ELECTRON

With the extensive range of software currently available for the BBC Micro, the Electron, at under £200, sounds like a wise investment. Steve Mann puts Acorn's new machine through its paces.

It's an exaggeration — but not too much of one — to say that the BBC Model A is the Ford Edsel of home computing. For those of you that don't know your history, let me say the the Ford motor company was convinced that the Edsel (named, incidentally, after Henry Ford II's father) was a sure-fire smash — not ultra-cheap but bringing the qualities of much more expensive cars into the mid-range market. The car was a lemon — it sold hardly at all and for years provided a generation of stand-up comedians with comic fodder. This was a bit unfair — it was certainly no Rolls-Royce but by all accounts the Edsel wasn't *that* bad.

When the BBC decided that its viewers should become computer-literate, it shopped around for a computer to go with the television series *The Computer Programme* and plumped for Acorn as the manufacturer. The machine that Acorn came up with — which became known as the BBC Model B — was designed with expansion in mind: extra processors could be hooked on and there was a whole host of spare sockets for additional ROMs, together with expansion ports and various other channels for communicating with the outside world. Of course, all of this cost money and the BBC was not a little worried at the thought that it could be pricing itself out of the lower end of the market. To rectify this situation, the Model A was designed. This was basically a chopped-down Model B, with all the B's features but minus the expansion facilities. It was £100 cheaper and, not surprisingly, was expected to be the bigger seller of the two computers. That was the theory, anyway. In fact, almost everybody wanted the Model B. Acorn was not geared up to produce the large numbers of Model Bs that were required and the horribly familiar production delays ensued. The Model A was soon all but forgotten.

Now Acorn has come up with what at first sight appears to be a revamped Model A. Externally, the Electron resembles Acorn's earlier Atom rather more than it does the BBC, but the new machine is very, very similar to the Model A or B. Aimed squarely at the Sinclair/Oric/Dragon market, the Electron offers all the advantages of BBC Basic at a very competitive price and the fact that large numbers of potential users will be familiar with the BBC

machines through their use in schools, together with the vast amount of software developed for the BBC Micro and able to run with little or no modification on the new machine, means that the Electron has a head start in this most cut-throat of markets.

Hardware

Manufactured in creamy beige textured plastic and measuring 34cms by 16cms by 5cms, the Electron gives the impression of being all keyboard. It has a satisfyingly solid feel to it and generally gives the impression of being a very classy product indeed. Acorn has no need to resort to the misleading phraseology other manufacturers adopt in an attempt to disguise keyboard deficiencies — 'typewriter pitch', `full-size moving keys', 'ergonomic design', etc, etc, *ad nauseam* — the Electron has a normal keyboard with spacebar that is perfectly suitable for touch-typing. The standard qwerty layout is augmented by the usual control keys; the Electron closely emulates the BBC in these but, in the interest of space-saving, several of these have to double up — the BBC Micro's ten function keys are all here but they each have to share a key with a numeral and are accessed by the CAPS LOCK/FUNCT1ON key. SH1FT LOCK is missing but CAPS LOCK is present, with a small LED to indicate when the lock is engaged. All keys are plainly marked in black, and in brown on the front of most of them is the keyword that can be obtained when pressing the key together with FUNCT1ON. The user thus has the option of typing keywords out in full or entering them using just two keys.

Unlike certain other manufacturers, which let out a squawk and invalidate the guarantee if you even so much as *think* of reaching for a screwdriver, Acorn positively encourages users to delve about inside. The top of the case lifts off after removal of four screws and the ribbon cable that connects the keyboard simply unplugs. The Electron's interior is divided in two — a small compartment to the right houses the electronic odds and ends that regulate the power supply; everything, in fact, bar the mains transformer itself, which is housed in the mains plug. The Electron power supply is decidedly non-standard:

the transformer in the plug puts out 19 volts, which is further stepped down on board, giving 18 volts AC to the expansion port and +5, —5, and 0 volt lines to the PCB.
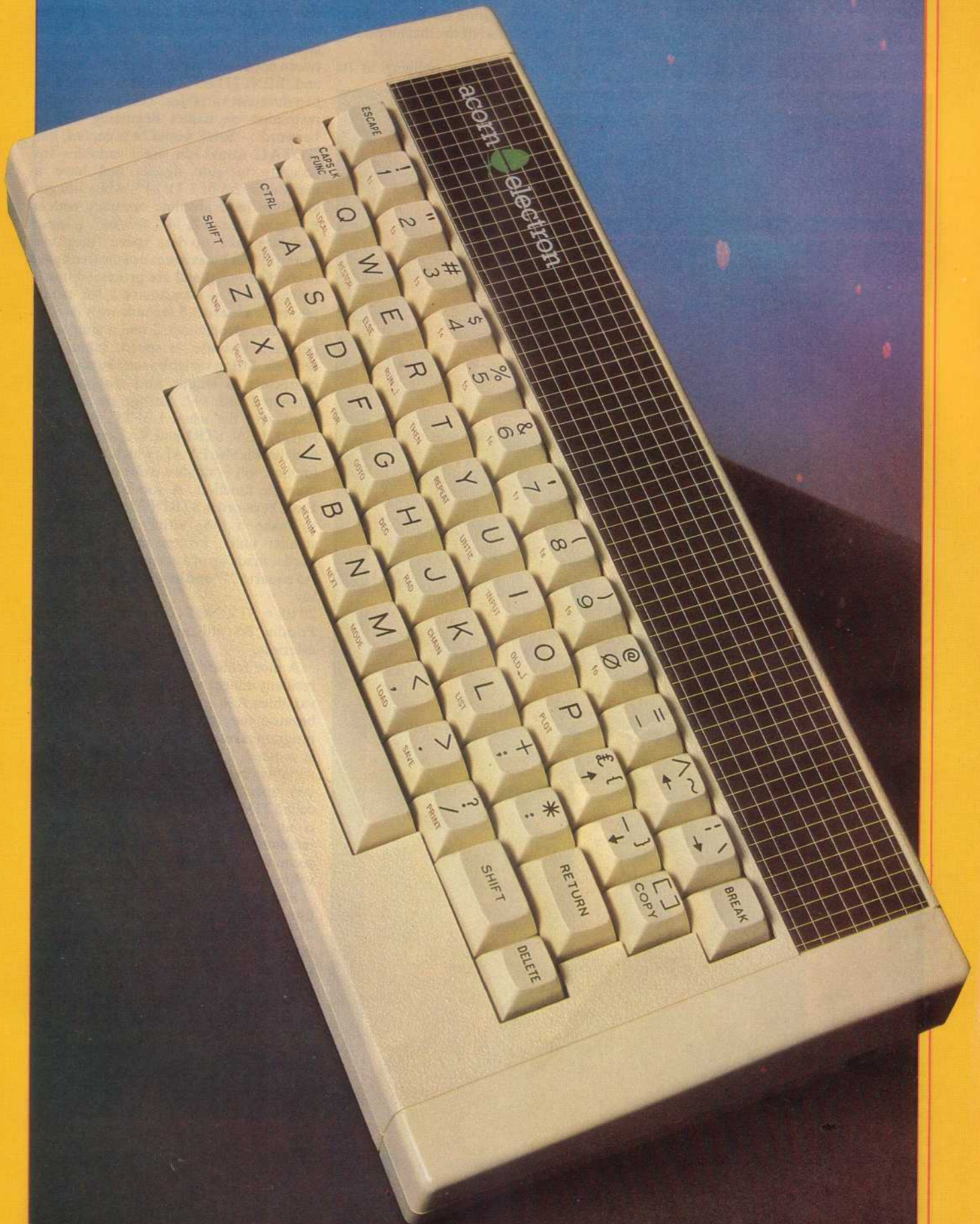
The board itself is beautifully laid out, with every component clearly labelled and plug-in connectors for the power lines and speaker leads. The processor is a 6502A running at 2MHz. It would appear that Acorn has been playing around with the Electron design right up until the moment it was launched — other review machines have apparently contained a single ROM chip in place of the BBC Micro's two, and have shown evidence of the odd patch on the board; *PCW's* version contained the full complement of ROMs and every component appeared to be in its final position with not a patch in sight.

At the rear of the Electron is a single expansion port — a gold-plated, Sinclair-style edge connector. To the right of the machine is the power lead socket; and on the left are sockets for TV, monochrome monitor (composite video input), colour monitor (RGB) and cassette. Sound comes from a small built-in speaker; volume, while hardly astonishing, is certainly more than adequate and can easily be amplified if desired.

Software

On power-up the screen displays the words ` Acorn Electron BAS1C' in white letters on a black background. The Electron runs BBC Basic, a very powerful 'structured' dialect supporting procedures and local variables.

When the Electron is switched on the computer is in Mode 6 with the CAPS LOCK on. Mode 6 gives 25 lines of 40 characters and uses 8k of RAM, making it the most economical of the Eleetron's seven modes. This is the major difference beween the Electron and the BBC Micro — the latter defaults to the teletext Mode 7: a mode which supports all the BBC's colours, uses standard teletext block graphics and which requires only 1k of user RAM. The teletext mode is something of an anomaly on the BBC Micro — it uses standard teletext control codes instead of BBC Basic's COLOUR, DRAW and MOVE commands and allows easy control of things like double-height text — but it is

The Electron: all the advantages of BBC Basic for under £200

excellent for programs that do not need fancy graphics but require large amounts of data. It is a pity that Mode 7 was not implemented on the Electron. As it is, in addition to the small amount of RAM needed for system 'housekeeping', the most economical mode on the Electron will take up a quarter of the machine's 32k of user memory.
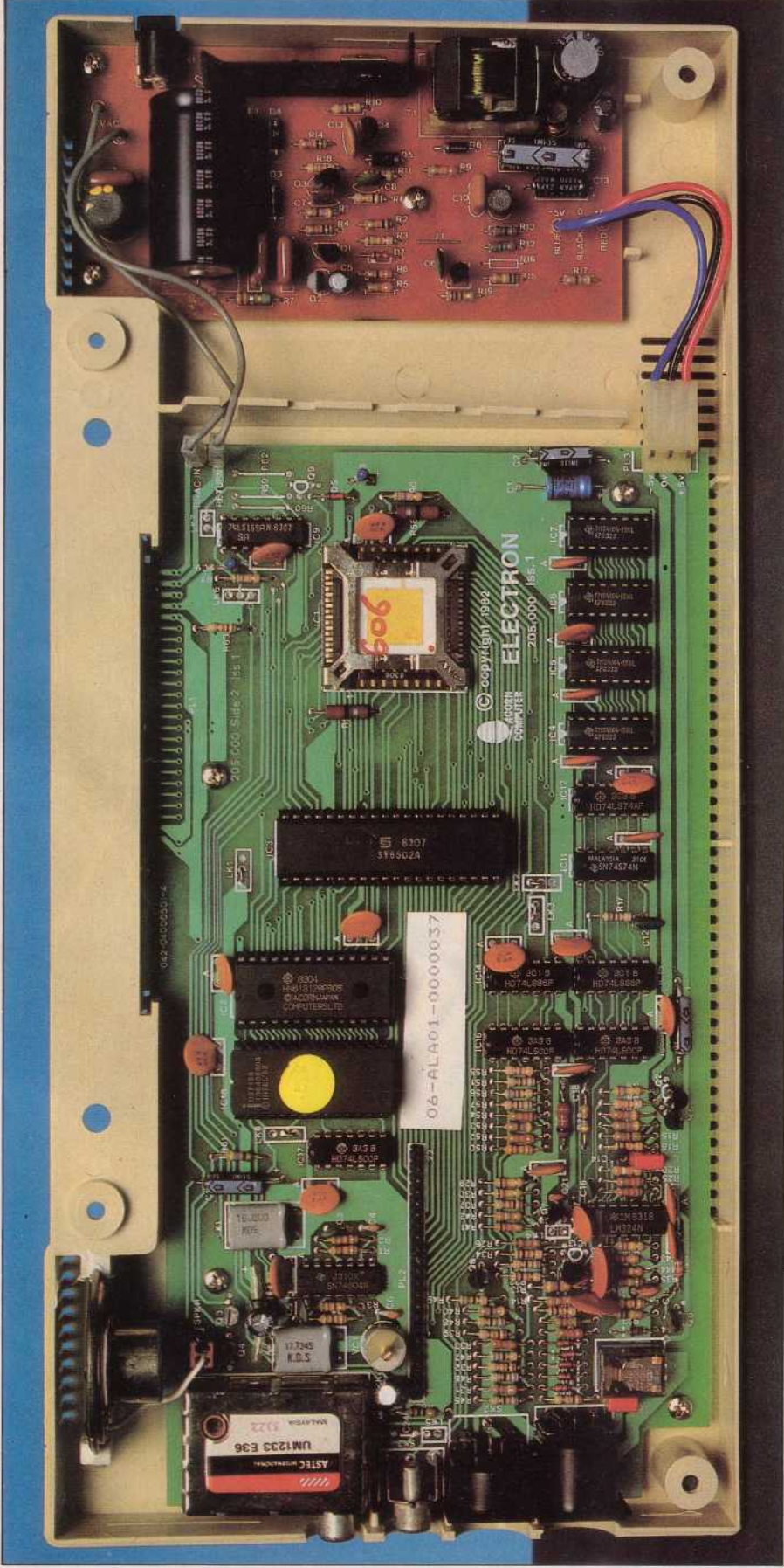
The Electron is very obliging in its acceptance of variable names. There are three types of variable — real, integer and string — and all the Electron insists on is that they begin with a letter, contain no spaces and do not start with a Basic keyword. The Electron differentiates between upper and lower case, so `benchtest' and `BENCHTEST' would be treated as two different variables. This gets over the restriction on names beginning with a keyword — 'total' would be accepted, but `TOTAL' would not. Real variables are accurate to nine figures and have a maximum size of 1.7 x 1038, while integer variables have absolute accuracy with a, maximum size of 2, 147, 483, 647. Integer variables take up less space (they are stored in four bytes instead of the five bytes needed by reals) and are processed much more quickly — Benchmark 1, for example, took a mere 0.5 seconds with integer variables; Benchmark 2 had a similar 50 per cent increase in speed, taking 2. 5 seconds. String variables may be up to 255 characters in length, and may be concatenated with ' . Strings are manipulated with the usual LEFT$, RIGHT$, MID$, LEN, STR$ and VAL commands, and there is also EVAL, which, as its name implies, will evaluate a string as if it were a numeric function. A somewhat unusual Basic keyword is STRING$, which is used to make a long string from repeated smaller units; for example, to print a border made up of asterisks, you would set A$ equal to " " and have a program line of, say,

20 B$ = STRING$ (20,A$)

Printing B$ will result in a string of 20 asterisks.

A nice feature of Electron Basic is the resident integer variables. The 26 integer variables A% to Z% are called resident because they are not cleared when the program is run or when NEW or BREAK is pressed. This allows the user to pass variables between programs. There is also one special resident integer variable, called TIME. T1ME, as its name suggests, is an elapsed-time clock; its value is incremented by one every hundreth of a second and it may be assigned a starting value or zeroed like any other variable.

Electron Basic derives its power from its use of procedures. These allow the longest of programs to be split up into easily managed sub-units which are labelled and may be called by name. This makes following the structure of a program—and debugging it — much easier than in a program which relies on unconditional

The board is beautifully laid out with every component clearly labelled

Sockets for TV, monitors and cassette

GOTOs and GOSUBs. Procedures are generally placed at the very end of a program, after the END statement, and are each given a name starting with PROC. It is good programming practice to use lower case when defining a procedure to distinguish the procedure's name from the word PROC, but the only restriction on procedure names is that they must not contain spaces _ the underline character is used to represent a space. All procedure definitions must end with ENDPROC. Variables may be defined as being local to a procedure or a function, and the user may have two variables with the same name coexisting quite happily in the same program, with one of them being restricted to its function within a procedure. Procedures may be recursive — ie, a procedure may call itself from within its own definition. Functions are similar to procedures, but give a single result. The Electron, like all computers, has many inbuilt functions — SIN, COS, RND, etc. The user may define any number of additional functions by using DEF FN.

The Electron has a good set of conditional statements. IF . . . THEN .. . ELSE, REPEAT . . . UNTIL and FOR . .. NEXT . . . STEP are all supported and the results of all conditional statements are evaluated with FALSE = 0 and TRUE = —1. NOT, AND, OR, and Exclusive-OR are all recognised, and the latter three will all also operate at the 'bitwise' level, which enables some interesting effects to be achieved with graphics.

Instead of using PEEK and POKE, Electron Basic relies on three 'indirection operators' for directly manipulating memory. '?' will either return the contents of a particular memory location or will POKE a value to that address; T will do the same, but . operates on four consecutive locations, while '$' is used to place a string directly in memory. CHR$ is used to obtain a character from its ASCII code; ASC does the reverse.

Formatting and editing

The Eleetron allows considerable flexibility in the formatting of the screen display. The default mode gives fields ten eharacters wide in all modes. Numerical items are printed 'ranged right' in a field, while characters are lined up from the left. This means that if a character is entered on one line and a numeral on the next the two will not line up. This is done so that numbers will line up in the units eolumn (or the least signifieant deeimal). The variation in number of characters across the screen in the various modes means that the number of fields will vary also. There are three different character sizes and these give either eight, four or two fields, depending on the mode selected.

The field width and the number of characters displayed are altered by using the @% integer variable. The default value is @% = &0090A. This gives nine significant figures and a field width of ten.

This may be changed to suit the user — for example, @% = &00A0C will give ten significant figures (the maximum the computer will print) with a field width of 12. The first figure after the '&' can take three values — 0 is the normal configuration, 1 gives numbers in exponent form (ie, an integer followed by a power of 10), and 2 gives numbers to a fixed number of decimal places. This formatting flexibility makes the effective display of information very easy. Numbers may be displayed in either hexadecimal or decimal format: PRINT 10 will give the result to hexadecimal base (A). Text is formatted onscreen by using TAB, semi-colon, comma and apostrophe. TAB can take either one or two parameters — PRINT TAB(n);"PCW" will print `PCW' n spaces across the screen; adding a second parameter allows specification of both column and row. The semi-colon, comma and apostrophe are used to format text in PRINT or INPUT statements — a semi-colon prints two items with no intervening space, a comma moves the following item to the next field, and an apostrophe moves the print position down one line.

Editing is one area in which I am not totally enamoured of the Electron. To amend a program line, the underline cursor is moved to the relevant line, leaving a block cursor at the bottom of the sereen. The COPY key is then pressed, and the correct parts of the program line are reproduced at the block cursor. It is then a matter of skipping the underline cursor over any incorrect parts and typing in the corrections. This seems an unusually long-winded way of going about things and it is all too easy to make a mistake while using COPY. Electron Basic does not reject incorrect program lines as they are entered — as the Spectrum does, for example — but errors are trapped at runtime. Electron Basic does support ON ERROR GOTO/GOSUB so the user may develop his own routines for error-handling. Error messages are mainly self-explanatory, and include the delightful 'Silly', which is the message that appears if you use the AUTO line-numbering command with a step size of less than 1 or more than 255.

AUTO is a labour-saving command that, together with RENUMBER and DELETE, allows manipulation of program lines on entry. Another useful command is LISTO, which allows a number of different formats when a program is listed. Depending on the parameter, LISTO inserts spaces to make FOR . . . NEXT and REPEAT . . . UNTIL loops stand out.

Graphics and sound

The Electron has seven modes, allowing the user to trade off memory for screen display. The modes are numbered 0-7 and are selected by using the command MODE. On power-up the Electron is in Mode 6, a text-only mode which gives 25 lines of 40 characters. Mode 5 has a chunky

character set that allows only 20 characters across the screen, but is 32 lines in depth. Graphic resolution in this mode is 160 by 256 pixels and up to four colours may be onscreen at the same time. Mode 4 gives 32 lines of 40 characters and 320 by 256 pixels with two colours. Mode 3 is, like Mode 6, text-only and supports 25 lines of 80 eharacters. Mode 2 gives 32 lines of 20 characters but is distinguished from Mode 5 by enabling all 16 'colours' to be onscreen at the same time. Acorn is a bit naughty in saying the Electron has 16 colours — as with the BBC Micro, eight of these `colours' are in fact flashing combinations of the eight true tints. Mode 1 has 32 lines of 40 characters and 320 by 256 pixels but differs from Mode 4 in that it allows four colours to be used. Finally, Mode 0 allows 32 lines of 80 characters or a very presentable 640 by 256 pixel resolution in two colours. In all modes the user memory takes a beating — Mode 6 takes up 8k, Modes 4 and 5 use 10k apiece, Mode 3 will use up half the available memory (16k) and Modes 0-3 require a massive 20k each.

To make things easier when switching between modes, the Electron's screen has a standard set of coordinates, regardless of mode. The screen is considered as having 1280 points across by 1024 points upwards — the bottom left hand corner is labelled 0,0 and the top right hand corner is 1279, 1023. Thus DRAW 600,500 will draw a line from the bottom left to approximately the middle of the screen, no matter what mode the computer is in. The graphics screen has a separate set of commands — GCOL is used to set foreground and background eolours, while MOVE, DRAW and PLOT are used to put designs on the screen. The Electron is very flexible in its use of graphics: text may be entered at any point, and graphics and text windows may be set up independently of each other. The PLOT command is especially versatile — using this enables lines and pixels to be plotted in foreground, background or complementary colours and, depending on parameters, solid triangles may be placed on the screen or areas filled in with solid colour.

The VDU command is used to set up windows, to join text and graphics cursors, to change colours and to define characters. It is also used to generate ASCII control codes for moving the cursor around and for sueh tasks as selecting paged mode when listing a program. A full list of VDU codes is shown in Fig 1. VDU is, in fact, shorthand for 'PRINT CHR$'. As an example of its use, consider the Electron's method of dealing with user-defined charaeters:

Charaeter codes 127-255 are initially undefined. 256 bytes of RAM are set aside for the definition of characters with codes from 224 to 255 (more may be defined if memory is reserved for this with an FX call). To define the character with code 224, say, VDU 23 is used, with the bytes that make up the new character entered after the code number. As a simple example, to redefine character 224 as a space it would merely be necessary to enter VDU 23,224,0,0,0,0,0,0,0,0 Now, every

time CHR$ 224 is entered a space will be printed.

The Electron follows BBC Micro syntax in handling sound. The SOUND command is used to produce a noise at the inbuilt speaker. To maintain compatibility with the BBC Micro, there are three tone channels and one noise channel although, unlike the BBC, the Electron can handle only one channel at a time, so chords are not possible. The 'shape' of the note produced is set up by using the ENVELOPE command. All sound commands rely on inputting a stream of numbers — ENVELOPE, for example, has 14 parameters — and the syntax seems needlessly complicated, especially when compared to something like the Sord M5 which makes entering of music very simple. The restriction of using one channel at a time only means that the Electron will not be used for any truly musical effects— as with the Spectrum, the best that can be achieved is simple sound effects for games, etc.

Cassette handling

Programs are saved to and loaded from cassette using a standard format of SAVE/ LOAD " program name". The Electron does not allow the baud rate to be changed, as the BBC Micro does, for example, but the 1200 baud rate is fairly fast and is compatible with the BBC machine's default mode.

Programs will auto-run on loading if CHAIN is used in place of LOAD. There is no VERIFY command, but *CAT will give a list of every program on a cassette and will indicate whether there is a complete

Not a patch in sight

recording on the tape. Files may be set up on cassette, and programs may be merged by saving one as an ASC1I file (using *SPOOL) and loading it back with *EXEC. Machine code is saved and retrieved with *SAVE, *LOAD and *RUN. *OPT is used to control error messages and set up parameters for saving blocks of code on tape.

Function keys

The numeral keys 1-0 may be set up as `soft' or programmable keys, allowing the user to place frequently used routines on a chosen key. Holding down CAPS LOCK/ FUNCTION together with the relevant key will then execute the routine. It is useful, for example, to have a key set up to list a program in Mode 6. The function keys are programmed with *KEY, and control characters are indicated by using the ' symbol. The BREAK key is also user-programmable and the five screen-editing keys (the cursor keys plus COPY) may be redefined by the use of the *FX command.

Documentation

Acorn seems to have learnt its lesson from the BBC *debacle* — at least as far as documentation is concerned. The Electron comes with a 290-page User Guide and a book entitled *Start Programming with the Electron* by Masoud Yazdani. Both books are designed to tie in with the `Welcome' cassette and, taken together with this, provide a comprehensive and painless introduction to Basic programming in general and the Electron in particular.

The User Manual follows the pattern set by the final version of the BBC Manual and is particularly strong on machine code, giving a thorough introduction to 6502 assembly language. This is one of the Electron's strong points — it features a

built-in assembler — and the User Guide devotes 60 pages or so to an easy-to-follow section to help the Electron user make the most of this powerful feature.

Start Programming with the Electron is designed to be read in conjunction with the User Guide and is helpful in amplifying points that may not be altogether clear in the Guide. It contains a section on turtle graphics (a turtle graphics program is also on the 'Welcome' cassette) and goes on to explain how to use this powerful graphics language to escape from a maze.

My only quibble as far as documentation goes is the lack of an index in the User Guide. There is a very full table of contents, but the BBC Micro's manual had an index and I fail to see why one could not, have been included in the Electron manual. Some more hardware details —like a `pin out' breakdown of the expansion port — would have been welcome. That apart, the documentation is well up to Sinclair standard and a considerable advance on most micro material.

Conclusions

The BBC Micro has done wonders for Acorn — its use in schools has ensured it a wide user base, and its expansion facilities have made sure that it will not become outdated in a hurry. With the Electron,

Technical specifications

Processor	6502A, 2MHz
Memory	32k RAM, 32k ROM
Screen	UHF TV — facility for RGB or composite video
Keyboard	55 key ASCII, incorporating 10 function keys
Disks	Not yet available
Language	BBC Basic

Acorn is making BBC Basic available to a wide range of users at a lower cost. As such, the Electron should take a large share of the sub-£200 market currently dominated by Sinclair. In many respects, the Electron scores heavily over the Spectrum: the Basic is considerably more powerful, the graphics are much more flexible and the maximum resolution is many times that of the Spectrum's. The built-in assembler will attract those keen to program in machine code and the physical aspects of the computer — the standard keyboard and all-round attention to detail in the construction — are especially welcome.

It's a pity that the teletext mode is not supported — the extra memory available in this mode would have been very useful for applications involving the manipulation of large amounts of data, and there are one or two other features that are present on the BBC machine but are missing here, such as the ability to scroll sideways. It would have been useful if the BBC Micro's

*TV command had been retained — this is used to centre the display on the television screen; on my TV the Electron's display loses half of the top line, and I have as yet found no way to cure this.

Overall, though, the Electron is one of the most impressive machines I have seen. I'll stick my neck out a bit here and forecast that this one will be *the machine* to challenge the Spectrum on its own ground. The Spectrum has the advantage of the Microdrives and by far the biggest range of software of any machine in this market sector, but the Electron positively oozes quality and there is a wide range of software currently available for the BBC Micro that will run with little or no adaptation on the new machine. The memory map is virtually identical for both machines and almost every BBC program I have tried on the Electron works a treat. The only real difference is the absence of Mode 7 on the Electron — and all that this means is that the title page of many BBC programs will contain a fair bit of gibberish as the Electron tries to make sense of the teletext control codes for things like double-height printing. I have successfully managed to load programs as diverse as the companion cassettes to *The Computer Programme* and a chess program from Computer Concepts.

Many people thought that the Oric would become the Spectrum's major challenger, but this has not happened. Ex-editor David Tebbutt used to indicate his approval of various computers by telling his readers whether or not he would buy one — as far as I can remember, the two machines to receive this accolade were the Atari 800 and Spectrum. As far as I am concerned, I'd plump for a BBC Model **B** — but if I couldn't raise the readies for that I'd be more than happy to settle for an Electron. This one will run and run ...

VDU code table

Decimal	Hex	CTRL	Meaning
0	0	@	Does nothing
1	1	A	Reserved
2	2	B	Reserved
3	3	C	Reserved
4	4	D	Write text at text cursor
5	5	E	Write text at graphics cursor
6	6	F	Enable VDU drivers
7	7	G	Make a short beep
8	8	H	Backspace cursor one character
9	9	I	Forwardspace cursor one character
10	A	J	Move cursor down one line
11	B	K	Move cursor up one line
12	C	L	Clear text area
13	D	M	Move cursor to start of current line
14	E	N	Page mode on
15	F	O	Page mode off
16	10	P	Clear graphics area
17	11	Q	Define text colour
18	12	R	Define graphics colour
19	13	S	Define logical colour
20	14	T	Restore default logical colours
21	15	U	Disable VDU drivers or delete current line
22	16	V	Select screen mode
23	17	W	Re-program display character
24	18	X	Define graphics window
25	19	Y	PLOT K,x,y
26	1A	Z	Restore default windows
27	1B	[	Reserved
28	1C	\	Define text window
29	1D	]	Define graphics origin
30	1E	△	Home text cursor to top left
31	1F	—	Move text cursor to x,y
127	7F		Backspace and delete

Fig 1

Benchmarks

BM1	1.11
BM2	4.01
BM3	11.12
BM4	11.76
BM5	12.40
BM6	18.72
BM7	28.71
BM8	72.53
Average	20.05

All timings in seconds. For a full explanation of Benchmark timings, see PCW November 1982.

Price

Acorn Electron. 32k RAM £199 **END**