VOL.2
NO. 6

STATUS 1500

JULY
1984

* 12 issues £10.50 (UK) * overseas £14.50 * published monthly *

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

JAM TOMORROW

The first of the PC 1500A are now in the country, and are initially being sold direct by SHARP (UK). Send them a cheque for £169.95 and see what happens. (Unfair! Stop knocking SHARP!) No doubt the price will come down eventually, but not for some time to come. The first of the CE 161 are also in Manchester, but no price is yet available. There is a rumour that they may perhaps be cheaper than previously expected, so keep your fingers crossed.

Keep them crossed also for the second half of this year's newsletter. There really is some good stuff on the way, and - who knows - it may actually get printed. DAVID RIHOY has written an exciting series, with fully detailed instructions, on the subject of Designing Your Own Characters for the CE 150. Of course, one could always do this graphically, if in no hurry for printout: but now the ROM routines have been cracked: and a very brief program (if worked out with care) can give you your own characters just as if they were in ROM. A.E.L.COX is also writing some provocative articles (anyway I am sure they will provoke me!) but they will be useful as well as provocative: one of the first will go to the roots of the problems associated with "Sort" routines, and explain some short cuts. In Germany the PC 1500 is booming (see 'Signals'). A new software house in Scotland intends to specialise in Programs for Portable Computers, with particular reference to PC 1500, and their first program will be reviewed next month.

At home, there is still a chance of the newsletter appearing on the 1st of the month - this year, next year.... 'Lets Write a Program' may eventually actually have some Program in it. At the beginning of the year 'Peek & Poke' was intended to become intermittent, but new material keeps turning up. 'Mindboggle Corner' is always on the verge of running out of steam, but keeps getting refuelled, unfortunately, by one minor disaster after another. Next month's competition will offer a small prize to the reader who can claim to have used his PC 1500 (usefully) in the most unlikely or exotic situation. Have an adventurous summer holiday!

SIMON EYRE finds the newsletter very useful, but has a problem with his CE 161. After loading software, any attempt to edit destroys the program.

I am glad that you enjoy the newsletter. Perhaps you would care to subscribe to it?

Both SYDNEY LENSSEN and R.J.COURT speak highly of the 'Business Graphics' module, CE 501B.

This is obtainable from Atlantic Northeast Marketing, but I am told that the cheapest place to get it is the Duty Free Shop at Schipol Airport, which also stocks other accessories for PC 1500.

DAVID BOWRING is pleased with his EP44 typewriter, which also acts as a terminal, and contains a serial interface for this purpose.

The print quality, although dot-matrix, is certainly excellent. However the special shiny paper one must use is somewhat expensive.

M.GREENING-LEWIS asks about the memory of the 1500A.

This almost certainly runs from 16K, where the unexpanded PC 1500 starts, to 24K, where the CE 155 ends. This would allow the CE 161, covering 0 to 16K, to slot neatly in front of it. Of course, the effect of this could be to leave data in the computer, while removing the program in the non-volatile memory module. The reverse might have been more useful. If my estimate of the 1500A memory is correct, it would seem that you could also use the CE 159 with it, but not the CE 151 or CE 155.

RON DAVIS thinks that C.P.UNDERWOOD was over-critical of our Volume 1. He finds it very useful indeed. He also claims authorship of the GOLF improvement on page 46. He asks about my answer to CHIA PIA LON (page 31). What happens if you POKE into the ME1 area. SHARP say this "may result in destruction". Destruction of what? Can the CPU be damaged?

Theoretically, no. According to the SHARP instruction book, you cannot damage the computer by pressing keys. But you could certainly inhibit CLOAD or CSAVE. You could perhaps give yourself quite a headache trying to sort things out. The greatest danger would be if there were no immediate visible effect, but only delayed-action problems. I am sorry that I am not prepared to conduct the experiment on your behalf.

CRISTER SKOGLUND has discovered many interesting advertisements in the German magazine CHIP. Over 20 of them, in fact, concerned with PC 1500, which is very popular over there, particularly with engineers.

Thanks for the information. I will try and publish some of these addresses in forthcoming issues.

IAN TRAYNOR says that I misunderstood his solution to the March Mindboggler. His routine is below. N is size of DIMensioned array A(N).

```
SAVE ROUTINE:
   a) get File name in F$ (not more than 11 CHR$ long)
   b) 1000: G$=F$+" spec": PRINT # G$;N
      1010: G$=F$+" Data": PRINT # G$;A$(*)
      1020: GOTO "menu"

LOAD ROUTINE:
   a) Check Data is to be cleared.
   b) 2000: CLEAR: get filename in F$
   c) 2010: G$=F$+" spec": INPUT # G$;N
   d) 2020: DIM A$(G)
   e) 2030: G$=G$+" Data": INPUT # G$; A(*): GOTO "menu"
```

This is much simpler to operate than it appears. But see PEEK & POKE!

PEEK POKE & MEMORY - XVI

More questions than answers this month: but the answer is a gem.

Address Finder is now obsolete. To find the address of any line, attach the computer to the CE 150. LLIST the line whose address is wanted. Then the address is: 256*PEEK 31202+PEEK 31203. So simple! I am indebted to the New Zealand newsletter, edited by ALLAN THOMAS, for this most useful piece of information. In the Technical Reference Manual these locations are given as "USER Y COUNTER". No doubt they are used both for that purpose and for LLISTing.

While on the subject of LLISTing, perhaps you can help me. If I print LLISTings on my printer, instead of on the CE 150, the results reproduce better in the smaller size: and many hours of mucking about with scissors and paste, copying and re-copying, are avoided. But there is one snag. If there are many short lines, much space on the page is wasted. An answer might seem to be to print in 2 columns, splitting the occasional long line. But this is not quite satisfactory. Unlike the CE 150, the CE 158 is not inhibited from splitting Reserved Words: nor can I control the format, since the second part of a split line starts printing from the lefthand margin, and intervenes between line numbers. The result is messy. Have any of you experimented with LLISTing? Any method of controlling the printout of LLISTings, perhaps a machine-code routine, would be gratefully received.

In order to prime the Reserve, it is necessary to know where it is. And any method of finding it should be fully portable. With so many different possible formats for memory (unexpanded, CE 151, CE 155, CE 159, CE 161, PC 1500A) some users may on occasion make use of more than one format. The Reserve always starts at beginning of memory +86 But where is the beginning of memory? Normally it is STATUS 2 -STATUS 1-197. But supposing you have more than one program in memory, and the second was installed after concealing the first by NEW STATUS 2 ? (see vol.1, p.118) Again, if you have the CE 161, you may start your programs after NEW 256, as recommended, or after the more normal and apparently harmless NEW 0 (=NEW 197) The following method of locating the start of Reserve is slow and cumbersome, but portable:

```
9000: FOR F=STATUS 2-STATUS 1 TO 0 STEP -1
9010: H=PEEK F
9020: POKE F,0
9030: IF PEEK F=255 GOTO 9090
9040: POKE F,H
9050: NEXT F
9090: "reserve" R=F+86+(F<>0)
```

You may also wish to know the end of your memory area. This is indicated by STATUS 3 (strictly speaking, STATUS 3-1). But that is only true before you DIMension, or use any 2-character variables. As you know, STATUS 3 always indicates the start of these. So how then do you discover the end of your memory area, without wiping out data? I suppose you could use something like the routine above, running in the opposite direction, but it could be dreadfully slow. The memory must certainly contain somewhere these Start and End addresses, but where?

I have experimented with CSAVE M to preserve the whole program area, including DIMensioned variables, and then preserving the pointers by CSAVE M "ä"; 30208,31500. This worked perfectly - once only! and never again! In vain I have tried to reproduce the original conditions. Any ideas?

LETS WRITE A PROGRAM - VI

I hope you are not intending to play Roulette on Boxing Day. I very much doubt whether this program will be finished by then. There is still so much to do. I said that this month we would tackle "Place Your Bets!" but I fear we shall not get through this part in one session, particularly since I am very puzzled as to how to tackle this: the vital spark of inspiration refuses to appear. In any case, I am beginning to lose interest a little, just as you are.

But when a problem appears insoluble the first thing to do is to break it down into its component parts. At least some of these can be dealt with, and the hardest, even if not solved, will be narrowed down.

So consider 3 or 4 routines. The announcement of bets. The calculation of odds. The amount of win or loss. The crediting or debiting of this. The display of the result. The next bet - or the next player. There are 2 methods of approach. Either take each player in turn, and take him through the possible sequence of bets - red/black - odd/even etc - columns and dozens - groups of numbers - single numbers. Or else let any player input his name, let the program identify him, and offer bets via a menu. I do not usually like menus, but here one may be appropriate, since we wish to offer the player specific limited choices, followed by further details of each choice. This method could be slower in practice than the other, if players dither,
On the other hand, it may be more flexible (except for the daunting problem of system betting).

As far as the actual bets are concerned, we are again faced with the prospect of a plethora of IF statements, which we want to avoid. The only way I can see to avoid these is to establish a set of conditions - odd/even - black/red etc. - N1, N2, N3..., attached to the result of the spin, others - P1, P2, P3..., for the bets, match them up, and credit or debit according to the result. Thus:

```
[bank] = [bank] + [stake] * (N1<>P1)
[player] = [player] + [stake] * [odds] * (N1=P1)   and so on.
```

This seems a bit cumbersome. On the other hand it should be consistent, and may therefore be susceptible to the use of a common subroutine, thus saving space. Also it will allow players to change their minds, or correct mistakes - by simply placing an equal - but minus - bet! Lets strike while the iron is hot, and sketch out a few lines of code. We can fill in the exact line numbers later, and do all the polishing after that.

```
INPUT "name";N$
FOR F=1 TO N
IF N$=A$(F) LET [player number] = F: GOTO "menu"
NEXT F
((name not found: GOTO [input name] again

"menu" WAIT 0: PRINT "1/1  2/1 group number next cancel"
A$=INKEY$: GOTO ASC INKEY$+XX00 ((then secondary menus>
XX17: PRINT "red black odd even low high"
A=ASC INKEY$: INPUT "stake?"S: GOTO XZ00+A
etc.

XZ17: P1=0: GOTO "calculate"
XZ18: P1=1: GOSUB "calculate"
```

This is obviously going to need a great deal of hammering into shape. At the moment it does not even add up: and when it does it will no doubt be infested with bugs. The best thing to do is to write and RUN a short pilot program, and get this sorted out as far as possible on its own, before trying to integrate it into the program proper.

POLL

Mike O'REGAN was in charge of a polling-station for the recent Euro-elections. He devised and used the following program to see how voting was going. Check that the real-time clock in your computer is correct before using the program. It may be used at any time during polling, and may be turned off between sampling.

Initialise by DEF =, and operate by DEF (space).

```
 20 " "TN=(TIME /100-INT (TIME /100))*100:INPUT "Last Ticket Issued ";LT
 30 ET=TN-OT:DT=DEG ET
 40 VC=LT-FT
 50 PV=VC/TV*100
 60 VR=VC/DT
 70 FV=PV/DT*TT
 80 LPRINT W$;" Ward": LF1
 90 USING "###.##":LPRINT "at";TN:USING :LPRINT "     with";VC
100 LPRINT "     votes cast"
105 USING "####.##":LPRINT " rate ";VR:LPRINT " votes per hour": LF 1
110 LPRINT "percentage of vote"
115 LPRINT "  = ";PV;"%":LF1
120 LPRINT "FORECAST TURNOUT = "
125 LPRINT "     ";FV;"%":USING
130 LF3:END
499 "="CLEAR
500 INPUT "ward name ";W$
510 INPUT "total voters on roll= ";TV
520 INPUT "Poll opens at (hh.mm) ";OT:OT=DEG OT
530 INPUT "open for (hh.mm) ";TT:TT=DEG TT
540 INPUT "first ticket no. issued?";FT
550 END
```

CHILD'S PLAY

Frank Odds has difficulty using his PC 1500. Not because of any fault in the machine, but because his 3-year-old daughter will not leave go of it! He has evolved a program which flashes letters onto the screen, and they must be identified by pressing the correct key. Here however is another program, which displays a number of objects: they remain on the screen until the correct number key is pressed.

```
5: A$="7F7F7F00000000"
10: X=RND 6: X=X+1: Y=RND 17: CLS
30: CURSOR Y: WAIT 20: FOR F=1 TO X: GPRINT A$;
60: NEXT F
70: B$=INKEY$ : B=VAL B$
75: IF ASC B$=0 GOTO 70
80: IF B=X BEEP 1,50,150: BEEP 1,25,250: BEEP 1,12,900: GOTO 10
90: IF B<>X BEEP 1,100,200: BEEP 1,200,200: GOTO 30
```

LABELS

A line which consists solely of a LABEL, e.g.:

```
10: "X"
```

gives ERROR 21 IN 10 on both my PC 1500s. Yet other readers report running such a line, both by RUN and by DEF "X" without any ERROR at all. Can anyone explain?

IMPROVEMENTS

title:	RESTRUCTURE
original:	RJC (vol.1, p.97)
improved by:	C.A.F.LEDSAM
purpose:	speed
comment:	brilliant

OPERATION: To Restructure a program, MERGE this routine with the subject program. You must have 1210 bytes free. Initiate by DEF =. Answer prompts with the number of the first line to be transferred, the last line to be transferred, and the new line number to be ascribed to the first line of the transferred block. The routine will renumber all lines in the block appropriately, and transfer them to their new position. The process may be repeated. Afterwards, delete the routine by DEF C. For the latter to work, it is essential that the length of the routine be exact.

The routine is almost foolproof, and in normal use cannot harm a program, which can be edited in the usual way after RESTRUCTURE. You should avoid transferring a block in such a way that some of the new line numbers duplicate old ones: but if you do make this mistake, no harm is done. Simply reverse the process, and transfer the block back to its original position. However you must be careful to avoid transferring a block when the transfer would give any line a line number greater than 66535. This could destroy the subject program.

The RESTRUCTURE operation takes only a few seconds to RUN. To transpose the two halves of an 8k program now takes 6 seconds instead of 6 hours!

WARNING: Do NOT use this routine for simple Renumbering, without any actual transposition of line positions. If you do, you may get trapped in an endless loop, only curable by RESET, or even taking the batteries out.

```
  1 "="S=STATUS 2+1
 10 INPUT "first line=? ";A:INPUT "last line=? ";B:INPUT "new block start no.? ";C
 15 A=A-65536*(A>32767):B=B-65536*(B>32767):E=C-65536*(C>32767)
 20 GOSUB "address":CALL S,B:B=B+PEEK (B+2)+3:F=INT (B/256):G=B-F*256:POKE &790C,&82,F,G
 25 C=C-A+65536*(C<A):D=INT (C/256):C=C-D*256
 30 CALL S,A:AA=A:CALL S,E:GOSUB "renumber":CALL S+31,AA
 35 IF (E<A)LET X=E:E=B:B=A:A=X
 40 B=B-A-1:AA=PEEK &7905:A=PEEK &7906:EE=PEEK &7925:E=PEEK &7926
 50 H=INT ((S+113)/256):HH=S+113-H*256
 60 GOSUB "restructure":CALL S+58,B:END
100 "C"N=STATUS 2-1047:M=INT (N/256):N=N-M*256
101 POKE 30823,M,N,PEEK 30821,PEEK 30822
102 BEEP 3:END
200 "address"POKE S,&FD,&88,&FD,&2A,&A5,&78,&66,&08,&A5,&78,&66,&0A,&BE,&02,&EA,&00
201 POKE S+16,&81,&00,&A5,&78,&A6,&08,&A5,&78,&A7,&0A,&46,&46,&46,&FB,&9A:RETURN
300 "renumber"POKE S+31,&44,&F9,&85,C,&03,&0E,&46,&85,D,&03,&0E,&44,&44,&05,&FD
301 POKE S+46,&CA,&44,&4C,F,&99,&15,&4E,G,&99,&19,&F8,&9A:RETURN
400 "restructure"POKE S+58,&FD,&88,&FD,&2A,&48,AA,&4A,A,&58,H,&5A,HH,&FD,&98,&FD
401 POKE S+73,&A8,&FD,&88,&F5,&88,&03,&FD,&1A,&F5,&4C,EE,&99,&05,&4E,E,&99,&09
402 POKE S+90,&FD,&2A,&FD,&0A,&F5,&88,&03,&FD,&62,&6C,&FF,&99,&29,&FB,&9A:RETURN
```

ERROR ON PAGE 37

Line 30, of the program in the lefthand column, should start:

```
30: B=P1*256+P2:....
```

In some copies it may read:

```
30: B=B1.....
```

Please amend your copy if necessary.

MINDBOGGLE CORNER

No puzzle this month, but in PEEK & POKE there are several Mindboggling questions. The most useful answer to any of these will qualify for the usual small prize. Time limit: a few weeks.

The immortal prose referred to in the May competition was of course:

THE QUICK BROWN FOX JUMPED OVER THE LAZY DOG

For the benefit of readers in Europe, perhaps I should mention that this is a standard typing exercise. On printing out the alphabet, it ran:

CCCDGGGHKKKL....

I did mention that I was using parallel interfacing, and what had happened was that a spot of solder had joined pins 2 and 3. These carry the 2 lowest bits. When both were low, or both were high, they did not affect each other, but when only one was meant to be high, it transmitted a signal on the other as well. So that both CHR$ 65 - A - (01000001) and B - CHR$ 66 (01000010) would become C - CHR$ 67 (01000011). And so on. Only one correct solution was received, from MIKE SMITH, who not surprisingly has rejected options a) and b) and will therefore receive a copy of SUPERTEXT (B).

subroutine 4 DATABASE EXAMINER

For a quick look at any series of a DIMensioned variable, too long for the Display, without recourse to the printer, use the subroutine below. It will scroll continuously through the variable, starting at its beginning: and will then repeat the process. The ▲ and ▼ cursors change the number of the variable. Any other key returns to the start of the variable currently being scrolled.

```
 10 " "WAIT 9:ON ERROR GOTO 150
 15 G=0
 20 FOR F=25TO 1STEP -1
 25 IF ASC INKEY$ <>0GOTO 120
 29 IF G<0LET G=0
 30 CURSOR F:PRINT G;">";A$(G)
 40 NEXT F
 50 LL=LEN A$(G)
 60 FOR F=1TO LL
 70 B$(1)=MID$ (A$(G),F,25)
 90 PRINT B$(1)
 95 IF ASC INKEY$ <>0GOTO 120
 99 NEXT F
120 K=ASC INKEY$
130 CLS :BEEP 1,19,999:G=G+(K=10)-(K=11):GOTO 20
150 G=G-1:GOTO 20
```

STATUS 1 = 256

FROM THE KEYBOARD

You want to initiate execution of a program. You do not wish to use RUN because it will wipe out data. No DEF [label] has been provided. You cannot remember the first Line Number, and do not wish to UNLOCK and examine the program, so as to use GOTO [N]. You do not need to do so! Key GOTO, and [ENTER], without any Line Number, and execution will start at the first available line.

SUPERTEXT (B) reviewed by Sydney Lenssen

PC 1500 owners are an idiosyncratic bunch, which is not to say actual idiots. We know the machine has remarkable capabilities for its size. Yet for most tasks another machine will usually do the job better. Our fascination with the pocket Sharp lies in the challenge of exploiting it to its limits. It's like trying to sail the Atlantic single-handed when any sensible person would choose the QE2.

Word processing on the 1500 must be seen in this spirit. If you really want to write a letter or report while being jolted in the train, or riding a camel, then why not use pen and paper? If your message really has to be typed, then why not use a typewriter?

Nevertheless SUPERTEXT (B), the latest W.P. program from STATUS SOFTWARE is a quite remarkable achievement. If you already have the original version, you will find that the new (B) version lifts you from the "Mini" class to the realms of Jaguar or Rolls. Most features have become friendlier and more usable, the instructions are far easier to follow, despite the host of extra features which do require some guidance. If you have already used Supertext or another W.P. program, then let me urge you to take the plunge with this (B) version because you're sure to find it fun, and very probably useful. If you enjoy ingenious subroutines, you will find a wealth of them in this 8K program.

If you haven't W.P.'d before with your Sharp, and don't know much about it, then let me try to give you a simple guide. The tape takes about 10 minutes to load. As usual it loaded first time. (Status Software will of course replace the cassette if you have trouble).

If it's the first time for your 1500, you must RUN a simple check routine before you start. But take care! If you use RUN again you will wipe out any text you have written! So it is as well to get used to starting with DEF F for the WRITE mode. You will need to cultivate a habit of writing 4 or 5 words at a time before keying ENTER. Avoid single characters because most of these act as commands, though there is a way of entering single characters if you must.

The display panel gives you a line number and length, and your text as you write it. You can backspace, and edit as you go, in the usual way. Most of the time, I find it best to type away as fast as possible, mistakes and all, and later switch to EDIT mode. Then I can scroll through the complete text, making any corrections, even changing or inserting complete lines, before justifying, centering, reformatting, or even switching lines and whole paragraphs around.

Usually I like to do a draft in PRINT mode: this prints out the entire text in CSIZE 1, with text line numbers, so that I can read the copy right through and spot anything I want to change, via EDIT mode - or return to WRITE mode and write a new chunk of text for insertion. Or else you can go straight into CSIZE 2 printing with DEF M. This prints out vertically in CSIZE 2, complete with reference margin for trimming to A4 width. Spacing is variable from 5 to 11 lines a strip: the wider spacing is clearer. All 4 colors can be used, though printing in colour does require simple data entries on line 61000.

Rather than go on about the 16 page book of instructions, I would refer you to the advertisement in the June issue. This lists mosts of the features in jargon not too hard to understand, and adequately explained in the instructions themselves.

For some of you it might have been better if this review had been written by a Sharpman who has followed our editor's love affair with the Juki printer. Unfortunately I have not been able to test this mode. Suffice it to say that the program is tailored for standard Diablo protocols, should present no great problems with any other printer, and proof of the pudding can be seen each month, since STATUS 1500 is now produced entirely on SUPERTEXT (B) and stored for posterity on a tiny mini-cassette!

Let it be enough for me to dispel any lingering doubts you may have. You may not find W.P. is suitable for all missives, but almost certainly you will find applications where your Sharp does a job impossible to achieve by alternative means. And at the special reader's price, the cassette represents a real bargain.

PC 1500 ELECTRICS

WARNING: The information contained in these articles on the electrics is believed to be entirely accurate, but no responsibility is accepted for any mishaps resulting from its use.

For the benefit of the electronically illiterate, (such as the editor) a general look at the terms used may be helpful. AMPS (A) describes the current. The VOLTAGE (V) is the force making the current flow, and the output (heat, light, sound, etc.) is measured in WATTS (W). When discussing domestic appliances, the term WATTS is often used to describe the consumption of current: this is not strictly accurate, but is tolerated because of the constant relationship between the 3 measures, as shown by the formula:

$$V * A = W$$

In the case of an ordinary 3KW (=3000 watts) 3-bar electric fire, this consumes 12 AMPS at 250 VOLTS when all 3 bars are on, 8 AMPS when 2 bars are on, or 4 AMPS when only 1 bar is on. If the voltage were only 125 VOLTS, then it would take 24 AMPS, 16 AMPS, and 8 AMPS respectively to produce the same wattage.

For the PC 1500 and its peripherals we shall be talking about very much smaller currents. These will be measured in milliamps (=one thousandth of an amp) and microamps (one millionth of an amp). Milliamp is usually abbreviated to **mA** , and microamp to µA .

The current requirements of the PC 1500/CE 150 vary enormously according to the function being performed. **The table overleaf illustrates these variations.** The battery parameter which determines the period for which a particular current can be supplied is CAPACITY, measured in AMPERE HOURS (Ah). The group of 5 rechargeable Nicad cells in the CE 150 has a capacity of 0.5 Ah, and can supply 500 mA for 1 hour, 50 mA for 10 hours, or about 165 mA for 3 hours. Each cell has a voltage of 1.2 V, but in the CE 150 they are used in series, "head to tail", thus giving 6 volts which is what the CE 150 runs at. The 9 volts of the EA 150 charger output is reduced to 6 volts inside the CE 150. Unless a special wiring connection were made directly to the internal battery pack, CAPACITY could only be increased by the use of a 9-volt battery connected instead of the EA 150 charger/mains combination. Any battery of lower voltage does not allow for the internal battery-charging control circuit.

The batteries in the PC 1500 are ordinary AA replaceable dry-cell batteries. They are NOT rechargeable, and when the PC 1500 is fed by the EA 150 charger, no attempt is made to recharge these: instead they are bypassed, and the PC 1500 is powered directly by the charger. If the PC 1500 is connected to the CE 150 printer, then the PC 1500 is powered directly by the batteries of the printer. You are warned against plugging the charger directly into the PC 1500 while the computer and printer are attached together. (See p.143 of the SHARP instruction manual). One reader reports that after inadvertently making such a connection, he had to send the printer to SHARP for repair - possibly a total discharge of the Nicad batteries in the printer?

Of course the current used by the PC 1500 is extremely small compared with the requirements of the CE 150. This is why the batteries in the PC 1500 last such a very long time, particularly if it is usually attached to the printer when in use. It has been suggested that the current used by the TRAMSOFT TOOL-2 may be greater than usually realised, and the frequent use of this attachment may cause the batteries in the PC 1500 to run down much sooner than expected.

[to be continued

CURRENT LEVELS IN PC 1500/CE 150/EA 150

CONFIGURATION	CURRENT DRAWN FROM	MODE				
		OFF	ON	KEY PRESSED	SCROLL	EXECUTE PROGRAMME
PC alone	PC battery	30 μA	6	7–10	13	19
PC + EA	EA	30 μA	6	7–10	13	19
CE alone	CE battery	130 μA				
CE + EA	EA	50				
				PEN MOTOR	LF MOTOR	LLIST
PC + CE	CE battery	160 μA	7	330	400	600 max
PC+CE+EA	CE battery	-50	-45	110	165	310
PC+CE+EA	EA	50	52	220	235	290 max

MINUS values indicate charging. All values are in mA except where specifically shown in μA. Currents are typical, but may vary according to state of battery charge. ON + OFF values are constant. Charging current varies by about 10 mA. Proportion of PEN and LF current varies, but total is fairly constant. LLIST varies over a wide range, average value about 400 mA.