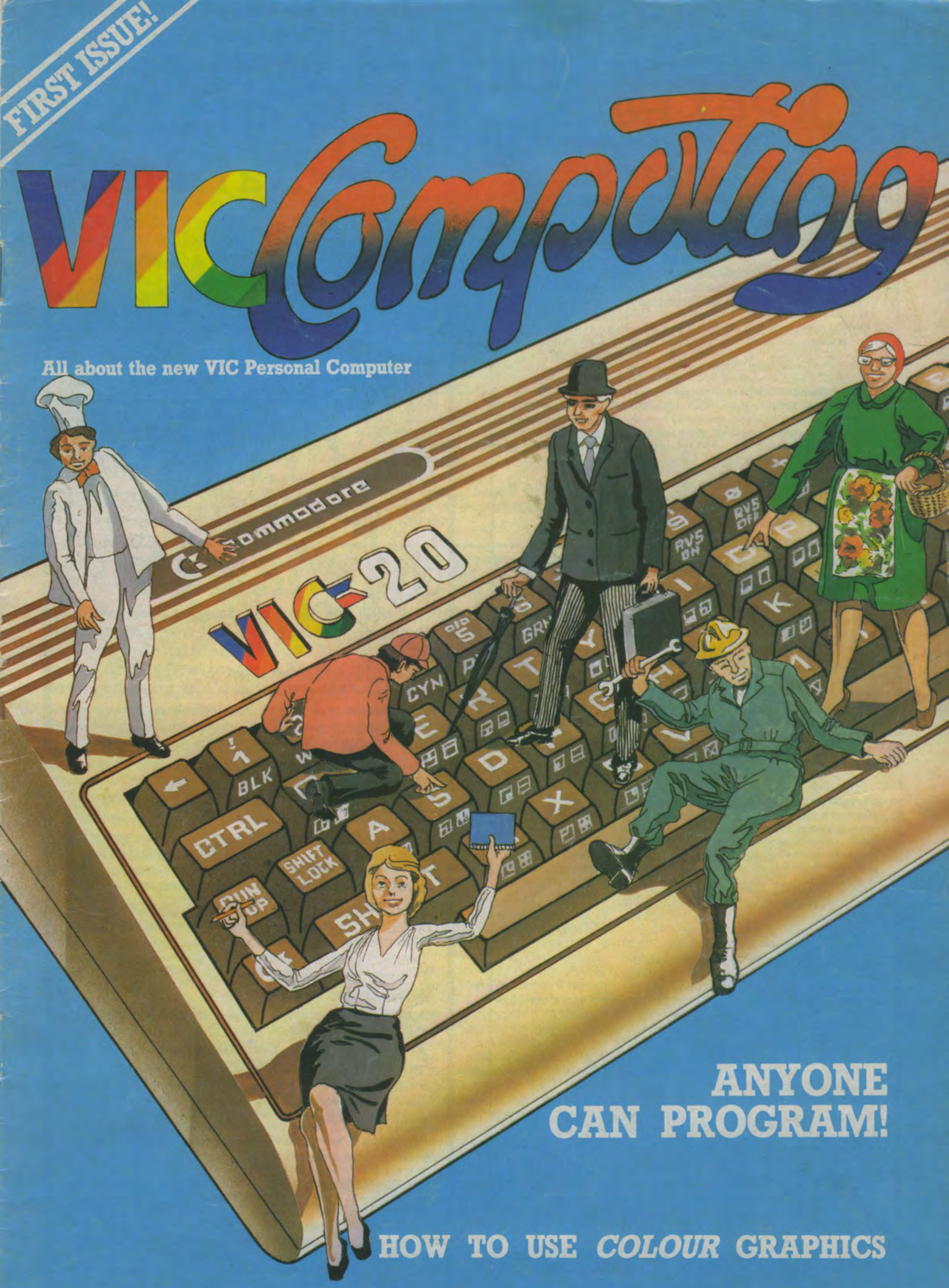
FIRST ISSUE!

VIC Computing

All about the new VIC Personal Computer



ANYONE CAN PROGRAM!

HOW TO USE *COLOUR* GRAPHICS

45p

VIC Computing

Publisher Julian Allason
Editor Richard Pawson
Programming Editor Mike Gross-Niklaus
Technical Editor Chris Preston
Graphics Denis Appleby
Production Manager Gill Atley
Address: P.O. Box 48, Newbury, RG16 OBD
Telephone: 0635 201131

Advertising Manager Jonathan Horne
Advertising Office VIC Computing. PRINTOUT Publications. North Warnborough, Basingstoke. RG25 1PB.
Telephone: Odiham (025671) 2724
Telex: 339393 Attn. VIC COMPUTING.

VIC Computing is published 6 times a year by PRINTOUT PUBLICATIONS, in association with Commodore Systems UK Ltd.

TO OUR READERS

Welcome to the world of the microcomputer! *VIC Computing* is a new magazine that will appear six times a year - devoted to owners of the Commodore VIC, and those considering the purchase of a low-cost computer.

Whether you already understand the intricate design of a microprocessor, or (more likely) aren't too sure what the keyboard is for - there is lots to learn. *VIC Computing* can help you - all of our contributors are experienced microcomputer users who remember their early days well enough to want to help you avoid the same problems! The result will be a heavy mix of tutorials, reviews and anecdotes that are written in a wonderful new language we have discovered - English - and not in impenetrable computer jargon.

Take a look at this first issue, hot off the press as the very first VICs go on sale. We have a tutorial that can teach you the beginnings of programming in *less than one hour,* a careful study of VIC's expansion capabilities and peripherals, as well as a straight-talking explanation of what *you* can make the VIC do.

VIC Computing comes to you from the same stable as *Microcomputer Printout,* the independent magazine about the PET and VIC. So why another publication? We felt that there was not only scope but a tremendous need for a magazine devoted solely to the VIC, which we believe will be Britain's most popular computer. Commodore agree and have joined us in putting together the first computer magazine that is truly written for beginners. Their co-operation will help to ensure that *VIC Computing* is the most up-to-date and informed magazine available.

Because the VIC is making its first appearance the amount of software and peripherals available is limited - although not for long, we suspect.

We have deliberately kept this issue - and the price - small. Expect to see *VIC Computing* grow quickly.

If you've already bought your VIC then a year's subscription is going to be essential, and if you are still hesitating then 45p is a small price to pay to learn about microcomputing the easy way.

BEGINNERS QUERIES

We welcome letters from both VIC users and from people considering the purchase of a microcomputer. If you would like a personal reply, please enclose a stamped, addressed envelope. For this first issue, we have answered some of the questions put to us while we were compiling the magazine.

How Much of VIC's Memory is actually usable? The basic VIC has 5K bytes of RAM - enough to store 5120 alphanumeric characters. However, VIC needs 1K worth for its own workings, and ½K to store the screen characters. This leaves you with 3½K or 3584 bytes to store your BASIC program and variables.

As well as playing games, could I use my VIC for storing Names and Addresses? Yes. Undoubtedly someone will shortly offer a program to do this or you might write your own. Because the cassette system is too slow for accessing such information, all addresses would have to be loaded into memory at the start of the day or after you have finished playing games! Allowing 500 bytes for the simple program, and 100 characters per address, you would only have room for 30 in memory at the same time. Loading would take about one minute. Our advice: use it for telephone numbers only, or buy the VIC Disk unit later for more and faster storage.

When I learnt to program at school, I was taught that BASIC allowed 80 characters per line. How can VIC work with only 22? On the VIC, a line of BASIC can constitute up to four lines on the screen, but with one line number -that's 88 characters in all! Unfortunately, this makes BASIC listings a little harder to read, unless you have the PRINTER which prints up to 80 per line. Our advice: keep line lengths short as we will in *VIC COMPUTING* listings.

The VIC Cassette Unit costs £49.95 - why can't I just plug in my own Audio Cassette player - the connection looks standard? Recording digital data on tape is not quite the same as recording music. The wave pattern is not so smooth, rather more square which requires more precision. Also, any imperfection in music is slightly annoying or often unnoticed by the ear, in data, it can cause the computer to go into a what looks like a 'coma' - we call it a "crash", and you must switch off and on again.

Consequently, though the VIC cassette has the same mechanics as your audio unit, the recording electronics are totally different. Whether that justifies the extra cost is arguable but we still rate the total system good value.

Could I use my VIC for accessing any of these Public Information Services like CEEFAX or PRESTEL? Not just yet - though it is quite likely within 6-9 months. (Incidentally, CEEFAX is a television-only system, PRESTEL uses the telephone). You need a way of connecting VIC to the telephone, which means either a Modern (fitted by British Telecom) or an Acoustic Coupler with two rubber cups into which you press the handset. Fortunately, VIC has the standard RS 232 interface so almost any make will do. Even so, it will set you back at least £200 unless Commodore comes out with a cheap VIC Modem. Once you have got one, however, the possibilities are almost limitless - not just of accessing public information, but also for private electronic mail and 'bulletin boards.' Our advice: wait a few months - *VIC Computing* will cover the availability of hardware and information sources, as well as the whole new range of Community Services and applications that this could open up.

Just how difficult is it for a complete beginner to learn BASIC from scratch? A lot easier than it was for new microcomputer users 3 years ago. To begin with, the VIC manual is of a much higher standard than any of Commodore's previous efforts. Secondly, BBC TV will be broadcasting courses teaching BASIC; although they will be illustrated by reference to a different machine, much of it will still be relevant. Thirdly, *VIC Computing* will be running a series of self-contained articles to take a complete beginner through to a competent level of programming. The first appears in this issue and will teach you how to write a simple twelve-liner after only one hour. If you want to proceed faster, most computer dealers stock a range of books on BASIC. There will shortly be released a cassette which teaches BASIC on the VIC itself.

I read in *Printout*, that VIC is very similar in design to PET, so will it run all the software that is available? The answer to this one is a qualified Yes. The VIC has essentially the same internal software as PET and runs BASIC 2.0, which is found on all 3000 series PETs. The recording format is identical which means that PET cassettes should load correctly. The main problems are as follows:

1) You only have 3.5K of memory (unless you have added some expansion) which limits you to the simpler games and utilities.

2) The 22 character-wide screen will tend to mess-up PET screen displays. And of course there will be no colour.

3) The memory map is not identical to PET so if there is any sophisticated POKEing in the program it will need to be modified.

Nevertheless, a lot of PET software is currently being modified by amateurs and suppliers alike to run on VIC. *VIC Computing* will be supplying tips on how to do this yourself.

Will it be possible to use my VIC to control my model railway? - I am an electronics enthusiast? VIC abounds in suitable interfaces, but you need to be adept at both electronics and programming to handle this. For example, the User Port has 8 lines at TTL levels which you can set either to 0V or 5V at will. This must be buffered from your electronics with a standard 7400 chip. You may then, however, use each line to control a set of points, signals etc.

To: ACT Microsoft Ltd.,
Shenstone House,
Dudley Road,
Halesowen,
West Midlands

Dear Sir,

Please send me details of your exciting new programs for the VIC.

Yours faithfully,

Name: .....

Address: .....

.....

Fill in your name & address and send this page to ACT Microsoft for information about the new VICSOFT range.

BUT WHAT CAN IT DO?

As publishers for two magazines about microcomputers, we are pretty used to answering the queries of our readers ranging from heavyweight technical problems down to questions about where to buy software and how to begin learning programming. If you are considering buying, or have just bought a VIC then you are in luck - we have anticipated some of your most likely questions and will be answering them one by one through the pages of this magazine.

To begin with, a question that many people are reluctant to ask, and which is at the same time difficult to answer - just what *can* you do with a VIC? After a long lunch at the Waggonload of Monkeys our panel decided this was really three questions rolled into one:

What is a microcomputer; what are its applications; and how do you go about making it do what you want?

The microcomputer developed out of the Mainframe Computer as technology allowed components to be made many orders of magnitude smaller and cheaper, and demanding of less support. Mainframes require large rooms, air conditioning, complex electrical services, operating staff, maintenance crew and supporting bureaucracy. And a large supply of white coats. Microcomputers usually require a mains plug! This evolution is causing attitudes towards computers to undergo a subtle change - away from the 'Big Brother' image and towards 'Personal Computing.'

That simply means a computer where *you* need it, doing what *you* want it to, without affecting other computer users and costing next to nothing to run - less than a domestic light bulb in fact.

Taking home as the *where* (though as we shall see, the VIC is not limited to that environment) let's move smartly on to *what* you might want to do. Our experience with other micros suggests three areas: Entertainment, Learning and Utility.

The first of those is the most spectacular and is purely a cover name for *Games* - albeit with a difference. Chess, Draughts and Backgammon are all transformed by the computer. The VIC will be able to play against you with a degree of skill or difficulty that you specify. Given the right software, you can record games, make trial moves or re-enact the positions of professional matches. Then there are the Arcade Games - imitations of the ones that have done so much to destroy Pub Life as we know it, complete in VIC's case, with sound effects and full colour. All that's missing is the coin slot! Space Invaders, Galaxian, Night Drive and Breakout are all under development, along with a few titles which you won't find at the Waggonload of Monkeys. For the VIC, these will be sold as ROM Cartridges that plug in the back, or tape cassettes to load from the VIC cassette unit.

The microcomputer is as much as innovator as an imitator, however, and is spawning a new family of games. Some are simulations that enable you to re-enact a scenario or try your hand at controlling a particular event. Our Editor has now failed to land a Jumbo Jet successfully no less than sixteen times. He had more success quenching a Forest Fire.

The latest arrival are the adventure games - not quite as visually stimulating or fast as the others but every bit as addictive. Typically you are given a task to perform such as finding and returning with treasure to the surface from a vast underground labyrinth. Each turn sets a new scene and potential course of action, such as:

LYING ON THE FLOOR OF THE VAST CAVERN IS A SHINY BRASS LAMP. YOUR COURSE OF ACTION?

RUB LAMP

THERE IS LITTLE TO BE GAINED FROM RUBBING AN ELECTRIC LAMP!

At this point we had better assure you that there is more to Personal Computing than playing games. Learning at Home conjures up images of school homework, and books that teach you Ballroom Dancing in Three Easy Lessons. Though there is no reason why a VIC couldn't do both, and better - there are a lot more interesting areas to explore.

This applies particularly to young children - where a suitable program can provide extra help at crucial points; simple arthimetic and spelling come to mind. The ability to respond to answers with appropriate praise or encouragement, coupled with the excitement of colour and sound effects, means that many children will respond faster and more willingly than they might to a normal pupil-teacher situation. Higher up the educational process, this simple idea can be extended to tutorials, quizzes, tests and simulations with which the student can supplement or clarify his curriculum.

And that brings us to adults - who are not exempt from this new form of learning! Remedial subjects, new languages and even handicrafts are potential Teachware - the appalling jargon term for such programs. The important thing to grasp is that your VIC will not replace or supercede formal class-

continued

ANYONE CAN PROGRAM!

They said it couldn't be done - an article that could teach you to program in less than one hour! Even if you previously thought that programming was not for you, read this first of a series by Mike Gross-Niklaus. After all, what have you got to lose?

As someone who lectures in Computing for Beginners, I very often find that I have only to mention my job and people immediately start to clam up. "I'm sure its all very interesting," they say "there's a computer in our office - but I could never learn to use it because I'm just not technically minded." The chances are that at some stage they have been shown a computer program - and one glance at the strange list of symbols and unrecognisable words has convinced athem that a technical degree must be a pre-requisite for programming. Nothing could be further from the truth, as the title maintains - anyone can program!

There are two aspects to writing a computer program: defining the task you want done in an orderly and unambiguous way, and then writing these instructions in a compact form or code that will save both space and time when the computer performs them. What most people don't realise is that they perform processes similar to these many times a day - in the office, in the car or in the kitchen! The classic example has to be a knitting pattern - an incomprehensible array of symbols, numbers and abreviations to most men - the practised knitter could take one look and immediately envisage the shape and design of the finished garment. This article will take you from the position of 'computer ignoramus' to being able to write a 10 line program in less than one hour.

Cooking an Omlette!

To get back to that mention of everyday programming tasks, consider the process of cooking an omelette.

First you probably plan out what you are going to do and assemble in one place the pan, bowl, spatula etc.

Then you get the raw materials together, eggs, butter, salt and pepper.

Having heated the pan, you break in the eggs, add the butter, salt and pepper.

Next you let it cook, perhaps scraping the outsides into the middle and tipping the pan to move the parts still liquid onto the heated pan.

Finally when it is ready, you fold it over and 'flup' it out onto a plate.

Writing down in a short hand form, as you might find it in a recipe, you might write:-

1. TIP IN EGGS, BUTTER, SALT, PEPPER.
2. COOK, TILTING PAN TO SOLIDIFY RUNNY BITS.
3. TIP OUT ONTO PLATE.

Many computer programs do no more than that! Input some information, cook it in some particular way and output the result for us to use. Using BASIC, you are now going to program the VIC in exactly this way.

As with cooking an omelette, let's assemble the bits and pieces required.

First we better decide what to 'cook'. Let's keep it simple. How about working out how many Miles Per Gallon tahe family car is doing?

Assemble the cooking utensils! Well, the television screen can display the answer of so many Miles Per Gallon. We can use the VIC keyboard to enter the information required to work out the answer, and the VIC itself can do the cooking!

Now, what information are we going to cook? The reading on the car milometer now and the reading when we last put in petrol? And we must know how many gallons have been used.

Miles Per Gallon?

Next, write down the steps for getting from the miles and gallons information to the required answer.

1. TYPE IN MILES NOW ON KEYBOARD
2. TYPE IN MILES BEFORE ON KEYBOARD
3. WORKOUT MILES DONE (= MILES NOW - MILES BEFORE)
4. TYPE IN GALLONS USED ON KEYBOARD
5. WORKOUT MILES PER GALLON (= MILES DONE DIVIDED BY GALLONS USED)
6. DISPLAY MILES PER GALLON ON SCREEN

If you had been asked to write down these steps without looking at the article, you would probably have written down the same actions using different words. For example, for step 3, you might have written:-

3. SUBTRACT MILES BEFORE FROM MILES NOW TO GIVE MILES GONE

It would be possible for a VIC to be designed to understand such a sentence and all its variants but it would be a very large and expensive computer, and it would take a long time to work out the result. As a compromise, BASIC allows you only a limited number of 'action words' (less than 60) such as those required for getting information from the keyboard and displaying the result. We shall use three of those action words, or KEYWORDS in this article:-

INPUT Obtains information from the Keyboard

PRINT Displays information on the T.V. screen

LET Calculates a value and keeps it

We will also use three BASIC words which tell you the VIC to do something with the instructions we give it. They are:-

NEW Tells the VIC to clear out any previous program instructions - ready to start a new program.

RUN Tells the VIC to carry out the instructions it has been given - i.e. 'RUN' the program.

LIST Tells the VIC to display the list of instructions on the screen.

VIC HINT : THE CURSOR

The "Cursor" is the name for the flashing square that appears on VIC's screen and moves about as you type in text.

Its purpose is to show you where the next character you type will appear and in that sense it works like a typewriter. When a program is running the VIC is busy and can't accept any information so the cursor disappears. It reappears when it needs some value to be input, and when the program is finished - in which case a READY message is printed also.

On the bigger computers you can still give the information and results you create any names you like, such as GALLONS USED and MILES NOW, but on micro computers, smaller names have to be used. On the VIC they consist of one letter, (e.g. F), or a letter followed by another letter, (e.g. XD), or a letter, followed by a digit, (e.g. G9). Usually a programmer will 'abreviate' so that GL might be Gallons Used etc. Note: there are a small number of combinations you can't use (such as IF, TO and ON) because they mean something special to BASIC. For each name the VIC reserves space in it's memory. It's as though it allocates some boxes into which you can put anything you like. Because the boxes are electronic, anything you put in replaces what was there before. If you take something out however, you take out an exact copy of what is in the box, leaving the contents in tact. I wish my bank account worked like that! Because the contents of the boxes vary as the program proceeds, they are called variables.

Your first program

Enough of the theory. Lets do something! Have a look at the following BASIC program and compare it to the set of instructions for calculating miles per gallon above.

```
10   PRINT "MILOMETER READINGS"
20   PRINT "NOW"
30   INPUT MN
40   PRINT "BEFORE"
50   INPUT MB
60   LET MD = MN-MB
70   PRINT "GALLONS"
80   INPUT GL
90   LET MG = MD/GL
100  PRINT "MILES PER GALLON"
110  PRINT MG
```

The first thing you can note is that five boxes or variables are used and can be perceived as:

MN: Miles Now

MB: Miles Before
MD: Miles Done
GL: Gallons Used
MG: Miles per Gallon

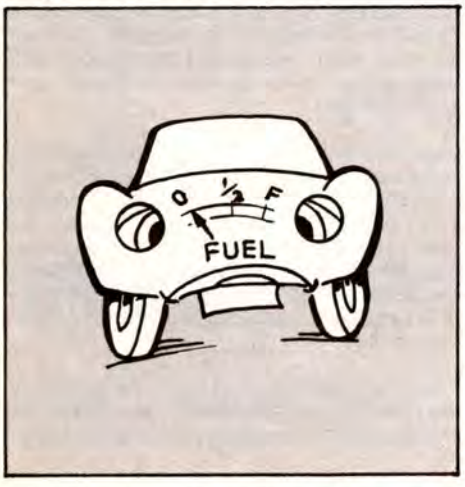
Lines 10 and 20 display the messages "MILOMETER READINGS" and "NOW" on the T.V. screen.

Line 30 causes a question mark to appear on the screen and the flashing cursor to appear. The VIC waits while you type in a number on the keyboard. You tell the VIC to carry on by pressing the RETURN key after you've typed in your number. (Return control to VIC). The VIC reads the number you've typed and puts it into a box (in it's memory cells), labelled MN, (for miles now).

Line 40 and 50 do the same as 20 and 30, but the message displayed is "BEFORE" and the value you type in goes into variable MB.

Line 60 causes the calculation 'value in MN - value in MB' to be done and the resultant value put into variable MD. You can read that line as:-
"Let value in MD become value in MN - value in MB"

Lines 70 and 80 do as 20 and 30 but put the gallons value in GL.



Line 90 calculates MD divided by GL (/ means divide in BASIC, * means multiply) and puts the result into MG.

Lines 100 and 110 display the result on the screen.

In other words, this list of BASIC instructions carries out the miles per gallon calculation in the way we first described it, using some of the limited set of KEYWORDS available and two letter names for the variables. Provided you stick to the KEYWORDS and variable names and don't add anything extra, (the VIC probably won't understand it and will complain with SYNTAX ERROR), then you too can write VIC programs in BASIC.

Typing it in

To give the VIC this list of instructions you must type them into the VIC's memory. After switching on, start off by typing NEW, which the VIC understands as an instruction to clear out any previous program it is holding in the memory.

Then provided you start each line with a number, VIC will assume it is a program instruction and enter it into the list of instructions to be obeyed. VIC keeps the lines in sequence according to the number at the start of the line. You can enter lines out of sequence and VIC will place them in the right place. You will see that I numbered the lines ten apart. Doing this allows you to squeeze in extra lines if you want to improve your program later on. If you find a mistake in a line you can replace it by typing a new line (correctly), with the same line number.

Type in the lines of the Miles per Gallon problem, ending each line by pressing the Return key to tell the VIC to store it away. If you don't press the Return key, the VIC won't store the instruction line, even though it is displayed on the screen when you type it.

Some of the lines (10, 70 & 100)are longer than 22 characters and so will not fit on a single line of the screen. Don't worry - when you get to the right hand edge of the screen, VIC automatically starts a new line but remembers that what you are typing follows on from the line above.

If you have a long program to type in, when the typing gets to the bottom of the screen, the top lines will scroll one by one off the top as you type in extra lines. You can look at any part of your program by typing LIST followed by the first line number you want to look at, then a dash followed by the last line number you want to look at. For example, you could type:-

LIST 20 - 50

and lines 20 to 50 inclusive will be displayed.

Once you've entered the program, you must tell the VIC to obey the instructions. The BASIC word is RUN. And because it's a general command and not part of the program, you can type it in without a line number, although you still need the Return key after the N.

If you made a typing error in the program line, you may see the words SYNTAX ERROR IN LINE followed by a line number. Display the line on the screen by typing LIST followed by the line number. See if you can spot the error. Whether you can or not, type in the line again, correctly, including the line number, and press RETURN at the end, then type RUN again followed by Return to try out the program once more. This immediate correction of errors is a special feature of BASIC which makes it a particularly good language for beginners.

Assuming the program has been typed in correctly, what will happen?

Getting Results

Here is a copy of what appeared on the screen when I ran the program and typed the information. The bits I typed are in bold type although they don't appear like that on the screen.

```
MILOMETER READINGS
NOW
? 10250                    Note: I had to press
BEFORE                     the Return key after each
? 10000                    number I typed in.
GALLONS USED
? 10
MILES PER GALLON
25
READY.
```

Ready indicates that VIC has finished what it was doing and is ready for some new instructions. RUN the program several times with different values for NOW, BEFORE and GALLONS USED. You will see that the same program can be used for different sets of readings. In other words it is a general purpose Miles Per Gallon program! If by accident, you typed in some letters instead of digits, the VIC will have displayed the message "REDO FROM START", put a question mark on the next line and awaited a number response.

We'll see next time how to INPUT words, and also how to use the VIC to make decisions. But until then, here are two problems for you. The first is to work out what the following program does, then type it in and see if you were right!

Homework!

The other is to prepare, enter and run a program which asks you for your calorie consumption for each day, Sunday to Saturday, and then tells you what your average daily calorie consumption is.

Who said they could never program!!!

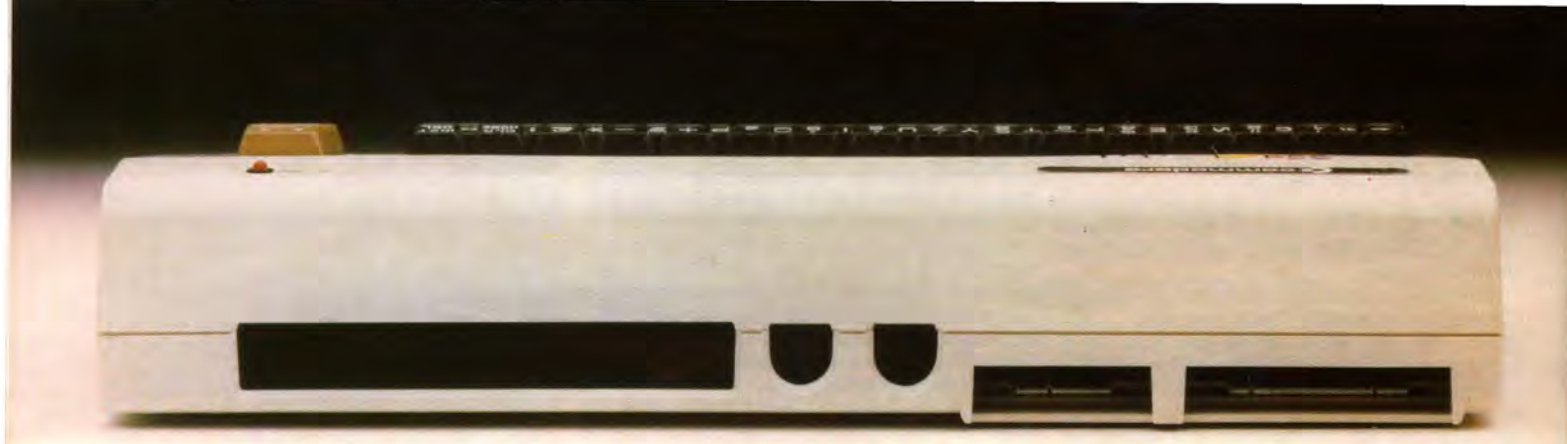
```
10   PRINT "WIDTH OF ROOM IN FEET"
20   INPUT RW
30   PRINT "LENGTH OF ROOM IN FEET"
40   INPUT RL
50   LET AR = RW*RL/9
60   PRINT "HOW MUCH IS THE"
70   PRINT "CARPET  PER  SQUARE
YARD"
80   INPUT CP
90   LET TC = AR*CP
100  PRINT "THE TOTAL COST OF"
110  PRINT "CARPETING THE ROOM IS"
120  PRINT TC
```

VIC HINT: THE RETURN KEY

One might justifiably say that the key marked 'RETURN' is the most important one on VIC's keyboard. It has two functions: the first is like the Return key on a typewriter. When you press VIC's Return key, the Cursor (which shows you where you are typing) finishes a line and jumps to the start of the next line.

The different feature, however, is that pressing this key 'returns' control to the VIC. In other words, it is only after pressing Return that VIC can get on with 'understanding' and then executing what it is that you have typed.

There are three cases when this is used. First, after you have typed in a command such as NEW, LIST or RUN which is to be executed immediately. Secondly, on entering a new line of a program which is always preceeded by a line number - pressing Return will store the line though not execute it. Thirdly, when you are feeding information into a program that is running - by means of INPUT. A question mark appears - indicating that you should type your answer. The program will only continue once you have finished typing and pressed Return.



EXPANSION OF THE VIC.....

The VIC was designed as a self-contained computer; that means all you need is a T.V. set and a mains socket to start using VIC for writing programs. Add a cassette unit and you have a system for storing the programs you have written for it. But the list of what you can add to VIC doesn't stop there. Indeed from the expansion point of view the VIC is one of the best planned microcomputers available.

To give you a better idea, of what VIC can be made to do we offer a whistle stop tour of the various expansion methods and ports, followed by a preview of what's likely to be available to fit on and in them.

We'll start with a look at the outside of the VIC - at the various slots and protrusions in evidence at the back and at the right hand edges of the casing. Starting at the side, there is, not unexpectedly an on-off switch and a power connector for the external mains transformer included with each VIC. The main feature of interest is the front most socket which is a standard nine-pin connector. Known as the Game Port, it is where you connect game paddles, a joystick or even a Light Pen - used both for games and selecting displayed options in Educational and Business programs. Though it seems unlikely that these devices will have as high a priority in Commodore's development department as, say, disk drives, VIC users will be interested to know that the units made by Atari are entirely compatible!

Moving round to the back panel, the large slot on the left (see top photo) is called the Memory Expansion Port. The connector is well recessed into VIC's housing, in order to support cartridges that are slotted into it.

These cartridges contain extra memory for the VIC - either permanent memory (ROM) or memory that may be used for storing your own programs (RAM). The former will include ready-written software, so that when you buy Space Invaders, Teach Yourself Maths or whatever, it will come in the form of one of these cartridges.

Since it is quite likely that you will want to add several cartridges concurrently (whether software or units of RAM expansion), a motherboard will be produced which 'reproduces' the expansion slot six times over. Of much the same size as the VIC itself, this motherboard will also accept additional interfaces for the VIC. Ultimately this will include the IEEE-488 which would enable VIC to address scores of laboratory instruments as well as peripherals for the Commodore Pet.

Next to the Memory Expansion slot come two DIN-type connectors which are commonly used on Hi-Fi and audio equipment. The left hand one is actually the audio and video output that feeds to the loud speaker and screen of your T.V. set. Between the T.V. and VIC is a small device called an R.F. Modulator which is needed to convert the electronic signal to the same format as would normally come from the aerial socket. VIC can be used without the modulator to drive the Video Monitors as sometimes found in schools and laboratories, but seldom in homes.

The second DIN socket is a special serial interface for driving the Commodore VIC peripherals such as Floppy Disk and Printer. Whether you know a lot about interfacing or nothing at all, it is difficult for us to explain exactly how this works since it is not a standard configuration.

Finally we have two "edge connectors" at the right hand end. The smaller one is for the VIC Cassette deck and is used for storing and loading programs. Incidentally, the connector is identical to the PET cassette which you might be able to borrow from an affluent micro-enthusiast colleague who has now upgraded to disks!

The larger of the connectors is called the User Port because it is up to you what you use it for! It contains 8 data lines with two handshakes - which in English means you should be able to control anytning from a non-standard printer to your model train set.

However, part of VIC's operating system converts this port into an RS232 interface which is the accepted Standard for many correspondence quality printers, and also for modems - which are used for sending information over the telephone. Though the physical connector is not standard (a low-cost plug-in convertor will be available, soon) all the standard communications functions are incorporated.

So much for the guided tour - you should by now have a picture of what VIC's potential is. But what add-ons can we really expect to see for the VIC?

The first will almost certainly be a printer. Commodore are being customarily tight-lipped about its specification but it would appear to be an 80-column dot-matrix 30 characters per second. This can be used for listing programs (in which case the 80 column width overcomes the 22 column screen restriction) or as the output from a simple word processor for notes and letters. *VIC COMPUTING* estimates that the price will be in the £200 to £250 range.

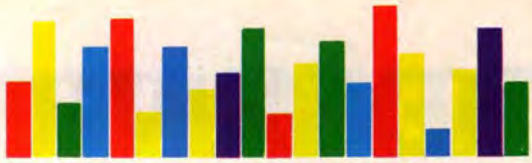
The floppy disk unit, also likely to be relatively low-priced, will be used for storing and retrieving large numbers of programs at high speed. Total capacity will be 140K bytes per disk, which is nearly 40 programs of 3.5K each. It can also be used for storing data - such as names, addresses, telephone numbers and account details. Because the unit will contain its own microprocessor and operating system, the action of buying a disk will not necessarily require you to add more RAM - although that would certainly increase the power of your system.

Finally, there is the Modem which connects your VIC to the telephone line for sending data to other VICs, other computers or accessing Public Information services like Prestel. The applications of such a system include Electronic Mail and the exciting new area of Community Services.

Quite when these peripherals will appear and in what order is as yet unknown.

Watch this space!

USING GRAPHICS



By Mike Gross-Niklaus

This article describes some of the techniques I use for colour programming in VIC BASIC. None are particularly clever, but they do provide a set of formulae which save you having to work out every time all the various POKE locations and values and colour control codes.

The Boarder and Background Colours.
On the VIC, you can set the border colour (around the text area) with any of 8 colours. Likewise the background (behind the text) can be set with any of 16 colours. The colours are specified by bit patterns in the colour control register at location 36879. You can chose the appropriate value for the pair of colours you want from the following table.

	Border							
Screen	BLK	WHT	RED	CYAN	PUR	GRN	BLU	YEL
BLACK	8	9	10	11	12	13	14	15
WHITE	24	25	26	27	28	29	30	31
RED	40	41	42	43	44	45	46	47
CYAN	56	57	58	59	60	61	62	63
PURPLE	72	73	74	75	76	77	78	79
GREEN	88	89	90	91	92	93	94	95
BLUE	104	105	106	107	108	109	110	111
YELLOW	120	121	122	123	124	125	126	127
ORANGE	136	137	138	139	140	141	142	143
LT. ORANGE	152	153	154	155	156	157	158	159
PINK	168	169	170	171	172	173	174	175
LT. CYAN	184	185	186	187	188	189	190	191
LT. PURPLE	200	201	202	203	204	205	206	207
LT. GREEN	216	217	218	219	220	221	222	223
LT. BLUE	232	233	234	235	236	237	238	239
LT. YELLOW	248	249	250	251	252	253	254	255

Suppose you want Black background and Black border then the code will be:-

```
POKE 36879,8
```

While for a green background with purple border (Ugh!) the code is:-

```
POKE 36879,92
```

By setting bit 3 of the same register from a one to a zero, you can reverse for each character position the colour of the graphic and that of the background. For example if you have a green A followed by a red B on a blue background, then by flipping bit 3 of location 36879, you will see a blue A on a green background next to a blue B on a red background. This colour reversing is very effective for portraying the moment when your Starship gets clobbered by yet another Klingon laser beam. I use code along the following lines:-

```
100 FOR I = 1 TO 20 : GOSUB 5100 : NEXT I
110 . . . .

5100 REM FLIP SCREEN COLOURS
5110 CC = 36879 : CP = PEEK (CC) : CR = CP AND 8
5120 IF CR = 0 THEN CP = CP + 8 : GO TO 5140
5130 CP = CP - 8
5140 POKE 36879, CP : RETURN
```

Printing Strings in Colour
Printing strings in colour is achieved by printing a control character for a particular colour at the start of the string. The control character doesn't put anything on the screen, but all characters printed from then on will be in the specified colour until you print a different colour control character. You can program the control characters by including them in the string in the same way as cursor movement. But often you want to print the same string in different colours at different points in your program. You can do this by printing say:-

```
PRINT CHR$ (CL) ; Z$
```

where CL is the ASCII value of the appropriate control character and Z$ is the string you want printed. ASCII values are as follows:

144	BLACK	156	PURPLE
5	WHITE	30	GREEN
28	RED	31	BLUE
159	CYAN	158	YELLOW

Alternatively, you can set up a string, say CC$, containing just the eight colour control characters, and select the one you want with:-

```
PRINT MID$ (CC$,CL +1,1) ; Z$
```

Screen Colours With Poke
There are occasions, typically in Race 'n Chase games, where the graphics are best poked to the screen rather than printed. The VIC maintains a colour map which parallels the screen character map. For each screen location there is a corresponding colour location. The screen memory starts at 7680 and the corresponding colour map at 38400. When you fit expansion memory onto the VIC, the locations of screen and colour memory change, so use variables for the two locations in the body of your program with their initial assignment at the start of the program. That way changing the program to cater for the different configuration is no hassle. If you want to POKE a red A into row R, column C of the screen, then you need code along the following lines:-

```
POKE SC + 22*R + C, 1 : REM CAPITAL A
POKE CM + 22*R + C, 2 : REM RED
```

where SC is 7680 and CM is 38400, and the Rows and columns are numbered from zero onwards rather than 1.

Intelligent Use of Variables
It's a good habit to allocate variable names in some ordered and meaningful way. You may have noticed that all the colour variables in the above example start with C. If you can you should standardise your allocations for all your programs. Just as RE$, QU$, SP$ and VT$ have for some time now meant to me - RETURN character, QUOTE character, Space string and Vertical Tab string - so now I recognise CL for colour value, CB and CG for border and background colour and so on.

Finally, here's a short program that demonstrates several of the techniques mentioned - to produce a random histogram on the screen.

```
 5  REM COLOUR HISTOGRAM
10  POKE 36879, 190
20  CL$ = "■■■■■■■■": REM COLOUR CONTROL
    CHARACTERS
30  VT$ = "hms ↓ 22 ↓ ": REM CURSOR DOWNS
35  DEF FNR(X) = INT (RND(1)*X) +1
40  PRINT "cls RANDOM HISTOGRAM"
50  FOR I = 1 TO 20: REM 20 COLUMNS
60  PRINT VT$; TAB (I-1); "rvs";
70  PRINT MID$ (CL$, FNR(8), 1);:REM RANDOM COLOUR
60  FOR J = 1 TO FNR(18)
90  PRINT " ↑ ← ";
100 NEXT J,I
110 GO TO 110: REM AVOIDS MESSING PICTURE
```

DEAR VIC....

How can I use Machine Code on my VIC?

Our resident genius replies:

The VIC is based on the 6502 microprocessor, like so many other microcomputers - which means that there is a lot of machine code software which will run on it.

The trouble is that unlike its big brother, the PET, VIC has no built in Machine Language Monitor. The latter is itself a machine code program which can display the contents of any block of memory, allow you to modify those contents and hence enter machine code. It also has a number of other features, such as the ability to load and save sections of memory onto tape and disk.

A cassette-loaded Monitor will probably be one of the first utility programs on sale for VIC. Trouble is, when you mess about with machine code, you are very prone to "crashing" the system. Since the only remedy here is to switch off and on again, you have to reload the monitor before attempting to debug the code.

The solution to this will be a Machine Code programmers ROM Cartridge - containing a resident Monitor plus Assember/Disassembler for writing Assembly Language, and some more RAM in which to hold your routines.

Meanwhile, you are going to have to load Machine Code *through* BASIC by means of PEEK and POKE statements; these deal with decimal values and byte addresses

A useful convention is to store the start address in a data statement, following by the number of bytes of code, followed by their actual values, e.g.:

```
100 DATA 826, 5, 92, 88, 172, 240 60
```

A short program to load this routine would look something like this:

```
10    READ SA, NB
20    FOR I = SA TO SA + NB-1
30    READ VL
40    POKE I,VL
50    NEXTI
60    END
100   DATA. . . . . .
```

Please note that the example used is not a real machine code program - though I will be passing on a few in future issues - so if you don't want to experience your first crash, steer well clear of Machine Code at first.

continued

BUT WHAT CAN IT DO?

room learning or tuition - but it does provide fun and powerful assistance if you want to progress faster or need extra help.

And so to applications that can pay-their-way in the home. This idea has been somewhat abused in the past - most frequently by enthusiastic husbands justifying a microcomputer purchase to sceptical wives on the grounds that it can do recipies or somesuch.

It is true, however, that your VIC can be used for storing useful information such as telephone numbers, insurance details or (at a stretch) recipes! If you intend to keep and manage a lot of this sort of data then a floppy disk would be essential - since the cassette is very slow to locate individual items.

VIC comes rather more into its own as an advanced calculator - which can prompt you with messages and display results graphically and in glorious technicolour. Apart from the dubious attraction of showing your bank balance changing from black to red, the VIC can actually be quite useful for plotting a monthly breakdown of household expenditure or cost of running your car. On the other hand you maybe appalled by the results. Though by no means essential, a Printer comes in handy here.

A trap to avoid - don't be mislead by all those ill-informed T.V. programmes that insist you should use your microcomputer to control the home heating! It is a nice idea in theory, but totally impracticable with the kind of heating installations in most housing. Nevertheless, if you are the type who enjoys experimenting then the VIC has plenty of interfaces for connecting to light bulbs, Hi-Fi systems, vibrating beds and the like.

One question remains unanswered - how is VIC to achieve all this? The answer is software: the name given to ready-written programs to perform a series of tasks. At the time of writing, software for VIC is still in its infancy, with many packages under development, and a number of key suppliers limbering up to support the VIC with cassettes and cartridges. Expect them in the High Street shops soon.

VIC Computing will review the best of them in the next and subsequent issues.

CONVERTING SOFTWARE FOR VIC

This article is aimed at the slightly more technical end of our audience and in particular at the thousands of Commodore PET Users who will now be buying VIC's for use at home. In it we aim to give a broad outline of the internals of the VIC, and how it compares with the PET. As space is short, we will concentrate on items which are not covered in the VIC manual.

By Chris Preston

Those of us (the ones with long white beards) who remember all that time ago when Commodore brought out the 'New ROM' PET will also remember the horror when we found out that all the ROM subroutines and zero-page locations had changed, wrecking our trusted 'Old ROM' programs. Well the bad news is it has happened again! This time however, VIC COMPUTNG has come to the rescue, with just about all the information you are likely to need to understand the new system, whether or not you are an existing PET addict, and to convert your existing programs to the VIC. A further, in-depth article will follow in a later issue but we hope that this short review will whet a few appetites and prevent some suicides. The material to be covered here mainly concerns the operating software. The hardware in the VIC really needs an article to itself, as there are several devastating changes, such as the switch from a parallel IEEE port to a serial one (to cut down on the cost of the interfaces) and the addition of various extras such as a Game I/O port for joysticks, a light pen interface and an audio/video output. In fact the only port which is unchanged is the cassette deck!

The Screen Display

The Screen Display memory on the VIC is divided into two parts. The first runs from 7680 to 8191 (that's 1E00 to 1FFF in hex) and the values in this area determine the character displayed. This means that to POKE a moving character on the screen requires two POKEs, which obviously slows the program down a bit.

Zero Page and other locations

As mentioned above, the VIC's zero page area is totally different to that on existing PETs. The following table covers the most commonly used locations on zero page, which should allow programmers to convert their favourite PET programs to the VIC. The memory locations are given in decimal first then hex. Those in the PET column refer to the 'New ROM' series, that is 3000, 4000 and 8000 machines.

VIC	PET	Function
0-2	0-2	JMP instruction for USR function
43-44/2B-2C	40-41/28-29	Start of BASIC program
45-46/2D-2E	42-43/2A-2B	Start of variables
47-48/2F-30	44-45/2C-2D	Start of arrays
49-50/31-32	46-47/2E-2F	End of arrays
55-56/37-38	52-53/34-35	Top of memory
97-98/61-62	94-99/5E-63	Floating Accumulator 1
122-123/7A-7B	119-120/77-78	Pointer to BASIC Text (TXTPTR)
144/90	150/96	I/OStatus (ST)
158/9E	192/C0	Tape Error Counter
198/C6	158/9E	No. of characters in keyboard queue
204/CC	167/A7	Flashing Cursor Switch
209-210/D1-D2	196-197/C4-C5	Pointer to start of screen line
211/D3	198/C6	Cursor Column No
214/D6	216/D8	Cursor Line Number
243-244/F3-F4	- - -	Pointer to colour code for current char.
631/277	623/26F	Keyboard Queue
646/286	- - -	Current Colour
649/289	- - -	Maximum length of Keyboard Queue
650/28A	- - -	If > 128 then all keys repeat
788-789/ 314-315	144-145/90-91	Interrupt Vector

Monitor Subroutines

It goes without saying that all the ROM routines now start at different locations; you cannot change anything without altering those addresses! However large parts of the ROM are unchanged apart from being moved around. The BASIC is almost identical to BASIC 2 apart from the new graphics commands. The screen, keyboard and cassette routines are almost identical, although the IEEE routines are completely different owing to the new serial bus. The keyboard decoding is completely different; you can no longer test for the SHIFT key. The old jump table from locations $FFC0 to $FFEA is unchanged, although it has been extended, and there is a new jump table in RAM from $300 to 332. In fact most of the old jump table entries point into this new RAM table, so 'fiddling' the I/O system will be a lot easier. One item of bad news is that it is no longer always possible to disable the STOP key by means of a single POKE instruction. The method works for the basic VIC, but not with some of the add-on cartridges. The addresses given below are all in hex and we have not given the corresponding PET addresses as these vary between BASIC 2 and 4.

VIC	Function
$73	CHRGET.
$76	CHRGOT.
$C474	BASIC warm start.
$D1BB	FLTINT.
$D391	INTFLT.
64802/$FD22	RESET (cold start).

We hope that this quick review of the VIC's operating system has removed many of the worries facing prospective buyers. We will be publishing more detailed information in the future, though the Commodore documentation looks like being a lot more comprehensive and comprehensible than in the past.

DATA SHEET

The VIC Sound

Built into its hardware, VIC has a sound generation system, with the output being fed into the audio channel of the TV set. This means that pleasant sounds and realistic effects may be produced with a minimum of user programming.

Three independent tone generators may be activated and mixed - each with a frequency spectrum of 3 octaves. Thus harmonies or 3 voice music are possible, simply by changing the frequencies of the generators. In addition, the total volume may be software controlled, plus there is a "White Noise" generator. The latter is used extensively in game sound effects - lasers, explosions, etc. In combination, it should be possible theoretically to produce a very crude form of speech synthesis - two tones defining a vowel, with White Noise for Fricatives such as "S" and "F".

Control is achieved by POKEing to registers within the VIC chip as follows:-

Decimal	Hex	Function
36874	$900A	Tone Generator 1: 0 gives no sound. 128-255 gives a tone (see below).
36875	$900B	Tone Generator 2.
36876	$900C	Tone Generator 3.
36877	$900D	This is a form of tone generator but with White Noise added, so that the output varies from an Aeroplane Buzz to whistling wind.
36878	$900E	Volume: 0 gives lowest volume and 15 the highest. Note that the volume control on the TV set must also be turned on.

Using the Sound Generator

Though sounds may be thus generated using simple POKE commands in live mode, the most common method will be from a BASIC program. Notes may be read as text strings from Data statements, converted to POKE values, and then activated inside FOR-NEXT loops to time the notes correctly. Alternatively, a small program may be written to GET characters from the keyboard - so that the VIC could simulate a simple 3-voice electronic organ! Doubtless, also, VIC users will build up a massive library of subroutines to generate special effects and games sounds.

Piano Program

The following is a useful program for converting VIC into a piano keyboard. The keys 1 to 8 provide the notes of the scale, with 0 or 9 halting the program.

```
10   S2=36587
20   V=36878
30   POKE S2,0
40   POKE V,3
50   FOR N=1 TO 8
60   READ A(N)
70   NEXT N
80   GETA$:IF A$="" THEN 80
90   N=VAL(A$)
100  IF N=0 OR N=9 THEN 200
110  POKE S2,0
120  FOR T=1 TO 25:NEXT
130  POKE S2,A(N)
140  GOTO 80
200  POKE S2,0
210  END
300  DATA 223,227,230
310  DATA 231,234,236
320  DATA 238,239
330  REM NOTES FOR B MAJOR SCALE
```

Note	POKE Value	POKE Value	Note
C#	240		
		239	C
		238	B
A#	237		
		236	A
G#	235		
		234	G
F#	232		
		231	F
		230	E
D#	228		
		227	D
C#	225		
		223	C
		221	B
A#	219		
		217	A
G#	215		
		213	G
F#	210		
		207	F
		204	E
D#	201		
		198	D
C#	195		
		191	C
		187	B
A#	183		
		179	A
G#	174		
		170	G
F#	164		
		159	F
		153	E
D#	147		
		141	D
C#	134		
		128	C

HSV Limited
Microcomputers

May Place,
Basingstoke RG21 1NX
Tel: 0256 62444
22 Southampton Street,
Southampton SO1 2ED
Tel: 0703 331422

B&B [Computers] Limited

The Consultants for the North West

Your VIC from B&B can communicate with Viewdata if required

SUITE 1,
124 NEWPORT STREET,
BOLTON BL3 6AB.
LANCASHIRE.
Tel: (0204) 26644, 382741.

Advanced Computer Equipment (Leeds) Ltd

MICROCOMPUTER SYSTEMS

95 MEADOW LANE LEEDS LS11 5DW · Telephone: 0532-446960

VIC 20 NOW IN OUR SHOWROOMS

TYPEWRITER CENTRES

for all your VIC requirements - branches throughout the MIDLANDS and BOURNEMOUTH

PET specialists - COMMODORE appointed dealers

BIRMINGHAM		
City Centre:	52 Bromsgrove Street	021 622 5385
Edgbaston:	92 Dudley Road (opp.hospital)	021 455 9111
Erdington:	52 High Street (opp.library)	021 382 0185
Kings Heath:	46 High Street	021 444 7439
Solihull:	54 High Street	021 704 1260
Sutton Coldfield:	Knights House, Gracechurch Centre	021 355 6789
LEICESTER:	34-36 Rutland Street	0533 538745
WALSALL:	7 Leicester Street	92 27589
WOLVERHAMPTON:	Worcester Street	90 27627
COVENTRY:	128 Far Gosford Street	0203 28799
BOURNEMOUTH:	75 Seamoor Road, Westbourne	0202 766997

---

robox

The Scottish Microcomputer and Calculator Centre

COUNT ON US

Units 14 & 15,
Anderston Centre,
Argyle Street,
Glasgow
Tel: 041 221 5401/2

HOLDENE LTD.

See
VIC 20
at

MANCHESTER UNITY HOUSE,
11/12 RAMPART ROAD, LEEDS
Tel: (0532) 459459 TWX: 556319

Wego Computers Ltd

22a High Street,
Caterham, Surrey, CR3 5UA

Tel: Caterham (0883) 49235
Telex: 933660 WEGO G

Currie & Maughan
MICRO COMPUTERS
204 DURHAM ROAD,
GATESHEAD.

TEL. GATESHEAD 774540

Yorkshire Electronics

COMMODORE APPOINTED
COMMERICAL SYSTEMS DEALER

NEW SHOWROOM NOW OPEN

CAXTON HOUSE,
17, FOUNTAIN STREET, MORLEY,
WEST YORKSHIRE. Tel: 0532-522181

EYS

ANY TIME NOW IT WILL BE WORTH BUYING A HOME COMPUTER.

You haven't seen a real home computer until you've seen the VIC 20 by Commodore.

But by August you'll get your chance.

Because that's when the first VIC 20's will be arriving at your Commodore dealer.

Then you can take a good, long look at what makes VIC the best.

Like the real typewriter keyboard with full graphics.

And eight background and 16 foreground colours.

And music in three voices and three octaves, as well as language and sound effects.

So don't think of buying a home computer until then.

Because it simply isn't worth it.

commodore COMPUTER

Commodore Home Computer Division, 818 Leigh Road Trading Estate, Slough, Berks. SL1 4BD. Telephone: Slough (0753) 74111.