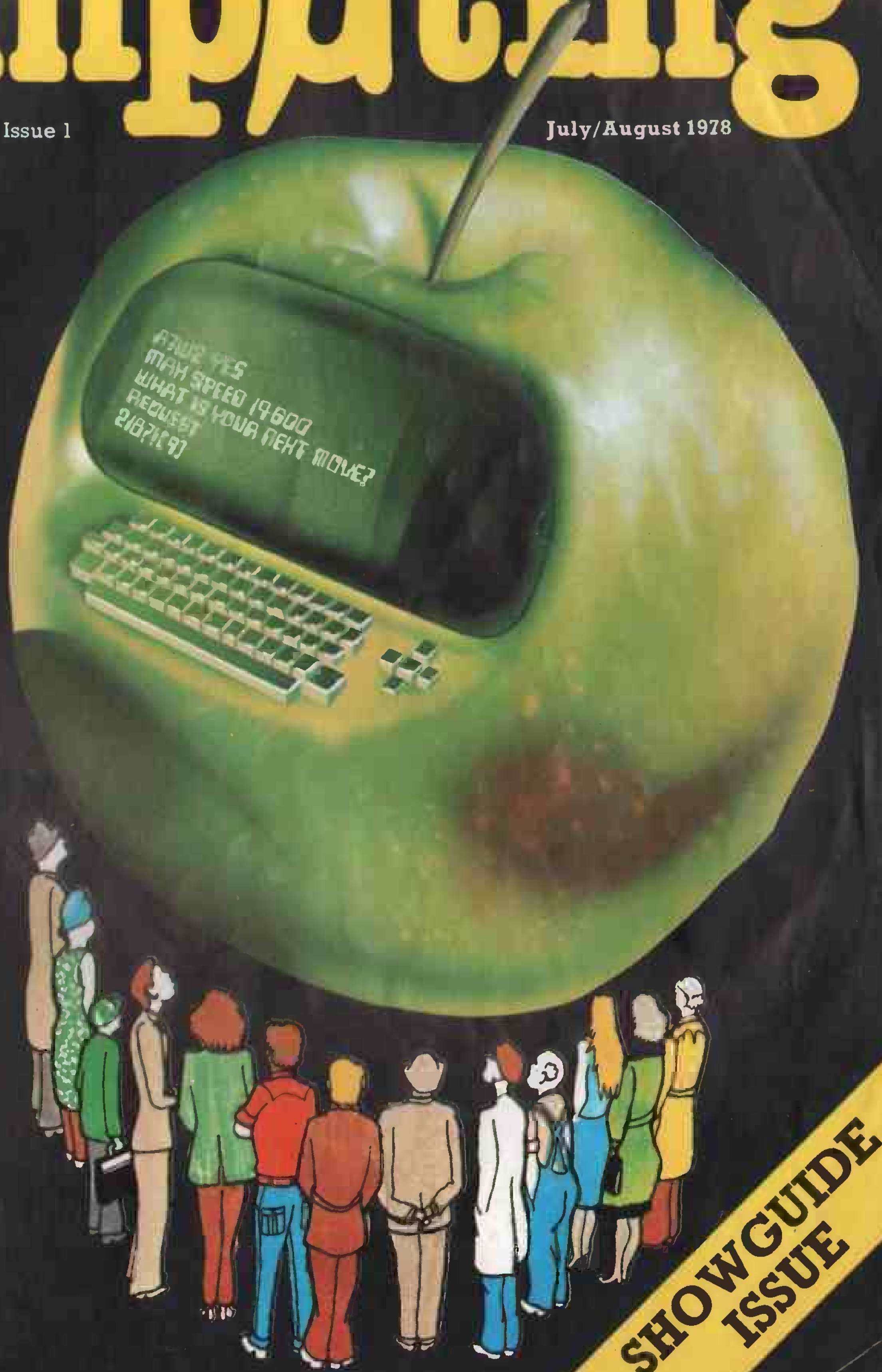
Practical Computing

50p

An ECC Publication. Volume 1 Issue 1

July/August 1978



Win a computer competition

We review Apple II

Business Applications you can run

Musical Micros

How to wire your home

SHOWGUIDE ISSUE

Circle No. 101

Contents

intel®

Packaged Systems | Single Board Computers |

The Answer To Your Microcomputer

SYSTEM 80/10 The System 80/10 is a fully packaged microcomputer utilizing the iSBC 80/10A Single Board Computer. Ideal for the OEM whose design requires low-cost 19" rack mountable packaging, the System 80/10 offers easy to use, fully programmable I/O, the computational power of and fully compatible with Intel's iSBC 80/10A, with both RAM and EPROM memory. The enclosed power supply is designed to support not only the Single Board Computer, but also a full complement of additional slots for expansion boards. The chassis houses the computer, power supply, fans, and has three additional slots for expansion boards.

Real-Time Applications served in the market sectors to date are:

Industrial Control
Automation
Instrumentation
Replacement of Wired Logic
Numerical Control
Distributed Intelligence
Data Acquisition and Logging
Data Communication
Terminal Concentration

These are just a sample of applications served by our Systems Division in both hardware and standard real-time software.

SYSTEM 80/20-4 The System 80/20-4 is a fully packaged microcomputer utilizing the iSBC 80/20-4 Single Board Computer, ideal for the OEM whose design requires low-cost 19" rack mountable packaging. The System 80/20-4 offers easy to use, fully programmable I/O, the computational power to the iSBC 80/20-4, and has both RAM and EPROM memory. The enclosed power supply is designed to support not only the Single Board Computer, but also a full complement of expansion boards. The chassis houses the computer, power supply, fans, and has three additional slots for expansion boards.

Real-Time Software
Application

iSBC 660 SYSTEM CHASSIS The iSBC 660 is an attractive, 7" high system chassis designed for use with Intel OEM computers. It has 8 slots for Single Board Computers, memory, I/O, or other expansion modules. The iSBC 660 is ideal for applications requiring multiple board solutions.

Compatible Single Board Computers

- iSBC 80/05, 80/20, 80/20-4 can operate in multiprocessor environment
- iSBC 016 16K byte RAM
- iSBC 094 4K byte CMOS RAM
- iSBC 104-4K byte RAM, 8K byte PROM, 48 programmable I/O lines, USART
- iSBC 108 8K byte RAM, 8K byte PROM, 48 programmable I/O lines, USART
- iSBC 116 16K byte RAM, 8K byte PROM, 48 programmable I/O lines, USART
- iSBC 310 High Speed Math Unit
- iSBC 416 16K byte PROM
- iSBC 501 DMA Controller
- iSBC 508 32 input lines / 32 output lines
- iSBC 517 Combination I/O Board
- iSBC 519 Programmable Parallel I/O Board
- iSBC 534 Serial Communications Board
- iSBC 556 Optically Isolated I/O Board
- iSBC 711 D/A Input Board
- iSBC 724 A/D Output Board
- iSBC 732 Combination A/D, D/A, I/O Board

Compatible Peripherals

- iSBC 201 Diskette Controller
- iSBC 202 Double Density Diskette Controller
- iSBC 211 Single Diskette System
- iSBC 212 Dual Diskette System

Compatible Hardware

- iSBC 530 Teletypewriter Adapter
- iSBC 956 Parallel I/O Cable Set

Compatible Software

- RMX/80 Real Time Multitasking Executive

intel®

The nuclei of Intel's **OEM** Computer Product Line are the iSBC 80 Single Board Computers and System 80 Packaged Systems. Each Intel Single Board Computer provides all the resources of a full computer (i.e., CPU, Read/Write Memory, Read Only Memory, Parallel I/O, and Serial I/O) on a single PC board. The System 80/10 and 80/20-4 extend these capabilities into low cost, fully packaged, rack-mountable computers. The Intel Single Board Computers and Systems are supported by a complete line of memory, parallel and serial digital I/O, analog I/O expansion boards, and peripheral and DMA Controllers which are all compatible with the standard bus... the Intel **MULTIBUS.** Prototyping packages, modular backplane/cardcages, power supplies, and chassis are also available to simplify product development. Intel's **RMX/80** Real-Time Multi-Tasking Executive with diskette file system, terminal driver, analog driver, high speed math unit driver and debugger have been designed and optimized specifically for the iSBC 80 series. The **OEM** Computer Systems products and Intel's Microcomputer components are supported by the most advanced system development tools available today. The heart of these development tools is the Intellec Microcomputer Development System, its In-Circuit Emulators, and its complement of advanced development software. Software packages available for use directly on the Intellec System include a resident relocatable and linkable macro-assembler, a text editor, operating systems, and utility programs. PL/M, the high-level programming language Intel specifically designed for 8080 based systems, is also available resident on the Intellec system. The PL/M-80 compiler provides the capability to program in a natural, algorithmic language, and eliminates the need to manage register usage or allocate memory. PL/M-80 program modules are relocatable and linkable with other language program modules.

For more information on Cost Effective Microcomputer Systems, contact:
intel® International
Rue du Moulin à Papier, 51 - Boîte 1,
B-1160, Brussels, Belgium.

● Circle No. 102

WELCOME

Dear Reader,

Welcome to the launch issue of *Practical Computing*.

You may have noticed from our cover that this issue is dated July/August. We had intended originally to begin publishing in September, but the organisers of the D–I–Y show asked us if we could publish a Showguide, so we decided to bring forward the launch of *Practical Computing* and incorporate that in it.

From September, though, we shall be publishing regularly every month on the first of the month.

In this issue are some of the ideas we have for the future of *Practical Computing*. There is, of course, the special Showguide to the D–I–Y exhibition and conference, as well as abstracts from many of the papers presented, which we have found extremely interesting and hope you do, too.

The magazine is really aimed at three audiences. For those of you in the hobby market, we shall be carrying regular reviews of computer equipment, looking at shops, prices, software, games and puzzles. We will be telling you how you can get the most from your home computer.

We will also be looking at how micros can be used in business and the opportunities open. We will be examining *practical* applications, showing how businesses can now take advantage of cheaper computing power, as well as ways you can develop systems for companies.

And in education we will be examining how computing is used and taught in schools, colleges and universities with *practical* ideas and suggestions.

Most of all, though, we want to hear from you. In this issue you will find details of our Win-a-Computer competition (page 52). We would also like letters from you telling us about your experiences and problems you may have. We will even try to answer them through our advice column. If you are looking for special software, hardware, or just information, write to us. We will even pay £5 for the best letter published.

Practical Computing will be about *practical* ideas, suggestions, and applications. And it will be fun. That we promise you.

Richard Hease

Richard Hease

Note: Letters should be addressed to: The Editor, Practical Computing, 2 Duncan Terrace, London, N.1.

Practical Computing

Managing Editor
Dennis Jarrett

Production Editor
Harold Mayes

Advertisement Manager
Erica Gibson

Subscription Manager
Annabel Hunt

Publisher
Wim Hoeksma

Managing Director
Richard Hease

Editorial, Advertising and Subscriptions: 01-278 9517
Practical Computing is published by ECC as a subsidiary of WHICH COMPUTER? Ltd at its registered office, 2 Duncan Terrace, London, N1 and printed by Bournehall Press Ltd., Welwyn Garden City.

ISSN 0141-5433
Subscription rates: Single copy: 50p. Subscriptions: U.K., £6 per annum (including postage); overseas, £12 (including airmail postage).

Using SWTP Micro System at Eltham College

Eltham College is a boys' school of some 500 pupils in South-east London. Formerly a direct grant school, it was forced into independence by the conditions of its foundation trust. The school is by no means over-endowed financially, and the presence there of four U.S. South-west Telecommunications Products (SWTP) micro systems is something of a tribute to the determination and commitment of two of its physics teachers, Jan Pringle, and the head of department, Gordon Teichmann.

COMPUTING in schools is still a fringe activity, despite many more or less organised attempts to introduce it into the curriculum, dating from the foundation of the ICL CES project in the 1960s. The cost of equipment has, of course, been a major handicap, with the result that most computers with a useful capacity are to be found in the better-off independent schools.

Even then there are cases where the presence of a computer seems to add more to the prestige of the school than to the education of the pupils, and where inter-departmental rivalry prevents full use being made of the potential.

Both Pringle and Teichmann feel that computing has reached the stage where it needs to be part of any general education, for scientists and non-scientists alike, but for some time that had to remain a pipe dream. They two looked enviously at powerful minis which would have been ideal had they been a quarter of the price, and with interest but pessimism at the early micros.

The relative cheapness of these systems was negated in almost every case by a lack of facilities—limited memory size, inability to attach peripherals, often an almost total lack of software—which made them unsuitable for educational use.

Pringle and Teichmann were looking for a system which would give the pupils hands-on experience, allow them to do

(continued on next page)

(*continued from previous page*)

realistic work, and grow to meet demand. They discounted time-sharing for two reasons. It prevented pupils having any direct contact with the computer; and even if they could have obtained free computer time, the size of the telephone bills for any reasonable amount of usage would have put it out of reach.

An advertisement first drew their attention to the South West Telecommunications micro kit in the Autumn of 1976. It seemed to meet all their requirements, and without more ado they bought one. Or rather, since there was no money immediately available from school funds, Pringle bought one, with his own money, and then set about trying to persuade the school governors that computing was a good idea.

Persuading the Governors

The best way to do it seemed to be to get the computer up and running in time for the next governors' meeting, two weeks later. The kit they had bought consisted of the computer, a keyboard terminal and cassette interface, to which they added a portable TV as visual display and a cheap audio cassette recorder.

The system was assembled and tested by Pringle in a week, and a program written and tested in the second week. It is not an undertaking he recommends to anyone else. "By the end of the fortnight", he says, "I was absolutely ashen".

The soldering times involved were about 12 hours for the computer, 14 hours for the terminal, and six or seven for the tape interface but, as he points out, that gives very little idea of the true scale of the problem.

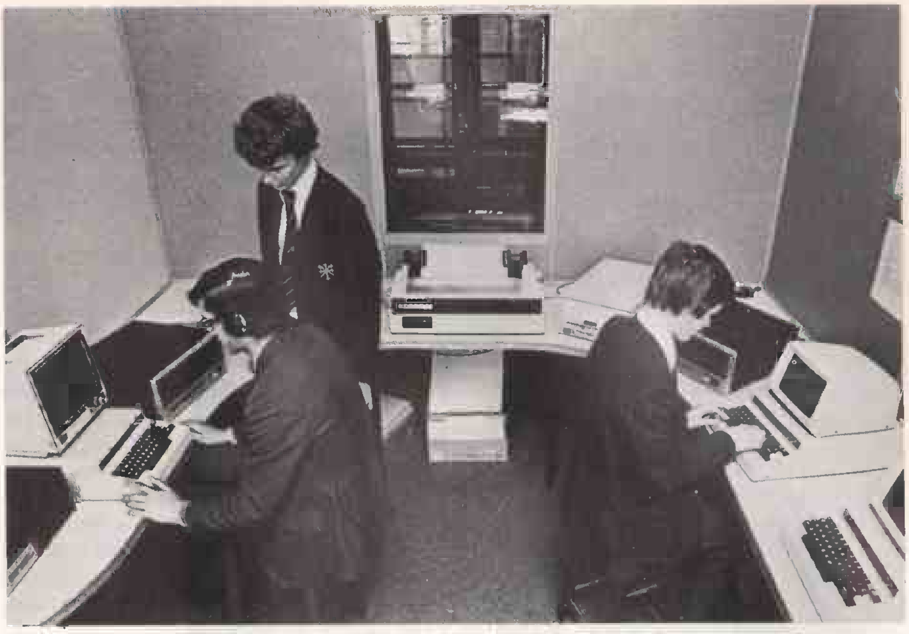
The instructions supplied with the kits are comprehensive and, apart from a few obscurities, comprehensible. Even so, deciphering and putting them into practise is a time-consuming business. "Just sorting out the components is a problem the first time you do it—nowadays I do it while I'm watching television".

Apart from that, much fine work is involved in the soldering, and the delicacy required might present difficulties to anyone accustomed to the standards of normal hobby work.

Success on two counts

The initial system was a success on two counts. The school agreed to meet the costs (about £800), and the computer generated immediate and lasting enthusiasm among the pupils. Within six months, however, the limitations of this admittedly fairly basic set-up were becoming all too apparent.

They concerned mainly the slow operating speed of the cassette recorder, which takes up to five minutes to load or dump a program. Reliability was another problem and audio equipment in general was felt not to be up to the necessary standards of reliability for computing. Thus the next addition was an MSI FD8 floppy disc drive, imported, like the computers, by Computer Workshop.



From that point, the hardware has been added-to piecemeal as money has been available. Even now, it is difficult to give an exact description of the system, since it is being augmented constantly. Of the four SWTP computers, three are equipped with mini-floppy disc drives—the original cassette recorder is still *in situ*. Two computers use TV displays, but two have SWTP monitors, which are felt to be ergonomically preferable. In addition, there are the FD8 disc drive and Centronics 701 and T1 Silent 700 printers.

Software includes a Basic interpreter, file-handling routines, operating system and a time-sharing program, which Pringle is adapting to handle discs.

How the cash was raised

The value of the equipment at the college is now about £10,000, which has been raised in various ways. The first computer was bought by the school, and the Parents' Association provided the money for a computer room, which was built and decorated by the staff and pupils. Money has also been raised by running courses for other schools, and by

(*continued on next page*)

(continued from previous page)

pupils' contributions to the school Computer Society. A large proportion of the equipment, however, is the property of Pringle and Teichmann, on more or less permanent loan to the school.

If that seems an extravagant amount of hardware for teaching purposes, it is put in perspective by the attitude of Pringle and Teichmann to teaching, which is, essentially, that computing is a real-life subject which needs to be presented in a realistic way.

Local business projects

With that in mind, they have arranged for pupils to carry out projects for local businessmen. A stock control program is already written and running; and invoicing, payroll and word processing programs are under development. Another project is the provision of graphic records of patients' teeth for a local dentist.

Nearly all pupils now have an introduction to computing in their first year. As they grow older and begin to specialise, the proportion naturally drops, but even so, 40–50 percent of the fifth and sixth year are studying or using computing in one way or another.

Computing is not taught as a classroom subject. "In fact", says Pringle, "teaching is hardly the right word for it at all", He puts forward the mildly heretical view that programming is very easy to pick up and that once the pupil has been shown how to cope with the machine, he can be left to "explore", more or less unsupervised, for some time. The structure of programming can be explained later and is easier to interpret as a result of the hands-on experience already gained.

So pupils are taken in groups of two or three, first by a teacher and then by an older pupil, and begin to write programs almost immediately. Some credit must be given to the simplicity of Basic, but Pringle stresses the advantages of micros in this respect—their compactness and even crudeness compared to larger computers is something he sees as an asset.

"Simple things like connecting peripherals and loading programs are a great help in making computers more real and understandable. It's something you simply can't get from any form of remote computing."

The fact that computing is under the aegis of the Computer Society rather than part of the official curriculum allows great flexibility of approach to the subject, and programming is certainly not regarded as the be-all-and-end-all. Some boys take an interest in electronics, others use the computer in connection with other subjects, such as geography, economics, and, of course, physics.

The academic side is not neglected. The school entered its first candidate for A level Computer Science last year—successfully—and the first real crop of seven candidates is now sitting the examination. Pringle and Teichmann have not entered candidates for O level, which they feel is hardly a worthwhile examination.

One problem which has arisen in the teaching of A level is the lack of an adequate textbook, due largely, no doubt, to the relatively small number of candidates taking the examination at present. Some Open University publications have provided a partial solution but there is clearly a gap which will become significant in a few years' time.

Approach with caution

Both Pringle and Teichmann are convinced that they made the correct initial decision and are very happy with the results. They continue to look at new computers to see if they are missing anything, but feel that, while other systems are potentially better, they are limited by a shortage of peripherals or software.

At the same time, they offer several words of caution to anyone considering the same approach. The delicacy of the soldering already mentioned is the first possible stumbling block, and anyone unused to such work might do better to

(continued on next page)

(continued from previous page)

get the system built for them.

"It's important in any case", says Pringle, "to have someone to run to. Maintenance is a real problem, since it's almost impossible to find the duff bits on a machine without another one on which to test them". So a standby machine, your own or someone else's, is almost essential.

"The manufacturers are very helpful, but they just can't provide an on-site repair service, which means that if you want them to do the job you have to pack up the whole kit and send it back. This could be a big advantage for the Tandy systems, with a potential repair shop in nearly every High Street."

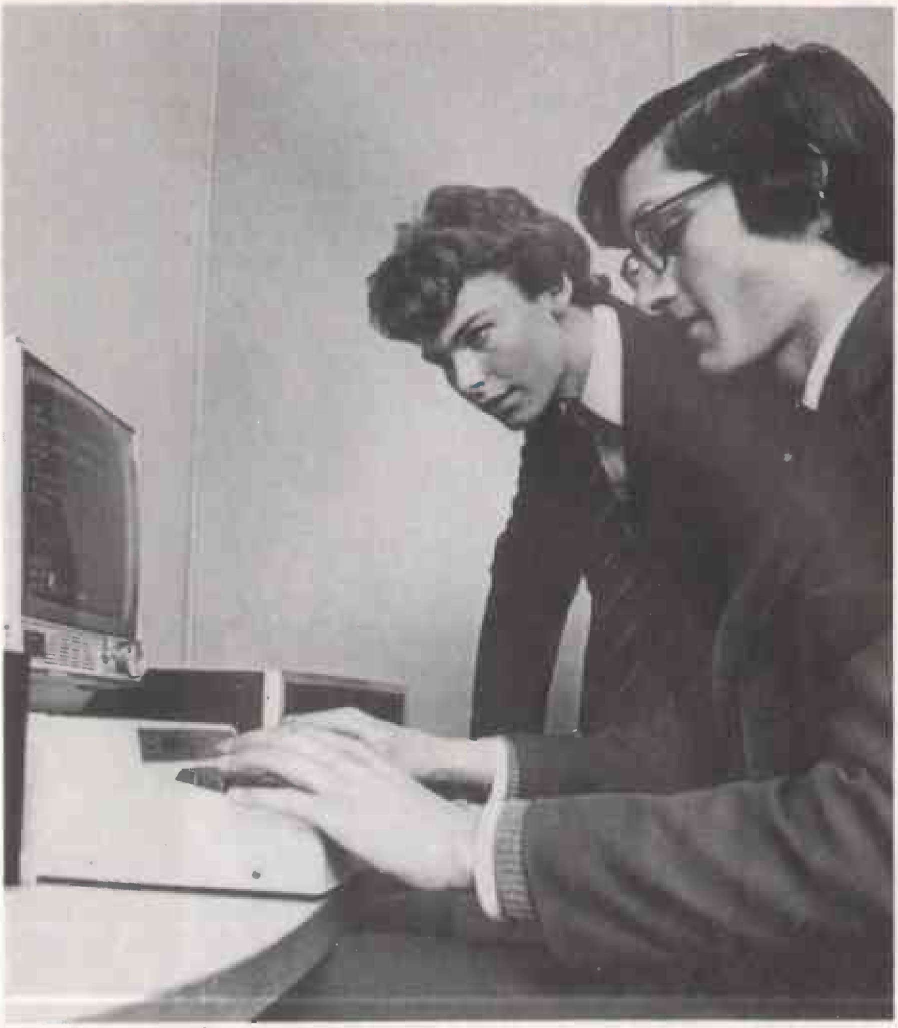
Finally, Pringle points out that "home computing" is bound to be expensive if you want worthwhile results. A limited price invariably means a limited system. He sees little prospect of a major change in this situation, for while the cost of electronic components may fall, it will be matched largely by a rise in the cost of mechanical parts in peripherals.

Considerable potential

There is a brighter side to the coin, though. For those who can afford such a system the potential is considerable. The excellent design of Basic means that other languages are hardly necessary, "though Fortran would be nice, and running costs are remarkably low.

Repair charges are at present "fairly nominal", though this is a condition unlikely to last if sales take off in line with predictions. Software is very reasonably priced, with Basic interpreters available from as little as £10.

The Eltham College installation is, of course, much bigger than most individual users, or even small businesses, would be likely to contemplate, but it illustrates several points about the current status of micro-based systems. First, remarkable results are possible from relatively small systems—Pringle and Teichmann consider 20K of memory to be the minimum useful size—provided they are equipped with adequate peripherals and software.



Second, basic system prices are a poor guide to the costs of a working system. The four Eltham basic kits, at £800 to £1,000 each, account for little more than one-third of the total cost. By far the largest part of the remainder has been spent on peripherals, with relatively little going towards software.

Third, users of this type of system are at present very much at the pioneering stage. Suppliers used to be vetted with care, and even the best can provide only limited support. While the situation is likely to change rapidly over the next few years, a sound knowledge of electronics is at present the home computer user's most valuable asset.

One Apple which grew without giving anyone the pip

EACH MONTH Practical Computing will be reviewing a microcomputer system. This month we look at Apple II, the prize we will be giving away in our competition (see page 49).

Apple II is now being sold extensively through a network of shops and distributors in Britain. It has even been adopted by the consumer products division of the international giant ITT, as an obvious adjunct to its colour TV sets.

In this review we look at Apple, the company, the product, the software, the kind of applications you can run, and, of course, the price you have to pay.

APPLE was formed in 1976 by two hobbyists working in a garage in California. Steve Jobs is one of the new American computercrats. He grew up with minicomputers at his school, he can visualise several lines of assembly code complete with hex equivalents and displacements, and he can make that program work by typing straight into a computer without writing down the code first. Steve Wozniak, who shared the garage, is an engineer; he had spent three years with Hewlett-Packard, an engineer-driven company reminiscent in some respects to our own Ferranti.

About that time there were hundreds of lads working in California garages. Jobs and Wozniak were different, however, not least because they succeeded in attracting a man named Markkula from Intel. Markkula had had four years there as marketing manager. Intel, like Fairchild, breeds capitalists, who get as part of their remuneration more shares they can properly dispose of. The result was that the new company had access to personally-controlled capital. Markkula also had good banking contacts—with Bank of America.

Market research exercise

The result was that the first Apple, basically a kit of parts for knowledgable hobbyists, was quickly converted into a market research exercise for a properly-funded, second-generation machine.

With a new product specified by March last year, the company had the cash needed to develop it as a fully-packageable product. There was no question of sticking badges on other people's boxes.

By January this year Apple could move into a proper factory, having sold around $3 million worth of the Apple II —formally announced on June 5, 1977. Sales of this machine in the U.S. are expected to reach five times that figure before the end of this year.

The U.K. Apple II is, thankfully, not the same product as sold in the States. A key factor in its appeal is the ability to drive a colour television. American television differs radically from the PAL colour television system used in Britain (and Germany) so a new colour driver circuit had to be designed for sale here. This design project was not exactly rushed, with the result that the Apple II computer is not yet truly six months old here.

It was launched officially at the Compec exhibition last November by the agent, Personal Computers Ltd, under Mike Sterland. There it was demonstrated using a U.S. standard television.

Apple II is based on the MOS Technology 6502 microprocessor, now made by Commodore—who bought MOS Technology—plus Synertek and Rockwell. This micro has many fans because of its relative ease of use, compared to Intel 8080-type micros. It is based on Motorola 6800 design, with a few modifications which are regarded by everyone, except possibly Motorola, as improvements.

Fast Basic interpreter

One effect is that the first feature which the user notices about Apple II is its very fast Basic interpreter. To be sure, a few tricks were used in making Apple Basic run that fast; it assumes that users would know no other test than a simple FOR I= 1 TO 1000, NEXT I, PRINT "END", END benchmark and would be impressed if the loop came out around 10 seconds—and it does.

The hardware consists of a complete boxed system including microprocessor, keyboard, power supply, high-level language interpreter (in firmware), and at least 4K bytes of read-and-write memory (RAM). You also get two non data processing 'peripherals'. They are two game paddles, which give the unfortunate impression that the Apple is a games machine.

To use Apple II, a standard colour television, a standard audio cassette recorder, and a standard 240V power point are sufficient. This means that the user need not open the smart moulded plastic case and plug in anything or do any soldering, or even adjust a single component unless the plug-in cable to the tape cassette machine does not match.

Expansion initially involves adding read-and-write memory. The machine uses the very dense memory chips now available from semiconductor companies, and this enhancement involves plugging more chips on to the main central processor board rather than adding memory boards.

This is important because it means that expanding the memory of the system does not mean you take up more PCB slots and thereby limit the number of peripheral boards you can add.

The other aspect of expanding memory is merely a question of writing the cheque. A comparison of the price for the machine in differing memory capacity up to its maximum 48K bytes shows that the memory costs approximately £55 for 4K of RAM—definitely good value compared to the cost of adding a memory board of similar capacity to other systems, while at the same time having a great deal of margin for manoeuvre as technology brings down prices.

The cost of buying the extra chips is not the same, however, as the cost of buying a system with more memory in the first place. An additional 4K bytes bought as chips will cost £100. The moral is simple—decide how big you want the machine to be eventually, and obtain one that size now.

The cassette tape is used as a program dump and load device, because the Apple semiconductor memory will not retain data with the power off. The system has the advantage of a fast data rate, at 1,500 baud, which makes the loading of your favourite program much quicker than on personal systems using the standard Kansas City CUTS system. This means, of course, that programs written by personal users who have CUTS will not load on to Apple.

Plug-in printer option

A diskette will be available at the D-I-Y Show at £395.

On the printer front, Apple II has a printer interface card which, says the manufacturer, will drive "almost any printer". One such is the Centronics 779, for those who want a commercial printer) but this is priced at £700, plus £100 for the printer card.

An amazing little printer which prints at 2,200 characters per second is, we understand, to be offered as a plug-in

(continued on next page)

(*continued from previous page*)

option. We are still waiting, however, to know the price.

Software

Firmware comprises 6 bytes of read-only BASIC interpreter, and 2K bytes of read-only monitor, including a mini assembler and disassembler. You also get Wozniak's brilliant pseudo 16-bit micro, Sweet 16.

The BASIC is integer BASIC and therefore fast, but a bit simple. A vastly improved Applesoft BASIC, described as "expanded, floating-point BASIC" (but in fact quite a bit more sophisticated than that implies) is available on cassette; this is for loading into read-and-write store (you need a lot) but it will "soon" be available in pluggable firmware (price is not known).

Reputation as games machine

Features of the firmware Basic include multiple statements on single lines and string arrays to 255 characters. Special commands are provided for COLOR and PLOT (self-explanatory) and HLINE (horizontal line) VLINE (go on, have a guess), SCRN (reads the screen colour at x, y) and PDL (controls the games paddle).

There are also the odd utilities such as cassette SAVE and LOAD. There are switchable I/O assignments, there are immediate execution features—on "most statements"—and there is a memory boundary adjust feature which means you do not lose the program when you would expect to do so.

The monitor, which needs to be taken with the documentation provided, gives the user control of character formation, cursor control, and other detailed graphics and screen features. It can be a little confusing here because the colours are designated by number but the numbers generate different colours on PAL sets, and you have to disbelieve everything you don't see on the screen. But we have been told you soon get used to that.

Application software is what has given Apple its reputation as a games machine. Some of them are available on two demonstration cassettes, some are in the excellent reference manuals, and some are for sale at costs ranging from £4 to £20—the latter for the Applesoft extended Basic.

The games are fun, however, and include Mastermind, Hangman, and Startrek; you also get some fun oddities like one in which a dragon chases you around an invisible maze, and if he gets close enough to you he cheats by climbing over the walls. You get eaten—no physical pain, but psychological angst is enough to make you reach for RESET.

"Many users will wish to take advantage of the maintenance plan," says. Personal Computers Ltd. "This is available at 7½ percent of purchase price per annum—so £1,000 worth of equipment costs £75 per year to maintain." This figure will seem daunting to many who will happily spend several times as much maintaining a car costing half the price.

Computer:	4K bytes	£995
	8K bytes	£1,050
	16K bytes	£1,250
	48K bytes	£1,900
Memory:	4K in 4K RAM chips	£100
	16K in 4K RAM chips	£400
	16K in 16K RAM chips	£450
Accessories:	RS232 card	£95
	voice recognition card	£165
	empty prototyping card	£18

Software prices are indicated in text.

Conclusions

The American market research company, Creative Strategies International, has predicted that "with the Apple hardware becoming increasingly popular, the firm may be one of the few which can compete with large consumer electronics companies in the future home market"—a market which the research company foresees as growing really fast after 1982, when home programmers are saturated and genuine domestic users start buying.

If you believe CSI, this means that Apple is one of the few companies the future of which can be anything like guaranteed; with Apple II you can, theoretically, look forward to grandchildren who will run the home programs you're writing now without modification.

Distributors and shops include the following:

Personal Computers Ltd.,
18/19 Fish Street Hill, London EC3R 6BY.
Tel: 623 1434.

Pademede Ltd.,
The Tuns, High Street,
Odiham, near Basingstoke.
Tel: Odiham (02–5671) 2434.

Keen Computers Ltd.,
58 Castle Boulevard, Nottingham.
Tel: 0602 412214.

Eurocalc,
Tottenham Court Road,
London, W1.

The Byte Shop, 426/428 Cranbrook Road,
Gants Hill, Ilford, Essex.

ITT Consumer Products,
Chester Hall Lane,
Basildon, Essex.
Tel: Basildon 3040.

Micro Digital Ltd.,
47 Paradise Street, Liverpool.
Tel: 051-708 8624.

Ready-to-run software is coming here fast

A payroll suite for £12; a mortgage package for £7; a life expectancy package £5. If you think these prices are crazy, read on.

PACKAGED, ready-to-run software for home computers is now coming on to the market here fast—and it's cheap, as one company in Fulham is proving.

The firm is PETSoft. It is a recently established division of the StereoSound Co, which claims to have introduced car stereo and, lately, videotape recorders to the U.K. consumer market.

As you might expect from the name of the company, the software it has developed is for the PET computer. It believes that the system is one of the most likely home computers to make an impact on the U.K. consumer market—the non-professional market.

That is not to say, though, that PETSoft is narrowing its sights to the PET computer only. At the moment it is, we understand, looking seriously at Tandy TRS80 with a view to producing and marketing software for both machines.

Freelance work for sale

PETSoft began by producing material generated in-house but is now selling software produced by freelances. It also wants PET users to submit their programs for evaluation. Those which prove worthwhile will be published, with a royalty paid to the "inventor" on sales.

All the programs are produced on Commodore format cassettes—they, incidentally, are not compatible with the CUTS standard from Kansas City, nor with the Tandy TRS80 format.

So far, PETSoft has developed an impressive range of low-cost business application software. The company believes that with hardware prices so low on microcomputers, there is little sense in a repeat of the cycle which occurred with minicomputers, where as much investment was required on software as hardware.

The firm believes that it can sell a high volume if the price is right—and some of its prices are really astounding.

It has a payroll suite for £12. It can accommodate up to 100 named employees, plus routines for bonus calculations, overtime and deductions.

Under development and to be released soon is a VAT package which is being inspected by the Department of Customs and Excise. The company says that a French TVA system is available from some other source on the market but that it supports only one rate which in the U.K. is not suitable.

A cashbook and an invoicing suite is in the debugging stages and will be available once the PET printer has been released to the market.

All software supplied by PETSoft has been copyrighted to protect the income on royalties accruing to freelances. The software itself has built-in safeguards and identifiers to deter attempts at infringements.

Although the majority of microcomputer vendors use Basic, many of them are syntactically different. PETSoft claims that PET Basic is the best, as it is a subset of MicroSoft Inc Basic.

What the PET Basic does not have is an automatic re-numbering facility,—where lines are inserted in between others, say 14 & 15 is inserted between lines 10 & 20. The re-numbering facility will convert lines 14 & 15 to lines 20 & 30 and line 20 onwards will be converted to line 40 onwards.

Mortgage package available

That facility is available from PETSoft for £35. An absolute code re-numbering routine in binary is available for £25. A selectable one- or two-pass assembler with full text editor will cost £49.

For the non-business professional user, a mortgage package is available for £7, comprising five programs which may be selected once loaded into memory. The screen flashes a number of questions such as the term of the mortgage, the capital sum, and the like, and the programs calculate interest rates or repayments according to variables which have been input.

Another home user program is a life expectancy package, costing £5, which produces a statistically-accurate assessment from a 40-question form, flashed interactively on the screen, relating to an individual's personal health, life-style and heredity.

Some esoteric mathematical programs are also offered:

- Prime factor program £4
- Cube & Quad (roots, quadratics) £7
- 2 Sim/3 Sim—A two-matrix method of evaluating 2 or 3 roots of any given simultaneous equation £6
- ComPerm/Super N—combinations and permutations, and factorials £6
- 2*InvMat—Inverse of 2 × 2 matrix £4

A disassembler is also available at £15 and a machine code handler allows for the input of hex codes from a given location to call subroutines using SYS verb, at a cost of £3.

Computing for the kitchen

Coming soon is a suite of tutorial programs for the instruction of PET Basic, occupying a total of 27K bytes of memory loaded in 7K byte stages. It will provide explanations for instructions, such as PEEK and POKE.

PETSoft has noted that some computer professionals-turned-hobbyists are usually more interested in games than in programming and in addition to the games available from Commodore, has added two of its own—Doctor Sinister's Personality Test for £9, and Alien Attack—not the same as Star Trek—£7·50.

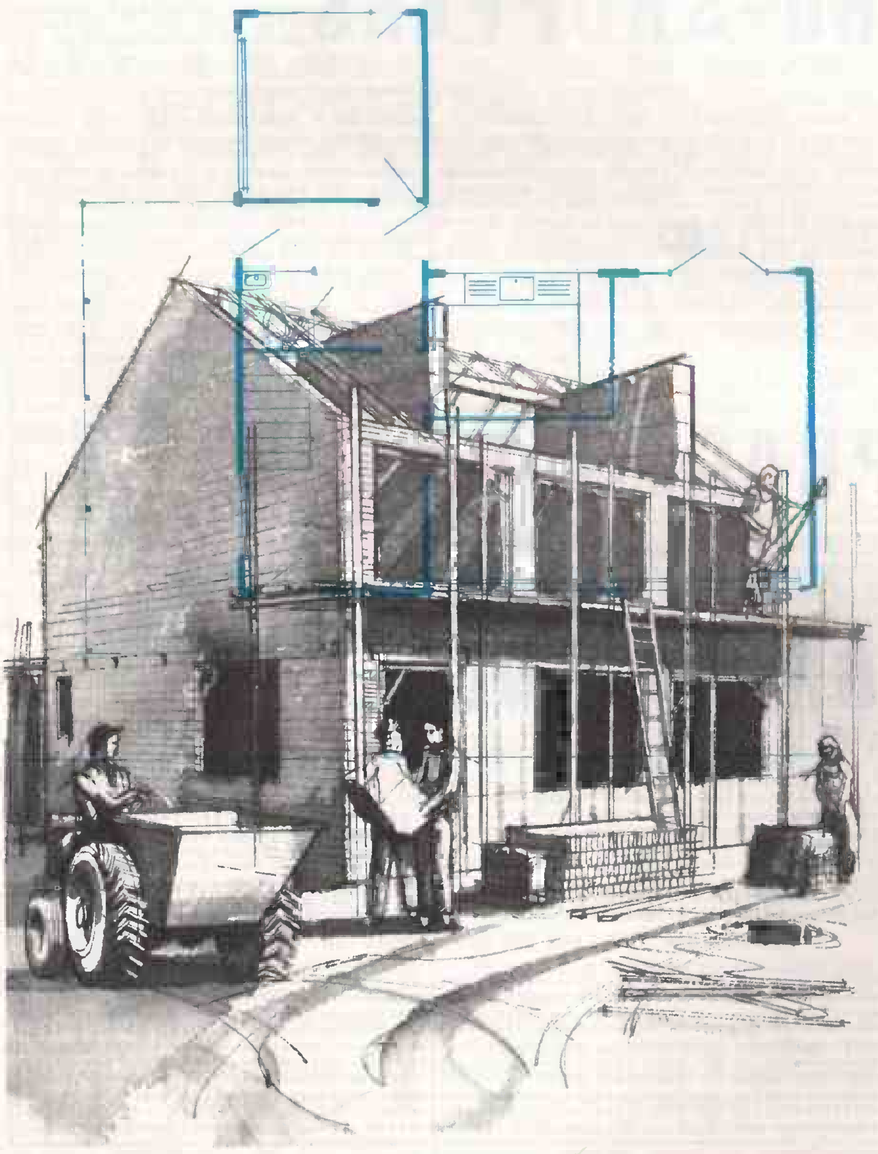
To entice the housewife to the wonders of home computing, interactive recipes will be available and so far the company has a suite of chicken recipes for sale, one of which struck us as particularly interesting—Silicon Valley Casserole—Tender bytes of chicken in a creamy sauce.

PETSoft is at 318 Fulham Road, SW6.

● Circle No. 105

How to build big the little way

Microcomputers are creating literally hundreds of new business application areas. Their low cost is bringing down the price of computer power so that you don't have to be a big company to take advantage. This article explains how small firms in the construction industry can compete on equal terms.



THERE is a very real need for the use of computers in the construction industry to improve the flow of data and to handle the increasing complexity of calculations and regulations even apart from the more exotic areas of computer-aided design.

Much work has been done in this field over a number of years. For example, two of the authors designed and developed a very large integrated system for West Sussex County Council in the early 1960s which covered many aspects of the design, production and construction data for buildings.

So much was excluded

All the work was done on large mainframe machines—IBM 360 and 370 Series with a light pen input/output device (IBM.2250).

The cost of the hardware and the software for such a system meant that this work was developed only in large organisations with large resources. As the construction industry is largely made up of small professional firms and small contractors, the major part of the industry was excluded from most of those developments.

Not only that, but some of the large organisations which developed large computer systems suffered the alienation of the user from the computer. Because the computer and its programming staff were often remote from the user and outside his control, time and resources were allocated only according to someone else's behest.

Even the best systems were often dogged by the "us and them" syndrome. Further, some people became worried about the power they were handling over to departments outside their control.

While the team recognised those problems at a very early stage, there was nothing to be done about it until the cost of computing dropped dramatically.

Initiative and enthusiasm

The microcomputer created this opportunity. One member of the team is a microcomputer hobbyist and it was his initiative and enthusiasm which helped to put together the first configuration working with a conventional colour television set. The original equipment and its cost was:

- Microprocessor with 12K memory £514
- keyboard £248
- cassette tape interface £70
- two cassette tape recorders £30
- colour television set.

With that equipment it became obvious that a great deal of the work could be done on the microcomputer if the systems were scaled down and the configuration enhanced.

(continued on next page)

(continued from previous page)

The original equipment was, therefore, enlarged by the addition of

- four direction cursor controls with joystick
- graphical symbol display
- increased display capability
- colour board
- floppy discs
- an extra 8K memory.

At that stage the biggest problem in transferring a system for a large IBM computer to the microcomputer was to stop thinking complex systems. The tendency with large machines is to design so that the maximum amount of work can be processed in the minimum time.

With a system such as the microcomputer we can afford to process work a small step at a time. The only problem with this is that without the overall control normal in a complex program one's steps may be in the wrong direction, therefore the philosophy of programming design has been to keep it simple.

In that way the user has been provided with a reasonably fast response to a series of mundane processing tasks. This means all the programs are written as simple, straightforward processes, often with data input in a conversational manner.

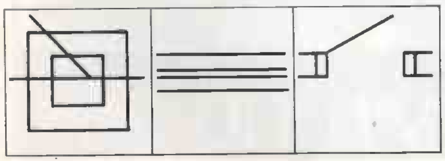
There has been no attempt to double-up the functions of programs the user can initiate another slightly different program for a slightly different task—for example, file creation versus file printing. This approach is thought to be valid because the decision-making of a user who knows the system is bound to be more effective and accurate than that of a program, however complex.

Separate entities

Design and construction in the construction industry are separate, as opposed to most other industries where they are part of one organisation, and, therefore, a convention of communication has to be established. The language of communication consists of:

Symbols which are put together as drawings to indicate the location and type of work to be performed. The drawings are not visual representations of the work to be performed but an amalgam of established symbols. For example, the following symbols are known to everyone in the industry although they do not look strictly like the object.

Manhole *External wall* *Door*



Unfortunately, the symbols leave a great deal open to misinterpretation and therefore need further definition.

Considerable repetition

That additional definition and specification is given in alphanumeric terms. The symbols and descriptions are recorded every time as though the situation had never been seen before, whereas in fact for most architects and contractors, there is a considerable repetition of many items.

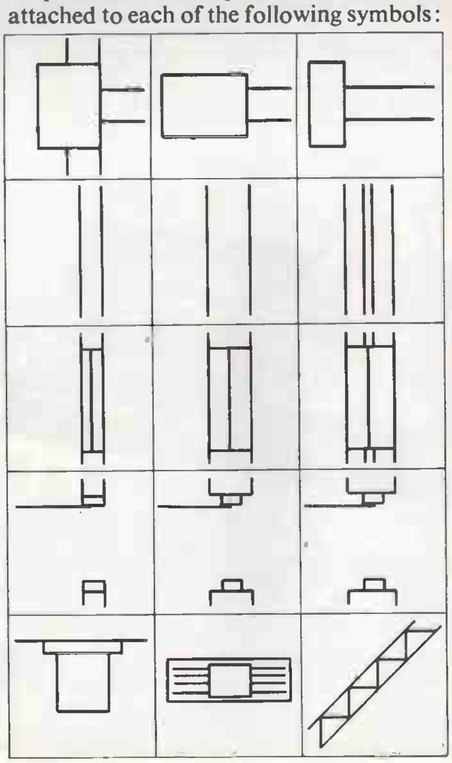
The rules for manholes are laid down clearly and there are few needs for variation. Similarly with brick cavity walls and so on. A personal range is often used while someone else may have his own different personal range.

Therefore if one prepares one's own set of, say cavity wall types, this can be related to the symbol and given a suffix of A, B or C type or 1, 2 or 3 type. The symbol for cavity wall can be called up on the TV screen and, by means of the joystick control, a plan produced showing where the walls are required. The type of wall can be entered by

- typing in description;
- typing in suffix
- using colour as a selector
- using speech input.

Volumes from heights

If heights are given as the plan is developed, the computer will be able to calculate volume. Therefore it is possible to build up data files which can be manipulated to provide a wide range of outputs. For example, values can be attached to each of the following symbols:



The following values are an example:

- a quantity surveyor's normal unit
- quantity
- the material content
- the manpower content
- the activity content;
- the plant content
- heat loss characteristics

Even taking those few simple items of data, it can be seen easily that simple algorithms can produce a bill of quantities for the 'drawing' on the TV screen; the material, manpower and plant requirements and activity content; critical path networks with outputs under those headings; the total heat loss of the building and from that the quantity and cost of energy input.

They are only some examples of possible usage. It is, of course, possible for a contractor to update his files as often as he wishes, to give himself an up-to-date cost. For example, if each symbol is related to material, manpower, activity and plant files, when any item within those files is updated the computer will update the value for each symbol. Only profit has to be added.

If files are kept on manpower—usually for PAYE purposes in any case—then they will be linked to the symbol description. Therefore, if bricklayers' wages increase, the computer will look for bricklayers throughout the file and update these, similarly with materials.

In that way the up-to-date cost of any building can be produced as it is drawn, to which profit is then added. It is obvious that a variety of designs can be tried and the comparisons between capital cost and running cost carried out quickly. Similarly with other resources.

Two problem areas

The building system has two main problem areas as far as data is concerned:

The size of the database is beyond the capacity of the twin floppy discs. Thus it is envisaged that the user will take the approach of separating information into smaller groups and performing multiple passes against different diskettes. That has the advantage that programs which break down infills into component materials, activity, manpower and plant are run in several passes, giving shorter run times and greater opportunity for control.

Some of the information used in the building industry is very volatile (rates) as opposed to other information which is fairly static (activity times, quantities). To speed processing, this volatile information has been separated from the files of non-volatile information. This means extra processing and easier file-handling in the long run.

The use of speech on this system was developed to ease communication.

(continued on next page)

(continued from previous page)

With the drawing joystick and keyboard, he has too much to do with his hands. Thus, a vocabulary has been developed so that he can further describe his symbols (the suffix) with speech.

The vocabulary has been paged. The first page contains:

- colour specification
- command words for picture
- manipulation
- page selection for spelling mode
- room description entry.

The analysis and correlation program is written in machine language as an interrupt routine.

Basic the language

The building system is intended to be used in architects', quantity surveyors' and contractors' offices. The computer system demonstrated would need a printer as well as the keyboard and TV display. No office could process Bill of Quantity information without having a hard-copy facility.

It is only budget limitations which have prevented the addition of one to this system. It is felt that a matrix printer is the most suitable type, because the dot matrix characters could then be used in the same way as on the screen to produce hard copy of plans.

Basic is the computing language of the system and has been found to be very easy to write and debug. Only in two cases has machine level programming been used—speech input and screen reading. In both those cases, speed is of the essence and so the Basic interpreter is not satisfactory.

The file handling of the disc operating system was also found to be rather limited. In a commercial situation it is also very necessary to have a rounding function. This could usefully be incorporated in the SWTPC Basic.

It is possible that this type of work may have some significance for the future. The public in the past has found little opportunity for involvement in the design of its own environment because the design/evaluation process takes so long that a commitment has to be made at initial sketch stage. Also enormous resources have been expended by the time a tender and other evaluations are available.

The technique described can be developed to give direct user interaction with his design. Further, with the advent of Teletex and Viewdata, it may be that details of materials and prices could be produced at a national level on the TV screen while using the microcomputer for manipulating the data at a personal level.

This could mean that the small firms of professionals and contractors would be able to compete with the large organisations, having mainframe systems; those larger organisations could break down the size of their units; the ordinary man may be able to become involved in his environment, if not to design some of it for himself.

The Microcomputer in the Construction Industry, by John Paterson, former deputy county architect, West Sussex, now principal of a private practice; Joanna Firth, former systems engineer with IBM and now a researcher at Reading University; and Ted Cogswell, architect in private practice and lecturer at Southampton College of Higher Education.

Hello, Houston —we are falling out of orbit

COMPUTER games can give excellent hands-on experience. They are usually the foundation on which much more complex mathematical models and techniques are built.

Chess programs are a well-known example. There are many general techniques in problem-solving which have resulted from the analysis of the chess game.

There is an entirely different type of game which can be played, however, and that is the simulation game. This technique is used in engineering or economic models and, of course, simulators. A prime example is the Lunar Landing Game.

Lunar landing games are at their best when they can be played in real-time. The idea is that the player must soft-land a lunar excursion module (LEM) on to the moon's surface. He can brake his fall by firing his main rocket motor and vary the thrust by altering the fuel-flow rate.

In real-time the player has a VDU screen displaying his height, rate of descent and other useful information which is changing constantly as time ticks by. If he does nothing to prevent it, his height will decrease and his rate of descent increase until he hits the ground, instigating the odd moonquake in the process.

So how does he stop it happening? He can cause the motor to fire by pressing a button on the keyboard. The computer program will monitor constantly for this event and, if no key is depressed, will continue calculating the height in the normal way. If the key is depressed, then the program will register that the rockets are on and make the necessary changes to the readout on the screen.

It's just like NASA

Different buttons can mean different fuel flowrates. The simulators used by NASA do exactly the same, except that they monitor the controls and display on screens in a life-sized model.

I have always found that this program can have an amazing effect on people. They really "live" the whole experience.

The equations which do the work are really simple. Consider the case where the LEM is free-falling to the surface. The velocity and acceleration are related by

CHANGE IN VELOCITY = ACCELERATION × CHANGE IN TIME

That is really only an approximation but it will suffice if the time increments are kept small enough. The acceleration is, of course, one-sixth of a G and we may choose to do this calculation every $\frac{1}{10}$ of a second and display the results.

Now what happens when you have a rocket to alter the thrust and hence the acceleration?

The equation will remain the same but the acceleration will change. There is a standard relationship between force and acceleration.

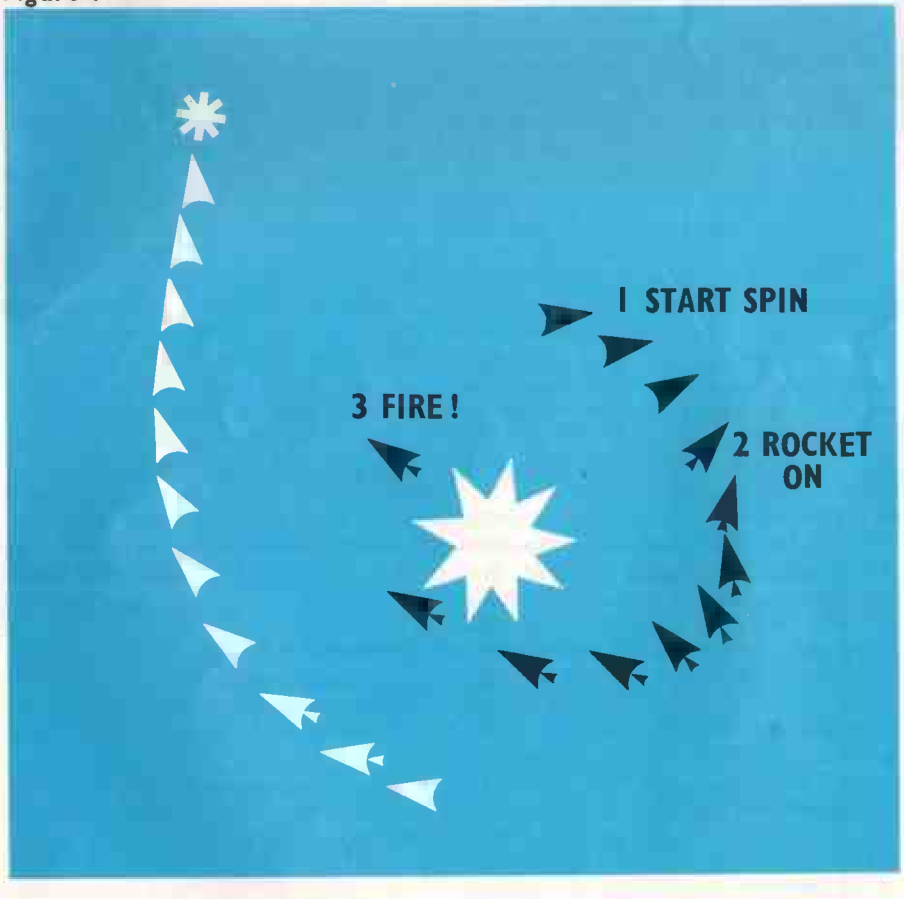
FORCE = MASS × ACCELERATION

And hence in our case

$$\text{ACCELERATION} = \frac{\text{MASS OF THE SHIP}}{\text{FORCE OF GRAVITY + ROCKET}}$$

We know the mass of the ship and the rocket thrust can be proportional to the fuel rate being burned. There is only one last thing to do and that is when the rocket is on, a reduction must be made of the mass to account for the loss of the fuel. That is all there is to a simulation—but there is a good deal more to a game.

Figure 1



Juggling with constants

Firstly, you have to juggle the constants to make it difficult to get down without running out of fuel. When you eventually land then, the program must display some congratulatory remark—or an obituary—on the screen to entice you to improve.

If you are using a personal computer, you could link the throttle input to a TV-type game controller. After a time it seems relatively easy to make soft landings every time. Then the game must be livened up.

I was lucky enough to have a graph plotter linked to a Wang 2200 computer. By the time the final version was completed, our simulation LEM took two people to fly—like the real thing.

A lunar map is drawn on the plotter and the object is to land in one of the craters. Players have control over the tilt of the craft as well as x and y movement. The game begins with the lunar module coming out of orbit "on its side"

(continued on page 21)

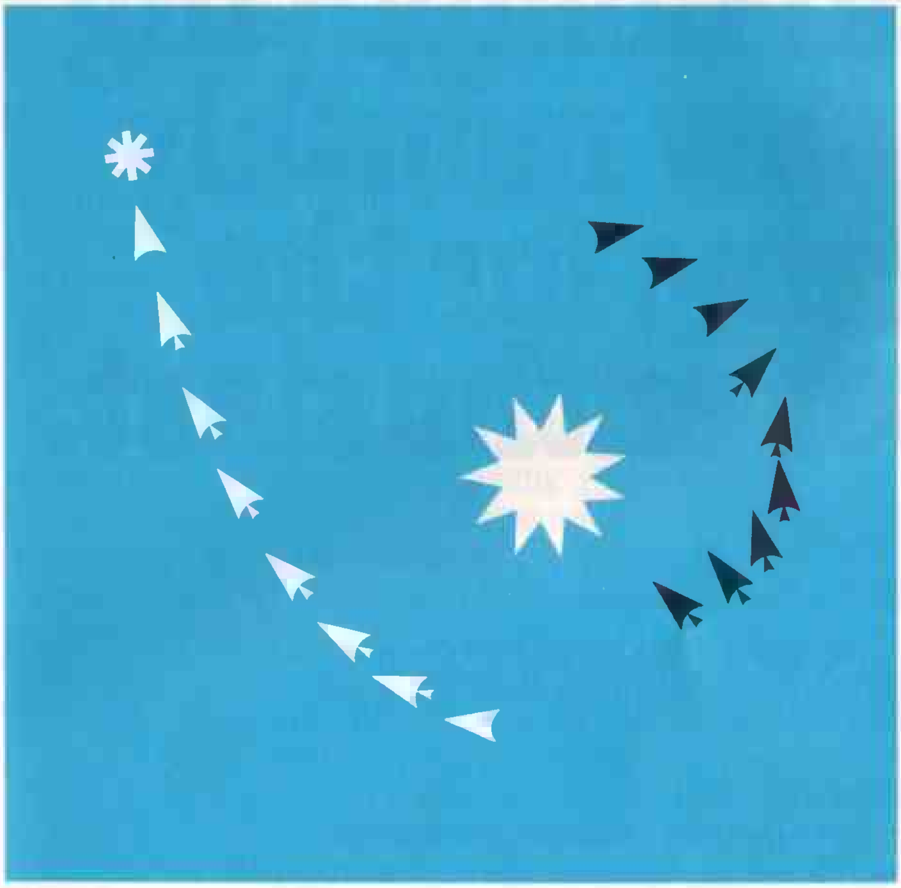


Figure 2

(continued from page 19)

and it is necessary to manoeuvre the craft upright. If control is lost at that stage, you could tumble all the way to the ground.

The next task is to judge the start of the main "burn". If this is too early you waste precious fuel; if you are too late then you never pull up.

There is a constant stream of messages being displayed by Houston command control. They inform of suspected electrical problems and height corrections due to Mascons. At any time all your controls could go off and you may suffer possible ignition failures trying to start up again. When the surface is finally neared Houston will tell you if the area is suitable for landing.

For this information you must be under 100 metres altitude. If the surface is not suitable you must abort and go to a new crater—look out for crater walls, they can be up to 50 metres high.

Never lacking in spice

If the pair of you are good you can usually find a landing site before running out of fuel but the game never seems to lose its spice. One slip in concentration and you are finished.

Writing a lunar landing game on a small microprocessor-based computer certainly teaches you much about the capabilities of that machine. You have to learn to control the output on to the screen and how to read the keys without interrupting the program.

Most of all, it teaches you how to program efficiently. The essence of this program is 'response'.

Graphics is a field which has always fascinated me and I believe that visual display of data by screen graphics will play a much larger role in the future of the small computer. From a computer-aided design viewpoint, this method of data display becomes invaluable.

One of the most difficult and expensive forms of graphics is termed "refresh graphics". Refresh graphics is really no more than the ability to erase selected lines which have been drawn previously on the screen. There is a game which utilises this technique as well as shape rotation and real-time simulation techniques called Space War.

It is a two-player game with each player in command of a spaceship in orbit around a central sun. You may fly your ship by spinning it around and using a main rocket motor for thrust. You have a forward-firing gun and the object is to destroy your opponent before he gets you or one of you blunders into the sun.

The ships and the sun are displayed on a screen and the battle is fought in real-time. The first thing this game teaches you is that buzzing around in orbit is not as easy as it looks. Fairly soon you go off of the edge of the screen and what happens next is what makes Space War so enjoyable to play. You come back on the other edge of the screen.

This is excused in the game by assuming wraparound space but this effect can cause the inexperienced player many headaches. There you are lining up for a shot and your opponent slips over the edge and comes up behind you.

The other major influence on the game board is the sun. The effect of the sun's pull diminishes with the square of the distance from it and using the sun can produce a few surprises for your opponent.

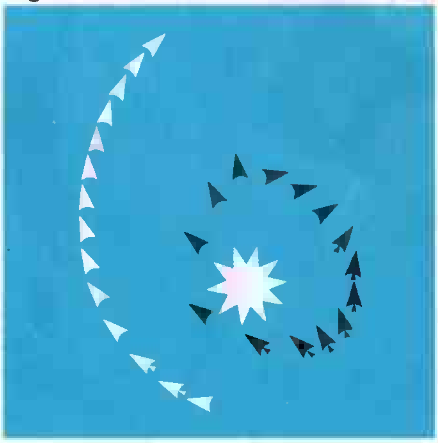
Figure 1 shows a very slick opening move by the good ship White Wedge. The ship spins to the left and starts the rocket—you can see this by the "exhaust" from the base of the triangle. This will brake him in his orbit and the sun will pull him in, accelerating the craft and eventually lining himself up with a sitting duck.

You'll find it refreshing

Several things can be learned from this plot about the necessary computing. Firstly, for purposes of clarification I have left a trail of White Wedges and Black Wedges on the map to explain the sequence of events. On a screen, of course, it is necessary to erase them and that is the refresh graphics part of the program. The other important point is the fact that these pre-determined shapes must be rotated to show the angle of the ship. This involves some useful exercises in vectors and transformation matrices.

The plotting of ships on the screen, although superficially straightforward, proves to be quite a complicated procedure. It is outside the scope of this article to go into detailed mathematics but writing a game of Space War can teach you much about refresh graphics. *By Dr Tim Keen.*

Figure 3



Practical Computing is the new monthly magazine for the computer enthusiast.

Each month we shall be looking at how computers can be used in the home, school and as small business systems.

<u>In The Home</u>

You don't have to be a programmer or computer professional to read Practical Computing. It's the magazine that tells you how to get started, who to buy from, and what you can do with computers in the home.

There are hundreds of applications that you can run on your own computer – and it doesn't cost the earth.

Each month we give a practical review of a home computer, detailing what it can do and how much it costs.

We look at peripherals. You can build your own or buy them off-the-shelf. We also look at what software is available and give you a step-by-step guide to programming your own system.

There's games to get you started. We show you how to develop TV games, space war games, puzzles, sports simulation, music and speech synthesis and how to put your dream games into reality.

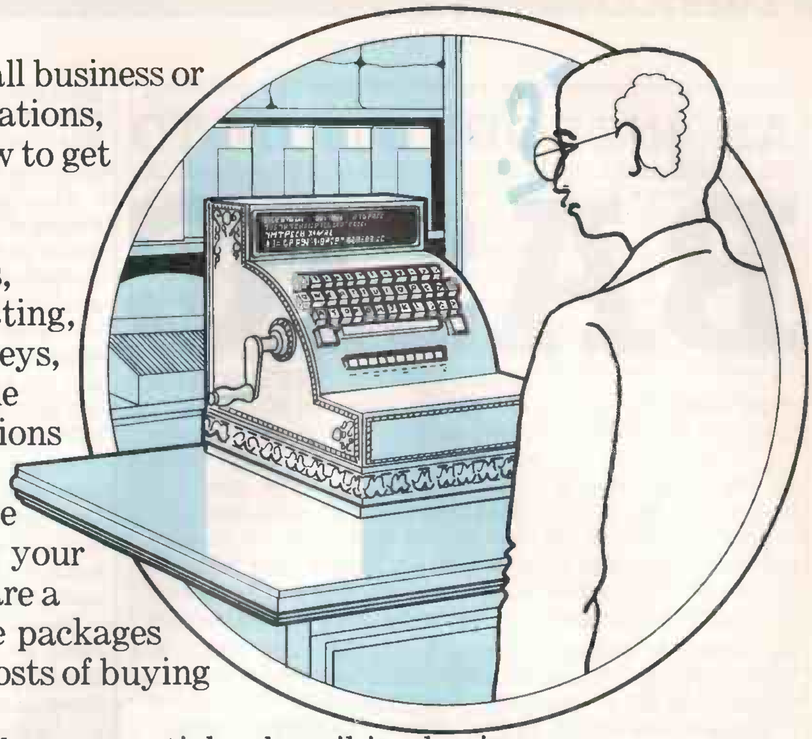
Then there's a host of other applications you can develop like automating your home, sorting out your finances, mathematics, and even how to speak to your computer.

In Business

Whether you are running a small business or you want to develop business applications, Practical Computing shows you how to get started and what you can do.

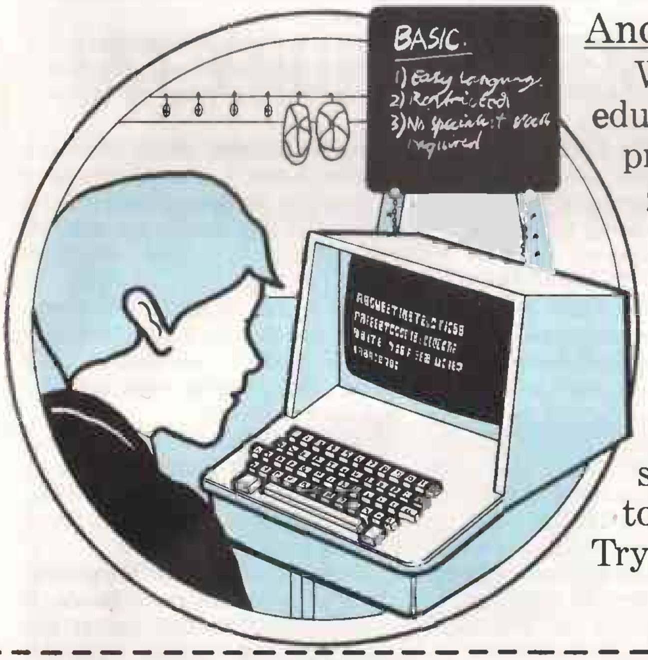
Word processing, accountancy, mailing lists, small business systems, stockbroking, stock control, typesetting, simulations, cash flow, payroll, surveys, bookkeeping, graphics, and real-time control are just some of the applications you could develop.

We show you how you can make money out of developing and selling your computer applications. And if you are a small business, we tell you about the packages that have been developed and the costs of buying your own system.



Each month there are articles describing business applications in detail. We review the equipment you need and how much it costs.

You can save money too! Our evaluations cover microcomputers, terminals, peripherals, software packages, periodicals, booklets and suppliers of equipment. We help you choose the computer that's best for you – before you spend money on one that isn't.



And in Education

We examine how computers can be used in education. Develop timetables, learn about programming, classroom computer projects and games, mathematics and business studies are some of the applications described.

We also look at how other schools, colleges and universities are using computers along with actual project descriptions and case studies.

Practical Computing is fun. We've got style, colour and we're packed with lots of ideas to help you get the most out of your computer. Try us now.

AN INTRODUCTION TO BASIC

BEGINNER'S All-purpose Symbolic Instruction Code. BASIC is a very simple, easy-to-use language which looks more like plain English than any other high-level language.

It uses words like IF, THEN, PRINT and GOTO. Further, it is usually interactive—that is it will tell the user if he has made a mistake and will permit him, very easily, to correct his program and re-run it. The easy style of BASIC and its interactive nature make it a natural choice for the inexperienced programmer. Consider, for example, this simple program:

```
0010 REM PHONE DIRECTORY PROGRAM
0040 PRINT "WHICH DIRECTORY DO YOU WANT";
0050 INPUT F$
0060 OPEN 1,F$: REM OPEN DISK FILE
0070 PRINT
0080 PRINT "NAME";
0090 INPUT C$
0100 IF C$= " " THEN CLOSE 1 : STOP
0110 IF C$= "STOP" THEN CLOSE 1 : STOP
0120 REM ** HERE TO SEARCH FILE FOR NAME
0130 RESTORE 1
0140 READ 1, N$, A$, P$: REM GET NAME, ADDRESS-
     TEL NO.
0150 IF N$ <> C$ THEN 140 : REM NOT THIS ONE
1000 REM HERE IF ENTRY IS FOUND
1030 PRINT "NAME:"; TAB(15); N$
1032 PRINT "ADDRESS:"; TAB(15); A$
1035 PRINT "PHONE NO:"; TAB(15); P$
1040 PRINT
1050 GOTO 40
```

It is quite clear, even at a casual glance, that this is a disc-based file look-up program and the user would have little difficulty in changing it. Contrast that with this short piece of assembly language source code:

```
NAM TEST3 CDAT3
ORG $0100
START LDAA 32
LDX 0DTAB
D10 CLR X
INX
DEC A
BNE D10
LDX # PROMPT
JSR PDATA
CLRA
LOOP9 LDX BEGA
STX XT
```

The contrast is obvious. One can write and debug large programs in BASIC very rapidly. Of course, there is a price to be paid for this simplicity. One finds that there are things which cannot be done with any degree of elegance in BASIC. That applies particularly to handling complex peripherals. The other price one often has to pay is in the execution time of the program.

BASIC is often, though not always, an interactive language. Each line of code is checked for syntax errors as it is entered into the computer. Errors are reported immediately to the user.

There are, however, two different ways of dealing with a user's program. The program can be interpreted or compiled. An interpreter stores the input text almost exactly as it is entered. When the program is RUN, each line is interpreted (dealt with) each and every time it is met. Thus a statement like

```
100 IF A ≤ 100*(3.14159 * (1.5)^2) THEN PRINT "TOO
BIG"
```

will be checked for errors, the constants 100, 3,14159. 1.5 and 2 will each be converted to floating point format and the arithmetic will be evaluated every time the line is evaluated.

With a compiler, on the other hand, the translation to machine code takes place only once, when the program is "compiled". One loses little of the flexibility of BASIC apart from, usually, the interactive editing. Once successfully compiled, the program behaves just like any other BASIC program, but executes very much faster.

The advent of "compiling" BASICs in recent years could well make BASIC much more attractive in time-sensistive applications. The combination of an interpretative BASIC for program development, followed by the use of a compiling BASIC for the production of the final working program, would seem to offer a very attractive system.

Unfortunately, at present there are very few compilers which have a "matching" interpreter. One discovers rapidly that code interpreted successfully in one version of BASIC is unlikely to compile in another version.

Of course, the possibilities of an "interpretive-compiler" have not passed unnoticed. The idea is that text is held both in its original English format, and in compiled—or at least partially compiled—form. If no program changes are made between RUNs then the compiled version is used. If program changes have been made then the program is re-compiled. Needless to say such programs tend to take up a lot of memory. The "incremental compiler" is a further improvement.

The available compiling BASICs are very individualistic. For example, consider two compilers for the M6800, one produced by Microwave Systems Corporation and the other by Software Dynamics.

The Microwave Systems compiler is small, under 8K, and is integer only. One's first impression is that it is probably one of those awful "tiny BASICs". Not so; it has extensive string handling, including a SUBSTR (substring) search, and essential statements like ON ERROR GOTO, ON OVR (overflow) GOTO, ON NMI GOTO, ON IRQ GOTO.

The whole package is designed to run under RT/68, a real-time multi-tasking operating system, though it can be run in other environments. The compiler produces, on request, a full source code and op-code listing. The compiler pulls out only those parts of the run-time package it requires.

So if there are no print statements, the print routines will not be put into the program. It IS somewhat limited; it has no trigonometrical functions, but for many things it will be ideal. The code produced is well optimised and, as a result, very fast. I am not aware of it being available in the U.K. at present.

The Software Dynamics compiler adopts a different approach. It has everything you can think of and a lot more beside. It supports full disc handling, extensive arithmetic, logical and string functions. The code produced on the printer is pseudo-machine code, which is a little difficult to follow at first.

The compiled program includes a full 10K run-time package. The compiler is nothing short of immaculate but using it on a small 20-line program would be like using the proverbial sledgehammer to crack a walnut.

Because of its complexity it is somewhat more complicated to run at present. As so often happens each has something to recommend it and one uses both if one can.

One of the obvious attractions of the BASIC language is that it

(continued on next page)

(continued from previous page)

should promote programme "portability". Any program written for one machine should run on any other machine with BASIC. Unfortunately that is far from the case. ANSI, the American National Standards Institution, has defined BASIC in its simplest form and most versions conform to ANSI standard except in one minor aspect.

The ANSI BASIC is a very simple BASIC and does not include many of the commands one would wish to use. As a result, there is little uniformity and, in practice, it is unlikely that a program written for one version of BASIC will run on another system without modification.

To give an idea of the range of commands and statements supported by the various versions of BASIC, here is a list of some of them;

ABS	AND	APPEND	ASC	ATAN	ATN
AUTO	BYE	CALL	CAT	CH	CHR$
CLEAR	CLOAD	CLOSE	CLR	CMD	COLOR
COM	CON	CONT	COS	CREATE	CSAVE
DATA	DATE$	DEBUG	DEF FN	DEL	DELETE
DIGITS	DIM	DOS	DSP	EDIT	ELN
ELSE	END	EOF	ERR	ERROR	ESCAPE
EXP	FIND	FOR	FORMAT	FRE	FRE$
GET	GOSUB	GOTO	GR	HEX$	HIMEM
HLIN	IF	IN	INP	INPUT	INT
KILL	LEFT$	LEN	LET	LINE	LINK
LIST	LLIST	LNUL	LOG	LOMEM	LPOS
LPRINT	LVAR	LWIDTH	MAN	MID$	MOD
MON	NEXT	NEW	NOT	NULL	NUM$
NUMF$	ON	OPEN	OR	ORIGIN	OUT
PATCH	PEEK	PDL	PI	PLOT	PLT
POKE	POP	PORT	POS	PR	PRINT
PROGRAM	RANDOMISE	READ	RENAME	RENUMBER	REL
REM	RES	RESTORE	RETURN	RIGHT$	RND
RUN	SAVE	SCRATCH	SCRN	SIN	SGN
SPC	ST	STEP	STOP	STR$	SQR
SWITCH	SWITCHES$	SYS	LINE	TAB	TAN
TAPPEND	TEXT	THEN	TI	TI$	TIME$
TLOAD	TO	TRACE	TSAVE	USING	USR
USER	VAL$	VERIFY	WAIT	WIDTH	WRITE
+	—	*	/	**	⌢
=	<	>	=>	<=	$
"	:	;	.	?	

With Charles Sweeten, one of my colleagues, I am 'compiling' a comparison of most of the available BASICs with a view to publishing a guide to indicate to programmers the most common functions and syntax used in BASIC.

Such a guide could, perhaps, help ensure that programs written in BASIC had maximum portability and indicate to users what changes would need to be made to run a program under their particular dialect of BASIC.

BASIC is ideal if you want quickly to write a small program for a specific job which does not require a great deal of arithmetic or string handling, or a very large database. Such jobs as payroll for up to, say, 100 employees, or stock control of perhaps 1,000 items. Order processing and invoicing of about 100 transactions a day would be feasible.

Clearly, the feasibility of these operations will depend on the sophistication of the operating system as much as on the quality of the BASIC interpreter or compiler.

A large proportion of small computer sales in the last year have been made to "small businesses and it is clear that the vast majority of the buyers will be working in BASIC. There is a dearth of good BASIC business software at reasonable prices at present, and most people seem to be writing their own. We are beginning to see a change in that situation, but much remains to be done.

It must be added in conclusion, that anyone selecting a system to run BASIC business software must look carefully at the total support available for the machine he is selecting. A good disc system and a flexible and efficient operating system are essential features of a practical BASIC system.

John Cole had been head of computing at Oundle School for five years, and has been responsible for the design and construction of a substantial system based on the M6800. He is a consultative member of MUSE, the Minicomputer Users in Secondary Education group.

This tutorial is based on his paper at the Do-it-yourself Computer Conference.

MEET YOUR COMPUTING NEEDS FROM
PADMEDE
We supply APPLE IIs

Apple II is a self-contained, ready-to-use computer

Its features include:

- Simple to use, yet portable and powerful (65K on-board memory)
- Includes Basic and monitor in ROM (8K bytes)
- Colour graphics display to home TV, audio cassette interface, game i/o connector, typewriter style keyboard, speaker.
- Eight peripheral board connectors which can be used for printers, voice recognition, terminals, floppy discs, etc.

Whether your applications are for the home or business, we at Padmede would like to understand your needs

If you want to buy an Apple II, or just have a demonstration, visit or contact Trevor Williams now at:

PADMEDE LTD
The Tuns, High Street, Odiham, nr. Basingstoke
Tel. Odiham (02-5671) 2434

Padmede is an Apple II agent for Surrey, Sussex, Hampshire, Berkshire and Oxfordshire

● **Circle No 110**

'78

SHOWGUIDE

The 1978 Do-it-yourself Computer Show

22-24 June 1978
West Centre Hotel,
London

online

Cleveland Road,
Uxbridge, Middlesex
(0895) 39262

Fun followed by disaster has spawned successful industry

Dr Adam Osborne's address to the Do-it-yourself Computer Show

THE TWO recurring characteristics of companies prominent in personal computing today are luck and a lack of professionalism.

None of those companies would exist had their founders been guided by the traditional market surveys and product analyses. In fact, a number of established companies—such as Intel, Rockwell, Fairchild and General Automation—looked at the personal computing industry in its infancy. Unanimously they dismissed the industry and its products as a passing aberration, with no chance for long-term success or growth.

That is why the well-known personal computing companies began as two or three naive engineers putting together hardware for the fun of it.

Personal computing began with a small company, MITS, in Albuquerque, New Mexico, which built a kit around the Intel 8080A microprocessor.

There were earlier kits available; for example, Scelbi, which publishes books today, had an Intel 8008-based kit earlier than the MITS 8080A-based kit.

In 1974, however, it was MITS which brought personal computing into prominence by advertising its product in *Popular Electronics* magazine.

The MITS founders looked upon their kits as an advertising sideline, not as their principal line of business. They were amazed at the response they received from their *Popular Electronics* advertising.

Having produced a sales forecast, with the usual optimism of entrepreneurs, they sold, in three months, the number of kits they had planned to sell in two years—and personal computing was born.

Now, by today's standards, which are not very high, the original MITS kit was an engineering disaster. Its logic cards had no side guides and a small jolt would cause them to rattle into each other and short-out.

Generating cash flow

There was no reasonable expansion ability and MITS blessed us with the S100 bus, which was ill-defined and badly-engineered. Nevertheless, without peripherals or software, the kits sold in good quantities.

IMS Associates was the next significant manufacturer of personal computing hardware. Its entry resulted from $250,000 that it was paid, up front, to interface the MITS kit to a PDP-11 minicomputer for certain instrumentation applications.

Based on the overwhelming demand MITS advertising generated, it was unable to supply IMSAI with systems. Instead of spending the $250,000 on building its own microcomputer system, IMSAI advertised its system to generate enough cash to fulfil its initial interfacing contract. A few advertisements generated more than sufficient sales and cash, and IMSAI gained a rapid and prominent position in the market-place.

The IMSAI entry into personal computing is not typical. More typical are companies like Apple Computer Corporation, Processor Technology, and Vector Graphics, whose initial products were put together by engineers working part-time, as much for the fun of it as for profit.

They looked at the kinds of products MITS and IMSAI were selling and concluded that, without trying very hard, they could build a better product.

The early microcomputer hobby customer base was amazing in the fervour with which it pursued products. Customers were so eager to buy that they established an industry payment practice which was initially necessary but which has become a liability.

Advertised in advance

It is the practice of paying in advance for products, frequently months in advance of delivery. In the early days of the personal computing industry, front-end payments were absolutely necessary, since manufacturers had no track record, no visibility, and no chance of obtaining loans or funding through any traditional sources.

Instead, they were forced to advertise a product before its existed, demand cash payments with orders, and then spend the cash to build the product which had been ordered.

This is called "forward financing". It works while business is increasing rapidly and products are designed on time. Business has, indeed, increased rapidly, but products have not always been designed on time.

Frequently, they have been very late or have not worked at all. There were, for example, numerous dynamic read/write memory boards which were never properly designed and were shipped when only partially working. In consequence, companies occasionally ran out of cash before a product for which there were many paid orders had been designed.

Larger companies with more than one product spent money which had arrived in payment for product A to complete the design of product B. In the U.S. those practices are illegal and can result in prison sentences for offending management.

The early personal computers were sold with no support software. That vacuum was filled initially by enterprising amateurs with insufficient background in either software design or business practices. The software they designed was not very good, and few people bothered to pay for it; they made a copy instead.

Hobbyists distinguished themselves as thieves. A few software designers out of

(continued on next page)

(continued from previous page)

the early hoard were experienced and understood business practices.

Applications software has been all but non-existent. This is to be expected, since application programs cannot be written until operating systems and language interpreters or compilers are available. It is worth noting, however, that most of the early applications software which has been sold is incredibly crude, paying little attention to error recovery, audit trail, or file maintenance.

That very bleak situation did not kill off personal computing in the U.S. because the customers did not know what to expect and therefore did not realise how badly they were being treated. They had products which, for the first time, were within their price range.

The early chaos of the home computing industry in the U.S. perhaps had a more detrimental effect in Europe than it did there. All the early participants—hardware manufacturers, software writers and book and magazine publishers—were too small to understand or cope with foreign sales. Handling the domestic market was already more than they could cope with.

Maximum in minimum

In consequence, U.S. companies sold in European markets through distributors, some of whom were interested in maximum profits in minimum time. Personal computing almost certainly made its slow start in Europe because the double overlay of avaricious distributors on top of incompetent suppliers was enough to discourage almost everyone.

Success, however engenders improvement. There were so many hardware and software manufacturers and the market has proved so buoyant that a significant minority of companies emerged with quality products.

That, in turn, caused two further developments—more qualified people moved into the industry and capital became available.

Stu Mabon founded Micropolis to build floppy disc drives. He had previously founded Pertec, which is one of the major companies in the minicomputer peripherals business. Pertec acquired MITS. Since the founding of Micropolis and the purchase of MITS, numerous established companies and well-known individuals have entered the business. Even IBM is selling the 5110 system as though it were a personal computing system.

Arthur Rock, who made his fortune backing Scientific Data Systems, recently invested a significant sum in Apple Computer Corporation, in return for an equity position in the company. Other venture capital companies tend to follow Rock's lead. Today, any established microcomputer system manufacturer in the U.S. can obtain as much capital as he needs, and probably more than he really wants. With the money comes professional management.

‘The commercial market will grow very fast and that, in turn, will give hobbyists greater opportunities to turn their hobby into profit’

The personal computing industry is, therefore, in the middle of a transition. Some companies are already run solidly and financed adequately; others are in the process of obtaining capital and management, while a few are still struggling the old way.

Within a year a generally sound industry operating in a responsible fashion can be expected, but it is improved products which will really make the industry grow.

While the Intel 8080A and Motorola 6800 microprocessors provide interesting computing capabilities, the Intel 8086, the Zilog Z8000 and the Fairchild 9440 offer more. As cassettes and small floppy discs certainly make simple applications feasible, a new class of "Winchester technology" fixed discs will be available in 1979. These fixed discs offer 10 to 20 million bytes of disc storage and will probably retail at between $2,000 and $3,000.

Low-cost printers are also on the horizon. Low-cost system software and applications software are also appearing rapidly.

What does this mean to the hobbyist? In reality, we have seen very few true "hobbyists" in the U.S. That is to say not many people spend a great deal of money buying their own computer systems simply for the fun of building the computer system and programming it. They usually intend to turn the hobby into a second profession, or at least into a profitable hobby.

Programming consultants and hardware design consultants have often started as hobbyists; in fact, many of the well-known companies were founded by hobbyists. Steve Jobs and Steve Wosniak, the founders of Apple Computer Corporation, for example, began as hobbyists. Therefore, the fact that products are improving rapidly in terms of quality and capabilities is of great interest to hobbyists, since it means the commercial market will grow very fast, and that, in turn, will give the hobbyists greater opportunities to turn their hobby into profit.

No discussion of personal computing in the U.S. would be complete without looking at the phenomenon of the computer store. Dick Heizer opened the first in Los Angeles in 1975. Today, there are approximately 600 computer stores in the U.S. At the present rate of growth, there will probably be 5,000 by 1980. The average computer store today grosses $75,000 to $100,000 per month; after paying for goods and operating expenses, it allows the store to make a modest though unspectacular profit.

Computer stores do not make their living serving hobbyists. Computer kits are all but dead and the amount of equipment individuals buy for their personal entertainment would not amount to any significant business. Computer stores are serving a commercial customer base; in 1978, it is a small business data processing customer base.

Calls generated no sales

In the past, the smallest business computer system cost $50,000, and after necessary options had been added the cost was more than $100,000.

Those high prices assumed that a sales-person would visit customers to sell computer systems and subsequently to make sure that the systems remained sold. Since each customer talked to many vendors before buying, much money was spent by manufacturers in making customer calls which generated no sales. That method of selling business computer systems is expensive. It is also frequently counter-productive, so the computer store has changed the rules of the game. Now the customer visits the store rather than the vendor going to the customer. At the store, the customer can see different computer systems and options.

What, then, is the future of personal computing? There is unlikely ever to be a microcomputer system in every home. The idea that people will buy microcomputer systems to balance their bank accounts or to look up recipes makes interesting journalism, but it is unrealistic.

Microcomputer systems will be bought, if they are cheap enough, to play intelligent games, or for educational purposes. But then, like Teletex, they become dedicated instruments supporting specific functions.

That is not to say that there will be no home computers, programmed purely for entertainment; there will, just as there are ham radio stations operated for the pure joy of talking to another ham radio station. Yet the true future growth of personal computing lies in that vast grey area between the personal and the commercial computer. There is no clear division between one market and the other; rather there is a continuum between one extreme and the other.

Dr. Adam Osborne, author of Personal Computing, where it came from and where it is going, is the founder of Osborne & Associates, Inc. a publishing and Consulting Company. He is the author of more than 50 books and manuals describing all aspects of minicomputers and microcomputers.

List of Exhibitors

Airamco,
(U.K. agents for Jade Computer Services),
30 Witches Linn,
Ardrossan,
Ayrshire.
Stand No: 28

Amateur Computer Club,
7 Dordells,
Basildon, Essex.
Stand No: 34

The Byte Shop,
426-428 Cranbrook Road,
Gants Hill,
Ilford, Essex.
Stand No: 1&2

CCS Microhire,
7 Pear Tree Dell,
Letchworth,
Herts.
Stand No: 51

Comart,
PO Box 2,
St Neots,
Huntingdon.
Stand No: 5&6

Commodore Systems Division,
446 Bath Road,
Slough.
Stand No: 37&38

Compelec Ltd.,
107 Kilburn Square,
London, NW6.
Stand No: 49&50

Computabits Ltd.,
41 Vincent Street,
Yeovil,
Somerset.
Stand No: 12

Computer Workshop,
38 Dover Street,
London W1.
Stand No: 29&31

Crellom Electronics Ltd.,
380 Bath Road,
Slough.
Stand No: 4

GRT (U.S.A.),
California.
Stand No: 48

Interam Ltd.,
59 Moreton Street,
London SW1.
Stand No: 35

G. Kewney,
Computing,
76 Dean Street,
London W1.
Stand No: 12

LP Enterprises,
313 Kingston Road,
Ilford,
Essex.
Stand No: 3

The Micronics Company,
1 Station Road,
Twickenham,
Middx.
Stand No: 7

Motorola Ltd.,
York House,
Wembley,
Middx.
Stand No: 12

Nascom Seals Ltd.,
92 Broad Street,
Chesham,
Bucks.
Stand No: 30

The Newbear Computing Store,
7 Bone Lane,
Newbury,
Berkshire.
Stand No: 44

Personal Computers Ltd.,
18-19 Fish Street Hill,
London EC3.
Stand No: 24

Personal Computer World,
62A Westbourne Grove,
London, W2.
Stand No: 26

Practical Computing,
2 Duncan Terrace,
London N1.
Stand No: 33

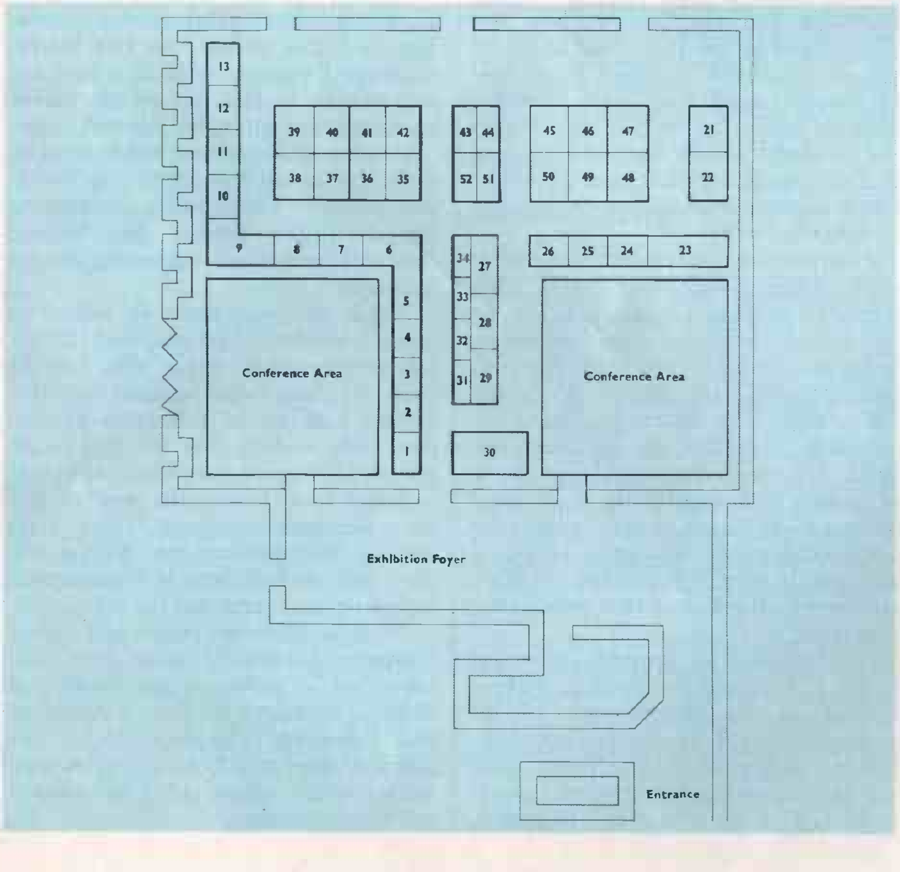
Sintrom Electronics,
14 Arkwright Road,
Reading,
Berkshire.
Stand No: 27

Strumech Engineering Ltd.,
Portland House,
Coppice Side,
Brownhills,
Walsall,
Staffs.
Stand No: 21&22

Wilcox Computers Ltd.,
Rackery Lane,
Llay, Clyde.
Stand No: 11

Xitan Systems,
31 Elphinstone Road,
Highcliffe,
Dorset.
Stand No: 13

Zartronix,
Unit 116B,
Blackdown Rural Industries,
Haste Hill,
Haslemere,
Surrey.
Stand No: 32



General Show Information

Hours of exhibition

Thursday, June 22, 11.00am to 7.30pm
Friday, June 23, 11.00am to 7.30pm
Saturday, June 24, 11.00am to 7.30pm

1978 Do-it-yourself Computer Show Organisers

The 1978 Do-it-yourself Computer Show is organised by Online Conferences Ltd, Cleveland Road, Uxbridge, Middlesex UB8 2DD.

Catering facilities

The exhibition foyer has a bar, and a snack bar will be provided. In addition, the hotel will supply a stand service, this being a service of drinks, tea and coffee on a cash-only basis. The West Centre Hotel has full restaurant facilities.

Telephone facilities

Coin boxes are situated around the hotel.

Company-by-company guide

Airamco & Jade

Stand No 28

A JOINT stand for Jade Computer Products and Airamco. Representatives of both companies will answer general enquiries and give advice on how to build and extend computer systems.

Some of the products on show are the Jade 8080A cpu board (S100); Jade 280 cpu board (S100); Jade serial/parallel board (S100); Jade real-time clock (S100); Jade RAM memory board 8K, 18K, 32K versions (S100); and the Jade Dynamic RAM board 8K, 16K, 32K versions (S100).

The company will also be showing an EPROM board for 2708/2716s, plus PROM SETTER programming boards suitable for programming the following EPROMs: 1702A, 2708, 2716, 5204 and 6834 (S100).

KIM kits and S100 mother boards and power supplies will also feature on the stand.

Customers may place orders on the stand for items on display with delivery after the show.

Amateur Computing Club

Stand No 34

THE AMATEUR Computer Club will be giving information about what the club does and provides for its membership. Special article on the club in this issue.

The Byte Shop

Stand No 1 & 2

THE BYTE SHOP will be exhibiting a selection of products on sale in its shop. They include Southwest Technical Products in both kit and assembled form, together with a variety of peripherals suitable for the 50 bus, audio cassettes, floppy discs, music boards, VDUs and printers.

The newly-introduced SOL computer will also be on show, showing how attractive the latest American packages can look. There will also be a range of Cromemco products, microcomputers, floppy discs (micro and 8 in), plus add-on memory from a variety of manufacturers, including assembled Dynabite.

A wide range of peripherals for the S100 bus will be readily available. Colour graphics and speech recognition will be demonstrated on the lightweight, versatile Apple computer.

As The Byte Shop is a Commodore distributor, the PETs will be there in abundance.

For the business user, a corner has been dedicated to commercial requirements and includes the Digital LSI-11 commercial system.

As The Byte Shop is tailoring its supplies to the needs of its users, its product range is expanding all the time. There will, however, be on the stand available from stock Lear Siegler VDUs, Centronics micro printers, Tally Line printers, Teletype 43, Cyfer VDUs, DECwriter terminals, a range of floppy discs from North Star, Smoke Signal Broadcasting, PerSci, Calcomp, Cromemco, SWT.

CCS Microhire

Stand No 51

CCS MICROHIRE will be explaining how you can hire a microprocessor system and get hands-on experience before you commit yourself to buying.

The company is offering hiring arrangement from one day to as long as you like at a starting price of £2 a day.

Among the equipment it will show you can hire is Apple II, Commodore PET, Nascom 1, Research Machines 380Z, SWTC 6800, and the Tandy TRS 80.

Comart

Stand No 5 & 6

COMART (Computer Mail Order and Retail) Ltd will be launching the SOL terminal computer system and exhibiting it for the first time.

The SOL, from Processor Technology Corporation, is based on the 8080 microcomputer and conforms to the S100 bus standard. Attractively packaged in a walnut and blue metal cabinet, the SOL includes keyboard, video, serial and parallel interfaces, and 19K bytes ROM and RAM memory. Basic and Extended Basic are available, together with an 8080 assembler.

SOL S100 compatibility allows it to be configured with other Comart products; the North Star diskette systems, Dynabyte Memory, and a range of interface and PROM boards from Cromemco.

Also making its U.K. exhibition debut is the Cromemco System 3. The CS is designed around the Cromemco Z2 processor, floppy disc controller and drives, 32K byte memory, and serial and parallel interfaces.

With powerful software for development, scientific and commercial use, the CS3 is unrivalled as a general-purpose microcomputer system. Software packages include Fortran IV, Extended Basic, Z80A Micro-Assembler, Editor, Debug and Trace—a simulator programme.

With the introduction of a new 8-slot S100 card cage, Comart is offering more of its products to the industrial and OEM user. The Z2-D Z80 development system, with its EPROM programming facility, encourages development engineers to the advantages of the range of S100 processors, memory and associated input output modules.

Compelec Ltd

Stand No 49 & 50

COMPELEC will be demonstrating a new range of computer systems.

On the stand will be the multi-user, dual disc system based on the Altair 800B. Price for the system, which includes 33K RAM, two Pertec mini floppy disc drives, PROM monitor, serial interface and turnkey processor, is £3,100.

There is also a single-user business system with Soroc vdu, Qume printer, and 64K RAM. Application packages available for this system include sales ledger, purchase and bought ledger, invoicing, property management, hotel booking and estate agents package. Price is approximately £9,000.

Also on display is a hard-disc system which uses Pertec 10Mbyte discs (five fixed/five removable), plus a range of languages like Cobol, Fortran and Basic and other packages such as mailing lists and addressing.

Computabits Ltd

Stand No 12

COMPUTABITS LTD will be showing its range of small computer systems, the smallest of which is the Commodore PET. The company is the authorised distributor in the South-west of England.

Its range of larger systems are all based on the industry-standard Altair/Imsai S-100 Bus. Manufacturers represented include Cromemco, Dynabyte, Godbout and Tarbell.

Products of these manufacturers which are available as either separate boards or complete systems include Z-80 based central processors, static or dynamic memory in 8K, 16K, and 32K configurations—these include some very low cost units. Also included are interface boards, serial, parallel and analogue, as well as floppy disc drives and controllers.

Software includes operating systems, Basic, Fortran IV and Assembler.

For the small business system, Computabits can supply a new low-cost, British-made computer, the Computabits System 8. It is an 8080 based S-100 Bus computer, incorporating 32K bytes of RAM (expandable to 64K), and one megabyte of floppy disc storage on two double-sided 8 in. drives. The computer power supply and disc drive are contained in a single cabinet designed as an attractive desk-mounted unit.

(continued on next page)

(continued from previous page)

Computer Workshop

Stand No 29 & 31

COMPUTER WORKSHOP will be highlighting its range of computer equipment. It sells both small and large systems. For £4,650 you can buy a cpu, one million characters of disc storage, vdu and a 60 cps printer with 132-column width. It has a disc operating system and sophisticated Basic. The operating system has facilities such as print job queuing and spooling while you are using the computer for other purposes.

At the smaller end of the scale, the company sells a cpu with 4K bytes of memory for £330. Additional memory is available at £70 per 4K; serial and parallel interfaces at £37 each; a vdu for £445; and a 40-column printer for £250.

Software such as a text editor and text processor, said to be as powerful as those found on large mainframes, is available for £25 each.

The majority of Computer Workshop equipment is manufactured by Southwest Technical Products in England. Maintenance contracts can be arranged with Computer Field Maintenance as well as leasing facilities.

Computer Workshop is also planning to start a series of three-day course costing £100, covering a range of topics from programming to the use of microcomputers and specialised applications. Details on the stand.

Interam

Stand No 35

INTERAM will be demonstrating the Equinox and Horizon computer systems, both of which use the Industry Standard B100 bus.

The Equinox mainframe incorporates an 8080A based CPU/front panel combination, Wunderbuss 20-slot Bus board and a 26 amp constant voltage (ferroresonant) power supply. The standard system configuration includes 32K bytes of static RAM, dual Memorex floppy disc drive, two parallel and three serial I/O ports, CP/M disc operating system and extended Basic compiler.

The Horizon Z-80A-based computer incorporates a 12-slot bus, 15 amp power supply, serial I/O ports and dual integral Shugart mini-floppy disc drive. A parallel and a second serial I/O port can be added at little extra cost. The Horizon system runs at 4 MHz and the standard configuration includes additionally 32K bytes of memory, North Star DISK operating system and extended Basic.

Printer terminals and VDUs can be added to either system to complete, in each case, a powerful compact inexpensive computer system.

Guy Kewney's own computer thinkshop

Stand No 12

ON his stand will be various loaned bits and pieces, including some kits he has built himself and does not understand—he says.

There will also be a constant supply of technically well-informed assistants of the calibre of Dave Leiper, Micro marketing manager at Motorola, and Brian Crank, Intel home-computer-owning public relations boffin.

Anyone who wants to answer, or indeed ask questions, is welcome to help, or visit the stand.

He will also be plugging his weekly personal computing column in *Computing*.

LP Enterprises

Stand No 3

LP ENTERPRISES will be showing its range of books and publications. The company handles subscriptions for a variety of American personal computing publications, as well as holding a large selection of books on nearly all the hobby computing topics.

Examples are books on introduction to micros, business programs in Basic, games, tricks and puzzles, the artist and the computer, and pocket guides for programmers.

The titles of the books and publications are too numerous to mention. A browse on the stand will reveal all.

Micronics Company

Stand No 7

MICRONICS will show the MICROS I system which is based on the Z80 cpu chip. The basic instrument housing contains cpu, power supplies, memory, keyboard, video interface, UHF modulator and audio cassette interface. Output may be displayed on a uhf TV or video monitor and programmes may be recorded and retrieved from a standard audio cassette.

A resident monitor enables programmes to be edited and manipulated easily; the command codes include single step, halt, store on cassette, retrieve from cassette, set breakpoint, tabulate memory locations, execute from N, move memory blocks, modify memory locations and load to/from Teleprinter.

Typical peripherals fully compatible include paper tape readers, punches, printers, PROM programmers, digital to analogue and analogue to digital converters. Specific status conditions can be programmed to interrupt the cpu, for example an alarm condition in a peripheral device. An RS232 serial input/output port is also included with a transfer rate up to 200,000 baud. Memory extensions will shortly be announced to expand the RAM to 64K bytes.

Price for MICROS is now listed as £399 assembled, and £360 in kit form. An impact printer for £150 is expected to be announced shortly.

Nascom

Stand No 30

THE MAIN exhibit is the NASCOM I microcomputer, which is one of the country's top low-priced systems. The basic NASCOM I kit costs less than £200 and has 2K of RAM, together with its 1K monitor EPROM. But the attraction of the system is its futuristic design which allows for expansion.

NASCOM RAM expansion boards have the option of 4K, 8K, 16K and 32K of dynamic RAM, depending on choice of RAM with 4K dynamic or 16K dynamic. The RAM board also has capability for 2 EPROMS and NASCO will soon offer a tiny Basic in two 2708s to fit this board.

Other developments in the pipeline include two board-based firmware packages—one with assembler, the other Basic—and a floppy disc drive kit. Prototypes, are also being designed, include I/O board which will allow a vast range of interface potential with the outside world. Kansas City interface is available for upgrading tape acceptance.

On the software front, there is the International NASCOM microcomputer club. Already several thousand users and non-users have expressed a readiness to enrol. The INMC will enable users to pool their programming talents and take advantage of the low cost programs.

NASCOM I is one of Britain's own microcomputers and is in demand throughout Europe. In the initial months, demand exceeded supply to such an extent that long delivery delays were inevitable. The situation is now reaching a state of equalibrium as suplies at the new high volume are reaching Chesham, and it is envisgaed that by the end of July NASCOM I will be on one-two weeks' delivery.

Newbear Computing Store

Stand No 44

THE Newbear Computing Store is a division of Newbury Laboratories. The company will be showing its range of equipment for the computing enthusiast, both professional and hobbyist.

Among the products is its own 77-68 (6800-based) system which starts at £49·50 and can be expanded with more memory, visual display units and floppy discs to a "top-flight" system. For the professional sector—or the hobbyist with plenty of money—Newbear offers the Cromemco Z2 system, which is Zilog Z80-based.

Also shown on the stand will be the

(continued on next page)

(continued from previous page)
company's Bear Bag concept. There are 10 which consist of the components necessary to build a part of a system. For example—PETIVID (vdu kit), Kansas City cassette interface, memory boards and the like.

A range of software and literature is available and expanding rapidly, says Newbear. For those with component problems, the store holds extensive stocks of electronic components, including low power SCHOTTKY TTL integrated circuits.

Personal Computers Ltd

Stand No 24

PERSONAL COMPUTERS is a U.K. company based in the City of London, and is the distributor for Apple Computers Inc.

The company will be showing its Apple II computer (reviewed in this issue) which is a very powerful 6502 microprocessor using Basic programming language with high-and-low resolution colour graphics facilitities.

Personal Computers says that it is a machine at the very beginning of its product life with spare input/output ports, ROM and RAM areas. Several cards to enhance the machine are already available.

Personal Computers describes the system as easy to use, reliable, simple to maintain, easily upgradable and very well documented and it will be demonstrating these points at the show.

Applications for which the machine is being used include time-sharing, and markets like education, engineering, scientific research, banking and to OEM computer companies.

Personal Computer World

Stand No 26

PERSONAL Computer World will be showing its monthly magazine which carries articles for the beginner, hobbyist, businessman and expert.

SEED

Stand No 21 & 22

SEED will be announcing several new disc products, including a 10 megabyte disc system (five fixed/five removable) for use with the Midwest Scientific Instruments computer systems. The disc system is complete with controller, and disc operating system.

Other new products on display include the ACT-1 keyboard and video monitor.

Several system configurations based on the MSI 6800 processor will be shown, including the FD-8 floppy disc system, with other supporting hardware and software.

SEED is the sole distributor in the U.K. for Midwest Scientific Instruments and Smoke Signal Broadcasting. The company can supply fully-customised systems including a comprehensive maintenance contract.

Sintrom Microshop

Stand No 27

SINTROM Microshop is showing a range of microcomputer equipment, from full systems using professional peripherals to individual hardware components.

Based on the Motorola 6800, the Southwest Technical microcomputers are now U.K. manufactured, as is the 780/8080 Perex system which is ideally matched to the Micropolis range of minifloppy discs, says the company.

Sintrom is based in Reading, with its own shop, where visitors can try the equipment and gain some "hands-on" experience before purchasing.

Xitan

Stand No 13

XITAN SYSTEMS is the south coast distributor for Comart Ltd, and an importer of various S100 Bus products. It is also a marketing agent for a range of software development tools for microprocessors. The company specialises in complete Z80 floppy disc-based systems, including support peripherals.

Among its major products is the cost-effective INFO 2000 half-megabyte floppy disc subsystem, which provides an operating system, dual drive, two RS 232 channels for console and line printer for £1,800. Xitan also intends to have on show the latest and cost-effective word processing Electric Pencil II.

On demonstration will be a pre-release version of the new TDL disc Basic interpreter, including such features as text replacement, real-time access of console, and a source code security feature.

Conference programme

Personal Computers in Business

Thursday, June 22

Adam Osborne, president, Osborne & Associate Inc
Home computing in the U.S.A.

Guy Kewney, *Computing*
Microprocessors—a review of the range and variety available to the hobbyist, including the more advanced/costly machine types S100 v SS50 v the rest.

John Marshall, Lynx Electronix
Manufacturer's evaluation—systems for the first-time personal computer user.

Martin Healey, consultant & University College, Cardiff
Instruction sets and system software.

John Coll, Oundle School
A review and comparison of BASIC.

Chris Mollor, Modular Technology
How to wire your house for total computer control.

Philip Barker, University of Durham
Application and case study—BINSYS, a personal information retrieval service.

Steve Glynn, CAP Microsoft
Musical microprocessors.

Personal Computing in the home

Thursday, June 22

Chairman's introduction—Martin Healey, consultant & University College, Cardiff
Micros in business. Development over the last few years and their value to the small businessman.

Keynote address—Adam Osborne, president, Osborne & Associates Inc
Recent developments in U.S.A. and probable future trends.

Guy Kewney, *Computing*
Review of the range and variety of equipment available on the U.K. market, with particular reference to the small business.

Ken Smith, Electrical Research Association
Introduction to microprocessor systems.

Vincent Tseng, ICL Dataskil
Micro manufacturers' development systems.

Chris Le Tocq, CAP Microsoft
MicroAde—A general-purpose approach for the small business.

Martin Healey, consultant & University College, Cardiff
How the small business can be developed through the use of microprocessors.

Friday, June 23

Chairman's introduction—David Clarke, Oxford University
A review and comparison of assemblers and macro assemblers.

John Coll, Oundle School
Software, a review and comparison of manufacturers' and suppliers' BASIC for the businessman.

Keith Baker, University of Sussex
A review of operating systems and other system software.

Stuart Kitchener, Texas Instruments
Memory systems—a review of the systems available and their applications for the small business.

John Paterson, University of Reading
Case study—The microcomputer and the construction industry.

Paul Woodward, University College, Cardiff
Case study—a custom-built word processor system on a personal computer.

David Broad, Comart
Cromemco—A personal computer as a development system.

Mike Gurr, BOC
Case study—a microprocessor system for estate agents.

Roger Pye, CSP
Teletext and Viewdata—their nature and relevance to the small businessman.

Learn to play the Micro way

WE LOOK at how you can develop a simple musical program to run on your micro. Playing tunes on a computer is of course, by no means a new development but Steve Glynn and friends have produced a method by which you can move your music program from one micro to another.

THIS ARTICLE aims to set out the design for, and one implementation of, a portable music interpreter for microprocessor usage.

As anyone involved with computers knows, the persuasion of a computer to play tunes is by no means new, some early machines even being able to accompany themselves in four-part harmony.

The arrival of the microprocessor, however, has brought plenty of computing power within the grasp of many people with a musical bent, to whom the idea of a musical microprocessor is a new and, hopefully, interesting one. It is to such people that the article is directed.

Anyone coming new to musical notation could be forgiven for assuming that it had been designed with digital computers in mind. Notes are arranged in groups of eight called octaves — well seven actually, since the first and last notes are the same, but who ever heard of a heptave (septave)?

If the intermediate notes (semitones) are included, there are 12 in each octave. Since the notes are arranged in a logarithmic sequence and the frequency of any octave is twice that of the preceding one, the frequency of any individual note may be found by multiplying the previous one by $\sqrt[12]{2}$ or approximately 1.0595.

The note lengths are based on an even simpler relationship, that of integral powers of twice a fundamental minimum length. It is also possible to extend a note by 50 percent by 'dotting' it.

This is virutally all that need be considered for the present discussion, and will suffice for most music, classical and modern.

Requirements for the program

The original idea was simply to write a music-playing program, more as an exercise than anything else. The constraints were that, firstly, there be the absolute minimum of extra hardware, or even modifications to existing kit.

Secondly, that the result be as musical as possible, within the limits imposed by the first criterion. This is something of a problem, because the micro, being a digital device, can produce only squarish waves, and they are not very musical. After going through a cheap amplifier and loudspeaker, however, there is not much left recognisably square about the tone. The first requirement was met by deciding that the interface would consist of the crocodile clips and a capacitor. Since the original conception of this program, Christopher Smith, a colleague, has proposed an even simpler interface but this has not been adopted for this article.

Turning to the software involved, the requirements were that the program be able to produce a reasonable subset of the notes on a piano keyboard—yes, even the black ones— and that the score of the tune to be played be fairly easy to convert to a binary code to enter into the memory of the microprocessor. It was also considered desirable to be able to implement some of the other features of printed music.

Thirdly, it was necessary that the binary representation of a tune be

portable—that a given piece sound exactly the same, regardless of the micro on which it is being performed. Lastly, the programs should allow for the improvement of their design, and for the addition of new and desirable reatures.

The design so far

Having established these criteria for the program, the following design points emerged:

Each note of the tune is represented by two bytes. It would have been pleasant to use only one but there simply is not enough information in 8 bits, and unless you have a IMS61000 micro, the next interval is 16 bits. This leaves some redundancy, which is good if there is to be room for expansion.

The music is to be portable, so the note pitch and duration are table-driven. This makes for easy tuning, or tempo adjustment.

There is an 'escape' sequence so that repeats and key changes may be implemented. The first of these is very important, and on many pieces of music can overcome easily the disadvantage of the two-byte notes.

In detail, the structure of the music word is as follows:

There are 12 semitones in an octave and at least three octaves are needed to produce any reasonable music. Six bits of the first byte therefore are used to represent any particular note, thus leaving room for another couple of octaves, when necessary.

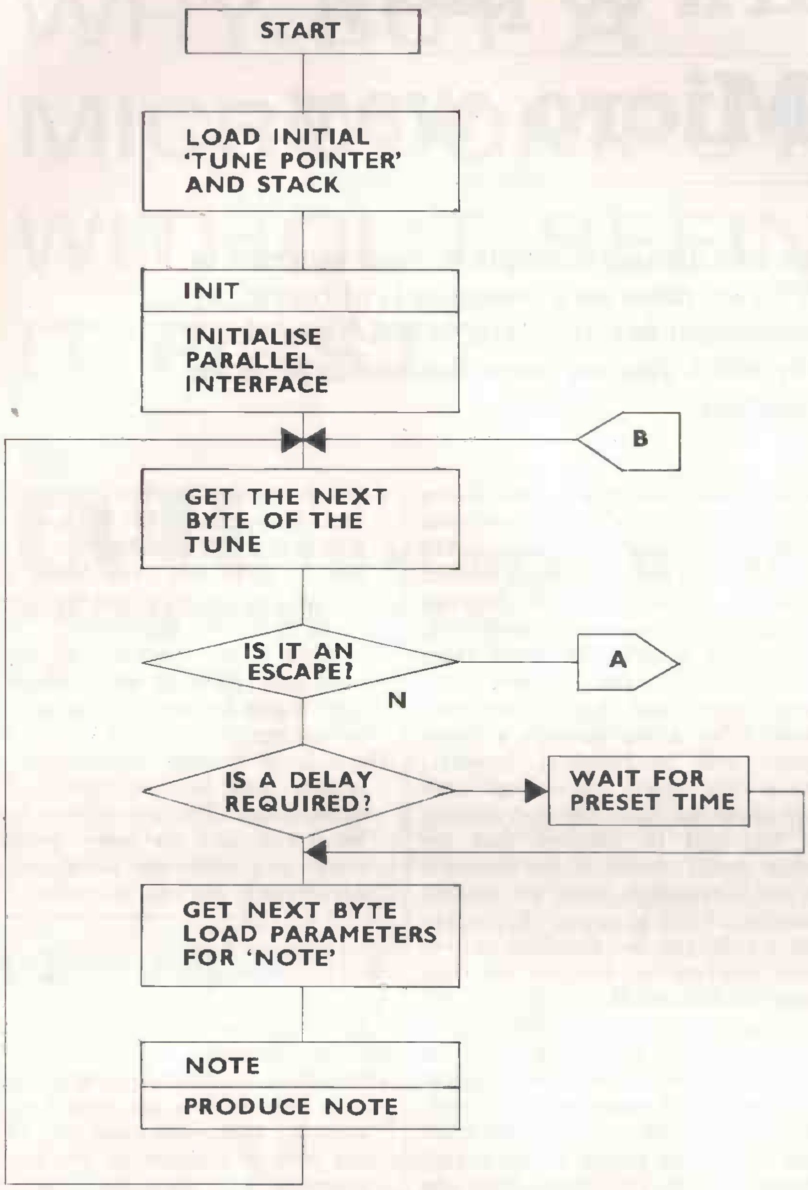
The initial design used the top two of the six as an 'octave-indicator' and used multiplication of a basic note frequency to obtain the actual pitch value. While being somewhat more elegant than the current method, there was a tendency for pitch accuracy to be suspect, especially in the higher frequency region.

The upper bits of the byte are used as a slur indicator and escape sequence indicator. Bit 6, if set, introduces a pre-set delay before the sounding of the note. Bit 7 indicates that the word is not a note but a control code. At present, this is used for implementing repeats and for indicating the end of the tune.

The second byte indicates the length of the note to be played. In the current version of the interpreter this is used to access a table of delays which will allow note lengths from hemi-demi-semiquaver

(continued on next page)

MAIN PROGRAM LOOP



(continued from previous page)

—that is one of these-♪—to a semibreve, which is 128 times longer.

Intermediate dotted values are also allowed. The square bits in this byte are intended for the future production of complicated things like triplets, quintuplets, trills and the like.

Escape sequences

These are so-called, not because it allows the audience to leave in the middle of a piece, but because the interpreter is made to switch context on encountering such a sequence, and do some housekeeping.

The first code, represented by a byte of = 81, causes a 'set-repeat' function. This signifies that the next note is the start of a piece of the tune which is to be repeated. This is implemented easily by simply pushing the address of the note on the micro's stack.

The second type is the complement of the set-repeat and is the 'repeat' or da-capo—for those programming in Italian. Its main function is to send the interpreter back to the note after the last set-repeat. There is, however, one small problem. Having played the last piece of the tune twice, we do not want to go back again and again. So this code (= 82) modifies itself to = 83 after its execution.

The third function is therefore, = 83 and is called for want of a suitable technical term the already-repeated indicator. This is effectively a no-op, but so that nested repeats may be implemented, it modifies itself back to = 82 for the next time round.

The end of tune byte is any byte with its top bit set which is not one of the above escape sequences. This makes for ease of programming and ensures that the interpreter does not try to play the operating system. The second byte of an escape code is currently a 'don't-care', but once again this allows scope for enlargement.

Note production

The heart of the system is the routine 'NOTE'. This produces a square-wave output of a specified frequency for a specified number of cycles. It consists of two nested loops, one of which delays for the time of one-half of a cycle, the other for the whole length of the note.

The inner loop, when complete, sends a byte to the parallel interface, then increments the byte. Thus, the next time round the least significant bit of the byte will, have changed and will produce the next half cycle. This also means that bits 1–7 will be flipping at progressively slower rates, and provide, simultaneously, the lower octaves, which might be useful.

The coding of 'NOTE' is one of those occasions when the table of instructions for the particular micro has to be studied very carefully. Every clock cycle saved means better high-frequency note resolution, with consequently greater 'musicality'. The example given, for the Intel 8080, may not win any prizes for style but the main delay loop takes only 28 clock cycles.

Using the system available to the author, the CPU is clocked at 2·048 MHz, which gives a loop execution time of around 13·7 microseconds. A quick reciprocate gives a top frequency of more than 73kHz, which sounds a lot, but means that the 'C' two octaves above middle C is represented by a count of around 50. Since the difference in frequency between any two semitones is about six percent it may be seen that this is nearing the upper limit of frequency resolution.

The rest of the program is straightforward, consisting of an initialisation section which sets up things like the stack pointer and the memory pointer to the start of the tune. It also programs the parallel interface so that it may be used as an output device.

The follows the main control loop which fetches sequential bytes of the tune, and having tested for non-notes, sets up the two parameters for the routine 'NOTE' described.

Following the code are the two tables which drive the whole program. Their

(continued on next page)

(*continued from previous page*)

values were decided by experimentation—empiricious rules—with the aid of reasonably accurate scope to calibrate notes at each end of the scale.

Intermediate values may be interpolated by using the relationship $f_n = f_{n-1} \cdot \sqrt[12]{2}$. Any discrepancies may be tuned out individually with the use of an 'ear' or maybe a tuning fork.

Programming a tune

If one can read music, this presents no problem. One simply prepares a table of the notes, with their hexadecimal equivalents and bashes away at the keyboard. On the other hand, an inability to read music can be overcome with half-an-hour's study, and though the process is slow, a reasonable tune may be entered in an hour or so.

Debugging

Having entered the 'score' into the microcomputer memory, the chances are that like all software, it will not be correct first time through. Since a wrong note

within a rapid succession of demi-semiquavers is difficult to identify, it was found that the most successful method of debugging was to change the table of note-lengths so that all possible values produced a long note. Thus the original music may be followed and the location in memory identified and corrected.

Flowcharts listings

The accompanying flowcharts are applicable for any micro, having deliberately left the exact mechanism of parameter establishment to the individual microprocessor. A listing for the 8080 is included as an example. It was prepared on a PDP-11 minicomputer using the MicroAde system of CAP Microsoft, but in all other respects is perfectly standard Intel source mnemonic code.

The program may be seen to occupy # E1 bytes, which in these days of tumbling memory prices is not excessive. Addresses of VO ports are for the SBC 80/10 board from Intel, using the 8255 parallel interface circuit.

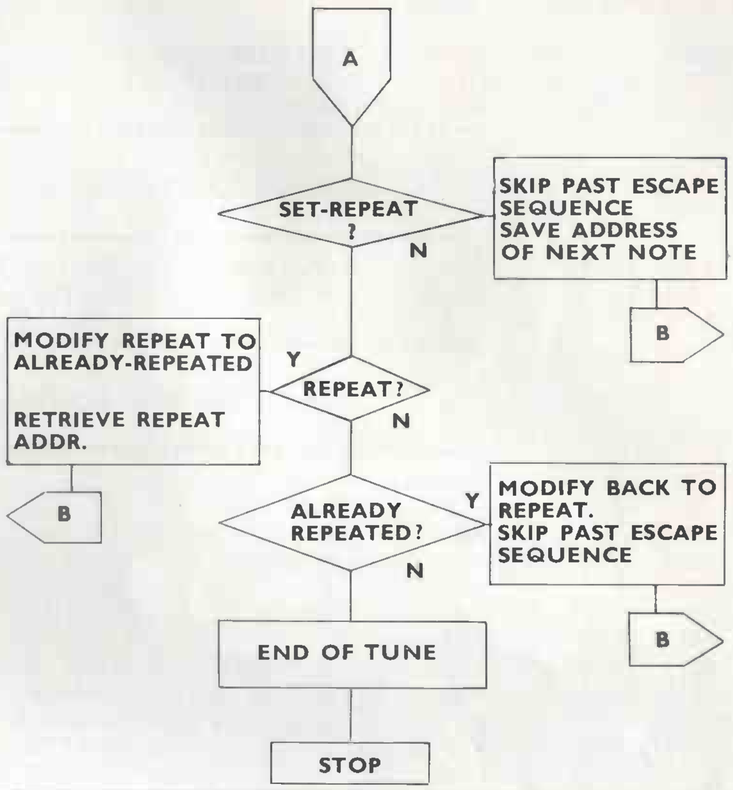
Conclusion

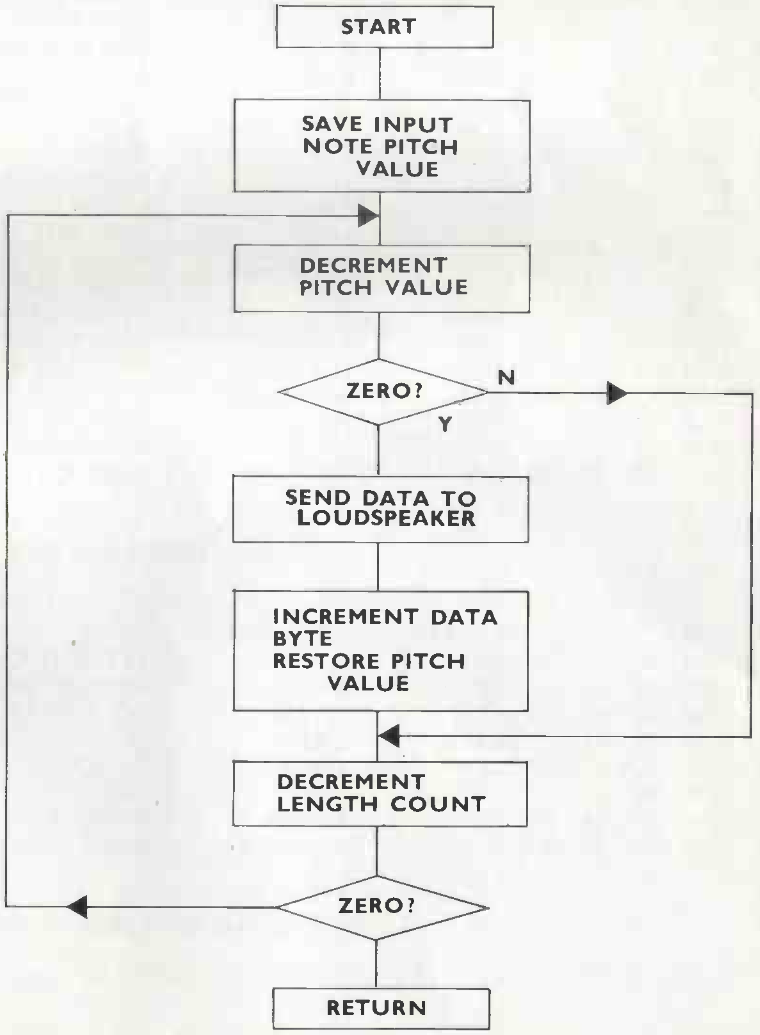
It is hoped that the foregoing will serve to provide some interesting ideas for musically-inclined DIY computer people, and the author Steve Glynn, would be pleased to receive any suggestions for future improvements or extra facilities.

Thanks are due to Rob Lowe for his invaluable assistance, and to CAP Microsoft for the use of its extensive facilities, without which development of the software described would have been a much slower business.

(*continued on next page*)

ESCAPE SEQUENCE PROCESSING





(continued from previous page)

```
MICROADE      RT-11 MACRO VM02-12     30-MAY-78 16:12:22 PAGE 2
TUNES INTERPRETER

0000                   CODE.       A,ZISTS$,50000
5000                   HEX.        2C,MONTOR
5000                   HEX. E5,    PORT
                       ;*************************************************************************
                       ;
                       ;           TUNES - A PORTABLE MUSIC INTERPRETER
                       ;
                       ;*************************************************************************
5000   31 00 50        LXI         SP,#5000        ; LOAD UP THE STACK POINTER
5003   2A 85 50        LHLD        SCORE           ; GET THE START ADDRESS OF THE TUNE
5006   CD 80 50        CALL        INIT            ; INITIALISE THE PARALLEL INTERFACE CHIP
                       ;*************************************************************************
                       ;
                       ;           THIS IS THE MAIN CONTROL LOOP OF THE PROGRAM
                       ;
                       ;*************************************************************************
5009                   NEXNOT:
5009   7E              MOV         A,M             ; GET THE FIRST BYTE OF THE NEXT NOTE
500A   87              ADD         A               ; PUT THE TOP BIT INTO CARRY
500B   DA 34 50        JC          ESCAPE          ; IF SET, THEN IT'S AN ESCAPE CODE
500E   87              ADD         A               ; NEXT BIT INTO CARRY
500F   DC 5C 50        CC          SPACE           ; IF SET THEN A WAIT BEFORE THE NEXT NOTE
5012   1F              RAR                         ; DIVIDE BY TWO
5013   11 92 50        LXI         D,PITCHT        ; LOAD ADDRESS OF PITCH TABLE
5016   06 00           MVI         B,O             ; CLEAR B
5018   4F              MOV         C,A             ; TO GET A WORD IN BC
5019   EB              XCHG                        ; SWAP DE & HL
501A   09              DAD         B               ; SO THAT WE CAN ADD THE OFFSET
501B   EB              XCHG                        ; SWAP BACK AGAIN
                                                   ; DE IS NOW A POINTER TO THE PITCH VALUE
501C   1A              LDAX        D               ; GET IT IN BC
501D   47              MOV         B,A             ;
501E   13              INX         D               ; . . . BOTH BYTES
501F   1A              LDAX        D
5020   4F              MOV         C,A             ; THERE . . .
5021   23              INX         H               ; POINT AT THE NEXT BYTE OF THE NOTE
5022   5E              MOV         E,M             ; FETCH IT
5023   E5              PUSH        H               ; SAVE THE TUNE POINTER
5024   21 87 50        LXI         H,LENGTB        ; POINT AT THE TABLE OF LENGTHS
5027   16 00           MVI         D,0             ; CLEAR THE D REG
5029   19              DAD         D               ; IN ORDER TO GET AT THE LENGTH VALUE
502A   5A              MOV         E,D             ; CLEAR E WITH D
502B   56              MOV         D,M             ; AND GET A WORD VALUE IN DE
502C   CD 68 50        CALL        NOTE            ; PRODUCE THE SOUNDS
502F   E1              POP         H               ; RESTORE THE SAID POINTER
5030   23              INX         H               ; ADVANCE IT
5031   C3 09 50        JMP         NEXNOT          ; AND PLAY ON. . .
                       ;
                       ;
                       ;           THIS CODE PROCESSES ESCAPE SEQUENCES
                       ;
                       ;
5034                   ESCAPE:
5034   1F              RAR                         ; RETRIEVE ORIGINAL BYTE
5035   FE 81           CPI         #81             ; PROCESS POSSIBLE ESCAPE CODES
5037   CA 47 50        JZ          SETREP          ; SET REPEAT SEQUENCE
503A   FE 82           CPI         #82             ;
503C   CA 4D 50        JZ          REPEAT          ; DA CAPO
503F   FE 83           CPI         #83             ;
5041   CA 54 50        JZ          ALLREP          ; CLEAR REPEAT MARKER
5044   C3 2C 00        JMP         MONTOR          ; IF WE ARRIVE HERE, IT'S THE END OF THE TUNE
                       ;
                       ;
                       ;           THIS IS THE SET-REPEAT FUNCTION
                       ;
                       ;
                       ;
```

(continued on next page)

(continued from previous page)

```
5047                SETREP:
5047  23            INX      H          ; POINT AT NEXT REAL NOTE
5048  23            INX      H          ;
5049  E5            PUSH     H          ; SAVE THE ADDRESS ON THE STACK
504A  C3  09  50    JMP      NEXNOT     ; CARRY ON. . . .
                    ;
                    ;
                    ;        THIS IS THE REPEAT FUNCTION. IT RESTORES THE ADDRESS SAVED
                    ;        BY THE ABOVE ROUTINE, AND MODIFIES THE BYTE THAT CAUSED IT
                    ;        TO BE EXECUTED.
                    ;
                    ;
                    ;
504D                REPEAT:
504D  F6  01        ORI      1          ; SET THE BOTTOM BIT AS THE ALREADY-REPEATED
                                        ; INDICATOR
504F  77            MOV      M,A        ; REPLACE NOTE
5050  E1            POP      H          ; GET BACK THE ADDRESS OF NOTES TO BE REPEATED
5051  C3  09  50    JMP      NEXNOT     ;
                    ;
                    ;
                    ;        THIS ROUTINE REVERSES THE EFFECT OF THE ABOVE, RESETTING THE
                    ;        BYTE IN MEMORY, FOR POSSIBLE LATER EXECUTION
                    ;
                    ;
                    ;
5054                ALLREP:
5054  E6  FE        ANI      #FE        ; CLEAR ALREADY REPEATED INDICATOR
5056  77            MOV      M,A        ; STASH THE NOTE
5057  23            INX      H          ; POINT AT NEXT NOTE
5058  23            INX      H          ;
5059  C3  09  50    JMP      NEXNOT     ;
                    ;
                    ;        THIS ROUTINE PRODUCES A PRESET DELAY BEFORE THE PRODUCTION
                    ;        OF THE NEXT NOTE
                    ;
                    ;
                    ;
                    ;
505C                SPACE:
505C  47            MOV      B,A        ; SAVE THE ACCUMULATOR, NOT THE FLAGS
505D  11  00  04    LXI      D,#400     ; LOAD A PRESET DELAY
5060                SPACE1:
5060  1B            DCX      D          ; DOUBLE DECREMENT
5061  7A            MOV      A,D        ;
5062  B3            ORA      E          ; AND TEST
5063  C2  60  50    JNZ      SPACE1     ; LOOP BACK
5066  78            MOV      A,B        ; RESTORE THE ACCUMULATOR
5067  C9                     RET        ; ELSE RETURN FOR MORE
                    ;
                    ;        'NOTE'
                    ;        THIS ROUTINE IS THE ACTUAL SOUND PRODUCER.
                    ;        IT CONSISTS OF TWO NESTED LOOPS, WHICH DECREMENT INPUT
                    ;        PARAMETERS, THUS PROVIDING A GIVEN NUMBER OF CYCLES
                    ;        AT THE REQUIRED FREQUENCY
                    ;
                    ;
                    ;
5068                NOTE:
5068  60            MOV      H,B        ; SAVE THE PITCH VALUE
5069  69            MOV      L,C        ;
506A                LOOP1:
506A  0D            DCR      C          ; DECREMENT LOW ORDER BYTE OF PITCH
506B  C2  77  50    JNZ      LOOP2      ; IF NOT YET ZERO, SKIP NEXT BIT
506E  05            DCR      B          ; ELSE DECREMENT THE HIGH BYTE
506F  F2  77  50    JP       LOOP2      ; IF THAT'S POSITIVE OR ZERO SKIP
5072                SOUNDS:
                    ;        THIS CODE ACTUALLY CHANGES THE LEVEL AT THE LOUDSPEAKER
                    ;
```

(continued on next page)

(continued from previous page)

```
5072  2F          CMA                    ; ELSE FLIP SOME BITS
5073  D3 E5       OUT      PORT          ; SEND TO LOUDSPEAKER
5075  44          MOV      B,H           ; RESTORE PITCH VALUE
5076  4D          MOV      C,L           ;
                  ;
                  ;        SECOND PART OF MAIN TIMING LOOP
                  ;
                  ;
5077              LOOP2:
5077  1D          DCR      E             ; DECREMENT LOW BYTE OF LENGTH
5078  C2 6A 50    JNZ      LOOP1         ; RETURN TO TOP OF MAIN LOOP
507B  15          DCR      D             ; DECREMENT HIGH BYTE
507C  C2 6A 50    JNZ      LOOP1         ; RETURN IF NOT DONE YET
507F  C9                   RET           ; ELSE WE'VE FINISHED
                  ;
                  ;
                  ;
                  ;
5080              INIT:
                  ;        THIS ROUTINE INITIALIZES THE PARALLEL INTERFACE
                  ;
                  ;
                  ;
5080  3E 89       MVI      A,#89
5082  D3 E7       OUT      #E7
5084  C9                   RET
                  ;
                  ;
                  ;
                  ;        THIS WORD CONTAINS THE START ADDRESS OF THE TUNE TO BE
                  ;        PLAYED
                  ;
                  ;
5085  00 60       SCORE    DW.   60000
                  ;
                  ;
                  ;
                  ;        THIS IS A TABLE OF BYTES, CORRESPONDING TO THE TEN POSSIBLE
                  ;        NOTE LENGTHS. THESE START AT HEMI-DEMI-SEMI-QUAVER AND GO
                  ;        THROUGH TO MINIM, WITH DOTTED NOTES IN BETWEEN
                  ;
                  ;
                  ;
5087              LENGTH:
5087  07 0A 0E    .BYTE    7,10.,14.,21.,28.,42.,56.,85.,114.,171.,228.
508A  15 1C 2A
508D  38 55 72
5090  AB E4
                  ;
                  ;
                  ;
                  ;        THIS IS THE TABLE OF NOTE PITCHES.
                  ;        THEY HAVE BEEN SET UP FOR A CLOCK RATE OF 2.048 MHZ, AND GIVE
                  ;        MIDDLE C AS 256 HZ, THREE OCTAVES ARE SET OUT HERE, WITH
                  ;        TWELVE NOTES IN EACH, FROM 128 TO 1024 HZ INCLUSIVE.
                  ;
                  ;
5092              PITCH:
5092  00 CA 00    .BYTE    0,202.,0,191.,0,180.
5095  BF 00 B4
5098  00 AA 00    .BYTE    0,170.,0,160.,0,151.
509B  A0 00 97
509E  00 8F 00    .BYTE    0,143.,0,135.,0,127.
50A1  87 00 7F
50A4  00 78 00    .BYTE    0,120.,0,113.,0,107.
50A7  71 00 6B
50AA  00 65 00    .BYTE    0,101.,0,95.,0,90.
50AD  5F 00 5A
```

(continued on next page)

(continued from previous page)

```
50B0 00  55  00  .BYTE      0,85.,0,80.,0,76.
50B3 50  00  4C
50B6 00  47  00  .BYTE      0,71.,0,67.,0,64.
50B9 43  00  40
50BC 00  3C  00  .BYTE      0,60.,0,57.,0,54.
50BF 39  00  36
50C2 00  32  00  .BYTE      0,50.,0,48.,0,45.
50C5 30  00  2D
50C8 00  2A  00  .BYTE      0,42.,0,40.,0,38.
50CB 28  00  26
50CE 00  24  00  .BYTE      0,36.,0,34.,0,32.
50D1 22  00  20
50D4 00  1E  00  .BYTE      0,30.,0,28.,0,27.
50D7 1C  00  1B
50DA 00  19      .BYTE      0,25.
                 ;
                 ;
     0001'       .END

MICROADE     RT-11 MACRO VM02-12   30-MAY-78 16:12:22 PAGE 24
SYMBOL TABLE

ALLREP     5054        ESCAPE  5034  INIT    5080  LENGTB  5087  LOOP1   506A  LOOP2    5077
MONTOR#    002C        NEXNOT  5009  NOTE    5068  PITCHT  5092  PORT  # 00E5  REPEAT   504D
SCORE      5085        SETREP  5047  SOUNDS  5072  SPACE   505C  SPACE1  5060  ZISTS$#  5000G
.ABS.      50DC 00
           0000 01
ERRORS DETECTED: 0
FREE CORE: 10574. WORDS

U:TUNES2[2],TUNES2[-]=ZIP,U:TUNES2
```

ICS TRAINING SYSTEM

WHEN one is given a microprocessor chip and told "There's your computer" it is like being given a crankshaft and told to build a motor car. The next questions are invariably "What do I do now?" and "How do I learn?".

Review by Boris Sedacca

THERE is an ample supply of development systems supporting most CPU chips available on the market and the dividing line between those and what some suppliers will call a training kit or system is very fine indeed.

There are various means by which you can learn about computers. One is to buy a kit and to solder each component on to a printed circuit board. That might provide the cheapest solution for the newly-converted hobbyist, but extensive testing may be required once the board has been assembled, assuming, of course, that no major mishap has occurred during the soldering stage.

At the other extreme one may buy a ready-configured system supporting a high-level language, normally Basic, with some means of mass storage such as cassettes or mini-floppy disc drives, a keyboard for input and a VDU, or a television set with special adaptor—a popular trend— for output.

With personal computer systems at this level of sophistication, the hobbyist is more concerned with his applications, i.e., what he puts into the sytem and what comes out of it, rather than what goes on inside the "guts" of the machine itself.

To illustrate the point, the Basic language is available on various makes of microcomputer "mainframes", and apart from some differences in syntax, is generally the same across those machines.

Their internal circuitry will vary substantially, however, let alone their use of different instruction sets for different CPUs.

The training system I am reviewing fits somewhere in between. It is a board without interfaces to applications-orientated peripherals, limited user memory, and an educational monitor stored in read-only memory (ROM).

This, however, is only one-half of a training system; of paramount importance are textbooks, manuals and documentation provided with the hardware.

The approach a supplier of training systems will take depends largely on his background, and the student, user or hobbyist should establish this from the beginning.

The student may, for instance, wish to learn about the electronics side of microprocessors with a view eventually to designing and putting together a board from each component. Alternatively, he might wish to take the assembled board for granted and concentrate on learning the instruction set of that particular microprocessor.

As the latter approach is less likely to require prior knowledge of electronics, we look at a system which falls roughly within this category for the purpose of this review.

My experience

The ICS training system requires little from its students by way of electronics experience. I used a primitive soldering iron to connect some leads from the power supply to three edge connector pins, threw the mains switch and the LEDs (light-emitting diodes) came to life.

The system was ready to accept its first instruction showing a hexadecimal address. The system is boot-strapped as soon as the power is turned on with the aid of an educational monitor contained within 1,024 bytes of electrically-erasable PROM memory.

I pressed some keys on the keyboard and the system reacted with results which were, naturally enough, unintelligible to me. The next step was obviously to start reading the manuals from the beginning.

Incidentally, the first manual does not provide the instructions on soldering and edge connection, which is the most logical place for them to be, but they are in an appendix at the rear of the second manual. In fact, the student is not even guided to this section so I called Computer Marketing and asked for instructions.

The first chapter gives a fair introduction to computer concepts such as hardware organisation, conversion from hexadecimal, and access to instructions stored in memory.

This approach is commendable in that the student is encouraged to learn the use of the microprocessor instruction set without concerning himself unduly with the complexities of logic gates, timing compatibilities and pin assignments from the beginning, as is the case with other textbooks on the subject.

Although the first chapter proceeds through the first steps meticulously to let the student grasp the implications of each key press, I was disappointed to see that this style was not maintained.

Mental block?

Some mistakes occur, however, in the documentation—inevitably off-course—and a student might well find himself facing a mental block if he is not made aware of these errors.

As I have detected such errors I have communicated them to Computer Marketing, who will no doubt act upon them. The main problem is one of communication, however, and it is sometimes a little difficult to understand what the engineer is trying to convey.

A case in point arises is chapter four, entitled "The other registers"—other than the accumulator, that is. At that stage I felt that documentation could have been more extensive and have shown the status of the registers where the detailed explanations of the program codes are given. In fact, their existence is rarely acknowledged.

Unfortunately, the student is not even encouraged to make notes for himself and as the facilities for the inspection of registers are amply provided for, it is difficult to understand why the author has chosen to ignore them instead of making adequate use of them.

I make this criticism because a beginner, for which the manuals have

(continued on next page)

(continued from previous page)
supposedly been written, might well not be able to work out this for himself, and a true training system ought not to take previous knowledge for granted.

Criticisms aside, the manuals provide a sound basis by which to help a student up his learning curve. The documentation is made up of photo-copied material which has mostly been typed manually, and whether by design or by dictates of economy, having the documentation in loose-leaf binding makes good sense.

Unfortunately, however, some sheets are not well reproduced in the copying process and parts of a page are occasionally cut off, or the text is slanted. Some characters are obliterated by punch-holes and ICS might consider using narrower-width text.

As the ICS training system is ready-assembled it is not necessary to give a detailed component (or chip) count normally given with a kit.

Instead, we look at the principal component groups of the board to give an idea of their particular functions.

The 8080 CPU sends high-or low-voltage signals via the 8228 systems controller along the data bus. The monitor program is contained within four × 256-byte D454 read-only memory (ROM) chips.

1024 bytes of user memory are provided in four pairs of D5101 random-access memory (RAM) chips, each pair providing 256 bytes of read/write memory. Each chip does not, in fact, provide 128 bytes of RAM but 256 half-bytes—or "nibbles" if you really like buzzwords.

Three ports

Pulses are provided by the 8224 system clock and its crystal to the 8080 CPU and the rate at which the pulses are generated has a direct bearing on processor cycle speeds.

The 8255 programmable peripheral interface provides three ports for 8-bit, parallel data transfer. When port A or port B is programmed to input mode 1—typically for a high-speed paper tape reader— it uses some bits in port C to handle the strobe and to give an interrupt signal to the 8080, and then responds with an acknowledgement to the input device when the data has been accepted.

All the aforementioned chips are friction-mounted on DIL sockets which allow for easy replacement, should the need arise, of faulty chips, although NEC chips are believed by a major mainframe manufacturer to be twice as reliable as those of other semiconductor manufacturers.

Indeed, it would be technically feasible to replace the 8080 CPU with the 8085 or Zilog Z80 CPU, although they would merely emulate the performance of the

8080. Before attempting this conversion, however, the student would be well advised to check that certain requirements, such as power supply, are complied with.

The rest of the chips, including the 8212 I/O port, the 2155 (NEC 4155) address decoder, and those making up the direct memory access (DMA) channel are soldered permanently to the board itself.

The 8255 peripheral interface provides three I/O ports of 8-bits each. In fact, only one port is available to the user unless he wishes to disconnect the keypad which uses a 3 × 8 matrix array, effectively occupying the eight bits of port A and three bits from port B. This services 24 keys, the 25th being the reset key, "RST", which has its own direct connection to a pin on the 8255.

Assuming the student becomes proficient enough in the technology to require more than the single available port, ICS has announced an interfacing system to be made available "soon".

This will require some soldering work and possibly some changes to the ROM monitor, although the company has not made itself clear on this point and the conversion might be chargeable. Otherwise the two boards are connected by a simple edge connector and ribbon cable.

The interface board contains two extra 8255 I/O chips and space for an extra 1,024 bytes of RAM, although it is not clear whether the address decoder of the training system will still be used.

In addition, the board supports "priority" vectored interrupt, an 8253 timer chip for real-time programming, and interfaces for cassette, Teletype and RS232C terminals via connector blocks.

A motor will be provided for control by software of pulse-width modulation, or by voltage level with digital-to-analogue conversion. A loudspeaker will allow for the keying-in and replaying of music.

All this might well sound dubious but I have heard computer music before, although I wonder whether all these applications are possible on the system and how much hand-holding the course notes actually provide.

Perhaps I am too cautious and I am told that some of these boards have already been ordered prior to their availability. I do not believe, however, that the training system is being all things to all men. It is an excellent training system and I feel that ICS can comfortably live up to such a claim.

Computer Marketing seems to have taken to the training system seriously, claiming to have ordered 1,000 training systems and 250 interacting systems to be delivered over the next year, at a total value of around £450,000.

Technical details

8080A microprocessor
8-digit hexadecimal display
Educational monitor program
1,024 bytes of user memory £349
Keyboard with 25 keys
650 pages of text and diagrams (2 manuals)

Optional power supply 5 & 12 volt £69

Available from Computer Marketing, sole U.K. distributors for ICS training courses.

The hexadecimal keypad is constructed from high-quality key mechanisms for maximum reliability and contains 25 keys.

The 16 keys from 0 to 9 and A to F correspond to hex values which are decoded by the board circuitry into binary signals when a key is depressed. Some of them serve a dual function when certain control keys are depressed first. The control keys make up the other nine keys on the pad.

Any register content within the CPU may be addressed directly at any instruction stage of a program execution and displayed by depressing the "REG" key, followed by the key relating to the required register.

Keys A to F refer to single-byte (8-bit) registers within the CPU. Keys 8 and 9 also refer to registers H and L (for high and low) respectively.

As the 8080 CPU has a 16-bit addressing capability—this means that 64K bytes may be addressed—certain registers are arranged in pairs to contain 16-bit addresses. They are register pairs B & C, D & E, and H & L.

By depressing keys "ADDR", "D" and "MEM", for instance, the LEDs will display an address to which the register pair is pointing, the letters DE (the register pair), and the contents of the address location.

The above addressing method is also available for displaying the address of the stack pointer by depressing "ADDR", "1" (or "P") and "MEM". Again the LEDs will display an address, the letters SP (for stack pointer) and one-half of the return address held in a single byte location. By depressing the "NEXT"

(continued on next page)

(continued from previous page)

key, the address is incremented to display the other half of the return address.

As the return address is in many cases more relevant in itself than the addresses at which the stack pointer is contained, Key "2" (or "T") may be depressed instead of Key "1" to show the full return address.

Where a student wishes to see intermediate results of a program execution without having to "STEP" through each instruction, particularly where a series of loops are encountered, a break key is provided, "BRK", to stop execution of the program at any stage.

Up to eight break-points are permitted by the monitor software. As an option, the brêak-point may be set to cut in only after an instruction has been executed a specified number of times.

For example, keys "ADDR", "8", "2", "1", "0", and "BRK" may be depressed to set the break-point at the instruction contained within address location 8210. If this is followed by depressing keys "2" and "4", for instance, the instruction will be executed 24 times.

The display will then show "8210", "BP' and "24". If the re-set key, "RST" is depressed, followed by "RUN", the display will then show "8210" and the instruction keyed in to that address. A look at the registers will show how they have been affected.

The breakpoints may be erased by depressing "CLR" and the display will show "0000", "BP" and "00".

All the keyboard routines are controlled by the monitor program. The keyboard features "de-bouncing". What this means in effect is that, with calculator-type keypads, a very common problem occurs on some occasions where, for instance, key "A" is depressed, a whole string of "A"s will be entered.

The "de-bouncing" feature minimises the risk of incorrect keying by locking-out any other entry until a key is released.

The "STEP" key is an essential feature for any training kit. It allows one to check the result of instructions one step at a time as a program is being executed.

Programs are entered through the keyboard in HEX code. The monitor presents Hex address 8200 for a program instruction to be entered.

Any memory location may be addressed and its contents desplayed by depressing "ADDR" and its address. Instructions may then be entered by depressing "MEM" and the Hex code for the address. This random addressing capability is essential for program modules and subroutines. There is no effect, however, on protected areas addressed by the ROM monitor programs.

When an instruction has been entered against a memory location, "NEXT" may be depressed to display the following location for the next instruction to be entered.

Where an incorrect instruction has been entered, the correct keys may be depressed again. Only that instruction displayed opposite the memory location will be stored in memory when the "NEXT" key is depressed. Completed programs may be verified after they have been entered by the use of this correction facility.

They could call him Lord Mike

The founder of the Amateur Computer Club is one of the real pioneers among computer hobbyists. He has seen the organisation grow from 50 members to 20 times that number in five years.

MIKE LORD has probably done more for the computer hobbyist than anyone else in this country, though he would be the last person to say so. Besides being one of the real pioneers—how many other amateurs have been building computers for 10 years?—he is founder of the Amateur Computer Club and the designer of two computers which even today cost less to build than the nearest commercial alternative.

Like most other amateurs, Lord is not employed in the computer industry but has a professional interest in electronics, working as a telephone systems engineer for ITT. His interest in computers was aroused in 1968 on a visit to the States, where the hobby had already reached recognisable proportions. He returned with a number of ideas and a handful of components and began building his first computer.

Time of rapid development

The result of his efforts was made public around 1974 as the Weeny Bitter, a very basic computer which used TTL circuitry and had a maximum memory addressing capability of 256 bytes.

Despite its severe limitations, the Weeny Bitter found favour with ACC members and others as a way of getting to grips with a working computer at a reasonable cost. The design was formalised and a handbook produced by Bear Systems—now New Bear—and the computer was built, and continues to be built, in relatively large numbers.

It was perhaps unfortunate that the Weeny Bitter appeared at a time of particularly rapid development in the industry, for it was soon made technically obsolete by the advance of micro-processors. Even so, there are still ACC members who feel it is a worthwhile project as a way of learning the basics of computers. Some have even been recorded as saying that they find micro-based systems boring, and that they prefer projects with plenty of wiring.

Meanwhile, on behalf of the rational majority, Lord went to work again, and in April, 1977 the first details were published of the 7768 microcomputer. Once again, cheapness and availability of parts were leading criteria in the design, but the 7768 is a much more sophisticated and potentially more power-ful machine.

To quote from the club newsletter: "The 7768 was designed to meet the need for a low-cost, introductory micro-computer which the beginner can build easily and use without having to buy expensive peripherals such as a teleprinter and VDU, but which can be expanded, as the constructor's time and pocket permit, to give a system comparable to any of the 8-bit microcomputer systems available.

"In its basic form, consisting only of a single board and a rudimentary control panel, the 7768 can be built for around £50, or less if the constructor has a reasonably deep junk box."

Has some kind of appeal

Thus the basic model has, again, a 256-byte memory and provision for data input by means of switches but can be expanded to accept up to 64K bytes of memory and peripherals such as a VDU or cassette recorder. Once again, New Bear has adopted the system and is producing the PCBs, and the system is still being developed.

The 7768 seems to have as much appeal to ACC members as the Weeny Bitter before it, but how many will find a use for its full potential is open to question. Browsing through the club newsletter,

(continued on next page)

(continued from previous page)

one gets the impression that members do not do very much with their computers—they just build them. Of course, there is no reason why a computer built for interest or amusement should have any useful function,—but to anyone used to commercial computing it seems a slightly bizarre phenomenon.

So what sort of animal is the computer hobbyist? The Amateur Computer Club was founded in 1973 with around 50 members. Today it has about 1,000, and looks like growing fast. Its members, according to general secretary Robert Warren, tend to fall into three main categories.

Looking into the unknown

There are the 15-to 25-year-olds who have been brought up on computers, are probably studying them at school or university, and also want one of their own with which to work. Second, and probably the largest group at present, are older members, mostly with a background in electronics but no direct connection with computing. They have seen computers grow around them, so to speak, and have taken to the hobby mainly as a way of investigating unknown territory. A third, and much smaller group, consists of those with an interest in using computers in their work—teachers, businessmen, doctors, and at least one veterinary surgeon.

Mike Lord's computer

On the whole, though, the indomitable spirit of amateurism reigns supreme. Comments appear in the newsletter such as: "I shall continue to write algorithms in PASCAL, even though I can't run them".

Tracking down cheap components seems as much an enjoyable challenge as a financial necessity, and members exchange snippets of information on likely secondhand sources—a stall in St Albans market is reputed to be one treasure trove.

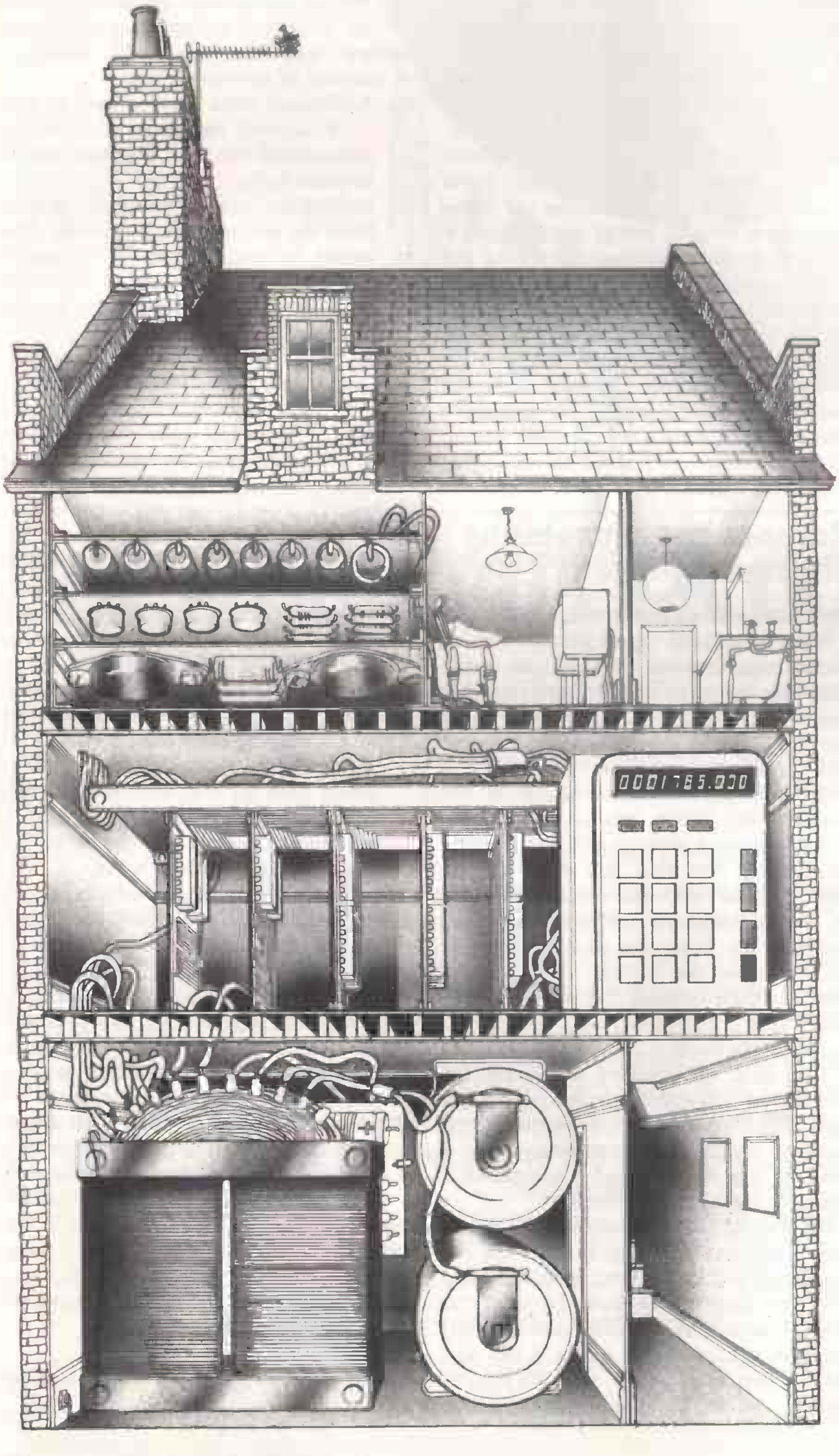
Some construction methods would make a professional goggle, like the member who built a plotter out of Meccano and then programmed his computer to generate random patterns by using the inevitable free play in the mechanism.

Popularity is a problem

Incidentally, not all members' computers are shoe-box sized. There are at least a dozen mainframes, including such veterans as an ICT 1301 and Elliott 803, chuntering away in sheds and garages across the country.

Even the club's increasing popularity poses a problem. At present the club is a fairly casual collection of individuals with a number of common interests, which on the whole do not include administration. It is already under pressure to become regimented and organised, and that pressure is likely to grow.

How to wire your home by the easy route



WELL, how do you control your home with a computer? The average house contains many static electrical and electronic appliances which could benefit from intelligent control.

A microcomputer can, in theory, easily control your central heating, according to time and climate. It can turn lights on and off, operate your hi-fi, act as an alarm clock, a shopping list memo pad, oven timer—and still have time to run games programmes or computer calculations.

Wiring all these appliances can cause headaches. So let us consider the requirements for a system, which I call the Simbus Scheme, which is practical to implement.

There are nine main requirements. The system should not require many wires; connectors should be cheap; individual devices should be removable without stopping the whole system; each appliance should have its own address, which it keeps, even when moved to a different site; the protocol must be simple and standardised; extending the system should not necessitate modifications to existing devices; appliances which do not contain low voltage d.c. to operate logic should not require individual interface power supplies; on computer failure, each appliance should revert to manual control; and it should be able to control analogue devices with only one DAC/ADC.

All requirements in one system

All these requirements can be met with a single system, the technical details of which have been described elsewhere. It is possible to devise a system which fulfils all these requirements using only two wires and earth.

One wire (the red) is a serial data wire, down which an address is sent, followed by a data byte or bytes. The other (black) is a bus-free line, which is at a positive voltage when the bus is free, and at ground when busy.

At various points around the house, the cables are brought out to three-pin sockets—I recommend 3-pin DIN as being cheap and satisfactory with added advantage mentioned later in which case 1= Data, 2=Ground, 3=Bus free.

Each device interface presents a high impedance to the bus except when sending data on to the bus, in which case, it shorts the line to earth. Data is sent "blind", so that the system does not wait for a

(continued on next page)

(continued from previous page)

reply, which might hold up the system if an appliance had been removed.

It is possible, however, in some circumstances for the computer to test for the presence of a particular device.

Each bus interface card has a row of eight dual-in-line switches on which addresses are selected. This permits 256 devices on the bus, although it is possible to use one address to access a further 256 and so on.

There are six types of peripheral interface card, and each has its own protocol, which is defined absolutely and does not change as the system grows or improves.

Appliances which do not often need attention can be driven from rechargeable nickel-cadmium cells which are trickle-charged off the bus-free line. There are some restraints on this facility.

It is possible to measure analogue voltages by arranging that each analogue device has the capability to pull the data line down to the analogue voltage for a set time, during which it may be sampled by the microprocessor. Judicious control of the bus-free line prevents digital devices interpreting it as spurious data.

The next important thing to discuss is the degree of standardisation which can be achieved in the bus interface modules. I consider that devices for connection to the bus may be categorised as six types:

On/Off devices

1A. One address turns on; a second turns off.

1B. The address set a monostable which turns on for a pre-set time.

In both these cases the output provided by the interface is simply a relay contact, which may be used to turn on or off any conventional appliance.

2. Data Receive

2A. Serial output.

2B. Parallel output.

These devices might include Teletype printers, VDUs and the like. Serial output could be 5V logic level, or RS232 or 20mA. A modification permits the address word to be sent at, say, 300 baud, and the data at a user-selectable speed. Any number of data words may be sent after the address word, provided that bus free is held low until after the last data word.

3. Send-on-Request

3A. Serial input.

3B. Parallel input.

On receipt of the address word, one or more data words are sent to the computer. I would recommend that a special version of 3B. should be produced which includes a 7-seg to BCD converter, thereby permitting direct interface of digital clocks and calculators. A digital clock somewhere on the system is essential for most applications.

4. Unsolicited Send

4A. Serial input.

4B. Parallel input.

This type of interface is for devices which generate data at unpredictable intervals, for example keyboards. When information becomes available, 'bus free' is pulled low, and as this has not been done by the controlling computer, this causes an interrupt to be generated at the computer to a routine which awaits an address word from the device, followed by one or more data words. These devices pose a number of problems, in particular:

a. If other devices on the system occupy the bus for long uninterrupted periods—e.g. printing a whole page on a Teletype with only one address word—then either data will be lost, or buffering must be provided at the interface.

b. If a number of type 4 devices are on the bus, they may all try to grab the bus as soon as it becomes free. A solution, if only a small number of devices are on the bus, is to arrange that each interface times 'bus free', before grabbing it. Devices with higher priority have to wait shorter times than those with lower priorities. If a large number of these devices are sending lots of data, then it is possible for two devices to grab the bus simultaneously, if it has been free for longer than either timing period. If this proves to be a problem, and it will only rarely, then it is arranged that all type 4 devices (and the µP interface) pull bus free low for 10mS and then release it, starting their timers as they do so. In this case the priority structure described above still works.

Remote measurement of quantities

5. Analogue Transmit

Many domestic applications involve remote measurement of analogue quantities–e.g. temperature, light level–and it is desirable to cope with all of these using one ADC. This may be done as follows: The analogue interface recognises a serial address on the data line in the normal way. This sets a monostable which enables a voltage-follower op-amp which pulls the data line down to the voltage to be measured for 10mS. After, say, 5mS to allow the level to stablise at the computer, the computer enables the ADC which samples the voltage. The analogue interface then releases the data line and 'bus free'.

6. Analogue Send

This kind of interface is needed for controlling volume controls, light dimmers and the like. The principle is the same as for 5., but in reverse, using a DAC at the µP end, and a sample-and-hold op-amp at the remote end.

Finally, though it doesn't really count as a peripheral interface, there is the all-important microcomputer interface. The 8-bit bus connects to both ports of the UAR/T which drives (and receives from) the data line, and also to the ADC and DAC if devices of type 5. or 6. are present on the system. A single output pin is required from the microcomputer for each of the following:

i Bus free.
ii Enable ADC.
iii Enable DAC.
iv Select UAR/T output.
v Select UAR/T input.

Also the following inputs should be provided:

i Interrupt (when Bus free goes low unexpectedly).
ii UAR/T Received Data Available.
iii UAR/T Transmit Buffer Empty.
iv ADC Conversion Complete.

I assume that provision of +5V at the microcomputer is no problem. Many of the remote appliances—particularly those of an electrical rather than electronic nature—will not have +5V available. A number of solutions exist.

i Additional wires may be used to pipe +5V around the system—I do not recommend this as it degrades signal-to-noise ratio.

ii Individual power supplies derived from the mains at the appliance. This would be appropriate for a central-heating boiler or a larger television set, where the unit could be housed inside.

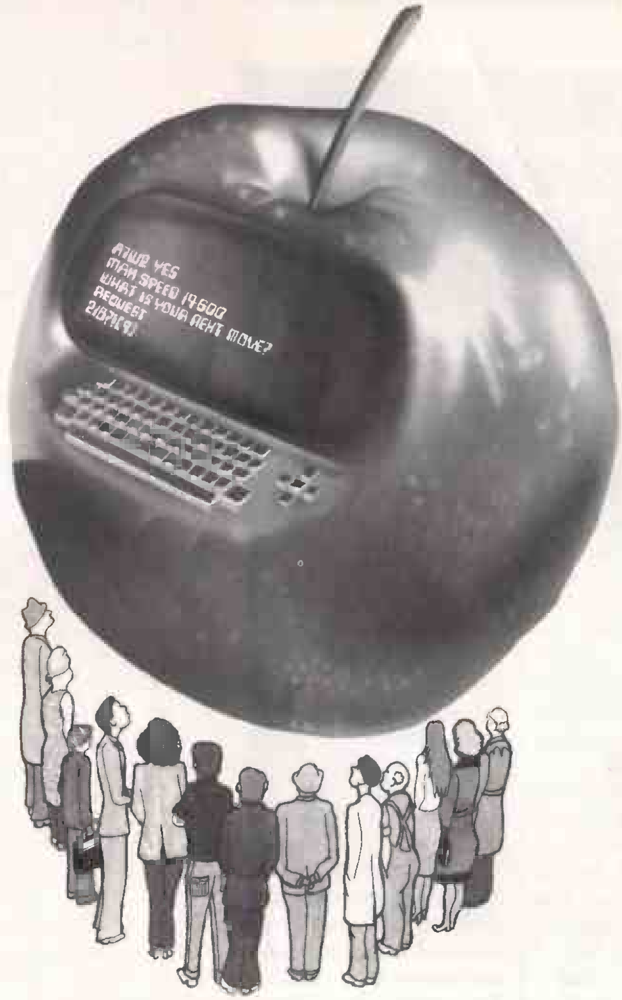
iii A fixed mains-driven power supply, supplying power to two extra pins on the DIN socket. This would be appropriate for units which (say) just switch lights on and off. In this case, pin 5 of the DIN socket has +5V provided on it, and pin 4 has −12V, if needed for the cheaper UAR/TS. The interfaces which require these have 5-pin plugs, which can plug only into the 5-pin sockets with power. Those sockets which do not have power laid on have only three pins, as do those plugs which do not need power.

iv If the system pull-up resistors pull up to a voltage greater than 5 volts, then bus free can be used to trickle-charge NiCd cells at the interface. In this case, however, protection circuitry is needed to protect the input to the logic from exceeding 5 volts.

There are, of course limitations to such a simple scheme. In particular, the 'wired-or' principle imposes severe restraints on speed, when capacitance of lines is high. Also, on the subject of speed, while data can be sent at any rate, the first (address) word must always be sent at the same rate. A rate of 300 baud would seem a sensible compromise. Also, to state the obvious, the bus cannot do two things at once. Consider this restraint carefully if you have type 4 devices on the system.

Unfortunately the interface modules described are not yet available commercially. Sketch diagrams for them, however, have been published—*Electronic Engineering*, May 1978—for anyone keen enough to make their own.

The author is Chris Mollor of Modular Technology.



WIN A COMPUTER COMPETITION

To mark the launch of Practical Computing we have decided to give away a computer.

It's the Apple II which we have reviewed on page 11.

All you have to do is think of a novel application which you would run on the Apple II.

It could be a new game that you have invented – or it could be an idea for a business application of some description.

We require a description of not more than 3,000 words on your application detailing the project and putting forward your solution.

And just to make it more inviting we shall also be offering TEN runner-up prizes of £25 for applications ideas which we shall publish in Practical Computing in later issues.

The closing date for the competition is September 30, 1978.

To enter, however, you must fill in the form at the bottom of this page.

We shall then post to you the official entry form which must accompany all articles.

Don't delay – post the form today.

Please send me the official entry form
(I enclose a stamped-addressed envelope).

Name ______________________________

Address ______________________________

Now post to: Practical Computing, 2 Duncan Terrace, London, N.1.

It's all Byte but it's worth barking about

STEP INSIDE The Byte Shop in Gants Hill, Ilford and you could be forgiven for thinking you were somewhere else. Apart from the bookstand and the till, it could be the computing department of a university or technical college, with hunched figures concentrating at VDUs, others deep in books or earnest conversation.

The one dynamic figure in the tableau is Vince Coen, pounding from one group or individual to another, to the telephone, occasionally to the till. Coen runs the shop and the slightly academic atmosphere is a result of his basic approach to the task of selling microcomputers. He believes that his customers, when they walk through the door, are unlikely to know exactly what they want and that the best way for them to find out is by hands-on experience of a range of equipment.

So a businessman who enters with a vague idea that he could make use of a computer will meet anything but the conventional salesman's approach. More likely, he will be introduced to a machine or two and left on his own to play Star Trek or Othello for a time.

Permutations for a decision

Coen draws a parallel with a good hi-fi shop, where even the expert is faced with an enormous number of possible combinations of equipment which need to be tested before a decision can be made. The Byte Shop, as one of the few computer shops in the country offering equipment from a significant number of manufacturers, is able to take the same approach.

Computers available include the Commodore PET, Digital LSI 11, SWTP, Processor Technology, Cromemco, Apple and New Bear equipment. Peripherals are from Centronics, Lear Siegler, Teletype, TI, Tally, Dynabyte and North Star, among others.

Coen and his business partner, Bill Cannings, are at pains to point out that their enlightened approach to customer relations is anything but altruistic. They are there to make money but they are also clearly anxious to establish a reputation for integrity and dependability.

Cannings tells horror stories of American trading practices, where a computer may exist only as a cardboard model until enough customers have paid out to allow development to begin: "Over-trading would be a very kind word for some of what goes on."

He feels customers should expect protection from this kind of activity. "Our customers are usually first-time users. They can't be assumed to know which manufacturers are reliable and which are sharks. We have to act as a buffer between the two."

The criteria for choosing suppliers is chiefly that the equipment can be proved to work, that it has adequate software support, and a good supply of spares. Delivery is another factor but the shop acts as a buffer. All equipment is sold "off the shelf" and Cannings and Coen hope to smooth delivery problems by careful ordering.

Certainly the philosophy seems to be working in practice. Two weeks after opening, all the computers except the Processor Technology SOL were on display, including no fewer than six PETs and five Cromemcos, as well as a respectable variety of peripherals.

The success of the venture so far has surprised both partners. "We've done as much business in the first two weeks as I expected to do in two months," says Cannings. "Vince is an optimist—he thinks we've done as well as we should have done."

Initially, they expected to be selling mainly to the impecunious computer hobbyist and were slightly taken aback at selling nine complete systems in the first two weeks, as well as "hundreds of bits and pieces."

The wide cross-section of customers has been another surprise, ranging from schoolchildren to doctors, university and college teachers, small retailers, insurance brokers, wholesalers and computer professionals. Three computer operators have already bought their own systems, and a group of schoolboys has combined to buy a computer between them.

The number of young people in the shop almost any time you visit, often in their very early teens or even younger, is something for which it is hard to find a mercenary motive. Once again, Cannings denies philanthropy—he sees them, he says, as future customers.

Program punched from memory

Children, get much the same treatment as adults. They are encouraged to play with the machines, to learn languages, and to key-in and run their own programs.

"Many of these kids," says Cannings, "are studying computing at school. They maybe get hands-on experience once a month at a local university. It is not enough". He tells of one 12-year-old who punched out a fairly complex program straight from memory and then was very upset because one line of coding was wrong.

Books form an important part of the shop sales. Nearly all are American imports, supplied by LP Enterprises—sole proprietor, Vince Coen. Among the best sellers he recommends are Osborne's *Introduction to Microcomputers,* the eight-volume *Basic Software Library* from Scientific Research, and Scelbi's *Understanding Microcomputers and Small Business Systems.* The shop also sells most of the worthwhile American magazines.

Applications software is also receiving attention. Much of it is already available in the form of listings in the books on sale and more will be available shortly on disc or cassette. The shop is also hoping to set up a library of software written by users for sale on a royalty basis,

(continued on next page)

(continued from previous page)

and would be glad to hear from anyone who has anything which might be of general interest.

One problem The Byte Shop seems unable to crack is on-site support and maintenance. Cannings also runs a systems house specialising in Digital equipment—Computer Aided Systems—which has offices in the same premises. Its software and engineering staff are available to customers but the cost of sending an engineer to a customer's site is prohibitive compared to the cost of the original system.

Cannings points out that many problems can be solved over the telephone, and that most systems are small enough to fit in the boot of a car but that, of course, is not much help if you live in West Lothian. At least, not at the moment; the next planned stage of development is to issue franchises, of which the first is due to be agreed.

Birmingham, the north of England and "a London suburb" are the first target areas, but Cannings sees the potential as much greater than that. "There are about 750 shops in the States. Let us say there is 10 percent of the demand here, and then be conservative and halve that. That is still a lot."

He points out that there are already some 4,000 micro systems in the U.K. including about 1,500 development systems and 1,000 built by hobbyists. He has commissioned a market survey to obtain a more accurate idea of the strength of the market.

Bill Cannings and Vince Coen.

In theory, The Byte Shop's technical expertise is provided by Coen—whose 17 years in EDP have given him experience of most of the major manufacturers—while Cannings handles the business side. In practice, both are clearly devotees of their business and to forestall the obvious question, they do use micros themselves; the stock control system is run on a Cromemco.

The Byte Shop is a new type of venture in this country and it would be rash to try to predict the extent of its success, but if enthusiasm were the only criterion, there would be absolutely no doubt of the outcome.

The Byte Shop is at 426/428 Cranbrook Road, Gants Hill, Ilford, Essex.

Mailing by the easy route

DR TIM KEEN, of Keen Computers, Nottingham, has developed some software to run addressing or mailing lists on the Apple II micro. The program allows the user to enter, amend, delete, list or printout addresses. There is no limit to the size of the address and neither is any storage space wasted if the address is a small one.

The records are held in numerical order and a number is printed-out with the address to allow quick reference. Also associated with a record are two numeric sorting codes which may be used for identifying the type of customer and nature of his business, for example.

Addresses may be printed in numeric, alphabetical and sorting code orders. They may be selected in a number of ways—all addresses, by postcode, by sorting code S1 by sorting code S2, or by a combination of them—by 2 and 3, 2 and 4, 3 and 4, 3 and 4 and 5, 2 or 3, 2 or 4, 3 or 4, 3 or 4 or 5.

It is an easy process to increase the capabilities of the sorting routines.

The addresses are stored on to cassette tape for backup. There is a limit to the number of addresses which may be held in the memory at any time, although there is no limit to the number of cassettes you may have. The number of addresses which may be held will depend on the sizes of the addresses. More than 500 may be stored in the memory if the address is about 60 letters.

Call for papers

IF YOU would like to write about your experiences with micros and give a paper at a conference, start thinking.

A call for papers has been made by Liverpool University, which is holding a conference called *The Third Annual Microprocessor Workshop on Applications* in September.

Suggested subjects include testing and hardware/software trade-offs. Abstracts of about 200 words should be sent to Dr Malcolm Taylor at the university.

Logical Mole Show

THERE are some strange applications for micros but this probably beats them all.

A micro is being used as the "brain" of a mole at the Science Museum and it's a logical mole at that.

Take a trip to the museum and you'll be able to marvel at this cunning little fellow and the tunnels he "digs".

The sequence of decisions taken by the mole before each new piece of tunnel is made can be followed on an animated flow chart. You can influence all sorts of tunnelling factors such as 'straight' or 'curved' and 'digging speed' by touching contact switches. The tunnels form an ever-changing pattern of light which is quite intriguing and artistic to watch.

The mole was thought of and designed by Colin Emmet, with assistance from CAP-CPP, a software house, as well as the Computer Arts Society. Construction and design of the interface for the Motorola 6800 micro was by Grazebrook Laboratories.

Two New Pets

TWO Commodore PET distributors have opened shop in Ireland. In Northern Ireland, M & S Computer Systems, of Altona Road, Lisburn, will be selling PET, and in Eire, Software Development Services of 84, Northumberland Road, Ballsbridge, Dublin 4 are the distributors.

A big hand for Heathkit

HEATHKIT is probably the world's best-known manufacturer of electronics kits. To date, it has made its money by selling TV, radio and audio kits. It has now added some micros, and some months after the States-side launch, the H8 and H11 are being made available in this country.

The H8 utilises the Intel 8080 and costs slightly more than £300 by mail order. An American personal computer magazine criticised it recently—Heathkit came back with some telling points in its favour, though—while applauding the H11. This is based on the LSI-11, which means it costs rather more—slightly over £1,100 by mail order.

Heathkit also has some peripherals for these devices, including a VDU (£503 in kit form) though an ADM-3A might be a better bet, with more characters displayed and current prices from Penny & Giles around the £530 mark for cash-and-carry. Cassette units are available now, a floppy disc is promised soon.

Time for Tandy

THE nationwide Tandy concern is best-known for domestic electronics and especially audio equipment, but it also has a microcomputer system built around the Z80. It is now becoming available in this country. At £499, the TRS 80 could be a potent competitor in the low-cost market.

Tandy hopes for sales to small businesses but, as with PET, that market will surely open-up only when random-access storage and hard-copy output are available—that £499 buys a system with a 12 in. display, keyboard, and cassette.

The system does, however, have sockets for attachment of appropriate peripherals and Tandy will doubtless be developing them.

The TRS 80 is dramatically cheap for a plug-in-and-go system. As far as we know, its only competitor here is PET. Potential buyers, however, should probably look at a U.S. hobbyist's magazine called *People's Computers*.

The March-April 1978 issue carried a reader's opinion of the micro which made some pretty damning conclusions. Tandy was invited to answer them in the same issue, however, and there is a shorter piece alongside by a satisfied customer.

Price cut

MICRONICS has reduced the price of the MICROS micro kit to £399 assembled and £360 as a kit. That is effective immediately.

The kit consist of Z80 CPU, 1K monitor EPROM, 47-key solid-state keyboard, 2K of RAM and two parallel I/O ports. There are interfaces for TV, cassette and Teletype, and you can fit 16K of mixed ROM and RAM in the housing and

(*continued on next page*)

(continued from previous page)

up to 64K outside. An impact printer will be available shortly, at around £150.

Managing with Micros

MANAGING with micros is the title of a two-day course to be given by Bleasdale Computer Systems and the Motorola Semiconductor Group. It is geared to company directors and managers concerned with micros and will attempt to highlight the areas where micros can be used in a company, and provide an understanding of the control of micro-based projects.

Cost is £150, with accommodation on top. Details from June Dove on 01-540 8611.

Or you might try "a new and different" course intended "to help people understand the elements of the electronic microprocessor". The organiser is Mektronic Consultants, the full title is *Microprocessors for production and mechanical engineers—a non-technical introduction*.

Mektronic charges £99 plus VAT and travel for the course. It covers up to six people and the 'travel' element arises because Mektronic will bring the course to you. Metktronic is on 061-224 7312.

Pets' corner

THE happily-named Commodore PET home computer now has a companion in the U.K. KIM is already a top-selling hobbyist computer in the States. The basic micro (MOS Technology 6502) is less than £150 in the KIM package and it needs only a power supply to be operational immediately. KIM is readily expanded, and will grow "until it has the equivalent power of a full computer system". Such a pity they have to grow up.

On a less anthropomorphic note, Commodore has opened a Personal Computer Centre in Euston Road, London. Apparently delivery times are good, despite heavy demand, and 300 PETs are already in the field. Commodore is in the process of appointing an initial dealer network.

PET is a fine home computer, of course, but its wider appeal—for example as a small business computer—is somewhat restricted by the lack of random-access storage and hard-copy output.

Well, Peripheral Hardware has just provided a PET interface for the Teletype 43, a high-reliability, high-quality 30cps matrix printer terminal, and though you can't use its keyboard with the PET operating system, the special interface might be a good thing for PET owners.

Alternatively, you could wait for the official PET printer. Commodore will be launching the 2020 in August: it is an 80-column 8 × 8 printer rated at 50cps and priced attractively at £459.

Commodore has an external add-on cassette unit available immediately as well—you get one cassette deck built into PET, of course. This will cost slightly less than £60, which can't be bad.

And, of course, there is a floppy disc under development, along with a modem and memory expansion units. That lot, when it arrives, should enhance the usefulness of this well-designed and very cheap microcomputer.

ITT aspires

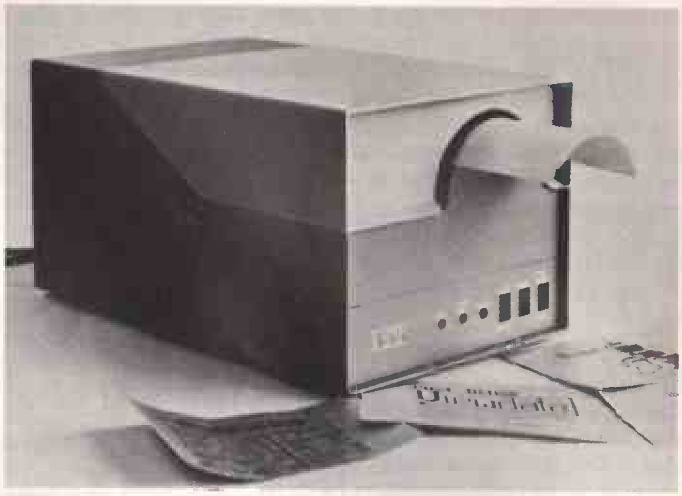
ITT is keen that more people should make more use of television sets—not least because ITT makes TV sets. Part of the company's endeavours in this line concentrate on providing Viewdata and Teletext facilities.

As well as modifying domestic TVs to take the appropriate controllers, ITT also has a neat little printer which will print out a Viewdata page.

The other angle of interest is the home computer, particularly those keyboard-plus-micro packages which plug into TVs and use the CRT as the display medium. ITT is to sell the Apple II, undoubtedly one of the better such systems. Apple scores by having a good Basic and by offering excellent colour graphics—black, white, violet and green in a high-resolution format of 280 × 192, or simpler graphics using 15 colours.

Cassette is the standard storage medium, though we gather that floppy disc is available, or will be soon.

The micro inside is the MOS Technology 6502.



See ITT aspires.

Sitting on the map

WHEN it comes to business applications on micros, CAP believes it has a year's lead over anyone else. It stems from the hefty investment already made in MicroCOBOL products and CAP is clearly doing all it can to maintain the lead.

MicroCOBOL has been developed to provide highly-portable software to run on micros and, recently, some minis. To date, MicroCOBOL has run on Intel 8080, Motorola 6800—as CAP originally-announced intentions—Intel 8085, Zilog Z80, IBM Series/1, and Digital PDP-11.

Most significantly, in terms of CAP marketing strategy, it has also been implemented on the AID-80F, which is a new Z80-based microcomputer from Mostek. MicroCOBOL has a portable operating system, a micro-resident compiler,

(continued on next page)

(continued from previous page)

and will have micro application products.

All good stuff, obviously, and very costly to develop—the company says it has had 50 programming staff working on it. Now there is involvement by NRDC and the NCC, through its Software Products Scheme.

NRDC involvement is to the tune of 50 percent of development costs, including implementation and marketing of MicroCOBOL on different computers, up to £350,000; the NCC is involved only in the applications development side, on the usual SPS terms of 50 percent of costs. SPS ceiling will be £170,000.

INSAC is contributing by taking a demo version of the product to its New York offices to assess the impact of MicroCOBOL on the U.S. market. None of these bodies is blundering into the unknown; the Central Computing Agency, the Government's chief arbiter on things computational, has had MicroCOBOL in its hands for the last six months.

Altogether, CAP reckons on spending £2 million on the project.

Don't buy—hire it first

DON'T buy your micro—hire it first. That is the idea behind a new company called CCS Microhire.

It has a range of micro systems which it will hire from £2 a day so that you can evaluate the system in your home before you decide to buy the kit.

CCS has a wide range of kit available. It includes the Commodore Pet, Apple II, Nascom 1, Tandy TRS 80, SWTC 6800, and Research Machines 380Z.

There are obvious advantages to hiring kit before you buy it. It gives you the chance to try the system first to see if you like it. And, of course, you are out of the "selling" environment of a shop.

CCS also sees a market for those who want to hire equipment for special advertising displays, events or promotions. Typical areas are car rallies and sailing races—where you may want to compute the results—to, of course, development of software.

CCS will hire a Nascom 1 complete with power supply for connection to your own TV for £2 a day, or a Commodore PET for £6 a day.

Details from CCS Microhire, Freepost, Letchworth, Herts SG6 4YA.

More cartridge

THE 3M DC300XL is an extra-length version of the tried and tested DC300A tape cartridge. It gives you 450 ft. rather than 300, is otherwise fully compatible, and costs about £15·30.

New field

AMPEX is, of course, well-known as a manufacturer and vendor of magnetic media and storage peripherals, including add-on memories, discs and tape decks for minis. Audio recording on mag tape is equally an area of specialisation for the company.

So it is not unreasonable to find the company putting that expertise to use in what is more of a mass market as far as the computer business is concerned, with the launch of a line of word processing media. They include cassettes, floppy discs and magnetic cards. There is also a voice cassette for disctation recording.

Ring binders for floppies

A RING BINDER to accommodate floppy discs has been introduced by Willis Computer Supplies Ltd., the supplier of removable magnetic data storage media.

Covered in white PVC, the binder contains eight frames, each of which has four pockets permitting a total of 32 flexible discs to be housed.

An ingenious, horizontally-hinged cover permits the binder, which costs £21, to become a free-standing desk unit.

Books

IF you are the type who tends to go boldly and that's the last anyone sees of you for some time, you will like the Super Star Trek version David H. Ahl has included in his *Basic Computer Games;* he includes a useful explanation of galactic quadrants.

If, however, you are more inclined toward the Haika—and that's *not* a new micro—you can turn your hand to poetry using his generator, although reciting the source listing could well be an end in itself.

David doesn't seem to have missed much from his collection, in fact, and it is a very well produced book. It has cartoons, as well. And the Basic in the title means BASIC, of course.

Grittier reading is Don Lancaster's *TV Typewriter Cookbook*, which shows you how to put your own words and pictures on ordinary sets and is also a book on cheap things which can be connected to a microprocessor to do genuinely useful tasks.

No cartoons, but plenty of diagrams, and he covers a good deal of ground—according to the preface, he's aiming at everyone from the hobbyist who likes video games to the electronics technician and the software specialist.

If your idea of computer art is those line drawings which look like string and nails, read Ruth Leavitt's *Artist and Computer*. Or rather, just look at it. It is concerned with what computers can produce themselves but, more important, with the way artists use computers and computerised techniques. The illustrations are attractive.

Basic Computer Games, Microcomputer edition, edited by David H. Ahl. Price £5·50.

TV Typewriter Cookbook, by Don Lancaster, £7·95.

Artist and Computer, edited by Ruth Levitt, £3·95.

All are available from LP Enterprises, 313 Kingston Road, Ilford, Essex, IG1 1PJ. Telephone 01-553 1001.

News round-up from Intel Fair

THIS year's Intel Fair proved to be more than just the formal announcement of the Intel 8086 microprocessor.

There were some important announcements on new software available for the now-established 8080 microprocessor and some impressive exhibits.

Held at the Wembley Conference Centre this month, the fair included a programme of three seminars running simultaneously through the day, although the paper on the Intel 8086 made it necessary to combine two of the suites, as the room was packed with some 500 delegates.

The 8086 is the highest-performance Intel microcomputer to date and, claims Intel, "capable of providing a performance equivalent to today's top-of-the-range minicomputers".

The 8086 is source-code compatible with the 8080A and the 8085 via a translator package which will convert most of the 8080/8085 source code into code for the 8086.

The 8086 provides all the features of the 8086A plus 16-bit arithmetic, signed 8- and 16-bit arithmetic (including multiply and divide), interruptable byte string operations and improved word manipulation.

Also included are such minicomputer-type features as re-entrant code, position-independent code, and dynamically relocatable programs.

The chip contains, remarkably, about 29,000 transistors and has an internal clocking rate of 5MHz (200 nsec), or on selected chips, 8MHz (125 nsec).

For commercial applications the 8086 is capable of addressing just over one megabyte of memory, by means of a bank of registers designated the segment register file which is used to augment the 8080-like 16-bit (64KB) addressing capability. Basically four bits are added to the addressing signals which allow for the selection of 16 × 64KB segments.

The bus interface unit maintains a 6-byte fetch-ahead instruction queue technique, for overlapping the execution and fetching of instruction. Price for the 8086 in the U.K. will not be known until production is fully under way.

More from the Fair

OTHER announcements included a new range of microcomputer development systems, a Fortran 77 compiler; a new microcomputer chip, the 8022, with on-chip A-to-D converter; two new PCM code chips; and a new version of MicroFocus CIS COBOL.

Current Intel development systems have been criticised and thought to be difficult to use. The new development systems, known as the Intellec Series II, will no doubt have some enhancements over the older systems.

Model 210 is used for the smaller types of projects using MCS 80, MCS 48 or MCS 85 microcomputers and is supplied with 32KB RAM, TTY interface, VDU, high-speed paper-tape punch/reader and universal PROM programmer, and has an 8-level priority interrupt system. Standard firmware is a ROM-based monitor, editor and assembler.

Model 221 has integral floppy disc drive, and upper- and lower-case keyboard/VDU; 4K of ROM contains the monitor, bootstrap and systems diagnostics. The ISIS-II diskette operating system has a re-locating macro-assembler, linker and locator.

Model 231 has an additional 32KB RAM and a one megabyte dual-drive, double-density disc system for the user who wishes to use high-level languages such as PL/M and Fortran.

The Fortran 80 is claimed to be the first ANS Fortran '77 offered by any minicomputer or microcomputer manufacturer and will run on all Intel development systems.

Competition

IN COMPETITION to the CAP MicroSoft MicroCOBOL demonstration, a new version of MicroFocus CIS Cobol was demonstrated on the Intel MDS-800 and the Tandberg TDV range of intelligent terminals. CIS COBOL compiles each COBOL source statement to a compact intermediate, code which is then interpreted by a run-time system, much like MicroCOBOL. Also offered is an ANSI '74 compatibility option.

Terminal

ALSO available on the Tandberg TDV-2114, an intelligent terminal with stand-alone computing capability, is the Logica Commercial BASIC. It occupies 24K of RAM and may be overlaid when the TDV-2114 is required to be used in terminal mode.

Hotel system

THE HOSKYNS Hotel System was a good example of a dedicated commercial system to which the use of a microcomputer is particularly suited.

A typical hardware configuration, costing £12,500 approximately, will comprise a processor containing two Intel 8080s and a bit-sliced disc controller, 20KBytes of memory, twin Shugart disc drives, Newbury Labs 7004 VDU, 80-column 200cps Westrex printer, and eight special mini terminals manufactured by Hoskyns.

The hotel system is part of a range of MAS Modular Application Systems and is designed to cover general accounting, administration, reservations, back office procedures and ledgers. Holiday Inns has signed a major order for a somewhat larger system.

A PRACTICAL GLOSSARY

Running the terminological gamut from A to B

There aren't too many businesses which actually had to invent a jargon word simply to describe its own jargon; but computing did. We present part one of a tour around the more common terms which both novices and initiates frequently find baffling: A to B this issue, more in the next, and so on—by the end of year one you will be armed with a Compleat Plain Person's Guide to words in the world of the computer.

Absolute Address
The *address* of a physical memory location, which in practice is a way of describing exactly where in the memory the required location is. The alternative is some indirect form of address, which describes how to find the location rather than precisely where it is.

Absolute code
(Synonymous with *object code*). Basically a program in object code is a sequence of instructions which can be read directly by the computer; absolute code is thus represented by a sequence of bit patterns. It is difficult for people to read, of course; it is also what a mnemoinc program is translated into before the computer runs it.

Access time
The time taken to reference a particular item in storage—to read from or to write to a memory location or a stored record on disc or tape. Not to be confused with cycle time, which means the total taken for a program instruction to reference a memory location, read from or write to it, and then to return to the next instruction in the program.

Accumulator
A dedicated storage location within the processor containing data on which the computer carries out an operation. The Intel 8080 cpu chip contains one accumulator, the Motorola 6800 cpu contains two; obviously the more accumulators you have available, the more arithmetic or logical operations that may be carried out in parallel.

ACK (Acknowledge)
A convention code meaning "I have received your request which will be dealt with as soon as conveniently possible". The 'request' might be any external event, like the arrival of a message at a terminal.

Acoustic coupler
Post Office telephone lines are intended for voice transmission signals. The coupler converts pulses of sound from a telephone line into the *digital* signals a computer can understand; and conversely it converts digital signals into the *analogue* pulses which can be sent down a phone line. Basically a coupler is the same as a *modem*, but it is not plugged directly into the telephone line; instead, it uses an ordinary telephone receiver. If you have a terminal with a coupler attached, and if you know that somewhere there is a computer with a telephone line of its own, you can dial the computer number. You then place the telephone receiver handset into a shaped recess on the coupler and you can use the terminal in the normal way to send and receive data. It all sounds wonderful, but your computer normally needs a telephone line of its own (not cheap); and you cannot send or receive any faster than 30 characters per second (usually much slower, in fact).

Acronym
Abbreviations of a group of words to form one symbolic word—like BASIC, COBOL, and FORTRAN.

Address
A way of referencing a memory location containing data or a program instruction.

Address Bus
In practice, this is a physical connection between a processor, the computer memory, and other parts of the system. A dual-bus approach normally separates the memory bus from the I/O bus, which means that two sorts of operation can go on at the same time—for instance, data can be moved between the processor and memory at the same time as data is being moved between the processor and a terminal. Physically a bus is likely to be a set of wires or a cable connection usually attached to the *backplane*. When you slot a printed circuit *board* into the backplane, the edge connectors on the board make contact with the bus—and if the circuit board holds memory, that effectively plugs some memory into the bus. Similarly, if it holds the controller for some peripheral, slotting-in the board connects that device to the bus.

Analogue
Practically all computer operations are *digital* in nature, which means they refer to distinct states which can be represented for instance by particular numbers—like 0s and 1s.

Analogue states change more smoothly and typically have a comparative value—like the position of the hands of a clock or a voltage level.

Charles Babbage's difference engine operated in a mechanical analogue mode; there are still analogue computers in use for highly-specialised scientific applications.

ALGOL
Acronym for ALGOrithmic Language. There are two different versions of the language, ALGOL 60 and ALGOL 68, neither of which has been implemented commercially on micros.

AND
The arithmetic operators are well-known—plus, minus, divide and multiply. Logical operators can also be used. AND is one of many possible logical operators. In binary terms, an AND will produce only a 1 where both inputs are 1; if one of the bits tested is not a 1, the result of the AND will be 0.

ANSI
American National Standards Institute. It operates somewhat like our own BSI but rather more aggressively. Among its areas of responsibility lies the specification of some of the high-level programming languages, including COBOL and FORTRAN (and soon, BASIC, too).

Applications
Applications are what a computer actually does (as opposed to how it does it). For instance, an *operating system* is not applications software since it does not produce usable end results A Star Trek game or a payrol run qualify as applications

Architecture
This word is vaguely defined as the way a computer is designed. It normally refers to the design of the internal physical structure of a system, the way its components are interrelated.

Arithmetical and Logical Unit (ALU)
A hardware function within the CPU which executes arithmetic and/or logical operations on the contents of an *accumulator* according to instructions in a program.

Arithmetic shift
A *bit* manipulation instruction which moves each bit in the accumulator one position to the left or right. In binary terms this has the effect of doubling or halving the value, depending on whether you shift left or shift right. That is the way binary numbers are; you will really understand this effect only if you understand binary arithmetic.

Array
An array is a set of *variables* which may be arranged within a logical relationship. It may refer to data, or to memory locations, or to components within a processor.

Arrays are used in the BASIC programming language, in which case the textbooks usually define an array something like "a whole set ol values designated by a single name." In effect, a BASIC array is a series of memory locations which probably will contain different values. As a block the array is given just

(*continued on next page*)

(continued from previous page)

one name, usually an alphabetic letter; to reference individual locations within the array, you use a *subscript*—so C(2) indicates the second element in the away named C.

ASCII
American Standard Code for Information Interchange. This was established by ANSI in 1963. It is one of the ways of representing alphanumeric characters in *binary* code, where specific binary patterns correspond to particular alphanumerics.

ASR
Automatic Send-Receive.

The simplest terminal is RO, which means receive-only. The terminal cannot *send* any data because it does not have the wherewithal—it is usually a printer. KSR means Keyboard Send-Receive; the terminal can receive information, and it can also send information because it has a keyboard for input.

An ASR terminal has some capability to store information, usually on paper tape, cassette, or internal memory; this can be used to receive incoming data without that data necessarily appearing on a screen or printer. More usually, an ASR terminal sends pre-prepared information which is available on paper tape or cassette. the 'automatic send means that the terminal operator does not have to key-in each data item —he or she can start the paper tape reader or the cassette unit and the input is automatic.

Assembler
The most direct way of programming a computer is by means of zeros and ones; that produces the 'absolute' code or '*objects*' code which the computer can understand. People do not do too well with it, however, and it is laborious to write out and understand. So a variety of shorthand forms more comprehensible to humans has been developed. One stage removed from binary coding is assembly code; it uses mnemonics which are de-coded (or 'assembled') by a special program (the assembler), into machine-readable code. For example, on the Intel 8080, the instruction move the contents of register L into register E will be keyed in at a terminal as "MOV E,L". After the assembly stage, though, this would be converted to an almost incomprehensible 8-bit binary instruction, 01011101.

Asynchronous
This term is normally applied to terminals. An asynchronous terminal cannot be connected to a *synchronous* connection. More specifically, it is a mode of working which is not dependent on the accepting device or the processor internal timing requirements. For instance, data signals transmitted over a telephone line do not have to be synchronised with a processor internal machine cycles.

Aficionados abbreviate this term to 'async'.

Audio Cassette
Normal voice and music cassettes can be used by some micro systems for storing *digital* data. An actual data cassette has much finer tolerances, but a microprocessor works relatively slowly and many can read and write data on audio cassettes without being affected by the various electrical interferences to which cheap and relatively simple devices are prone.

Babbage
Charles Babbage was born in 1791, and effectively invented the *digital* computer. He evolved all the basic concepts of computing, even though not all of them could be put into practice at that time he used wood and metal, some of his ideas demanded electronics. He had help from Lady Lovelace. She was related to Byron and wanted to use Babbage's machines to predict the results of horse races.

Backing store
Tapes and discs in all their manifestations; paper tape too, sometimes. They are also called backing memory, secondary store, and auxiliary store.

Backplane
The backplane is what circuit boards slot into. It is a piece of the computer chassis, which holds the boards in slots and provides connections for the circuits with other system elements—for example, it may well incorporate *buses*.

BASIC
The Beginner's All-purpose Symbolic Instruction Code is popular, easy to learn, easy to use, and available widely on minis and micros. Not the most elegant or space-saving

(continued on next page)

(continued from previous page)

of programming langugaes, it compensates for this by being one of the easiest to handle. BASIC programs are not portable from one machine to another, but re-coding different computers is straightforward: and once you have learned it for one machine, it is easy to convert yourself to another.

Batch

One of the ways in which work may be done on a computer is to group together all the information the system will need to do a particular job. It is then loaded on to the computer so that it can be worked on without further intervention. That is batch processing, or working in batch mode because it involves batching everything required to do the job. The usual contrast is with *interactive* processing, where you have to supply instructions or data while processing is taking place.

Baud

The capacity of a communications line is measured in bauds. The number of bauds tells you the rate at which information will pass down that line. You can often assume that baud is synonymous with bits per second, so that a line with a baud rate of 2,400 will carry that number of bits in a second. That is not always true, though, as it is possible to have bits travelling down a line two or more abreast. There are standard baud rates at which terminals communicate with computers.

BCD

BCD stands for Binary Coded Decimal, a way of expressing decimal numbers using bits. Each digit of a decimal number is allocated four bits for its representation. Extended BCD Interchange Code, EBCDIC, is an IBM code which allows you to represent characters, too. (This is not the only way of representing alphanumerics in *binary* code—compare *ASCII*).

Benchmark

A means of measuring what a computer can do and how fast it can do it—its power, in other words. Normally the benchmark is a program or group of programs whose results are thoroughly known and which can be used to compare the rate at which different computers run that kind of work.

Binary

The binary system is a way of representing numbers to the base two—as opposed to the decimal system, to base 10, or the ancient Babylonians' system (base 16). It uses only the symbols 1 and 0, which makes it a useful system for representing numbers in a computer. It can also be used to represent characters, where specific patterns of 0s and 1s can be used for individual characters. Best known of these codes are *ASCII* and *EBCDIC*.

Bipolar

There are many ways of making *semiconductors*, and this is one of them. It makes for fast operation, but it is expensive and generates a lot of heat. The best-known type of bipolar logic is Schottky (after Mr Schottky).

Bistable

Bi-stable as a term may be applied to various parts of a computer; it means essentially that what it refers to is capable of assuming either one of two stable states. Now you can forget this word, or you can make jokes about how it sounds like a village in Dorset.

Bit

The bit is the basic unit of information storage in a computer. The name comes from *binary* digit (get it?) and any bit can represent a 0 or a 1. Which it represents depends on the voltage level in that physical location in the computer. A bunch of bits is a *byte*.

Bit Density

This is usually the number of bits stored per inch (bpi) on tape or disc.

Block

It is often convenient for information to be transferred between components of a computer system—e.g. terminals, disc drives—in regular-sized chunks called blocks. When stored on media such as tapes or discs, a physical area of the medium may also be termed a block; it will be separated from other blocks by an inter-block gap.

Board

A printed circuit board, or PCB, is sometimes called a printed circuit card. Individual electronic components can be plugged or soldered into the board to make contact with those circuits. At the other end of the circuit is the edge of the board, which is equipped with *connectors*. Those engage with further circuitry in the *backplane*. The processing functions of a computer and its main memory will be held on a small number of *PCBs*.

Boolean

Boolean algebra describes a set of logical instructions, which centre on the ability of statements to be true or false—precisely the kind of two-state operation a computer likes. The name comes from a Mr Boole.

Bootstrap

Before you can get a computer to do anything, you need to *load* a program into it. So you need a way of loading a program before there is a program in the computer which can load programs. While it would be satisfactory if programs did load themselves by their own bootstraps, a bootstrap is, in fact, a set of resident instructions which are initiated by a special manual switch; they then call in the program which is to be loaded. This facility is often called a bootstrap loader.

Branch

The great thing about computer processing is that it allows for alternative courses of action to be taken, depending on certain measured values. In fact, that is what distinguishes computers from other forms of calculating machinery. When in a sequence of instructions there are two possible courses of action to be followed, then by taking one of them the program is said to branch.

Breadboard

A printed circuit board in its most basic state is termed a breadboard—that is, the circuitry is there, but the components which connect to it are not.

Breakpoint

It is often useful to build natural breaks into programs, where they can be interrupted either by an outside agent (you) or by a monitoring program within the system. Such breaks are called breakpoints or checkpoints; the program is prepared for interruption when such a point is reached.

BSC

Binary Synchronous Communications is an IBM communications *protocol*, which defines how information is parcelled for transmission along communications lines. You probably do not need to know any more about IBM communications than this just yet.

Bubble

You do not find much bubble memory around commercially yet, but the term crops up very frequently. It uses a form of magnetic technology for mass storage of data. Basically, it is one of two promising memory technologies, which fit between external devices like discs and *semiconductor* store. The other one is CCD—see next month's instalment. Both are faster and more reliable than disc storage but are cheaper than semiconductors.

Buffer

A buffer is a halfway house between one part of a computer and another, where information can be stored temporarily. It is useful because it cushions the impact of the different speeds at which different parts of a computer system can handle data—for example, a printer terminal may accept data from a high-speed communications link at 240 characters per second but actually print it at 30 characters a second. So the data has to go somewhere while it is waiting to be printed, and a buffer is one place where it might go. This buffered operation allows different parts of the system to work at optimum speed.

Bug

Bugs cause things to go wrong. In particular, bugs are what should not happen when you run a program but invariably do. You have to run it enough times to identify and stamp on all of them, a process known as debugging.

Bus

Information travels from one part of a computer system to another by bus—sometimes called a trunk on larger computers. A bus is a set of connections which enable the hardware components of a system to link or interface.

COMPUTER WORKSHOP

BRITAIN'S LONGEST ESTABLISHED COMPUTER SHOP

HARDWARE

COMPUTER MP-68 with 4K Memory & Control Interface Kit **£275.00 Assembled £330.00**
MEMORY MP-4K Memory Kit **£70.00 Assembled £90.00** MP-8 8K Memory Kit **£170.00 Assembled £195.00**
DISC SYSTEM DMA1 1.2 megabyte twin system Kit **£1450.00 Assembled £1525.00**
MF-68 160K byte twin minifloppy Kit **£800.00 Assembled £860.00**
TERMINAL ASR43 Upper/Lower case 132 Char. 30 cps KSR **Assembled £915.00**
CT-64 Upper/Lower case visual display terminal Kit **£230.00 Assembled £315.00**
CT-VM Video Monitor for CT-64 **Assembled £140.00**
PRINTER RICOH RP-40 Daisy wheel printer 30 cps **Assembled £1800.00**
CENTRONICS 701 Dot Matrix printer 60 cps **Assembled £1400.00**
PR-40 40 column tally roll printer 55 cpm Kit **£200.00 Assembled £250.00**
CASSETTE AC-30 Cassette Interface for 2 audio cassettes Kit **£65.00 Assembled £100.00**
GRAPHICS GT-61 Graphics Terminal with P197 Power Supply **Kit £92.50 Assembled £120.00**
SUNDRIES MP-R EPROM Programmer for 2716's **Kit £36.00 Assembled £43.00**
MP-N Calculator Interface Kit **£37.00 Assembled £44.00**
MP-T Real Time Clock Kit **£32.00 Assembled £40.00**
MP-L Parallel Interface Kit **£30.00 Assembled £37.00**
MP-S Serial Interface Kit **£30.00 Assembled £37.00**
PPG-J Joystick Kit **£32.00 Assembled £40.00**
MOD1 UHF Modulator to link CT-64 to T.V. Kit **£4.50 Assembled £6.00**

PRICES DO NOT INCLUDE V.A.T. or carriage.

SOFTWARE
Software includes 3K†, 4K and 8K Basic*, Coresident Editor/Assembler, Text Editing System†*, Text Processor†* TSC Assembler†*, Games and Scientific packages.
* Disk Based versions available.
† Full source listing included.
Prices vary between £5 and £20.

Both Disk Systems come complete with the comprehensive 'Flex' Disk Operating System. Commands include -

APPEND	**COPY**	**LINK**	**RENAME**
ASSIGN	**DELETE**	**LIST**	**SAVE**
BACKUP	**EXECUTE**	**MONITOR**	**STARTUP**
BUILD	**GET**	**NEW DISK**	**VERIFY**
CATALOG	**JUMP**	**PRINTER**	**VERSION**

Applications include: • **Travel bookings • Word Processing • Questionnaire analysis • Process control • Teaching • Accounts analysis**
Existing customers: **Schools — Universities — hospitals — M.O.D. — business — industry.**

Please write for full details to:-
COMPUTER WORKSHOP, 38 DOVER STREET, LONDON W1. Tel 01-491 7507
COMPUTER WORKSHOP, 12 TRESHAM ROAD, ORTON, SOUTHGATE, PETERBOROUGH. Tel 073 323 4433 OR
COMPUTER WORKSHOP, 29 HANGING DITCH, MANCHESTER, M4 3ES. 061 832 2269

● **Circle No. 309**