JUNE/JULY 1984
An independent magazine published by ECC Publications
95p

SINCLAIR PROJECTS

THE COMPLETE HARDWARE COMPANION

CONTROL YOUR OWN RAILWAY

25 percent off the Kempston Interface E

JOYSTICK REVISITED
NEW: ADVICE COLUMN
SPECTRUM DIGITISER
INSIDE THE MOTOROLA 68008

NEW NEW NEW NEW NEW NEW NEW NEW

The most imaginative graphics programming software for your Spectrum

PAINTBOX is a must for every 48K SPECTRUM owner! If you want to exploit the full graphics capability of your machine you can do so . . . simply and easily with PAINTBOX.

Take a look at the screen prints shown above. They are the sort of thing you could produce on your SPECTRUM. PAINTBOX gives you an entire suite of graphics programming aids in one integrated program. For instance:

UDG EDITOR
The facility to define and re-define up to 84 graphics characters for your programs or for use in screen-planning.

UDG DRAWING BOARD
A fully-integrated UDG planner for defining up to 4 Banks of characters. Planning facilities include MIRROR, INVERSE, ROTATE, FILE.

SKETCHPAD
An experimentation "window" for developing the UDG set.

PRECISION PLOTTER
An amazingly versatile high resolution drawing board which includes PAPER choice INK choice PLOT, DRAW, CIRCLE, FILL, ARC, OVER, ERASE, and STORE!

SCREEN PLANNER
Combining PRECISION PLOTTER and UDG characters! A multi-purpose graphics facility to enable you to produce incredible screen graphics. All work can be sent to a Printer and SAVED as SCREEN or as CODE with its own built-in machine code routine for instant recall from BASIC.

The program comes with a DEMO program and a 28 page book that's packed with hints and tips on how to get the best from PAINTBOX.

PAINTBOX is ideally suited for use with the Print 'n' Plotter Spectrum Jotter Package — the first and best graphics planning pad for the Spectrum! So why not place an order today? Write, call, or see your local dealer.

01-660 7231
24 HOUR CREDIT CARD ORDERING

Post to: Dept S Print 'n' Plotter Products Ltd., 19 Borough High Street, London SE1 9SE.
Please send me:

.... "PAINTBOX" SOFTWARE @ £7.50 (+75p p+p total £8.25)
.... SPECTRUM JOTTER PADS @ £7.50 (£1.50 p+p total £9)
.... SPECTRUM KEYBOARD OVERLAYS @ £2.60 (35p p+p total £2.95)
.... 5 ROLLS ZX PRINTER PAPER @ £11.55 (95p p+p total £12.50)

☐ I enclose remittance in full
☐ Please bill my Access/Barclaycard/Visa/Mastercard No:

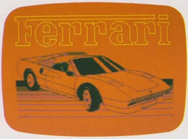
Overseas orders please add 25% for additional surface mail rate.

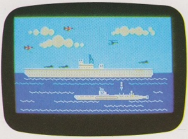
NAME ____________

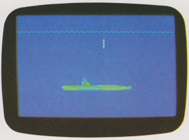
ADDRESS ____________

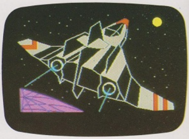
DEALERS: Phone 01-403 6644 for enquiries

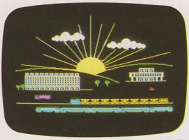
Print 'n' Plotter Products











SINCLAIR PROJECTS

ALL MANUFACTURERS of hardware like to keep people interested in their products by informing them what is to be launched. That long-awaited, low-cost mass storage device might just be around the corner; that high-powered new micro could be only weeks away from starting production.

Everyone has heard the claims, both outrageous and modest, and believed or rejected them depending on the amount of trust placed in the speaker. Given the record of Sinclair Research, there can have been few people who expected to receive QLs as soon as they were planned to go on sale at the end of February, but no-one expected to wait as long as they have before receiving a promise of seeing the results of their £400 investment.

It must be clear to everyone now that at the time of the launch in January, Sinclair did not have a QL. It had a box of assorted components and an idea of the kind of software it would like to go with it—but in between there was a yawning gap.

All those hundreds of people who paid £400 and placed their orders have taken a quantum leap in the dark. They would have been better-advised to keep their money in the bank earning interest which is now being earned by Sinclair Research. They have been promised a gift to sweeten their bitterness but Sinclair is able to pay for that with ease from the special account in which the money is being held.

The company is trying to enter the professional market. It will have to show less naivety and more professionalism before it can hope to gain a toe-hold in an area where there is already a large number of companies with products to sell and the knowledge of how to go about it.

Meanwhile, if there is anyone who is still interested in what the QL is capable of doing, we will supply as much information about it as we can. In this issue John Mellor has been investigating the Motorola 68008 chip.

Following the new format we introduced in the last issue, we have another two hardware projects. For the Spectrum there is a digitiser which allows the direct input of graphics and for the ZX-81 and Spectrum, Brian Lee has built a model railway controller.

Continuing the development of the magazine we have started a problem-answering service. We have asked Trevor Marchant, a man with a wealth of knowledge of both the ZX-81 and the Spectrum, to write a regular column based on your queries. He is a busy man and cannot undertake to answer problems personally or on the telephone. He intends, however, to deal with areas which are of importance to all of you.

Managing editor Nigel Clark **Consultant editor** David Buckley **Managing production editor** Harold Mayes MBE **News writer** John Lambert **Design** Elaine Bishop **Advertisement manager** Holly Fleming **Advertisement Executive** Robert Marcus **Editorial assistant** Colette McDermott **Production assistant** Dezi Epaminondou **Assistant managing director** Barry Hazel **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 3525. 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, 196–200 Balls Pond Road, London N1 4AQ. 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.

Disc system is twice as fast as Microdrive

OF THE MANY new disc systems which have appeared recently for the Spectrum, one of the first is the Viscount system from Interactive Instruments.

It comprises an interface and a 5¼in. Shugart drive with a hard-wired connecting cable. As there is no through connector on the interface, users who wish to add a printer interface will also need some kind of extender connector which Interactive can also supply. The system will, at present, support only the one drive and uses a single-sided, single-density format. It is configured to give 107K of storage per disc.

The interface is housed in a very solid metal box; care should be taken when inserting the power lead, which plugs in the back, as it can short-out. Inside is a professionally-laid-out PCB. It is surprising that one of the standard disc operating chips is not used but instead a multitude of TTL chips. That is the reason for the non-standard storage capacity.

On power-up the contents of an 8K EPROM is loaded into the top 8K of memory; a number of variables taking another 110 bytes are loaded into the variables area and you are presented, on screen, with the message FIZ 83/540.1 OPERATING FIRMWARE, © 1983 Macronics Systems Limited.

The variable nd, used to format a new disc, is set to 64003 and used in the form RAND USR nd. For most of the commands f$ is first filled with the file name, plus any parameters. Those parameters are used to signify the auto-run line with Basic programs and the start, length and run addresses, in decimal, of CODE.

With arrays any number of DIMentioned arrays can be saved provided the total length per save is fewer than 2,816 bytes. Large arrays therefore must be split into parts and saved separately. The usual commands to obtain a directory, delete a file and duplicate a disc are also available.

The system is roughly twice as fast as a Microdrive, even allowing time to type the command, but it uses a great deal of RAM space. Business users could not use it for many of the usual business programs for that reason, although some specially-written software is becoming available.

Overall, the system works well and is very useful, especially if you write your own programs. The system costs £245 inc. and is available from the Spectrum chain of shops or direct from the manufacturer, Interactive Instruments Ltd, Unit 6, Pilot House, King Street, Leicester.

Interface DOS now uses only 1K

THE DISC interface from Technology Research, reviewed in the last issue of *Sinclair Projects,* has been upgraded. The main changes are that you can now SAVE the variables by preceding the save name with #, that when auto-running the program does a GOTO 1 rather than RUN, and that the DOS uses only the upper 1K of memory. Programs saved using an old interface are compatible with the new one. Users with the old interface can buy an upgrade.

The price of the new interface is the same as the old one, £85 plus VAT, and it can be obtained from Technology Research Ltd, 356 Westmount Road, London SE9 1NW. Tel: 01-856 8408.

Pen has positional error

AMONG Add-On Electronics Hardware is a light pen for the Spectrum. Included in the software provided with the pen are 16 routines to allow you to draw pictures.

The light pen has two parts. The pen contains a BPW 148 photo-transister with one leg cut off. The interface box, into which the pen is plugged, takes its power from the Spectrum power lead and has another lead to the Spectrum power socket. Yet another lead plugs into the EAR socket after the software has loaded.

The pen works by timing the flying spot of light on the TV from the start of the scan to where it is detected. The interface amplifies the signal and uses the EAR socket as a port.

With a light pen the timing is all-important, as a small time difference leads to a large positional error. This pen has a random positional error of up to 2in. which makes it almost impossible to use.

It was used on Sony Trinitron and Fidelity TVs but may work correctly with other sets.

At £30 the pen costs £10 more than others on the market. It can be obtained from Add-Ons Electronics, Units 2, 3 & 4, Shire Hill Industrial Estate, Saffron Walden, Essex CB11 3AQ.

Getting 16K on tap

FROM Camel Products, well-known for its EPROM products for the ZX-81, is the new ROM SP, an EPROM-to-Spectrum loader. The unit can hold either one or two 2764s or one 27128, giving nearly 16K on tap.

The unit is housed in a custom-designed ABS case and has a flexible connector to the back of the Spectrum; there is also an extender card on the back of the unit. On the top is a LED to show when a program is being transferred and a pushbutton switch. Inside the unit are two chips, a few discrete components and two sockets for the EPROMs. Three Minicon pins are used to hard-link the unit to transfer on either power-up or by pressing the pushbutton. Another three are used to select the type of EPROM.

Users with an Issue 1 Spectrum may experience difficulties if the unit is set to transfer on power-up. That is because that issue is slow to set up initially.

No difficulties were found if the unit is set to work via the push-button or on any other issue Spectrums.

The unit provides a welcome addition to the Camel range of products and is designed to work alongside its PROMER-SP, a Spectrum EPROM blower, which has a program to optimise space on the EPROM which can be used for Basic CODE and DATA.

The ROM SP costs £29.95, as does the PROMER-SP, both plus VAT. EPROM erasers are also available from £18.95 plus VAT, all inclusive of p&p. Camel Products is at 1 Milton Road, Cambridge CB4 1UY. Tel: 0223-314814.

Light rifle is accurate up to 6ft.

A NEW IDEA on an old theme is the Stack Light Rifle from Stack Computer Services. It is a four-part sniper's rifle supplied with three games tapes, **High Noon, Grouse Shoot** and **Shooting Gallery** for the 48K Spectrum. Other versions of the Rifle are available for the Commodore 64 and Vic-20.

The main pistol is attached, via 12ft. of cable, to a dead-ended ZX-81-size edge connector which plugs into the rear user port of the Spectrum. To the pistol can be fitted a barrel, stock and telescopic sight.

The rifle works like some light pens in that it detects the flying spot of light on the TV screen. A photodiode in the pistol barrel picks up the spot and the software, when the trigger is pulled, finds the X/Y position. It does so accurately from up to about 6ft. away.

Overall the rifle is well-made and the games highlight the kind of uses to which it can be put. At the time of writing other games are being written to use the rifle. The Stack Light Rifle costs £29.95 inc. VAT with the three games and can be obtained either direct from the manufacturer, Stack Computer Services Ltd, 290-298 Derby Road, Merseyside L20 8LN or local retailers.

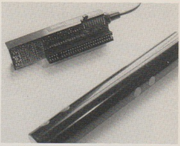
Tricky simulations of joystick actions

THE TRICKSTICK from East London Robotics could best be described as a joystick simulator. It is a 7in.-long black plastic cylinder with six touch-sensitive pads which simulate the usual four joystick positions plus two fire buttons.

It is supplied with its own interface and a software trainer tape. The interface mimics the Kempston joystick standard and can therefore be used with many existing programs. In addition, it is possible to use up to eight Tricksticks at once and have proportional movement, selected by pins on top of the interface.

The touch pads are drawing pins which use the capacitance of your body to effect a voltage level. That is amplified by circuitry in the stick and is gated with a Spectrum supplied reference voltage. The relevant bit of the data byte is therefore set, or not, accordingly. Once set, the bit will revert to 0, the rate of change of the bit will depend o the capacitance, i.e., is proportional to the proximity of your finger to the drawing pin. The overall sensitivity can be changed by a small variable resistor at the top of the stick.

When using more than one Trickstick, the pins on top of the interface decide at which address the stick will sit; A5 and A6 are held low and A8 to A15 denote the address, all the other address lines being held high.



With no moving parts, the Trickstick should last a long time and, as it is complete with interface, is reasonably-priced at only £34.50. It is available, by mail order, from East London Robotics Ltd, Gate 11, Royal Albert Docks, London E11.

Professional finishing

KELAN is producing a prototyping kit which can be used with the ZX-81, Spectrum or Jupiter Ace. The kit, part No HB/2090, will give a professional finish to any project and can be used for many of the designs in *Sinclair Projects*.

In each kit is a prototype PCB with space for 12 16-pin ICs, 10 in comfort, and a Veroboard-style scratchpad. There is also a Spectrum-style 28-way edge connector which can be cut down for use on other machines, an extender card and a 9-pin D-type Atari-style PCB mounting socket. A case, similar to that used by Cambridge Computing for its joystick interface is provided.

Costing only £9.50 inc., the kit is very good value. For further information, Kelan (Hobbyboard), North Works, Hookstone Park, Harrogate, North Yorkshire HG2 7BU. Tel: 0423 883672.

UPDATE

FEBRUARY/MARCH **Letters,** page 12, line seven of M Farnsworth's letter should read '7620 has "p)" " should be'.

Digital electronics page 14, col. 2, line 11— '$\overline{A}$ is pronounced 'A bar' or as'; col. 3, line 29—' remember that $\overline{0}$ is 1:'; page 15 figure five—the & should not be in the NOT gate; col. 2, line 4 '$\overline{\overline{A}} = A$'; col. 3, line 29— 'the function $\overline{D}.\overline{E}$. If the theorem is'; page 16, col. 1, line 29—'$\overline{A.B} = C$'; col. 1, line 30—'$\overline{\overline{A.B}} = A.B$'; page 17, col. 3, line 26—'$(\overline{A}.\overline{B}) = (\overline{A+B})$; col. 3, line 29—'$(\overline{A.B}) = (\overline{A}+\overline{B})$; col. 3, line 36—'$\therefore C = (A+B+(A+B)$'; page 18, col. 2, line 5 '$\therefore G = (1.0)+1$'.

Automatic re-start, page 22, figure three—The NOR gate is joined to 5v and IC1b; capacitor C3 should be labelled 470μF; page 24, figure four—at the top end of the capacitor C1 there should be '+'.

Sound Board; page 38, figure two—the capacitor at the top left of the figure should be labelled C2 and 10μ, the variable resistor to the right of that is VR1; the capacitor further to the right is C1, beneath that C9 should be labelled 330p, R4 is 470R and R5 is 470R. Page 39, figure two—the resistor in the top left corner marked 1K is R1: the AY 38910 is IC1 and to the left of that is socket SK1; page 40, program 5, line 180—'POKE 16516, B.

RAM re-set; page 42, figure seven—column headings are 'A13, $\overline{Q2}$, A13, $\overline{A13}$'; page 43, figure eight—hatching in column A13 should be in the top half with the bottom half being clear; page 45, figure three—the arrow at the top near line 25 should be labelled RESIN, on IC9 the fourth pin from the top on the right-hand side is connected to 5v and the connections near 4K7 are shown incorrectly. The connection on line 18 should be with capacitor C4 and there should be a connection between lines 19 and 26; page 46, figure six—to the right of resistor R1 should read 'A15 to pin 6 of '138'; page 46, figure four—labelled warm-reset circuit diagram, on IC9a the top is connected to +5v and the bottom of resistor R3 is connected to 0v.

NEWS

Deliveries delayed

SINCLAIR RESEARCH is hoping its problems with the delivery of the QL will be over soon. At the time of going to press the company was promising that all delivery dates which have been given to customers would be met.

It refused, however, to indicate the dates for the first deliveries. Journalists were able to test production models at the company offices in London from April 17.

Sinclair Projects was told by a reader who placed an order at the end of February that he has been given a delivery date of the end of July. Sinclair said in January that production would begin at the end of February and soon afterwards orders should take no more than 28 days to satisfy.

A spokeswoman said that a gift was being considered for people who had paid by cheque and whose money was in a holding account.

Pigment printout

COLOUR PRINTING can now be done from the Spectrum. Euroelectronics of Cheltenham has produced Copy Four software which allows users of the company's ZX Lprint Centronics and RS232 interface to print in four colours on the MCP 40 or Tandy CGP 115 printers.

Costing £5.50, the cassette is in two versions, machine code for the Mk111 model and Basic for earlier models or other interfaces, such as those built by Kempston and Tasman.

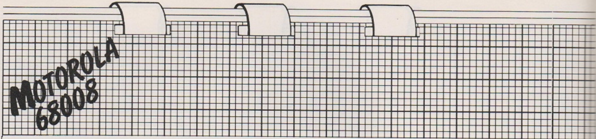
Peripheral of the Year

THE MICRONET system is receiving increasing acclaim. Following its success in the RITA awards at the WHICH COMPUTER? Show, the Prism VTX 5000 modem was named as Peripheral of the Year in the British Microcomputing Awards.

It was selected because of its ability to put low-cost telecommunications before micro users. It won narrowly from the Epsom RX80 printer and the Torch disc pack.

The Spectrum and BBC B shared the What Micro? Home Microcomputer Award, with most of the other hardware titles going to the ACT Apricot.

The VTX 5000 allows Spectrum owners to access viewdata services, including Micromet 800 and Prestel.



Getting to the heart of the Quantum Leap

The 68008 chip from Motorola is one of the main reasons for the low cost of the QL. John Mellor discovers its capabilities and how it is organised with a 16-bit microprocessor and an eight-bit data bus while retaining the 32-bit architecture

AT THE HEART of the Sinclair QL computer are two processors. One is the 68008 microprocessor chip and the other the 8048 single-chip micro-controller.

The 68008 is a member of the 68000 microprocessor family. The microprocessors are designed and manufactured by Motorola and second-sourced by Mostek Corp, Hitachi Ltd, Philips, Rockwell Int, Signetics and Thomson EFCIS. Second-sourced means that those manufacturers also manufacture them by agreement with the chip designer Motorola.

It is important for Sinclair or any microcomputer manufacturer that it can obtain components from more than one source to minimise the chances of industrial action or the policy decisions of the suppliers affecting production.

The 68008 is a 16-bit microprocessor with an 8-bit data bus. The 8-bit data means that it is cheaper and

Figure 1. 68008 register.

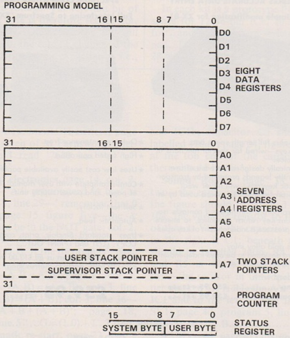


Table 1.

Mnemonic	Description
ADBC	Add Decimal with Extend
ADD	Add
AND	Logical And
ASL	Arithmetic Shift Left
ASR	Arithmetic Shift Right
BCC	Branch Conditionally
BCHG	Bit Test and Change
BCLR	Bit Test and Clear
BRA	Branch Always
BSET	Bit Test and Set
BSR	Branch to Subroutine
BTST	Bit Test
CHIC	Check Register Against Bounds
CLR	Clear Operand
CMP	Compare
DBCC	Test Condition, Decrement and and
DIVS	Signed Divide
DIVU	Unsigned Divide

INSTRUCTION SET

Mnemonic	Description
EOR	Exclusive Or
EXG	Exchange Registers
EXT	Sign Extend
JMP	Jump
JSR	Jump to Subroutine
LEA	Load Effective Address
LINK	Link Stack
LSL	Logical Shift Left
LSR	Logical Shift Right
MOVE	Move
MOVEM	Move Multiple Registers
MOVEP	Move Peripheral Data
MULS	Signed Multiply
MULU	Unsigned Multiply
NBCD	Negate Decimal with Extend
NEG	Negate
NOP	No Operation
NOT	One's Complement
OR	Logical Or

Mnemonic	Description
PEA	Push Effective Address
RESET	Reset External devices
ROL	Rotate Left without Extend
ROR	Rotate Right without Extend
ROXL	Rotate Left with Extend
ROXR	Rotate Right with Extend
RTE	Return from Exception
RTR	Return and Restore
RTS	Return from Subroutine
SBCD	Subtract Decimal with Extend
SCC	Set Conditional
STOP	Stop
SUB	Subtract
SWAP	Swap Data Register Halves
TAS	Test and Set Operand
TRAP	Trap
TRAPV	Trap on Overflow
TST	Test
UNLK	Unlink

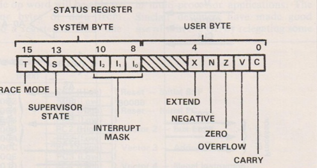
easier to use to design a system and that standard 8-bit peripheral chips can be used readily. That is one reason for the low price of the QL, which is advertised as a 32-bit computer. Internally the 68008 retains the the full 68000 32-bit microprocessor architecture. The 68008 uses the same instruction set and codes as the 68000, so that programs written for one processor will run on the other.

The address bus has also been reduced in size and two of the less important control signals are not brought to the outside of the package. By doing that the 64-pin 68000 has been squeezed in a 48-pin dual-in-line package.

Non-segmented space

There are 20 address lines, so the QL can address a non-segmented 1MB address space directly. That provides plenty of scope for memory expansion and should be more than adequate for most applications. One megabyte, for those who are a little rusty on the system international standard prefixes, is 1,000K bytes or 1,048,576 bytes. There are 20 address lines giving $2^{20} = 1,048,576$ addresses. If we are to move to 16-bit computing with 16-bit precision we require two bytes to store each 16-bit word. In fact, on the big brother of the 68008, the A_0 line is not used and so the number of available addresses is halved.

Figure 2. Status register.



Even so, the storage capacity in bytes is a useful guide when describing and comparing systems. You will need 10 Microdrive cartridges to store all the information contained in 1MB of RAM. According to the QL specification, 640K of that memory space, less whatever the system uses, will be readily available to the user if he decides to upgrade a QL with the 0.5MB RAM pack.

The 68008 uses memory-mapped input/output, unlike the Z-80, which uses a separate I/O map and I/O instructions. So long as the peripheral devices have high-speed data latches, memory and peripherals can be decoded in the same way and accessed using the same instructions. That makes programming more straightforward but means that some of the memory space will be occupied by peripheral addresses.

The 68008 has non-multiplexed address and data buses. That simplifies the external hardware required to build the system, because we do not require demultiplexors and latches to create valid data or addresses. The data in memory can be treated in many ways. It can be considered to be a bit, a byte of eight bits, a word of 16 bits or a long word of 32 bits.

Data can also be considered to be a 4-bit binary coded decimal character or it can represent an address, in which case it will be 32 bits long. The way the memory is used is carefully ordered so that data cannot be placed at random. That simplifies program-

Table 2.

Instruction Type	Variation	Description
ADD	ADD ADDA ADDQ ADDI ADDX	Add Add Address Add Quick Add Immediate Add with Extend
AND	AND ANDI ANDI to CCR ANDI to SR	Logical And And Immediate And Immediate to Condition Code And Immediate to Status Register
CMP	CMP CMPA CMPM CMPI	Compare Compare Address Compare Memory Compare Immediate
EOR	EOR EORI EORI to CCR EORI to SR	Exclusive Or Exclusive Or Immediate Exclusive Or Immediate to Condition Code Exclusive Or Immediate to Status Register

Instruction Type	Variation	Description
MOVE	MOVE MOVEA MOVEQ MOVE from SR MOVE to SR MOVE to CCR MOVE USP	Move Move Address Move Quick Move from Status Register Move to Status Register Move to Condition Codes Move User Stack Pointer
NEG	NEG NEGX	Negate Negate with Extend
OR	OR ORI ORI to CCR ORI to SR	Logical Or Or Immediate
SUB	SUB SUBA SUBI SUBQ SUBX	Subtract Subtract Address Subtract Immediate Subtract Quick Subtract with Extend

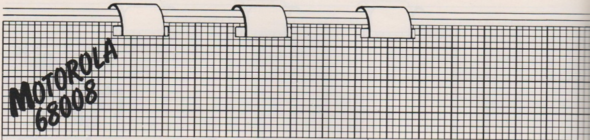
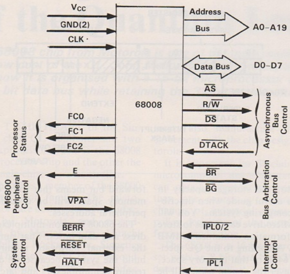


Figure 3. 68008 Signal lines



ming and system operation. Internally, the 68008 information is stored in 17 general-purpose registers. Each of them is 32 bits long and they can all be used as an accumulator, i.e., like the A reg on a Z-80 system. Arithmetic and logical operation can be performed on any register, with the result being stored in that register and the flags being set.

True 32-bit

That fairly large number of registers means that true 32-bit, high-speed complex processing can be carried-out internally to the 68008 with the minimum transference of data to or from memory. Figure one is a programming model of the layout of the registers. In addition to the general-purpose registers grouped as eight data registers, seven address registers and two stack pointers, there is a 32-bit program counter and a 16-bit status register.

Full-signed 32-bit arithmetic can be performed on the address registers but only the lower 20 bits will appear on the address bus. The status register is divided into two bytes. The user byte contains the usual flags, plus an extended flag which is used to indicate a borrow or a carry in multiple-precision arithmetic. The extend flag is similar to the carry flag in most other microprocessors.

The system byte contains the interrupt mask. Those three bits are used to define the current interrupt priority. The 68008 has only two interrupt control inputs as opposed to the three inputs available on the 68000 — figure three. The inputs IPLO and IPL2 have been combined so that the three interrupt levels supported are 2, 5 and 7. The interrupt pins on the 68008 indicate the encoded priority level of the interrupting device; level seven is the highest priority and is non-maskable.

The other levels are maskable; any interrupt higher than the current mask level will be recognised. Interrupts, re-set, traps and software restarts or traces are dealt with in a similar way. Collectively they are called exceptions and the way the processor deals with them is called exception processing. The interrupts or exceptions are vectored — once the processor decodes the type of exception and its priority it will jump to a particular service routine, the starting address of which is stored in a memory location. That address is called a vector because it points to the start of the exception service routine. There are 255 vectors occupying 1K of RAM from 000000h to 0003FFh — figure four.

Increased complexity

One of the biggest problems with the larger, more advanced processors is the increased complexity of the hardware and the instruction set. That is often compounded by different modes of operation. The 68008 was designed with the machine code programmer in mind, so that the number of different instructions is small — only 56, see table one.

That does not mean that the 68008 is limited, because each of those instructions can have up to 14 addressing modes, listed in table three, resulting in more than 1,000 useful instructions when you consider the different data types on which they can operate. Table two lists some

Table 3.

ADDRESSING MODES

Register Direct Data Register Direct Address Register Direct	**Register Indirect** Register Indirect Postincrement Register Indirect Predecrement Register Indirect Register Indirect with Offset Indexed Register Indirect with Offset
Absolute Data Absolute Short Absolute Long	**Immediate Data** Immediate Quick Immediate
Program Counter Relative with Offset Relative with Index and Offset	**Implied** Implied Register

additional instructions which are variations of those in table one. Most of the instructions will operate on bytes, words — 16 bits — or long words — 32 bits — and some of them will also operate on bits or BCD digits. Bits 7 and 6 of the op code are often used to specify the data type, unless it is implied in the instruction.

Instructions are from one to five words long — two to 10 bytes. The first word of the instruction is the operation word which specifies the type of instruction to be performed. The remaining words are either the operands, which may be two words long, or the source and destination addresses, which each may be two words long.

Start at even address

The way the data is held in memory follows the standard pattern for an 8/16-bit microprocessor. A 16-bit system will address one word at a time and transfer all 16 bits along the data bus to the CPU. The 68000 system can transfer a byte at a time along the upper or lower eight bits of the data bus, depending on whether it is an even or odd byte. Thus instead of having an A_0 pin the 68000 has an upper data byte stroke — UDS — and a lower data byte stroke — LDS. The 68008 uses the A_0 line which is available on the chip rather than UDS and LDS. A_0 is used to select the odd and even bytes, selecting the odd bytes when it is high and the even bytes when it is low.

Words and long words must always start at an even address. Figure six shows how that data may be organised. A single op word will be able to transfer four bytes of data from memory into a register or from one four-byte location in memory to another. An important aspect of the 68008 is its ability to work efficiently in multi-processor applications. The Sinclair designers have made good use of it in the QL by relegating some

Figure 4.
Memory Addresses

(Hex)			
000000	SSP (High)	Reset — Initial SSP	
000002	SSP (Low)		
000004	PC0 (High)	Reset — Initial PC	
000006	PC0 (Low)		
000008	PC2 (High)	Vector 2 — Bus Error	
00000A	PC2 (Low)		
00000C	PC3 (High)	Vector 3 — Address Error	
00000E	PC3 (Low)		
000010	PC4 (High)	Vector 4 — Illegal Instruction	
000012	PC4 (Low)		
000014	PC5 (High)	Vector 5 — Divide by 0	
000016	PC5 (Low)		
000018	PC6 (High)	Vector 6 — CHK Instruction	
00001A	PC6 (Low)		
00001C	PC7 (High)	Vector 7 — TRAPV Instruction	
00001E	PC7 (Low)		
000020	PC8 (High)	Vector 8 — Privilege Violation	
000022	PC8 (Low)		
000024	PC9 (High)	Vector 9 — Trace	
000026	PC9 (Low)		
000028	PC10 (High)	Vector 10_{10} — Opcode 1010 Emulation	
00002A	PC10 (Low)		
00002C	PC11 (High)	Vector 11_{10} — Opcode 1111 Emulation	
00002E	PC11 (Low)		
000030	PC12 (High)	Vector 12_{10}	Reserved by Motorola
000032	PC12 (Low)		
00005C	PC23 (High)	Vector 23_{10}	
00005E	PC23 (Low)		
000060	PC24 (High)	Vector 24_{10} — Spurious Interrupt	
000062	PC24 (Low)		
000064	PC25 (High)	Vector 25_{10} — Level 1 Interrupt	Auto-Vector Interrupts
000066	PC25 (Low)		
000068	PC26 (High)	Vector 26_{10} — Level 2 Interrupt	
00006A	PC26 (Low)		
00006C	PC27 (High)	Vector 27_{10} — Level 3 Interrupt	
00006E	PC27 (Low)		
000070	PC28 (High)	Vector 28_{10} — Level 4 Interrupt	
000072	PC28 (Low)		
000074	PC29 (High)	Vector 29_{10} — Level 5 Interrupt	
000076	PC29 (Low)		
000078	PC30 (High)	Vector 30_{10} — Level 6 Interrupt	
00007A	PC30 (Low)		
00007C	PC31 (High)	Vector 31_{10} — Level 7 Interrupt	
00007E	PC31 (Low)		
000080	PC32 (High)	Vector 32_{10}	TRAP Instruction Vectors
000082	PC32 (Low)		
0000BC	PC47 (High)	Vector 47_{10}	
0000BE	PC47 (Low)		
0000C0	PC48 (High)	Vector 48_{10}	Reserved by Motorola
0000C2	PC48 (Low)		
0000FC	PC63 (High)	Vector 63	
0000FE	PC63 (Low)		
000100	PC64 (High)	Vector 64	User Interrupt Vectors
000102	PC64 (Low)		
0003FC	PC255 (High)	Vector 255	
0003FE	PC255 (Low)		

Figure 5. Instruction format.

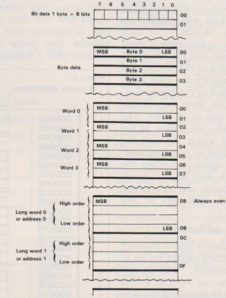
Op word		
Op word	Operand — one or two words	
Op word	Source — one or two words	Destination — one or two words
Op word	Operand — one or two words	Destination — one or two words

MOTOROLA 68008

of the system control functions to the pre-programmed 8084 micro-controller. A second processor can request use of the microcomputer buses and signal lines using a bus request signal on the BR pin. The 68008 will put all its addresses, data and control output lines into the high-impedance state, so that the second processor can transfer data to and from memory and peripherals along the computer buses until the second processor relinquishes control of them. To increase co-processor speed the 68008 has a sophisticated bus arbitration scheme. Internal logic on the 68008 can process bus request signals while data transfers are taking place.

The 68008 has control signals which can be gated externally, to provide the control signals for the synchronisation of data transfer to 6800 peripheral devices. Later we will offer a project to add parallel I/O to the QL using the 6821 PI0 chip.

Figure 6. Data organisation in memory.



ADVICE

Beware of the pitfalls in improving your Sinclair

To begin our new regular feature Trevor Marchant warns of the difficulties involved in adding extra hardware. If you have any queries write to him at Sinclair Projects

THE SPECTRUM, like most other Sinclair computers before it, is always attracting small add-ons and, like most other users experimenting with them is, I find, part of the fun of owning a computer. While this is a popular pastime, I wonder how many people realise the dangers involved? In advertisements you read in magazines, very rarely are you warned of the pitfalls which adding a keyboard, expansion RAM pack, sound unit, joystick or expansion port can produce.

Lose guarantee

If you open a Spectrum case it invalidates any guarantee. The chips inside are static sensitive, although the probability of blowing any is minimal, unless you handle the board inside. Using the power rails on the edge connector can blow up the internal power components. That is one of the most common faults, unless a few simple rules are followed:

Always try to stick to the outside edge connector; try to supply power via the Spectrum power supply unit—it supplies 9V approximately at 1.4 Amps, although the 48K Spectrum uses 600–700mA; if it is necessary to use the power from the edge connector, keep to the +5V line and avoid the −5V line.

One reader, Anthony Kirke, is interested in using his Spectrum to control his workshop equipment. As a first note I offer a warning—there are two things of which to be careful—do not try to connect the mains to the Spectrum in any way and do not try to go beyond the limits of your skill, since you have to learn to walk first.

It is possible to control Kirke's equipment by using a Spectrum without opening it. For I/O ports you need to contact Harley Systems Ltd, Box 7, The Pepperboxes, Great Missenden, Bucks. The company supplies several types of interface for connection to almost anything.

Safest method

There are several ways of doing what you ask; one of the safest is using a relayed controlled interface. That would allow you to switch on and off, in and out, up and down, using the relays to handle the 5 Amps at 250V. Programming the interface would give the desired results. I understand from his letter that he is not conversant with electronics, so please be careful.

The options which I/O ports and interfaces offer users is immense. Using a computer to control a house is a growing hobby and many dedicated systems are sold in the U.S. so it will not be long before they appear on our market.

It is not necessary to spend thousands of pounds, since armed with a little knowledge, some tools and a computer it is possible to do almost anything. Controlling things in the house could mean having a computer regulating the central heating system, turning it up or down as the inside temperature dictates, and it can even take into account the outside temperature. That would allow you to increase the inside temperature slowly as it becomes colder outside and so possibly lower your heating costs.

It is possible to install a very intelligent burglar alarm system, monitoring windows, doors, pressure pads, even infra-red detectors. The lights, television, radio or video could be pre-set to go on and off to give the impression that someone is at home. It is possible to develop the idea further. I could use a computer to turn on an electric blanket.

Modern-day slave

Simple problems, too, can be overcome by using a modern-day slave to control, monitor, switch various items such as printers, joysticks, tape players. Imagine having a system

ADVICE

which will record only when music is playing, or a video which records only the film, not the advertisements, or a power drill which drills to pre-set programs. All that and more is possible; the only limiting factors are the time taken to set it up—and your own imagination.

Kirke's idea is typical of today's computer users; we all want to do something more than play games and with the add-on market booming it is possible.

Problems answered

If you have a particular problem with your Spectrum or ZX-81, Microdrives, Interface 1 and 2, write to me at *Sinclair Projects* and I will try to help. I work for a company which makes the computers—not Sinclair or *Sinclair Projects*—and know most of the problems associated with the computers. I will also deal with particular add-ons and how you can use them to their full potential.

Current projects include a programmable joystick interface, variable power supply 0-30V/0-3 Amp, mother board for expansion, tape controller unit, and others at the ideas stage. My main difficulty is finding time to put all that into operation at home.

Useful add-on

One useful cheap add-on for the Spectrum/ZX-81 is a re-set switch. If you have an edge connector block, fit a piece of Veroboard into it with solder and fit a push-on/release-off switch between OV and re-set. When pushed it will erase any program from memory, regardless where RAMTOP may be set. That obviates the need to turn off your computer every time, which can cause the PSU 9V plug to become intermittent.

Now for answers to a number of queries:

While using my computer it suddenly stopped working and now all I get are black and white bars, with random rubbish in with the main bars on my TV.

In simple terms you have no –5V to your 16K RAM. That could be caused by plugging add-ons to your Spectrum without a keyway in the connector. To cure it depends on your skill in electronics, and whether or not your computer is still under guarantee. It is necessary to replace TR4 (ZTX650), cut the –5V track to IC6-13, check the voltage on the supply end. If that is satisfactory check each chip one by one. Cut pin one on all chips and re-connect one at a time,

measuring –5V line which should be re-connected. Replace any ICs which lower the –5V line—note it may be more than one.

Dead keyboard

Part of my keyboard has stopped working. The QWERTY keys do not work at all, although the rest seems to be satisfactory.

There are two possibilities. Either the tab inside is not inserted correctly or it is broken. Inside the computer there are two ribbon leads, a five-way and an eight-way. The five-way controls down the lines, e.g., 1, Q, A, cap shift, and 0, P, enter, space break, for one of the lines. The eight-way controls across the lines, e.g., Q, W, E, R, T for one line, so that shows which is your problem.

Open the computer and replace the tab carefully and firmly in its black box, ensuring it lines-up correctly. If it is broken, send the computer to Sinclair.

How many volts/Amps does the 48K Spectrum require to run it, as I wish to build a power supply and run my home-built add-ons.

An average 48K requires between 600–700mA to run, rising to about 800mA depending on what it is doing at that time. The only thing of which to be wary is the 9V supply, which must be well smoothed, although it need not be too well regulated.

A Spectrum will run on anything between 7.5V and 13V DC, although stay near to 9V if you can, as that leaves less for the internal regulator to drop. A perfect PSU would be capable of supplying up to 3 Amps (0–3) and up to 12V (0–12); that would give scope for additions. Use a bridge rated at 3 Amps and 4400 going to +vF capacitors (2×2200) to give the required smoothing.

I need a proper keyboard, as I want to use my 48K as a word processor. What do you suggest?

I have looked at most of the available keyboards and eventually bought a Fuller FDS. My reason was that it has keys which feel like a real keyboard, unlike some others.

CPU fault

When I have finished programming my new computer with the keyboard, I find that I have to re-set RAMTOP *which is still where I placed it originally. Am I correct in thinking that is incorrect?*

Your problem is IC2. The CPU unit is at fault—it is not re-setting on new command from the keyboard. It is a fairly common CPU fault, although CPU faults are rare. The only answer is to change it.

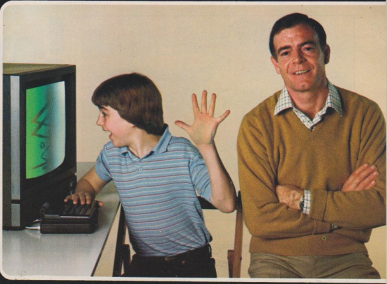
If, however, the Spectrum is still under guarantee, return it to Sinclair, stating the fault. Be careful removing the IC as the board is through-hole plated.

Schmitt trigger

I own a ZX-81 which seems to dislike loading programs. It has never been easy to load but sometimes it is better than others. I would be grateful for advice.

Unfortunately the ZX-81 always appears to have some problem in loading, even if you own an excellent quality cassette unit. I recall in the early days of the Spectrum a unit called the 'Q' save, which I believe was made by Panda. The unit helps a great deal with loading and saving. The same effect can be achieved by using a Schmitt trigger circuit, which squares the signal in or out of the ZX-81

Today, we talked to our user group, booked our holiday, zapped nine monsters, checked the football results, bought two games, looked at share prices, learnt some French, and conquered the universe!



Prism VTX 5000
winner
– Peripheral of the year –
British Microcomputing
Awards 1984

Prism gi

information, commu

Games

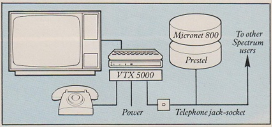
Swapping

Educational software

News

Competitions

Telesoftware



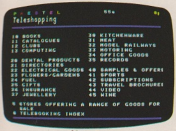
...and so m

Access a world of information with your Spectrum.

Simply by connecting it to the telephone line, via the award winning Prism VTX 5000 communications modem, you can link into a wide variety of services, including Prestel and the popular Micronet 800. Access hundreds of software programs – many completely free. Talk to fellow users. Keep up to date with all the latest national, international and micro news. Play on-screen games. Check sports results. Book travel and

ves you
nication, education...

BRITISH MICROCOMPUTING AWARD
Peripheral
VTX 5000
of the Year.
1984

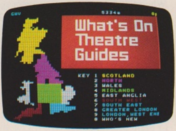


Shopping

Travel

Teach yourself programming

Sport



Booking

Mailbox

uch more!

entertainment. Learn a language. Check share prices.

The choice is simply vast.

Plus, your Prism VTX 5000, allows you to communicate directly with other Spectrum owners.

And now, our special offer means that you can do all this for just £11* a month. So complete the coupon and send it to us today, and we will let you have full details of our unbeatable offer.

FOR JUST £11* A MONTH YOU WILL GET:

- **A Prism VTX 5000 worth £99.95.**
- **One year's subscription to Prestel worth £20.00.**
- **One year's subscription to Micronet worth £32.00.**
- **Free user-to-user software – normally retailing at £19.00.**
- **Quarterly hardware and software offers.**
- **Discounted software through Micronet (up to 20% off retail prices).**

Prestel and the Prestel symbol are trademarks of British Telecommunications.

SPJ6

Please send me full information on your special offer to Spectrum users.

Name: ____________________

Address: ____________________

Return to: **Prism Microproducts Ltd.**
Prism House, 18-29 Mora Street, City Road
London EC1V 8BT. Tel: 01-253 2277

PRISM

*All prices quoted refer to domestic use only. For business use the Prestel annual standing charge is £66. Therefore business users will be billed £44.50 per quarter.

DIGITISER

Making an automatic point on the Spectrum

Drawing designs on a micro can be made much easier by using a digitising tablet. Corin Howitt shows how to build one.

THIS IS the first of a series detailing how to build and use a digitiser for the Spectrum. A block diagram of the complete system is shown in figure one. It is divided into three circuit blocks. The first is the digitising tablet — figure two. That consists of a board made from wood or other material. Pivoted across the board are two moving arms. By moving the writing tip at the end of the second arm, the voltage output from the two potentiometers located at the two pivot points varies in sympathy with the degree of rotation of each arm, allowing us to work out mathematically the position of the writing tip on a pre-defined x,y grid.

The second circuit block is a two-channel analogue-to-digital converter which converts the two analogue signals from the digitising tablet — block one — into binary values for use by the Spectrum. The third circuit block is the interface circuitry which allows the Spectrum to control and use the ADCs — block two.

Let us start by considering the interface circuitry — block three. Each ADC in block two requires two control signals from the computer. The first is a start conversion pulse — $\overline{SC}$. When that line goes low it causes the ADC to start converting the analogue voltage at its input to a binary value for the computer. The second is an output enable pulse — $\overline{OE}$. That tells the ADC to place the binary value stored in its output buffer on to the Z-80 data bus. IC1 is a 13-input NAND gate. The output of a NAND gate is only at logic level 0 when all of its inputs are at logic level 1. Looking at the circuit — figure three — you will see address lines A_0-A_4, A_9-A_{11} are all connected direct to inputs of IC1. Those address lines therefore must be at logic 1. Address lines A_5-A_7 are unconnected, which makes them also at logic level 1. Address lines A_{12}-A_{15} can be taken to IC1 via

PARTS LIST

Semiconductors
IC1:74LS133
IC2:74LS04
IC3:74LS32
IC4:ZN449
D1,2:1N914

Resistors
R1:4K7 ¼W ± 2%
R2:47K ¼W ± 2%
R3:390R ¼W ± 2%
R4:5K6 ¼W ± 2%
R5:8K2 ¼W ± 2%
R6:680K ¼W ± 2%
Pre-sets are ± 20% tolerance

Capacitors
C1:100pF ceramic
C2,4-5:100nF ceramic
C3:4u7 tantalum

+5V input circuit
RV1:5K pre-set (Lin)
RV2:1MO pre-set (Lin)

Substitute if using alternative input circuit.

Miscellaneous
28 + 28way 0.1in. Spectrum edge connector
3.75 × 5in. Veroboard
2 × 16-pin DIL sockets
2 × 14-pin DIL sockets
1 × 18-pin DIL socket
3.5mm. PCB-mounting jack socket
Single-core connecting wire
Solder.

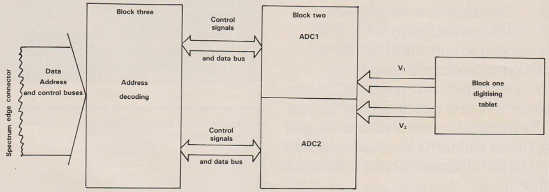
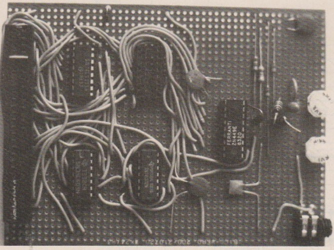


Figure 1. System block diagram.

DIGITISER



inverters — IC2. When you are constructing the circuit it is for you to decide whether those address lines pass through converters before they connect to IC1. If an address line passes through an inverter, it must be at logic level 0 to provide the input logic level 1. If the address line does not pass through an inverter, it must be at logic level 1. That method allows you to specify which address the board occupies in the Z-80 64K I/O space.

$\overline{IORQ}$ is passed to IC1 via an inverter to provide the correct logic level. So far we have the output of IC1 which goes low whenever a specified I/O address appears on the address bus. From that signal and the Z-80 $\overline{RD}$ and $\overline{WR}$ signals we need three ADC control signals. $\overline{SC}$ starts both $\overline{ADC1}$ and ADC2 conversion cycles. ADC1 enables the output buffer of ADC1, clocking its data on to the data bus. $\overline{ADC2}$ enables the ADC2 output buffer, clocking its data on to the data bus. The $\overline{SC}$ pulse is generated by ORing together $\overline{WR}$ and the correct address output of IC1. Thus when those two inputs go low, both ADCs start their conversion cycles.

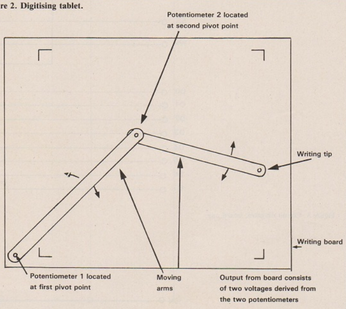
$\overline{ADC1}$ and $\overline{ADC2}$ are generated by ORing together $\overline{RD}$ and the correct address output from IC1. That output is then passed to a single input of two more OR gates. Address line A_8 is passed to the second input of one of the two OR gates and is also passed via an inverter to the second input of the second OR gate.

To enable ADC1, the correct address output from IC1 and $\overline{RD}$ must be low, A_8 must be high. To enable ADC2, the correct address output from IC1, $\overline{RD}$ and A_8 must all be low. As ADC2 will be constructed on a separate board, the control signals $\overline{ADC2}$ and $\overline{SC}$ are connected to a 16-pin DIL socket along with the data bus lines A_0-D_7 and the two power lines +5V and 0V. The advised pin-out of that socket is given with the IC pin-outs in figure four.

We will now look at the circuitry of ADC1. It is built around IC4, a ZN449 ADC with an accuracy of $\pm$1LSB. A simplified internal system diagram is shown in figure five. When $\overline{SC}$ first goes low an 8-bit binary counter is re-set in the system control logic block. The outputs of that counter control eight analogue switches which switch resistors in and out of a chain in the resistor chain and decoding block.

At each clock stage, the analogue switches switch resistors in and out of the chain producing a voltage, Vr, which varies from 0V to 5V in 256 individual steps. At each stage a comparator compares Vr with Vin, the analogue input voltage. The largest resistor is switched in and out first to determine whether Vin is greater or

Figure 2. Digitising tablet.



DIGITISER

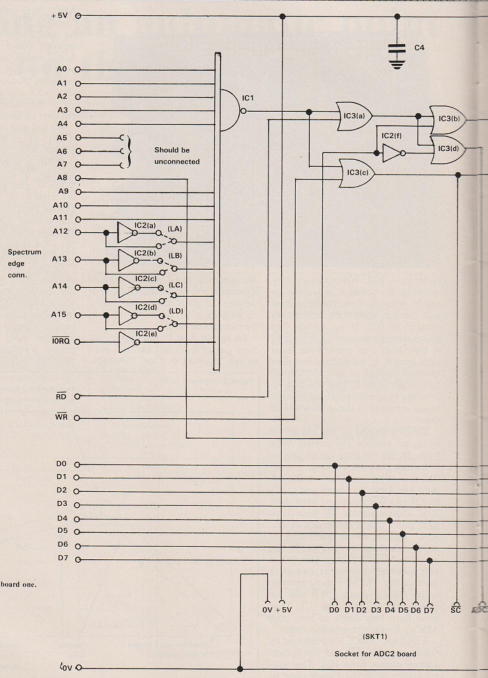
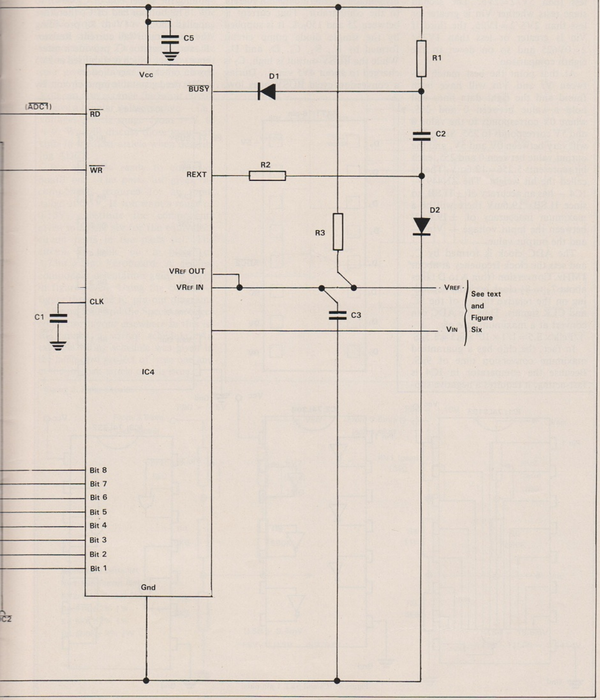


Figure 3. Circuit diagram, board one.



C5
Vcc
D1
BUSY
R1
(ADC1)
RD
C2
WR
R2
REXT
D2
R3
VREF OUT
VREF IN
· VREF ·
See text
and
Figure
Six
CLK
C1
C3
VIN
IC4
Bit 8
Bit 7
Bit 6
Bit 5
Bit 4
Bit 3
Bit 2
Bit 1
Gnd
IC2

less than 5V/2 = 2V5. The second stage tests whether Vin is greater or less than 2V5/2 = 1V25, the third if Vin is greater or less than 1V25/2 = 0V625 and so on down to the eighth comparison.

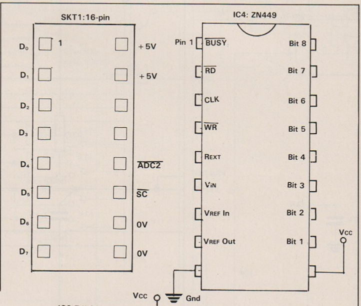
At that point the best match between Vr and Vin will have been found and the eight data lines will hold a value between 0 and 255, where 0V corresponds to the value 0 and 5V corresponds to 255. Since Vin will vary between 0V and 5V, and the output value between 0 and 255, each bit represents 5.256 = 19.6mV. That is called the bit weight. The ZN449 — IC4 — has an accuracy of ±1LSB, so since 1LSB = 19.6mV there will be a maximum inaccuracy of ±19.6mV between the input voltage — Vin — and the output value.

The ADC clock is formed by C_1 and sets the clock frequency at about 1MHz. Conversion from A to D takes about $7\frac{1}{2}$ to $8\frac{1}{2}$ clock periods, depending on the relative timing of the $\overline{SC}$ and CLK signals. Thus the ADC can convert at a maximum rate of:

$1/Fclk \times 8.5 = 1/1 \times 10^6 \times 8.5 = 8.5\mu S$

In fact, the chip has a guaranteed maximum conversion time of 9µS. Because the comparator in IC4 is fast-acting, it requires a negative supply voltage to generate the tail current to the comparator. That current is between 25 and 150µA. It is supplied by the simple diode pump circuit formed by R_1, R_2, C_2, D_1 and D_2. While the $\overline{BUSY}$ output is high, C_2 is charged to about 4V5 via R_1. During a conversion cycle $\overline{BUSY}$ goes low, pulling the upper end of C_2 down to 0V. The bottom end of C_2 therefore applies about −4V to R_2 providing the comparator tail current. Resistor R_3 and capacitor C_3 provide a reference voltage which is stabilised at 2V5 by an on-chip zener diode.

We need a suitable input circuit. By



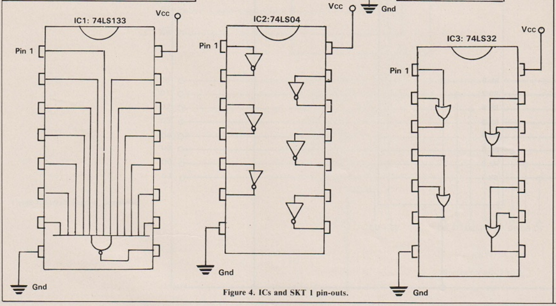


Figure 4. ICs and SKT 1 pin-outs.

choosing different input circuits we can alter the voltage range of the ADC. Note that the final digitising system will use ADCs with a +5V unipolar range. Two input circuits are provided in figure six. Input circuit one gives a range of 0-5V. Input circuit two gives a range of 0-10V. You can also arrange the input circuit to give a bipolar voltage range — i.e., voltages which range from −V to +V. We will discuss those input circuits in the next article when describing ADC2.

We are now ready to construct board one. The parts list gives the components required for an input range of 0-5V. If you want a range of 0-10V, substitute the components given in figure six for the equivalent circuit parts in the parts list. The circuit is built on a piece of 3.75in. × 5in. Veroboard. A general component orientation guide is given in figure seven. Using the circuit — figure three, the IC pin-out diagrams — figure four and the Spectrum edge-connector layout elsewhere in this issue, make a wiring schedule. An example wiring schedule was given in the latch-card project of issue one and in the joystick article of this issue.

Figure 5. Simplified internal block diagram of ADC.

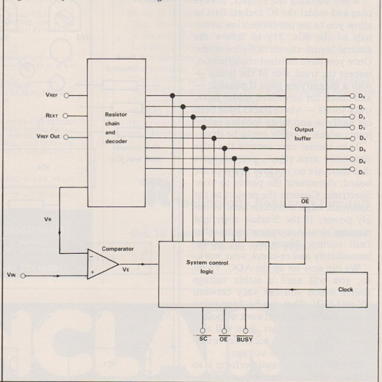
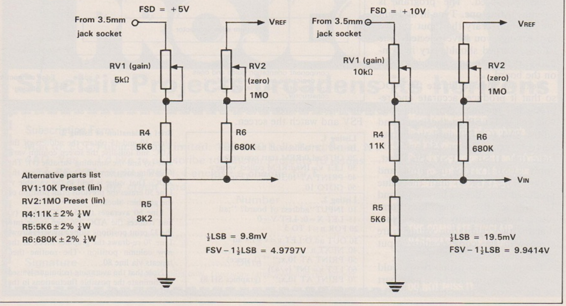


Figure 6. Input circuits.



When building the project, always plug and solder the IC sockets first to allow you to see the connection terminals of the ICs. Try to follow the general layout shown in figure seven. Once you have finished construction, inspect the track side of the board — with a magnifying glass if possible — and check for unwanted solder links, track joints — where tracks have been cut badly — and misty-looking solder joints which possibly could be dry.

Take special care around the edge connector area. Once you are sure that there are no obvious faults in the board, disconnect the power to your Spectrum. Connect the project to the Spectrum edge connector and re-apply power. If the Sinclair copyright message does not appear or another fault occurs, disconnect the power immediately and re-check your work.

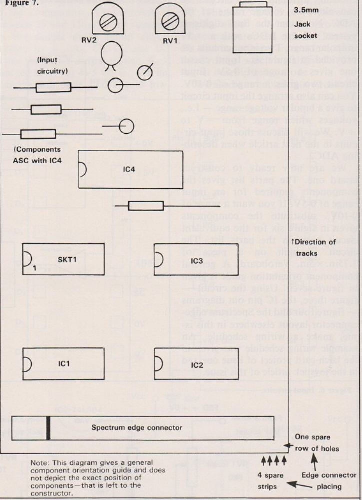
We can now set up the ADC. To do so, you will need a stable voltage source which you can vary between 0V and FSV. That can be done using a dry battery and a network of potentiometers and resistors. You will also need a reasonably accurate multimeter to measure the voltage.

The first test we must perform is to apply continuous convert — $\overline{SC}$ — pulses to the ADC and monitor the values produced. The program is given as listing one. Type-in and run the program. Vary the input voltage in the range you have selected. The values returned should vary in sympathy. If they do not, there is a fault on the board.

Next, we must calibrate the ADC so that it produces accurate and reproducible readings. Apply full-scale voltage — $1\frac{1}{2}$LSB to the input and adjust RV1 (GAIN) until the output value flickers between 254 and 255. Note that 1LSB = FSV/256. Next, apply $\frac{1}{2}$LSB (V) to the input and adjust RV2 (ZERO) until the value produced just flickers between 0 and 1. With all those operations complete, ADC1 is ready for use. Note that the two calibrating voltages are given with the corresponding input circuits in figure six.

The program in listing two should keep you amused until the next part of the system is described. Just vary the input voltage between 0V and FSV and watch the screen.

Figure 7.



Listing 1.

```
10 INPUT"address of board? ";ad
20 OUT ad,0:REM start conversion
30 LET v=INad:REM collect value
40 PRINT AT 10,10;v;"  " (2 spaces)
50 GOTO 10
```

Listing 2.

```
10 INPUT"address of board? ";ad
15 LET X=0: LET V=0
20 FOR n=1 TO 5
30 OUT ad,O:LET v=v+IN ad
40 NEXT n
50 PRINT AT 10,x;" " (1 space)
60 LET x=INT (v/42)
70 PRINT AT 10,x;" " (graphics SH 8)
80 GOTO 20
```

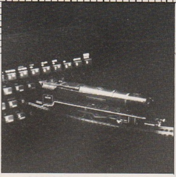
Brief explanation of listing 2.

Lines 10 and 15 obtain the address of the ADC and initialise the screen column variable (x) and the summing variable (v). The following loop sums five readings from the ADC. That value is held in the variable v. Line 50 erases the previous print position of the graphics block.

Line 60 averages the 5 ADC values (v) and reduces the ADC 256 step range into the 32 print positions along any screen row. Line 70 re-draws the graphics block at the new column position. The routine then repeats via line 80.

Note that the averaging technique is used to eliminate the possible fluctuations in the analogue input to the ADC.

LOCOTROL

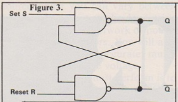


Develop your own micro-controlled model railway system

Brian Lee describes how it is possible to use a ZX-81 or Spectrum to run model trains. Here he considers the problems of speed and direction and the ways of adjusting them

THE MODEL TRAIN control system to be described in this series is designed to be operated via any digital I/O port, so it is applicable to both the Spectrum and ZX-81. The prototype was controlled via the I/O ports of the Programmable Sound Generator featured in earlier issues of the magazine and this device offers the addition of being able to create realistic sound effects, such as locomotive noise, in proportion to speed, steam noise, whistles and the like.

Figure 3.



S	R	Q	Q̄
O	I	I	O
I	O	O	I
I	I	NO CHANGE (REST STATE)	
O	O	NOT ALLOWED	

Initially, the system will require an 8-bit output port but more additions to the project will require an 8-bit input port as well. The article describes the construction of the speed and direction control, which also incorporates buffered lines to drive to be added later.

Each function of the system is switchable to manual control, so that the railway may be operated in a conventional manner when required. It is assumed that most readers who intend to build the project will already have a model train layout, so a mains power supply is not included. Details of a suitable supply will, however, be discussed later.

Anyone who has operated electric model railways will know that there are two main difficulties in achieving realistic speed control. The first is starting without too much of a jerk and the second is maintaining a very slow speed necessary for such operations as moving the train into a siding.

The track voltage necessary to start the train is usually much higher than that required for slow running; so to start a train manually, it is necessary to turn the speed controller well up, causing the unrealistic jerk forwards, and then to turn it down rapidly to

COMPONENTS

ICs
IC1 4011BE
IC2 μA7805
IC3 NE555
IC4 74LS93
IC5 74LS85
IC6 74LS08

Transistors
TR1, TR2, TR5 and TR6
TR3
TR4
BC107B (4 off)
BFY50
2N3055

Diodes
D1, D5 and D6 IN4001 (3 off)
D2
3mm. red L.E.D.
D3, D4, and D7 IN4148 (3 off)

Resistors
VR1 1K linear pot.
R1, R2, R8, R9, R11 10K (5 off)
R3, R4, R10 100K (3 off)
R5 1K
R6 270K
R7 OR27 3W. wire-wound
¼W.

Capacitors
C1 2200μF 25V Elect. Axial leads
C2 4μ7 25V Elect. Radial leads
C4, C5, C6 0μ1 min. disc ceramic (3 off)
C3 10n polycarbonate
C7 0μ68 polyester

Switches
S1 S.P.S.T. ultra min. toggle. Maplin FH97F.
S2 D.P.D.T. ultra min. toggle. Maplin FH99H.
S3 Push to make. Maplin FH59P.

Relay
RL1 Ultra min. relay 12v. D.P.D.T. Maplin YX95D

Case
Maplin desk console style 2. (ABS console M6005)

Control knob
Maplin type F11 (HB26D)

Veroboard
1 piece 36 strips × 32 holes.

IC sockets
8-pin 1 off
14-pin 3 off
16-pin 1 off

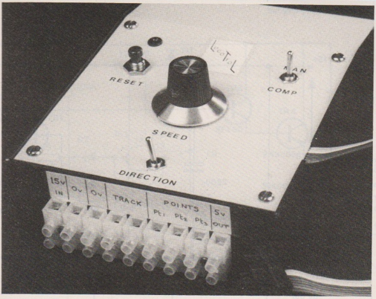
DIL header plug
16-pin. 1 off

Ribbon cable
1 piece, 9-way 12 in. long

Connector block
9-way, 1 off

3mm. LED panel mounting clip, 1 off.
Single-ended Veropins.
Mounting kits for TR4 and IC2, 1 of each.

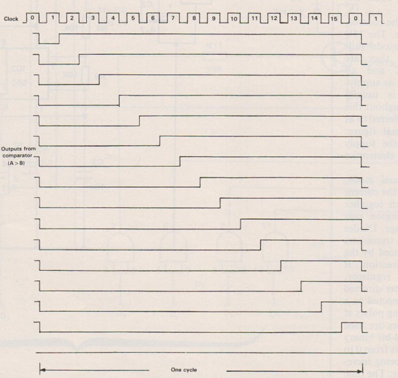
LOCOTROL



Program 1.

```
10 OUT 191,7: OUT 223,192 Sets both ports to output.
15 OUT 191,14: OUT 223,0: STOP Outputs zero speed to port A.
20 OUT 191,14: OUT 223,15 Full Speed (15).
30 OUT 191,14: OUT 223,1 Lowest speed (1).
40 STOP
50 FOR a=1 TO 15
60 PRINT AT 10,15;a: OUT 191,14 OUT 223,a Steps through speeds from 1 to 15
70 PAUSE 100: CLS
80 NEXT a
90 FOR a=15 TO 0 STEP -1
100 PRINT AT 10,15;a: OUT 191,14:OUT 223,a Decreases speed down to stop.
110 PAUSE 50: CLS
120 NEXT a
130 OUT 191,14: OUT 223,31 16 added to output number setting bit 4 of data to 1, thus operating direction relay.
140 OUT 191,14: OUT 223,17
```

Figure 2.



Speed control data			
D_3	D_2	D_1	D_0
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	0
1	0	1	1
1	1	0	0
1	1	0	1
1	1	1	0
1	1	1	1

the required speed. By using suitable software, advantage may be taken of the computer speed to carry-out the operation much more smoothly.

In its most basic form, a speed controller is simply a rheostat wired in series with the power supply and that is used to vary the track voltage and hence the speed of the train. The problem of slow running is that the slightest amount of oxide build-up on the track will prevent the low voltage reaching the loco and thus the train stops.

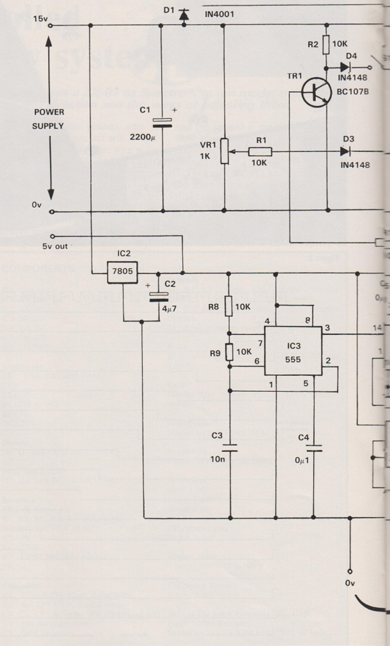
Varied pulses

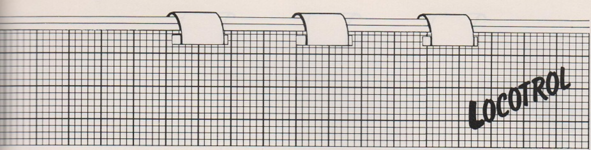
Several methods have been used to overcome the difficulty. In the controller to be described, power to the track is controlled by being applied in pulses of varying length — but always at maximum voltage. That improves the slow-running performance but it should be borne in mind that dirt, oxides and the like which build up on the track rapidly can still cause unprogrammed stoppages.

The circuit diagram of the controller is shown in figure one. The 15V input is obtained from a model railway power supply and they are usually unsmoothed DC and although generally regarded as supplying 12V, their output is usually several volts higher. Throughout this article the supply will be referred to as 15V but that is a nominal figure. Capacitor C1 smoothes the supply and diode D1 isolates the electronics from the railway supply.

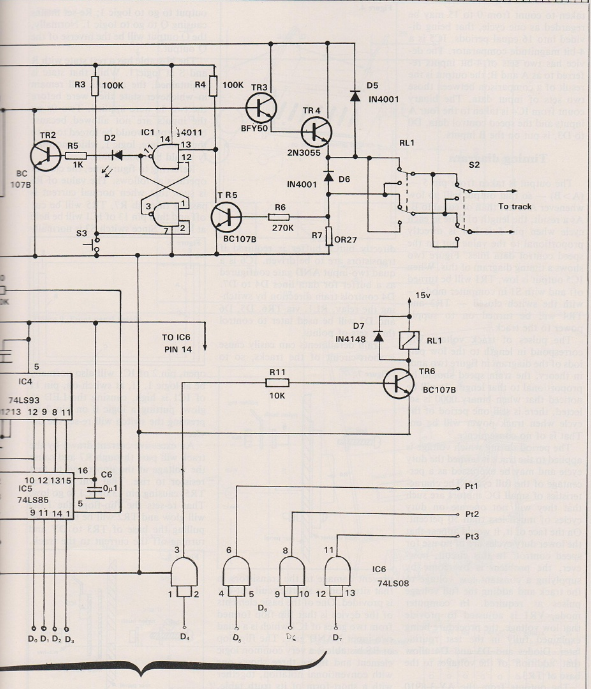
With the unit in manual mode, with S1 open, VR1 varies the current to the base of TR3 which together with TR4 forms a Darlington pair controlling the track voltage. Diodes D5 and D6 protect the transistors from voltage spikes produced by the train motor. Train direction is switched by S2. The 5V regulator, IC2, supplies the computer-directed speed control. IC3 is connected as a free-running clock providing pulses at around 5kHz. Those pulses are used to trigger IC4 which is a 4-bit binary counter. That device counts from 0 to 15 repeatedly, the count being incremented by each clock pulse. The time

Figure 1. Circuit diagram.





LOCOTROL



R3 100K
R4 100K
TR2
BC 107B
R5 1K
D2
IC1 14011
14
13
12
1
2
7
S3
T R5
BC107B
R6 270K
R7 OR27
TR3 BFY50
TR4 2N3055
D5 IN4001
D6 IN4001
RL1
S2
To track
15v
D7 IN4148
RL1
R11 10K
TR6 BC107B
TO IC6 PIN 14
5
IC4 74LS93
1213 12 9 8 11
16
10 12 1315
IC5 74LS85
5
9 11 14 1
C6 0µ1
3
1 2
6
4 5
8
9 10
11
12 13
IC6 74LS08
Pt1
Pt2
Pt3
D0 D1 D2 D3
D4
D5
D6
D7
D6

taken to count from 0 to 15 may be regarded as one cycle, that being divided into 16 equal periods. IC5 is a 4-bit magnitude comparator. The device has two sets of 4-bit inputs referred to as A and B; the output is the result of a comparison between those two sets of input data. The binary count from IC4 is taken to the four A inputs and the speed control data, D0 to D3, is put on the B inputs.

Timing diagram

The output is taken from pin 5 — (A > B) — so that output will go low whenever A is less than or equal to B. As a result, the length of time in each cycle when pin 5 is low is directly proportional to the value put on the speed control data lines. Figure two shows a timing diagram of this. When IC5 output is low, TR1 will be turned off and with S1 in computer mode — with the switch closed — TR3 and TR4 will be turned on to supply power to the track.

The pulses of track voltage will correspond in length to the low periods of the diagram in figure two and, in theory, the train speed should be proportional to that length. It will be noticed that when binary 0000 is selected, there is still one period of the cycle when track power will be on. That is of no consequence.

The period during which voltage is applied to the track is termed the duty cycle and may be expressed as a percentage of the full cycle. The characteristics of small DC motors are such that they will not operate on duty cycles of much less than 50 percent. On the face of it, it would appear that the lower duty cycles are of no use for speed control. In this circuit, however, the problem is overcome by supplying a constant low voltage to the track and adding the full voltage pulses as required. In computer mode, VR1 is adjusted to provide that low voltage, the procedure being explained fully in the test routine later. Diodes and D3 and D4 allow that 'addition' of the voltages to the base of TR3.

The outputs from the AY-3-8910 sound chip will drive TTL devices directly but a buffer is required if transistors are to be driven. IC6 is a quad two-input AND gate configured as a buffer for data lines D4 to D7. D4 controls train direction by switching the relay, RL1, via TR6. D5, D6 and D7 will be used later to control three sets of points.

Train derailments can easily cause a short-circuit of the tracks, so to prevent damage to the transistors in that situation, a short-circuit cut-out is provided. One of the basic elements of the device is that flip-flop formed from two gates of IC1 which is a quad two-input NAND gate. The flip-flop or RS bistable is a very common logic element and figure three shows one with conventional notation, together with a short-form of its truth table. The term set means causing the Q output to go to logic 1; Re-set means causing $\overline{Q}$ to go to logic 1. Normally, the $\overline{Q}$ output will be the inverse of the Q output.

The bi-stable has a rest state with R and S at logic 1. While that state is maintained, the outputs will remain in whichever state they were before the two 1s were applied. Two 0s on the inputs are not allowed because both outputs would be forced to go to the same state, logic 1, which normally would be undesirable.

Referring to figure one, the cut-out operates as follows. The value of R6 is such that when normal current is passing through R7, TR5 will be cut off and thus pin 13 of IC1 will be held at logic 1. Since switch S3 is normally open, pin 2 of IC1 will also normally be at logic 1. If, at switch-on, pin 11 of IC1 is high, causing the LED to glow, putting a logic 0 on pin 2 by pressing the switch will re-set the device.

An excessive current drawn by the track will pass through R7 and cause the voltage at the upper end to that resistor to rise. That will switch-on TR5, causing pin 13 of IC1 to go low. That re-sets the flip-flop; the LED will glow and TR2 will be turned on, pulling the base of TR3 to OV and turning-off the current to the track.

Figure 4.

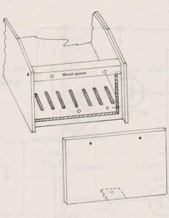


Figure 7.

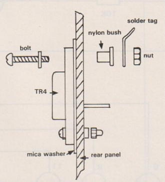


Figure 5.

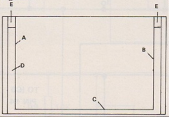


Figure 8.

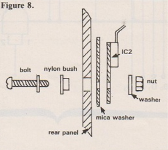


Figure 6.

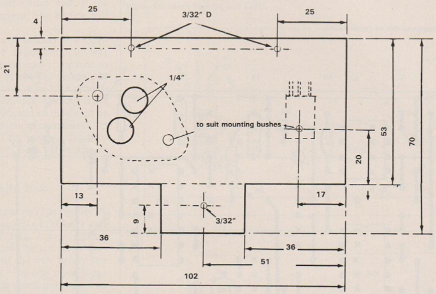
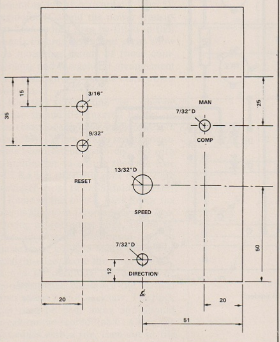


Figure 9. Front panel details.



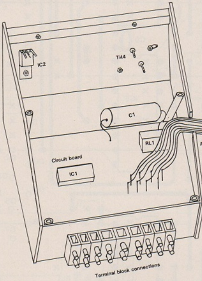


Figure 11. General arrangement of components in case.

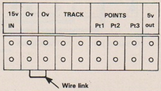
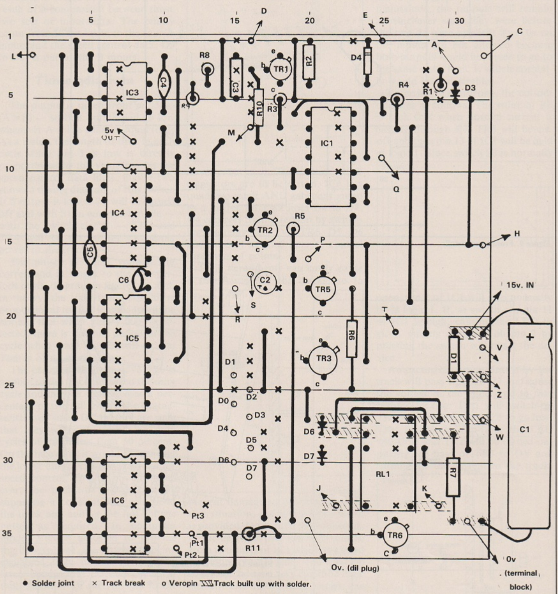


Figure 10. Component side of circuit board.



Current may be restored only by pressing the re-set switch and, if the fault is still present, the device will operate again.

The controller is housed in a Maplin-type M6005 type box. It was chosen for its lower cost compared to all-metal cases of similar design. As purchased, the box is an ABS moulding with an aluminium sloping front panel. The box should be modified by fitting an aluminium rear panel on which to mount the power transisitor and 5V regulator. Figure four depicts the modification.

Rigid box

The spacer maintains the rigidity of the box after removal of the back and is made from 5/16in. square section wood. That should be cut to fit between the sides of the box before the back is removed. The rear panel is cut from 1mm. thick aluminium sheet to the dimensions given in figure six and drilled as shown. The location of one bolt hole for the transistor is given; the remaining holes may be located using the mica washer from the mounting kit as a template. The 5V regulator IC2 and the power transistor TR4 should then be fitted to that panel. Figures seven and eight show the mounting details of those components and the general arrangement shown in figure 11 should clarify their position.

Heatsink compound or silicone grease should be smeared lightly on both sides of the mica washers before assembly. When those components have been fastened securely, test with a multimeter to ensure that the case of TR4 and the metal part of IC2 are insulated from the aluminium panel.

Remove the plastic back from the box as shown in figures four and five, leaving a protruding edge, D, on either side for the aluminium back to fit against. Using a sharp blade and straight-edge, score the three lines A, B and C, outlining the area to be removed; then cut down the vertical lines with a junior hacksaw. The back can then be bent away and should break on the scored line C. Clean the cut edges with a file, then cut or file

Figure 12. Inside view of rear panel.

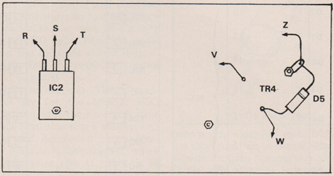


Figure 13. Wiring on underside of front panel.

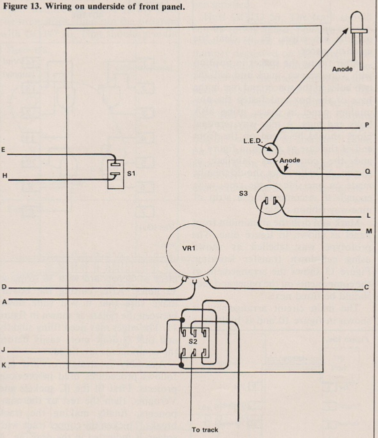


Figure 14.

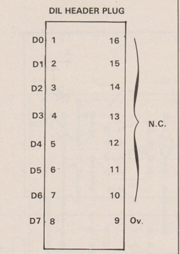


Figure 15b.

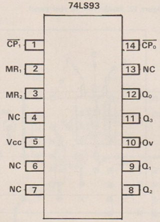


Figure 15c.

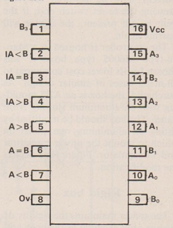


Figure 15d.

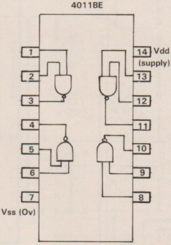
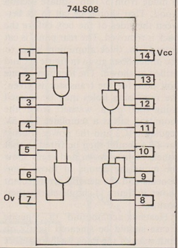


Figure 15e.



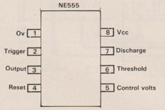
out the top edges, E, to clear the wooden spacer.

After fitting the spacer in position with wood screws, mark and drill the two holes in the spacer and one in the base of the box and fasten the aluminium panel in place, using 8BA bolts and nuts. A nine-way terminal block should be fitted on the front end of the case as shown in figure 11 and the connections labelled as shown. A shallow slot should then be made on one side of the case, wide enough to accommodate a strip of nine-way ribbon cable.

Mark and drill the aluminium front panel as shown in figure nine. The prototype was labelled as shown using rub-down transfer lettering. Figure 13 shows the arrangement of the components on the panel and they should be fitted next.

The main circuit arrangement is shown in figure 10 and is built on a

Figure 15a.



piece of Veroboard with 36 strips × 32 holes. After cutting the board, make sure that it will slide down between the pillars as shown in figure 11. The edges may need filing slightly and that is done more easily before the components are fitted.

Assembly is straightforward and follows procedures used in previous projects. First fit the IC sockets and Veropins, then the rest of the components, finally making the track breaks. Thicken the copper track with solder as indicated in the drawing. A piece of nine-way ribbon cable is used to connect the controller to the computer port board, one end being connected to the Veropins marked D0 to D7 and OV — DIL plug. The other end is fitted with a DIL header plug connected as shown in figure 14.

A 7in. flying lead made from flexible wire is soldered to each of the remaining Veropins. At that stage the circuit board should be checked for omissions and solder bridges across the tracks and it would be advisable to check with a multimeter to ensure that the 15V, 5V and OV lines are not short-circuit at any point.

If all is satisfactory, the ICs may be fitted. IC1 is a CMOS device and the

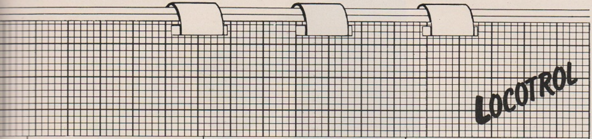
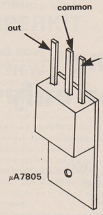


Figure 15f.



usual precautions should be taken to avoid damage by static. Avoid touching the pins and work on an earthed metal surface while fitting the device. Next fit the circuit board in the case as in figure 11. All the flying leads not marked by letters — figure 10 — are passed through the slots at the front of the case, trimmed to length, and connected to the appropriate terminal. Figure 12 shows the connections to be made to the rear panel. Having done that, the front panel should be placed alongside the case and the remaining wires connected as in figure 13. That done, the panel may be fitted to the case, taking care not to trap any wiring.

Any model railway DC power supply should be suitable for the controller but care must be taken to ensure that the polarity is correct, otherwise the circuit will suffer damage and, in particular, the smoothing capacitor, C1, will over-heat and explode dramatically.

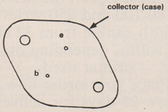
I used a Hornby R915 as a power source for the prototype and that unit has a built-in reversing switch which could lead to more confusion. If such a reversing switch is fitted to your unit, first tape it down firmly in one direction; then, using a voltmeter, ascertain which terminal is positive and connect red and black leads appropriately. Set up the railway and connect the controller as in figure 16. C7 helps to limit interference from the track end.

With S1 in the manual position, test the speed control and direction switch. With the train running, short-out the tracks; the train should stop and the LED light. Press the re-set button and the train should re-start. If all is well, the computer control should be tested.

Enter program one. The port addresses are for the original PSG board featured in the June/July issue and will need changing if another I/O port is used. Before switching to computer mode, run the program, which will stop after line 15. That is to ensure that the train is stopped when the computer takes control. Switch S1 to computer then use VR1 to start the train. Reduce the speed slowly until the train stops, then run the program with GOTO 20. That should give the train a very short full-speed burst — too short to notice — then stop at line 40 with the train running very slowly. If the train stops or is moving too quickly, repeat the precedure, making small adjustments to VR1 until you are satisfied.

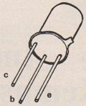
Next GOTO 50. The train should run up to full speed, pausing at each speed increment with the increment number displayed on the screen. The speed should then reduce in a similar manner until the train stops, at which point it should start again in the opposite direction, running at speed one. The three-point control outputs may be tested using a voltmeter. Putting the values 32, 64 and then 128 on port A should result in a logic 1 appearing on Pt1, Pt2, and Pt3 respectively. Finally, check that 5V is available on the 5V OUT terminal.

Figure 15h.



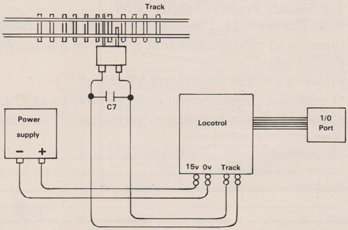
2N3055 (pin side)

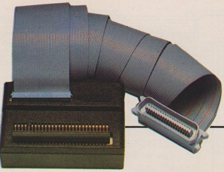
Figure 15g.



BC107B and BFY50

Figure 16.





Save £££s
Improve your printing
Kempston Interface E for only
£41.25 (plus p&p)

Recently-released, the latest printer interface from Kempston Microelectronics for the Spectrum normally costs £55. We have negotiated an exclusive discount of 25 percent. All you have to do with the Interface E is to plug it in and it will reproduce high-resolution graphics on most Epson (or derivatives) and Seikosha 80,100 and 250 series printers. The print width can be from 32 characters to full width, depending on the printer used.
Place your order now.
Allow 28 days for delivery.

The kit is available by mail order only. No callers, please. The prices apply to the United Kingdom only. For overseas orders please add £4.

To: Sinclair Projects Special Offer,
ECC Publications, 196–200 Balls Pond Road, Islington, London N1 4AQ

Please send me Interface E at the special *Sinclair Projects* price of £41.25 plus £1 p&p.

Please tick if you require a VAT receipt ☐

*I enclose a cheque/postal order payable to ECC Ltd for £ ____________

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

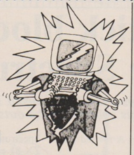
*Please delete/complete as applicable

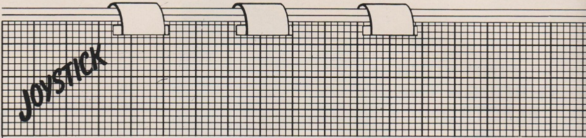
Signature ____________

Name Mr/Mrs/Miss ____________

Address ____________

____





Avoid your joystick blocking the sound on the Spectrum

In the October/November issue we built a joystick which caused some problems. Corin Howitt has followed his previous article with this solution to the outstanding difficulties

SINCE PUBLISHING the joystick conversion project in the October-November issue of *Sinclair Projects,* we have received a number of problem letters from readers. The most common problem has been the project causing screen border colour changes and sound loss while in operation.

To understand why that is happening look again at the original circuit diagram in figure one. The circuit detects when the Spectrum is addressing port 254 (A0 and $\overline{IORQ}$) and which keyboard half-row (A15-A18). As page 160 of the Spectrum manual will indicate, port 254 addresses the keyboard IN INPUT, i.e., when $\overline{RD}$ goes, low, port 254 in OUTPUT: when $\overline{WR}$ goes low, it controls the border colour (D0-D2), the loudspeaker (D4) and the MIC level (D3).

So the project did not detect whether the Spectrum was using port 254 in input or in output and consequently the border colour and loudspeaker were affected. The solution is straightforward. Simply connect a second diode between the $\overline{RD}$ line and the junction of D1 and R1. That adds the condition that the buffer (IC3) will be enabled only when A0,$\overline{IORQ}$ and $\overline{RD}$ go low, showing that the Spectrum is addressing port 254 in input. A revised circuit diagram is shown as figure one. A revised parts list is also included.

Two readers, A W Stanworth and R Morris, pointed out that the Atari and Commodore joysticks mentioned at the end of the previous article have switches which are commoned at one end. You cannot use them therefore in the manner stated. There are two possible methods of using those types of joystick switches as "control" inputs to a second switch which connects the appropriate row and position terminals. Brief explanations are given under each circuit.

Many readers have also asked whether the joystick project can be used with the ZX-81. The circuit remains unaltered for the ZX-81, except that you use a 23 + 23-way ZX-81 connector in place of the 28 + 28-way Spectrum connector and follow the ZX-81 edge connector pin-out given in this magazine.

The keyboard layout is also different from that of the Spectrum. Noting those points, the circuit will function in exactly the same way except that you must follow the ZX-81 keyboard layout — figure three — when selecting which keys the joystick will mimic.

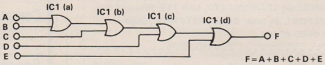
Two or more joysticks can be used simultaneously so long as they are isolated from each other and do not interact. That can be done by using a diode (1N4148) arrangement or by using one of the circuits outlined in figure two.

When using diodes, take care to see that they do not prevent ICs 1 and 2 sinking the pull-up current from R2-

REVISED PARTS LIST

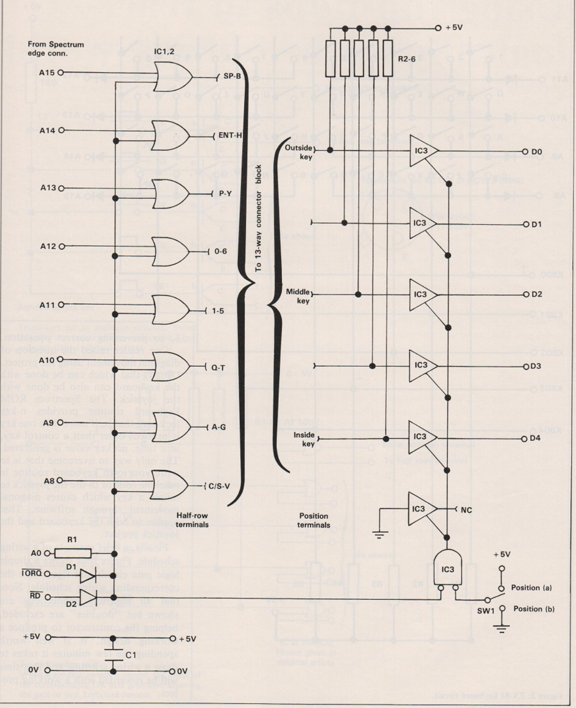
- ICl, 2 are 74LS32
- IC3 is 74LS365
- Rl is $\frac{1}{4}$W carbon 470R (E12)
- R2-6 are $\frac{1}{4}$W carbon 100K (E12)
- Dl, 2 are 1N4148
- Cl is 100nF ceramic disc
- SW1 is spdt toggle
- 28 + 28 way Spectrum edge connector
- 13-way connecting block
- 2 6BA nuts and bolts
- Vero VQ board
- Connecting wire, solder, etc.

Figure 4: Example circuit.



Terminal	IC1:74L532 (pin no.)	Wiring Schedule
A	1	
B	2	
C	5	
D	9	
E	12	
F	11	
IC1 (pin)		
1	—	
2	—	
3	4	Each element represents a connection
4	—	
5	—	
6	10	
8	13	
9	—	
10	—	
11	—	
12	—	
13	—	

Figure 1: Revised joystick circuit.



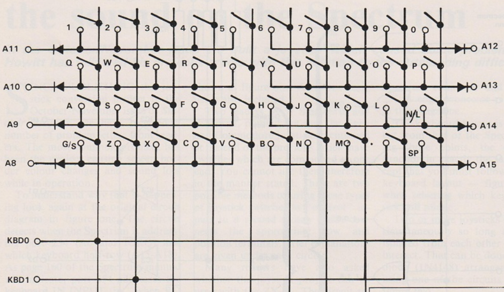
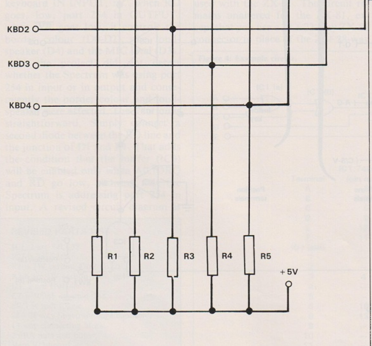



Figure 3: ZX-81 keyboard circuit.

6, so preventing correct operation.

One reader raised the question of diagonal movement using the project. Usually that which can be done with the keyboard can also be done with the joystick. The Spectrum ROM keyboard routine provides n-key lock-out; that is, if more than one key is pressed, other than a control key, at a time, no key value is generated. The only way to overcome that is to write your own keyboard routine in machine code or to use the joystick to mimic a key which causes diagonal movement through software. That applies to both the keyboard and the joystick project.

Finally, a brief example of a wiring schedule. Figure four shows a simple logic gate circuit and figure five the corresponding wiring schedule. Note that all electrical connections are shown but "doubles" are excluded, helping the constructor to produce a working design. It is well worth spending the few minutes it takes to draw a wiring schedule and that time will be rewarded with a working project.

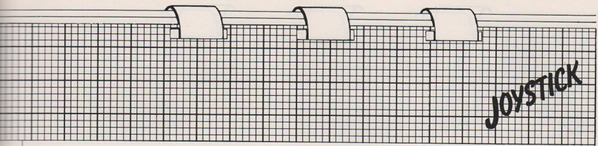
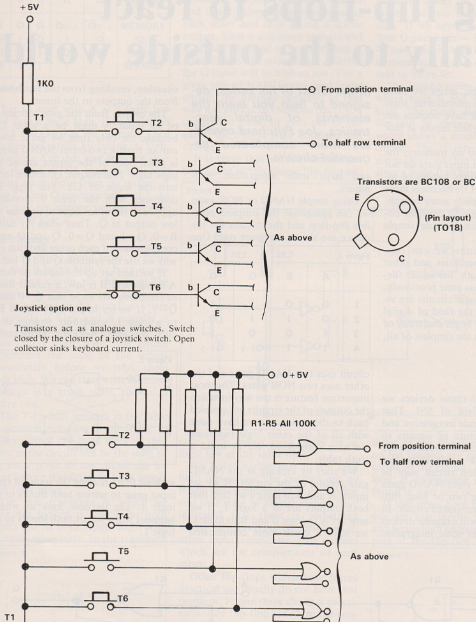


Figure 2.



Joystick option one

Transistors act as analogue switches. Switch closed by the closure of a joystick switch. Open collector sinks keyboard current.

Joystick option two

When a joystick switch is closed, the input of the corresponding OR gate goes low, allowing the gate to sink keyboard current.

Using flip-flops to react logically to the outside world

In the latest of his series, designed to help you learn the elements of digital electronics, Joe Pritchard considers more complicated sequential circuits.

SO FAR in this series we have looked at combinatorial logic circuits which have outputs dependent solely on their inputs at that moment.

The circuits have no memory; the outputs for a given input state do not vary with the previous history of inputs to the device. Life becomes more interesting and slightly more complicated when we begin to consider circuit elements which provide simple memory functions.

With those circuits we can construct counters, memories and other logic functions with behaviour depending on what has gone previously. Those sequential logic circuits are vitally important in the field of digital electronics and we begin our study of them by looking at the simplest of all, the flip-flop.

Field of SSI

In the study of those devices we move into the field of SSI. That stands for small scale integration and it is the manufacture of circuits to perform logic functions which require a few gates on a single chip. For example, a flip-flop circuit may be built out of six or eight NAND gates and there may be two or four flip-flops on a single integrated circuit. In future articles we will examine devices built with medium scale integration and large scale integration techniques.

Using simple NAND or NOR gates we can synthesise the simplest possible flip-flop and the circuits for the device are shown in figure one. One circuit uses two NAND gates and the other uses two NOR gates. The most important feature is the way in which the outputs of the circuits are coupled back to the inputs. As we have done with all of the other logic functions we have considered, let us build a truth table.

Figure 2.

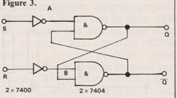
	A	B	Q	$\overline{Q}$
1	0	0	1	1
2	0	1	1	0
3	1	0	0	1
4	1	1	Q_{T-1}	$\overline{Q}_{T-1}$

We start by looking at the NAND gate version of the circuit. If we set inputs A and B to logic 0 we find that both outputs are at a logic 1. If we now set A to logic 0 and B to logic 1, we have a much more complicated situation, resulting from the feedback from the outputs to the inputs.

The output from the gate connected to input A must be a logic 1, because the only time we get a low output from a two-input NAND gate is when both of the inputs are at a logic high. That output, Q, is then fed into the input of G2. The logic 1 combines with the logic 1 already applied to the other input to provide a low output at $\overline{Q}$. Thus when A1 and B = 0, Q = 1 and Q = 0, Q and Q are complements of each other and that is why we use the symbols Q and Q-bar.

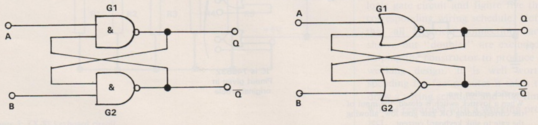
If we then set up the inputs so that A is high and B is low, a similar line of reasoning will show that Q = 0 and $\overline{Q}$ = 1. If we try to put both A and B to a logic 1, something strange occurs. We cannot work out the outputs

Figure 3.



without first knowing the states of the input prior to setting both inputs to a logic 1. Let us now look at what happens when we set both inputs to a logic 1.

Figure 1.



S	R	Q	$\overline{Q}$	MODE
1	0	0	Q_{T-1}	$\overline{Q}_{T-1}$	MEMORY
2	I	0	I	0	SET
3	0	I	0	I	RESET
4	I	I	I	I	INVALID

Figure 4.

Let us start with A at a logic low and B at a logic high, thus giving us Q=1 and $\overline{Q}$=0 at the outputs. Then change the inputs by taking A high. Changing A in that way does not alter the output of G1 because it is still seeing a low input from the output of gate 2. Thus Q retains a value of 1. As that does not change, neither does the output of gate 2.

No change

So, in brief, going from A being low and B being high to both inputs being high causes no change at the outputs. A similar line of reasoning can be applied to the situation where A initially is high and B initially is low. Thus, whenever we take the two inputs to a high state, the circuit remembers the state of the inputs immediately before we took them both high. We can thus write all those findings in a truth table, shown in figure two.

The T-I which appears in the table is common notation in this field. It indicates that the logic state of that point in the circuit will be the state of the same point in the circuit the instant before transition to both inputs high took place.

The one transition at which we have not looked is that from both inputs low to both inputs high, with no intervening state. In the transition period the outputs oscillate between 1 and 0 until they are complementing each other and thus giving a stable output. That is a random process and it is not possible to predict how a given circuit will settle down, whether the Q input will be high or low. For a given pair of NAND gates used in the circuit, the outputs obtained in that way will always be the same but if you change the gates, or even one of the gates, the output state obtained will change.

It is there that we find one of the limitations of this simple flip-flop circuit; both inputs must never be at logic low at the same time.

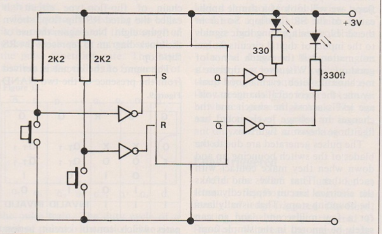


Figure 5.

We noted earlier that if you take both inputs low, the outputs are no longer the complements of each other and that state of affairs should have indicated that there is something wrong with this input condition, as all other input conditions give outputs which are the complements of each other.

Other flip-flops which have found practical use usually do not have that problem. Figure three shows a practically-used flip-flop based on the configuration we have just examined. The only changes are the inverters on the two inputs. That gives a slightly different truth table. The new inputs are labelled SET (S) and RESET (R).

Figure four shows the new truth table. The names of the inputs result from the effect on the Q output of taking the two inputs to logic 1 in turn. If we apply a logic high to S, then Q goes high — it is said to be set — and a logic high applied to the R terminal sets Q to a low state — it is said to be re-set.

The invalid mode in that circuit is where both S and R are at a logic high. The block digram for an SR flip-flop is shown in figure five, which shows a simple demonstration circuit. The inverters on the inputs are present, so that pressing a switch gives a logic L to either the S or the R input of the flip-flop. The inverters on the outputs are part of the monitoring of the circuit. A more detailed look at monitoring outputs can be found in part one of the series.

If you press both buttons at once, you get low inputs to the inverters and hence high outputs to the S and R pins of the flip-flop. That gives the

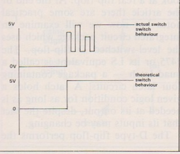


Figure 6.

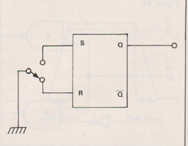


Figure 7.

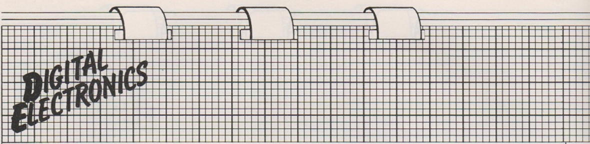
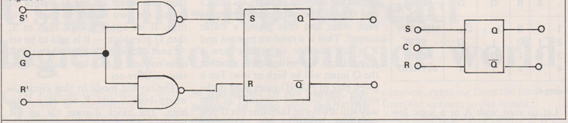


Figure 8.



invalid state. You might like to build the circuit on a breadboard and confirm the truth table, remembering that pressing a button gives a logic 1 to the relevant input to the flip-flop.

Before proceding to other flip-flops, we will look at a simple application of the SR flip-flop. So far in the series, when applying logic signals to the inputs of digital circuits, it has not mattered if the switch is not of good quality. Whenever we operate a mechanical switch, even a good quality one, the theoretical change in voltage levels across the switch and the changes in voltage levels noted are like those shown in figure six.

The pulses generated are due to the blades of the switch bouncing up and down when they make contact with each other. That makes and breaks the electrical circuit repeatedly until the bouncing stops. That usually lasts for a few milliseconds and so can safely be ignored in the simple combinatorial circuits we have so far seen.

When we begin to deal with circuits which count pulses or circuits which can remember previous inputs, it is important to stop these pulses. The process is called switch debouncing and can be carried-out with a simple SR flip-flop, as in figure seven. Thus by actuating the switch we can get a clean change in logic state at Q. Relays exhibit similar behaviour and there it is called contact bounce. Flip-flops can be used, therefore, in a similar way with relays.

The next step in the evolutionary chain of flip-flop-type circuits is called the gated RS flip-flop, shown in figure eight. Note again the use of the block diagram to represent the RS flip-flop.

The name of the circuit is derived from the presence of the two NAND

Figure 9.

G	S	R	Q	$\overline{Q}$
0	X	X	Q_{T-1}	$\overline{Q}_{T-1}$
1	0	0	Q_{T-1}	$\overline{Q}_{T-1}$
1	0	1	0	1
1	1	0	1	0
1	1	1	INVALID	INVALID

gates which control circuit action. The circuit has a control input, the G or gate input which, when bought to a logic low, allows the circuit outposts to ignore any changes in the logic states of the inputs.

On allowing that line to go high, we get normal RS flip-flop behaviour. The introduction of the G input, which is also known as the clock input, gives two memory states in the truth table, whereas in the standard RS flip-flop there is only one memory state. The truth table for the configuration is shown in figure nine. Note how we still have an invalid state.

We now move to a type of flip-flop which disposes of this disadvantage. The circuit for the level-switched D flip-flop is shown in figure 10. Note how the inverter between input D and input R prevent both S and R attaining the same logic state. As the invalid state occurs only for an SR flip-flop when the two inputs are high, that can never occur. One of the memory modes of a gated affair can no longer happen; the only memory mode for that type of flip-flop is when the G input is low.

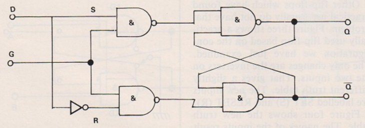
Logic state changes

Thus this circuit has evolved from the previous one but now has only one input, D, and a control input G. D stands for data and an examination of the truth table shows how the outputs respond to changes in the logic state of that line and the clock line. Whenever the clock line is at a logic low, the outputs remain unaffected by changes at the D input. If we take the G input high, the Q output will follow the D input logic states.

Let us now pause for breath and look at a real flip-flop. At the end of the article there are some practical notes but now we will examine an integrated circuit package which uses the level-switched D flip-flop. The 7475 or its LS equivalent is called a quad latch — a package containing four latch circuits. A latch holds a given logic condition for as long as is needed at its output, despite the fact that its inputs may be changing.

The D-type flip-flop performs the

Figure 10.



function of a latch when we bring the G line to a logic low. The pin-out of the 7475 package is shown in figure 12. Lines D1 to D4 are the inputs to the four latches and Q1 to Q4 and $\overline{Q1}$ to $\overline{Q4}$ are the outputs from the latches.

The observant ones may have noted the presence of only two clock lines for the four latches. The explanation for this is that each clock line controls the action of two latches. Line G1 controls the behaviour of flip-flops 1 and 2 and line G2 controls the behaviour of flip-flops 3 and 4. By connecting G1 and G2 we can apply a clock signal to all the latches at once, thus latching all four input lines simultaneously. The 7475 or its LS equivalent is often used in non-microcomputer applications to store four bits of data in a logic circuit.

The flip-flops we have looked at so far have been what are called level-switched or gated flip-flops. The logic outputs depend on the logic level applied to the clock line in the circuits. Those devices form a minority grouping in the family of flip-flops and the majority are called edge-triggered flip-flops.

To understand what is meant by that description, we have to make a short excursion into the world of electrical pulses and pulse trains to examine the jargon involved. Figure 13 shows diagrammatically the descriptions applied to an electrical pulse. A pulse is best described as a short-lived disturbance in an electrical circuit which has a regular shape. The pulse shown in figure 13 is rectangular, although they they can be triangular or of a fairly complex shape. A pulse train is a series of pulses of a similar type. For a single pulse there is a rising edge and a falling edge, the rising edge being that part of the pulse which goes from a logic low to a logic high and the falling edge being that part of the pulse which goes from high to low.

The distance between the two edges is the pulse width and the number of pulses which occur in one second is the pulse repetition rate. The latter number is also called the frequency of the pulse train. The duty cycle of a train of pulses is the ratio of the time that the signal is at a logic high to the time that a signal is at a logic 1 low. The time the pulse is high, between the rising and falling edges of a single pulse, is called the mark period. The time between the falling edge of one pulse and the rising edge of the next pulse is called the space period. That is shown in figure 13b.

D	C	Q	$\overline{Q}$	MODE
X	0	Q	$\overline{Q}_{T-1}$	MEMORY
0	I	0	I	INPUT
I	I	I	0	INPUT

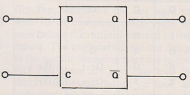


Figure 11.

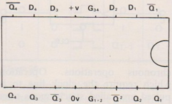


Figure 12.

The pulse height, in volts, is the difference between the voltage of a pulse in its mark state and that in its space state. To be useful in digital logic circuits, a pulse must have a mark state with a voltage above the logic 1 threshold for the devices being used and a space state which has a voltage below the threshold for a logic 0.

In the diagram, the pulse goes straight from logic 0 to logic 1 instantaneously. Unfortunately, that is not the case in real life. Figure 14 shows how a real pulse might look. The rise and fall times of a pulse are important only when we are working at high speeds.

Edge-triggered

After the brief interlude, let us return to flip-flops and examine the family of edge-triggered flip-flops, starting with the edge-triggered D flip-flop. Whereas the previous devices had to be put into the memory mode by the user providing a certain set of inputs to the data and the clock lines, this type is in the memory state most of the time.

The only time the outputs change their logic state is in the short period needed for the signal applied to the clock line to go from logic 0 to logic

Figure 13.

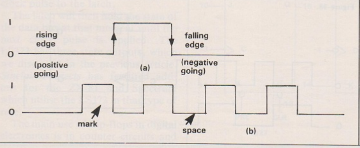
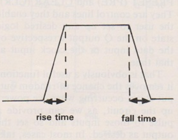


Figure 14.



D	C	Q	$\overline{Q}$
1	R	1	0
0	R	0	1
X	0	Q_{T-1}	$\overline{Q}_{T-1}$
X	1	Q_{T-1}	$\overline{Q}_{T-1}$

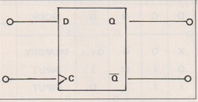


Figure 15.

1. Thus it is triggered on the rising edge of the clock pulse. Figure 15 shows the logic symbol for that device. From now we shall treat the flip-flops we encounter as black boxes; it is not necessary to know the intricate details of their internal construction to use them in circuits.

Figure 15 also shows the truth table for this flip-flop and has a new symbol in it. R indicates a rising edge at a given point in the circuit; here it shows that the change occurs where we have a rising edge on the clock line. Alternative symbols for the rising edge are ↑ and ↑ . Similar symbols ↓ and ↓ are used to indicate that the changes occur when we have a falling edge. From the truth table we can see that when the clock line is at either logic high or logic low; the outputs do not change.

One point to note with regard to edge-triggered D flip-flops is that some of them have inputs designated PRESET (PRE) and CLEAR (CLR). They are control lines and they enable the user to set up any desired logic state on the Q output irrespective of the data input or the clock input at that time.

That is obviously a useful function; it removes the chance of random output states occurring when we power-up the circuit, as we can provide a pulse to those inputs and so set the output as desired. In most cases, taking the PRE input to a logic low will result in a Q attaining a high state; taking CLR low will result in Q attaining a logic low.

The pre-set and clear operations occur with no interventions from the clock line. They are said to be asynchronous operations. Operations which require the clock line to be pulsed are called synchronous operations.

Figure 16.

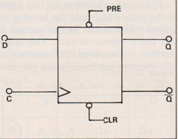
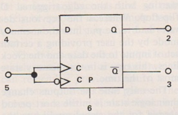


Figure 17.



The logic symbol for a D-type flip-flop with pre-set and clear is shown in figure 16. There are various edge-triggered D-type flip-flops available in TTL and CMOS logic; the 74174 is a hex flip-flop with CLR but no $\overline{Q}$ output. That is no loss, as the $\overline{Q}$ output is the complement of the Q output. The 74175 is a quad flip-flop with CLR but no pre-set. They are available in the LS series as well.

Moving to CMOS devices, the 4013 is a dual flip-flop with pre-set and clear functions. In this case the CLR and PRE lines must be taken to a logic high to operate; that is the reverse of the general situation.

Back to TTL and the 7475 is a dual flip-flop chip which has PRE and CLR inputs which need to be taken to a logic low to operate.

The final flip-flop of this type we will look at is the 4042, a CMOS device. Its interesting feature is that we can program it to respond to either a rising or a falling edge depending on the logic state on its programming input.

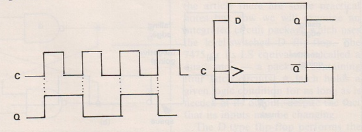
CMOS flip-flops

The logic symbol for the device is shown in figure 17. If P is taken to a logic low, the flip-flop responds to a rising or positive edge and if it is at logic high, the flip-flop responds to falling or negative edges. The 4042 has no asynchronous — pre-set or clear — functions.

Before looking at other members of the flip-flop family, we will examine a few applications. As usual, the CMOS flip-flops are slower than their TTL counterparts, the TTL devices having a maximum clock rate of 35MHz.

The propagation delay of the CMOS flip-flops is of the order of 200 to 300 nanoseconds, as opposed to the 20 to 30 nanoseconds typical for a TTL flip-flop. Those figures are the times which elapse between the application of a PRE signal and the outputs acquiring the desired logic states. It should be noted that the CMOS times are dependent on the supply voltage. The flip-flops are slower at lower supply voltages than they are at higher supplies.

Figure 18.



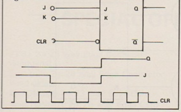


Figure 19.

One simple application of an edge-triggered D-type flip-flop is as a simple frequency divider circuit, a circuit which accepts a train of pulses as an input at one frequency and provides an output pulse train whose frequency is less than that of the input.

In our simple example the output frequency is half the input frequency. The circuit for that is shown in figure 18. On the first rising edge, Q attains a high state. That assumes we start with Q at logic low. The flip-flop then keeps that output condition until the next rising edge. Then the input on D is put on to the output Q. The input to D comes from Q, which will be low if Q is high. Thus Q goes low until the next rising edge, and Q, and hence D, are taken high. On the next rising edge, the high on D will be put on to Q and the cycle will be repeated.

Latching data

A more common use for flip-flops in computer circuits is for latching data in a similar fashion to the 7475. The 74LS373 is an octal latch which can thus remember the states existing on eight inputs when it receives a clock pulse edge. We can use it to interface devices to the computer data bus which in the Sinclair machines, QL excepted has an 8-bit data bus. The computer puts the information on to the data bus and then sends a clock pulse to the latch.

The latch will then hold the state of the data bus at that moment until the next clock pulse is applied. The LS373 has three-state outputs, which we discussed in the previous article. *Sinclair Projects* has featured add-ons for the ZX-81 and Spectrum which utilise the LS373 in that type of role.

The main use of flip-flops in digital electronics is in counter circuits and frequency dividers. Those two areas of application will be looked at in greater detail next time.

On to the most versatile type of flip-flop used in digital electronics. Any other type of edge-triggered flip-flop can be synthesised from this device and in many circuits that is how this device is used. The name of the device is the master slave JK flip-flop. The master-slave part of the title indicates that there are two flip-flops inside each device. Figure 19 shows the logic symbol for that type of flip-flop.

If we take both the J and K inputs to a logic low and then allow a complete clock cycle with the two inputs in this low state to occur, the outputs

J	K	C	Q	$\overline{Q}$
1	1	⎍	$\overline{Q}_{T-1}$	Q_{T-1}
0	1	⎍	0	1
1	0	⎍	1	0
0	0	⎍	Q_{T-1}	$\overline{Q}_{T-1}$

Figure 20.

will remember the state they were in at the end of the clock pulse prior to that we have just allowed to occur. That is the memory mode of the device.

If J and K are the complements of each other, Q takes on the input state of the J input after a complete clock pulse, of both rising and falling edge, has occurred. Contrast that behaviour with that exhibited by the RS flip-flop, where we did not have to wait for a complete clock cycle to pass before the outputs reacted to the inputs. The explanation for the delay in the output changing state lies in the fact that there are two flip-flops in the JK flip-flop. Before we look at how that works, look at the truth table shown in figure 20. The only line in it not yet considered is the state where both J and K are at a logic high.

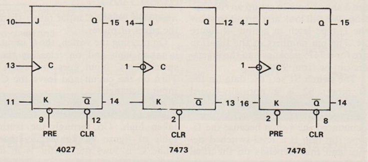
Toggling

In that case, after one complete clock pulse, output Q takes on the previous state of Q and Q takes on the previous state of Q. That process is called toggling. Let us look at the internal structure of the JK master-slave flip-flop and see how it operates. The master section of that flip-flop connects to the inputs J and K. On a positive-going clock pulse edge, those inputs are read and placed on the outputs. During the negative-going edge of the clock pulse, the data present at the output of the master is transferred to the inputs of the slave, and hence to the slave outputs, which are those we monitored to build the truth table.

Should the J and K inputs change during the clock cycle, there will be no change to the outputs of the master flip-flop until the next positive edge. In summary, the data is accepted at the inputs on the positive-going clock pulse edge but does not appear at the output until the negative-going clock pulse edge.

JK flip-flop can have pre-set and

Figure 21.



DIGITAL ELECTRONICS

clear inputs, or only clear inputs. Typical examples of the JK flip-flop are the 7473, the 7476 and the 4027. All of those integrated circuit packages have two flip-flops in them, as can be seen from figure 21. The numbers shown are the pin numbers of the device.

Clock pulse

The small circle on the clock input of the TTL devices indicates that the circuit responds to clock pulses in the way described. If the circle is absent, it indicates that the data is read from the J and K inputs on the negative-going edge of the clock pulse and not the positive-going edge.

Similarly, the data is put on the outputs of the slave on the positive-going clock edge. Some JK flip-flops have logic functions on the J and K data inputs.

An example is the 7472 device in figure 22. The three J inputs and the three K inputs are AND-ed together before reaching the flip-flop part of the device.

Thus to apply a high input to the J or K input of the flip-flop, all three inputs to that input must be high to satisfy the AND gate logic.

One of the main points of interest about the JK flip-flop is that you can synthesise other flip-flop types from it.

So, before looking at the practical demonstrations of flip-flop circuits, figure 23 shows how we can form other types of flip-flop from this versatile unit.

Note how the PRE and CLR inputs are used as S and R inputs when this flip-flop is being constructed.

As to practical demonstrations, figure 5 shows how to set up an SR flip-flop. We will look at a circuit to show the D-type latch in action — figure 24. The pin numbers of the flip-flop are for a 7475 package and the inverters can be from 97404.

The switch S1 sets the data on the D input; S2 is a normally-closed switch which provides a clock pulse when pressed. Note that we have not debounced the switch because the D-type flip-flop is a level-switched device and thus depends on the final logic state on the C input rather than the edges of pulses.

On pressing the switch, the C input is taken high and on releasing the switch the C input goes low again. Turn on the circuit and leave the C input unchanged.

Altering the D input causes no change at the output until the C line is taken high. Go through the truth table of figure 11, confirming that while C is low, the circuit is in a memory mode. Try the other circuits, remembering that if you use CMOS flip-flops you must interface the CMOS outputs to the 7404 gates used for monitoring.

By doing these experiments, you will soon realise that clocking digital circuits using pushbuttons is a little clumsy because of switch bouncing. What is required is a means of providing single pulses and pulse trains when required.

Figure 22.

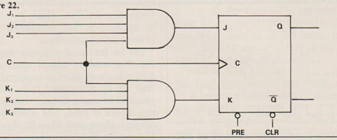


Figure 23.

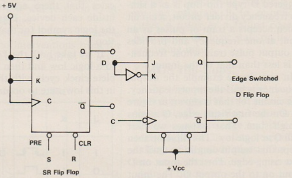
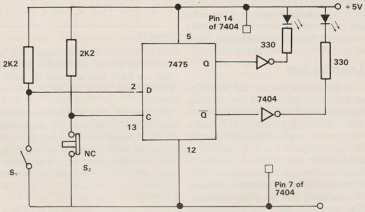


Figure 24.



INNOVONICS

Robotics — Design & Marketing

As a reader of Sinclair Projects your commitment to microcomputing is something more than that of the 'talk only' enthusiast. To build is to learn and every builder needs his materials, so take advantage NOW of our

SPECIAL OFFER

	5 Off	*10 Off*
ZX81 2 × 23 way Edge Connector (long pins)	£8.25	£15.00
Spectrum 2 × 28 way Edge Connector (long pins)	£8.75	£16.50
ZXTONGUE 2 × 23 way Edge Card	£4.25	£7.50
SpecTONGUE 2 × 28 way Edge Card	£4.75	£8.50

As ever all prices inclusive of P&P etc, no extras

STOP PRESS! — Write now for inclusion in the INNOVONICS Robot Bits mailing list for the parts your brainchild needs!

All enquiries/orders by POST please to:

147 Upland Road, East Dulwich, London SE22 0DF

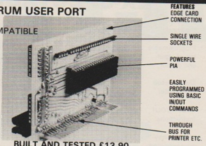
MULTITRON

ZX SPECTRUM SPECIALISTS
HIGH QUALITY — LOWEST PRICES

No 1

ZX SPECTRUM USER PORT

24 I/O LINES
μ–DRIVE COMPATIBLE



BUILT AND TESTED £13.90

Ideal for automation, control, robotics etc etc.

★ **NEW** ★ 16 line I/O port. Features: Flexible addressing, crash reset button, full handstarting & 16 I/O lines. Ideal for centronics printer port & robotics, control, etc. Price: Built & Tested £13.90, Kit £11.90.

I/O PORT SUPPORT MODULE 1

8 Channel A/D Controller Features:

1) 8 analogue input channels
2) Fast conversion time
3) Choice of accuracy
4) Plugs onto the Multitron I/O port
5) Manual supplied

★ Transducers, Control IC's, books etc. SAE for lists. ★

1 L.S.B. £14.60 — ½ L.S.B. £17.75 — ¼ L.S.B. £18.95

CONNECTORS

2×28 way Spectrum female	£2.50
2×28 way Spectrum spade	£0.75
Single wire socket (per 20)	£1.75
Wire pack for bread-boarding	£0.99

PLEASE ADD VAT AND 60p P/P TO TOTAL ORDER

MULTITRON
5 MILTON CLOSE, HEADLESS CROSS, REDDITCH B97 5BQ

FOUR SINCLAIR SPECIALS FROM NEWNES

ZX SPECTRUM USER'S HANDBOOK

R J Simpson and T J Terrell

This is an introductory book on the Spectrum that takes you a lot further, including extensive coverage of the Spectrum hardware and machine code programming.

Softcover *210 pages* *0 408 01323 0* *£6.95*

INTRODUCING Z-80 ASSEMBLY LANGUAGE PROGRAMMING

Ian Sinclair

Overcome the limitations of BASIC, using this introductory guide to assembly language by top author Ian Sinclair. You can write programs which are more versatile and much faster using assembly language, and this book includes tips on program design and techniques.

Softcover *128 pages* *0 408 01338 9* *£6.50*

ZX81 USER'S HANDBOOK

R J Simpson and T J Terrell

This book takes you from the first steps with your ZX81 to more advanced features like machine code programming – ideal for first-time users who want a book that they won't grow out of quickly.

Softcover *144 pages* *0 408 01223 4* *£5.95*

ZX81 BASIC BOOK

Robin Norman

An ideal introduction to BASIC programming on the ZX81 for the complete beginner. Robin Norman explains clearly and carefully all the different BASIC commands.

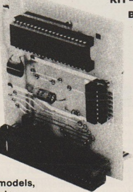
Softcover *176 pages* *0 408 01178 5* *£5.95*

Available from bookshops or direct from the publishers

Newnes Technical Books, Borough Green, Sevenoaks, Kent TN15 8PH

INPUT/OUTPUT PORT FOR ZX SPECTRUM

* 3 Channels (24 lines)
* Programmable mix of input and output
* May be used with printer
* Uses IN/OUT instructions
* Up to 8 may be connected



KIT - £16.95

BUILT - £18.95

(Output connector £3.25 if required)

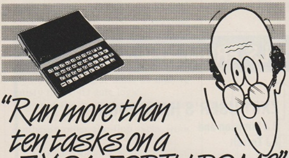
Includes VAT and Inland Postage

Ideal for controlling models, motors, robots, automatic equipment, security systems etc. practical applications circuits included.

Export postage (surface) £2.50. Send SAE for full catalogue of Spectrum and ZX81 accessories. Cash with order or ACCESS.

REDDITCH ELECTRONICS

21 FERNEY HILL AVENUE, REDDITCH, WORCS B97 4RU



"Run more than ten tasks on a ZX81-FORTH ROM?"

Sure! More than 10 tasks simultaneously and, in some cases, up to 300 times faster! That's what replacing the basic ROM with the new FORTH does for the ZX81 – and more!

The brains behind the breakthrough belong to David Husband, and he's building Skywave Software on the strength of it. Already orders are flooding in and it's easy to see why.

The ZX81-FORTH ROM gives you a totally new system. In addition to multi-tasking and split screen window capability, you can also edit a program while three or four others are executing, schedule tasks to run from 50 times a second to once a year, and with a further modification switch between FORTH and BASIC whenever you like.

The ZX81-FORTH ROM gives you a normal keyboard with a 64 character buffer and repeat, it supports the 16k, 32k, 64k RAM packs, it is fig-FORTH compatible and it supports the ZX printer.

The price, too, is almost unbelievable. As a "fit it yourself Eprom", complete with manual, it's just £25+VAT.

Add £2 p&p UK (£5 Europe, £10 outside Europe) and send your order to the address below.

Skywave SOFTWARE

David Husband
73 Curzon Road, Bournemouth, BH1 4PW, ENGLAND.
Tel: (0202) 302385.
International +44 202 302385.

ZX PROGRAMMERS. . ™ OCP look no further than

Whether you write MACHINE CODE or BASIC we have the very latest "state of the art" programming tools for you, try them and see why our product is widely regarded by professionals as the very best available.

FULL SCREEN EDITOR/ASSEMBLER (16/48K) voted THE MOST POWERFUL MACHINE CODE PROGRAMMING TOOL YET SEEN by HOME COMPUTER WEEKLY

- Editing facilities comparable to the most sophisticated word processor with MOVE, COPY and/or DELETE lines or blocks of code.
- LOCATE, CHANGE or DELETE specified strings or characters, full Z80 instruction set supported, comprehensive syntax check, powerful expression evaluator, 8 derivatives including stand alone and quash, symbol table display, assembly to screen, printer or save to tape and **"SNAKE"**, a fully notated source code demonstration program.
  (PLUS 80 version for the KEMPSTON CENTRONICS 80 COLUMN PRINTER INTERFACE now available).

MACHINE CODE TEST TOOL (16/48K) tutor and de-bug program, co-resides with the FULL SCREEN EDITOR/ASSEMBLER in 48K to give a COMPLETE MACHINE CODE DEVELOPMENT ENVIRONMENT that is second to none. The programmer can switch between programs within moments.

- Allows easy entry and testing of machine coded instructions.
- Pages and displays memory registers so you actually see what's happening, displays Main and Alternate register sets. Breakpoints can be Set, Viewed and Nullified. HEX:DECIMAL conversion. STOP and return to BASIC. MOVE a memory block. GOTO address. Character Generator and full supporting documentation and tutorial.

MASTER TOOLKIT (16/48K). YOUR BASIC WILL NEVER BE THE SAME AGAIN!

This program adds a whole range of really powerful commands and facilities for your Spectrum.

- Real time clock and alarm with off/on/set and print commands.
- BLOCK MOVE, COPY, DELETE and MERGE two lines. FIND and CHANGE character string. RENUMBER. 10 programmable keys (up to 255 chars. each). TRACE with continuous execution display. VARIABLE display and dump. COMPRESS, REMKILL and PACK to minimise program bytes. CHANGE CASE upper to lower and vice versa. RAMTOP ADDRESS and PRINTER output for vectors. Comprehensive manual and instructions supplied.

Available from selected branches of W H SMITH, BOOTS and MENZIES and other good software stockists.

OCP ™ SOFTWARE . . . SIMPLY THE BEST

If you experience any difficulty obtaining your copy of these programs send a cheque or postal order for £9.95 per program (£19.95 for the PLUS 80 version of FULL SCREEN EDITOR/ASSEMBLER)

Oxford Computer Publishing Ltd.
4 High Street, Chalfont St. Peter, Bucks. SL9 9QB

GRAFPAD

for Sinclair Spectrum ... Drawing into the future!

Now you too can have the benefit of drawing your own applications by the simple use of our NEW ZX Spectrum version of Grafpad!

THE GRAFPAD comes complete with Cassette Programme, Special Keyboard, Overlay, Plug, Light Pen and, of course, a comprehensive manual full of tips and instructions . . . all at the usual high quality expected from **BRITISH MICRO!**

STARTING UP is simple. Connect the Grafpad, switch on your Spectrum and connect the cassette player, type "LOAD" and press ENTER, then start the cassette player, place the keyboard overlay, and by the time you are ready, the "MENU" will flash on to your screen! Usage? Well there is . . .

AS MANY USES AS YOU CAN IMAGINE by first following our easy manual, then, in no time, you will be able to utilise as you Think best!

A UNIQUE PRODUCT that will improve your skills and give you endless hours of excitement!

Size
25mm height
55mm width
260 depth
weight 1.2k (gross)

★ **Dealer inquiries welcomed**

★ **Special discounts to educational authorities & government departments**

★ **Export inquiries invited**

HOW TO ORDER
BY TELEPHONE: (0923) 48222
If you are an American Express, Barclaycard, Diners Club or Access Card Holder simply telephone us giving your Card No., Name, Address and item(s) required and your order will be despatched within 48 hours!

BY POST:
Simply fill in the coupon enclosing your cheque/P.O. made payable to: BRITISH MICRO or use the special section for Credit Card Holders and post to the address below.
OVERSEAS ORDERS:
Please add £10 per item ordered.

BRITISH MICRO

A HEGOTRON GROUP COMPANY

Unit Q2, Penfold Works,
Imperial Way, Watford, HERTS WD2 4YY
TEL: (0923) 48222. TELEX: 946024

Post to: BRITISH MICRO, UNIT Q2, PENFOLD WORKS, IMPERIAL WAY, WATFORD, HERTS. WD2 4YY.

Please send me GRAFPAD for: SINCLAIR SPECTRUM
☐ BBC MODEL B ☐ COMMODORE 64
If you require details of the above versions, please tick.

Qty.	Item	Ex. VAT	Inc. VAT	Total
	Grafpad Complete	£125.00	£143.75	
			Postage, Packing & Insurance	£5.00
			TOTAL	£

I enclose my cheque for / P.O. £............

I prefer to pay with my American Express, Barclaycard, Diners Club, Access Card.
(Please delete whichever is not applicable)

Card No. ....

Signature ....

Name ....

Address ....

....

Address above must be the same as Card Holder

EMS

★ INQUIRE ABOUT BBC VERSION.