

* 6 issues £4.50 * 12 issues £8.50 * Single copies £1.50 * Published Monthly *

* edited by RONALD COHEN, 62 BLENHEIM CRESCENT, LONDON W.11 *

IN GENERAL

A computer in the hand is worth two back home. This is why the PC 1500 is such a marvellous little machine. But there are disadvantages. For most machines there is a mass of software available. There are many programs published in the many magazines. There are many add-ons available from the manufacturers of the computers, and from other sources as well. The costs of running the PC 1500 are not entirely negligible. And programs for other machines are seldom convertible: most are designed for a screen, not for paper.

STATUS 1500 can make life easier in many ways. But only if you help. We need programs, particularly those using the graphics capabilities of the machine. (On cassette if possible). We need useful subroutines, mini-programs, tips and techniques. We need articles about special uses, and about the extra hardware. Most of all, we want to hear about your problems: if we cannot answer them maybe other readers can.

Finally, apologies to almost every reader. For the beginner, the contents will be found too advanced. For the more sophisticated, they will be found elementary and trivial. The answer? Let us know what you want, and we will try to supply it.

Happy programming!

```
1: GRAPH %ROTATE
   %;CSIZE %9:
   GLCURSOR (100,
   0)
2: FOR F=LN %E%TO
   LN %E%:READ Q%
   (F):NEXT F
3: READ U:U=%J%U:
   K=2.0)^((ASC
   LEFT% (U%,1))^
   13):K=%INT K
4: LPRINT Q%;U%;
   RIGHT% (U%,1);
5: LPRINT W%;T%;
   CHR% K;S%;CHR%
   K;RIGHT% (R%,2
   );LEFT% (U%,1)
   ;RIGHT% (R%,1)
   ;MID% (R%,2,2)
   ;U%;Q%
6: LPRINT CHR% (K
   +1):TEXT :LF 3
7: DATA "DUST", "T
   O", "COME", "ALL
   ", "WE", 3.17609
   1259, 94
```

*** CONTENTS ***

- | | |
|---------------------------|---------------------------|
| 1 IN GENERAL | 5 RESCUE OPERATION |
| 2 JARGON | 5 "STARTER" by GT Chaplin |
| 2 FROM THE KEYBOARD | 6 BOOK REVIEW |
| 3 STATUS CODES | 6 MINDBOGGLE CORNER |
| 3 DISTRESS SIGNALS | 7 RESERVE |
| 4 PEEK, POKE & MEMORY - I | 8 ASCII CODES |
| 9 advertisements | |

JARGON - WHICH WAY IS UP?

You will be happy to hear that I know very little about computers. This means that I shall make mistakes; a friendly critic will put me right, and the mistake can be corrected - but I cannot go charging off towards distant horizons, blinding with science everyone but fellow initiates.

All disciplines have their own language. It is not so bad when they invent words: the difficulties arrive when they take ordinary English words, and give them meanings alien to - even contrary to - their ordinary everyday meaning.

The computer world has an unenviable record of arrogance in this respect. Take "parameter" for instance. Ordinarily, this means the limits of a field within which something is valid; but in the language of computers it means, not the field, but "a variable within a fixed field". I was foxed by this for a long time.

"High and low, up and down" are the most confusing of all. Is "2" a lower number than "10"? It is certainly a lower card; but when racing commentators say "The high numbers are drawn on the inside" they mean that "2" is on the inside, and "23" on the outside. "Class 1" is higher than "class 3"; but the Sixth Form is the top of the school. In America the 7th grade is at the bottom. So what is a "high line number"? Where is the "top of memory"?

I do not know the answer to the first question, and do not wish to. I shall say "greater" or "smaller" number, and avoid confusion. As for the "top of memory", this at the end. The "top of memory" for the PC 1500 is at the end, 65535. The bottom is at the beginning.

When it comes to languages, it is easier to remember. Quite simply, upside down. BASIC, an easy language, almost like English, designed for low-level programmers, is a "high level language". Machine code, used by programmers with a high level of sophistication, is of course, a "low level language".

But STATUS 1500 will try to use ordinary common English, as far as this can be done.

FROM THE KEYBOARD

Did you know that many instructions can be given by keying them in directly, without having to write them into a program? You know about CLEAR and GOTO. But did you know about COLOR, WAIT or WAIT n, GRAPH and TEXT? TEXT is particularly useful, since it returns the pen to the left, and restores the co-ordinates to that point.

DIM is also useful from the keyboard, if you are taking DIMensioned variables from cassette. They will not go in, unless you have DIMensioned space for them. RESTORE unfortunately cannot be written from the keyboard. If you execute a program with RUN, it RESTOREs your DATA, which will be read from the beginning: but if you start with GOTO line n, or with a DEF label, DATA will be read from where you left off.

Contrariwise, did you know that TEST, and LLIST or LLIST n (but not LIST) can be written into a program?

STATUS CODES

STATUS CODES 0 to 4 all have important uses, although only STATUS 0 and STATUS 1 are mentioned in the Instruction Manual. STATUS 0 gives the program space remaining: STATUS 1 gives the program space used .

STATUS 2 gives the address of the first free byte for program. Thus with no program in memory, (and 8K attachment) STATUS 2 - STATUS 1 will give 14534. With a program in memory, STATUS 2 - STATUS 1 will give the true beginning of the program area: 14533. Without 8K , these addresses are 16582 and 16581 respectively. See "PEEK, POKE & MEMORY" for details of the 8K attachment with battery. Thus reference to STATUS 2 - STATUS 1 will ensure portability of programs containing PEEK and POKE (within the program area) whether add-on memory is attached or not.

STATUS 3 gives the address after the end of program space. With 8K add-on this is not 18432 , but 24576. This has one important use. When you write a program which includes DIMensioned variables, STATUS 0 takes no account of this, and can return as available space a quantity which is not truly available; so that when you run your program you may get ERROR 10. However if you run that part of your program which does the DIMensioning, and then key STATUS 3 - STATUS 2, the screen will display the true space still available, since STATUS 3 moves to the beginning of the space you have reserved for your DIMensioned variables.

STATUS 4 contains the Line Number of the last line executed. With this, you can turn ON ERROR GOTO into ON ERROR GOSUB. However the number contained in STATUS 4 is sometimes the number of the line in which the ERROR occurs, and sometimes that of the previous line.

Suppose a program in which you wish to ignore errors, and go straight to the line following the ERROR. The following instruction will not work:

```
10: ON ERROR GOTO (STATUS 4 + 10)
```

However the following pair of instructions will work:

```
10: ON ERROR GOTO 100
20: ..... (line containing error)
30: .....
100: GOTO (STATUS 4 + 10)
```

In this program, assuming an ERROR in line 20, the program will immediately execute line 30. (This does not apply to Syntax Errors, and certain other types)

STATUS 5 to STATUS 255 also exist. Fortunately these are all identical with STATUS 4.

DISTRESS SIGNALS

D.G.BOWRING, of Cardiff, has some long programs which have been MERGED
He is having some difficulty modifying them for the 'Software Board'

Can anyone help?

SIMON COX, of Exeter, would like to get hold of the Processor Instruction Set, and the Monitor Listing.

Sorry, Simon! SHARP have not released this information - at least not in English. Several other readers are keen to obtain these details. We will continue to pursue the matter.

PEEK, POKE, & MEMORY

The memory of a computer consists of a number of locations, or 'pigeonholes'. The quantity is measured in so many 'K'. 1K is approximately a thousand, actually 1024, which is 2 to the power of 10. Each location has an address: in the case of the PC 1500 these addresses run from 0 to 65535.

Some of the areas of memory consist of ROM (read-only memory). The contents of the locations in ROM are fixed: they are freely available, but cannot be changed. The rest consist of RAM (Random-Access Memory). The contents of locations in RAM can change according to the instructions the computer receives, whether through a program, or by direct command.

The locations only hold numbers. No characters, just numbers. This is why ASCII codes are necessary. When you write a character in your program, it is stored in the form of its code. Obviously, if you know the address at which it is stored, and have access to that address, you can put a different number in, and the character will be different. In fact, since the instructions the machine receives are all stored as numbers in the same way, it is even possible to change the computer's actions.

This is what PEEK and POKE are all about. We will concern ourselves first with the program area. For the PC 1500 when it has no extra memory, this area starts at 16581. With 8K add-on (non-battery) this area starts at 14533 (a difference of 2K). With the 8K battery add-on, this address would be another 6K less, i.e. 8389. In all cases, so long as you have a program in the machine, the address of the first byte of program is given by $\text{STATUS 2} - \text{STATUS 1}$. The addresses I shall mention will relate to the machine with normal 8K extra: you may have to do a few sums if you have a different set-up.

A very important thing to remember is:

PEEK is a noun & POKE is a verb.

PEEK 12345 is a value. It means "the contents of location whose address is 12345". It does not mean "please see what is at that address." PEEK 12345 can be multiplied or divided or what you will, just like VAL A\$, or SQUARE ROOT OF 4. It is a thing.

POKE is different. The form is : POKE 12345, n (any number). Remember, not a quantity, but the ASCII code for a character or instruction. It means "Into the location whose address is 12345, please insert the number n!"

The first 3 bytes of each line hold the line number, and the quantity of bytes in that line. The last address of the line holds 13, meaning ENTER, or 'new line!'

Go into PRO mode, write NEW. Write the following program: *(line 20 is optional)*

```
10: REM ABC
20: LPRINT PEEK 14538;CHR$ PEEK 14538; PEEK 14539; CHR$ PEEK 14539;
   PEEK 14540; CHR$ PEEK 14540
30: POKE 14538,88: POKE 14539,89: POKE 14540,90
40: LLIST
```

(to be continued next month)

RESCUE OPERATION

Have you ever deleted a program, by writing NEW, and then regretted it? This emergency operation will bring your program back to life, provided you have not over-written it with other code.

1) Write the original first line again, including line number, in PRO mode, and ENTER it in the usual way. If you cannot remember the exact line, it is possible to dig for it by PEEKing along the program from the beginning until you reach the results 13 and 255, or 13 and 0. I find it helpful to start all my programs with a simple 2-byte word line such as 1: TEXT or 1: CLEAR. It essential that the newly-input line be exactly the same length as the original.

2) In RUN mode, from the keyboard, POKE STATUS 2 - 1 , 0
Your program is now restored.

3) You can scroll through it, in PRO mode, but the command LLIST will only produce the first 2 lines. You will need to write:
LLIST , 64000
or a number greater than your highest line number.

4) You will not be able to edit the revived program in its present state. Indeed if you try to do so, you may well destroy the program, beyond revival.

(You can change any characters, but must not alter the number of lines, nor the length of any line).

"STARTER"

A mini-program by G.T.CHAPLIN

```
1: ARUN : BEEP 3
2: "ON ERROR
   GOTO 1
3: T=TIME : A=INT
   (T/100): B=T-(A
   *100): C=INT (A
   /100): D=A-C*10
   0: WAIT 0
4: PRINT "ready";
   D; C; " 83";
   USING "####.#
   ##"; B: USING
5: IF B=FBEEP 5, 1
   00, 200
6: Z$=INKEY$ : IF
   ASC Z$(>0)BEEP
   1
7: GOTO Z$
8: "M"B$=LEFT$ (
   STR$ B, 5):
   PRINT B$:
   CURSOR 6: INPUT
   "set alarm hh.
   mm>"; F: BEEP 1:
   GOTO 2
9: "E" WAIT : CLEAR
   : END
10: "P" BEEP 2: WAIT
   : CLEAR : REM
   main program
```

This pleasant little program is intended to go at the head of any other main program. When you turn on the computer, you will see the date - translated into English style - and the time. The clock will go on running until you press any key. M lets you set an alarm. (Note format!). This will only sound while the clock is visibly running. Line 5 could be amended to initiate execution of a main program at a chosen time. E clears instructions, and ends the program. Any other key directs the program to a label, (without the use of DEF key) and this label can be any alphanumeric symbol. You will hear 3 BEEPs when you turn on the machine, and again on any ERROR, such as pressing a key for which there is no label.

NOTE the use of a 'window' in Line 8 which shows the time before the INPUT. The WAIT 0 instruction is essential. Note also in Line 2 the label "" without a space. This is ASCII 0; and INKEY\$ gives this when there is no entry.

POCKET COMPUTERS

This is an irritating little book. Much of it is written in the sort of style that appeals to average Americans, or to English 9-year-olds: "Wow! Look what fun I'm having with my new Supersonic Pocket Computer! Zap! Bang!" or even (believe it or not) "Look what the naughty dog did to the lampost!". However those parts of the text which deal with facts and techniques are quietly and sensibly written. The objectionable style is superficial. The print is large, and well-spaced, so there is not that much in the book, particularly since 5 pages are devoted to the same program, repeated in minutely different forms. This happens several times. It may be educational, but is hardly economical.

The most irritating thing I found about the book was that I was able to learn quite a lot from it. I thought I knew most of the elementary techniques for handling the PC 1500. I was wrong. It annoyed me to learn from a book which seems so elementary, because it is so clearly expounded. I discovered, for instance, that all the keys, even ENTER and OFF, have ASCII numbers. I discovered how to prevent moving graphics from vanishing at the end of a run.

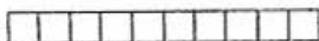
Most of the book consists of mini-programs rather than direct techniques. These can be extracted: you do not have to dig very deep. For the sake of understanding, no doubt, some of the code is not as compressed as it might be. There are many unnecessary brackets, and WAIT instructions are repeated although still in force.

In spite of all its faults, I wish I had acquired this book earlier. If you have had your PC 1500 since last spring, it is still worth borrowing, and if you have only had the computer since Christmas, the book is very well worth buying.

99 TIPS AND TRICKS FOR THE NEW POCKET COMPUTERS, by Jim Cole. (ARCISOFT, Maryland). 128 pp., paperbound. Distributed by ELKAN ELECTRONICS.£6.95

.....

*SOFTWARE Received for Review: EASICALC by Ian Traynor.
This fascinating program will be reviewed in the March issue.*

MINDBOGGLE CORNER

```

99: CLEAR
100: ""BEEP 1: WAIT
      50: PRINT : BEEP
          1, NOT BAND X, X
          : GRAPH : B=160-
          X: LINE (X, 0)-(
          X+20, 20), , , B: X
          =X+20
101: IF NOT BEND
102: GOTO "

```

The program on the left is intended to draw a series of boxes across the paper, each box accompanied by a pair of BEEPs, one constant and the other varying in tone. But will the program do this? Indeed, will it RUN at all? Or will you get ERROR 1? Pick out the Syntax Errors, if any; and then RUN the program just as it is, to see if you are right. No prizes.

RESERVE

It is useful to be able to preserve one's RESERVE programs on cassette, so as to be able to replace them quickly - perhaps because one is experimenting, and keeps 'crashing', or because it is found useful to have various RESERVE programs for various purposes - one for graphics, and another for scientific calculations, for example. Here are 3 methods of CSAVEing RESERVE programs.

- 1) Go into RESERVE mode, and having set up your cassette recorder, key CSAVE without a file-name. This can be disadvantageous if you wish to record a variety of RESERVE programs.
- 2) CSAVE "RESERVE" (or any other file name) when in RESERVE mode. Remember to press the SHIFT key before the inverted commas, or you will just bring up F2: since you are in RESERVE mode.
- 3) By POKEing. The following program will tell you what you have in your RESERVE program, in terms of ASCII characters:-

```
10: SX=STATUS 2 - STATUS 1
20: FOR F = SX-111 TO SX-1
30: LPRINT PEEK F;
40: IF PEEK F = 0 AND PEEK (F + 1) = 0 GOTO 100
50: NEXT F
100: BEEP 3: END
```

This will give your DATA. -- The following set-up is my own favorite RESERVE program. Go into RESERVE mode, write NEW. Return to RUN mode, CLOAD and RUN as a normal program.

```
1: TEXT
5: SX=STATUS 2-
  STATUS 1
10: ON ERROR GOTO
  100
20: FOR J=1 TO 6
30: READ K
40: POKE K, 73
50: NEXT J
60: FOR X=SX-111 TO
  SX-1
70: READ Y
80: POKE X, Y
90: NEXT X
100: BEEP 3: END
110: DATA SX-189, SX
  -163, SX-162, SX
  -137, SX-136, SX
  -135
120: DATA 1, 58, 3, 24
  1, 164, 64, 5, 59,
  6, 44, 14, 44, 9, 5
  8
130: DATA 13, 41, 45,
  40, 12, 240, 183,
  40, 10, 241, 111,
  11, 241, 161
```

- a) Lines 20,30,40, & 110 display I,II, & III when you switch rows.
- b) Lines 120 and 130 contain the DATA for the RESERVE program. Figures 1 to 6 refer to the locations on Row I. Figures 17 to 22 would refer to locations on Row II. Figures 9 to 14 are for locations on Row III.
- c) Please note that it does not matter in which order these locations are given; you will see that 14, 44 in Line 120 (a comma at the end of Row III) is given before 9,58 (a colon at the beginning of that row).
- d) However it is essential to start at the correct Address. With 8K(non-battery) this is 14422. Without 8K it is 16470. (In all cases these addresses are taken care of by STATUS 2-STATUS 1-111)

ASCII CHARACTER CODES

Here is a list of ASCII characters, in more accessible form than in the Instruction Manual (page 144).

Some characters print on the screen, but will not LPRINT on paper. An asterisk is placed against these numbers.

CHR\$ 0 is null. CHR\$ 1 to CHR\$ 31 do not print, but are used in the execution of programs. CHR\$ 13 indicates 'NEW LINE'. You will notice that it has indented itself, as has CHR\$ 0.

CHR\$ 17 to CHR\$ 22 refer to the 6 'Software Keys'. A simple program using these:

```

10: WAIT 0: A$ = " " : ON ERROR GOTO 20
20: A$ = INKEY$
30: IF A$ = CHR$ 17 BEEP 1,255, 100
40: IF A$ = CHR$ 18 BEEP 1, 127, 200
    etc.
100 GOTO 10

```

You could then attempt to play a scale on these keys.

CHR\$ 32 is a SPACE. So I think is CHR\$ 96. CHR\$ 39 is the 'Insertion' symbol. CHR\$ 95 is the ordinary cursor. CHR\$ 127 is the cursor that flashes. CHR\$ 126 is the 'tilde' which appears when the computer is given an instruction it cannot fulfil owing to lack of an attachment (i.e. LPRINT without the printer). This symbol also appears in the program replacing any instructions such as LPRINT which are improper if the printer is not attached.

CHR\$ 128 to CHR\$ 255 are identical to CHR\$ 0 to CHR\$ 127 - except for the fact that they do not LPRINT on paper, but will only print on the screen. They are however used in connection with the RS 232 interface.

All command keys, such as the Cursors, MODE, ENTER and even OFF, have ASCII codes between 1 and 31.

0	16	32*	0 48	@ 64	P 80	96*	p 112
1	17	! 33	1 49	A 65	Q 81	a 97	q 113
2	18	" 34	2 50	B 66	R 82	b 98	r 114
3	19	# 35	3 51	C 67	S 83	c 99	s 115
4	20	\$ 36	4 52	D 68	T 84	d 100	t 116
5	21	% 37	5 53	E 69	U 85	e 101	u 117
6	22	& 38	6 54	F 70	V 86	f 102	v 118
7	23	^ 39*	7 55	G 71	W 87	g 103	w 119
8	24	< 40	8 56	H 72	X 88	h 104	x 120
9	25	> 41	9 57	I 73	Y 89	i 105	y 121
10	26	* 42	: 58	J 74	Z 90	j 106	z 122
11	27	+ 43	; 59	K 75	[91	k 107	{ 123*
12	28	, 44	< 60	L 76	\ 92	l 108	! 124*
13	29	- 45	= 61	M 77	^ 93	m 109	} 125*
14	30	. 46	> 62	N 78	_ 94	n 110	~ 126
15	31	/ 47	? 63	O 79	— 95*	o 111	■ 127*

Next Month: RESERVED WORDS & their 2-byte tokens
 RENUMBER - EASICALC reviewed - more
 PEEK & POKE - RS232 INTERFACE -
 WHAT TO BUY AND WHERE TO BUY IT
 and some programs.....

THE FIRST BOOK FOR THE PC-1500

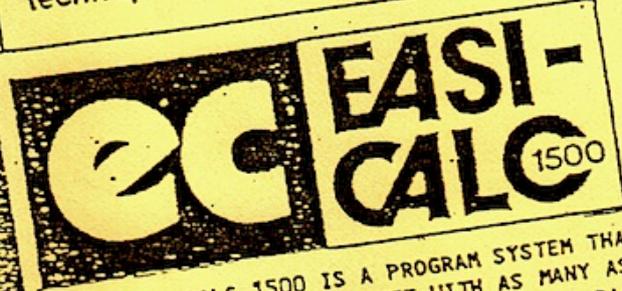
99 Tips & Tricks for the New Pocket Computers

by Jim Cole

99 all-new ready-to-run programs for the TRS-80
PC-2 and Sharp PC-1500 pocket computers,
including LCD and printer/plotter graphics
techniques.

£6.95 + 0.50p P&P

PENS AND PAPER ROLLS...£5.84
CSPT50 (20 PAPER ROLLS)...£3.09
EA850B 4 BLACK PENS...£3.09
EA850C 4 COLOURED PENS...£3.09



OUR MOST POPULAR PROGRAM
A SPREADSHEET PROGRAM FOR THE SHARP
PC-1500 AND TANDY-PC-2.
NEEDS PC-1500 8K RAM CE-150

£19.95 INCLUDING VAT

EASI_CALC 1500 IS A PROGRAM SYSTEM THAT ALLOWS YOU TO USE YOUR COMPUTER
AS A LARGE SPREADSHEET WITH AS MANY AS 26 COLUMNS ACROSS... UP TO 99 ROWS
DEEP. YOU DECIDE ON THE SIZE OF THE PAGE WITHIN A MAXIMUM LIMIT OF 305 CELLS
EASI-CALC COMES WITH A DATA-QUALITY TAPE AND A 30 PAGE MANUAL AND TUTORIAL.

AN ELKAN SECRET.....
IF THE BATTERIES RUN DOWN IN YOUR PC-1500 WITH THE CE-159 (8K BATTERY RAM)
YOU DONT LOSE THE PROGRAM.....WORTH KNOWING. THE CE-159 FITS IN THE SAME
COMPARTMENT AS THE CE-155 AND INCLUDES A COMPREHENSIVE MANUAL..OUR PRICE £79.95

LET YOUR PC-1500 COMMUNICATE.....GET A CE-158 RS-232 and PRINTER INTERFACE.
FEATURES RECHARGEABLE BATTERY FOR PORTABILITY. USE A FULL SIZE PRINTER..LINK
TO OTHER MICROS...DATA CAPTURE...LET YOUR SYSTEM GROW.....OUR PRICE...£139.95

HAVE YOU TRIED THE CE-153 SOFTWARE PANEL? THIS PANEL HAS 140 TOUCH SENSITIVE
SQUARES . INCLUDES A MACHINE CODE OPERATING TAPE AND 2 SAMPLE PROGRAMS...£79.95

TEXT HANDLING PROGRAM OUR FIRST PROGRAM FOR THE CE-153 SOFTWARE PANEL.
NEEDS 4K OR 8K RAM. EACH OF THE 140 SQUARES CAN HOLD UP TO 45 CHARACTERS.....

SILICON NOTEBOOK by GEMINI A DATABASE IN MINITURE. 140 NOTES OF 50 CHARACTERS
WITH 8K RAM OR 60 NOTES WITH 4K RAM. YOU CAN ADD*DELETE*FIND*SAVE*ETC....£19.95

ARE YOU INVOLVED WITH SELLING? THIS PROGRAM IS THE GREATEST SALES AID EVER!
SALES PRESENTATION PROGRAM BY MIKE LINDLEY..USES 4 COLOUR GRAPHICS AND SOUND.
THIS PROGRAM TOOK 4 MONTHS TO WRITE AND CAN BE TAILORED TO YOUR NEEDS.WRITE
: TO ELKAN ELECTRONICS GIVING DETAILS OF YOURSALES PROBLEM.

ELKAN ELECTRONICS

FREEPOST
NO STAMP REQUIRED
PRESTWICH
MANCHESTER M25 6LZ
TELEPHONE:
061-798 7613(24 HR. SERVICE)
PRESTEL 258 2022