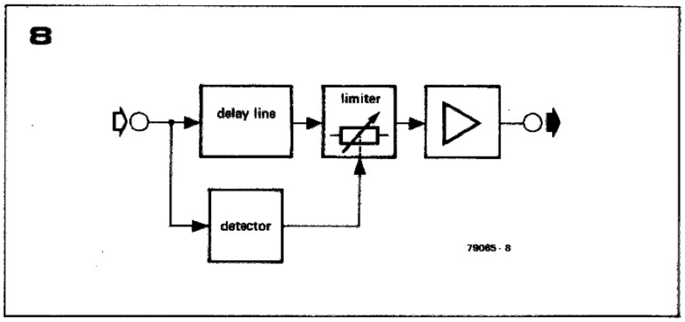
Not every TV has an input for an unmodulated video signal, but in those cases where one is present, or if a video monitor is being used, the amplitude of the terminal output signal has to be adjusted to suit the sensitivity of the input in question. This is best done as follows:

The video signal is fed to the TV receiver or monitor via a length of coaxial cable (50 . . . 75 Ω). The 'receiver-end' of the cable should terminate in a low impedance. A 100 Ω potentiometer is ideally suited for this purpose. The potentiometer can then be used to adjust the signal amplitude to a suitable value. Naturally, an alternative solution will have to be found in the case of sets which are provided with an internal terminal impedance.

If an input for unmodulated video signals is not available the output must be fed to a VHF/UHF modulator such as that published in the October issue of Elektor this year. Due to the large bandwidth of the video signal a certain degradation of picture quality is inevitable, however the resultant definition is still quite acceptable for the type of application for which the Elekterminal is intended. With or without a modulator, the signal amplitude should be adjusted so that the picture is 'sync-ing' with both positive and negative polarity video signals. This can easily be checked by changing the polarity several times in succession. One should also first ascertain that the line oscillator of the TV receiver is correctly tuned.

## Supply

Using normal memory ICs the current consumption of the circuit is around 750 mA (5 V). If low power memories are used, however, consumption drops to around 550 mA (5 V). The −12 V supply draws well under 100 mA. This means that the circuit could be powered using the spare capacity of an SC/MP system, assuming the latter was not driving a large amount of additional memory. Alternatively, a better idea might be to use the SC/MP power supply design to form a separate supply for the Elekterminal, especially if one bears in mind that extending the number of character memory pages will push up the current consumption. ◼



signal is slowed down to a frequency of 100 Hz (signal IV). The section of signal containing the first four cycles is compressed into half its original period (signal V), the resulting 'hole' or time gap is filled by repeating the first four cycles, which have been specially stored for this purpose (signal VI). Since the pitch and speech rhythms (at half their normal speed) of the original signal are preserved, the extra information is not important.

In practice the speech signal is processed by feeding it through a bucket brigade memory and continuously varying the clock frequency. A simplified block diagram of a variable speech processor is shown in figure 7b. A sawtooth generator, the frequency of which is determined by the speed of the tape recorder, is used to modulate the clock generator of the delay line. In the case of faster than normal playback the sawtooth ramps negative. During each period of the sawtooth the clock frequency is continuously varied from a maximum to a minimum value. The lower the clock frequency, the longer each successive sample takes to travel through the delay line. The result is that the time domain of the output signal is extended (its frequency is reduced), whilst leaving the shape of the waveform unaffected. Since all the frequency components of the original signal were 'slowed' by the same relative proportion, the harmonic structure and therefore the tonal character of the signal are preserved.

In the case of speech expansion (the time domain of the speech signal is expanded by playback at a slower than normal speed) the opposite occurs. The sawtooth ramps positive and the clock frequency varies from an initial minimum to a maximum value, with the result that the pitch of the signal is increased.

The variable speech processor can also be used to falsify the pitch of signals played at their correct speed, i.e. real-time pitch shifting. Thus by expanding the time domain of the speech signal the effect is of increasing its frequency and pitch, a trick which can be used for cartoon voices etc. Conversely, by compressing the time domain of the speech signal its frequency can be lowered. This technique is useful in unscrambling the voice signals of divers working in helium-filled atmospheres.

**Figure 8. Basic principle of a level control circuit incorporating a delay line. The detector monitors the input signal for overloads, and as soon as a signal peak is detected, activates the limiter circuit. The delay line ensures that the gain reduction occurs before the input signal reaches the limiter, thereby preventing initial transient distortion. The same principle can be used for click suppressors etc.**

Finally, two closely- related applications of delay lines in specialised studio equipment: level control units and click eliminators. In both cases the principle involved is the same, an audio signal is monitored for a particular irregularity. In one case it is signals above a preset maximum level, and in the other case a particular type of noise or distortion (clicks or pops caused by scratches, old recordings etc).

Delay lines are used to give the control circuits sufficient time to respond to signal overloads or noise transients. The basic arrangement is illustrated by the block diagram of figure 8. The input signal is fed to a delay line and to a detector circuit which controls the limiter or noise suppression circuit. Since the signal fed to the detector circuit is undelayed in the case of e.g. a signal overload, gain reduction sufficient to prevent overshoot will have occurred before the delayed signal (and signal peak) arrive at the limiter. Due to the reduction in the cost of bucket brigade memories, click suppressors are now a feasible proposition for the amateur, who can make more or less noise-free recordings of old records which previously were 'unlistenable-to'.

# CAPITALS from the ASCII keyboard

## BASIC made easy

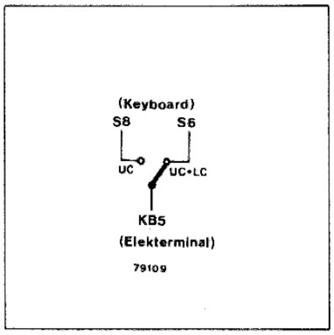
The ASCII keyboard (Elektor, November 1978) is more versatile than may appear at first sight. Some readers have commented that a 'shift-lock' would be useful, particularly when it is used for programming in BASIC. In actual fact, we can go one stage further: an 'upper-case lock'!

In the ASCII code, capitals and lower-case letters are distinguished by the value of the sixth bit (S6 on the character generator). For capitals, this bit is logic 'Ø'; for lower-case letters it is logic '1' (see table 1 in the original article).

The character generator used, the AY-5-2376, not only provides the usual 7-bit ASCII code: it has an eighth output (S8). Although this is not made clear in the data sheet, S8 can be used instead of S6. The result is all that could be desired: the shift key operates normally for numerals, punctuation marks etc. – but only CAPITALS are printed when a letter key is operated!

This facility can be extremely useful – for instance, when programming in NIBL. The 'shift' key need only be used when special symbols are required; it is no longer required for printing text.

A single-pole change-over switch can be added as shown in the figure.



# interface for µPs

**The specifications for a serial interface between computer and terminal are given by the so-called RS232C and V 24 standards — among others. Although these standards are in widespread use, this is not to say that all (micro-) computers include the corresponding interface.**

**Only a few components are required for a 'standard' interface. The circuit described in this article can be used in conjunction with both the Elektor SC/MP system and the popular KIM 1.**

The main difference between computer- and RS 232C/V24 signals is the definition of the signal levels. Within a computer system it is common practice to use TTL levels, with logic 'Ø' corresponding to 0 V and logic '1' to +5 V. The interface standards are rather different: logic 'Ø' may be anything between +5 V and +25 V, and logic '1' is 'defined' as between −5 V and −25 V. Note the level inversion: positive voltages for logic 'Ø' and negative voltages for logic '1'! The supply voltages in the SC/MP system are +5 V and −12 V, so it is 'logical' to use these levels for 'Ø and '1' respectively.

No negative supply voltage is available in the KIM 1 system, so an additional power supply must be added (giving a voltage between −5 V and −25 V). Two positive voltages are present (+5 V and +12 V). Either of these could be used for the positive logic level, but the higher voltage is to be preferred since it gives better noise immunity. The only disadvantage is that the power dissipation is higher in this case.

## SC/MP interface

The input/output software for the SC/MP normally uses the sense B input and flag Ø output for serial data transfer. A suitable interface for these connections is shown in figure 1.

The input interface (figure 1a) consists

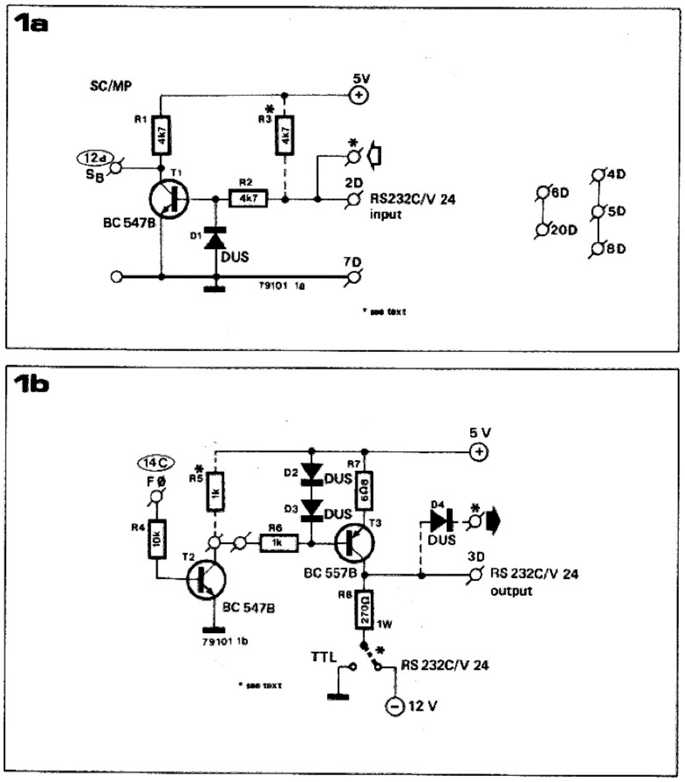
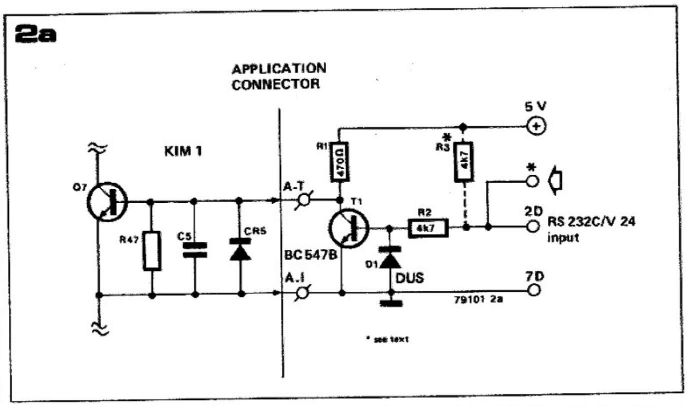
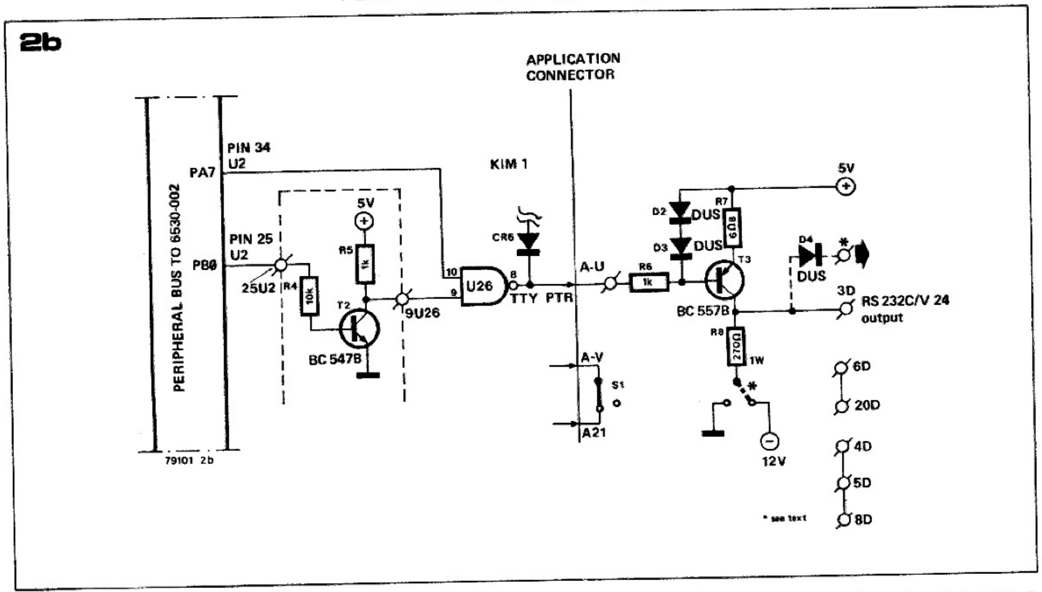


**Figure 1. Interface for the SC/MP system. The input side is given in figure 1a, and the output in figure 1b.**

**Figure 2. Modified communication interface for the KIM 1 microcomputer. Input and output sections are given in figures 2a and 2b, respectively; the relevant sections of the KIM circuit are also shown.**

**Figure 3. Multi-purpose printed circuit board for the communication interface (EPS79101). The track layout is given in figure 3a; figure 3b is the component layout for use with the Elektor SC/MP system and figure 3c is for use with the KIM.**





of four components. A diode (D1) and resistor (R2) limit the input signal, after which the transistor performs the conversion to TTL levels. Resistor R3 is not required if the input signal conforms to the official standards. However, the interface can also be fed from an opto-coupler or open-collector gate; in either of these cases R3 can be used as pull-up resistor.

The output side of the interface is slightly more complicated. The TTL levels from the SC/MP must be converted to +5 V and −12 V. A low output impedance is a must, since lines of up to 10 m (30') are quite common. Futhermore, the circuit must be short-circuit proof.

Figure 1b gives the circuit. A current source (T3) is used to obtain the low output impedance; it has the added virtue of being short-circuit proof. A second transistor (T2) is included as an inverter, to obtain the correct logic level relationship between the flag Ø output and the interface output. Resistor R5 is not strictly necessary: it improves the switching characteristic of the interface, giving sharper edges. The second output, via diode D4, can be used to drive the LED in an opto-coupler. The output current will have to be reduced in that application, by increasing the value of R7 to 15 Ω. The same circuit can also be used for buffering the flag Ø output, without altering the levels. In that case R8 should be connected to supply common instead of negative supply. The printed circuit board (figure 3) is designed to cater for all possible applications.

## KIM interface

Only a few modifications are required if the interface is to be used in conjunction with the KIM 1. The TTY (teletype) interface in the KIM system will also have to be modified slightly.

The serial data input is no problem: the same circuit can be used. The only difference is that the value of R1 (in figure 1a) must be reduced to 470 Ω to cope with the heavier load requirement. The circuit is therefore as shown in figure 2a; it can be connected to the KIM's 'application connector' as shown. For the serial output, some minor surgery on the KIM board is required. The TTY output on the KIM is intended for teletypes with a so-called current loop, but if it is to be used with the interface described here the polarity of the signal after the output gate must be inverted. Transistor T2 on the interface board is used for this, as shown in figure 2b. The track between PBØ (pin 25 of U2) and pin 9 of U26 on the KIM board must be broken, after which T2 can be wired in series as shown. The signal at output A-U on the application connector now has the correct polarity to drive the current source (T3 on the interface board).

## The printed circuit board

All the options described, both for

3a b c

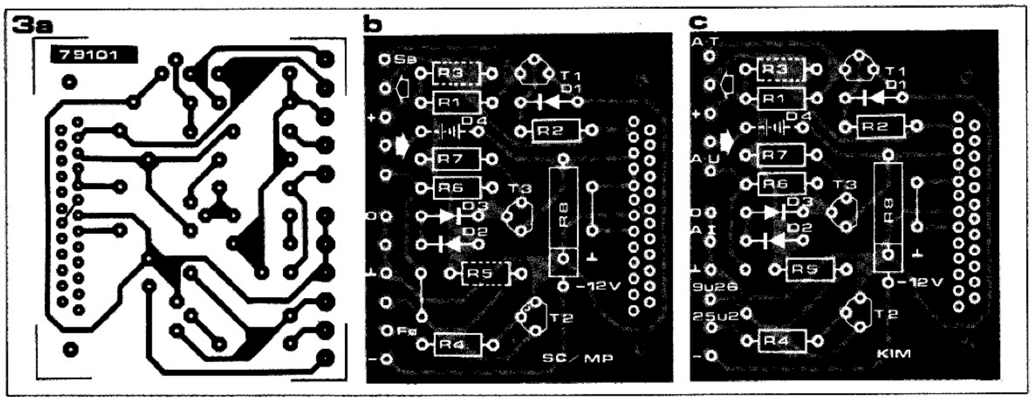


Table 1

```
              MEMORY-DUMP ROUTINE BY D. HENDRIKSEN

ØCØØ  C4 ØC 35 C4 ØØ 31 C4 ØC 37 C4 86 33 C4 ØC 36 C4
ØC1Ø  95 32 C4 ØD CB FD C4 ØD 3E 8F 8Ø C4 ØA 3E C4 2Ø
ØC2Ø  3E 35 Ø1 4Ø 35 4Ø 1C 1C 1C 1C Ø1 C3 8Ø 3E 35 Ø1
ØC3Ø  4Ø 35 4Ø D4 ØF Ø1 C3 8Ø 3E 31 Ø1 4Ø 31 4Ø 1C 1C
ØC4Ø  1C 1C Ø1 C3 8Ø 3E 31 Ø1 4Ø 31 4Ø D4 ØF Ø1 C3 8Ø
ØC5Ø  3E C4 2Ø 3E C4 1Ø CB FE C4 2Ø 3E C1 ØØ 1C 1C 1C
ØC6Ø  1C Ø1 C3 8Ø 3E C5 Ø1 D4 ØF Ø1 C3 8Ø 3E BB FE 9C
ØC7Ø  E5 BB FD 9C A1 C4 ØC CB FF Ø8 Ø8 Ø8 BB FF 9C F9
ØC8Ø  ØØ 9Ø 8F Ø5 ØC ØØ 3Ø 31 32 33 34 35 36 37 38 39
ØC9Ø  41 42 43 44 45 46 Ø1 C4 64 8F Ø6 Ø6 DC Ø1 Ø7 C4
ØCAØ  Ø9 C8 2Ø C4 FØ 8F Ø2 B8 1A 98 1Ø 4Ø D4 Ø1 C8 14
ØCBØ  Ø1 1C Ø1 Ø6 DC Ø1 EØ ØC Ø7 9Ø E8 Ø6 D4 FE Ø7 3E
ØCCØ  9Ø D4 ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
```

**Parts list**

Resistors:
R1 = 4k7 (470 Ω)
R2 = 4k7
R3 = 4k7*
R4 = 10 k
R5 = 1 k* (1 k)
R6 = 1 k
R7 = 6Ω8 or 15 Ω*
R8 = 270 Ω/1 W

Semiconductors:
T1,T2 = BC 107B, BC 547B or eq.
T3 = BC 177B, BC 557B or eq.
D1 . . . D3 = DUS
D4 = DUS*

* see text.
Where values for SC/MP and KIM differ, the values for the KIM are given in brackets.

SC/MP and KIM systems, can be mounted on the p.c. board shown in figure 3. The component layout for use with the SC/MP system is given in figure 3b; figure 3c corresponds to use in a KIM system.

A so-called modem connector can be used, if required. The mounting holes for the connector correspond to those of the p.c. board, so that the complete unit can then be mounted with only two bolts. The only thing to watch, in this case, is that the components must be mounted as nearly flush with the board as possible – there will not be much room between the board and the panel on which it is mounted! An alternative is to use a right-angle modem connector, so that the board can be mounted horizontally.

## Software

The monitor program for the KIM 1 already includes a teletype routine. The selection between a hexadecimal keyboard or teletype input is made by a wire link on the 'application connector'. In this case, since the TTY input is required (even if it is actually used for the Elekterminal), the wire link (or a switch) between pins A-V and A-21 on the connector must be included.

In the Elektor SC/MP system, no provision was made for connecting a teletype. However, only small programs will be required to obtain the necessary functions. As an example, a memory-dump routine is given in table 1. By means of this program, the memory contents will be printed out in hexadecimal – starting at a specified address. The length of the block is determined by the number of lines specified for the print-out. When the complete block has been 'dumped' the processor goes to the 'HALT' mode; operating the HALT-reset key causes a further block to be printed out, etc. The 'modify' routine is used to enter the new start address in memory locations ØCØ1 (upper address byte) and ØCØ4 (lower address byte). In the same way, the desired block length can be stored in location ØC13. The program itself is started at ØCØØ.

The program listing given in table 1 was actually printed using this memory-dump routine, as can be seen from the three underlined data bytes.

The transmission rate is 300 baud. ■

# SHIFT-LOCK for ASCII keyboard 35

The following SHIFT-LOCK circuit should prove a useful addition to the ASCII keyboard published in Elektor 43 (November '78). The circuit is also suitable for use with most other types of keyboard which are not already provided with this facility.

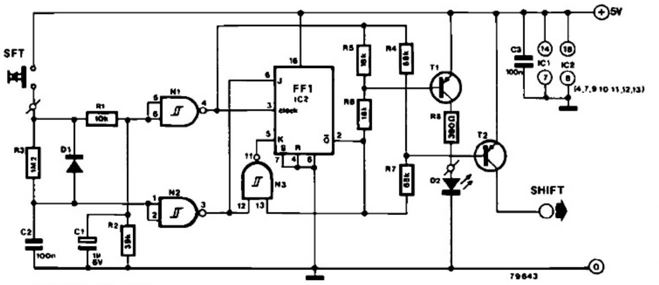
There is no need for an extra key to be mounted on the keyboard, since the original SHIFT key performs both the SHIFT and SHIFT-LOCK functions; the length of time for which the key is depressed determines which function is selected. If the SHIFT key is held down for longer than 0.2 seconds, the shift output will go low (inactive) as soon as the key is released. If, however, the key is only depressed briefly (i.e. for less than 0.2 sec), the shift output is held high (active) until pressed a second time, i.e. the key functions as a SHIFT-LOCK.

The timing for the circuit is provided by the RC constant of R3 and C2. As soon as the voltage across C2 reaches approximately 45% of the supply voltage, N2 will change the input conditions of the JK flip-flop, IC2. Assuming that the $\overline{Q}$ output is initially high, the circuit functions as follows:

Momentarily depressing the SHIFT key has no effect upon the output of N2. The J input of the flip-flop therefore remains high, and the K input low. Shortly after the key is pressed, the output of N1 goes low, taking the SHIFT output (via T2) high. When the SHIFT key is released, a positive going edge triggers the flip-flop, so that, given the state of the J and K inputs, the $\overline{Q}$ output goes low, taking the K input high, and ensuring that the SHIFT output is held high (via T1, T2). The next time the SHIFT key is held down briefly, the flip-flop is reset, i.e. the $\overline{Q}$ output is returned high, taking the SHIFT output low.

If the SHIFT key is originally depressed for longer than 0.2 seconds, the output of N2 goes low and takes the K input high, thereby holding the $\overline{Q}$ output high. The SHIFT output will go low as soon as the key is released. R1, C1 and R2 eliminate the effects of contact bounce, whilst LED D2 provides a visual indication of when the SHIFT-LOCK function is selected.

If the circuit is used in conjunction with the Elektor ASCII keyboard, the track to pin 4 of the AY-5-2376 should be broken and the circuit connected between the SHIFT key and the IC.



N1,N2,N3 = IC1 = 4093
FF1 = ½IC2 = 4027
D1 = DUS
T1,T2 = TUP

T. Frankemolen (The Netherlands)

# 2 switches - 2 lamps - 1 wire 36

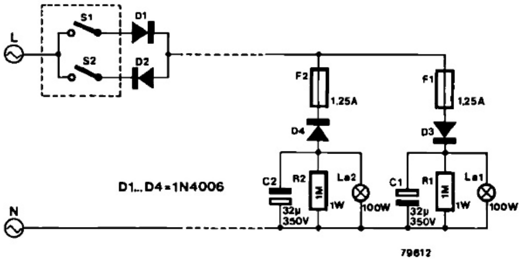
When housewiring, the addition of an extra switch and light to an existing circuit using the same power supply point would not normally cause any problems. However, the situation can arise where it is not possible to 'run' an extra cable between the additional switch and light thereby making it impractical to fit them.

The circuit described here is a simple but effective method of solving this problem by replacing the missing wire with a little ingenuity.

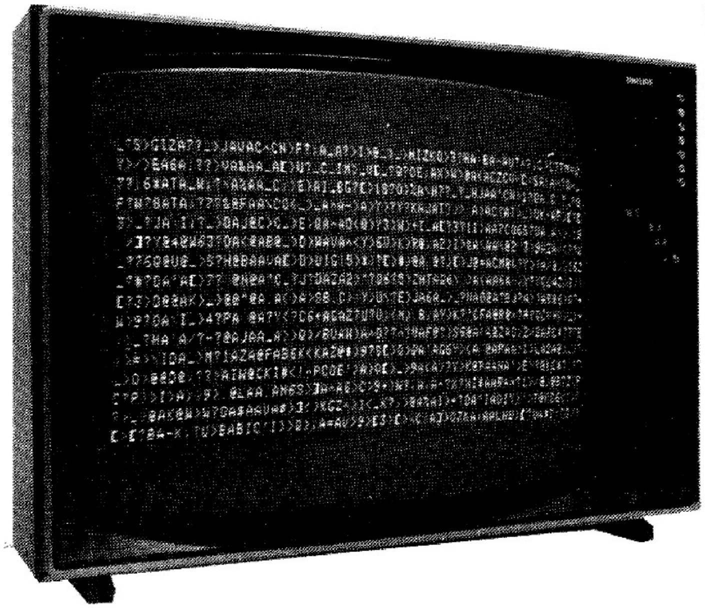
It will be seen from figure 1 that diodes D1 and D2 ensure that switch S1 controls lamp La1, whilst S2 controls lamp La2. The half-wave rectified mains voltage is partially smoothed by capacitors C1 and C2, so that an RMS voltage of approximately 240 V appears across the lamps, which therefore burn at normal intensity. The value of these capacitors is determined by the power rating of the lamps used. The appropriate value can be calculated by using the following equation:

$$C_X = 32 \cdot \frac{P_X}{100}$$

where $C_X$ is the new value of the capacitor (in $\mu$F) and $P_X$ the power rating (in W) of the corresponding lamp.

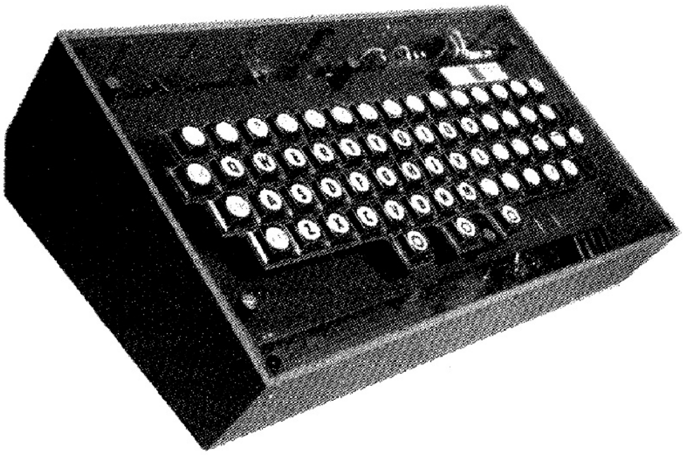


W. Richter (Germany)



# page extension for elekterminal

**With the aid of the extension board described here, the memory capacity of the Elekterminal can be expanded to 4 pages (each of 16 lines x 64 characters). Interconnecting the two boards is not a problem, since they can be mated quite simply by means of connectors.**



A memory capacity of 16 page lines is in practice somewhat restricting. Even a simple BASIC program will generally require additional lines. For this reason expansion of the video memory is highly desirable.

To increase the number of pages in the VDU's memory it is first necessary to provide a control circuit which will select the correct page, bearing in mind that the 16 lines displayed on the screen may be composed of sections of two successive pages. To this end a *page counter* is required, which selects the desired page by enabling the appropriate memory IC. The basic principle is illustrated in the block diagram of figure 1. Pages 1, 2 and 3 are accommodated on the extension board, whilst page Ø is housed on the Elekterminal board. The page counter is in turn controlled by the CRTC of the Elekterminal and by the up and down keys of the ASCII keyboard.

To be able to manipulate several memory pages satisfactorily the following functions are necessary:

- the page counter must be capable of counting up and down.
- the memory must 'wrap round', i.e. upon reaching the end of the last page, the start of the first page must reappear on the screen.
- conversely, when 'counting down', the last page must follow the first.
- it should be possible to reproduce sections of two successive pages on the screen.

The above facilities can be summarised by representing the memory as a drum, on which the four pages are spread out. The drum can be revolved in either direction, and any 16 successive lines can be displayed on the screen.

## Page counter

The operation of the page counter can best be explained with reference to the CRTC in the Elekterminal circuit. The latter contains a page-end comparator which provides two output signals, $\overline{RP}$ and RS. The RS output is used to indicate the transition somewhere in mid-screen from one page to another. If a complete page is on the screen, the RS output is high. If however sections of two pages are on the screen, then the page at the bottom of the screen is taken as the 'actual page'. During this portion of the page the RS output is high, whilst during the portion of the previous page it is low. For example, if lines 7 . . . 16 of page 2 and lines 1 . . . 6 of page 3 are on the screen, then the RS output is low for the first 10 lines and high for the last 6 lines.

The $\overline{RP}$ output provides a '0' pulse when a page boundary is exceeded at the bottom of the screen. This pulse is only generated if pressing the LF (line feed) or ESC (escape) key will result in the transition to the next page. Together, the RS and $\overline{RP}$ signals are used to control the page counter.

## Circuit

As can be seen from figure 2, the circuit of the page counter is quite straightforward, and consists of an up-down counter (IC1), a 4-bit full adder (IC2), and a 2-to-4 line decoder (IC3). The three additional pages of memory are formed by 18 RAM's, type 2102A4 (figure 3). It is also possible to use low power memories for this application (type 2102AL4), which would result in a saving of roughly 30% in current consumption. The extension board also includes an anti-bounce circuit (round N3 . . . N6) for the page-up and page-down keys on the ASCII keyboard, which could not be used until now. Their purpose is to enable the user to 'turn over' a complete page of memory at one go, i.e. scroll a full 16 lines up or down, regardless of whether it is one complete page or formed by sections of two successive pages.

When the $\overline{RP}$ output of the CRTC goes low, or the page-up key is pressed, the up-down counter is incremented by 1; pressing the page-down key causes the counter to decrement by 1. The full adder then determines the binary sum of the counter contents and the RS signal. Depending upon the result, the decoder takes the corresponding output low, thereby enabling the appropriate memory IC.

When a complete page is on screen the RS output is high, with the result that the page numbers are all increased by 1. The page numbering recognised by the page counter is shown in the block diagram of figure 1. As already mentioned, page Ø is situated on the Elekterminal board. If sections of two successive pages are on-screen, the RS output will be low for the first page, and high for the second, so that the counter will automatically 'turn the page' at the correct point. For a description of the operation of the page memories the reader is referred to the article on the Elekterminal (Elektor 44, December 1978).

## Printed circuit board

The printed circuit board for the extension to page memory (see figure 4) is provided with two connectors thereby facilitating interconnection with the terminal board. The 26-way connector should be soldered to the underside of the extension board, so that it mates with the connector socket on the terminal board. A number of connections however are not made via this connector. These are B0 . . . B4, B6 and the connections to the page-up and page-down keys. Provision is made for an 8-way connector, the pins of which are connected to the corresponding pins of the *second* connector on the terminal board. Of course the connector is not essential, it is equally possible to make these connections simply using ribbon cable.

1

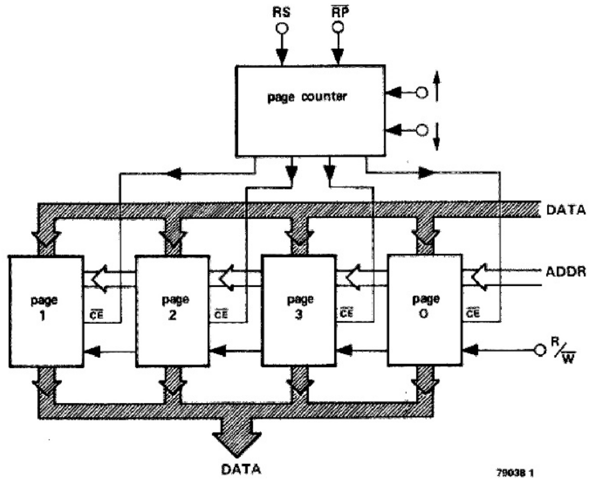


Figure 1. Block diagram of the extension to page memory. Page Ø is accommodated on the Elekterminal board.

2

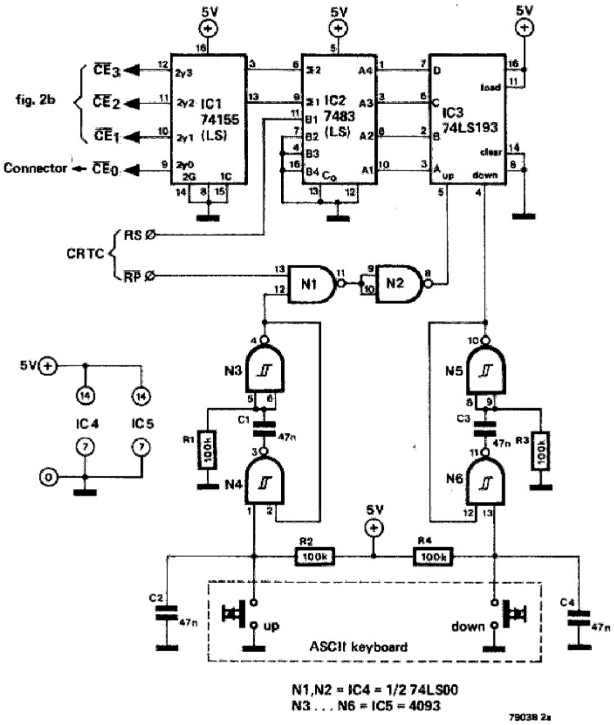


Figure 2. Circuit diagram of the page counter and anti-bounce logic. The numbering of the input and output connections corresponds to that used on the Elekterminal board.

3

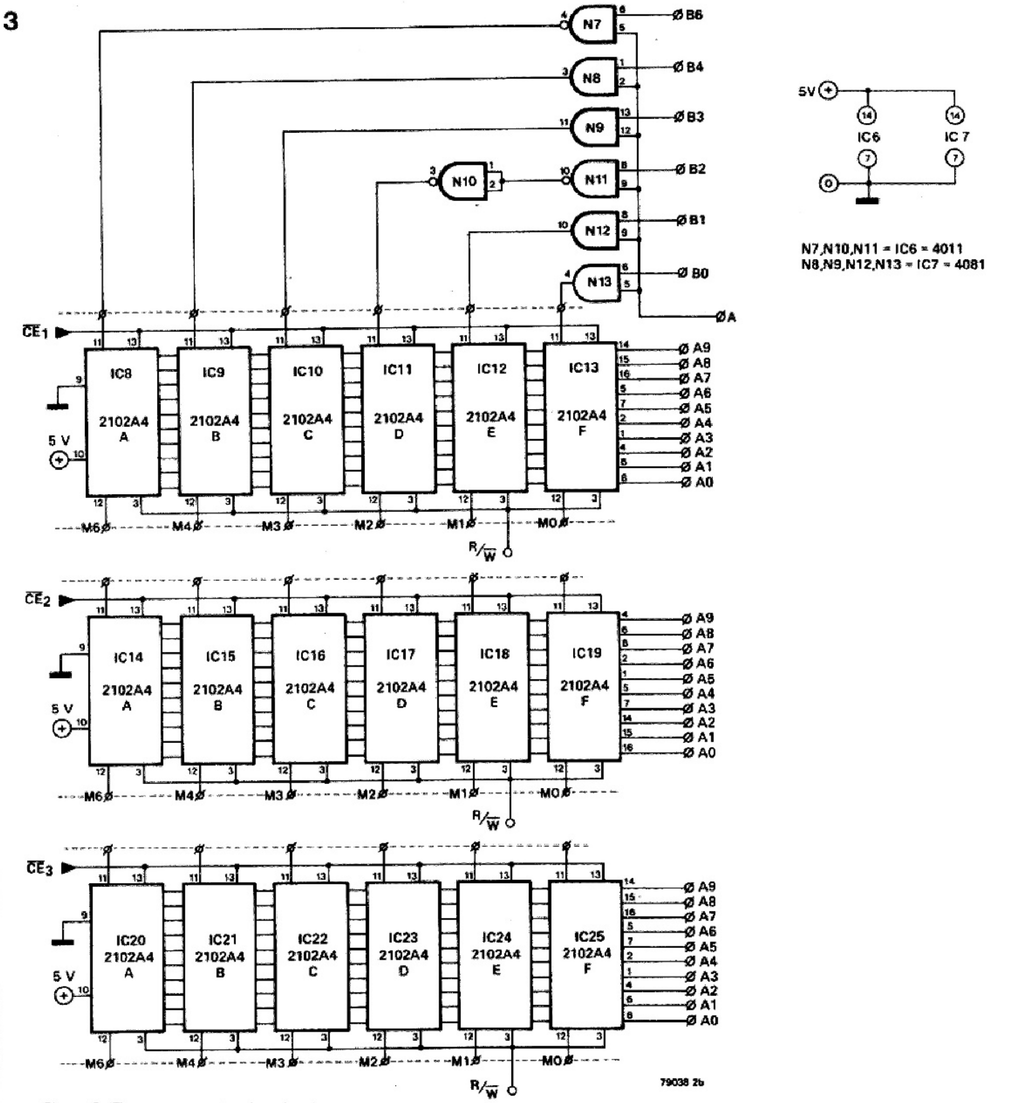


Figure 3. The memory extension circuit.

For the connections to the page-up and page-down keys there are two possibilities: either the key contacts can be connected directly to the extension board, or they can be routed via the extension board. If connectors are being used, then the latter option is the simplest. A small modification to the terminal board is also required before the memory extension is complete, namely the wire link between $\overline{CE}$ of IC3 and ground (see figure 5) should be removed.

## Scrolling

Once the memory is provided with extra pages, scrolling the text up a line at a time will normally be done with the aid of the ESC key. If the LF key is used, the text will scroll up, but the following line will be blanked, i.e. the line will appear vacant whilst the contents of the line are also erased from page memory. As mentioned, with the aid of the page-up and page-down keys the text can be scrolled in either direction a page at a time. Upon reaching the end of page memory (64 lines), the page counter wraps round to the start of the first page.

## Power supply

If normal memories are used, the current consumption of the extension circuit is roughly 600 mA. By employing low power memories this figure can be reduced to approximately 400 mA. It may prove necessary in some cases to uprate the Elekterminal power supply. Readers are referred to the article on the SC/MP power supply contained in Elektor 36, March 1978. ■

4

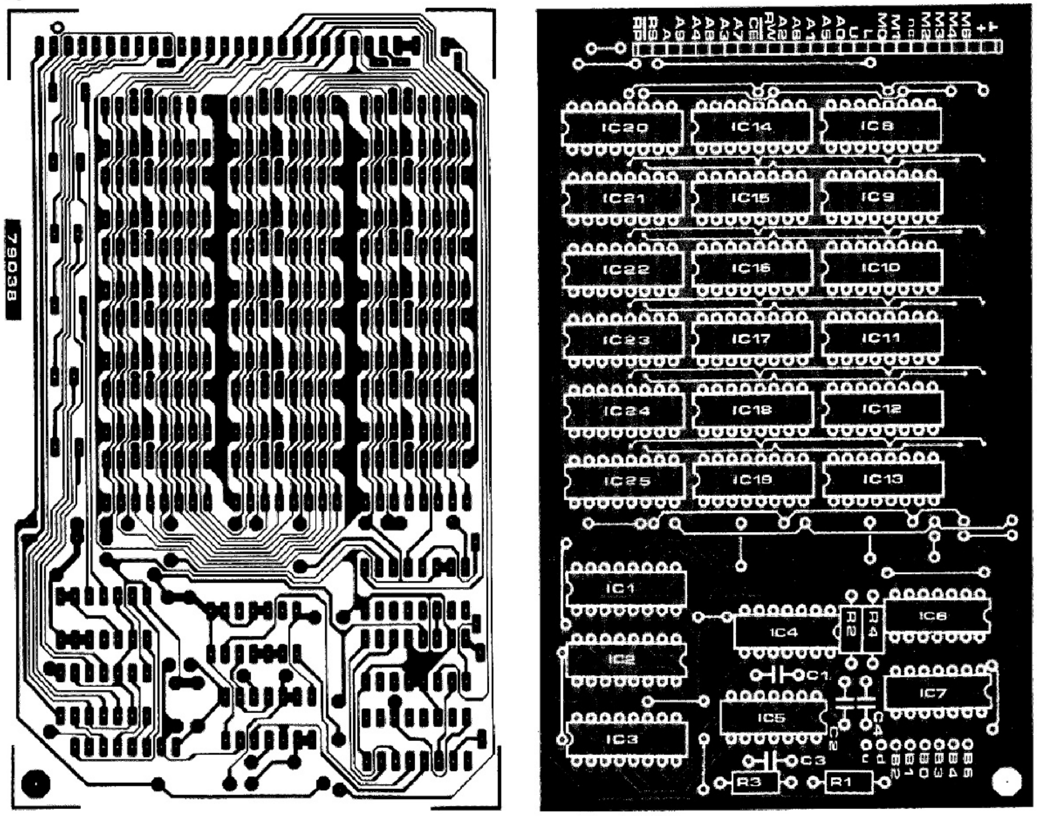


**Figure 4. Track pattern and component layout of the printed circuit board for the extension of page memory.**

**Parts list:**

Resistors:
R1 . . . R4 = 100 k

Capacitors:
C1 . . . C4 = 47 n

Semiconductors:
IC1 = 74LS155*
IC2 = 74LS83*
IC3 = 74LS193
IC4 = 74LS00
IC5 = 4093
IC6 = 4011
IC7 = 4081
IC8 . . . IC25 = 2102-1, 2102A4, 2102AL4

Miscellaneous: **
male connector mounted on underside of p.c.b. ITT-Cannon G09A45C4DBAA
1 x 26 way (if using ribbon cable, type G09A45C4DCAA)

*Low power Schottky is preferrable, however conventional TTL ICs can also be used.
**Note that the use of connectors is not essential

5

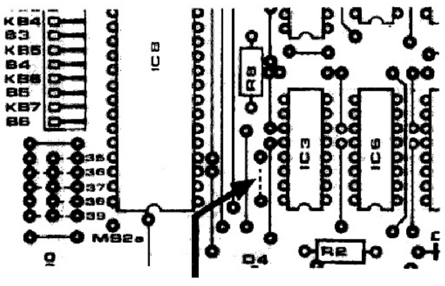
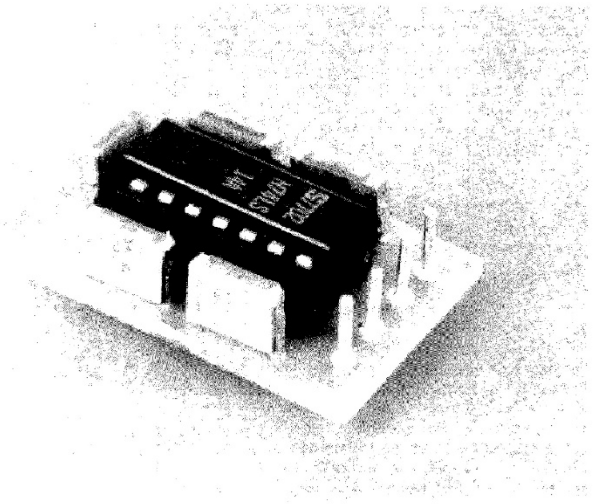


**Figure 5. This figure clearly shows the wire link on the Elekterminal board which must be removed.**

# the elekterminal width extender

## It's getting wider all the time . . .



W. Menzel

The elekterminal published in December 1978 meets the criteria set at that time. Users may, however, wish to adapt the row length to the screen width of the TV. This involves moving the beginning of a row to the left and adjusting the length of the row. With the elekterminal the dot clock generator is synchronized from the CRTC with the aid of the $\overline{\text{INI}}$-signal. Between the sync and the $\overline{\text{INI}}$ signals there is a fixed delay of 11 µs. By reducing this period the picture may be moved towards the left. In figure 1 a simple circuit has been drawn in which the picture can be moved in this way. The circuit is no more than a delay line with which the sync signal is delayed by 4 µs on its way to the video combiner. As a result, the $\overline{\text{INI}}$ and the sync signals are separated by about 7 µs.

The screen width is adjusted by lowering the frequency of the dot clock generator with the aid of capacitor C2 on the elekterminal. This modification enables the width of the letters to be slightly increased, so that they are easier to read. Furthermore, the video bandwidth is then also reduced, thereby bringing the letters in better focus.

The values of capacitors C given in figure 1 need to be empirically established. The indicated value of 1n5 proved to work well in most cases.

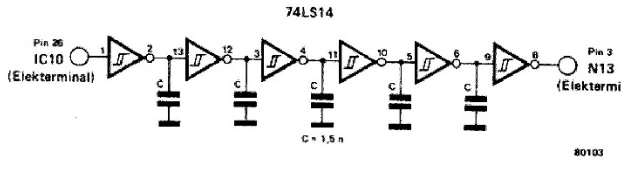


Figure 1. The circuit for delaying the sync signal.

# MPG meter

Due to unforseeable circumstances we have not been able to publish the fuel consumption meter as announced on the contents page in the last issue. The type of transducers required by the circuit are not available at present – contrary to assurances made to us – and we are therefore endeavouring to modify the circuit, so that a more readily obtainable type may be used. The finished result will be published as soon as possible.

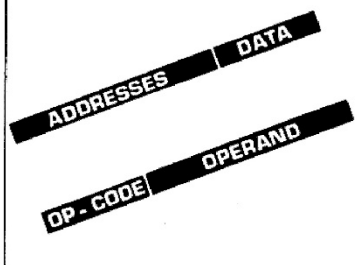
# junior computer sticker

A special sticker is available for the Junior Computer to increase display readability. It is made from red, transparent celluloid and can be affixed onto a perspex plate. This may then be fitted on top of the display printed circuit board with the aid of four bolts and spacers.

The words 'ADDRESSES' (left four displays) and 'DATA' belong to the normal use of the JC; 'OP-CODE' (left two displays) and 'OPERAND' (right four displays) are for editing and assembling.

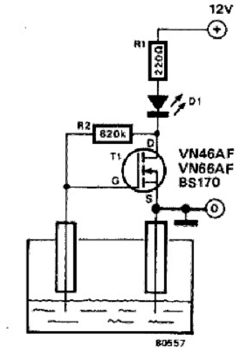
The sticker is delivered free with the EPS printed circuit boards for the JC. Those of you who have already got a JC board may apply for the sticker in writing (please enclose a stamped addressed envelope).

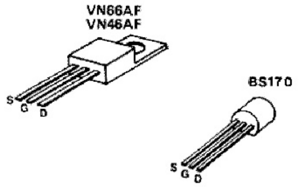
We would also like to point out that a complete kit of parts (including the programmed 2708) is available from Technomatic Ltd. Details are given in their advertisement on page UK 27.



# 27 water detector

The level of water in a water tank can be measured in various ways, some of which can of course be more complicated than others. The circuit published here lights a LED whenever the level of water drops below the electrodes. With a high water level, the FET hardly conducts or not at all, because the gate is then connected to earth and there is no voltage difference between gate and source. When the water level drops, the gate/source connection is interrupted. The gate is then at a positive potential by means of the 820 k resistor, thereby causing the FET to conduct. The LED will now light. If the reverse operation is needed, that is the LED lights when the electrodes are short circuited by the water, just connect the 'earthy' electrode in the circuit to the positive and wire R2 between gate and source.





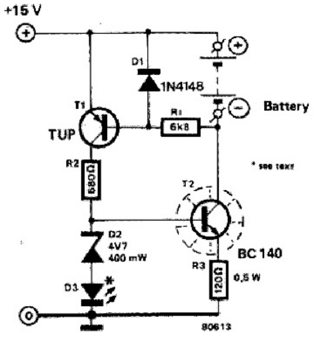
*ITT Applications*

# 28 intelligent NICAD charger

Like all things, NICAD chargers are subject to human error, that is, the batteries can be placed in the holder in two ways, correctly and incorrectly. This charger will refuse to operate unless the cells are fitted correctly. The charger consists of a current source (T2) to maintain the output current at about 50 mA. The zener diode D2 and LED hold the base drive of T2 at a constant level and thus also the voltage across R3. The current through R3 is therefore also constant, providing the correct conditions for charging NICADs at the collector of T2.

The protection circuit includes T1, D1 and R1. The terminal voltage of an incorrectly fitted NICAD will turn off T1 preventing the charger from operating. An indication of this will be given by the LED – it will not light. When the battery is fitted correctly T1 will turn on and the charger will function normally.

Although the charger is capable of charging up to four penlight cells, it will not detect a single cell being the wrong way round if two or three others are connected correctly at the same time. A small transformer, bridge rectifier and electrolytic capacitor are all that is required for a power supply. The circuit works well providing the NICADs are not completely discharged.

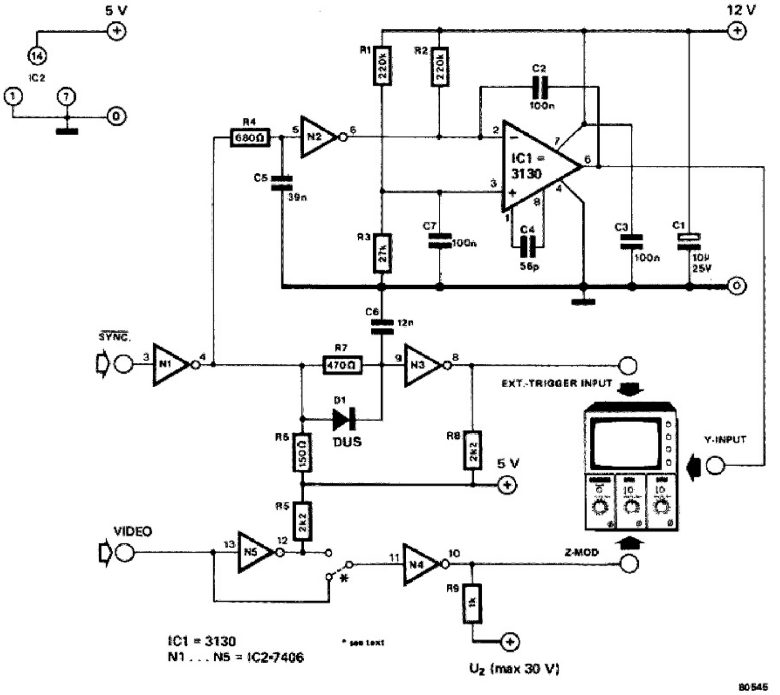


# 29 more scope for the elekterminal

Sometimes it can be very useful to connect the elekterminal (Elektor November and December 1978 issues) to an oscilloscope instead of to a TV. This of course, takes some doing, but if the circuit described here is used there should be no problems. The principle is very simple. If the oscilloscope has a Z input (which is usually the case, although the connection will probably be at the rear), the beam's intensity may be regulated by an external voltage. Thus, a video signal can be directly fed to the Z input of an oscilloscope.

If care is taken that the beam moves from left to right at the correct moment and obtains the right vertical deflection, a "TV picture" will appear on the screen. At the end of each row a horizontal synchronisation pulse is required. In the case of the elekterminal this is available at pin 4 of IC18 (N13). This output also contains the synchronisation pulses which make the beam flyback at the end of each raster. This combined signal is fed to the sync input of the circuit shown here. Each frame pulse triggers the ramp generator (IC1) via N2. The output voltage of IC1 will fall from 12 V to 0 V within 20 ms. This will be connected to the Y input of the oscilloscope.

The line pulses, which indicate that a new line must be written, are delayed by means of R7, D1 and C6, after which they are shaped by N3. Every 64 μs therefore, a pulse will be produced at the output of N3 which is fed to the external trigger input of the oscilloscope. This causes the oscilloscope's time base (which should be preset to about



6 µs per division) to start at every horizontal sync pulse and to write a line. Since the voltage at the Y input constantly falls during a raster period (20 ms) the lines (312 altogether) are written one below the other.

While these lines are being written, the beam's intensity is being varied. This is done by the picture information which is derived from pin 7 or pin 9 of IC12 and passed via a buffer to the oscilloscope's Z input. Not all Z inputs have the same sensitivity which is why any voltage up to a maximum of 30 V may be connected to the point marked $U_Z$. This is then switched by the open collector output of N4.

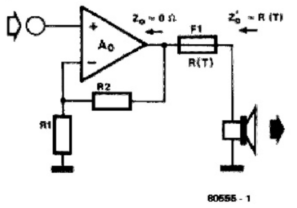
Depending on the type of oscilloscope, the voltage connected to $U_Z$ may have to be increased or decreased to obtain optimum intensity. The video signal can be inverted, or not, as required, by placing the wire link in the appropriate position.

# 30 loudspeaker fuse

The loudspeaker protection circuit described elsewhere in this issue can be used to protect multipath loudspeaker systems effectively from users with destructive tendencies. It can be done even more simply by using an old-fashioned glass fuse in series with the loudspeaker wiring. The melting value (in A) of the fuse is based on a compromise between a high value for the bass speaker, a less high value for the medium range and a low value for the treble speaker.

To place the fuse in series with the loudspeaker wiring (figure 1) as it stands would cause considerable problems. This is because a fuse has a relatively high series resistance. It is not too good for the muting factor of the amplifier or for the bass reproduction quality for that matter.

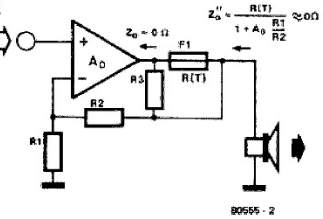
1



But there is more to it. When a current passes through the fuse it gets hot causing non-linear thermal behaviour – and the quality of the bass will show a negative temperature coefficient.

Something can be done about this. Include the fuse in the negative feedback loop (figure 2), in other words, tap off the negative feedback voltage from a point behind the fuse. The fuse is by-passed by means of resistor R3 which is small compared to R2 (slight influence on the DC setup of the amplifier), but large compared to the 4 Ω or 8 Ω load impedance. A value of 220 Ω (1 W) for R3 is fine.

2


$Z_o'' \approx \frac{R(T)}{1 + A_o \frac{R1}{R2}} \approx 0\Omega$
$Z_o \approx 0\ \Omega$
$A_o$
F1
R(T)
R3
R2
R1
80555 - 2

# high speed readout for elekterminal

**With a minor modification to the Elekterminal it is possible to 'store' the entire contents of the display (TV screen) on a cassette tape. The majority of the connections can be wired to the existing expansion sockets. For the remaining connections just three of the copper tracks between the UART and the CRTC on the main board of the Elekterminal have to be broken.**

The heart of the extension circuit consists of two multiplexers, IC1 and IC2. The information at one or the other set of inputs is passed to the outputs of the multiplexers depending on the (logic) state of the select input. The data lines of the keyboard and those of the memories are each connected to a separate 'group' of inputs. When the select input is taken low the keyboard data will be passed through to the outputs, and when the select input is high the data from the memories will pass through. To be able to store the memory contents on tape, therefore, the select input will have to be taken high. This is accomplished as follows:

When the start button, S1 of figure 1, is depressed, the $\overline{Q}$ output of FF1 goes low. As the Q output of FF1 is high, the memory WRITE signal is inhibited by N4. When one of the keys on the ASCII keyboard is depressed, preferably the space key or a control key, a strobe pulse (KS) will be generated. The leading edge of the strobe pulse will cause the data from the keyboard to be entered into the UART in parallel. With switch S1 in figure 2 closed, this information will be passed out of the transmitter section of the UART and back into the receiver section in series.

Once a complete character has been transferred in this manner the Data Available (DAV) output, pin 19, of the UART will go high for a short period of time. The trailing edge of this pulse will set the RS flip-flop formed by N1 and N2 which in turn will take the select inputs of the multiplexers high. The memory data, together with the R/$\overline{W}$ signal, will now be present at the multiplexer outputs. Shortly after the DAV pulse, an R/$\overline{W}$ pulse is generated which acts as a substitute for the strobe pulse. This means that the data held in memory will be shifted in and out of the UART repeatedly. This information passing out of the UART in series can now be stored on cassette. Since the recorder will have to be running before a key is depressed, the character which starts the cycle will also be recorded. This is why it is advisable to use the space key or a control key, as these will have very little effect on the actual display.

A DAV pulse is generated after each complete byte has been shifted out (and back in again). The leading edge of this pulse also increments the memory address counter so that the data stored in the next memory location becomes available. When the WRITE pulse occurs (immediately afterwards) the UART will operate once more.

The entire cycle is repeated until the complete page has been 'dumped'. The 'end-of-page' pulse ($\overline{RP}$) inhibits both the DAV pulse and the R/$\overline{W}$ pulse via FF2, N6 and N7.

By pressing the reset button (S2 of figure 1) FF1 and the RS flip-flop (N1/N2) are both reset and the Elekterminal can be operated normally once more. The information stored on cassette can be re-entered via the serial input.

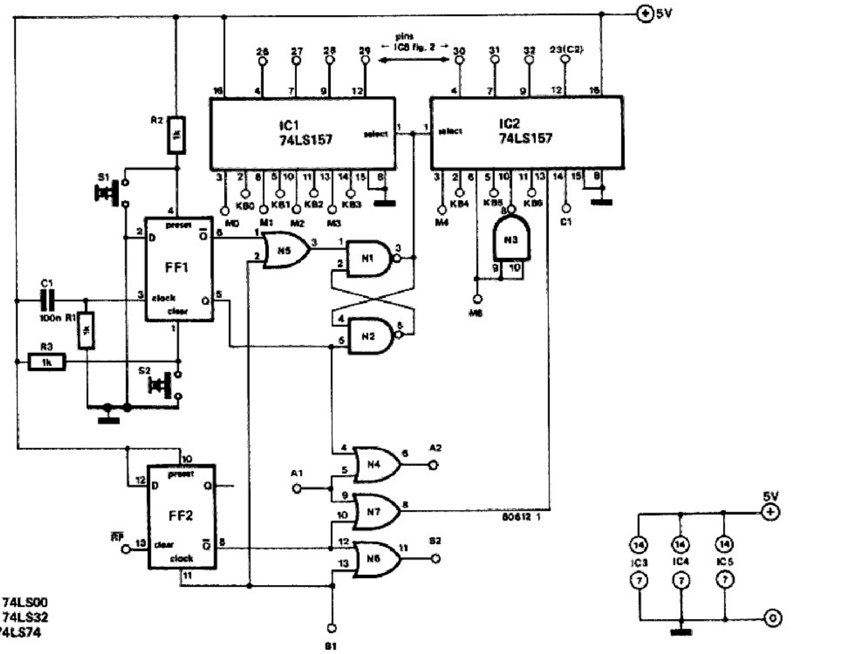
## The modifications

1. Break the copper track between pin 6 of IC19 (N11) and pin 3 of IC1 . . . IC6.
2. Break the copper track between pin 16 of IC10 (CRTC) and pin 19 of IC8 (UART).
3. Break the copper track between pin 3 of IC16 (N12) and pin 23 of IC8 (UART).
4. Connect points A1, A2, B1, B2, C1 and C2 of figure 1 to the corresponding points in figure 2.
5. Connect pin 27 of IC10 ($\overline{RP}$) in figure 2 to the point marked $\overline{RP}$ in figure 1.
6. Connect the following:

| in figure 1: | | in figure 2: |
|---|---|---|
| pin 3 of IC1 | to | point MØ (IC6) |
| pin 6 of IC1 | to | point M1 (IC5) |
| pin 10 of IC1 | to | point M2 (IC4) |
| pin 13 of IC1 | to | point M3 (IC3) |
| pin 3 of IC2 | to | point M4 (IC2) |
| pin 6 of IC2 | to | point M6 (IC1) |

7. Disconnect points KBØ . . . KB6 between the keyboard and IC8 in figure 2 and re-connect as follows:
   KBØ from keyboard to pin 2 of IC1
   KB1 from keyboard to pin 5 of IC1
   KB2 from keyboard to pin 11 of IC1
   KB3 from keyboard to pin 14 of IC1
   KB4 from keyboard to pin 2 of IC2
   KB5 from keyboard to pin 5 of IC2
   KB6 from keyboard to pin 11 of IC2
8. Finally, connect the outputs of the two multiplexers in figure 1 to the UART (IC8) in figure 2 as follows:
   pin 4 of IC1 to pin 26 of IC8
   pin 7 of IC1 to pin 27 of IC8
   pin 9 of IC1 to pin 28 of IC8
   pin 12 of IC1 to pin 29 of IC8
   pin 4 of IC2 to pin 30 of IC8
   pin 7 of IC2 to pin 31 of IC8
   pin 9 of IC2 to pin 32 of IC8

1



N1 . . . N3 = IC3 = 74LS00
N4 . . . N7 = IC4 = 74LS32
FF1,FF2 = IC5 = 74LS74

Figure 1. The complete extension circuit.

2

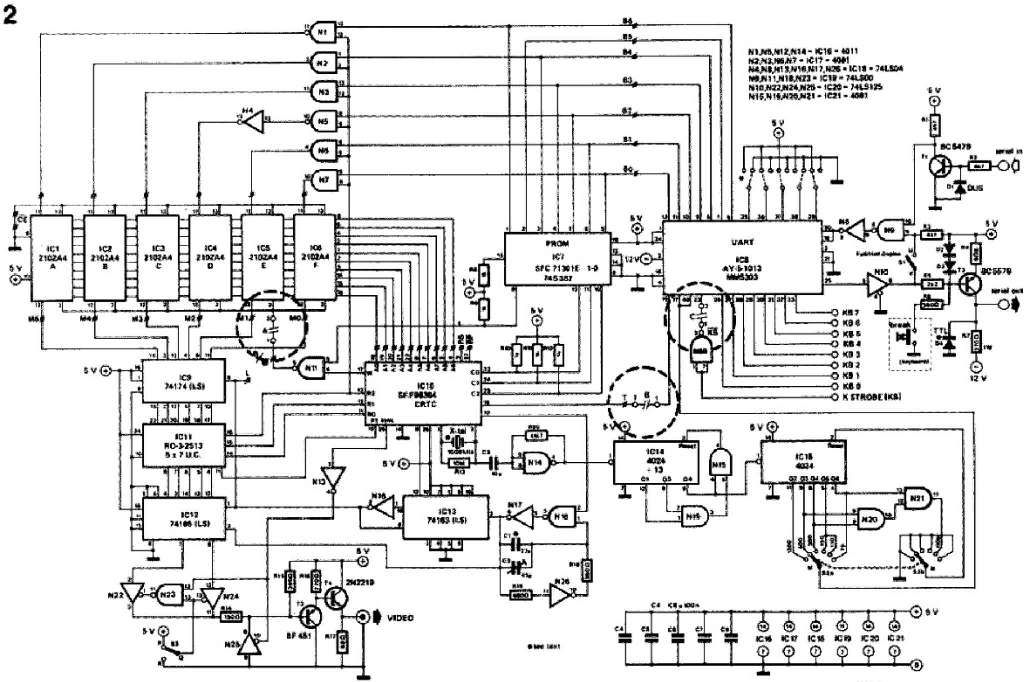


Figure 2. The original circuit, with the various connection points clearly marked. The points where copper tracks must be broken are shown inside the dotted circles.