February/March 1983

The magazine for the serious Sinclair user

sinclair projects

95p

SIX GREAT PROJECTS TO IMPROVE YOUR MACHINE

Financial decision-making

Printer Interface
Radio Teleprinter
Input/output Port
Spectrum Control Console
Machine code logical operations

sinclair projects

6 **MONEY MAKER**
A software project which helps the taking of major financial decisions

10 **PRINTER OUTLINE**
We take a close look at the ZX Printer

12 **RADIO TELEPRINTER**
A way to get the international news direct on your television

20 **SPECTRUM CONSOLE**
How to hide all those unsightly wires

24 **INPUT/OUTPUT PORT**
A new use for the cassette player port

30 **MACHINE CODE GUIDE**
An easy way to overcome the major problems of learning machine code

34 **RS232 INTERFACE**
A cheap way to allow you to connect your ZX-81 to a large printer

43 **LETTERS**
Your comments on our first issue

46 **EDGE CONNECTOR**
Our regular page illustrating the connections from both the ZX-81 and the Spectrum

47 **UPDATE**
We look back at our first issue and make some additions and corrections

FROM THE EDITOR

WELCOME to the second edition of *Sinclair Projects*. We hope that you find this more stimulating than the first one. Although a number of problems in the first issue meant that some diagrams were omitted, we hope that we have overcome those problems. Thank you to all those who wrote to offer valuable advice on how the magazine could be improved; there is obviously a need for a magazine such as this, as the response was beyond our expectations.

In this issue we are concentrating on a number of projects which, although slightly less hardware-based than those in the last issue, still provide the constructor with plenty to do. For example, we look at how we can use a ZX-81 to run a radio teleprinter. That will allow those with an appropriate receiver to intercept signals from news-agencies such as Reuters and TASS, without having to wait for the morning papers. We also look at a very simple input/output port which uses the cassette interface.

For those of you who are disenchanted with all the bits and pieces needed to get a Sinclair system going but do not want to pay £30 or more for a console to put them in, we offer a very simple control console for you to make.

Finally, on the construction side, we look at an interface, mainly in software, which will let you use commonly-available RS232 printers with Sinclair computers.

On the software side an interesting article on financial decision-making is coupled with a series of demonstration programs which illustrate the logical operations of the Z-80 processor. We also look at the inner workings of the ZX printer.

This marvellous list of items is supplemented by your letters an update on our first issue and our usual page illustrating the edge connections of the ZX-81 and the Spectrum. If you would like to contribute articles to *Sinclair Projects* we intend to publish guidelines for their writing in a future edition.

Editor Nigel Clark **Consultant editor** Robin Bradbeer **Technical assessors** Richard Larkin, Peter Whittle, Dave Buckley, Bazyle Butcher **Production editor** Harold Mayes MBE **Design** Elaine Bishop and Bill Scolding **Editorial director** John Sterlicchi **Advertisement director** Simon Horgan **Advertisement manager** John Ross **Advertisement Executive** Annette Burrows **Editorial/production assistant** Margaret Hawkins **Managing director** Terry Cartwright **Chairman** Richard Hease

Sinclair Projects is published bi-monthly by ECC Publications Ltd. It is in no way connected with Sinclair Research Ltd.

Telephone all departments: 01-359 7481. If you would like to contribute to any of the *Sinclair User* group of publications please send programs, articles or ideas for hardware projects to Sinclair User and Projects, ECC Publications, 30-31 Islington Green, London N1 8BJ. We pay £50 per 1,000 words for each article used.

 ISSN 0264/0449. Printed and typeset by Cradley Print PLC, Warley, W. Midlands. Distributed by Spotlight Magazine Distribution Ltd, 1 Benwell Road, Holloway, London N7. 01-607 6411.

Sinclair ZX Spectr

16K or 48K RAM... full-size moving-key keyboard... colour and sound... high-resolution graphics...

From only £125!

First, there was the world-beating Sinclair ZX80. The first personal computer for under £100.

Then, the ZX81. With up to 16K RAM available, and the ZX Printer. Giving more power and more flexibility. Together, they've sold over 500,000 so far, to make Sinclair world leaders in personal computing. And the ZX81 remains the ideal low-cost introduction to computing.

Now there's the ZX Spectrum! With up to 48K of RAM. A full-size moving-key keyboard. Vivid colour and sound. High-resolution graphics. And a low price that's unrivalled.

Professional power–personal computer price!

The ZX Spectrum incorporates all the proven features of the ZX81. But its new 16K BASIC ROM dramatically increases your computing power.

You have access to a range of 8 colours for foreground, background and border, together with a sound generator and high-resolution graphics.

You have the facility to support separate data files.

You have a choice of storage capacities (governed by the amount of RAM). 16K of RAM (which you can uprate later to 48K of RAM) or a massive 48K of RAM.

Yet the price of the Spectrum 16K is an amazing £125! Even the popular 48K version costs only £175!

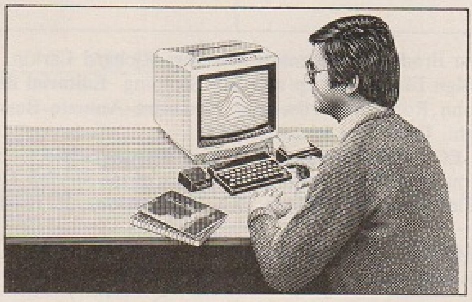
You may decide to begin with the 16K version. If so, you can still return it later for an upgrade. The cost? Around £60.

Ready to use today, easy to expand tomorrow

Your ZX Spectrum comes with a mains adaptor and all the necessary leads to connect to most cassette recorders and TVs (colour or black and white).

Employing Sinclair BASIC (now used in over 500,000 computers worldwide) the ZX Spectrum comes complete with two manuals which together represent a detailed course in BASIC programming. Whether you're a beginner or a competent programmer, you'll find them both of immense help. Depending on your computer experience, you'll quickly be moving into the colourful world of ZX Spectrum professional-level computing.

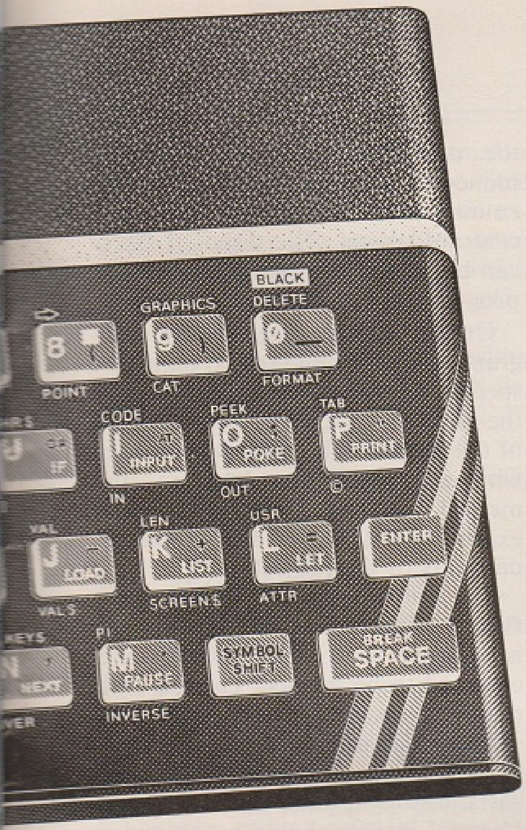
There's no need to stop there. The ZX Printer–available now–is fully compatible with the ZX Spectrum. And later this year there will be Microdrives for massive amounts of extra on-line storage, plus an RS232 / network interface board.



Key features of the Sinclair ZX Spectrum

- Full colour–8 colours each for foreground, background and border, plus flashing and brightness-intensity control.
- Sound–BEEP command with variable pitch and duration.
- Massive RAM–16K or 48K.
- Full-size moving-key keyboard–all keys at normal typewriter pitch, with repeat facility on each key.
- High-resolution–256 dots horizontally x 192 vertically, each individually addressable for true high-resolution graphics.
- ASCII character set–with upper- and lower-case characters.
- Teletext-compatible–user software can generate 40 characters per line or other settings.
- High speed LOAD & SAVE–16K in 100 seconds via cassette, with VERIFY & MERGE for programs and separate data files.
- Sinclair 16K extended BASIC–incorporating unique 'one-touch' keyword entry, syntax check, and report codes.

um



The ZX Printer – available now

Designed exclusively for use with the Sinclair ZX range of computers, the printer offers ZX Spectrum owners the full ASCII character set – including lower-case characters and high-resolution graphics.

A special feature is COPY which prints out exactly what is on the whole TV screen without the need for further instructions. Printing speed is 50 characters per second, with 32 characters per line and 9 lines per vertical inch.

The ZX Printer connects to the rear of your ZX Spectrum. A roll of paper (65ft long and 4in wide) is supplied, along with full instructions. Further supplies of paper are available in packs of five rolls.

The ZX Microdrive – coming soon

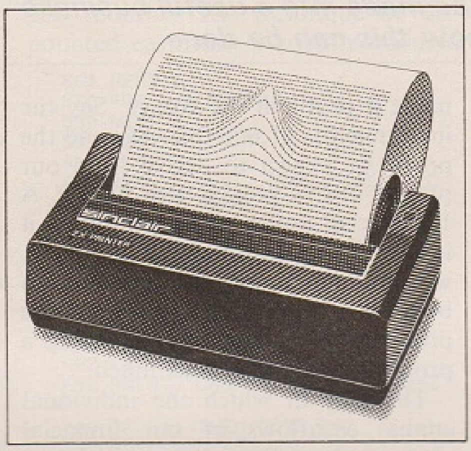
The new Microdrives, designed especially for the ZX Spectrum, are set to change the face of personal computing.

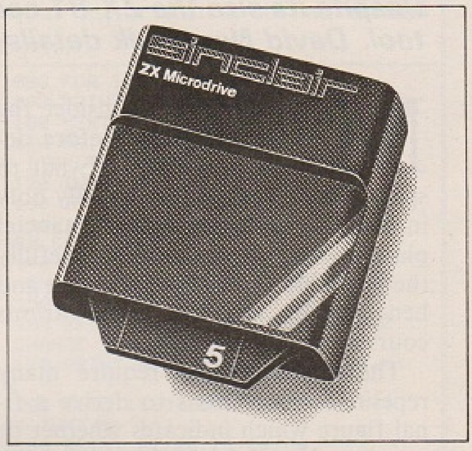
Each Microdrive is capable of holding up to 100K bytes using a single interchangeable microfloppy.

The transfer rate is 16K bytes per second, with average access time of 3.5 seconds. And you'll be able to connect up to 8 ZX Microdrives to your ZX Spectrum.

All the BASIC commands required for the Microdrives are included on the Spectrum.

A remarkable breakthrough at a remarkable price. The Microdrives are available later this year, for around £50.





How to order your ZX Spectrum

BY PHONE – Access, Barclaycard or Trustcard holders can call 01-200 0200 for personal attention 24 hours a day, every day. BY FREEPOST – use the no-stamp needed coupon below. You can pay by cheque, postal order, Access, Barclaycard or Trustcard.

EITHER WAY – please allow up to 28 days for delivery. And there's a 14-day money-back option, of course. We want you to be satisfied beyond doubt – and we have no doubt that you will be.

To: Sinclair Research, FREEPOST, Camberley, Surrey, GU15 3BR. **Order**

Qty	Item	Code	Item Price £	Total £
	Sinclair ZX Spectrum – 16K RAM version	100	125.00	
	Sinclair ZX Spectrum – 48K RAM version	101	175.00	
	Sinclair ZX Printer	27	59.95	
	Printer paper (pack of 5 rolls)	16	11.95	
	Postage and packing: orders under £100	28	2.95	
	orders over £100	29	4.95	
			Total £	

Please tick if you require a VAT receipt ☐

*I enclose a cheque/postal order payable to Sinclair Research Ltd for £ ____

*Please charge to my Access/Barclaycard/Trustcard account no.

*Please delete/complete as applicable

Signature

PLEASE PRINT

Name: Mr/Mrs/Miss

Address

MEG812

FREEPOST – no stamp needed. **Prices apply to UK only.** **Export prices on application.**

ZX Spectrum software on cassettes – available now

The first 21 software cassettes are now available directly from Sinclair. Produced by ICL and Psion, subjects include games, education, and business/household management. Galactic Invasion...Flight Simulation...Chess...History...Inventions...VU-CALC...VU-3D ...47 programs in all. There's something for everyone, and they all make full use of the Spectrum's colour, sound and graphics capabilities. You'll receive a detailed catalogue with your Spectrum.

RS232/network interface board

This interface, available later this year, will enable you to connect your ZX Spectrum to a whole host of printers, terminals and other computers.

The potential is enormous. And the astonishingly low price of only £20 is possible only because the operating systems are already designed into the ROM.

sinclair

Sinclair Research Ltd, Stanhope Road, Camberley, Surrey GU15 3PS. Tel: Camberley (0276) 685311.

Planning the cash for better decisions

Despite its size the ZX-81 can be made into a useful business tool. David Nowotnik details how this can be done

MOST PEOPLE consider the options carefully before deciding how and on what to spend their income. It is usually only in business decisions that a financial plan is made, considering carefully the probable income/expenditure and hence profit, resulting from various courses of action.

Those plans often require many repetitive calculations to derive a final figure which indicates whether or not a project appears financially viable. Computers now provide considerable help in carrying-out the calculations and so speed the decision-making process. While the Sinclair ZX-81 cannot match the convenience and flexibility of more expensive micros, there is no reason why it cannot also carry-out the repetitive business calculations and present the results in a form which may be used in making decisions.

A program called Investment Appraisal uses the technique of discounted cashflow and allows the ZX-81 to become a modest business machine.

When we want to decide whether a project is worth considering, a financial plan is made on the cashflow expected from that project. That is, for each year of the project, all items of income and expenditure are estimated and the nett cash flow for each year is calculated.

A simple approach in making the final decision would then be to sum all the individual nett cashflows to derive a figure which indicates whether the project will or will not show an overall profit.

In real terms, the approach is unrealistic as it ignores the time value of money. That may sound confusing but consider that, in most cases, we will want to invest money now to make a profit in the future. Say the investment is £1,000 this year and the profit we expect to derive from our project in five years is also £1,000. A 100 percent return sounds good but it is over five years. In the same period, we could have put the money in a building society and without all the problems of producing and selling a product, made almost as much.

The way in which the individual annual cashflows of our financial plan can be adjusted for their time value is by the technique of discounted cashflow. To consider how it works, say we invested £100 at 10 percent interest. At the end of one year we would have £110. So we could say that £110 in one year's time is worth £100 now, i.e., the discounted value of £110 in the second year of the project is £100. In that way, all cashflows of our financial plan can be discounted to year one, so that their time value is eliminated.

The ZX-81 program Investment Appraisal appears in figure one. The first part of the program sets up the matrix for all the data, then allows for entry of that data. When the program is RUN, the project name and term are the first two pieces of information required. The project term is, usually, the development time plus the product life, in years.

The next piece of information required is the number of income/expenditure items. That is the total number of items which will contribute to the individual annual cashflows—e.g., sales, overheads, production costs, development, advertising. The final piece of information required on the first page is the factor for discount. Arguments for the choice of discount factor are complex but a simple solution would be to choose a figure which reflects the percentage cost of borrowing money to finance the project. Discount factors of between 10–25 percent are typical. The discount factor can be altered, if desired, later in the program.

On the second page of the program, you will be asked to name the specific income/expenditure items. In the remaining pages of the first part of the program the table is completed with entry of all the data, i.e., numerical values for all the income/expenditure items for all the individual years of the project.

The unit of this data is not specified; it could be £ or £'000. Whatever is selected as the unit, all data is expected to be integer. There is little point in decimal-point accuracy in values which are predicted. Essentially they are a mixture of sales forecasts, historical data, and experience. Items of income should be entered as positive values; items of expenditure should be negative.

Once the data entry is complete, a year-by-year cashflow forecast will be produced. That will include the discounted cashflow, and the cumulative discounted cashflow which is the sum of all the annual cashflows up to the year shown on the screen.

In many projects, the initial CDCF will be negative as development outweighs nett income. When CDCF becomes zero, the break-even point has been reached. CDCF on the final year of the project provides a guide of the overall profitability of the project. The 32-column format of the ZX printer does not allow the complete table of results to be printed in one attempt. Instead, each page of results on the screen may be copied, then scissors plus Sellotape used to obtain a table of results if required, say, for a report.

When the results sequence has been completed, several options are available. The results display can be repeated, the data modified, or the program plus data saved on tape. The ability to modify the data allows for "what if" tests to be made; e.g., what if the production costs are 10 percent greater than expected? What if additional sales are generated? That flexibility allows the decision-maker to test the project at the possible extremes of variation from that expected. Results are available within seconds on the computer, which would take minutes or hours calculated manually.

The save program and data facility are added for convenience; the decision-maker may want to re-examine the figures at a later date and avoid the tedious re-entry of data.

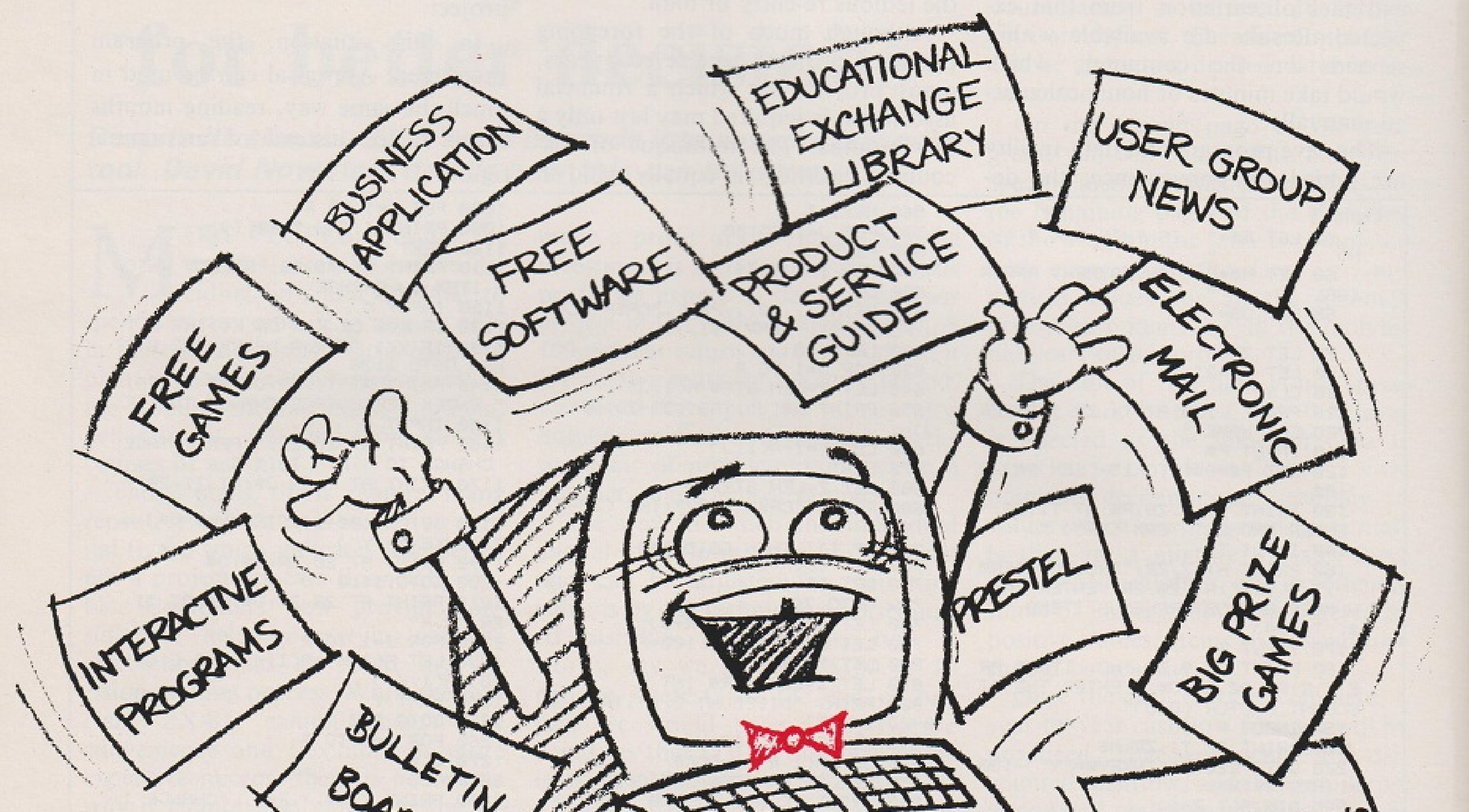
Although much of the foregoing considers projects over several years, many projects for which a financial assessment is required may last only a few months. The technique of discounted cashflow is equally valid in these cases; interest still has to be paid on money borrowed to finance the project.

In this situation, the program Investment Appraisal can be used in much the same way, reading months count factor instead of an annual figure.

Figure 1

```
  50 LET M$="  IS THIS O.K.? (Y/
N)        "
  60 LET N$="   INVESTMENT APPRA
ISAL          "
  65 LET O$="
                 "
  70 LET ZR=0
  80 LET T1=21
  90 CLS
 100 PRINT ,,N$;AT T1,ZR;" ENTER
 PROJECT NAME "
 110 INPUT P$
 120 LET P$=O$( TO 15-(LEN P$)/2
)+P$
 130 PRINT AT 3,ZR;P$;AT T1,ZR;"
 ENTER PROJECT TERM (YRS) "
 140 INPUT T
 160 PRINT AT 7,ZR;"PROJECT TERM
 ";T;" YRS";AT 20,ZR;"ENTER THE
NUMBER OF COST/REVENUE ITEMS ";O
$( TO 20)
 170 INPUT R
 180 PRINT AT 9,ZR;"NO. ITEMS AR
E ";R;AT 20,ZR;O$;" ENTER THE DI
SCOUNT FACTOR (O/O)"
 190 INPUT F
 200 PRINT AT T1,ZR;M$
 220 GOTO 220+5+(INKEY$="Y")-130
*(INKEY$="N")
 225 DIM A(T,R+1)
 230 DIM B$(R,14)
 240 CLS
 250 PRINT ,,N$,,,P$,,,"ENTER TH
E NAMES OF THE COST/     REVENUE
ITEMS",,,
 260 FOR I=1 TO T
 270 PRINT I
 280 INPUT C$
 290 LET B$(I)=C$
 300 PRINT B$(I)
 310 NEXT I
 320 PRINT AT T1,ZR;M$
 330 GOTO 330+10*(INKEY$="N")+70
*(INKEY$="Y")
 340~GOSUB 5000
 350 PRINT AT T1,ZR;"ENTER THE N
EW NAME OF ";X
 360 INPUT X$
 370 PRINT AT 7+X,16;(X$+O$)( TO
 )
 380 LET B$(X)=X$
 390 GOTO 320
 400 FOR I=1 TO T
 410 CLS
 420 PRINT ,,N$,,,P$,,," ENTER D
ATA FOR YEAR ";I,,,
 430 FOR J=1 TO R
 440 PRINT J;" ";B$(J);TAB 16;
 450 INPUT Y
 460 PRINT Y
 470 LET A(I,J)=Y
 490 NEXT J
 500 PRINT AT T1,ZR;M$
 510 GOTO 510+40*(INKEY$="Y")+10
*(INKEY$="N")
 520 GOSUB 5000
 530~GOSUB 5050
 540 GOTO 510
 550 IF I=T THEN PRINT AT T1,ZR;
" DATA ENTRY COMPLETE           "
 560 NEXT I
 570 FOR J=1 TO 50
 580 NEXT J
 590 FOR I=1 TO T
 600 CLS
 610 PRINT ,,N$,,,P$,,,"CASH FLO
W FORECAST, YEAR ";I,,,
 620 LET Y=0
 630 FOR J=1 TO R
 640 LET Z=LEN STR$ A(I,J)
 650 PRINT B$(J);TAB (30-Z);A(I,
J)
 660 LET Y=Y+A(I,J)
 670 NEXT J
 680 LET Z=LEN STR$ Y
 690 PRINT "CASH FLOW";TAB (30-Z
);Y
 700 IF I>1 THEN GOTO 740
 710 LET CDCF=ZR
 720 LET DF=1
 730 GOTO 760
 740 LET CDCF=A(I-1,R+1)
 750 LET DF=DF*100/(100+F)
 800 LET Y=Y*DF
 810 LET Z=LEN STR$ INT Y
 820 PRINT "DISCOUNT C/F";TAB (3
0-Z);INT Y
 830 LET CDCF=CDCF+Y
 840 LET A(I,R+1)=CDCF
 850 LET Z=LEN STR$ INT CDCF
 860 PRINT "C.D.C.F";TAB (30-Z);
INT CDCF
 870 PRINT AT T1,ZR;" cOPY,  OR
newline FOR NEXT PAGE "
 880 GOTO 880+10*(INKEY$="C")+30
*(INKEY$=CHR$ 118)
 890 PRINT AT T1,ZR;O$
 900 COPY
 910 NEXT I
 920 CLS
 930 PRINT ,,N$,,,P$,,,,,"
  OPTIONS:-",,,"2. SAVE ON TAPE
 ",,,"3. MODIFY THE DATA",,,"4.
STOP"
 940 PRINT AT T1,ZR;"    SELECT
1,2,3, OR 4 "
 970 GOTO 970-380*(INKEY$="1")+1
0*(INKEY$="2")+70*(INKEY$="3")+6
0*(INKEY$="4")
 980 CLS
 990 PRINT "SET YOUR TAPE RECORD
ER AS FOR   SAVE. PRESS NEWLINE
WHEN READY."
1000 GOTO 1000+10*(INKEY$=CHR$ 1
18)
1010 SAVE "INVESTMENT"
1020~GOTO 920
1030 STOP
1040 CLS
1045 PRINT ,,N$,,,P$,,,"     MOD
IFY THE DATA",,,"      SELECT ",
,,"1. CHANGE INDIVIDUAL ITEMS",,
,"2. PERCENTAGE CHANGE",,,"3. CH
ANGE DISCOUNT FACTOR",,,"4. RETU
RN"
1050 GOTO 1050+350*(INKEY$="3")-
130*(INKEY$="4")+10*(INKEY$="2")
+210*(INKEY$="1")
1060 CLS
1070 PRINT ,,N$,,,P$,,,"
PERCENTAGE CHANGE"
1080 FOR I=1 TO R
1090 PRINT I;TAB 5;B$(I)
1100 NEXT I
1110 PRINT AT T1,ZR;"SELECT WHIC
H ITEM TO CHANGE"
1120 INPUT K
1130 IF K<1 OR K>R OR K<>INT K T
1130~IF K<1 OR K>R OR K<>INT K T
HEN GOTO 1120
1140 PRINT AT T1,ZR;O$;AT T1,ZR;
" ENTER PERCENTAGE CHANGE TO ";K
1150 INPUT G
1160 PRINT AT 20,ZR;" PERCENTAGE
 CHANGE TO ";K;" IS ";G
1170 PRINT AT T,ZR;O$;AT T1,ZR;M
$
1180 GOTO 1180+10*(INKEY$="N")+4
0*(INKEY$="Y")
1190 PRINT AT 20,ZR;O$;O$
1200 GOTO 1110
1210 PRINT AT 20,ZR;O$;O$;AT T1,
ZR;"O.K"
1220 FOR I=1 TO R
1230 LET A(I,K)=A(I,K)+INT ((G/1
00)*A(I,K))
1240 NEXT I
1250 GOTO 920
1260 FOR I=1 TO T
1270 CLS
1270~CLS
1280 PRINT ,,N$,,,P$,,,"CHANGE I
NDIVIDUAL ITEMS, YR ";I,,,,,
1290 FOR J=1 TO R
1300 PRINT J;" ";B$(J);TAB 16;A(
I,J)
1310 NEXT J
1320 PRINT AT T1,ZR;" ANY CHANGE
S? (Y/N)        "
1330 GOTO 1330+10*(INKEY$="Y")+5
0*(INKEY$="N")
1340 GOSUB 5000
1350 GOSUB 5050
1360 GOTO 1320
1380 NEXT I
1390 GOTO 920
1400 CLS
1410 PRINT ,,N$,,,P$,,,"PREVIOUS
 DISCOUNT FACTOR = ";F;AT T1,ZR;
" ENTER NEW DISCOUNT FACTOR"
1420 INPUT F
1430 PRINT AT T1,ZR;O$;AT 10,ZR;
1440 GOTO 1440-40*(INKEY$="N")+0
*(INKEY$="Y")
1450 GOTO 920
4990 STOP
5000 PRINT AT T1,ZR;"ENTER THE I
TEM TO BE CHANGED    "
5010 INPUT X
5020 IF X>R OR X<1 OR X<>INT X T
HEN GOTO 5010
5030 PRINT AT T1,ZR;"ENTER THE N
EW VALUE OF ";X
5060 INPUT Y
5070 PRINT AT 7+X,16;(STR$ Y+O$)
( TO 14)
5080 LET A(I,X)=Y
5090 PRINT AT T1,ZR;M$
5100 RETURN
9000 SAVE "INVESTMENT"
9010 RUN
```

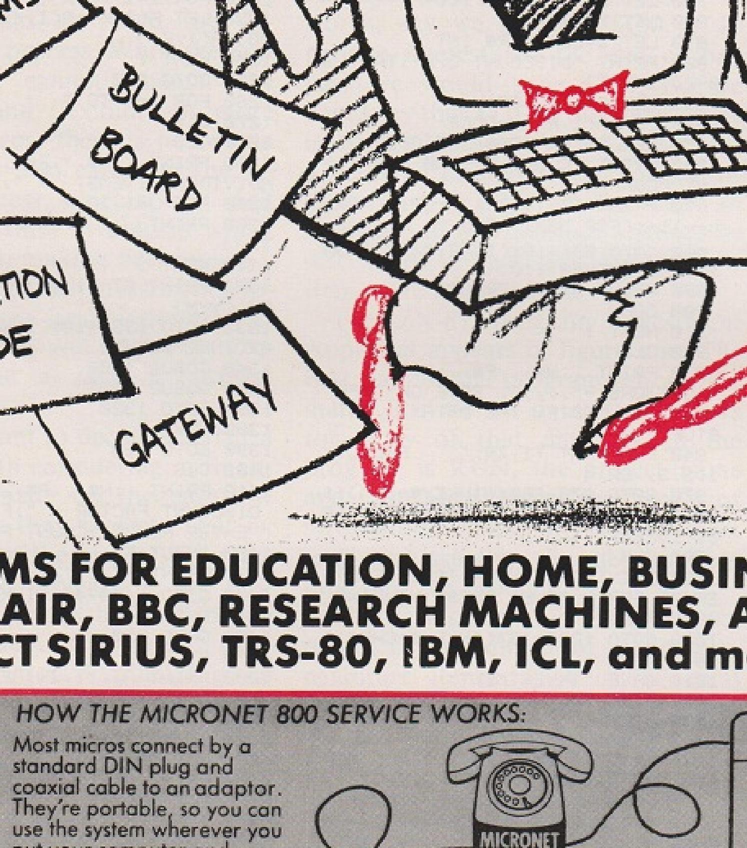
MORE IDEAS THA
MORE PROGRAMS THA
LESS MONEY THA



PROGRAMS FOR EDUCATION, HOME, BUSINESS, AND GAMES – FOR SINCLAIR, BBC, RESEARCH MACHINES, APPLE, COMMODORE, ACT SIRIUS, TRS-80, IBM, ICL, and more each month. . .

HOW THE MICRONET 800 SERVICE WORKS:

Most micros connect by a standard DIN plug and coaxial cable to an adaptor. They're portable, so you can use the system wherever you put your computer and telephone.



Simply dial Micronet's number and enter your subscriber number into the keyboard. A 'welcome to Micronet' message appears on the screen.
You're on-line to one of the world's largest databases.

PRESTEL
MICRONET 800's
thousands of ideas – packed pages – games, education, business software, club news and information.

When you have all the information you need – or you've finished 'downloading' software to your computer's memory – just switch off the telephone link at your keyboard and replace the receiver.

Taking an educated look inside the ZX printer

Sinclair Research's printer is one of the cheapest hard-copy devices on the market. At the moment it can only be used on Sinclair machines—the ZX-81 and the Spectrum—but it would be useful if it could be interfaced with other companies computers. Here John Connor describes how the machine works on the basis of both printed information and guesswork.

THE ZX PRINTER is one of the cheapest hard-copy devices available. It would obviously be useful to be able to interface it to non-Sinclair computers. So it is useful to describe the printer operation and interface. One warning—much of the detail is from the printer manual but there is educated guesswork for some things, especially the pin-outs.

The printer uses thermo-sensitive paper and has two styli on a continuous belt. The belt and paper feed roller are driven by the same motor; the belt, presumably is angled upwards across the paper to compensate for the paper movement. The styli are on opposite sides of the belt, so that while one is printing the other is moving round for the next line. It follows that the printer has no such thing as a 'carriage return'.

The styli generate individual dots on the paper. The position and horizontal width of a dot is controlled entirely by the timing and duration of power applied to the stylus; that, in turn, is entirely software-controlled by the host computer. Thus the printer naturally is suited to graphics but will need some kind of external character generator for text.

The printer interface appears as a standard Z-80 I/O port with an address of FB (hex). Timing diagrams for read and write are in figure one.

Write bit assignments:

D7 (msb): '1' stylus on (print)
'0' stylus off (blank)
D2 : '1' stop motor
'0' start motor
D1 : '1' motor slow
'0' motor normal speed.

Data latched till next write; can produce continuous line by leaving stylus on. D1 and D7 low on power-up or after pressing feed button; D2 high after feed.

Read bit assignments:

D7 : '1' stylus on paper
D6 : '0' printer connected
D0 : Encoder pulses; low to high transition defines start of print column.

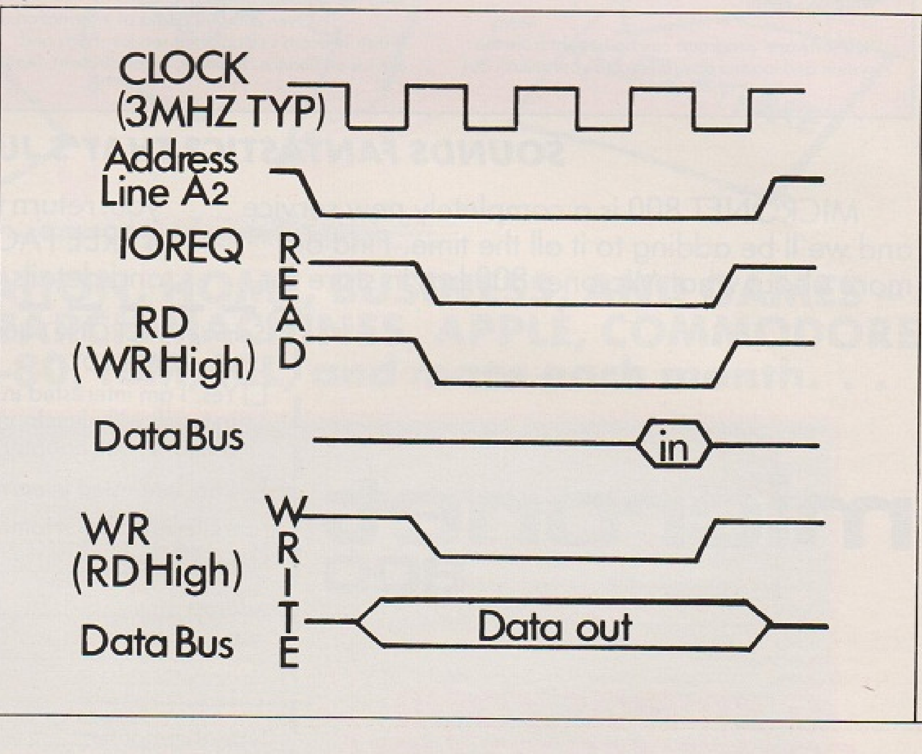
D0 and D7 latched on going high till write to printer re-sets them. For valid timing, follow every detected high with a write. D7 is high whenever stylus is on.

The Sinclair manual says that each pen is on the paper for 32ms at full speed and off it for 16ms. In fact, the individual stylus must be off the paper longer than it is on; i.e., on for 16ms and off for 32ms. Taking that and the stylus inter-lacing into account, D7 will have the timing shown in figure two.

D0 is fed by an ENCODER disc giving 256 pulses across the printable width allows 4mm. margins. The minimum period between successive pulses is 60μs at full speed. They key to using the printer is using D0 pulses to align dots on successive lines. D0 timing is shown in figure two.

There are two basic areas for generating text—generating a dot matrix representation of a line of characters, as for a VDU; and writing the dots to the printer with precise synchronisa-

Figure 1: Printer I/O



tion to preserve the character shapes.

The dot matrix rows could be produced by pure software; by access to the output of the VDU character generator—that seems possible on the BBC machine—or by making a hardware character generator.

A natural representation would be 8 × 8 for each character, giving 32 characters per line; 6 × 8 should also be possible, giving 42 characters per line.

Writing dots to the printer must be done in tight time sync to the encoder translations. Timing jitter will appear as jitter in the horizontal dot position; 30ms will shift a dot by half a dot column, so it cannot use interrupts except in a 'wait for interrupt' loop. The basic algorithm for a row of dots will be:

Detect stylus on paper (D7 high); detect encoder pulse (D0 high); write high (print) or low (no print) to stylus—this also re-sets D0 and D7 ready for next translation; write low to stylus; go back to detect encoder pulse until there have been 256 pulses; look for stylus off paper (D7 low); get set up next row—if last two rows before stopping set motor to slow (D1 high) so that stylus will stop off paper when printer is stopped; if not, last row go to detect stylus, or stop.

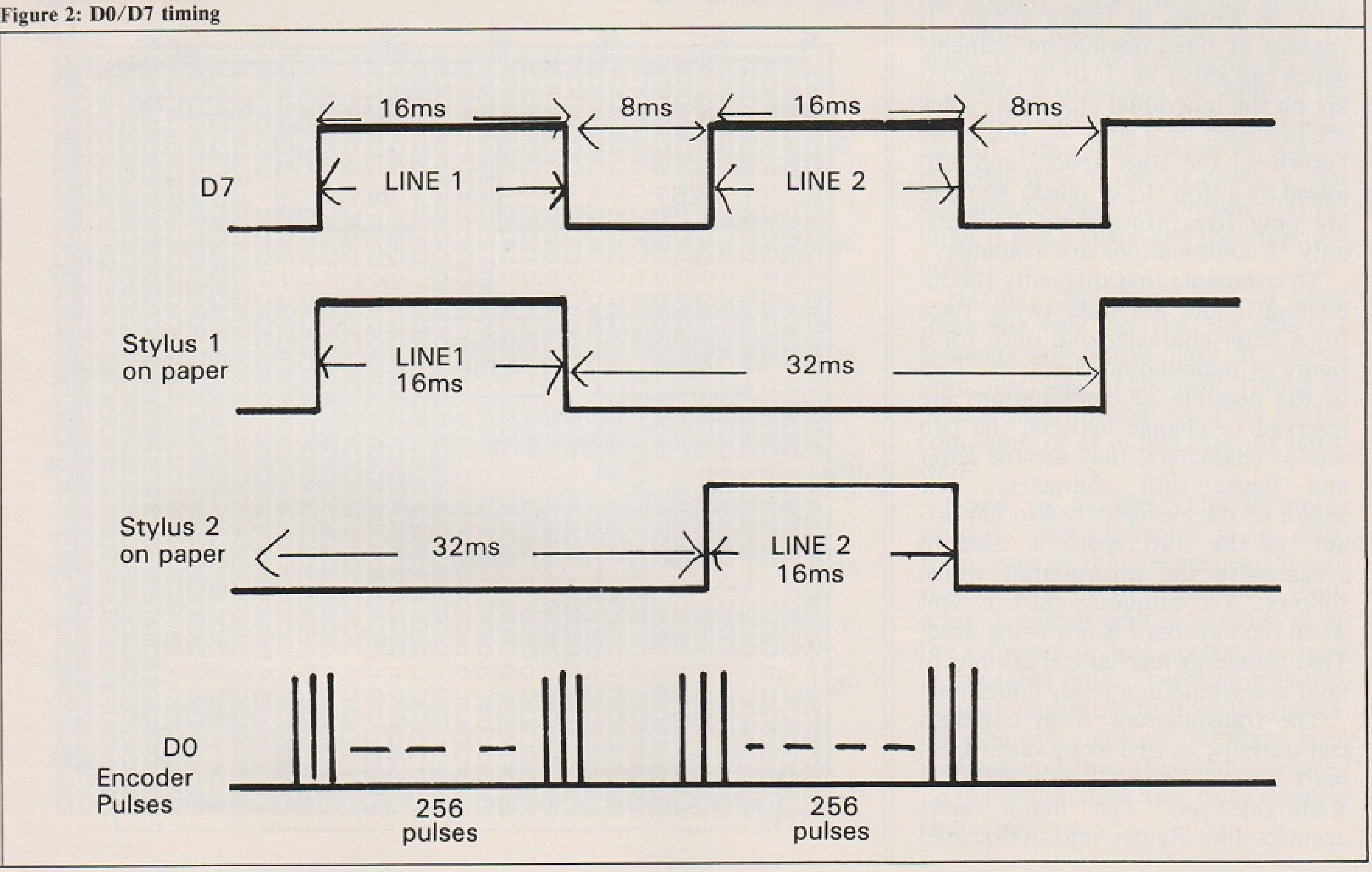
The first two can be done with polling loops. The loop duration will produce its own jitter relative to the asynchronous transitions on D0 and D7, so the loops should be as short as possible. The third and fourth are best done using the same instructions for dot and no dot, i.e., no branches. You may need NOPs between those to to ensure the stylus is on for long enough.

If you do not have a Z-80 system you will have to arrange to generate the signals shown in figure one, either using your existing bus signals plus logic or by programming a PIA or equivalent port.

PINOUT

Pin	Top	Bottom	Pin	Top	Bottom	Pin	Top	Bottom
1	D7	5V pos.	9	D3	A2	19	—	A7
2	—	9V pos.	10	D4	A3	20	—	A6
3	SLOT	—	11-14	—	—	21	—	A5
4	D0	OV	15	$\overline{\text{IOREQ}}$	—	22	—	A4
5	D1	OV	16	$\overline{\text{RD}}$	—	23	—	—
6	D2	CLOCK	17	$\overline{\text{WR}}$	—			
7	D6	A0	18	—	—			
8	D5	A1						

Figure 2: D0/D7 timing



Getting the latest news by ZX-81

To permit reception of radio teletype signals on a ZX-81 four separate problems have to be overcome. In this article John Eggleton gives, in detail, the instructions needed to find the solutions and build an interface which allows the ZX-81 to translate the Murray code used by teleprinters with the television replacing the normal paper.

THE SYSTEM still in use to transmit printed information telegraphically was invented in 1874 by Emile Baudot. At each end of a link is a teleprinter, capable of converting the key depressions into a code. That code is then converted into audio tones and transmitted by radio to the distant terminal, where the reverse process takes place.

It is important to understand the formation of the Murray code character as shown in figure eleven. It consists of five information elements which can either be '1' or '0', depending on the individual character. They are preceded by a start '0', otherwise known as the start space, and followed by a stop '1', or mark. As there are only five information elements, only 32 combinations are available.

To overcome that difficulty the individual codes are used twice, once for a letter character and once for a figure or punctuation character. Two of the possible 32 combinations are reserved to change between the two sets of characters; they are the letter and figure shift characters. The length of the elements is also important, as the start space is used to synchronise the information which follows. A continuous mark is sent when the keyboard is not being used. That is used by mechanical printers to hold everything in a ready condition.

The transmissions from commercial stations, as well as by radio amateurs are known as RTTY—RadioTeleType. The major news agencies like Reuter and Associated Press, broadcast almost continually using this form of transmission. Those sources can be monitored easily using the hardware and program described in this article, with radio amateurs being found easily around 14,080kHz during most of the day. The RTTY transmission produces its own particular sound, consisting of two audio frequencies alternating

Figure 1: Hex loader.

```
10   REM (170 characters long)
20   FOR I=16528 TO 16684
30   INPUT H$
40   LET D=CODE H$ (1)-28
50   FOR J=2 TO LEN H$
60   LET D=16*D+CODE H$ (1)-28
70   NEXT J
80   POKE I,D
90   NEXT I
```

Figure 2: Checking program.

```
15   DIM H$ (2)
20   FOR I=16514 to 16684
30   LET C=PEEK I
40   LET D1=C
50   FOR J=2 TO 1 STEP -1
60   LET D2=INT (D1/16)
70   LET H$ (J)=
     CHR$ (D1-16*D2+28)
80   LET D1=D2
90   NEXT J
100  PRINT I; "="; H$ (1); H$ (2)
110  FOR Z=1 TO 100
120  NEXT Z
130  SCROLL
140  NEXT I
```

Figure 3: Top of PCB.

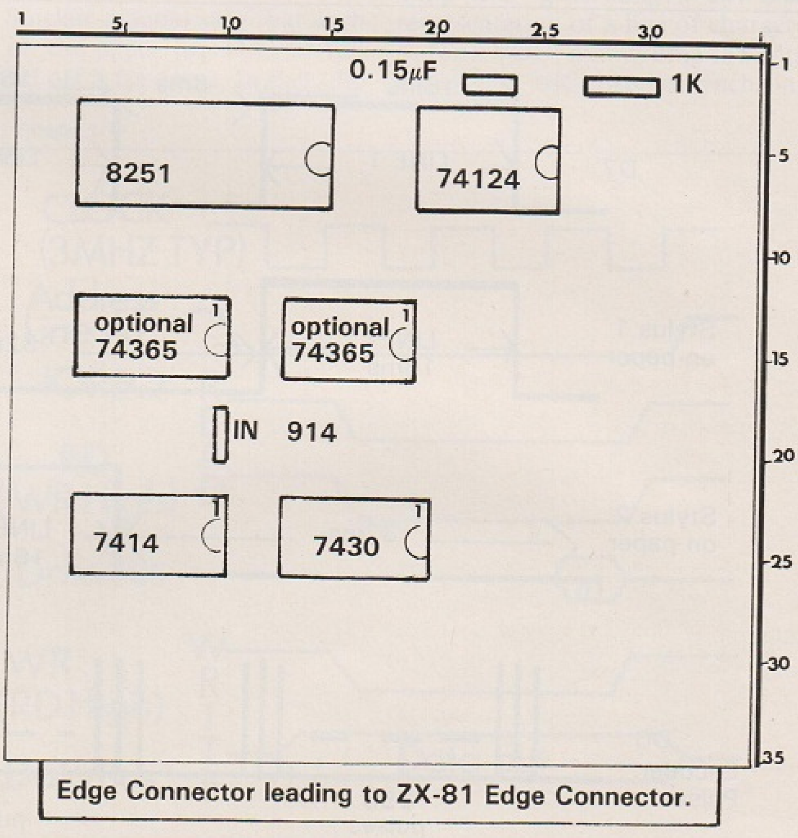


Figure 5:

	7	6	5	4	3	2	1	0	Hex	Port
Sinclair	1	1	1	1	1	0	1	0	FB	Printer
I/O	1	1	1	1	1	1	0	1	FD	Fast mode
Ports	1	1	1	1	1	1	1	0	FE	Slow mode
	1	1	1	1	1	1	1	1	FF	Tape
8251	0	1	1	0	0	0	1	1	63	Data
Port	0	1	1	1	0	0	1	1	73	Control

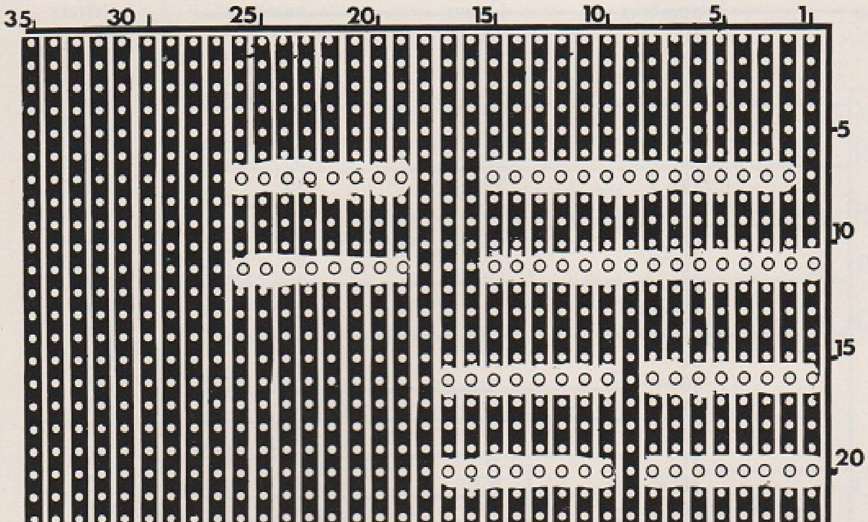
With any extra port, bits 0 and 1 must both be logical '1' because only partial decoding is used for the slow and fast modes. Likewise, bits 2 and 3 should be logical '0' to avoid conflict with the printer and tape ports.

continuously while characters are sent and a single tone during conditions of no information.

Nowadays mechanical printers have been replaced by VDUs and some systems have dispensed with Murray code and use ASCII instead. The system which is described uses the ZX-81 1K to translate Murray code with the TV replacing the teleprinter paper, therefore making it much quieter. It is now possible to have the news direct from the correspondents in the living room using a ZX-81. Figure seven shows a block diagram of the complete system.

To permit reception of RTTY signals on a ZX-81 or any other computer, four separate problems have to be overcome.

Figure 4: Veroboard.



Audio to logic level conversion. The RTTY signal output at the earphone jack of a receiver consists of two tones. In the amateur case the mark tone is 1,275kHz and the space tone is 1,445kHz. Unfortunately a computer cannot distinguish between the tones and so conversion is necessary.

The circuit needed to perform the conversion consists usually of a filter which ensures that only the two tones are used, followed by a detector circuit. The detector in most cases is a Phase Locked Loop (PLL) with the frequency of operation adjusted to one of the two frequencies in use.

The demodulated output of the circuit is then high or low depending on whether the PLL is in lock—i.e., frequency in equals frequency of PLL—or out of lock—i.e., frequency in equals other frequency in use. A number of circuits are available to achieve this, one of which can be found in the Teleprinter Handbook published by the Radio Society of Great Britain.

Start of character detection. While the sending keyboard is not active a continous mark is sent and so to detect the start of a character all that is needed is to sample the input to the computer looking for a space or logical 'O'. Once that change of state is detected it is usual to check a few milliseconds later to confirm that the input is still a space and therefore not noise.

Having confirmed that a start space has been received, the next five elements will arrive later at known time intervals. Therefore timing is needed and this can be done by the computer, or as in this case, by hardware.

Serial to parallel. The incoming Murray code character is in serial form; each of the five elements is sent sequentially. Inside a computer, characters are dealt with in parallel, as all the elements or bits needed are used at the same time. Therefore it is necessary temporarily to store the incoming elements until all five have been received and then they can be placed on the system data lines as a complete character.

RADIO TELEPRINTER

There is another difficulty, for instance that the letter 'a' is not the same in Murray code as it is in ZX-81 code and so translation is required.

Display results. Once the ZX-81 character has been formed from the original Murray one, all that remains to be done is to call the appropriate routine in the ZX-81 ROM. The display should also be examined to place NEWLINEs where appropriate and scroll the display when the screen is full. That is because a teleprinter page is about 60 characters wide compared to the 32 of the ZX-81.

To deal with these four problem areas, a mixture of hardware and software is utilised which can again be split into three areas.

Terminal unit. This is the device which changes the incoming tones into logic levels. Any design on the market can be used so long as it produces logic levels at the output. Two such circuits can be found on page 93 of June 1981 *Personal Computer World* or on page 62 of September, 1979 *Wireless World*. The lead from the device to the computer should be kept reasonably short to avoid noise being induced into the circuit.

8251 port. The heart of the solution is the chip and interfacing to the ZX-81 is minimal. The 8251 is a Universal Asynchronous Receiver Transmitter—UART—and is very flexible, having the added advantage of requiring only +5V to power it. As its name suggests, it is capable of transmitting as well as receiving and therefore with some addition to the hardware and software it will be possible to transmit RTTY from the ZX-81 keyboard.

The 8251 is a programmable device and as such requires two words to set it up; the machine code initialisation routine achieves this. The mode word is set up for one-and-a-half stop bits, no parity, five-bit character length and 64 times baud rate.

The baud rate indicates the speed of the incoming information and in this case the hardware clock input on TxC pin 9 and RxC pin 25 is 64 times the chosen input speed. In the case of amateur transmissions, the baud rate is 45.5 and commercial ones are 50 and 75. The clock produced by the 74LS124 is varied by means of a 20k multi-turn potentiometer to enable various speeds of transmission to be copied.

The command word is set for error re-set and receive enable, all other options being off. Once the 8251 has been programmed it remains so until powered down or it is re-set.

Figure 6:

Numbers	Letters	Murray code	Hexadecimal ZX-81 numbers	Hexadecimal ZX-81 letters	Decimal ZX-81 numbers	Decimal ZX-81 letters
3	E	01	1F	2A	31	42
L/F	L/F	02	EE	EE	238	238
—	A	03	16	26	22	38
SPACE	SPACE	04	00	00	0	0
"	S	05	0B	38	11	56
8	I	06	24	2E	36	46
7	U	07	23	3A	35	58
C/R	C/R	08	76	76	118	118
WRU	D	09	EE	29	238	41
4	R	0A	20	37	32	55
BELL	J	0B	EE	2F	238	47
,	N	0C	1A	33	26	51
%	F	0D	EE	2B	238	43
:	C	0E	0E	28	14	40
(	K	0F	10	30	16	48
5	T	10	21	39	33	57
none	Z	11	EE	3F	238	63
)	L	12	11	31	17	49
2	W	13	1E	3C	30	60
£	H	14	0C	2D	12	45
6	Y	15	22	3E	34	62
0	P	16	1C	35	28	53
1	Q	17	1D	36	29	54
9	O	18	25	34	37	52
?	B	19	0F	27	15	39
@	G	1A	EE	2C	238	44
FIGS	FIGS	1B	EE	EE	238	238
.	M	1C	1B	32	27	50
/	X	1D	18	3D	24	61
=	V	1E	14	3B	20	59
LETTERS	LETTERS	1F	EE	EE	238	238

EE = code for no direct ZX-81 equivalent.

As the other I/O ports of the ZX-81 are not fully-decoded it is important to ensure that there is no conflict of addresses, and figure five shows how that is achieved. The 8251 requires two I/O ports to operate, one for control and the other for normal data transfer.

The chip is enabled by $\overline{CS}$ pin 11 when the 74LS30 output goes low, which corresponds to an address of either 63H or 73H. A4, which determines whether it is 63H or 73H, is connected to C/$\overline{D}$ pin 12 and that puts the 8251 into the control or data mode.

The data bus is connected directly to ZX-81 data lines, as are the $\overline{RD}$ & $\overline{WR}$ lines. The re-set has to be inverted as the 8251 expects that line to remain low; that is done by part of the 74LS14. The clock line is connected to Ø of the ZX-81; that is because that 8251 is a dynamic device. A gate on the 74LS14 is used to buffer the 8251 from the terminal unit and has a limiting diode on its input.

All the other pins are not used and not connected. The circuit can be built on Veroboard and the supply taken from the +5V on the ZX-81 edge connector.

Driving program. The programming necessary to receive RTTY by way of the 8251 is written in machine code and a listing is given in figure nine. Figures one and two give the only Basic programming needed. One is in the form of a hexadecimal loader and figure two can be used to check that the machine code has been entered correctly.

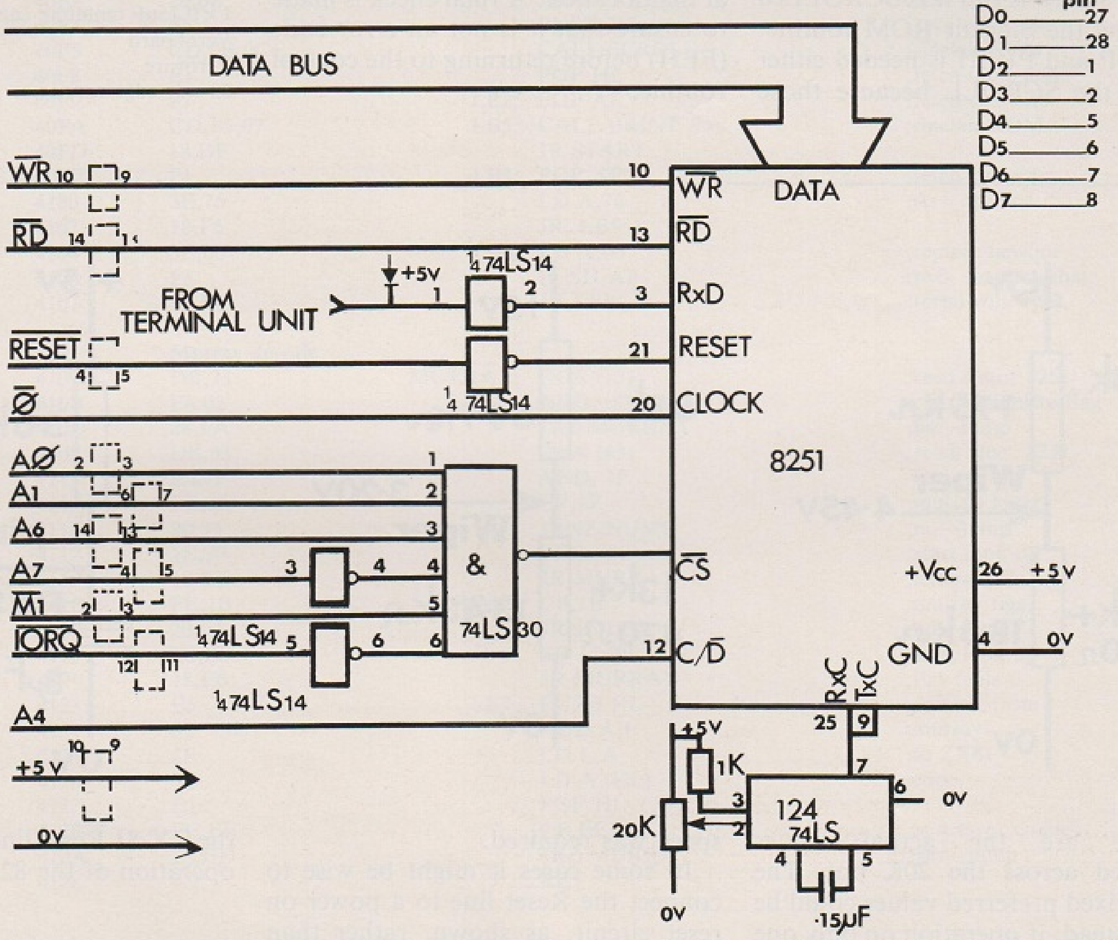
Once the machine code has been POKEd into the REM statement in line 1 and it has been checked it should then be SAVEd.

Machine code programming is needed for two reasons. First, the 8251 is not on the ZX-81 memory map and so it is not possible to send or receive instructions by PEEK and POKE commands. The second reason is that a 50 baud character takes a total time of 150msec to receive, which is too fast for Basic.

If it were used, characters would be lost while the ROM Basic interpreter was still determining what the next line number should be. The whole machine code program is re-locatable except for the underlined values which require altering if the program is placed elsewhere in memory. The program is split into four seperate areas:

Look-up table. The first 14 bytes of the REM statement are not used by the program but are there so that the look-up table starts at a round hexadecimal figure. If the table is placed elsewhere in memory, say in EPROM in the 8–16K area, it is essential that

Figure 7: Block diagram.



RADIO TELEPRINTER

the top pair of the hexadecimal address of the look-up table remain the same throughout the table—40H in this case. That is because the value is stored in the H register and is used together with the L register to form an address. The relationship between address, Murray code and ZX-81 characters is given in figure five.

Initialise. This routine sets up the 8251 for use by sending the two words AYSNC and READIN to the control port (73H). It also ensures that the ZX-81 is in the slow mode and it puts the look-up table address into the HL register pair.

Control routine. This collects in the A register a ZX-81 character from the Murray routine and checks to see if it is a NEWLINE. Then it checks the character position on the screen, first by column and then by line, taking action as required. To fetch the information use is made of the fact that the IY register in the ZX-81 is always 4000H, which is the start of RAM and the variable area in particular.

If the screen is full it is SCROLLed by calling the Sinclair ROM routine. The POP and PUSH is needed either side of the SCROLL because those registers are used in this routine.

Murray decode. This routine overcomes the three problems of detecting when a character is ready, remembering whether a figure or letter shift has been sent and, finally, conversion between Murray and ZX-81 codes. When a character has been assembled inside the 8251 it signals the program by raising bit 1 of the status word to a high. All that is then needed is to loop continuously reading the status word, inspecting that flag until it is set; at that point the program moves on to read in the assembled Murray code character. The character is then checked to see if it is the letter shift (1FH) or figure shift (1BH).

If it is one of those, the appropriate value is loaded into the L register, which contains the lower part of the look-up table address. If it is not, conversion from Murray code value to ZX-81 is performed by adding the Murray code value to the sum as a pointer to a position in the look-up table and extracting the value found at that location. A final check is made to ensure that it is not an error code (EEH) before returning to the control routine.

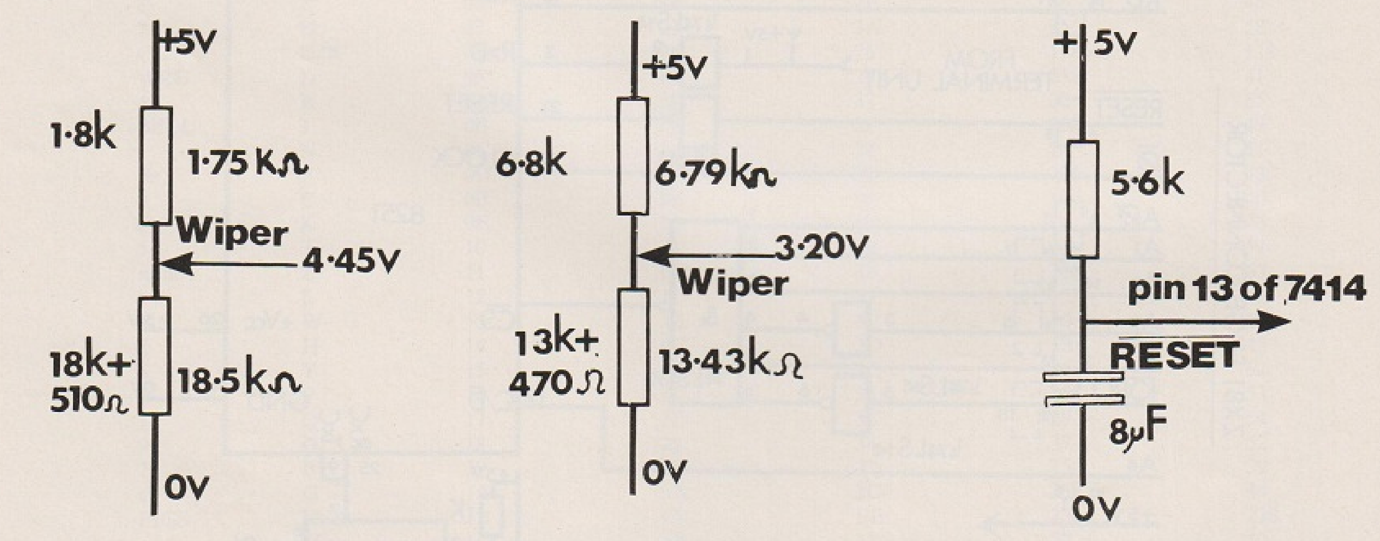
To run the program all that is required is the direct command RAND USR 16592. Control will then be passed to the machine code and the only way to return to Basic is to pull out the power plug. Ensure therefore that a copy of the program is on tape before running. The screen will then clear and random characters appear at the top left, running across the screen, if pin 1 of the 74LS14 is continuously and rapidly taken to OV, thereby simulating a Murray character.

Set the 20K multi-turn potentiometer, which should be of the clockface

Components

1 × 8251
1 × 74LS30
1 × 74LS14
1 × 74LS124
1 × 1k¼W5% resistor
1 × 20k multi-turn variable resistor
1 × 1N914 diode
1 × 0.15uF tantalum capacitor
Veroboard
Verowire
23-way edge connector

Figure 8:



These are the actual values measured across the 20K pot. The ringed fixed preferred values could be used instead, if operation on only one speed was required.

In some cases it might be wise to connect the $\overline{\text{Reset}}$ line to a power-on reset circuit, as shown, rather than the ZX-81 $\overline{\text{Reset}}$ line to ensure correct operation of the 8251.

Figure 9: Machine code program.

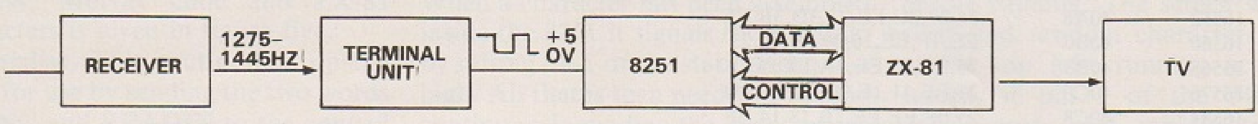
Dec	Hex	Data		Assembler	Comment
16514	4082	00,00,00,00,00,00			
		00,00,00,00,00,00,00			
16523	4090	EE,2A,EE,26,00,38,2E,3A			;look up table
16536	4098	76,29,37,2F,33,2B,28,30			
16544	40A0	39,3F,31,3C,2D,3E,35,36			
16552	40A8	34,27,2C,EE,32,3D,3B,EE			
16560	40B0	EE,1F,EE,16,00,0B,24,23			
16568	40B8	76,EE,20,EE,1A,EE,0E,10			
16576	40C0	21,EE,11,1E,0C,22,1C,1D			
16584	40C8	25,0F,EE,EE,1B,18,14,EE			
		Initialise			
16592	40D0	3E,83		LD A, AYSNC	;mode for 8251
16594	40D2	D3,73		OUT A (73)	
16596	40D4	3E,14		LD A, READIN	;command 8251
16598	40D6	D3,73		OUT A (73)	
16600	40D8	CD,28,0F		CALL SLOW	;sinclair ROM
16603	40DB	21,90,40		LD HL,4090	;HL : = look up add.
		Control routine			
16606	40DE	CD,09,41	START:	CALL MURRAY	;fetch character
16609	40E1	FE,76		CP,76	;is it newline?
16611	40E3	28,1F		JRZ,LB4	;yes—jump
16613	40E5	F5		PUSH AF	;save character
16614	40E6	FD,7E,39		LD A (IY + 39)	;A: = col count
16617	40E9	FE,03		CP,03	;end of line?
16619	40EB	20,0C		JRNZ, LB2	;yes—jump
16621	40ED	FD,7E,3A	LB1:	LD A (IY + 3A)	;A: = line count
16624	40F0	FE,03		CP,03	;bottom line?
16626	40F2	20,0B		JRNZ, LB3	;yes—jump
16628	40F4	E5		PUSH HL	;save table add.
16629	40F5	CD,0E,0C		CALL SCROLL	;sinclair ROM
16632	40F8	E1		POP HL	;fetch table add.
16633	40F9	F1	LB2:	POP AF	;fetch character
16634	40FA	CD,F5,07	LB5:	CALL PRINT	;sinclair ROM
16637	40FD	18,DF		JR START	
16639	40FF	F1	LB3:	POP AF	;fetch character
16640	4100	3E,76		LD A,76	;A: = newline
16642	4102	18,F6		JR, LB5	
16644	4104	3E,00	LB4:	LD A,00	;replace newline
16646	4106	F5		PUSH AF	;with space so that
16647	4107	18,E4		JR LB1	;scroll will work
		Murray decode			
16649	4109	DB,73	MURRAY:	IN A (73)	;read status 8251
16651	410B	E6,02		AND, 02	;char. complete flag
16653	410D	28,FA		JRZ MURRAY	;no—jump
16655	410F	DB,63		IN A (63)	;fetch char. 8251
16657	4111	E6,1F		AND, 1F	
16659	4113	FE,1F		CP,1F	;murray letter?
16661	4115	20,04		JRNZ NUMB	;no—jump
16663	4117	2E,90		LD L, 90	;start look up
16665	4119	18,EE		JR MURRAY	;letters table
16667	411B	FE,1B	NUMB:	CP,1B	;murray figs?
16669	411D	20,04		JRNZ LET	;no—jump
16671	411F	2E,B0		LD L, B0	;start look up
16673	4121	18,E6		JR MURRAY	;figs table
16675	4123	E5	LET:	PUSH HL	;convert from
16676	4124	85		ADD A,L	;murray
16677	4125	6F		LD L,A	;to ZX81
16678	4126	7E		LD A (HL)	;code.
16679	4127	E1		POP HL	
16680	4128	FE,EE		CP,EE	;is it error codde?
16682	412A	28,DD		JRZ MURRAY	;yes—jump
16684	412C	C9		RET	

RADIO TELEPRINTER

type, to 3.30 for 45.5/50 baud or 1.00 for 75 baud transmissions. Then connect the terminal unit, connect to a receiver and tune into a RTTY broadcast.

Some practice is necessary for the proper balance but if an oscilloscope is available, connect it to the output of the terminal unit and tune for an equal mark/space ratio. If the display is still unintelligible, switch to the other sideband on the receiver and try again. After a short time it should be possible to tune the signal by ear and dispense with the oscilloscope.

Figure 10:



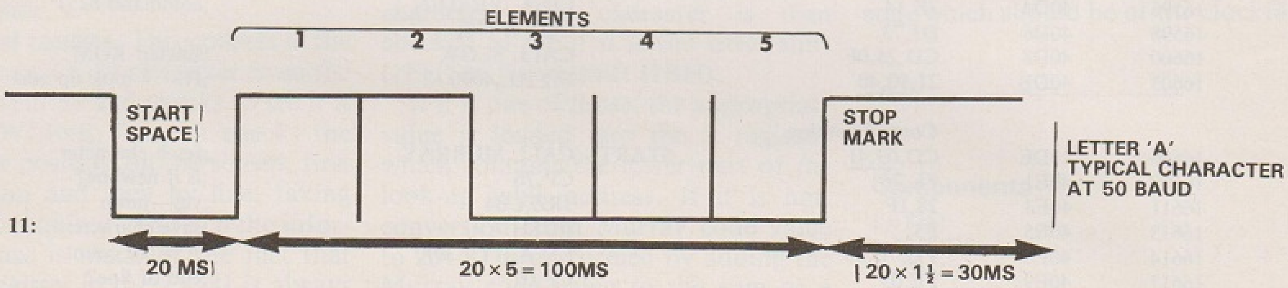


Figure 11:

Spectrum console tidies vast numbers of annoying trailing leads

When Bruce Binder received his Spectrum he was delighted by its capabilities but found the number of loose wires a problem. Here he details his solution by building a special unit to house his machine and the other items he hopes to add to it.

WHEN I received my Spectrum I was delighted with it once I had it set up but like so many of the micros there are always wires trailing all over the place. I decided to design a control console to house it and all the other items I hope to add to it. First, let me say that this project will not cost a fortune. In fact, most of the timber was offcuts from my workshop. The measurements may have to be adjusted to suit the dimensions of the tape deck you are using.

I started with a piece of ¾in. blockboard 27¼in. × 11¾in. Looking down on the board I marked the location of the power pack in the top right-hand side. A hole 2½in. × 3¾in. is cut with the sides of the hole sloping inwards so that the power pack fits in like a wedge. The hole should be 1½in. down and 2½in. in from the edge. Next to that I fitted a small terminal block to handle all the power cables—see circuit diagram.

The next thing is to construct the first of the hinge-up panel units; the angle of slope of the panel is governed by the height of the ZX power pack. I found 3in. sufficient but if your tape unit is taller you will have to adjust accordingly if you want the line of the console to follow through.

Once that has been decided, cut out three triangular pieces from ¾in. blockboard—see Figure three. One of them will have two holes cut in it to accommodate the input/output feeds from the tape unit and a bigger hole for the mains feed to the tape unit. Once again you may have to adjust to suit your tape deck. Next is the back piece. It is of ½in. ply, 3in. × 18in. I cut one large hole of 1in. and a small one for the mains cable—I used a curly cable from Habitat. Then pin and glue the two end-pieces to the back and check for square. At the front I used a piece ¾in. × ¼in. by 18in. long with pin and glue put to one side to stick.

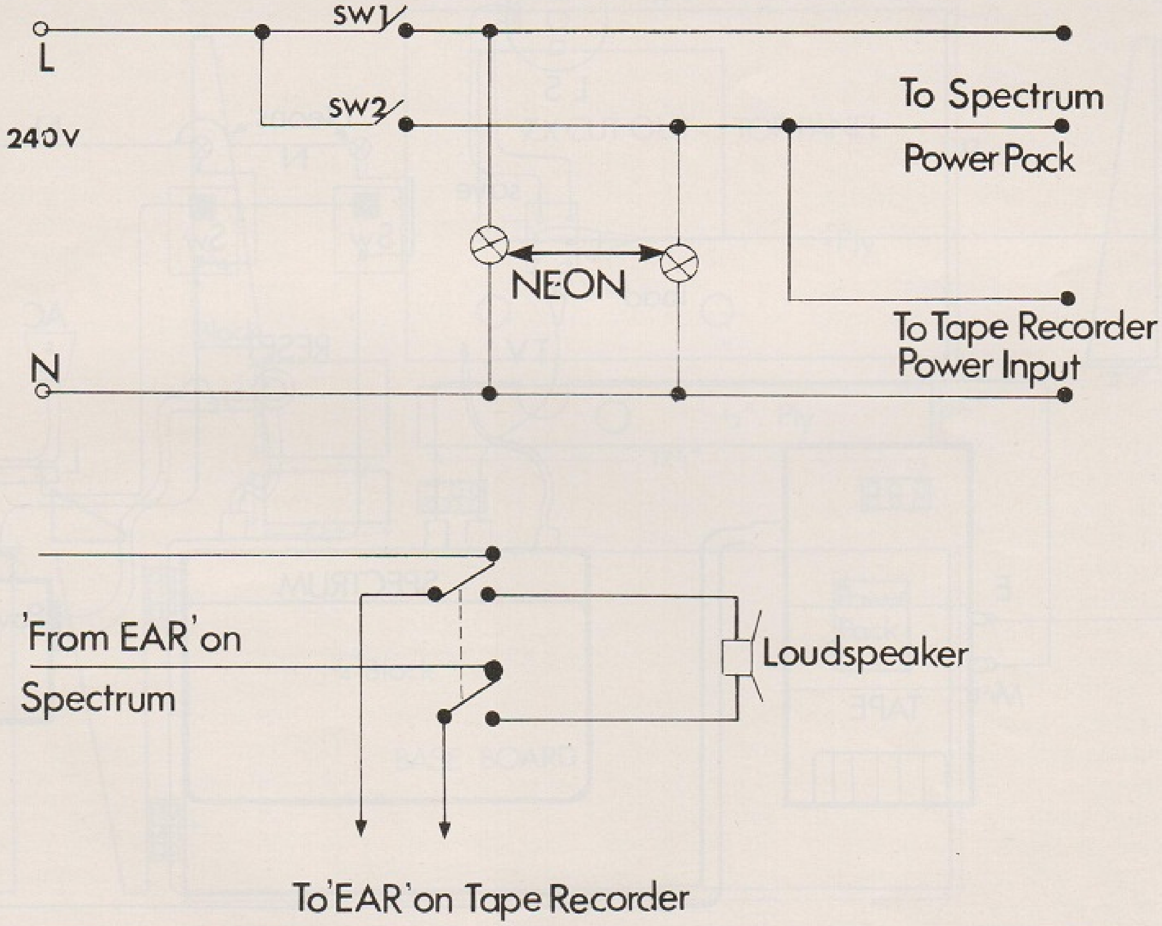
For the top I used ¼in. ply. Cut to size approximately 19¼in. × 12in. and on it mark the position of the Spectrum. If you look on the underside of the Spectrum you will see a lip running down each side. Cut the hole to that width, about 8¾in., then it will fit neatly. At the top cut the hole a good 1½in. bigger than the Spectrum so that all the plugs have plenty of room. To the right and left of that I drilled two 1in. holes for power cables. Once all holes are cut and you have tried the Spectrum for size, pin and glue into place. Then place the completed unit on the base board so that it lines up with the front, back and right-hand side. On the back you can mark the positions for the hinges and screw into place.

The top switch panel took a little time to make because of all the funny angles. I did not want it to look just like a box on top, so I made the front panel slope back by ¾in. at the top and the whole unit tapers down to the back. The construction is much the same as the bottom unit—two end-pieces, a top, and a back.

The front panel is a little more involved. As most panel-mounted switches are designed only to go through thin-gauge metal and will not go through ¼in. ply, I had to cut out the ply to the correct size and a piece of Formica to match. Then I worked-out the positions of all the switches but before sticking the Formica to the ply I cut the holes in the ply big enough for the switches, body and all to go through, then drilled the fixing

SPECTRUM CONSOLE

Figure 1:



SPECTRUM CONSOLE

holes for the switches in the Formica.

When they are stuck together you have the thin panel to take the switches but supported by the ply for strength. The switches can be bought from any good electronics shop. Those required are a double-pole change-over one-off, two single-pole on-off for tape and computer power, two neons (240V), one press to break push-button, a $1\frac{1}{2}$in. 8ohm speaker and one terminal block.

The front panel is screwed to two strips of wood $\frac{1}{4}$in. × $\frac{1}{4}$in. glued at an angle inside the top unit. When you have completed the construction work of the top unit, place it on top of the bottom unit so that it lines up with the left-hand side and back. Then mark the second set of hinge positions but do not fit hinges yet, as the thickness of the Formica on the bottom unit will make a difference.

When all glue is dry, sand ready for laminating. If laminating is not your forte you could use a Fablon material. Do not laminate the back of the unit—it only makes it more difficult to cut holes later. It is best painted matt black. When you have fitted the hinges on the top unit cut a slot between them to allow the TV cable to pass through.

All that is then left to do on the top unit is to drill the speaker holes in the left-hand end-panel. That is best done by putting masking tape over the Formica. Draw two circles, mark the holes and drill. The tape prevents the drill slipping on the Formica.

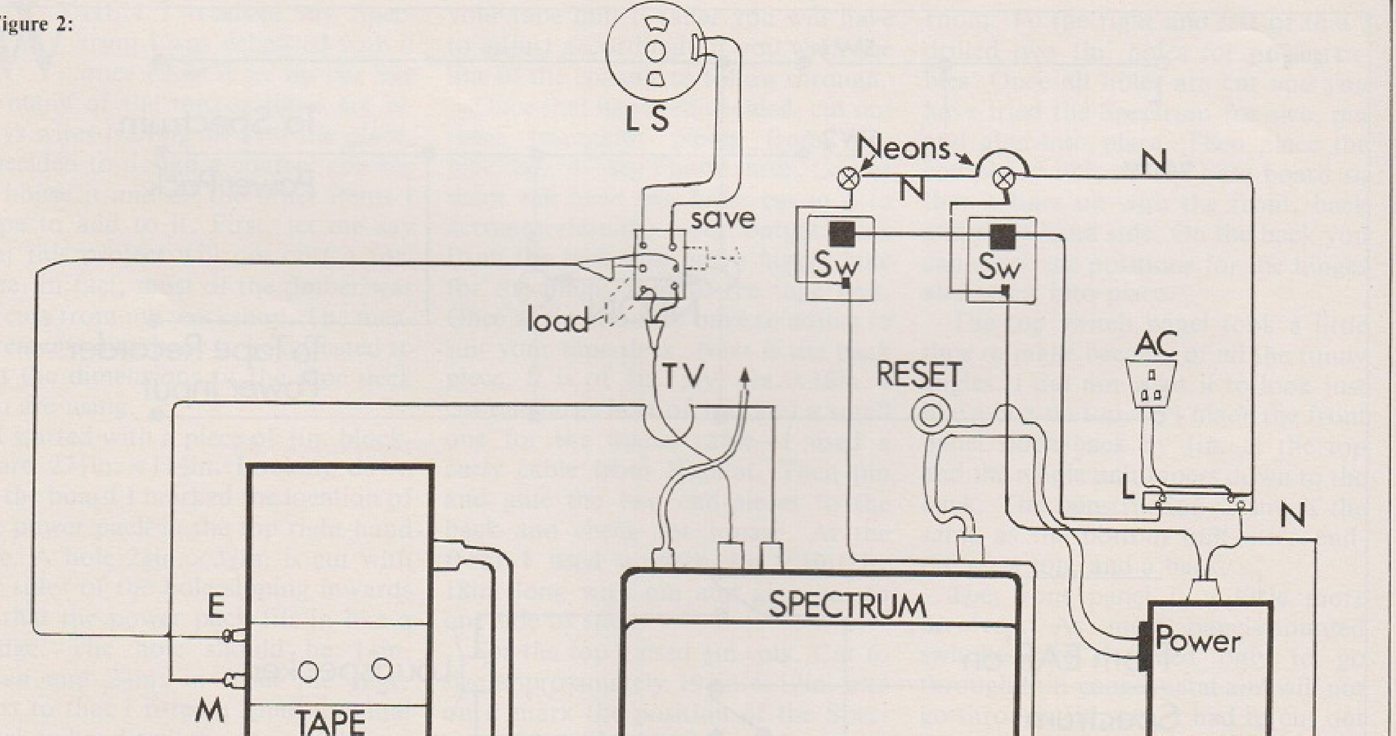
The tape unit I used had the output on the left-hand side and the power feed on the right, so putting the tape unit tight to the left-hand side of the bottom panel left me with about 2in. on the left, just sufficient to take the third of the triangular pieces with space for the jack plugs.

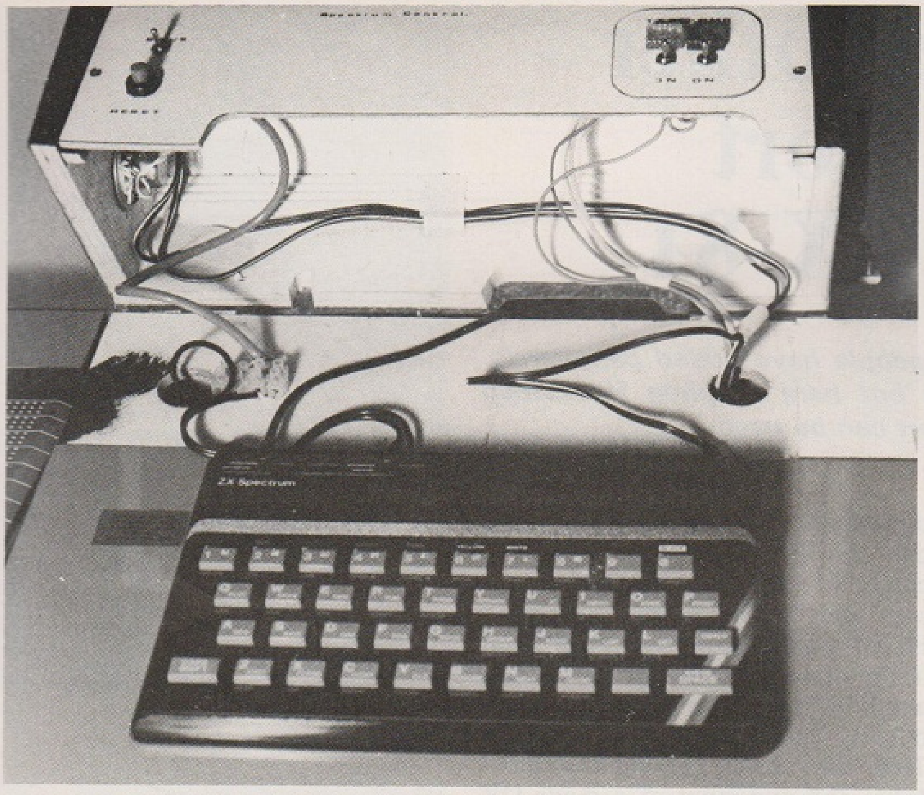
To make it easy to remove the tape unit the whole of the end section hinges down sideways. I will not go into much detail for that part of the console, as it all depends on the tape unit used.

The wiring is reasonably straightforward. Starting with the power feeds, the curly mains cable is passed through the lower back panel and strapped down; the live and neutral are fed in to the terminal block. From the other side of the live terminal a feed is taken to the poles of the two on/off switches. The other side of the switches are run back to the next two spare terminals on the terminal block.

Cut down the cables on the ZX power pack and tape unit mains lead and attach them to the same two terminals. Run the two neutrals into

Figure 2:



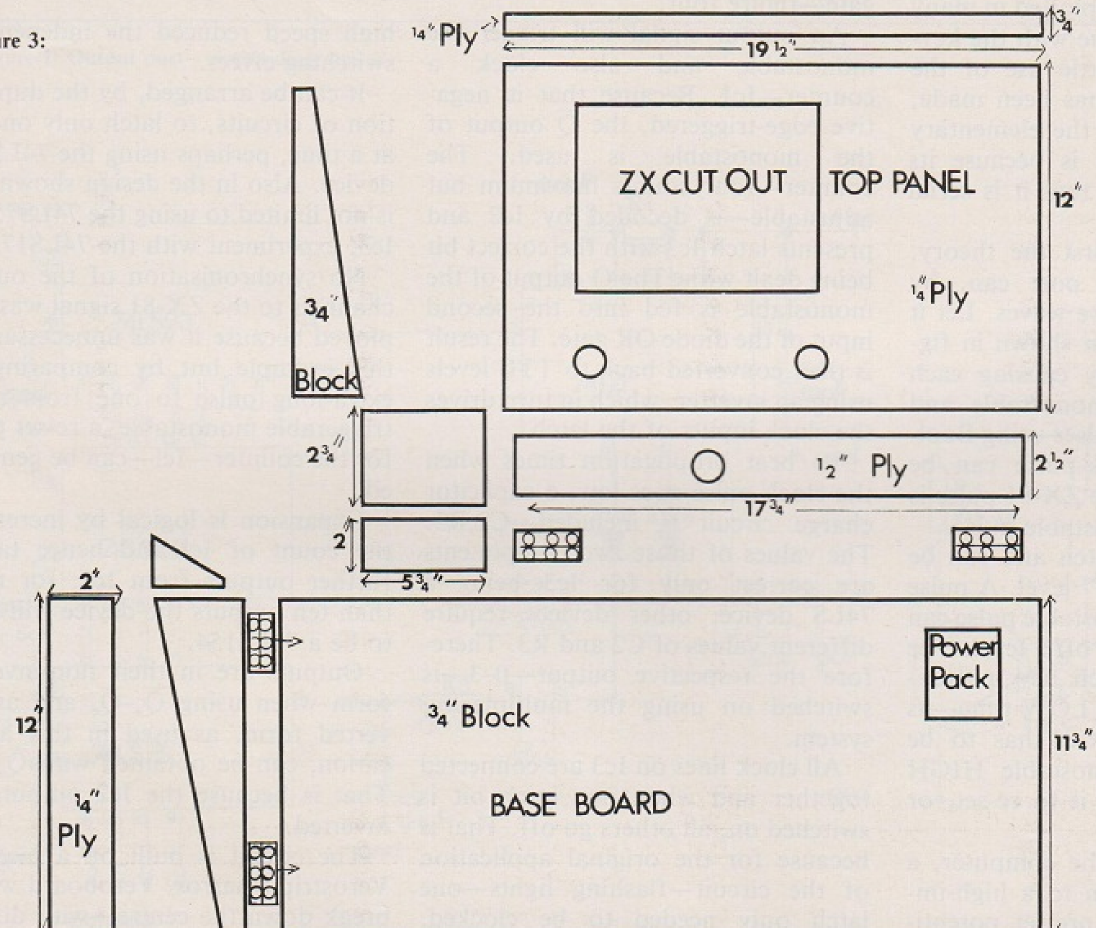


the incoming neutral terminal. Take two other wires from the output side of the switches and link to the neons, then a lead from the other side of the neons to neutral. That completes the power side of things.

In the re-set button, take the low-voltage output lead from the ZX power pack. Cut off one of them and solder it across the press button, so that if the computer is completely hung-up, pressing that button momentarily cuts off all power to the Spectrum.

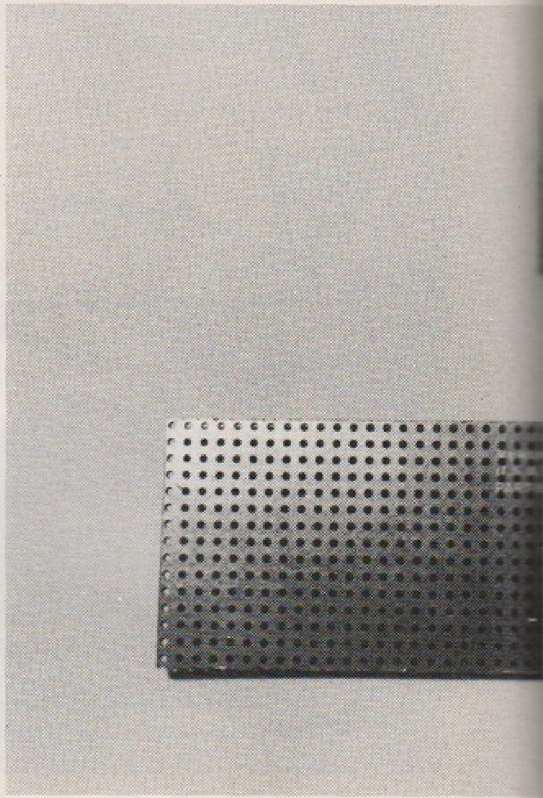
When you are saving a program on tape it is necessary to pull out the earphone plug from the Spectrum because when the tape unit is recording the earphone output acts as a monitor. The save-load switch does the same as pulling-out the plug but instead of letting the feedback go into thin air, it feeds it into the speaker so that you can hear the data transfer.

Figure 3:



Making another useful port for the ZX-81

Since the ZX-81 was introduced people have added peripherals. Most have used the rear socket but here Stephen Huckstepp shows how the cassette output port can be used.



ADDING PERIPHERALS to the ZX-81 can prove to be a cumbersome and daunting task with all those connections to be made to the rear socket, especially if the printer and/or RAM pack is to be used, too. So what can be done?

The ZX-81 is already equipped with three ports, two in and one out, for the cassette recorder and keyboard. They can be exploited in many ways, as has been done with the keyboard input. Very little use of the cassette output port has been made, however, apart from the elementary tone generator. That is because its main disadvantage is that it is serial rather than parallel.

To overcome it, first the theory. The cassette output port can, by nature, produce square waves. Let it produce the waveform shown in figure 10 as "out". By causing each pulse to trigger a monostable and comparing the two pulses using Boolean algebra, a clock-pulse can be formed only when the ZX-81 pulse is longer than the monostable pulse.

That can gate a latch and can be considered as an "on" level. A pulse shorter than the monostable pulse can be considered as an "off" level. The duration between each new pulse—i.e., HIGH time plus LOW time—as produced by the ZX-81 has to be longer than the monostable HIGH time, so as to allow it to re-set, or chaos reigns.

The output from the computer, a square wave, is taken to a high-impedance load—1MΩ pre-set potentiometer—from which a proportion is tapped-off and amplified, using a pair of BC184L transistors. The choice of transistors is not compulsory and various configurations of BC107-9 and 2N3704 have been tried in the prototype stage. The inverting amplifier then feeds a schmitt trigger buffer, made from two inverters. That then feeds the monostable—Ic4— and one input of a diode OR gate—figure four.

An output signal will trigger the monostable and also clock a counter—Ic1. Because that is negative edge-triggered, the $\overline{Q}$ output of the monostable is used. The counter—four counts maximum but adjustable—is decoded by Ic2 and presents latch Ic3 with the correct bit being dealt with. The Q output of the monostable is fed into the second input of the diode OR gate. The result is then converted back to TTL levels using an inverter, which in turn drives the clock inputs of the latch.

To 'beat propogation times when the clock pulse goes low, a capacitor charge circuit is included—C2,R3. The values of those two components are correct only for Ic3 being a 74LS device; other devices require different values of C2 and R3. Therefore the respective output—0-3—is switched on using the multiplexing system.

All clock lines on Ic3 are connected together and when one latch bit is switched on, all others go off. That is because for the original application of the circuit—flashing lights—one latch only needed to be clocked. Strobing all outputs continuously at high speed reduced the independent switching effect.

It can be arranged, by the duplication of circuits, to latch only one bit at a time, perhaps using the 74LS273 device. Also in the design shown one is not limited to using the 74LS75 for Ic3; experiment with the 74LS173.

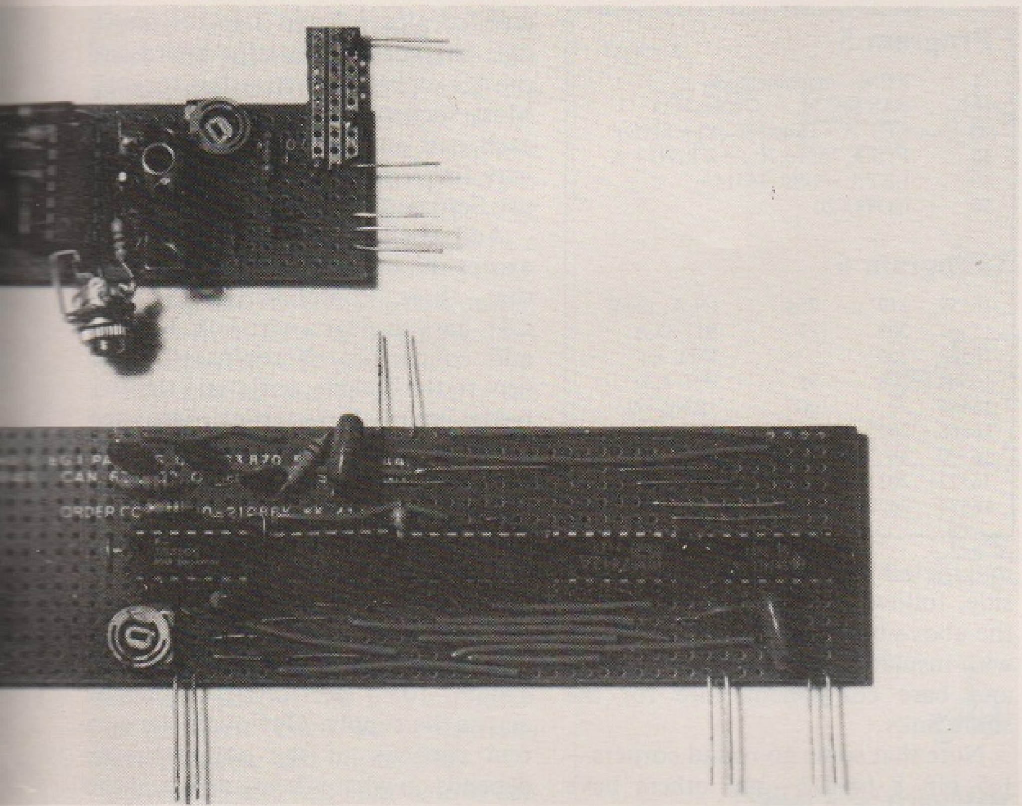
No synchronisation of the output channels to the ZX-81 signal was employed because it was unnecessary in this example but by comparing an extra-long pulse to one from a re-triggerable monostable, a re-set pulse for the counter—Icl—can be generated.

Expansion is logical by increasing the count of Icl and hence taking further outputs from Ic2; for more than ten outputs the device will have to be a 74LS154.

Outputs are in their non-inverted form when using $\overline{Q}_a$-$\overline{Q}_d$ and an inverted form, as used in this application, can be obtained with $\overline{Q}_a$-$\overline{Q}_d$. That is because the Ic2 outputs are inverted.

The circuit is built on a piece of Verostrip—narrow Veroboard with a break down the centre—with dimensions of 43 by 15 holes. Start by

INPUT OUTPUT PORT



Program 1

```
1     REM—any character, 62 of them
110   FOR F = 16514 TO 16528
120   SCROLL
130   PRINT F; "(space)";
140   INPUT I
150   POKE F, I
160   PRINT PEEK F
170   NEXT F
```

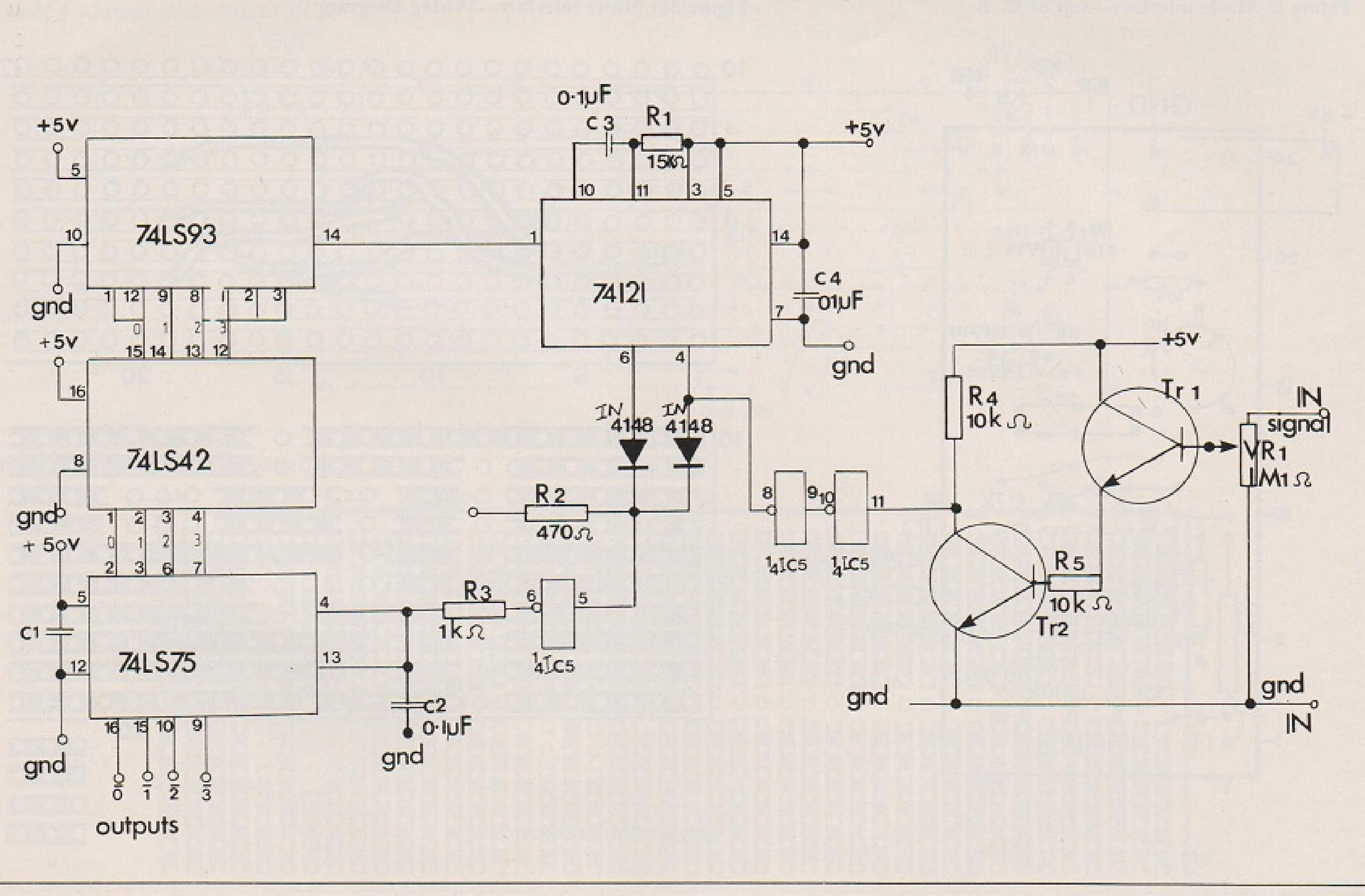
Program 2

```
16514  211   255   OUT (255), A
16516  6     0     LD B, 0
16518  0           NOP
16519  16    253   DJNZ-1
16521  219   254   IN A, (254)
16523  6     32    LD B, 32
16525  0           NOP
16526  16    253   DJNZ-1
16528  201         RET
```

Program 3

```
1     REM—machine code
10    FAST
20    POKE 16517,0
30    POKE 16524,32
40    LET L = USR 16514
50    POKE 16517,128
60    POKE 16524,160
70    LET L = USR 16514
80    GOTO 20
```

Figure 1: Output port—circuit diagram



INPUT OUTPUT PORT

Program 4

```
16514  17    17  17   LD DE, 4369
16517  6     16       LD B, 16
16519  122            LD A, D
16520  211   255      OUT (255), A
16522  163            AND E
16523  230   240      AND 240
16525  40    5        JRZ +7
16527  14    128      LD C, 128
16529  13             DEC C
16530  32    253      JRNZ -1
16532  14    128      LD C, 128
16534  13             DEC C
16535  32    253      JRNZ -1
16537  219   254      IN A, (254)
16539  122            LD A, D
16540  163            AND E
16541  230   240      AND 240
16543  32    5        JRNZ +7
16545  14    128      LD C, 128
16547  13             DEC C
16548  32    253      JRNZ -1
16550  14    32       LD C, 32
16552  13             DEC C
16553  32    253      JRNZ -1
16555  203   3        RLC E
16557  16    216      DJNZ -38
16559  21             DEC D
16560  122            LD A, D
16561  230   15       AND 15
16563  32    208      JRNZ -46
16565  201            RET
```

Program 5

```
1   REM—machine code
10  FAST
20  LET A=16★INT (15★RND)
30  POKE 16516,10+5★RND+A
40  LET L=USR 16514
50  GOTO 20
```

Program 6

```
16559  219  254  IN A, (254)
15561  203  71   BIT O,A
16563  200       RET Z
16564  203  95   BIT 3,A
16566  32   205  JRNZ-49
16568  203  34   SLA D
16570  48   201  JRNC-53
16572  203  226  SET 4,D
16574  24   197  JR -57
```

making all the breaks on the underside, followed by the insertion of all the above-board links—26 of them—with insulated wire for the long links and bare component wire for the short links.

Note that some go round corners—Ic5 pin 8 to D2—and others have multiple connections made to them—Ic2 pin 8 to GND (in) to Tr2 emitter, and Ic5 pins 1,3 and 7 to Ic4 pin 7. One link goes between the board and the IC sockets—when using sockets. Most sockets have a notch beneath each end and they can contain the wire link comfortably. Alternatively use Soldercon IC pins.

Add sockets and all components except the integrated circuits, noting that three components—R1, R3, C3—have to clear a wire link. Finally, add connectors—Veropins—to connect to the outside world and the five below-board links using insulated wire. Then insert all integrated circuits the correct way round.

The power supply—that is independent of the ZX-81 PSU so as not to overload it—was one which happened to be spare and was fully-regulated and smoothed at 5V. If the supply you use is not smoothed, place a 470uF 10V electrolytic capacitor across the supply. Obviously the current capacity of the power supply depends on what is being driven at the outputs, through buffers if necessary.

The ZX-81 cassette output is a sine

Figure 2: Music interface—top of PCB

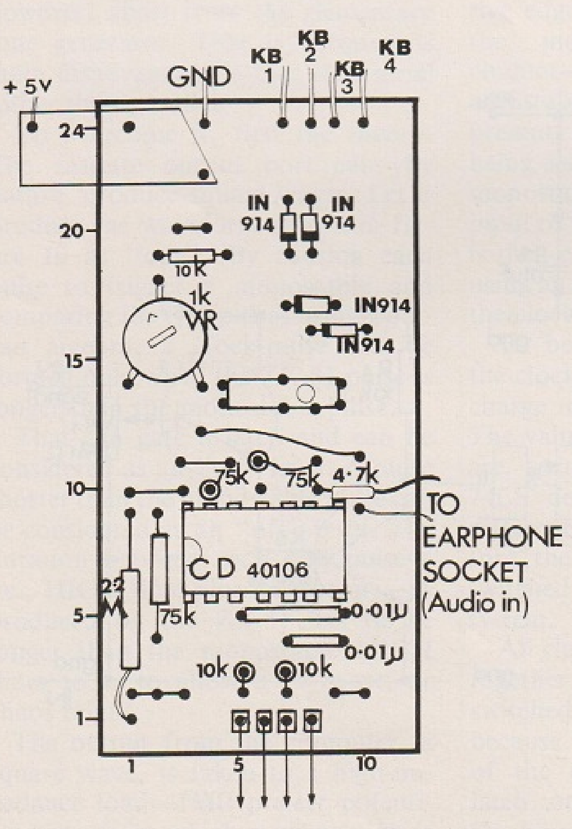
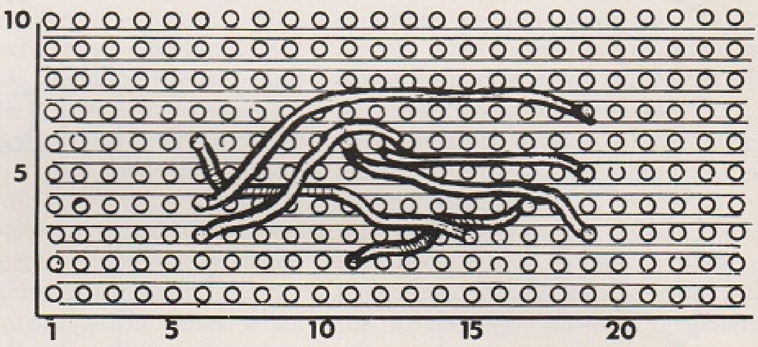


Figure 3a: Music interface—Wiring Diagram



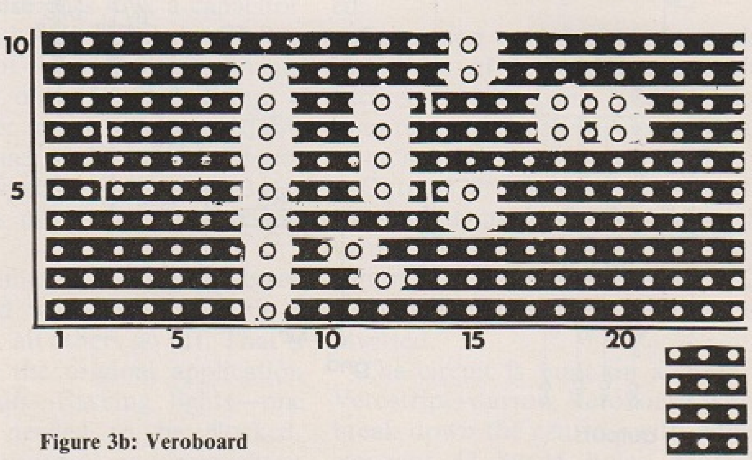


Figure 3b: Veroboard

INPUT OUTPUT PORT

wave of a low magnitude, resulting from the alterations made to the square wave by R27/29 C11/12 inside the computer. Thus a change has to be made. There are two ways and one is shown in figure seven. All the circuitry to the right of the R29–C12 junction is removed and placed in a small adaptor made up of a 3½mm. jack plug and socket. The new output is taken from R29, by putting a short link across the C12 position.

When using the output port, the port input is plugged straight into the MIC output. For saving a program, the adaptor is plugged into the computer and tape into the adaptor. Note that the automatic protection built into the computer if the incorrect plugs are put into the wrong sockets is now no longer there for the MIC socket. There is some protection evident. Whether or not it is sufficient has not been tested, so be cautious when plugging and unplugging.

Alternatively, two connections can be made to ground and the junction of R29–C12 can be run to an extra

Figure 6

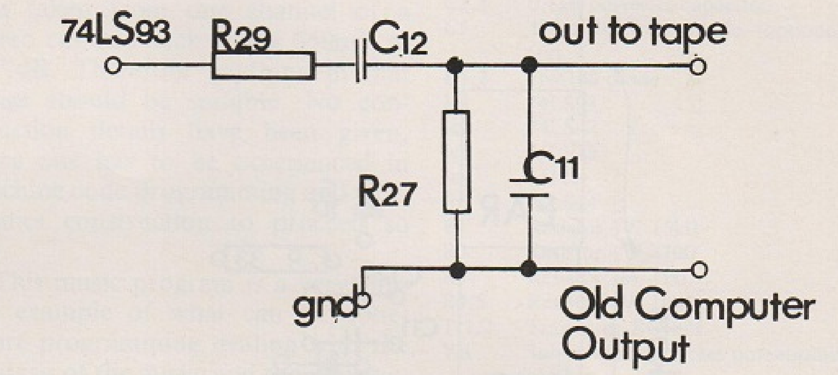


Figure 7

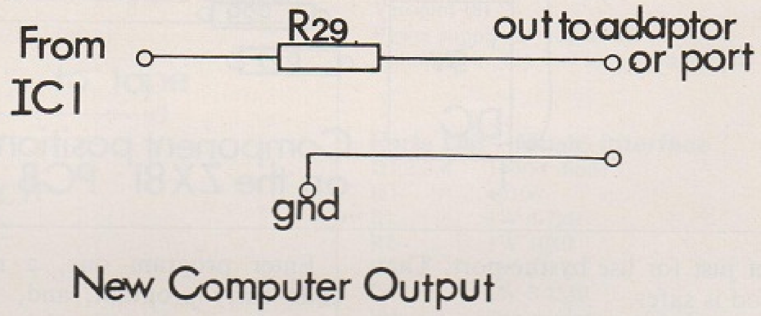


Figure 4: Output port—top of PCB

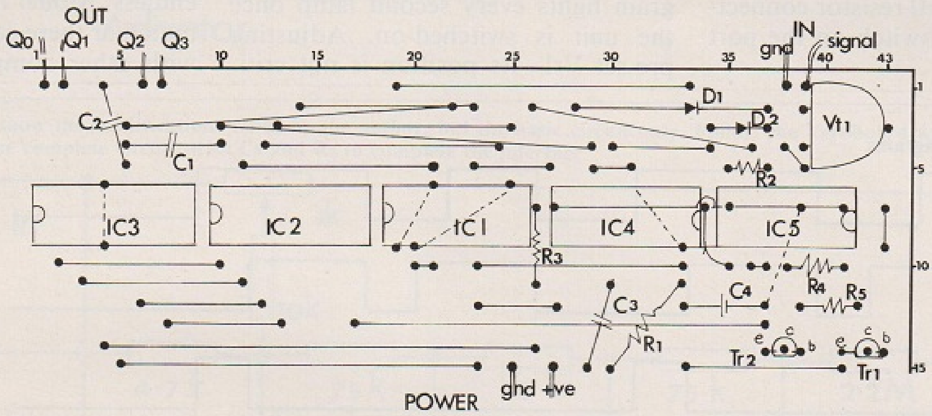
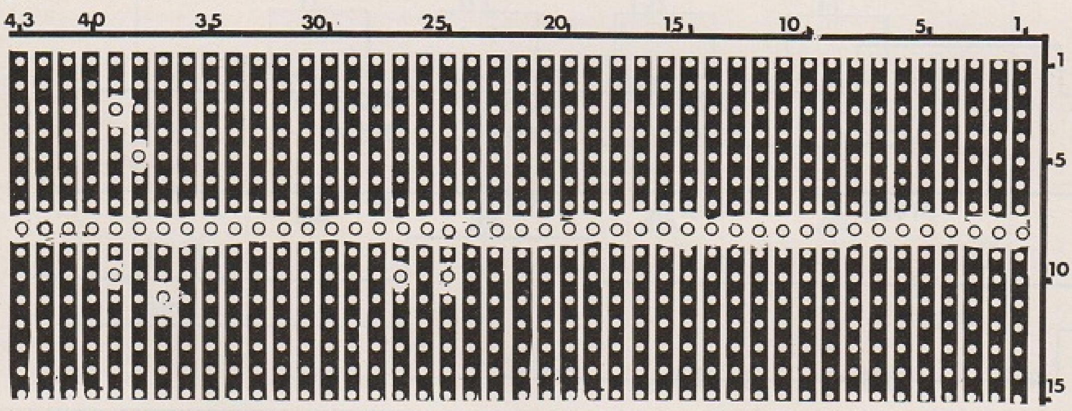


Figure 5: Output port—Veroboard



INPUT OUTPUT PORT

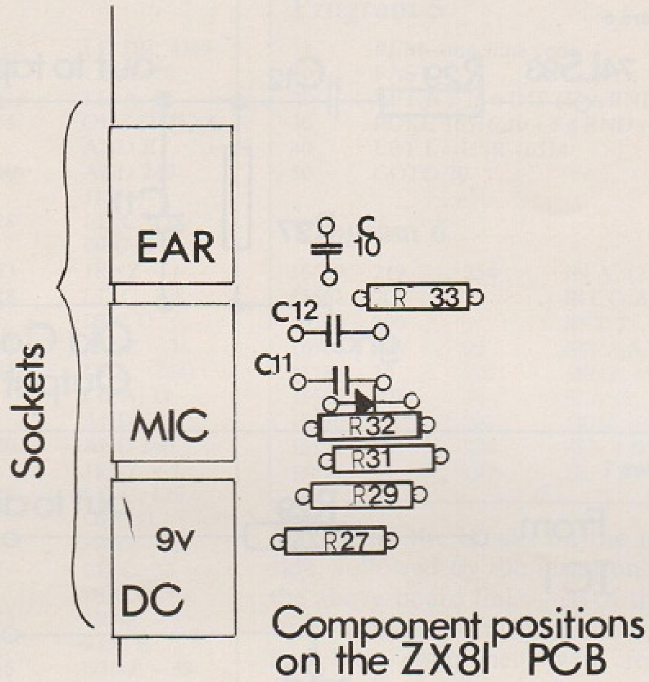


Figure 8

socket just for use by the port. That method is safer.

Once all alterations are complete, connect everything including power and four outputs, e.g., a light emitting diode and a 1kΩ resistor connected to 5V. Do not switch on the port power supply yet.

Enter program one, a machine-code entry program, and, using it, input the numbers listed in program two. Then enter the Basic lines in test program three and RUN. The program lights every second lamp once the unit is switched-on. Adjusting pre-set Vrl—its position is not critical—will ensure this. If not, switch-off and check for faults.

By BREAKing the program and RUNning again, it is possible, because of no synchronisation, to light alternate LEDs.

Once everything is working, it is possible to move to a more useful program. Change line 110 to:

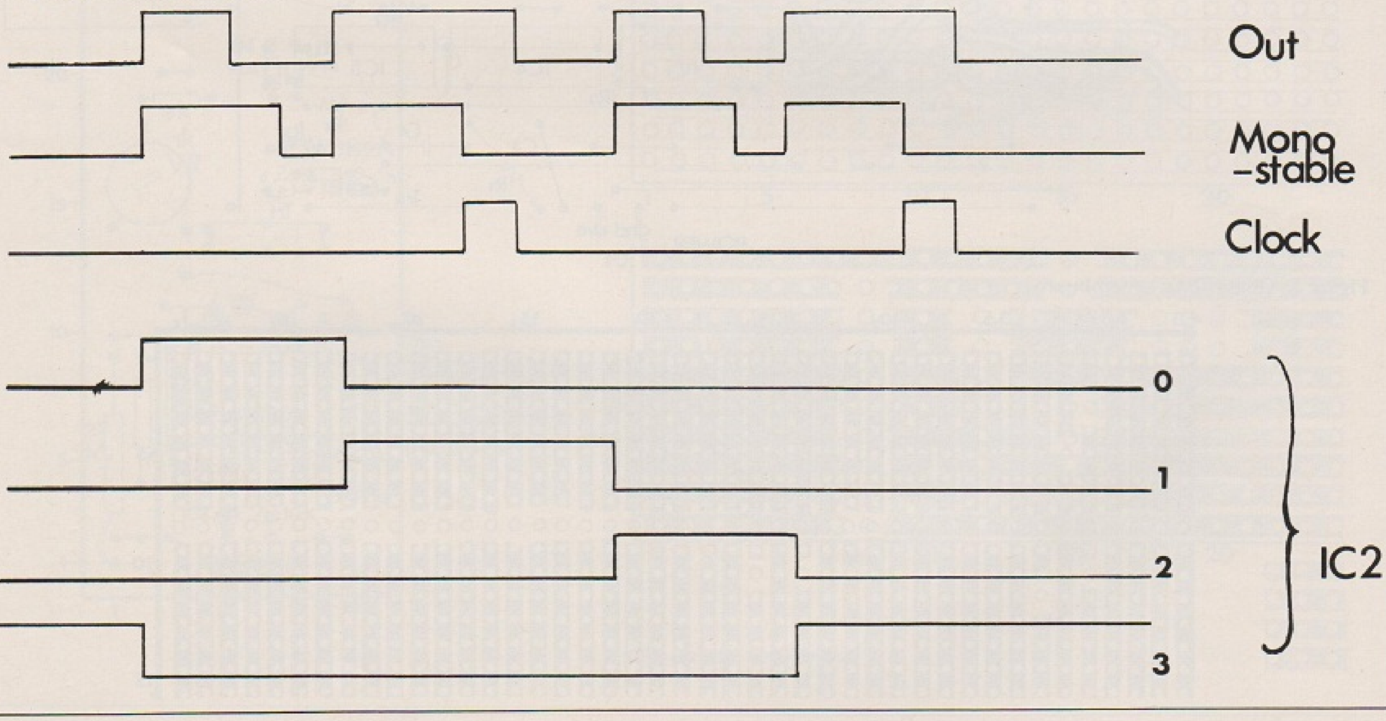
110 FOR F = 16514 TO 16565

and enter the data listed in program four by RUNning line 110.

That program in machine-code looks at the last four bits in location 16516, where each bit is a respective output. If a bit is high (1), that output is switched-on, and vice versa if it is low (0). Therefore the decimal number 240—16 × 15 and binary 1111—is all outputs on, and zero—16 × 0 and binary 0000—is all outputs off. There is also a duration counter in the first four bits of 16516. Enter lines 20-50 of program five and that will turn on a random four bits—line 20—for a random time—line 30—when RUN. That gives a random pattern of flashing lights if still connected to the test LEDs.

The limits of that output port are endless. Model railways can be automated or there can be communication with other computers via a parallel

Figure 10: Timing Diagrams



INPUT OUTPUT PORT

input. For example, by building the input described in figure two, the light outputs can be synchronised with music. Do not erase program five but change line 110 to:

110 FOR F = 16559 TO 16575

and enter the data and program six by RUN 110.

Erase lines 20–60 and enter the command:

POKE 16516, 17

Then connect the music interface once the program is running; it hangs up the keyboard otherwise. Pressing BREAK will still allow one to escape from the program. Adjust the pre-set so that the lights do not fly about too much when in use. In testing, a signal was taken from one channel of a stereo cassette deck giving 550mV at "0"dB. Therefore anything in that range should be suitable. No construction details have been given, since one has to be experienced in machine code programming and electronics construction to proceed so far.

This music program is a very simple example of what can be done. More programming dealing with the analysis of the music and the sequencing of the lights can make this unit far superior to anything else available.

Parts List—Output port

C1,4	0.1μF ceramic capacitor
C2,3	0.1μF polyester capacitor
C5	470μF 10V electrolytic—optional, see text
D1,2	1N4148 diode
Ic1	74LS93
Ic2	74LS42
Ic3	74LS75
Ic4	74121
Ic5	74LS14
R1	Resistor ¼W 15kΩ
R2	Resistor ¼W 470Ω
R3	Resistor ¼W 1kΩ
R4,5	Resistor ¼W 10kΩ
Tr1,2	Transistor BC184L
Vr1	Sub-miniature preset potentiometer 1MΩ

Miscellaneous
IC sockets (3 × 14 pin, 2 × 16 pin)
Veropins (8)
Power supply (5v regulated)
3½mm. jack plug and socket—optional

Parts List—Music Interface

D1,2,3,4	1N914 diode
Ic1	40106
R1	¼W 4.7kΩ
R2	¼W 10kΩ
R3,4,5	¼W 75kΩ
R6	¼W 2.2MΩ
Vr1	Sub-miniature pre-set potentiometer 1kΩ

Figure 9

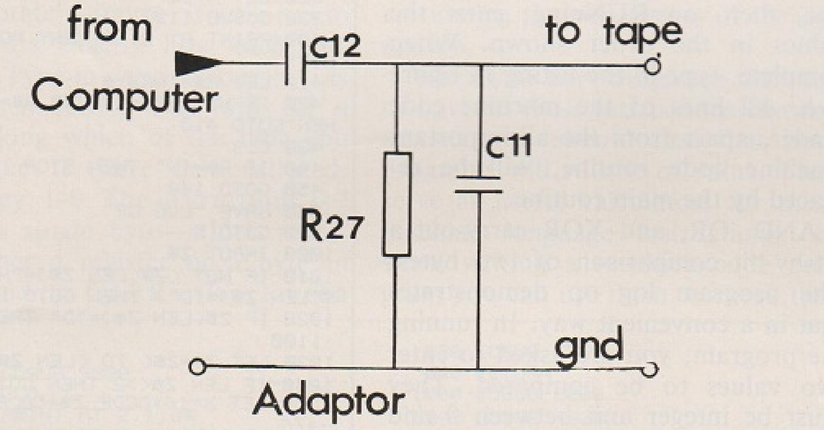
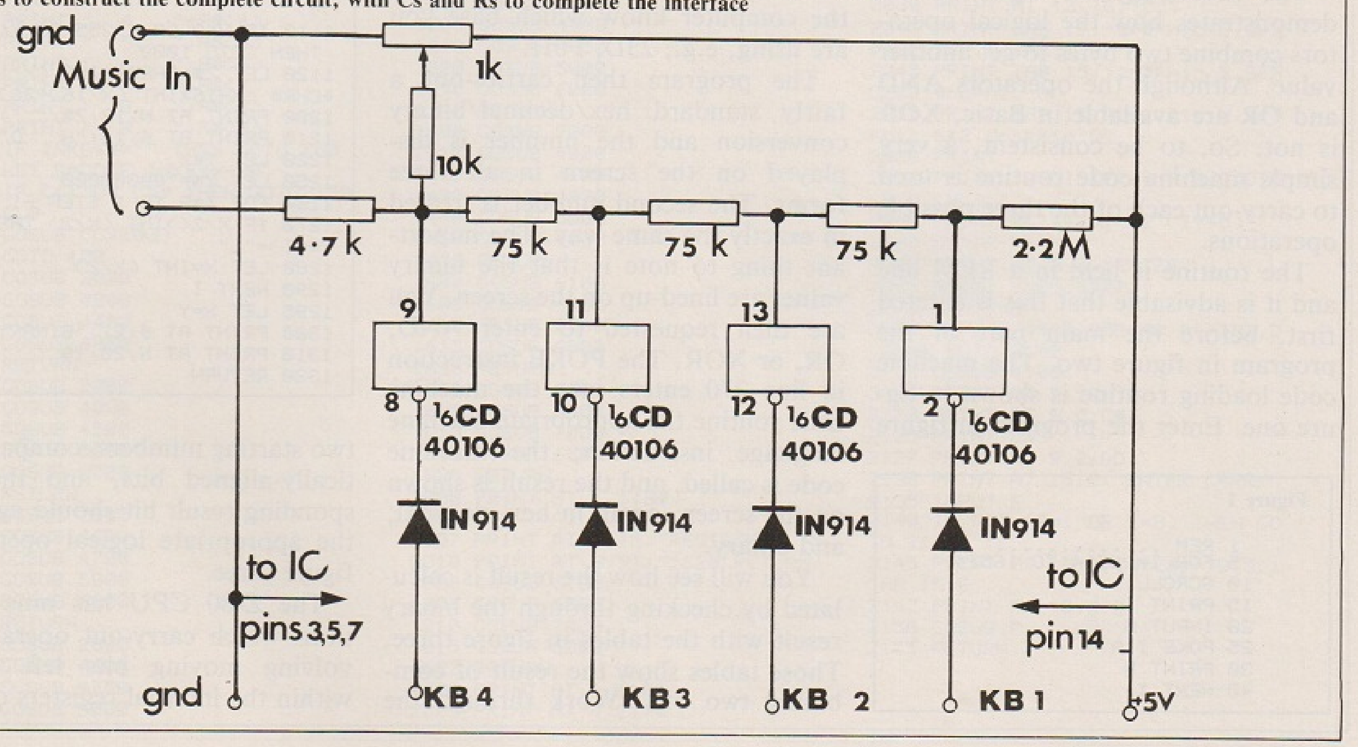


Figure 11: Music Interface
This diagram does not show the modifications made by the author, but the basic circuit only. Follow the Veroboard wiring component diagrams to construct the complete circuit, with Cs and Rs to complete the interface



Operating logically to improve Basic

David Nowotnik explains how to overcome two problem areas when using machine code to speed up the working of hardware projects. The programs apply to the ZX-81 but can be adapted for the Spectrum.

MANY OF the machine language instructions can be compared to the simple Basic commands e.g., LD can be compared to POKE or LET, RET with RETURN. That helps considerably in learning but there are some which are more difficult to comprehend.

Logical operators—AND, OR, XOR—and the shift and rotate instructions can be troublesome. To help overcome that hurdle, two programs will help the understanding of those concepts. The ZX-81 listings for the programs are shown in figures two and four.

The first program—figure two—demonstrates how the logical operators combine two bytes to get another value. Although the operators AND and OR are available in Basic, XOR is not. So, to be consistent, a very simple machine code routine is used to carry-out each of the three possible operations.

The routine is held in a REM line and it is advisable that this is entered first, before the main part of the program in figure two. The machine code loading routine is shown in figure one. Enter the program in figure one, then, on RUNning, enter the values in the order shown. When complete, type-in the listing in figure two. All lines of the machine code loader, apart from the all-important machine code routine, will be replaced by the main routine.

Figure 1

```
  1 REM ...........
  5 FOR I=16514 TO 16524
 10 SCROLL
 15 PRINT I
 20 INPUT A
 25 POKE I,A
 30 PRINT A
 40~NEXT I
```

AND, OR, and XOR carry-out a bit-by-bit comparison of two bytes. The program log op demonstrates that in a convenient way. In running the program, you are asked to enter two values to be compared. They must be integer and between 0 and 255. You may enter the values in decimal or hexadecimal; add the suffix D or H to the value entered to let the computer know which base you are using, e.g., 23D, F0H.

The program then carries-out a fairly standard hex/decimal/binary conversion and the number is displayed on the screen in all three forms. The second number is treated in exactly the same way. The important thing to note is that the binary values are lined-up on the screen. You are then requested to enter AND, OR, or XOR. The POKE instruction in line 270 enters into the machine code routine the appropriate machine language instruction; the machine code is called, and the result is shown on the screen, again in hex, decimal, and binary.

You will see how the result is calculated by checking through the binary result with the tables in figure three. Those tables show the result of combining two bits. Work through the two starting numbers, comparing vertically-aligned bits, and the corresponding result bit should agree with the appropriate logical operation in figure three.

Figure 2

```
 140 PRINT AT 4,5;"DEMONSTRATION
 "
 150 PRINT AT 4,5;"OF LOGICAL OP
ERATORS"
 170 LET N=8
 180 GOSUB 1000
 190 POKE 16514,X
 200 PRINT AT 10,0;"FIRST OPERAN
D ";
 205 LET N=10
 210 GOSUB 1000
 220 POKE 16515,X
 230 PRINT AT 21,4;"AND OR XOR -
 SELECT"
 240 INPUT A$
 250 IF NOT (A$="AND" OR A$="OR"
 OR A$="XOR") THEN GOTO 240
 260 LET A=(A$="AND")*166+(A$="O
R")*182+(A$="XOR")*174
 270 POKE 16522,A
 280 RAND USR 16516
 290 PRINT AT 21,0;"
 300 PRINT AT 12,5;A$;" ="
 310 LET X=PEEK 16515
 320 LET N=12
 330 GOSUB 1120
 400 PRINT AT 21,0;"ANY MORE (Y/
N)?"
 410 LET A$=INKEY$
 420 IF NOT (A$="Y" OR A$="N") T
HEN GOTO 410
 430 CLS
 440 IF A$="N" THEN STOP
 450 GOTO 140
 500 SAVE "LOG OP"
 520 GOTO 5
1000 INPUT Z$
1010 IF NOT (Z$(LEN Z$)="H" OR Z
$(LEN Z$)="D") THEN GOTO 1000
1020 IF Z$(LEN Z$)="D" THEN GOTO
 1100
1030 LET Z$=Z$( TO (LEN Z$-1))
1040~IF LEN Z$<>2 THEN GOTO 1000
1050 LET X=16*CODE Z$+CODE Z$(2)
-476
1060 IF X<0 OR X>255 OR X<>INT X
 THEN GOTO 1000
1090 GOTO 1200
1100 LET X=VAL (Z$( TO (LEN Z$-1
)))
1110 IF X<0 OR X>255 OR X<>INT X
 THEN GOTO 1000
1120 LET Z$=CHR$ (INT (X/16)+28)
+CHR$ (X-16*INT (X/16)+28)
1200 PRINT AT N,12;Z$;" ";X
1210 PRINT AT 6,12;"H   D"
1220 LET Y=X
1250 LET Y$="00000000"
1260 FOR I=8 TO 1 STEP -1
1270 IF X/2<>INT (X/2) THEN LET
Y$(I)="1"
1280 LET X=INT (X/2)
1290 NEXT I
1295 LET X=Y
1300 PRINT AT 6,21;"BINARY"
1310 PRINT AT N,20;Y$
1320 RETURN
```

The Z-80 CPU has nine instructions which carry-out operations involving moving bits left or right within the internal registers or RAM.

Figure 3: AND, OR and XOR

AND
0 AND 0 = 0
0 AND 1 = 0
1 AND 1 = 1

OR
0 OR 0 = 0
0 OR 1 = 1
1 OR 1 = 1

XOR
0 XOR 0 = 0
0 XOR 1 = 1
1 XOR 1 = 0

How they differ can be very confusing. The purpose of the program 'Shift/Rotate'—figure four—is to demonstrate exactly how each one works. Type-in the program as listed, then RUN. You will get a menu asking which of the nine you would like to have demonstrated. Press a key, 1–9. The instructions 1–7 involve a single byte—which represents either a register or a byte in RAM—and the CARRY flag. Enter a value for the byte—0 to 255, in decimal only—and 0 or 1 for the CARRY flag. Then watch the screen. You will see a slow-motion display of the bits in the register/byte and CARRY moving about to create new values. The program also converts the new binary value in the register to decimal. Trying each instruction in turn will demonstrate the differences. If you select 8 or 9, you will obtain the machine language instructions which shuffle the value in the A register and a byte in RAM.

Enter again, in decimal, the two values for the register and byte, and the program will do the rest, moving the bits around in slow motion and calculating the two results of the operation.

When you have outgrown the need to have all these instructions demonstrated, you should find that both programs will still be of value. They serve as useful calculators for these machine language instructions, so when writing a machine code program you can use them to check that your use of the instructions produces the desired effect.

Figure 5

```
 30 GO TO 500
190 POKE 32000,X
220 POKE 32001,X
270 POKE 32008,A
280 RANDOMIZE USR 320002
310 LET X=PEEK 32001
500 CLEAR 31999
510 LET X=32000
520 FOR I=32000 TO 32010
530 READ A: POKE I,A
540 NEXT I
550 DATA 0,0,38,125,46,0,126,35
166,119,201
560 GO TO 140
```

As a final note for Spectrum owners, the program 'Shift/Rotate' can be entered unchanged, although you may want to add a little colour. Because the program storage area is not a convenient place to store machine code on the Spectrum, "log op" will have to be altered slightly, with the machine code held above RAMTOP.

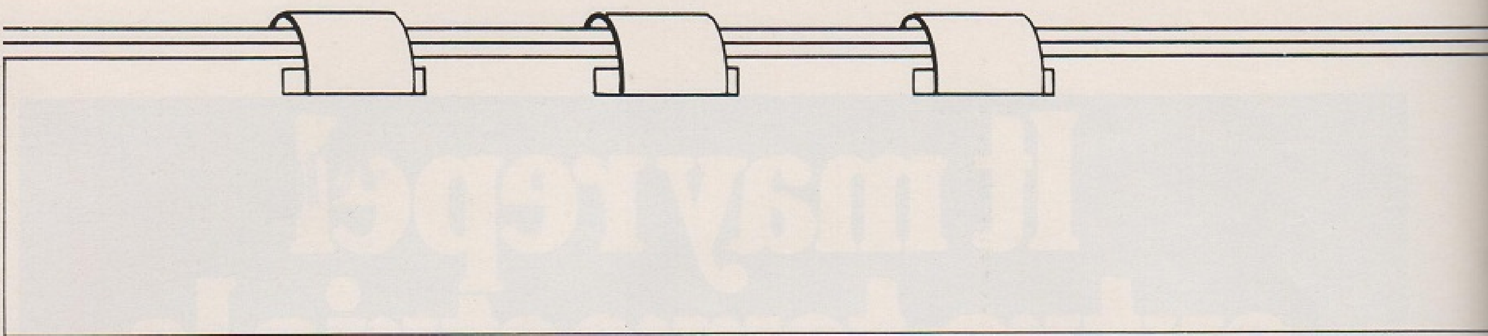
For the Spectrum, do not use the routine in figure one to enter the machine code routine. Instead enter the program as listed in figure two, with the changes and additions listed in figure five. The program can be started with RUN.

Figure 4

```
  50 GOSUB 9000
 100 CLS
 105 PRINT AT 2,1;B$
 110 PRINT AT 6,2;"SELECT :-"
 125 PRINT
 130 PRINT "   1.   ";A$(1)
 132 PRINT "   2.   ";A$(2)
 134 PRINT "   3.   ";A$(3)
 135 PRINT "   4.   ";A$(4)
 137 PRINT "   5.   ";A$(5)
 138 PRINT "   6.   ";A$(6)
 139 PRINT "   7.   ";A$(7)
 140 PRINT "   8.   ";A$(8)
 143 PRINT "   9.   ";A$(9)
 150 IF INKEY$="" THEN GOTO 150
 155 LET C=CODE INKEY$-28
 160 IF C<1 OR C>9 THEN GOTO 150
 170 CLS
 180 GOSUB (C*200)
 190 GOTO 100
 200 GOSUB 2000
 210 GOSUB 4000
 220~GOSUB 4400
 230 GOSUB 6000
 240 RETURN
 400 GOSUB 2000
 405 GOSUB 4000
 410 GOSUB 4500
 415 GOSUB 5000
 420 GOSUB 6000
 425 RETURN
 600 GOSUB 2000
 610 GOSUB 4600
 620 GOSUB 4700
 630 GOSUB 5000
 640 GOSUB 6000
 650 RETURN
 800 GOSUB 2000
 805 GOSUB 4600
 810 GOSUB 4750
 820 GOSUB 5000
 830 GOSUB 6000
 840 RETURN
1000 GOSUB 2000
1005 GOSUB 4300
1010 GOSUB 4800
1015 GOSUB 5000
1020 GOSUB 6000
1030 RETURN
1200 GOSUB 2000
1205 GOSUB 4600
1210 GOSUB 4850
1220 GOSUB 5000
1230 GOSUB 6000
1240 RETURN
1400 GOSUB 2000
1410 GOSUB 4900
1420 GOSUB 4600
1430 GOSUB 4930
1440 GOSUB 5000
1450 GOSUB 6000
1460 RETURN
1600 GOSUB 6100
1605 GOSUB 6400
1610 GOSUB 7000
1615 GOSUB 6000
1620 RETURN
1800 GOSUB 6100
1805 GOSUB 6500
1810 GOSUB 7000
1815 GOSUB 6000
1820 RETURN
2000 PRINT AT 1,1;B$
2005 PRINT AT 4,2;A$(C)
2007 PRINT AT 7,12;"REGISTER"
2010 PRINT AT 8,11;"1620 RETURN
1800 GOSUB 6100
1805 GOSUB 6500
1810 GOSUB 7000
1815 GOSUB 6000
1820 RETURN
2000 PRINT AT 1,1;B$
2005 PRINT AT 4,2;A$(C)
2007 PRINT AT 7,12;"REGISTER"
2010 PRINT AT 8,11;"graphic E,gr
aphic 7*9,graphic R"
2015 PRINT AT 9,11;"graphic E,SP
ACE*9,graphic 8"
2020 PRINT AT 10,11;"graphic W,g
raphic 6*9,grpahic 8"
2025 PRINT AT 11,12;"76543210"
2030 PRINT AT 13,16;"C"
2040~PRINT TAB 15;"graphic E,gra
phic 7*2,graphic R"
2045 PRINT TAB 15;"graphic 5,SPA
CE*2,graphic 8"
2050 PRINT TAB 15;"graphic W,gra
phic 6*2,graphic Q"
2060 PRINT
2070 PRINT "REGISTER = A,B,C,D,E
,H,L,(HL),"
2080 PRINT "(IX+D), OR (IY+D)"
2085 GOSUB P
2090 PRINT AT 8,0;"ENTER"
2095 PRINT "REGISTER"
2100 INPUT D
2105 IF D<0 OR D>255 OR D<>INT D
 THEN GOTO 2100
2110 GOSUB 4100
2115 PRINT AT 9,12;Y$
2120 PRINT AT 8,0;C$
2125 PRINT C$
2127 PRINT AT 9,2;DD
2130 PRINT AT 15,0;"ENTER CARRY"
2135~INPUT E
2140 IF NOT (E=1 OR E=0) THEN GO
TO 2135
2145 PRINT AT 15,0;C$;C$( TO 5);
TAB 15;E
2147 PRINT AT 15,2;E
2150 GOSUB P
2155 RETURN
```

MACHINE CODE GUIDE

```
4000 FOR I=1 TO 25
4010 NEXT I
4020 RETURN
4100 LET Y$="00000000"
4102 LET DD=D
4105 FOR I=8 TO 1 STEP -1
4110 IF D/2<>INT (D/2) THEN LET
Y$(I)="1"
4115 LET D=INT (D/2)
4220 NEXT I
4225 RETURN
4300 PRINT AT 9,12;CHR$ (CODE Y$
+128)
4310 GOSUB P
4320 PRINT 9,12;" ";AT 9,6;CHR$
(CODE Y$+128)
4330 GOSUB P
4335 FOR I=2 TO 8
4340 PRINT AT 9,(10+I);Y$(I)+" "
4345 FOR J=1 TO 10
4350 NEXT J
4355 NEXT I
4360 GOSUB P
4365 RETURN
4400 PRINT AT 9,6;" ";AT 9,19;CH
R$ (CODE Y$+128);AT 15,16;CHR$ (
CODE Y$+128)
4405 GOSUB P
4410 PRINT AT 9,19;Y$(1);AT 15,1
6;Y$(1)
4415 LET Y$=Y$(2 TO )+Y$(1)
4420 GOSUB 5000
4425 RETURN
4500 PRINT AT 15,16;CHR$ (E+156)
4505 GOSUB P
4510 PRINT AT 15,16;" ";AT 9,19;
4515 GOSUB P
4520 PRINT AT 9,6;" ";AT 15,16;C
HR$ (CODE Y$(1)+128)
4525 GOSUB P
4530 PRINT AT 15,16;Y$(1);AT 9,1
9;CHR$ (E+28)
4540 LET Y$=Y$(2 TO )+CHR$ (E+28
)
4550 RETURN
4600 PRINT AT 9,19;CHR$ (CODE Y$
(8)+128)
4605 GOSUB P
4610 PRINT AT 9,19;" ";AT 9,24;C
HR$ (CODE Y$(8)+128)
4615 GOSUB P
4620 FOR I=7 TO 1 STEP -1
4625 PRINT AT 9,11+I;" "+Y$(I)
4630 FOR J=1 TO 10
4635 NEXT J
4645 NEXT I
4650 GOSUB P
4655 RETURN
4650 GOSUB P
4655 RETURN
4700 PRINT AT 9,24;" ";AT 9,12;C
HR$ (CODE Y$(8)+128);AT 15,16;CH
R$ (CODE Y$(8)+128)
4705 GOSUB P
4710 PRINT AT 9,12;Y$(8);AT 15,1
6;Y$(8)
4715 LET Y$=Y$(8)+Y$( TO 7)
4720 RETURN
4750 PRINT AT 15,16;CHR$ (E+156)
4755 GOSUB P
4760 PRINT AT 15,16;AT 9,12;CHR$
(E+156)
4765 GOSUB P
4765~GOSUB P
4770 PRINT AT 9,24;" ";AT 15,16;
CHR$ (CODE Y$(8)+128)
4775 GOSUB P
4780 PRINT AT 9,12;CHR$ (E+28);A
T 15,16;Y$(8)
4785 LET Y$=CHR$ (E+28)+Y$(7)
4790 RETURN
4800 PRINT AT 9,6;" ";AT 15,16;C
HR$ (CODE Y$+128)
4810 GOSUB P
4815 PRINT AT 9,23;"inverse 0"
4820 GOSUB P
4825 PRINT AT 9,23;" ";AT 9,19;"
0"
4830 GOSUB P
4835 PRINT AT 9,19;"0";AT 15,16;
Y$(1)
4840 LET Y$=Y$(2 TO )+"0"
4845 RETURN
4850 PRINT AT 9,24;" ";AT 15,16;
CHR$ (CODE Y$(8)+128)
4855 GOSUB P
4860 PRINT AT 9,6;"0"
4865 GOTO P
4870 PRINT AT 9,6;" ";AT 9,12;"0
"
4875 GOSUB P
4880 PRINT AT 9,12;"0";AT 15,16;
Y$(8)
4885 LET Y$="0"+Y$( TO 7)
4890 RETURN
4900 PRINT AT 9,12;CHR$ (CODE Y$
+128)
4905 GOSUB P
4907 PRINT AT 9,6;CHR$ (CODE Y$+
128)
4910 GOSUB P
4915 PRINT AT 9,12;Y$
4920 GOSUB P
4925 RETURN
4930 PRINT AT 9,24;" ";AT 15,16;
CHR$ (CODE Y$(8)+128)
4935 GOSUB P
4935~GOSUB P
4940 PRINT AT 9,6;" ";AT 9,12;CH
R$ (CODE Y$+128)
4950 PRINT AT 9,12;Y$(1);AT 15,1
6;Y(8)
4955 LET Y$=Y$(1)+Y$( TO 7)
4960 RETURN
5000 LET X=0
5005 FOR I=8 TO 1 STEP -1
5010 LET X=X+(CODE Y$(I)-28)*(2*
(8-I))
5020 NEXT I
5030 PRINT AT 9,22;"= ";X
5040 RETURN
6000 PRINT AT 21,2;"PRESS ANY KE
Y TO COTINUE"
6010 IF INKEY$="" THEN GOTO 6010
6020 RETURN
6100 PRINT AT 2,1;B$
6105 PRINT AT 4,2;A$(C)
6115 PRINT AT 8,11;"graphic E,gr
ahic 7*9,graphic R"
6120~PRINT AT 9,11;"graphic 5,SP
CE*9,graphic 8"
6125 PRINT AT 10,11;"graphic W,g
raphic 6*9,graphic Q"
6130 PRINT AT 11,12;"76543210"
6135 PRINT AT 13,16;"A"
6140 PRINT AT 14,11;"graphic E,g
raphic6*9,graphic R"
6145 PRINT AT 15,11;"graphic 5,S
PACE*9,graphic 8"
6150 PRINT AT 16,11;"graphic W,g
raphic 6*9,graphic Q"
6155 PRINT AT 17,12;"76543210"
6160 GOSUB P
6165 PRINT AT 9,0;"INPUT (HL)"
6170 INPUT D
6180 IF D<0 OR D>255 OR D<>INT D
 THEN GOTO 6170
6185 GOSUB 4100
6190 PRINT AT 9,0;C$+"  ";AT 9,2
;D
6197 PRINT AT 9,12;Y$
6200 LET Z$=Y$
6205 PRINT AT 15,0;"INPUT A"
6210 INPUT D
6215 IF D<1 OR D>255 OR D<>INT D
 THEN GOTO 6210
6220 GOSUB 4100
6225 PRINT AT 15,0;C$;AT 15,12;Y
$;AT 15,2;DD
6235 LET X=128
6330 LET R$=CHR$ (CODE Z$+X)+CHR
$ (CODE Z$(2)+X)+CHR$ (CODE Z$(3
)+X)+CHR$ (CODE Z$(4)+X)
6335 LET S$=CHR$ (CODE Z$(5)+X)+
CHR$ (CODE Z$(6)+X)+CHR$ (CODE Z
$(7)+X)+CHR$ (CODE Z$(8)+X)
6340 LET T$=CHR$ (CODE Y$(5)+X)+
CHR$ (CODE Y$(6)+X)+CHR$ (CODE Y
$(7)+X)+CHR$ (CODE Y$(8)+X)
6345 RETURN
6400 PRINT AT 15,16;T$
6405 GOSUB P
6410 PRINT AT 15,16;"    ";AT 15,
6415~GOSUB P
6420 PRINT AT 9,12;R$
6425 GOSUB P
6430 PRINT AT 9,12;"    ";AT 15,1
6;R$
6435 GOSUB P
6440 PRINT AT 9,16;"    ";AT 9,12
;Z$(5 TO )
6445 GOSUB P
6450 PRINT AT 15,23;C$;AT 9,16;T
$
6455 LET T$=Z$(5 TO )
6460 LET R$=Y$(5 TO )
6465 LET Y$=Y$( TO 4)+Z$( TO 4)
6470 LET Z$=T$+R$
6472 GOSUB P
6475 PRINT AT 9,12;Z$;AT 15,12;Y
$
6480 RETURN
6500 PRINT AT 15,16;T$
6505 GOSUB P
6510 PRINT AT 15,16;"    ";AT 15,
6515 GOSUB P
6520 PRINT AT 9,16;S$
6525 GOSUB P
6530 PRINT AT 9,16;"    ";AT 15,1
6;S$
6535 GOSUB P
6540 PRINT AT 9,12;"    ";Z$( TO
4)
6545 GOSUB P
6550 PRINT AT 15,20;C$;AT 9,12;T
$
6555 LET T$=Z$(5 TO )
6560 LET R$=Y$(5 TO )
6565 LET Y$=Y$( TO 4)+T$
6570 LET Z$=R$+Z$( TO 4)
6572 GOSUB P
6575 PRINT AT 9,12;Z$;AT 15,12;Y
$
6580 RETURN
7000 LET X=0
7005 FOR I=8 TO 1 STEP -1
7010~LET X=X+(CODE Z$(I)-28)*(2*
*(8-I))
7020 NEXT I
7030 PRINT AT 9,23;"= ";X
7040 LET X=0
7045 FOR I=8 TO 1 STEP -1
7047 LET X=X+(CODE Y$(I)-28)*(2*
*(8-I))
7050 NEXT I
7055 PRINT AT 15,23;"= ";X
7060 RETURN
9000 DIM A$(9,8)
9010 LET A$(1)="RLCA/RLC"
9020 LET A$(2)="RLA/RL"
9030 LET A$(3)="RRCA/RRC"
9040 LET A$(4)="RRA/RR"
9050 LET A$(5)="SLA"
9060 LET A$(6)="SRL"
9070 LET A$(7)="SRA"
9080 LET A$(8)="RLD"
9090 LET A$(9)="RRD"
9100 LET B$="DEMONSTRATION OF SH
9110 LET C$="      "
9120 LET P=4000
9130 RETURN
```



High quality printing at a reasonable cost

Although the ZX-printer is comparatively cheap it has its limitations for heavy use. John Cussons shows how to make a RS 232 interface to allow a better printer to be used with the ZX-81 with at least 4K RAM.

ALTHOUGH the Sinclair printer is excellent value at £60, it has a number of limitations which make it less than ideal for protracted heavy-duty use. The metallised paper which it uses becomes expensive if you need reams of output instead of just the odd program listing, and the 32-column format is not really suitable for word processing. In addition, the print quality tends to deteriorate from time to time, which can play havoc with machine code listings where absolute accuracy is vital.

The interface described was designed for use with a KSR 33 terminal which fortunately was acquired for £50, although they are generally sold at prices around the £100 mark. During 1960s and 1970s the KSR 33 was one of the mainstays of the computer industry and consequently many of them are available second-hand. They are rather slow and noisy by today's standards but produce clear upper-case print on 8.5mm. plain paper in up to 80-column format and are fairly robust.

If your are fortunate enough to own something more sophisticated in the way of printers, this interface should still be suitable provided that it uses an RS232 ASCII code input.

Construction cost should be less than £8, the most expensive item being a suitable rear edge connector adaptor to enable you to use the interface in conjunction with the RAM pack and other expansions. The software enables you to produce complete program listings, or listings between selected line numbers and to output messages and variables to the printer under program control in any desired column setting.

The task of converting parallel data to serial format is carried-out by software, which outputs one bit at a time on the least significant bit of the data bus (DO). With reference to figure one, this data is stored in a latch formed by two cross-coupled NAND gates, IC2(8,9,10) and IC2(11,12,13). Gates IC1(1,2,3) and IC1(4,5,6) are address decode logic, requiring that IORQ, WR and A2 be simultaneously low to produce a 'low' output on IC1(4). That 'low' allows gates IC1(8,9,10) and IC1(11,12,13) to gate data and 'not data' signals through to both halves of the latch—gate IC2(1,2,3) acts as an invertor to provide the 'not data' signal.

IC3 is an op-amp which converts the OV and 5V logic levels to +10V and −10V as required by the printer.

The most straightforward method of construction is to use 0.2in. Veroboard, the layout illustrated being on a piece 3.5in. by 3.75in. That is a fairly generous size to make construction easier and you could probably condense the layout on to a much smaller board if required.

It is best first to solder the two ICs to the board, followed by VR1 and the two resistors, since they can then be used as a guide to the correct location of the PVC insulated wire links. That sequence has the disadvantage that the ICs could be overheated when soldering nearby links, so care should be exercised. If you do not wish to take the risk, the ICs could be replaced by DIL sockets into which the ICs are plugged only when construction is complete.

All the tracks which pass beneath the three ICs should be cut to avoid shorting IC pins and another track cut should be made between VR1 upper connection and IC3 pin 2 in the position marked.

Inputs from the ZX-81 are on the left hand side and they should be taken to the appropriate points on the edge connector adaptor. Multicore cable is ideal for it but might be difficult to obtain in short lengths, in which case you could use a bundle of separate wires held neatly together with insulating tape or cable ties. The connections should be kept short, 1ft. or less, since they are effectively an extension of the CPU bus which is sensitive to electrical noise.

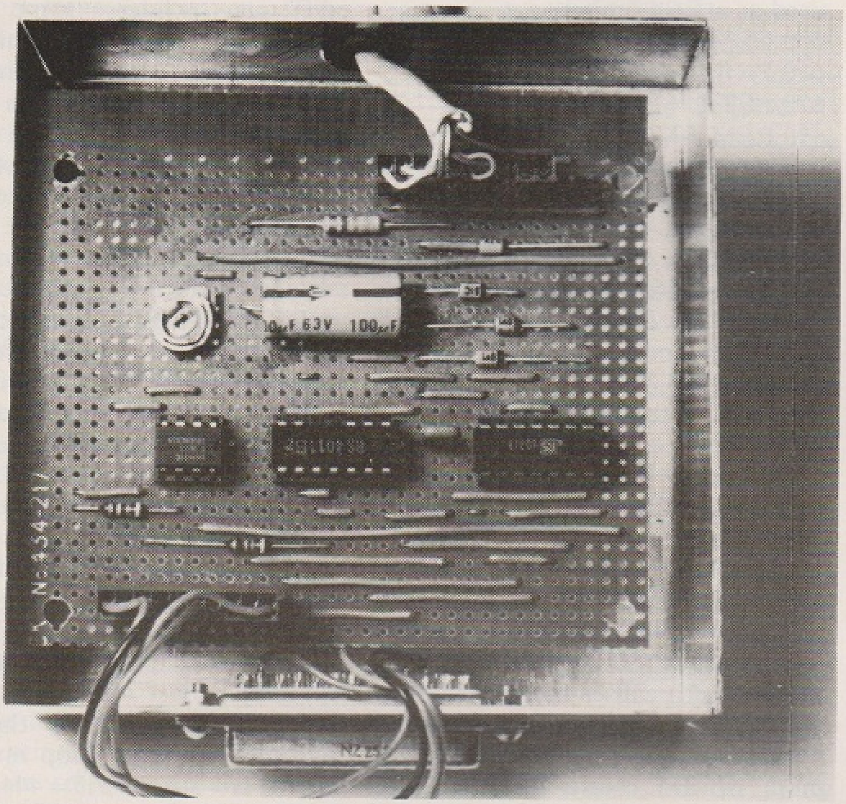
Two connections on the right hand side, 'data out' and 'common', go to the printer. The standard RS232 connector is a 25-way 'D' type plug, and 'data out' should be taken to pin 3, 'common' to pin 7.

The +10V and −10V power supplies for IC3 are only nominal values, since an RS232 interface should function in the range −3V:0:+3V to −15V:0:+15V. A pair of small 9V batteries therefore would make a suitable power supply; alternatively you could make up a mains power supply. Whatever you use the OV rail should be connected to the 'common' track.

In the case of the KSR 33, +10V and −10V are available on the selector magnet drive board and they could be brought-out externally and connected to the interface. That should be undertaken, however, only by an experienced constructor with access to the printer technical manual.

Potentiometer VR1 provides a reference level for IC3 and that should be adjusted so that the voltage on IC3 pin 3 is +2.5V.

The program consists of 1,049 bytes of machine code which initially



```
P9000 PRINT "INPUT STARTING
ADDRESS"
9010 DIM Y$(3)
9020 INPUT A
9030 FOR X=A TO 17559 STEP 5
9040 SCROLL
9050 LET CS=0
9060 PRINT AT 20,0;" ";
9070 FOR Y=1 TO 5
9080 INPUT Y$
9090 PRINT Y$;" ";
9100 POKE A,VAL Y$
9110 LET A=A+1
9120 LET CS=CS+VAL Y$
9130 NEXT Y
9140 PRINT "(";C$;")"
9150 PRINT AT 21,0;"IS THE CHECK
 SUM CORRECT?(Y OR N)"
9160 LET B$=INKEY$
9170 IF B$<>"N" AND B$<>"Y" THEN
 GOTO 9160
9180 PRINT AT 21,0;"
9190 IF B$="N" THEN GOTO 9230
9200 NEXT X
9210 PRINT AT 21,0;"LOADING COMP
LETE"
9220 STOP
9230 PRINT AT 20,0;"
                              "
9240 LET A=A-5
9250 GOTO 9050
  10 SAVE "PRINTER DRIVE"
  20 POKE 16389,123
  30 LET L=USR 16514
  40 PRINT "PRINTER DRIVE ROUTIN
 LOADED OK. ENTER ""NEW"" COMMAN
D TO RESET   RAMTOP."
  50~STOP
```

is resident in a REM statement in line, then down-loaded to above RAMTOP. There is an 'organiser' routine for program listings and another for writing under program control, called LISORG and RITORG. The organisers work by making calls to a common set of subroutines, performing 'housekeeping' tasks such as stack control where necessary.

Conversation of Sinclair codes to ASCII codes is done with the aid of two look-up tables, one for upper-case characters and one for lower-case—the latter is not used for the KSR 33, which produces upper-case print only . A more detailed description of the various routines is in the appendix.

To load the software, enter a REM statement at line number 1 containing 44 dummy characters—Xs, Ys or anything else. The total length of the line is then 50 bytes, allowing five bytes for the line overhead—two for line number, two for line length and one for final NL character—and one byte for the REM code. Use the edit facilities to produce a copy of this line and repeat until you have a total of 20 lines numbered 1 to 20. Enter line 21 with 49 dummy characters, then use the following direct commands:

POKE 16511,27

POKE 16512,4

That effectively converts the 21 lines you have entered into a single large REM statement with 1,049 bytes of space available for machine code storage; when listed it will still appear to be separate lines and the cursor may finish occasionally in the middle of it, but do not worry about that.

A word of caution at this point; when editing programs containing REM statements bigger than 768 bytes, never delete the line immediately following the REM statement, since that causes the ROM to be caught in an endless LIST/CLS loop and the only way of recovering is to pull out the power supply jack plug.

You should then enter the loader program—program 1—and use it to enter machine code into the REM statement from the decimal listing—figure four. The numbers in brackets after each block of five bytes are checksums and they should not be entered but simply compared to the checksums produced by the loader. After each block the program will output: "Is the checksum correct? (Y or N)". Pressing Y allows you to enter the next block and pressing N erases the last block and allows you to re-enter it.

Once all the machine code is entered, key-in the Basic section of the printer drive routine, lines 10 to 50—program 2. The machine code loader can then be deleted and the completed routine saved on tape using the command GOTO 10.

The format for using the printer is a program line with a USR call to either LISORG or RITORG, followed immediately by a REM statement containing appropriate arguments. FAST mode must always be entered before calling LISORG or RITORG.

The listing

10 LET L = USR 31930

20 REM 0,9999, will list the the entire Basic program in memory. If you wish to list only part of a program, insert the appropriate start and finish line numbers in the REM statement. Note that both commas must be present in this statement—it is very

RS232 INTERFACE

easy to omit the final comma which will result in an incorrect listing.

Graphics characters in the listing will be replaced by spaces, so that you have the opportunity to draw them by hand should you desire. Printer output for program listings is in 32 columns, the same as the TV screen. Although more columns could be used, it is easier to check listings against the screen if both use the same format. Sinclair ZX-81 Basic does not allow multi-statement lines, and most lines are consequently short; an 80-column output, therefore, would consist mainly of short lines with an occasional long line, which looks very untidy.

Terminating a program listing manually can be done by holding-down the Break key; printing then finishes at the end of the current line.

General-purpose output to the printer, such as strings, variables, messages, must first be written to the display file, then copied to the printer as follows:

```
500 LET L=USR 31690
501 REM 60,10,
```

Here line 500 calls RITORG, while line 501 sets the number of columns to 60 and copies the first 10 characters of the display file top line. If you wanted to copy the complete top line you should use:

```
501 REM 60,32,
```

The limit to the size of the second argument is 768 characters, i.e., the size of the complete display file. Using that value is equivalent to the Copy command.

Calling RITORG at address 31690 causes the printhead position to remain unchanged between calls—i.e., a carriage return/line feed occurs only when the specified number of characters have been printed. To start each print output with a CR/LF, line 500 should be amended to:

```
500 LET L=USR 31698.
```

There may be occasions on which the number of characters to be sent to the printer is unknown. Loading argument two in the REM statement with a large value is one way of dealing with it, but the print may then contain unwanted trailing spaces. To solve the problem the '£' character is used as a field limiter, e.g.:

```
600 CLS
610 PRINT AT 0,0;X;CH$ 12
620 LET L=USR 31690
630 REM 60,768,
```

That program will not copy the complete screen; instead, print output is terminated when all the characters of variable X have been printed. It is possible to alter the particular character used for limiting by entering in command mode:

```
POKE 31775,Y
```

where Y is the code of the desired character.

Outputting blank lines to the printer for the purpose of spacing can be achieved by:

```
1010 LET L=USR 31698
1020 REM 0,5,
```

Those two lines cause two CR/LFs to be output. Setting the first argument to zero causes the second argument to be interpreted as the desired number of CR/LFs, up to a maximum of 255. Note that RITORG must be called at address 31698 to use this facility.

The drive routine was written specifically for a KSR 33 operating at 110 baud, equivalent to 10 characters per second—11 bits/char. To drive different printers some software modifications may be required.

The baud rate is determined by the DELAY subroutine which is called-up after outputting each bit of a character. DELAY consists of two nested loops, an inner one which is executed 35 times and an outer one which is executed nine times. Those values produce the required 9.1msec delay for 110 baud operation but can be changed to give different baud rates, using the outer loop for coarse tuning and the inner one for fine tuning. The following calculations illustrate this:

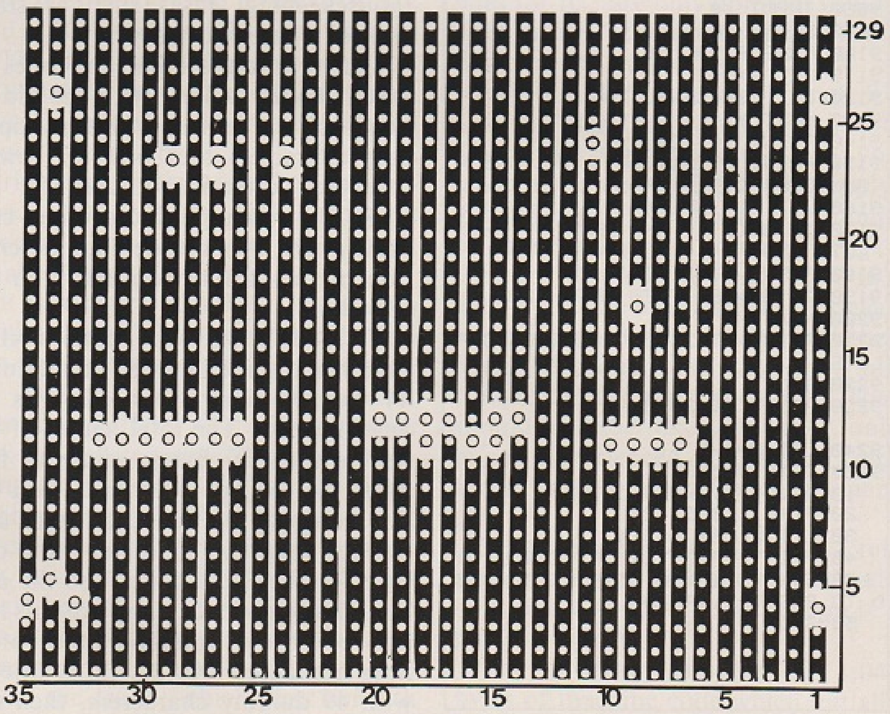
Total no. of times inner loop is executed is $9 \times 135 = 1,215$.

This corresponds to 9.1msec delay, thus inner loop execution time is 9.1 $1,215 = 7.5\mu$ sec.

For a 300 baud rate, say, the required bit time is 1/300 sec. = 3.333msec. Thus the number of times the inner loop must be executed is $= 3,333 \div 7.5 = 444$.

That could be achieved with an outer loop value of five, and an inner value of 89, or an outer value of six and an inner value 74. Which of the many possible combinations you choose is arbitrary but the inner loop

Figure 1: Veroboard.



will require some final adjustment to match the printer exactly, since the calculations are only approximate. The final trimming is done on a trial-and-error basis. Note that the range for the loop values is 1 to 255.

To alter the inner loop, POKE location 17370 and for the outer POKE location 17376. To down-load the modification to RAMTOP use the command RAND USR 16514.

The software includes no provision for setting parity; the first three bytes of the character output routine—CHOUT—merely set the parity bit to 1. Should you require to set odd or even parity, those three bytes should be replaced by a call to a subroutine. Most printers have settable parity—i.e., odd, even or none—and the last option should be chosen if available.

A carriage return/line feed combination on the KSR-33 is achieved by outputting decimal 10 to the interface but some printers may execute only a line feed in response to it. If that occurs check your printer manual; you may find that the CR/LF combination response is available as a hardware option, either on switches or wire links.

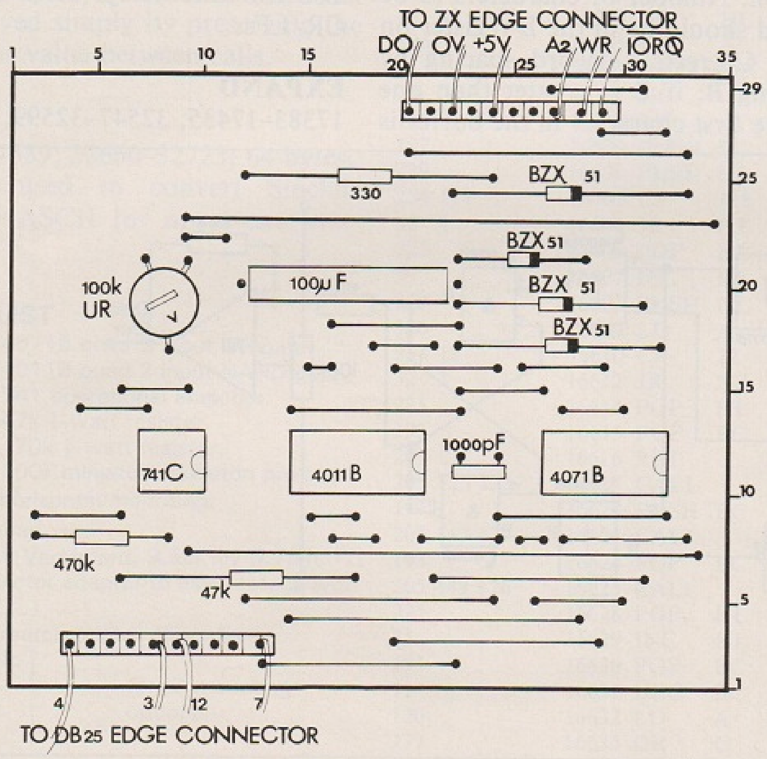
If you are unfortunate you might have to modify LISORG and RITORG to output a CR character—decimal 13—after each LF character—decimal 10. To find the locations where that is necessary, check the listings for the instruction LD A,10 followed by CALL 32275.

Look-up tables are used by the translation subroutine, STRANS, to convert Sinclair codes to ASCII. There are two tables, one for upper- and one for lower-case, both 64 bytes long. STRANS adds the Sinclair code to the table base address; the resulting address then contains the ASCII equivalent code.

Location 32725 contains a flag, CSFLAG, which determines whether or not the lower-case table should be used by STRANS. If CSFLAG is not equal to zero, both tables are used; inverse characters in the display file are then output to the printer as upper-case and normal characters as lower-case.

If CSFLAG is zero, however, only the upper-case table is used, both inverse and normal characters being printed as upper-case. To use lower-case, therefore, you must enter the command POKE 32725,1.

Figure 2: Top of PCB.



Appendix

Each of the programs following has two addresses, a REM address at which it is initially resident and a RAMTOP address which it occupies after down-loading; to convert between those addresses add or subtract 15164. The dissembled program listing gives REM addresses. A few of the instructions, those prefixed ED or CB, are not dissembled, since the program used to produce the listings requires that they be done manually.

DOWNLOADER
16514–16525; 31678–31689 12 bytes.

Simple routine using the Z-80 LDIR instruction to transfer the rest of the routine to RAMTOP.

RITORG
16526–16636; 31690–31800; 111 bytes.

Organiser program for copying display file to printer.

LCLUT
16637–16700; 31801–31864; 64 bytes.

Table used to convert Sinclair code to ASCII lower-case. Identical to upper-case table (UCLUT) except that the addresses corresponding to letter codes have contents incremented by decimal 32—hex 20. Some printers may have slight variations from standard ASCII, particularly punctuation marks.

CSHIFT
16701–16749; 31865–31913; 49 bytes.

Routine to enter UCLUT or LCLUT address into the Sinclair translation routine — STRANS. CSHIFT is not used directly by the organisers but is called-up by the subroutine which loads Sinclair characters into the output buffer—SCLOAD. Uses location 32725 as CSFLAG. If CSFLAG iz zero, only UCLUT is used.

LISTING COMPLETE
16750–16765; 31914–31929; 16 bytes.

Sixteen-byte memory area containing above message, which is output after completing a program listing.

LISORG
16766–16960, 31930–32124; 195 bytes.

RS232 INTERFACE

Main organiser routine for program listing.

SCLOAD
16961–16994; 32125–32158; 34 bytes.

Subroutine to load Sinclair characters to the output buffer—32757–32767. Loads only a single character at a time, which must be in the A register when SCLOAD is called. If A register contains a keyword code, however, the complete expansion is loaded to the buffer. If A register contains a graphics code, loads a single space to buffer; control codes and unused codes are loaded as a Sinclair '?' character; otherwise the character is loaded unchanged. Uses subroutines EXPAND and CSHIFT. Returns with a count in the B register of the number of characters loaded to the buffer—normally one unless a keyword expanded.

GETARG
16995–17092; 32159–32256; 98 bytes.

Subroutine to obtain two arguments from the REM statement following the USR call to LISORG OR RITORG. Uses location 32746 to flag first or second argument. Returns with binary value of first argument in 32747 and 32748, value of second in 32749 and 32750. Uses Subroutine DECADE for multiplication by 10.

CHOUT
17093–17130; 32257–32294; 38 bytes.

Outputs a single character to the interface at address FB. The character must be in the A register when CHOUT is called. Outputs one start bit, eight data bits and two stop bits-(mark) on the least significant bit of the data bus. Uses DELAY to produce correct bit timings.

DUN
17131–17162; 32295–32326; 32 bytes.

Program used by LISORG to terminate listings. When called DUN outputs 20 CR/LFs followed by the message 'Listing complete', then executes a 'dummy' POP to return program control to the Basic interpreter. Uses CHOUT and DELAY.

BINASC
17163–17240; 32327–32404; 78 bytes.

Subroutine which takes a binary value in the DE register pair and converts it into its ASCII equivalent in the output buffer. Returns with a count of the number of ASCII characters in the B register. Uses location 32745 as a leading zero suppression flag.

PRINT
17241–17322; 32405–32486; 82 bytes.

Prints the ASCII contents of the output buffer, which should be either a single character or a keyword expansion. Number of characters to be printed should be in the B register on entry. Corrects keyword spacing by checking B; if B is greater than one and the first character in the buffer is a space, will suppress this if the last character printed was also a space.

Formats printed lines by reference to COLSET and PCOUNT. COLSET (32754) is the number of columns specified in the REM statement—loaded in by the organiser; PCOUNT (32755) is the number of characters which reamin to be printed in the current line and is updated by PRINT before returning. Should PCOUNT go to zero, PRINT outputs a CR/LF and re-sets PCOUNT to equal COLSET. Uses DELAY when printing CR/LFs.

LOCLIN
17323–17364; 32487–32528; 42 bytes.

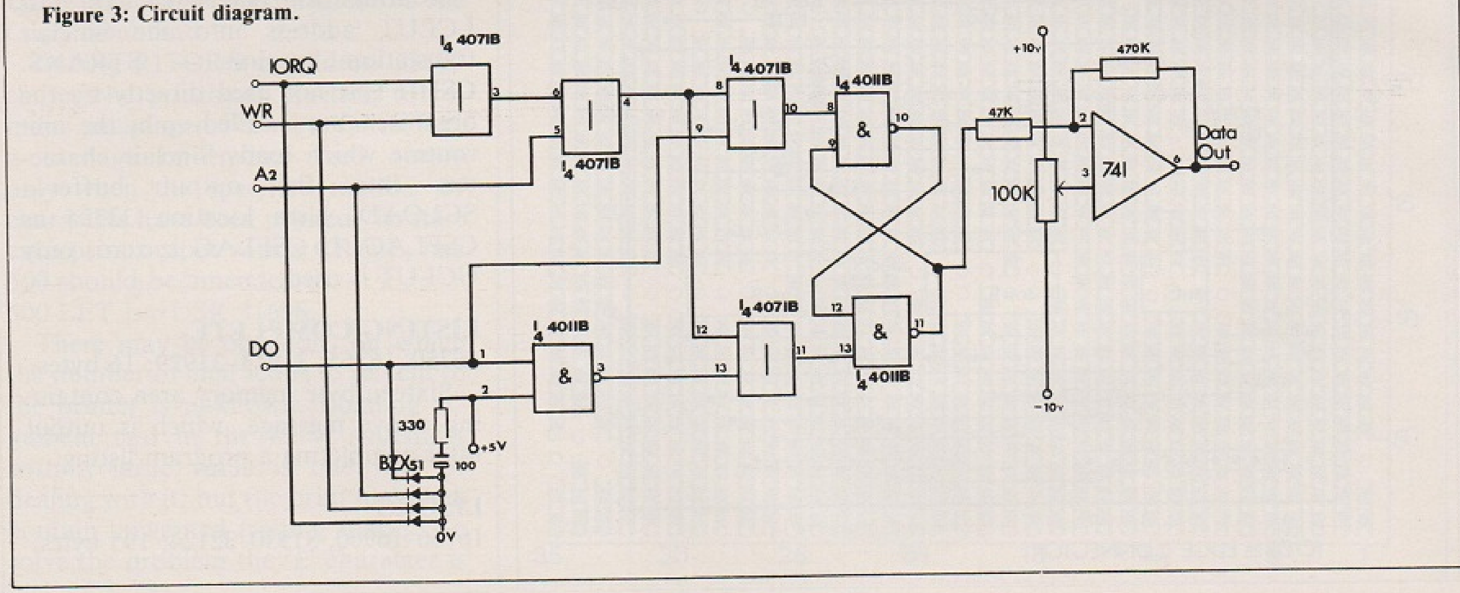
Used by LISORG to locate the first line to be listed, by searching the program file for a match with the binary line number stored in location 32747 and 32748. Returns with the address of the line start in the HL register pair. Calls DUN to return to Basic if line not found.

DELAY
17365–17382; 32529–32546; 18 bytes.

Produces time delays in 9.1m sec; increments according to the value in the B register—i.e., if B = 10 on entry provides a 91m sec delay. Used to set the 'Bit time' for serial output and also for time delays after outputting CR/LFs.

EXPAND
17383–17435; 32547–32599; 53 bytes.

Figure 3: Circuit diagram.



RS232 INTERFACE

Takes a keyword code in the A register and loads the expanded code along with appropriate leading and trailing spaces to the output buffer. Uses monitor routine 2421 (decimal) to locate the expanded form from with the monitor tables.

CHECK
17436–17464; 32600–32628; 29 bytes.
Called by CSHIFT to check the nature of a character in the A register. Sets or re-sets the carry and zero flags before return to differentiate between printable characters, graphics characters, keyword codes or control codes.

STRANS
17465–17479; 32629–32643; 15 bytes.
Translates the output buffer contents from Sinclair to ASCII. Number of characters to be translated must be set into the B register before entry.

DECADE
17480–17495; 32644–32659; 16 bytes.
Multiplies the initial contents of the DE register pair by 10 to the power C, where C is the content of the C register on entry. Increments C before return so that multiplication in the sequence 1, 10, 100 and the like can be achieved simply by preserving the C register value between calls.

UCLUT
17496–17559; 32660–32723; 64 bytes.
Table used to convert Sinclair codes to ASCII for upper-case output.

PARTS LIST

IC1	—	4071B quad 2 input OR gate.
IC2	—	4011B quad 2 input NAND gate.
IC3	—	741 operational amplifier.
R1	—	47k $\frac{1}{4}$-watt resistor.
R2	—	470k $\frac{1}{4}$-watt resistor.
VR1	—	100k miniature skeleton pre-set, horizontal mounting.

Miscellaneous
0.1in. pitch Veroboard, 3.5in, by 3.75in.
Edge connector adaptor to use inteface with RAM pack.
Socket to match printer plug.
2 14 pin DIL sockets.
1 8-pin DIL socket.
} for IC mounting—optional.

DOWNLOADER

```
33  142 64     16514 LD    HL  16526
17  202 123    16517 LD    DE  31690
1   12  4      16520 LD    BC  1036
237 176        16523 (ED)
201            16525 RET
```

RITORG

```
24  0              16526 JR         0
62  51             16528 LD    A    51
50  203 123        16530 LD    (31691) A
0                  16533 NOP
205 159 125        16534 CALL       32159
33  235 127        16537 LD    HL   32747
126                16540 LD    A    (HL)
254 0              16541 CP    O
32  21             16543 JR    NZ   21
35                 16545 INC   HL
35                 16546 INC   HL
70                 16547 LD    B    (HL)
4                  16548 INC   B
24  12             16549 JR         12
197                16551 PUSH  BC
62  10             16552 LD    A    10
205 1   126        16554 CALL       32257
6   20             16557 LD    B    20
205 17  127        16559 CALL       32529
193                16562 POP   BC
16  242            16563 DJNZ       -14
201                16565 RET
50  243 127        16566 LD    (32755) A
62  10             16569 LD    A    10
205 1   126        16571 CALL       32257
6   20             16574 LD    B    20
205 17  127        16576 CALL       32529
205 159 125        16579 CALL       32159
58  235 127        16582 LD    A    (32747)
50  242 127        16585 LD    (32754) A
237 75  237 127    16588 (ED)
42  12  64         16592 LD    HL   (16396)
126                16595 LD    A    (HL)
3                  16596 INC   BC
24  32             16597 JR         32
126                16599 LD    A    (HL)
197                16600 PUSH  BC
229                16601 PUSH  HL
254 118            16602 CP    118
32  4              16604 JR    NZ   4
225                16606 POP   HL
35                 16607 INC   HL
229                16608 PUSH  HL
126                16609 LD    A    (HL)
254 12             16610 CP    12
32  3              16612 JR    NZ   3
225                16614 POP   HL
193                16615 POP   BC
201                16616 RET
205 125 125        16617 CALL       32125
197                16620 PUSH  BC
205 117 127        16621 CALL       32629
193                16624 POP   BC
205 149 126        16625 CALL       32405
225                16628 POP   HL
35                 16629 INC   HL
193                16630 POP   BC
11                 16631 DEC   BC
120                16632 LD    A    B
177                16633 OR    C
32  219            16634 JR    NX   -37
201                16636 RET
```

CSHIFT

```
229                16701 PUSH  HL
245                16702 PUSH  AF
58  213 127        16703 LD    A    (32725)
254 0              16706 CP    O
32  12             16708 JR    NZ   12
33  148 127        16710 LD    HL   32660
34  122 127        16713 LD    (32634) HL
241                16716 POP   AF
225                16717 POP   HL
205 88  127        16718 CALL       32600
201                16721 RET
241                16722 POP   AF
254 166            16723 CP    166
56  12             16725 JR    C    12
254 192            16727 CP    192
48  8              16729 JR    NC   8
33  148 127        16731 LD    HL   32660
34  122 127        16734 LD    (32634) HL
24  6              16737 JR         6
33  57  124        16739 LD    HL   31801
34  122 127        16742 LD    (32634) HL
225                16745 POP   HL
205 88  127        16746 CALL       32600
201                16749 RET
```

LISORG

```
33  213 127        16766 LD    HL   32725
54  0              16769 LD    (HL)0
33  242 127        16771 LD    HL   32754
54  32             16774 LD    (HL)32
43                 16776 DEC   HL
43                 16777 DEC   HL
43                 16778 DEC   HL
54  0              16779 LD    (HL)0
6   5              16781 LD    B    5
62  10             16783 LD    A    10
197                16785 PUSH  BC
205 1   126        16786 CALL       32257
6   20             16789 LD    B    20
205 17  127        16791 CALL       32529
193                16794 POP   BC
16  242            16795 DJNZ       -14
205 159 125        16797 CALL       32159
205 231 126        16800 CALL       32487
229                16803 PUSH  HL
58  239 127        16804 LD    A    (32751)
254 0              16807 CP    O
32  10             16809 JR    NZ   10
62  10             16811 LD    A    10
205 1   126        16813 CALL       32257
6   40             16816 LD    B    40
205 17  127        16818 CALL       32529
225                16821 POP   HL
86                 16822 LD    D    (HL)
35                 16823 INC   HL
94                 16824 LD    E    (HL)
35                 16825 INC   HL
229                16826 PUSH  HL
205 71  126        16827 CALL       32327
197                16830 PUSH  BC
33  243 127        16831 LD    HL   32755
62  28             16834 LD    A    28
128                16836 ADD   A    B
119                16837 LD    (HL)A
```

RS232 INTERFACE

```
62 5            16838 LD   A    5
144             16840 SUB  B
71              16841 LD   B    A
24 7            16842 JR        7
62 32           16844 LD   A    32
197             16846 PUSH BC
205 1 126       16847 CALL      32257
193             16850 POP  BC
16 247          16851 DJNZ      -9
193             16853 POP  BC
205 149 126     16854 CALL      32405
225             16857 POP  HL
94              16858 LD   E    (HL)
35              16859 INC  HL
86              16860 LD   D    (HL)
237 83 240 127  16861 (ED)
35              16865 INC  HL
126             16866 LD   A    (HL)
254 126         16867 CP   126
32 18           16869 JR   NZ   18
17 6 0          16871 LD   DE   6
25              16874 ADD  HL   DE
68              16875 LD   B    H
77              16876 LD   C    L
42 240 127      16877 LD   HL   (32752)
167             16880 AND  A
237 82          16881 (ED)
34 240 127      16883 LD   (32752) HL
96              16886 LD   H    B
105             16887 LD   L    C
126             16888 LD   A    (HL)
237 91 240 127  16889 (ED)
27              16893 DEC  DE
237 83 240 127  16894 (ED)
71              16898 LD   B    A
122             16899 LD   A    D
179             16900 OR   E
40 17           16901 JR   Z    17
120             16903 LD   A    B
229             16904 PUSH HL
205 125 125     16905 CALL      32125
197             16908 PUSH BC
205 117 127     16909 CALL      32629
193             16912 POP  BC
205 149 126     16913 CALL      32405
225             16916 POP  HL
35              16917 INC  HL
24 202          16918 JR        -54
35              16920 INC  HL
237 91 12 64    16921 (ED)
235             16925 EXX  DE   HL
167             16926 AND  A
237 82          16927 (ED)
235             16929 EXX  DE   HL
204 39 126      16930 CALL Z    32295
86              16933 LD   D    (HL)
35              16934 INC  HL
94              16935 LD   E    (HL)
235             16936 EXX  DE   HL
68              16937 LD   B    H
77              16938 LD   C    L
42 237 127      16939 LD   HL   (32749)
167             16942 AND  A
237 66          16943 (ED)
220 39 126      16945 CALL C    32295
235             16948 EXX  DE   HL
43              16949 DEC  HL
229             16950 PUSH HL
205 187 2       16951 CALL      699
124             16954 LD   A    H
173             16955 XOR  L
225             16956 POP  HL
192             16957 RET  NZ
195 223 124     16958 JP        31967
```

SCLOAD

```
205 121 124     16961 CALL      31865
122             16964 LD   A    D
33 245 127      16965 LD   HL   32757
6 1             16968 LD   B    1
40 7            16970 JR   Z    7
56 2            16972 JR   C    2
119             16974 LD   (HL)A
201             16975 RET
54 0            16976 LD   (HL)0
201             16978 RET
56 3            16979 JR   C    3
54 15           16981 LD   (HL)15
201             16983 RET
205 35 127      16984 CALL      32547
17 245 127      16987 LD   DE   32757
167             16990 AND  A
237 82          16991 (ED)
69              16993 LD   B    L
201             16994 RET
```

GETARG

```
175             16995 XOR  A
111             16996 LD   L    A
103             16997 LD   H    A
34 235 127      16998 LD   (32747) HL
34 237 127      17001 LD   (32749) HL
34 234 127      17004 LD   (32746) HL
42 41 64        17007 LD   HL   (16425)
17 5 0          17010 LD   DE   5
25              17013 ADD  HL   DE
22 0            17014 LD   D    0
74              17016 LD   C    D
6 7             17017 LD   B    7
16 2            17019 DJNZ      2
55              17021 SCF
201             17022 RET
126             17023 LD   A    (HL)
254 26          17024 CP   26
40 4            17026 JR   Z    4
20              17028 INC  D
35              17029 INC  HL
24 243          17030 JR        -13
229             17032 PUSH HL
43              17033 DEC  HL
126             17034 LD   A    (HL)
214 28          17035 SUB  28
213             17037 PUSH DE
22 0            17038 LD   D    0
95              17040 LD   E    A
229             17041 PUSH HL
205 132 127     17042 CALL      32644
58 234 127      17045 LD   A    (32746)
254 2           17048 CP   2
32 9            17050 JR   NZ   9
42 237 127      17052 LD   HL   (32749)
25              17055 ADD  HL   DE
34 237 127      17056 LD   (32749) HL
24 7            17059 JR        7
42 235 127      17061 LD   HL   (32747)
25              17064 ADD  HL   DE
34 235 127      17065 LD   (32747) HL
225             17068 POP  HL
209             17069 POP  DE
122             17070 LD   A    D
145             17071 SUB  C
32 215          17072 JR   NZ   -41
58 234 127      17074 LD   A    (32746)
254 2           17077 CP   2
32 3            17079 JR   NZ   3
225             17081 POP  HL
167             17082 AND  A
201             17083 RET
62 2            17084 LD   A    2
50 234 127      17086 LD   (32746) A
225             17089 POP  HL
35              17090 INC  HL
24 177          17091 JR        -79
```

CHOUT

```
203 255         17093 (CB)
0               17095 NOP
245             17096 PUSH AF
175             17097 XOR  A
211 251         17098 OUT  251  A
6 1             17100 LD   B    1
205 17 127      17102 CALL      32529
241             17105 POP  AF
6 8             17106 LD   B    8
211 251         17108 OUT  251  A
197             17110 PUSH BC
6 1             17111 LD   B    1
205 17 127      17113 CALL      32529
203 47          17116 (CB)
193             17118 POP  BC
16 243          17119 DJNZ      -13
203 199         17121 (CB)
211 251         17123 OUT  251  A
6 2             17125 LD   B    2
205 17 127      17127 CALL      32529
201             17130 RET
```

DUN

```
6 20            17131 LD   B    20
197             17133 PUSH BC
62 10           17134 LD   A    10
205 1 126       17136 CALL      32257
6 20            17139 LD   B    20
205 17 127      17141 CALL      32529
193             17144 POP  BC
16 242          17145 DJNZ      -14
33 170 124      17147 LD   HL   31914
6 16            17150 LD   B    16
126             17152 LD   A    (HL)
197             17153 PUSH BC
205 1 126       17154 CALL      32257
193             17157 POP  BC
35              17158 INC  HL
16 247          17159 DJNZ      -9
241             17161 POP  AF
201             17162 RET
```

BINASC

```
175             17163 XOR  A
50 233 127      17164 LD   (32745) A
71              17167 LD   B    A
79              17168 LD   C    A
33 245 127      17169 LD   HL   32757
229             17172 PUSH HL
33 232 3        17173 LD   HL   1000
```

```
235              17176 EXX    DE  HL
167              17177 AND    A
237 82           17178 (ED)
56  8            17180 JR     C    8
12               17182 INC    C
62  1            17183 LD     A    1
50  233 127      17185 LD     (32745) A
24  243          17188 JR          -13
25               17190 ADD    HL  DE
58  233 127      17191 LD     A    (32745)
254 0            17194 CP     0
40  14           17196 JR     Z    14
229              17198 PUSH   HL
225              17199 POP    HL
225              17200 POP    HL
4                17201 INC    B
121              17202 LD     A    C
198 48           17203 ADD    A    48
119              17205 LD     (HL)A
35               17206 INC    HL
229              17207 PUSH   HL
59               17208 DEC    SP
59               17209 DEC    SP
225              17210 POP    HL
14  0            17211 LD     C    0
22  0            17213 LD     D    0
203 123          17215 (CB)
40  4            17217 JR     Z    4
30  100          17219 LD     E    100
24  211          17221 JR          -45
203 83           17223 (CB)
40  4            17225 JR     Z    4
30  10           17227 LD     E    10
24  203          17229 JR          -53
203 75           17231 (CB)
40  4            17233 JR     Z    4
30  1            17235 LD     E    1
24  195          17237 JR          -61
225              17239 POP    HL
201              17240 RET
```

PRINT

```
58  243 127      17241 LD     A    (32755)
87               17244 LD     D    A
33  245 127      17245 LD     HL   32757
16  58           17248 DJNZ        58
4                17250 INC    B
88               17251 LD     E    B
126              17252 LD     A    (HL)
205 1   126      17253 CALL        32257
1   239 127      17256 LD     BC   32751
175              17259 XOR    A
2                17260 LD     (BC) A
66               17261 LD     B    D
16  19           17262 DJNZ        19
62  10           17264 LD     A    10
205 1   126      17266 CALL        32257
1   239 127      17269 LD     BC   32751
2                17272 LD     (BC) A
0                17273 NOP
6   40           17274 LD     B    40
205 17  127      17276 CALL        32529
58  242 127      17279 LD     A    (32754)
71               17282 LD     B    A
35               17283 INC    HL
80               17284 LD     D    B
67               17285 LD     B    E
16  219          17286 DJNZ        -37
122              17288 LD     A    D
50  243 127      17289 LD     (32755) A
62  1            17292 LD     A    1
50  244 127      17294 LD     (32756) A
43               17297 DEC    HL
126              17298 LD     A    (HL)
254 32           17299 CP     32
200              17301 RET         Z
62  0            17302 LD     A    0
50  244 127      17304 LD     (32756) A
201              17307 RET
58  244 127      17308 LD     A    (32756)
254 1            17311 CP     1
32  191          17313 JR     NZ   -65
126              17315 LD     A    (HL)
254 32           17316 CP     32
32  186          17318 JR     NZ   -70
35               17320 INC    HL
24  184          17321 JR          -72
```

LOCLIN

```
17  125 64       17323 LD     DE   16509
26               17326 LD     A    (DE)
103              17327 LD     H    A
19               17328 INC    DE
26               17329 LD     A    (DE)
111              17330 LD     L    A
237 75  235 127  17331 (ED)
167              17335 AND    A
237 66           17336 (ED)
48  22           17338 JR     NC   22
19               17340 INC    DE
26               17341 LD     A    (DE)
79               17342 LD     C    A
19               17343 INC    DE
26               17344 LD     A    (DE)
71               17345 LD     B    A
235              17346 EXX    DE   HL
9                17347 ADD    HL   BC
35               17348 INC    HL
235              17349 EXX    DE   HL
42  12  64       17350 LD     HL   (16396)
167              17353 AND    A
237 82           17354 (ED)
32  224          17356 JR     NZ   -32
241              17358 POP    AF
205 39  126      17359 CALL        32295
27               17362 DEC    DE
235              17363 EXX    DE   HL
201              17364 RET
```

DELAY

```
197              17365 PUSH   BC
6   9            17366 LD     B    9
197              17368 PUSH   BC
6   135          17369 LD     B    135
0                17371 NOP
0                17372 NOP
0                17373 NOP
16  251          17374 DJNZ        -5
193              17376 POP    BC
16  245          17377 DJNZ        -11
193              17379 POP    BC
16  239          17380 DJNZ        -17
201              17382 RET
```

EXPAND

```
245              17383 PUSH   AF
33  245 127      17384 LD     HL   32757
254 217          17387 CP     217
40  8            17389 JR     Z    8
254 218          17391 CP     218
40  4            17393 JR     Z    4
254 222          17395 CP     222
56  4            17397 JR     C    4
6   0            17399 LD     B    0
112              17401 LD     (HL)B
35               17402 INC    HL
205 117 9        17403 CALL        2421
10               17406 LD     A    (BC)
203 127          17407 (CB)
203 191          17409 (CB)
119              17411 LD     (HL)A
3                17412 INC    BC
35               17413 INC    HL
40  246          17414 JR     Z    -10
241              17416 POP    AF
254 216          17417 CP     216
200              17419 RET         Z
254 219          17420 CP     219
200              17422 RET         Z
254 220          17423 CP     220
200              17425 RET         Z
254 221          17426 CP     221
200              17428 RET         Z
254 193          17429 CP     193
216              17431 RET    C
175              17432 XOR    A
119              17433 LD     (HL)A
35               17434 INC    HL
201              17435 RET
```

CHECK

```
245              17436 PUSH   AF
203 119          17437 (CB)
40  8            17439 JR     Z    8
203 127          17441 (CB)
40  13           17443 JR     Z    13
175              17445 XOR    A
63               17446 CCF
209              17447 POP    DE
201              17448 RET
209              17449 POP    DE
203 186          17450 (CB)
122              17452 LD     A    D
254 11           17453 CP     11
192              17455 RET    NZ
183              17456 OR     A
201              17457 RET
254 43           17458 CP     43
48  239          17460 JR     NC   -17
175              17462 XOR    A
209              17463 POP    DE
201              17464 RET
```

STRANS

```
33  245 127      17465 LD     HL   32757
126              17468 LD     A    (HL)
17  57  124      17469 LD     DE   31801
131              17472 ADD    A    E
95               17473 LD     E    A
26               17474 LD     A    (DE)
119              17475 LD     (HL)A
35               17476 INC    HL
16  245          17477 DJNZ        -11
201              17479 RET
```

DECADE

```
12               17480 INC    C
65               17481 LD     B    C
16  1            17482 DJNZ        1
201              17484 RET
235              17485 EXX    DE   HL
41               17486 ADD    HL   HL
84               17487 LD     D    H
93               17488 LD     E    L
41               17489 ADD    HL   HL
41               17490 ADD    HL   HL
25               17491 ADD    HL   DE
84               17492 LD     D    H
93               17493 LD     E    L
24  242          17494 JR          -14
```

RS232 INTERFACE

16514	33	142	64	17	202	(458)
16519	123	1	12	4	237	(377)
16524	176	201	24	0	62	(463)
16529	51	50	203	123	0	(427)
16534	205	159	125	33	235	(757)
16539	127	126	254	0	32	(539)
16544	21	35	35	70	4	(165)
16549	24	12	197	62	10	(305)
16554	205	1	126	6	20	(358)
16559	205	17	127	193	16	(558)
16564	242	201	50	243	127	(863)
16569	62	10	205	1	126	(404)
16574	6	20	205	17	127	(375)
16579	205	159	125	58	235	(782)
16584	127	50	242	127	237	(783)
16589	75	237	127	42	12	(493)
16594	64	126	3	24	32	(249)
16599	126	197	229	254	118	(924)
16604	32	4	225	35	229	(525)
16609	126	254	12	32	3	(427)
16614	225	193	201	205	125	(949)
16619	125	197	205	117	127	(771)
16624	193	205	149	126	225	(898)
16629	35	193	11	120	177	(536)
16634	32	219	201	32	0	(484)
16639	0	0	0	0	0	(0)
16644	0	0	0	0	34	(34)
16649	35	36	58	63	40	(232)
16654	41	62	60	61	43	(267)
16659	45	42	47	59	44	(237)
16664	46	48	49	50	51	(244)
16669	52	53	54	55	56	(270)
16674	57	97	98	99	100	(451)
16679	101	102	103	104	105	(515)
16684	106	107	108	109	110	(540)
16689	111	112	113	114	115	(565)
16694	116	117	118	119	120	(590)
16699	121	122	229	245	58	(775)
16704	213	127	254	0	32	(626)
16709	12	33	148	127	34	(354)
16714	122	127	241	225	205	(920)
16719	88	127	201	241	254	(911)
16724	166	56	12	254	192	(680)
16729	48	8	33	148	127	(364)
16734	34	122	127	24	6	(313)
16739	33	57	124	34	122	(370)
16744	127	225	205	88	127	(772)
16749	201	76	73	83	84	(517)
16754	73	78	71	32	67	(321)
16759	79	77	80	76	69	(381)
16764	84	69	33	213	127	(526)
16769	54	0	33	242	127	(456)
16774	54	32	43	43	43	(215)
16779	54	0	6	5	62	(127)
16784	10	197	205	1	126	(539)
16789	6	20	205	17	127	(375)
16794	193	16	242	205	159	(815)
16799	125	205	231	126	229	(916)
16804	58	239	127	254	0	(678)
16809	32	10	62	10	205	(319)
16814	1	126	6	40	205	(378)
16819	17	127	225	86	35	(490)
16824	94	35	229	205	71	(634)
16829	126	197	33	243	127	(726)
16834	62	28	128	119	62	(399)
16839	5	144	71	24	7	(251)
16844	62	32	197	205	1	(497)
16849	126	193	16	247	193	(775)
16854	205	149	126	225	94	(799)
16859	35	86	237	83	240	(681)
16864	127	35	126	254	126	(668)
16869	32	18	17	6	0	(73)
16874	25	68	77	42	240	(452)
16879	127	167	237	82	34	(647)
16884	240	127	96	105	126	(694)
16889	237	91	240	127	27	(722)
16894	237	83	240	127	71	(758)
16899	122	179	40	17	120	(478)
16904	229	205	125	125	197	(881)
16909	205	117	127	193	205	(847)
16914	149	126	225	35	24	(559)
16919	202	35	237	91	12	(577)
16924	64	235	167	237	82	(785)
16929	235	204	39	126	86	(690)
16934	35	94	235	68	77	(509)
16939	42	237	127	167	237	(810)
16944	66	220	39	126	235	(686)
16949	43	229	205	187	2	(666)
16954	124	173	225	192	195	(909)
16959	223	124	205	121	124	(797)
16964	122	33	245	127	6	(533)
16969	1	40	7	56	2	(106)
16974	119	201	54	0	201	(575)
16979	56	3	54	15	201	(329)
16984	205	35	127	17	245	(629)
16989	127	167	237	82	69	(682)
16994	201	175	111	103	34	(624)
16999	235	127	34	237	127	(760)
17004	34	234	127	42	41	(478)
17009	64	17	5	0	25	(111)
17014	22	0	74	6	7	(109)
17019	16	2	55	201	126	(400)
17024	254	26	40	4	20	(344)
17029	35	24	243	229	43	(574)
17034	126	214	28	213	22	(603)
17039	0	95	229	205	132	(661)
17044	127	58	234	127	254	(800)
17049	2	32	9	42	237	(322)
17054	127	25	34	237	127	(550)
17059	24	7	42	235	127	(435)
17064	25	34	235	127	225	(646)
17069	209	122	145	32	215	(723)
17074	58	234	127	254	2	(675)
17079	32	3	225	167	201	(628)
17084	62	2	50	234	127	(475)
17089	225	35	24	177	203	(664)
17094	255	0	245	175	211	(886)
17099	251	6	1	205	17	(480)
17104	127	241	6	8	211	(593)
17109	251	197	6	1	205	(660)
17114	17	127	203	47	193	(587)
17119	16	243	203	199	211	(872)
17124	251	6	2	205	17	(481)
17129	127	201	6	20	197	(551)
17134	62	10	205	1	126	(404)
17139	6	20	205	17	127	(375)
17144	193	16	242	33	170	(654)
17149	124	6	16	126	197	(469)
17154	205	1	126	193	35	(560)
17159	16	247	241	201	175	(880)
17164	50	233	127	71	79	(560)
17169	33	245	127	229	33	(667)
17174	232	3	235	167	237	(874)
17179	82	56	8	12	62	(220)
17184	1	50	233	127	24	(435)
17189	243	25	58	233	127	(686)
17194	254	0	40	14	229	(537)
17199	225	225	4	121	198	(773)
17204	48	119	35	229	59	(490)
17209	59	225	14	0	22	(320)
17214	0	203	123	40	4	(370)
17219	30	100	24	211	203	(568)
17224	83	40	4	30	10	(167)
17229	24	203	203	75	40	(545)
17234	4	30	1	24	195	(254)
17239	225	201	58	243	127	(854)
17244	87	33	245	127	16	(508)
17249	58	4	88	126	205	(481)
17254	1	126	1	239	127	(494)
17259	175	2	66	16	19	(278)
17264	62	10	205	1	126	(404)
17269	1	239	127	2	0	(369)
17274	6	40	205	17	127	(395)
17279	58	242	127	71	35	(533)
17284	80	67	16	219	122	(504)
17289	50	243	127	62	1	(483)
17294	50	244	127	43	126	(590)
17299	254	32	200	62	0	(548)
17304	50	244	127	201	58	(680)
17309	244	127	254	1	32	(658)
17314	191	126	254	32	32	(635)
17319	186	35	24	184	17	(446)
17324	125	64	26	103	19	(337)
17329	26	111	237	75	235	(684)
17334	127	167	237	66	48	(645)
17339	22	19	26	79	19	(165)
17344	26	71	235	9	35	(376)
17349	235	42	12	64	167	(520)
17354	237	82	32	224	241	(816)
17359	205	39	126	27	235	(632)
17364	201	197	6	9	197	(610)
17369	6	135	0	0	0	(141)
17374	16	251	193	16	245	(721)
17379	193	16	239	201	245	(894)
17384	33	245	127	254	217	(876)
17389	40	8	254	218	40	(560)
17394	4	254	222	56	4	(540)
17399	6	0	112	35	205	(358)
17404	117	9	10	203	127	(466)
17409	203	191	119	3	35	(551)
17414	40	246	241	254	216	(997)
17419	200	254	219	200	254	(1127)
17424	220	200	254	221	200	(1095)
17429	254	193	216	175	119	(957)
17434	35	201	245	203	119	(803)
17439	40	8	203	127	40	(418)
17444	13	175	63	209	201	(661)
17449	209	203	186	122	254	(974)
17454	11	192	183	201	254	(841)
17459	43	48	239	175	209	(714)
17464	201	33	245	127	126	(732)
17469	17	57	124	131	95	(424)
17474	26	119	35	16	245	(441)
17479	201	12	65	16	1	(295)
17484	201	235	41	84	93	(654)
17489	41	41	25	84	93	(284)
17494	24	242	32	0	0	(298)
17499	0	0	0	0	0	(0)
17504	0	0	0	34	35	(69)
17509	36	58	63	40	41	(238)
17514	62	60	61	43	45	(271)
17519	42	47	59	44	46	(238)
17524	48	49	50	51	52	(250)
17529	53	54	55	56	57	(275)
17534	65	66	67	68	69	(335)
17539	70	71	72	73	74	(360)
17544)	75	76	77	78	79	(385)
17549	80	81	82	83	84	(410)
17554	85	86	87	88	89	(435)
17559	90	0	0	0	0	(90)

Good idea is damaged by a number of errors

We have received a large number of comments and interesting points about our first issue. Here David Buckley answers many of the more important questions raised.

AS AN OWNER of a ZX-81, I have bought *Sinclair User* since its first issue, in the hope that you would deal with add-ons for home construction. So I was delighted to read about the advent of *Sinclair Projects;* it sounded just what I wanted. So I was disappointed, not by the contents but by the mistakes.

Page 14, figure 5. $\overline{\text{IORQ}}$ and $\overline{\text{RD}}$ transposed; $\overline{\text{RD}}$ is pin 18.

Page 16, figure 2B. Pin C is common to both joysticks; therefore the lead from the top left should go to pin C and the wiper to pin B.

Page 20, figure 1. $\overline{\text{MBEQ}}$ on the connector should be $\overline{\text{MREQ}}$. The 74LS373 is omitted from the parts list and the 74LS04 is omitted from the cost list.

Page 22, line 5, 3682 is incorrect.

Page 27, figure 7. The motor when in reverse will run until the battery runs down. Many figure numbers do not agree with those in the text.

Page 28, figure 12. I suggest this is added to the soldering article and entitled How not to Solder. Figure 11 looks as if they are removing the solder bridges shown in figure 12. Figure 14—connections to 12-way connector strip are incorrectly marked.

Please ask Dave Buckley where he can get edge connectors for £2.25 and extender cards for 25p; I should be interested. Incidentally, Watford seems the cheapest for 74LS373 at 60p plus VAT.

D. S. Anthes,
Bridport,
Dorset.

● *Thanks for your comments. Unfortunate production problems meant that unchecked diagrams were published. Regarding the edge connector, £2.25 was the Maplin catalogue price in September. In the new catalogue it is £2.39. The extender card was obtained from Technomatic; unfortunately it now seems to be discontinued.*

Soldering on

THERE IS great need for a magazine to cater mainly for owners of Sinclair computers, so news of the *Sinclair Projects* was welcome and the first issue eagerly anticipated. But there are problems, more specifically soldering for a perfect finish—a mixture of good and bad advice and no photographs of good and dry joints.

Joystick controller—Why do I have to read two inches of text to discover that it is for the Spectrum?

Word processor, very good; but then Randle Hurley is good.

Latch Card Errors in text, e.g., where is 741s373 in parts list and 741so4 in cost. What is a KB914? Diagram is poor and the description of circuit operation is poor.

Christmas lights—diagrams poor; T1 T2 T3 and T4 have some odd connections. The use of mains equipment in the manner shown is to be discouraged; mains and low voltage equipment, not specifically insulated for mains operation such as the ZX-81, should not share the same PCB. Optical separation would be better.

Graphics generator,—again poor diagrams. No circuit of connections between RAM and ROM. Some ZX-81s have a 1K RAM, so what do I do? The logic circuit uses too many components, the requirement can be achieved with two ICs, a 74LS30 and a 74LS10 and also not give two positions in the memory map for the $\overline{\text{CHRS RAMCS}}$.

Eprom blower—diagram titles incorrect; logic circuit poor.

F. A. Richards
Arrington,
Hertfordshire.

● *Thank you for your letter which contains some valid points. Regarding some of them, dry joints do not always look dry and the sketch was intended to depict a good solder fillet. People who can solder will develop their own style but beginners do not need style—they need clear instructions which, if followed, will produce a sound joint.*

There is a limit to the amount of circuit explanation which can be included in an article; figure two and table one give a complete picture of the theory and practice for operating the Latch Card. The input and outputs of opto-isolators share the same PCB in the way the connections to relays do, so where is the advantage except in speed of operation?

The Graphics Generator figure nine is the program in the centre column. Those with a 1K RAM chip will have to wait for a future article. If you built your ZX-81 from a kit and have the circuit diagram the changes are obvious but are too lengthy to detail. Regarding the component count, almost any circuit can be improved either by head-scratching or using the latest super-whizz-bang chip; for the hobbyist the most important thing is that it can be built easily and if it does what the designer intended.

Most people would not wish to solder on the ZX-81 board and so add-ons for £1 or £2 are difficult to devise.

Thank you for your suggestions, which we will bear in mind for future issues.

Graphic

AT LAST, a completely different type of magazine and if you can keep up the range of projects in subsequent issues, you are sure to have a best-seller.

I built the ZX-81 character generator from the first issue and can describe it as unbelievable, especially as it costs less than £5, including

Continued on page 44

LETTERS

Continued from page 43

everything from solder to ICs.

I encountered several problems in construction which perhaps caused others to hesitate building the project. I have listed them:

A lack of a wiring/cutting diagram for the board. There are no track-cuts except between IC pins.

I found it necessary to cut the track on either side of pin 20 of the ROM. Also, the connection to $\overline{\text{ROMCS}}$ would not work when connected to pin 23B of the edge connector—I had to fasten it to the right-hand side of R28, the resistor above the heatsink.

After that modification the circuit worked. Honestly, the effect of this board just has to be seen to be believed. Keep up the good work.

David Oram,
Liversedge,
West Yorks.

● *Thank you for your praise of the project. I cannot agree with your comments except about the lack of a wiring/cutting diagram. You state that all track cuts are between IC pins but the two track cuts necessary are between the ICs and the rear edge connector. Cutting the track on the R28 side of ROM pin 20 will prevent the ULA controlling the ROM and the ZX-81 will never work.*

Are you sure you cut the track between pin 20 and R28 and not one of the adjoining tracks to pins 5 and 6?

The article says that the $\overline{\text{ROMCS}}$ *output from the* $\text{CHR\$}$ *circuit should go to ROM pin 20; connecting it to the edge connector 23B will not work, since one of the track breaks is between ROM pin 20 and edge connector 23B.*

Latch card

I WAS very interested on reading the Latch Card project. As I understand it the Latch Card effectively gives the ZX-81 output ports from anyone of the memory locations 36832 to 36863.

Being something of a novice I wonder if you would be able to offer advice, as I also wish to input signals to memory location? The application of such a device would be for sequence circuit testing by switching to send a signal/pulse to each circuit to be tested.

P W Beddoes,
Uxbridge, Middlesex.

● *To input signals to the computer you could use basically the same circuit using the* $\overline{RD}$ *line instead of the* $\overline{WR}$ *line at the edge connector, swapping the* Q_0–Q_7 *and* D_0–D_7 *lines to the 74LS373, connecting pin 11 directly to +5V instead of to the 1K resistor and connecting* $\overline{OE}$ *pin 1 to the output of the 74LS133 leaving the invertor, 1K resistor and diode connected as before.*

If you can afford to wait, a future project will be a more versatile input/output memory-mapped port with 24 lines of input/output (three bytes).

MAIL ORDER ADVERTISING

British Code of Advertising Practice

Advertisements in this publication are required to conform to the British Code of Advertising Practice. In respect of mail order advertisements where money is paid in advance, the code requires advertisers to fulfil orders within 28 days, unless a longer delivery period is stated. Where goods are returned undamaged within seven days, the purchaser's money must be refunded. Please retain proof of postage/despatch, as this may be needed.

Mail Order Protection Scheme

If you order goods from Mail Order advertisements in this magazine and pay by post in advance of delivery, *Sinclair User* will consider you for compensation if the Advertiser should become insolvent or bankrupt, provided:

(1) You have not received the goods or had your money returned; and

(2) You write to the Publisher of *Sinclair User* summarizing the situation not earlier than 28 days from the day you sent your order and not later than two months from that day.

Please do not wait until the last moment to inform us. When you do write, we will tell you how to make your claim and what evidence of payment is required.

We guarantee to meet claims from readers in accordance with the above procedure as soon as possible after the Advertiser has been declared bankrupt or insolvent (up to a limit of £4,250 per annum for any one Advertiser so affected and up to £10,000 per annum in respect of all insolvent Advertisers. Claims may be paid for higher amounts, or when the above procedure has not been complied with at the discretion of *Sinclair User*, but we do not guarantee to do so in view of the need to set some limit to this commitment and to learn quickly of readers' difficulties).

This guarantee covers only advance payment sent in direct response to an advertisement in this magazine (not, for example, payment made in response to catalogues etc, received as a result of answering such advertisements). Classified advertisements are excluded.

EDGE CONNECTOR

Edge Connector signal allocation

SPECTRUM

BOTTOM	TOP
A$_{14}$	A$_{15}$
A$_{12}$	A$_{13}$
5v	D$_7$
9v	
Slot	Slot
Ov	D$_0$
Ov	D$_1$
CK	D$_2$
A$_0$	D$_6$
A$_1$	D$_5$
A$_2$	D$_3$
A$_3$	D$_4$
$\overline{\text{IORQGE}}$	$\overline{\text{INT}}$
Ov	$\overline{\text{NMI}}$
Video	$\overline{\text{HALT}}$
Y	$\overline{\text{MEMREQ}}$
V	$\overline{\text{IOREQ}}$
U	$\overline{\text{RD}}$
$\overline{\text{BUSRQ}}$	$\overline{\text{WR}}$
$\overline{\text{RESET}}$	–5v
A$_7$	$\overline{\text{WAIT}}$
A$_6$	+12v
A$_5$	–12v
A$_4$	$\overline{\text{MI}}$
$\overline{\text{ROMCS}}$	$\overline{\text{RFSH}}$
$\overline{\text{BUSAK}}$	A$_8$
A$_9$	A$_{10}$
A$_{11}$	

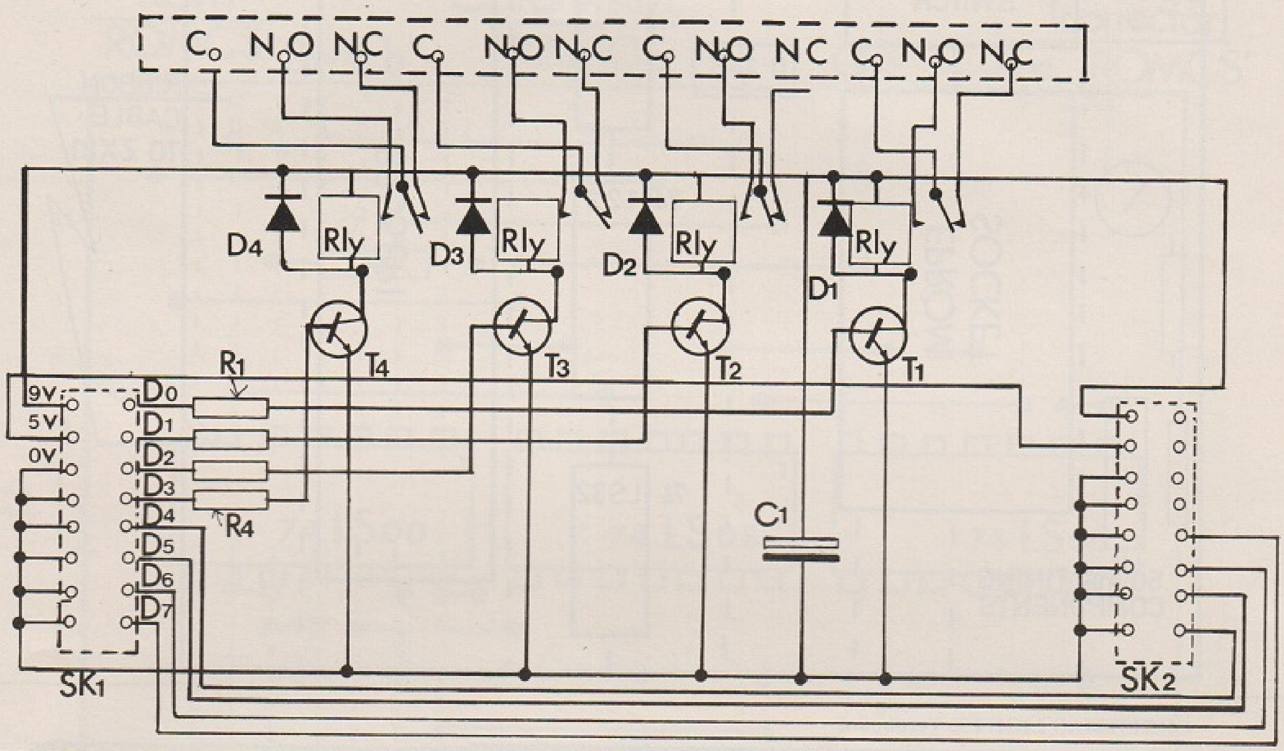
ZX-81

BOTTOM	TOP
5v	D$_7$
9v	$\overline{\text{RAM CS}}$
Slot	Slot
Ov	Do
Ov	D$_1$
$\overline{\varnothing}$	D$_2$
A$_0$	D$_6$
A$_1$	D$_5$
A$_2$	D$_3$
A$_3$	D$_4$
A$_{15}$	$\overline{\text{INT}}$
A$_{14}$	$\overline{\text{NMI}}$
A$_{13}$	$\overline{\text{HALT}}$
A$_{12}$	$\overline{\text{MREQ}}$
A$_{11}$	$\overline{\text{IORQ}}$
A$_{10}$	$\overline{\text{RD}}$
A$_9$	$\overline{\text{WR}}$
A$_8$	$\overline{\text{BUSAK}}$
A$_7$	$\overline{\text{WAIT}}$
A$_6$	$\overline{\text{BUSRQ}}$
A$_5$	$\overline{\text{RESET}}$
A$_4$	$\overline{\text{MI}}$
$\overline{\text{ROMCS}}$	$\overline{\text{RFSH}}$

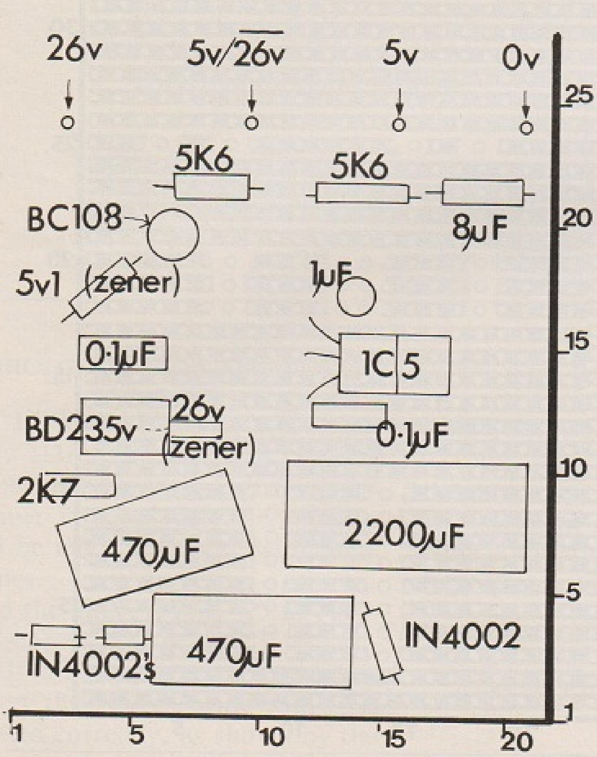
UPDATE

IN THE first issue of *Sinclair Projects* there were a few unfortunate errors in some of the diagrams. In this section we have reprinted the diagrams with the necessary corrections. We also inadvertently omitted the Veroboard layout in the graphics generator. This is now shown below.

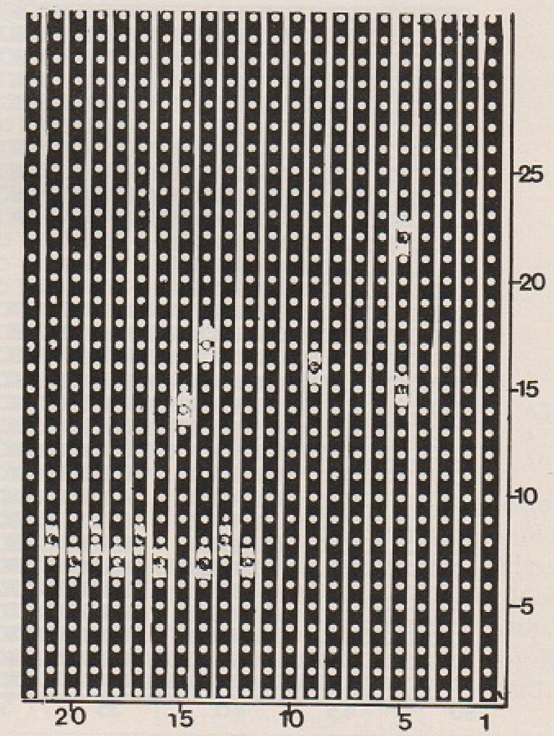
CHRISTMAS LIGHTS: Power-card schematic



EPROM BLOWER:
PCB layout for large board



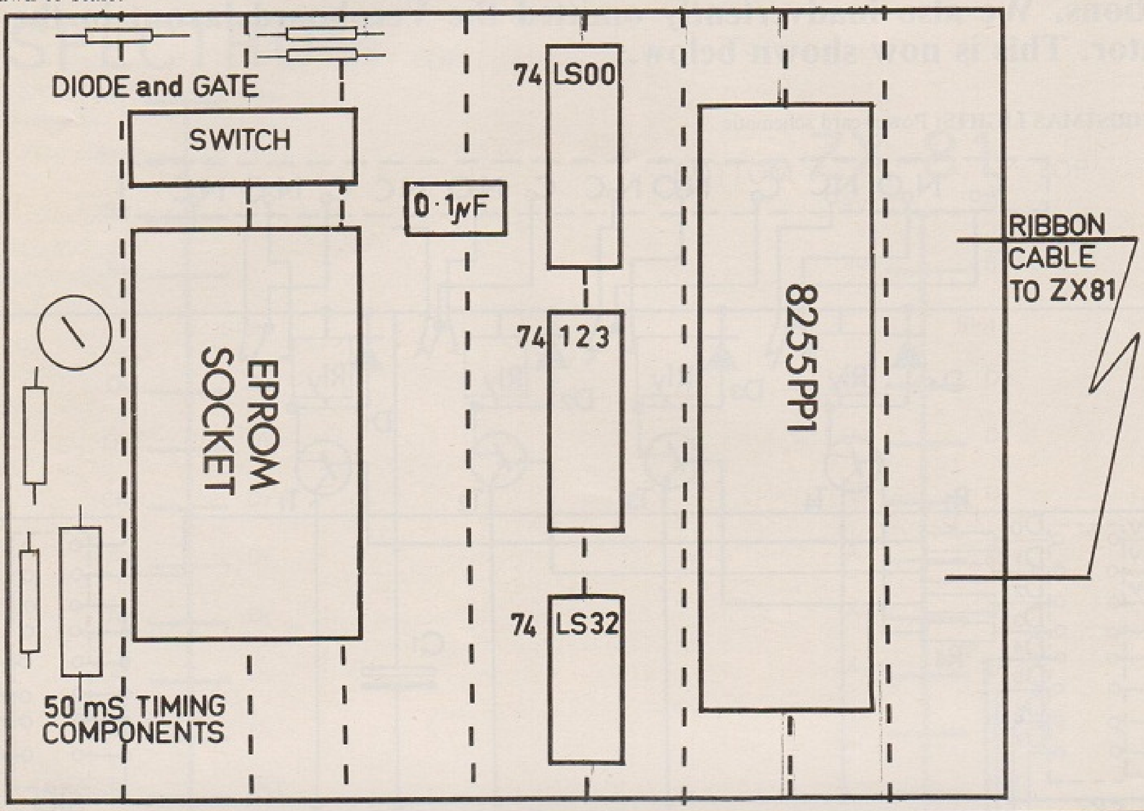
Veroboard



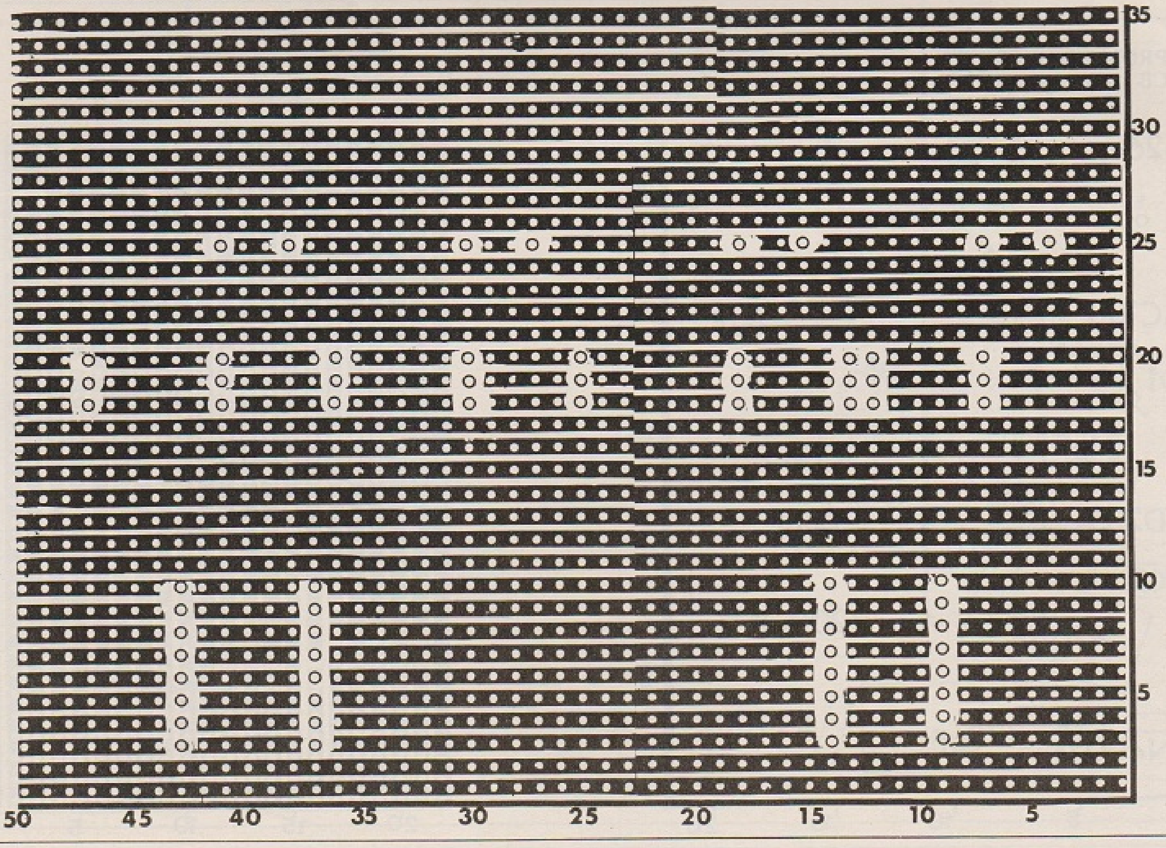
UPDATE

EPROM BLOWER

Component lay-out for the small board of the eprom blower. The dotted lines denote the lines of the cuts on the underside of the type of Veroboard which was used. The lines are equally spaced with four holes between them. They are not shown as such because the components are not drawn to scale.

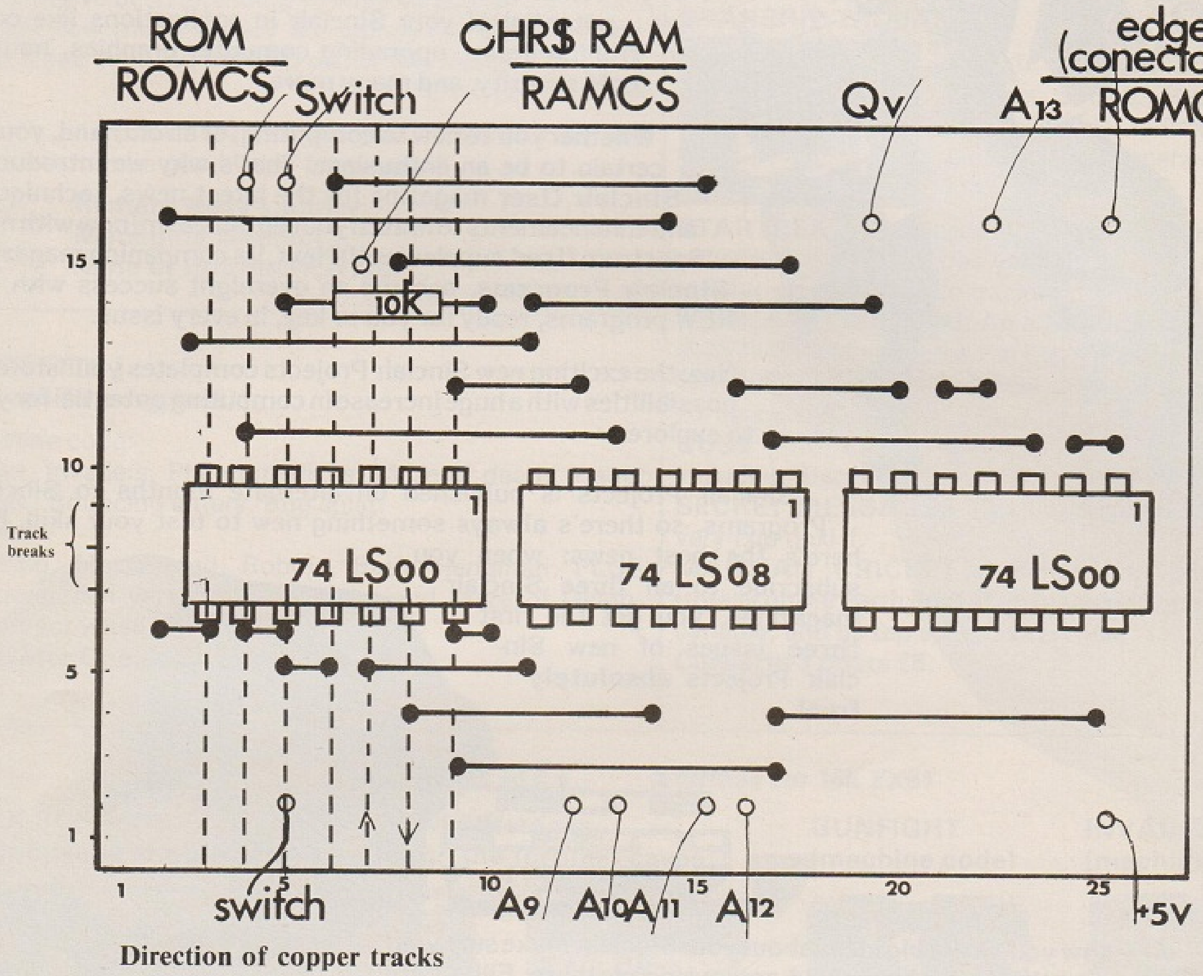


CHRISTMAS LIGHTS: Veroboard



UPDATE

GRAPHICS GENERATOR: PCB layout



GRAPHICS GENERATOR: Adaptation for ZX-81s with single 4118 RAM chip

VERY FEW ZX-81s were made with a single 4118 RAM chip. The following modifications to the article in the December issue of *Sinclair Projects* should be made for those with these machines.

Bend the address pins A0 to A8—pins 8, 1, 7, 22, 6, 23, 5, 4, 3—upwards until they are at right angles. Re-insert the IC, ensuring that it is orientated correctly, as shown by the identifying notch.

Using fine insulated single-core wire and a fine-tipped soldering iron, link the RAM IC to the address pins A0 to A8 of the ROM, as follows:

RAM IC pin	ROM IC pin	Address line
3	23	A8
4	1	A7
5	2	A6
23	3	A5
6	4	A4
22	5	A3
7	6	A2
1	7	A1
8	8	A0

Then continue as detailed previously.

Some 2114 RAM chips are soldered into the PCB. In this case, unless you are very good at de-soldering ICs, it is best to cut out the ICs with wire cutters, clear the holes of any leads, solder in some IC holders and then insert some new 2114s.

MICHAEL ORWIN'S ZX81 CASSETTES

THE BEST SOFTWARE (BY VARIOUS AUTHORS) AT LOW PRICES

QUOTES

"Michael Orwin's £5 Cassette Two is very good value. It contains 10 stolid well designed games which work, offer plenty of variety and choice, and are fun."

From the ZX Software review in
Your Computer, May '82 issue.

"I had your Invaders/React cassette . . . I was delighted with this first cassette."

P. Rubython, London NW10

"I have been intending to write to you for some days to say how much I enjoy the games on 'Cassette One' which you supplied me with earlier this month."

E.H., London SW4

". . . I previously bought your Cassette One and consider it to be good value for money!"

Richard Ross-Langley
Managing Director
Mine of Information Ltd.

CASSETTE 1
(eleven 1k programs)

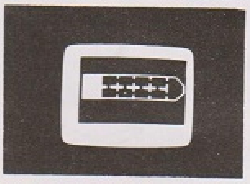
machine code:
React, Invaders, Phantom aliens, Maze of death, Planet lander, Bouncing letters, Bug splat.
Basic:
I Ching, Mastermind, Robots, Basic Hangman. PLUS Large screen versions of Invaders and Maze of Death, Ready for when you get 16k.
Cassette One costs £3.80

CASSETTE 2
Ten games in Basic for 16k ZX81

Cassette Two contains Reversi, Awari, Laser Bases, Word Mastermind, Rectangles, Crash, Roulette, Pontoon, Penny Shoot and Gun Command.
Cassette Two costs £5.

CASSETTE 3
8 programs for 16k ZX81

STARSHIP TROJAN

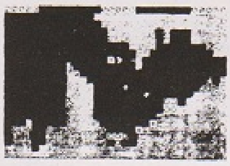


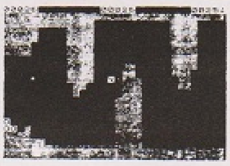
Repair your Starship before disaster strikes. Hazards include asphyxiation, radiation, escaped biological specimens and plunging into a Supernova.

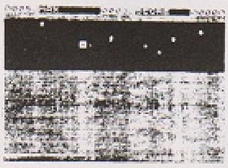
STARTREK This version of the well known space adventure game features variable Klingon mobillity, and graphic photon torpedo tracking.
PRINCESS OF KRAAL An adventure game.
BATTLE Strategy game for 1 to 4 players.
KALABRIASZ World's silliest card game, full of pointless complicated rules.
CUBE Rubik Cube simulator, with lots of functions including 'Backstep'.
SECRET MESSAGES This message coding program is very txlp qexi jf.
MARTIAN CRICKET A simple but addictive game (totally unlike Earth cricket) in machine code. The speed is variable, and its top speed is very fast.
Cassette 3 costs £5.

CASSETTE 4 **8 games for 16k ZX81**

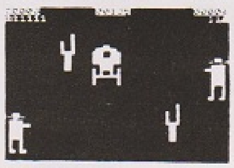
ZX-SCRAMBLE (machine code) with 3 stages.
Bomb and shoot your way through the fortified caves.



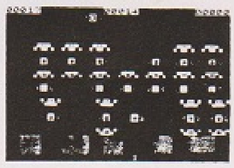


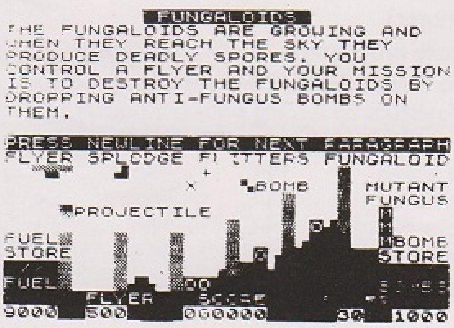


GUNFIGHT
(machine code)



INVADERS
(machine code)





GALAXY INVADERS (machine code)
Fleets of swooping and diving alien craft to fight off.
SNAKEBITE (machine code)
Eat the snake before it eats you. Variable speed. (very fast at top speed).
LIFE (machine code)
A ZX81 version of the well known game.
3D TIC-TAC-TOE (Basic)
Played on a 4 × 4 × 4 board, this is a game for the brain. it is very hard to beat the computer at it.

7 of the 8 games are in machine code, because this is much faster than Basic. (Some of these games were previously available from J. Steadman). Cassette 4 costs £5.

Recorded on quality cassettes, sent by first class post, from:
Michael Orwin, 26 Brownlow Rd., Willesden, London NW10 9QL (mail order only please)

COMPETITION-PRO JOYSTICK

• PRECISION GAME CONTROL!

For **ZX SPECTRUM** Complete with stylish interface unit simply plug in for instant arcade action.

ONLY £25.00 inc. VAT

Outstanding Features

- Super strong nylon and steel construction, built to last.
- A total of 8 directions plus 2 large fire buttons.
- Arcade proven moulded leaf switches ensure incredible reliability.
- Complete with self contained boxed interface which simply plugs into the Spectrum expansion port (with or without printer).
- Easily programmed in BASIC or M/C

These arcade games, designed exclusively for the **Joystick** by leading software houses, are now available for Spectrum:

Time Game	M/C	48K	£6.95
Meteroids	M/C	16K	£4.95
Spooky Man	M/C	16K	£4.95
Cosmos	M/C	16K	£4.95

Watch this space for new software!

I wish to order:

Quantity	Item	Price
	Post & Packing	£1.00
	Total (VAT inclusive)	

Cheques/Postal orders should be made payable to Kempston (Micro) Electronics

Name ________________________

Address ________________________

SU2

Post to Kempston (Micro) Electronics, Dept
180a Bedford Road, Kempston, Bedford MK42 8BL

KEMPSTON
MICRO ELECTRONICS
180a Bedford Road, Kempston, Bedford MK42 8BL