

IPSO FACTO

NEWSLETTER OF THE INDEPENDENT PSION ORGANISER USER GROUP

VOLUME 1 * * * NUMBER 1 * * * MARCH 1987

EDITORIAL

Welcome to the first issue of IPSO FACTO, the newsletter of IPSO GROUP. I hope that this will prove to be the first of many such newsletters and that we will all benefit from pooling our ideas in this way. To put the record straight from the outset, I am NOT an Organiser "expert" - in fact there are probably many of you out there with much more knowledge of the machine. However, I am prepared to share what little knowledge I have although the idea is to collate and distribute information from many sources. I am sure that you will have something to contribute, if it is only constructive criticism! I must emphasise that the continued existence of IPSO FACTO and indeed IPSO GROUP depends on the support of its members.

From the information which I have gleaned from your Membership Applications it seems that we are a mixed bunch, with, I suspect great disparities in our levels of expertise, so I hope that we can include something of interest for everybody. If any of you have experience with machine code, bar-code, card-swipe or communications with other micros etc I would be pleased to hear from you.

When PSION launched the ORGANISER I, there was a big splash in the media hailing it as "the world's first pocket computer" (which it was not!) and generally praising what was a very limited machine at the time. What is hard to understand is that the ORGANISER II (which is a very different proposition) has received relatively little publicity - outside the micro world. I have personally introduced the ORGANISER to a few dozen people, all of whom were impressed, and few of whom had even heard of either PSION or the ORGANISER. As a group, we must try to put the record straight.

1. Editorial
3. Books, Hints & Tips
5,6,7. TEXTBASE Review

2. Beginners Please
4. Your Letters
8. Your Programs & Procedures

BEGINNERS PLEASE.

In this section of IPSO FACTO I propose to develop a suite of simple OPL procedures (with full explanations) as an aid to those members who are just starting out to use OPL and know little about its features, apart from the examples in the User Guide. So that these procedures will have some practical value when the suite is complete, I am proposing to develop a simple VAT accounting set, complete with printout (to an Epson compatible printer). Each month I will add to the basic master procedure and I hope that this will prove instructive. If any of our more experienced members would like to comment on or improve any of the procedures as they are developed, please feel free to do so, so that we may all learn from the process.

	LINE NUMBERS (do not key in!)
MVAT:	1
GLOBAL filename\$(10),device\$(1)	
LOCAL option%	
START::	
option%=MENU("OPEN,CREATE,QUIT")	5
IF option%=1 or option%=2	
CLS	
PRINT "Device (A,B, or C)",	
INPUT device\$	
PRINT "Filename (8 chars max)"	10
INPUT filename\$	
filename\$=device\$+"."+filename\$	
ENDIF	
IF option%=1	
IF EXIST(filename\$)	15
OPEN filename\$,A,supplier\$,product\$,net,vat,gross	
ELSE	
CLS	
BEEP 200,300	
VIEW (1,".....no such file exists.....")	20
GOTO START::	
ELSEIF option%=2	
IF EXIST(filename\$)	
CLS	
BEEP 200,300	25
VIEW (1,".....file already exists, choose another....." name.....")	
GOTO START::	
ELSE CREATE filename\$,A,supplier\$,product\$,net,vat,gross	
ENDIF	
ELSEIF option%=3	30
STOP	
ELSE GOTO START::	
ENDIF	
MMENU:	35

This is the first procedure in the set. It provides a simple way of CREATing or OPENing the VAT file. Lines have only been numbered so that I can refer to them by number in this explanation. LINE 2 - filename\$ is declared as 10 characters to allow a full 8 character filename to be preceded by e.g. A and : (the other 2 characters) to form the complete file name. LINE 12 - is the line which combines your chosen device\$ and filename\$ separated by the required : to form the complete file name. LINES 15 - 21 check that you are OPENing an existing file, and tells you if the file does not exist before returning you to the menu at LINE 5.

LINES 22 - 27 check that a file which you have chosen to CREATE does not already exist - if it does you are returned to the menu to choose the OPEN option. LINE 35 - takes you to the next procedure which will appear in the next issue of this newsletter.

BOOKS

In case you haven't yet seen it, there is a handy book on the ORGANISER II on sale. It is especially aimed at the beginner and explains the various features of OPL in greater detail than the official handbook. As this book's author was involved with the design of the ORGANISER II, he should know what he is talking about. The book contains some handy procedures including an ingenious action game.

The book is "Using and Programming the Psion Organiser II" by Mike Shaw. It is published by Kuma Computers Ltd, Pangbourne, Berks. at £9.95 (It should be available from your local Psion Dealer)

USER HANDBOOK

There are TWO versions of the USER HANDBOOK, issued with the ORGANISER II. They are almost identical, except that the example programs have been totally changed in the latest edition and the handy skeleton database programs, which were in the early edition have now been omitted. There are, however, some new and quite useful programs included in the latest version. Psion sell both versions of the handbook at £4.00 each. Version 1 has got the parameters for the BEEP command reversed (pitch and duration). This has been corrected in the later version.

HINTS & TIPS

A PADDED CASE FOR YOUR ORGANISER

You may say that the Organiser is self-contained and needs no other case, but the DISPLAY is the only vulnerable area which is not protected when the case is closed. I have already seen one badly scratched. My Organiser lives in a padded pouch which is a nice tight fit, has a slot for a label on the outside and a soft strip to protect the display. The case was designed to be used repeatedly for sending AUDIO CASSETTES through the mail. If you have any trouble obtaining one, I have a limited number for sale at £1.50 (including postage)

STORING YOUR DATAPAKS

ROBIN FENNELL stores his datapaks in old AUDIO CASSETTE cases, which he then keeps in racks. The cases each hold three datapaks. DAVID RIHOY crams FOUR paks into a cassette case by breaking off the lugs inside the case. He also recommends that you pad the cases with plastic or cotton wool. (All this is helpful if you can afford to buy large quantities of datapaks!!)

EPROM ERASURE (DATAPAK FORMATTING)

You probably started off by taking your datapaks back to your dealer for "formatting". Others have already invested in one of Psions formatters. (I hope to have a review of these in later issues of IPSO FACTO). What is required is an EPROM ERASER. These are available from various sources, but perhaps the best value is the UVIPAC, available from Solidisk Technology Ltd, 17 Sweyne Avenue, Southend-on-Sea, Essex (tel. number - with credit card ordering facility - is 0702-354674). The price is reasonable at £20.00. As the slot in the case of this device is rather narrow, you must first gently dismantle your datapak by prising the sides apart and carefully removing the PCB with the EPROM mounted on it. This is then erased for 15 minutes, then re-assembled. I have never had one fail and the procedure is quite simple. It has the added advantage that the datapak label can remain in place.

PRINTER CONVERTER

Eidersoft Software Ltd (see Review Section for address) have just announced a Serial to Centronics Converter, to enable your parallel (centronics) printer to be used without the expense of fitting a serial interface board. You still need the Psion RS 232 Interface and the price of the converter is £39.95.

YOUR LETTERS

Q. ALEX WAYNARN would like to know if PSION have any plans to release more Application Paks.

A. I know of no plans for further paks from Psion, but "third party" developments are appearing from EIDERSOFT and WIDGET (see Review section of this issue)

Q. ERIC PEAK complains of the lack of a "hard" reset key, slow access to datapaks and small main memory.

A. The first point puzzles me rather. Surely RESET from the main menu produces the desired effect? I have not found that ACCESS to datapaks is slow, but WRITING to paks is (necessarily) rather slow (compared to the average disc access on other computers). This is probably connected to the power required to program an EPROM (21v?). I have found that using the Mains Adaptor speeds things up a little - and saves your batteries too. 16K RAM is rather small by today's standards, but many Pocket Computers get along with much less. Judicious use of SMALL procedures accessed as required from datapaks can work wonders.

Q. Dr E.L. HINES is interested in a "prioritised note pad"

A. There is a suite of procedures in the FIRST EDITION of the Psion User Handbook called "Things to do" which may fit the bill. PSION will sell you a copy of either edition for £4.00.

Q. NEIL HARRISON has an RS 232 Interface and wonders whether there is a need to buy a separate Mains Adaptor or whether it would be possible to use one of a range of adaptors which he has accumulated over the years.

A. PSION say that it is unlikely that other adapters can deliver the constant voltage without fluctuation and also say that the use of a spurious power unit will negate their warranty.

The following members are seeking any advice about interfacing the Organiser with their respective computers:

WOII J.D ROBINSON - Atari 800XL and ST Series
 DEREK GRIFFIN - Amstrad PCW 8512
 PETE TAYLOR - Amstrad PC 1512 (with Psion Exchange/PC
 Four)
 C.CROYDEN - Amstrad PC 1512

TIM ELKINGTON has an Organiser I, with 1 blank 8k datapak, all in pristine condition, for sale and is open to any reasonable offer.

WILL CHAPMAN, who runs a firm specialising in Event Organisation, would like to contact any member who would like to put his Organiser II to good use timekeeping road running and cross-country events. WILL supplies the necessary software. Those living in the London area have the best chance, but there may also be opportunities in other parts of the country.

Contact:

Will Chapman
 GBFR Limited
 Alexandra Lodge
 Alexandra Gate
 Hyde Park
 London SW7 1QH

Any helpful answers to the above, or any comments, are welcome for both publication in IPSO FACTO and/or transmission to the original enquirer.

FACTO REVIEW No.1 SOFTWARE - "TEXTBASE" from Eidersoft Software Ltd

OVERVIEW

One of the first requirements of the Organiser user is a database which offers more than the built-in one. Textbase (hereafter referred to as TB) goes some way to filling this requirement. The suite of programs is supplied on a snazzy yellow 16K datapack complete with a 23 page handbook. It is one of THREE "third party" software packages available from Eidersoft Ltd, the other two being Numbase I & II, which will be reviewed in subsequent issues of Ipsos Facto.

The first reaction after plugging in the datapak is to run it without reference to the handbook. As the package is wholly menu driven this is quite possible to anyone acquainted with computer databases, as the menu messages are self-explanatory. However, for those who want to get the most out of TB, a few minutes reading the handbook will not come amiss.

IN USE

With the datapak installed, it is possible to access TB either through PROG and RUN or by inserting TEXTBASE into the main menu. After the title credits the next menu is used to design a database FORMAT (or use one which has previously been designed). The menu offers a prompt "A:F_" to remind you that it recommends you to be logical in naming your files, with "F" being suggested as the first letter of FORMAT files. You may add a further 7 characters after the "F" to name your own FORMAT file (or you can be lazy and just use the "A:F" as your FORMAT filename,

in which case you cannot use it again unless you add further identifying letters). Assuming you are designing a format, pressing EXE after you have entered your FORMAT filename produces the message "Make New Format OK (Y/N)". To choose the "Y" option, you may press either the "Y" key or EXE.

The next menu says "Field 1" and waits for the title of your first field to be keyed in. On being given a field title TB then displays the options "Text or Numeric" (flashing cursor is over Text which will be selected if EXE or "T" is keyed. This provides TB with a clue to select KSTAT1 or KSTAT3 when data is entered later. When you have entered enough field titles pressing EXE will allow you to end the FORMAT setup (or continue to add fields if you wish).

There are two "special" fieldnames, BARCODE and DATESTAMP, and, as you would expect the first accepts barcode input via the Barcode Reader. The second automatically datestamps each record.

Next your FORMAT setup is displayed and you may proceed to name a DATA file to use the FORMAT you have just designed. Again you are prompted with "A:D" (this time plus the name which you allocated to the FORMAT which you can edit or use as required). Keying EXE produces a fleeting "Opening file" message which brings you to the next menu - the main one from which to use your new database. The options with their uses are:

```
Input Search Display Files Labels Print Off Mode
Erase and Quit
```

Input - is largely self-explanatory allowing you to input your records until you (or the memory device which you selected to hold your database) run out of steam. Text (upper-case) or Numeric are automatically selected according to your FORMAT setup and, naturally, SHIFT may be used to enter numbers into Text fields, although it is wise to remember that the reverse (Text into Numeric fields) will NOT BE ACCEPTED. For instance, if you had designated a field "Phone No" to be Numeric, it will not accept "Long Eaton 5482" as input. After each record is entered, you are asked "All correct (Y/N)".

Search - allows you to search on up to SIX "clues". If you select more than one clue then you are asked "All" or "Only-one" - allowing you to find records which meet either requirement (see later comments).

Display - Pressing "D" displays (scrolling if necessary) "(No.1)" and the contents of ALL fields of your first record on Line 2. Line 1 of the display has the legend "A/F/N/B/L/J/" plus a figure indicating the total number of records in your file. The letters stand for, respectively, Alter (current record), Forward, Next, Back, Last, Jump (to any record selected by record number).

Files - allow Copy, Delete, Export, Dir(ectory), Rename, Space. Any of these options is allowed except "Delete" on the current file (for obvious reasons).

Labels - lets you "design" a simple label layout with a default of nine lines per label and allowing you to designate which fields are to print on which lines and which to leave blank.

Print - allows you to print your records on a suitable printer, with each record using a separate line. You may print "All" or "Some" records (selection of extract similar to the Search facility described above).

Off - is interesting. It switches off the display "to conserve the battery". ON/CLEAR switches the display on again.

Mode - another interesting one. It allows selective retention of fields from the previous record - very useful for setting up a database which contains a lot of repeated information!

Erase - allows you to find a record and delete it

It is possible to escape from every option by using the ON/CLEAR key.

Finally, TB allows import/export from Sinclair QL and IBM PC, but this aspect was not tested as my other computer is a BBC Master.

COMMENT

As you may have gathered, Textbase is quite simple to use and allows the setting up and manipulation of a simple database with a minimum of trouble. However, in certain areas, there have been some surprising omissions:

1. NUMERIC fields are not able to be used in any form of calculation - they just accept numeric input. So it would not be possible to use Textbase to set up a simple VAT file, for instance.

2. The main PRINT function is very basic, just printing each record on a separate line, with fields just separated by a single space and no attempt at tabulation. I would have liked to be able to design my own format for the printed page with field selection, tabulation, headings etc.

3. The LABEL printing function is equally basic being only capable of printing single labels under each other. My printer is unable to handle such narrow tractor feed output so I like to print my labels THREE ACROSS.

4. The SEARCH function, which allows up to six search clues and appears to be quite powerful at first sight, would be much more effective if it had more options than just ALL or ANY. For instance it would be fine to be able to select various combinations of AND and OR, to allow A AND B OR C AND D OR E AND F etc.

5. The SPACE utility, which lets you see how much free space is still available on all three devices opens a "small file called LOOKUP". This is not required, as each device automatically has the MAIN file which can be used for this purpose.

6. I would have liked to have an ORDER option (for example to sort the whole file into alphabetical order). I realise this is difficult, as there is no OPL function which will do this automatically, but I still think it could have been provided.

Overall, for its price, Textbase is reasonable value. It is to be hoped that this and the other two datapak-based software from the Eidersoft stable are the first of many. I would like to see any future update offering some or all of the features mentioned above.

It is available from:

Eidersoft Software Ltd
The Office
Hall Farm
North Ockendon
Upminster
Essex RM14 3QH

PRICE £34.95

My thanks to Eidersoft for their prompt response to my request for review copies of their programs TEXTBASE and NUMBASE and to Bennetts Ltd, Nottingham for the loan of a Psion Bar Code Reader.

EPSON by Mike O'Regan

This procedure is a totally menu-driven utility for setting up an EPSON (or compatible) printer for various print styles. I have altered the NAMES of the various modes so as not to clash on the menu - e.g. "tiny" is superscript. "Length" will set the form length to 70 and perforation skip to 5 lines (for continuous stationery) and may be altered by changing the (70) which appears in the listing. Combinations may be chosen from the menu - for instance "Cond" and "Tiny" is an ideal combination for printing datapak labels!

```
EPSON:
LOCAL x%,s$(1)
s$=CHR$(27)
menu::
x%=MENU("Elite,Pica,Wide,Cond,Tiny,Ital,Bold,Length,Quit")
IF x%=1
LPRINT s$;"M";
ELSEIF x%=2
LPRINT s$;"P";
ELSEIF x%=3
LPRINT s$;"W";CHR$(1);
ELSEIF x%=4
LPRINT s$;CHR$(15);
ELSEIF x%=5
LPRINT s$;"S";CHR$(0);
LPRINT s$;"A";CHR$(5);
ELSEIF x%=6
LPRINT s$;"4";
ELSEIF x%=7
LPRINT s$;"G";
ELSEIF x%=8
LPRINT s$;"C";CHR$(70);
LPRINT s$;"N";CHRS$(5);
ELSEIF x%=9
RETURN
ENDIF
BEEP 200,400
GOTO menu::
```

MEMORY by Les Ball

This neat little set of procedures allows you to examine the contents of the WHOLE of memory (but not change anything!). Entering any hex number between 1 and FFFF will display Decimal, Hex and (if possible) the ASCII character in the address chosen. The left and right arrow keys may be used to scroll up and down memory. The EXE key allow you to insert a new number and ON/CLEAR exits from the program. Try FFCC upwards for an interesting message. Can anyone say why \$180 shows 13 \$D when first accessed and 77 \$4D M when scrolled back to!?!?


```

MEMORY:
GLOBAL a,ta,b%,d%,c$(1),h$(2)
ad::
inx:
sm::
conv:
d%=PEEKB(ta)
h%=HEX$(d%)
c%=CHR$(d%)
CLS
dispmem:
b%=GET
IF b%=6
a=a+1
ELSEIF b%=5
a=a-1
ELSEIF b%=1
STOP
ELSEIF b%=13
inx:
ENDIF
GOTO sm::
GOTO ad::

```

```

dispmem:
AT 1,1 :PRINT a;
conv:
AT 6,1 :PRINT "$";HEX$(ta)
AT 12,1 :PRINT "ASCII"
AT 3,2 :PRINT d%;
AT 7,2 :PRINT "$";h%;
AT 13,2 :PRINT c$
RETURN

```

```

inx:
GLOBAL f,s,r,p%
LOCAL h$(1),d$(4)
CLS
PRINT "Address: $";
KSTAT(3)
INPUT d$
p%=LEN(d$)
DO
h%=MID$(d$,p%-f,1)
r=ASC(h%)
IF r<64
r=r-48
ELSE r=r-55
ENDIF
r=r*16**f
s=s+r
f=f+1
UNTIL f=p%
IF s>15 OR s<11
s=s+.5
ENDIF
a=INTF(s)
RETURN(a)

```

```

conv:
IF a>=32768
ta=a-65536
ELSE ta=a
ENDIF

```

Getting at the Clock by Will Chapman

As event organisers, one of our major uses of the Psion is as a timing device and in the process of writing programs for that purpose we have discovered that there is more than one way to access the internal clock.

The manual suggests using the OPL words HOUR, MINUTE and SECOND and Mike Shaw's useful and informative book (see Books section - Ed) provides a routine on Page 192 that does just that.

However, we find that a more direct approach is to set the time to zero by POKEing the appropriate memory addresses, watching the elapsed time by displaying through the DATIM\$ function and resetting the clock back after use.

The following routine gives elapsed time on the screen.

```

TIME:
LOCAL st$(8)
st$=RIGHT$(DATIM$,8) : REM store start time
POKEB $20CA,0 : REM set hours to zero
POKEB $20C9,0 : REM set minutes to zero
POKEB $20C8,0 : REM set seconds to zero
DO
AT 5,1 : PRINT RIGHT$(DATIM$,7) : REM print HOURS:MINUTES:SECONDS
UNTIL KEY=13 : REM stop by hitting EXE key
POKEB $20CA,VAL(RIGHT$(st$,2))+VAL(RIGHT$(DATIM$,2)) : REM reset clock
POKEB $20C9,VAL(MID$(st$,4,2))+VAL(MID$(DATIM$,20,2))
POKEB $20C8,VAL(LEFT$(st$,2))+VAL(MID$(DATIM$,17,2))
RETURN

```

This routine is accurate to the second, as it is limited by the OPL word SECOND. If greater accuracy is required, it is possible to access the Psions counter which increments every 50 milliseconds (20th second). I will print the routine to do this in the next newsletter.

Conventions: All variables are printed in lower case to distinguish them from OPL words.

IPSO FACTO

NEWSLETTER OF THE INDEPENDENT PSION ORGANISER USER GROUP

VOLUME 1 * * * NUMBER 2 * * * APRIL 1987

EDITORIAL

IS THERE ANYONE OUT THERE?

Since sending out the first issue of IPSO FACTO, I am beginning to think that most of the post must have been hi-jacked that day, having had signs of life from only THREE members. It would have been nice to have had a few more, if only to offer criticism or advice about the newsletter itself. Remember, this newsletter belongs to the members and I expect SOME active participation. I would particularly like to hear from our less experienced members, with any problems they may have and would also like some programs and procedures (particularly games) from the more experienced.

I must offer an apology for a couple of slight boops on the BEGINNERS PLEASE pages of Issue 1 (nobody has complained!). Although I have the choice of two or three dot-matrix printers and one daisywheel, I finally decided to use my own AMSTRAD DMP 2000 for IPSO FACTO. As this printer has only a parallel interface, I was unable to print any of the listings in Issue 1 DIRECTLY from the ORGANISER, hence the slight mistakes in the first VAT program on the Beginners pages. However, since then I have acquired one of the neat little parallel-to-serial converters (which I mentioned before) from EIDERSOFT. In future, wherever possible, ALL listings will be printed directly from the ORGANISER in an effort to reduce (even eliminate!) errors.

I spent a day at the WHICH COMPUTER show and met quite a few interesting people. Something I found to be curious. Having left 100 IPSO Leaflets on the PSION STAND and checked that most of them had gone by early afternoon, I have only heard from ONE prospective member as a result of these leaflets. This confirms a suspicion I have had for some time - that Show visitors seem to collect leaflets like magpies, and presumably chuck them away at the earliest opportunity.

Contents:	11. Editorial	12. Beginners Please
	13. Bookfare	14. Beep Beep, Labels
	15. Secs, Keyboard	16. POKES
	17. Review No.2	18. Diary Printout
	19. Printers	

BEGINNERS PLEASE.

Not a very auspicious start to this series, as I have to admit that there were several mistakes in the first listing last month. I was pondering whether to wait for someone to draw my attention to them, but, as no-one has, I am not getting around the problem by pretending that I was just testing you. You will have got the general idea from the accompanying notes last time, but I am completely re-writing the first procedure, MVAT, below with various improvements. To start you off, I can point to the fact that I have shortened the variable names to save memory and also arranged the display of messages to be more convenient.

```
MVAT:
GLOBAL fn$(10),d$(1)
LOCAL o%
START::
o%=MENU("Open,Create,Quit")
IF o%=1 OR o%=2
CLS
PRINT "Device"
PRINT "(A,B, or C)",
INPUT d$
PRINT "Filename(8 chars max)",
INPUT fn$
fn$=d$+"."+fn$
ENDIF
IF o%=1
IF EXIST(fn$)
OPEN fn$,a,sp$,pr$,n,v,g
ELSE
CLS
BEEP 200,300
VIEW(1,".....no such file exists.....")
GOTO START::
ENDIF
ELSEIF o%=2
IF EXIST(fn$)
CLS
BEEP 200,300
VIEW(1,"...file already exists...choose another name...")
GOTO START::
ELSE CREATE fn$,a,sp$,pr$,n,v,g
ENDIF
ELSEIF o%=3
RETURN
ELSE GOTO START::
ENDIF
mmenu:
```

Now the next two procedures:

```
MMENU:
LOCAL o%
o%=MENU("Append,Display,Delete,Listall,Print,Quit")
IF o%=1
append:
ELSEIF o%=2
display:
ELSEIF o%=3
delete:
ELSEIF o%=4
list:
ELSEIF o%=5
print:
ELSEIF o%=6
CLOSE
RETURN
ENDIF
```

Just about self-explanatory?

```
APPEND:
WHILE 1
CLS
PRINT "To or From"
INPUT a.s$ :REM this is the name of the customer
IF LEN(a.s$)=0 :REM stop inputting records by pressing EXE
RETURN
ENDIF
PRINT "Prod/Svce"
INPUT a.pr$
PRINT "Total (inc VAT)"
INPUT a.g
a.n=a.g/1.15 :REM calculates NET from GROSS
a.v=a.g-a.v :REM calculates VAT
APPEND
ENDWH
```

Note: The procedures so far will allow you to CREATE and/or OPEN a VAT file and also enter records. The remaining procedures for completing the suite will be published in future issues of IPSO FACTO. Any questions or suggestions for improvements are welcome.

BOOKFARE

PSION say that the TECHNICAL MANUAL for the Organiser is almost ready and having seen a draft copy I can say that this enormous volume is going to be great value for money for anyone with more than just a passing interest in the Organiser. I hope that by the time the next newsletter is published there will be more definite news.

There is also advance notification of a newsletter to be published by PSION sometime in the summer.

BEEP BEEP

In case you hadn't realised, the Organiser has quite a powerful sound facility. The single channel sound generator is capable of producing more or less musical sounds, between 28 and 11815 Hz. I have included a simple procedure below to enable you to listen to the full range available. For instance, entering the number 2105 will produce a standard "A" at "concert pitch" (A=440 hz). It is quite interesting to test out your friends - you will probably find that some (older?) people cannot hear the top end of the range. The theory is that age reduces the ear's response to high frequencies, so it seems that, by the time you can afford the best quality hi-fi, you are too old to be able to enjoy the full range of sound.

There are some interesting sound experiments in "Using and Programming the Psion Organiser II" (mentioned in Bookfare).

```
TONE:
LOCAL T%,F%
DO
PRINT "1 - 32000"
INPUT T%
F%=INT(921600/(78+T%))
PRINT "Freq.",F%;"Hz"
BEEP 500,T%
GET
UNTIL T%=0
```

Now for a bit of work for YOU. Can you write a procedure to produce a real-time (digital) clock which produces "WESTMINSTER CHIMES" every quarter of an hour and which synchronizes exactly with the correct time when "chiming" the hours. I will publish the procedure which I consider to be the neatest way of achieving this feat of music and timing!

PRINT YOUR OWN LABELS

I wrote the following very simple routine to do the specific job of printing address labels of not more than seven lines, in expanded mode, on an "Epson compatible" printer.

```
LABEL:
GLOBAL a$(7,25),c%
c%=1
DO
PRINT "Line",c%
INPUT a$(c%)
IF LEN(a$(c%))=0
GOTO PR::
ELSE c%=c%+1
CLS
ENDIF
UNTIL c%>7
PR::
LPRINT CHR$(27);"W";CHR$(1);
c%=1
DO
LPRINT a$(c%)
c%=c%+1
UNTIL c%>7
```

SECONDS by Will Chapman

Following the TIMING procedure in the last issue, here is an extension which will allow timing to 1/10th second. As you can see, it uses the SYSTEM CLOCK to achieve this. The procedure may be combined with the previous one which will then enable elapsed timing to 1/10th second. Beginners need not fear crashing the system with the POKES in these routines - Will assures us that they are quite safe!

```
SECS:
LOCAL A%,B%,C%,D$(13)
C%=PEEK($20C8)
IF PEEK($20C8)=C%+1 :REM SYNCHRONISE SYSTEM & TIME CLOCKS
POKE($20CC),0
ENDIF
DO
IF PEEK($20CC)>=20 :REM EVERY (20*50ms)
POKE($20CC),0 :REM ZERO COUNTER
ENDIF
B%=PEEK($20CC)/2 :REM SET VARIABLE B% TO 10ths OF A SECOND
D$=RIGHT$(DATIM$,5)+". "+GEN$(B%,1)
AT 11,1 :PRINT " "
AT 1,1 :PRINT D$ :REM DISPLAY MINS,SECS,and 10ths ON SCREEN
UNTIL KEY=13
RETURN
```

KEYBOARD by Steve J.T. Knight

A tiny procedure which will return the ASCII code (in decimal and hex.) for ANY key (except SHIFT) on the Organiser keyboard. Q will abort the routine.

```
KB:
LOCAL ch$(1),v%
PRINT "Enter char"
DO
v%=GET
ch$=CHR$(v%)
CLS
PRINT "DEC: "
AT 6,1 :PRINT v%
AT 12,1 :PRINT ch$
AT 1,2 :PRINT "Hex: "
AT 6,2 :PRINT HEX$(v%)
UNTIL UPPER$(ch$)="Q"
GET
RETURN
```

MAIN ORGANISER POKES

\$0097 current device
\$00A2 which device for FIND, SAVE, ERASE (0=A, 1=B, 2=C)
\$00A3 0=SAVE, 1=FIND, 2=ERASE

\$2016 PEEKW(\$2016) is start of file buffer (first byte is
length of record <=254)

\$2018 b

\$201A c

\$201C d

\$FFE9 current version of operating system

RS 232 Link POKES

\$20F9 byte gives RS 232 version number in hex (18=12 hex=V.1.2)

\$2150 baud rate (8=4800, 9=9600)

\$2151 parity% (4=NONE)

\$2152 bit% (8=8 bits - most others seem to give 7)

\$2153 stop bits

\$2154 HANDSHAKE (0=NONE, 1=RTS/CTS, 2=XON/XOFF, 3=XMODEM)

\$2155 ECHO (0=HOST, 1=LOCAL)

\$2156 WIDTH

\$2157 REOL number of chars 0,1,2 then \$2158-9 REOL\$

\$215A REOF\$ " " "

\$215D RTRN\$ " " "

\$2160 TEOL\$ " " "

\$2163 TEOF\$ " " "

\$2169 TTRN\$ " " "

\$241B length of string below

\$241C string for comms e.g. A:FRED

"Cecil's POKES" (to alleviate RS 232 strain - only required with
early versions of RS 232 interface, check your version by
PEEKW-ing \$2169)

POKEW \$2169,\$108

POKEW \$216B,\$56

POKEB \$E4,0

FACTO REVIEW No.2
SOFTWARE - FILE UTILITY PROGRAMS from WIDGET SOFTWARE

As you no doubt noted from the advertisement mailed with your first copy of IPSO FACTO, WIDGET SOFTWARE are filling an obvious gap by providing inexpensive utilities to enhance the Organiser's built-in filing system (SEARCH, SORT, SELECT and PRINT) and also provide a readymade alternative database (VU).

SEARCH - allows you to search any MAIN file and, unlike the built-in FIND option it allows you to narrow the choice by providing up to 5 SEARCH clues. These work on an AND basis - i.e. finding records which contain clue 1 AND clue 2 AND clue 3 etc. I feel it would have been more helpful to have included an OR option, enabling combinations such as (clue 1 AND clue 2) OR (clue 3 AND clue 4) etc.

SORT - because of the way that data is stored on a datapak there is no simple way of sorting records (say alphabetically) using any of the OPL commands. Widget's SORT utility goes some way to rectifying this by providing a means of sorting a whole file into ascending alphabetical order (using the first "field"). It is recommended that you allocate a blank pak to receive the SORTed file. Again it would be more useful to have the option to be able to sort the file on the basis of one of the other fields

SELECT - enables you to produce an "extract" file from your MAIN file and write this to another datapak. This apparently simple device can be extremely useful in practice, enabling you, for instance to split an oversize file into a number of logical sub-files.

PRINT - again an obvious, and very practical utility. This gives you the options of printing the whole of any MAIN file or of viewing the record on the Organiser's display before deciding whether to print the record or not. This utility also has a simple alternative to the SETUP subroutine contained in the RS 232 COMMS program. This utility would be much more useable if it allowed you some choice of format for your printout e.g. order of fields printed and choice of whether in tabulated columns or with each record having its own mini printout occupying several lines etc.

COMMENT - all the above utilities ONLY work with MAIN files which have been set up either by the Organiser's built-in SAVE commands or input via the other Widget program VU mentioned below. Although I realise that there are some practical difficulties, I would have liked the above utilities to be available for ANY file set up in OPL.

VU

VU is a small simple self-contained filing system which can use either:

1. The MAIN files (on any device) - provided that they have been structured to only four fields.

or

2. Another file which you may name yourself, again on any device and restricted to four fields.

Having selected one of the MAIN files or started a new one you may select your next option from a small menu on the top line, which allows you to EDIT, FIND, INSERT, scroll back and forth through your database using the < and > keys, or QUIT. VU is quite convenient to use and quick to set up a new file, provided you can restrict your records to just four fields, and character fields at that. I would like to see a rather more complex version which would allow you to select the number and types of field and then, for instance, allow some sort of maths functions to work on numeric fields.

Just to remind you the above procedures are available from:

Widget Software
 Freepost
 21 Landrock Road
 London N8 9HR.

IPSO member and head of WIDGET, MARK NEEDHAM, informs me that he will give a free copy of his latest DIARY PRINTOUT program DPRINT to any IPSO member ordering WIDGET SOFTWARE. Order forms were included with IPSO FACTO 1 or are available on request from WIDGET.

DIARY PRINTOUT PROGRAM by Barry Thomas (courtesy of PSION)

```

PRDIR:
LOCAL base%,len%,a$(10),c%
a$="Alarm set "
base%=PEEKW($2004)
A::
len%=PEEKb(base%)
IF len%=0
PRINT " END OF DIARY"
PRINT " Press any key"
LPRINT CHR$(12)
RETURN
ENDIF
LPRINT (PEEKb(base%+3)+1);"/";
LPRINT (PEEKb(base%+2)+1);"/";
LPRINT PEEKb(base%+1);" ";
LPRINT PEEKb(base%+4);":";
IF PEEKb(base%+5)=0
LPRINT "00 ";
ELSE
LPRINT "30 ";
ENDIF
IF PEEKb(base%+6)=0
LPRINT "No ";a$
ELSE LPRINT a$;PEEKb(base%+6)-1;" minutes early"
ENDIF
c%=PEEKb(base%)
base%=base%+7
LPRINT " ";
DO
LPRINT CHR$(PEEKb(base%));
c%=c%-1
base%=base%+1
UNTIL c%=0
LPRINT :LPRINT
GOTO A::

```

PRINTERS

Although the Organiser is quite a handy self-contained computer, its use is greatly extended by coupling it to a printer (or another computer) through the RS 232 interface. This fact is obviously recognised by many of our members, judging by the number who already own an interface, so this little piece is written with the others in mind.

The first limitation which is imposed by the RS 232 is that it is a SERIAL device, and most common and/or cheap printers have only a PARALLEL connection. Of course, if you have another computer (and many members have at least one!), you may already have a printer - probably with a CENTRONICS (PARALLEL) connection. It is possible with most printers to fit an internal SERIAL board to provide a dual interface, but by and large these are fairly expensive. I have a BBC Master 128 computer and it is possible to pass the material to be printed from the Organiser to the Beeb through the RS 232/423 interface and thence to the printer via the normal parallel connection, but this is a roundabout way of achieving printout from the Organiser. A good compromise is the SERIAL to PARALLEL converter (which I have mentioned elsewhere) which is available from Eidersoft Software Ltd at £39, which is certainly cheaper than adding the necessary conversion to your printer (unless you are clever enough to do your own conversion). This device is quite small (about 2"x3"x1") and works at 9600 baud, 8 data bits, 2 stop bits and RTS/CTS.

If the Organiser is your only computer and you want a printer for its exclusive use, then it is possible to buy one with just a serial interface, but you will still need a PSION "gender-bender" to convert the male end of the RS 232 to match your printer interface. Reasonable full-sized (A4) printers can now be had for less than £100 if you shop around.

A rather unusual printer which is used by one or two members is the TANDY CGP-115. It is unusual in that it is a three colour PLOTTER which actually draws everything using tiny ballpoint pens. It is capable of being driven by the Organiser to produce some quite sophisticated graphics, as well as being able to produce text in many different sizes (and any of the four colours red, blue, green or black). One of the drawbacks is that the pens are quite expensive, as are the rolls of 5" paper.

STOP PRESS

As you will have seen, I am now accepting advertising leaflets for inclusion with each copy of IPSO FACTO. If you wish to take advantage of this service, please contact me about this rather effective way of advertising your services or sales

IPSO FACTO

NEWSLETTER OF THE INDEPENDENT PSION ORGANISER USER GROUP

VOLUME 1 * * * NUMBER 3 * * * MAY 1987

Editorial Address:

130 Stapleford Lane, Beeston, Nottingham, NG9 6GB

EDITORIAL

First a "thank-you" to those members who drew my attention to the fact that there was NO ADDRESS anywhere in the newsletter - which probably accounts for the lack of response so far. I hope that including it in the newsletter heading above will encourage more to write in. Actually, I am now beginning to get some response, with programs, queries, etc., although the same few names keep cropping up. I am particularly indebted to those members who have pointed out blunders in the first two numbers. It is quite a job to make sure that there are no mistakes in some 10 pages of print, when you are your own proof-reader. My wife kindly reads through the whole of each newsletter checking for typos or spelling mistakes, but is unable to check that the contents are up to scratch.

Last week I made my first visit to PSION, along with a blind friend, LES BALL, who is also an enthusiastic member of IPSO. He was demonstrating the Organiser attached to a DOLPHIN Speech Synthesiser (which is an excellent little unit, hardly bigger than the Organiser, with comprehensive speech facilities). I found all the people at Psion House to be most helpful and friendly. It is hoped that before this newsletter is published there will be news of some important Organiser developments (this accounts for the delay beyond the normal 1st of the month publishing date, for which I apologise.)

STOP PRESS - NO NEWS YET FROM PSION, in spite of their promise of a press release with details of "exciting new developments" - This is the LAST time I will delay publication for any reason, except for World War III or other emergency.

CONTENTS

- | | |
|-----------------------|------------------------------|
| 20. Editorial | 21. PHONE Review |
| 22. On-Line Organiser | 23. Bits & Pieces |
| 24. Beginners Please | 25-30. Procedures & Programs |

FACTO REVIEW No.3
SOFTWARE - PHONE from LANGDALE BUSINESS BUREAU LTD

For anyone who makes a lot of phone calls, especially those in business who have to charge the cost of phone calls to their customer, PHONE could become quite indispensable, as it provides an accurate record of the duration and cost of ANY phone call.

IN USE

PHONE comes on an 8k datapak, complete with detailed instructions including a table of standard charges rates. After inserting the datapak in drive C the program is booted, either from the PROG menu or from an entry in the main menu as normal. The first thing it to set up the rates for each zone from the table provided. At this stage you may prefer to make any additions to cover VAT or other charges. Once this is done (a matter of a minute or so) PHONE is ready to use.

This program has been very well thought out so that its use is a piece of cake. From now everything is menu-driven. The menus are unusual, as each entry takes up a complete screen. Further options are obtained by using the DOWN ARROW key before pressing EXE. The sections start with LOCAL OR NATIONAL calls and the various options within this group (e.g. calls less than 56 km, over 56 km, calls to Irish Republic, calls to Mobile Phones). BTs new low-cost long distance rates may also be called up, and here it is possible to get a list of the destinations which apply from your area.

The INTERNATIONAL CALLS menu is particularly comprehensive with a choice of 171 countries, which are very easily selected by keying in the initial letter, then using the UP and DOWN ARROWS to find the exact country, if required. For example, keying in P immediately brought up PAKISTAN.

Once you have selected the right zone from the menu, a further press of EXE displays your choice and an ingenious moving IKON of a telephone, accompanied by a realistic trim-phone sound. All that remains to be done is to press any key again when your call is answered and the timing begins (with a suitable click). When your call is finished the whole process is stopped by pressing any key.

The really ingenious part of this program is that it uses the Organisers built-in CLOCK-CALENDAR to automatically select the appropriate rate for the time-of-day and day-of-week of your call!

SUMMARY

PHONE is one of those obvious uses for the Organiser which is so obvious that it makes you think "why didn't I think of that?" The author is to be congratulated on devising a complete and well thought out application.

PHONE is obtainable from:

Langdale Business Bureau Ltd
41 Bath Road
Old Town
Swindon SN1 4AS

Tel: 0793 617397

The cost is £15 + VAT (or the program is available FREE to anyone buying an Organiser through the Bureau)

ON-LINE ORGANISER

It wasn't until my main computer went down, that I had even considered using my Psion Organiser as a terminal to a modem. The problem was, that being a subscriber to Microlink, mail collects in your mailbox, and incurs storage charges, unless it is read and then deleted. As the Psion was the only other machine I had with communication capabilities, I decided to see if it could be used to access Microlink, and other bulletin boards. The first problem was that of connecting the Psion to the modem. I use a Miracle Technology WS2000 manual modem, which has a standard 25 way 'D' socket. The Psion RS232 also has this socket, so two 'D' plugs were obtained. After much trial and error (especially error), I found the following connections worked.

MODEM	PSION
2 -----	3
3 -----	2
6 -----	6
20 -----	20
7 -----	7

Pins 4 and 5 at the Psion end should be connected together. Having solved the hardware problems, next was the communication protocols. The Psion RS232 cannot handle split baud rates, ie. 1200/75 so I set up for 300 baud, no parity, one stop bit. This is pretty much standard for most bulletin boards, although Microlink is 7 bits, even parity, 1 stop bit. Handshaking is XON/XOFF, and REOL is <CR><LF>.

The first board I contacted was a local bulletin board. The biggest limitation in using the Psion is the small display. Because most boards are designed to be read by a 40 or 80 column terminal, text on the Psion tends to be fragmented, and you have to keep pretty alert to read it all. Of course, what is really needed is a dedicated Psion bulletin board with 16 column format! (Anyone out there interested?) After this, I contacted Microlink, and was able to read and delete mail.

The other limitation of the Psion is the small keyboard. I refrained from trying to send mail while on Microlink, because of the time it would have taken to type the message in on the Psion. However, Microlink does have a facility called WPMAIL, which enables you to prepare a message or a number of messages beforehand, and then upload them. I thought this could be a useful facility, so firstly, I wrote a short routine to enable me to enter and store a text file on the Psion. I then prepared a message, stored in on drive 'A' and logged on to Microlink. The problem with sending a file is that you have to exit the TERM routine to TRANSMIT a file. In practice, this didn't prove to be too much of a problem, as long as you go back to TERM as soon as the message has been sent.

All in all, a very successful experiment. I had achieved what I had wanted to do, read and delete e-mail, and had also been able to send it. I am next going to obtain one of the small battery operated modems for truly portable communication.

Simon Webb
Telecom Gold mailbox 72:MAG80001

BITS & PIECES

SPECIAL INTEREST SUB-GROUPS

Many of our members have special interests, mainly to do with interfacing the Organiser to their particular desk-top computer, etc. I am therefore considering publishing names and addresses of members with details of their interests, so that they will have the facility of contacting other members with similar interests or problems. However, I am reluctant to publish ANYONE'S details without their permission, so I am asking anyone who has informed me of their special interests, but WOULD NOT LIKE ANY DETAILS PUBLISHED, to contact me before the next issue of IPSO FACTO is due for publication (1 June '87).

FOR SALE

Maths Pak for the Organiser II complete with manual- £20 (ono) - Please ring ALLEN BROWN on Sandy (0767 80311), evenings.

ORGANISER II XP £100 and several 16K (£12) and 32K (£24) datapaks - apply to: STEVE J.T. KNIGHT, GEAC Computers, Hollywood Tower, Cribbs Causeway, Bristol BS10 7TW.

PROCEDURE TIPS

SIMON WEBB points out that the BEEP parameters as listed in the Psion Handbooks are different to those in "Using & Programming the Psion Organiser II". He says that he prefers the latter. Here they are for comparison:

```
Psion Handbooks - f%=INT(921600/(78+t%))
U & P t P O II - f%=INT(921600/(39+t%))
```

Try altering my TONE procedure.

PSION FILES by Simon Webb

It is probably a little known fact that any data file stored on the Psion Organiser can be read without knowing its original field definitions.

For instance, say we create a file:

```
CREATE "A:TEST",A,A$,B%,C.
```

This creates a file with a logical file name "A" and three fields A\$-a string, B%-an integer and C-a floating point number. This file can be read without knowing this information, by simply designating each field to be a string. So the file can be opened and read with:

```
OPEN "A:TEST",A,A$,B$,C$
```

You are probably wondering what you do in you don't know the number of fields that have been defined. In that case, you simply define "too many" string fields:

```
OPEN "A:TEST",A,A$,B$,C$,D$,E$,F$,G$,H$
```

In this case, only A.A\$,A.B\$ and A.C\$ will be assigned values and the rest will return empty strings. The above example will read any data file up to 8 fields, but as a file can have up to 16 fields, to read ANY file would require this number of string fields.

ORGANISER DESIGN

In the POCKET COMPUTER field there are only two instruments (as far as I know) which use the UPRIGHT (calculator-like) format - the Organiser and the Hewlett-Packard 41 series. It seems that all the others have opted for a "landscape" format, usually with a QWERTY keyboard. DEREK GRIFFIN asks whether "Psion had considered (perhaps at the expense of some compactness) orientating the whole instrument through 90 degrees and placing a longer display on top of a QWERTY keyboard....". As far as I can ascertain, Psion have no such plans. In any case, having used a few "horizontal" pocket computers, I feel personally that the upright format is much easier to use when actually HAND-HELD. To make the best use of a QWERTY keyboard, both hands are needed to press the keys which means resting the computer on some handy surface. DEREK also wondered why the Organiser cover could not have extended over the display, thereby protecting the LCD glass. He remarks that, when the case is closed, the display serves no useful purpose. I would say that the only exception to this is when the Organiser switches on under DIARY control. Well, members, what do you think?

BEGINNERS PLEASE

Here are the next two procedures in our VAT suite. The first allows you to VIEW the contents of the whole file and the second is for the purpose of FINDing any record and ERASEing it if required.

DISPLAY:

FIRST

DO

```
VIEW(1,a.sp$+" "+a.pr$) : REM first two fields on the top line
VIEW(2,"Gross "+FIX$(a.g,2,10)+"....Net"+FIX$(a.n,2,10)+"....VAT
 "+FIX$(a.v,2,10)+"....") : REM remaining fields on line 2
UNTIL EOF
```

DELETE:

LOCAL cl\$(8),yn\$(1)

PRINT "Clue",

INPUT cl\$

WHILE FIND(cl\$)

VIEW(1,a.sp\$+a.pr\$)

PRINT "Erase (Y/N)";CHR\$(63)

yn\$=GET\$

IF UPPER\$(yn\$)="Y"

ERASE

ENDIF

ENDWH

Next month I will complete the suite, incorporating any worthwhile improvements to any of the routines.

PROGRAMS & PROCEDURES

We have a mixed bunch this month from yet another Timing procedure from Will Chapman to a routine for the electricians among you. The TUNE program by SIMON WEBB is very ingenious in that it simulates TWO PART HARMONY using the Organiser's SINGLE channel.

DELAYED-ACTION PROGRAM RUNNING

It may not have occurred to everyone that it is possible to use the Organiser's built-in ALARM or DIARY functions to control the delayed-action running of a program. In other words, it is possible to tell the Organiser to SWITCH ON at a predetermined time and date, run a specified program, then quietly switch itself off! Using, say, another device connected through the RS 232, this could be an extremely useful routine. It relies on the ability of the Organiser to resume operation at exactly the right spot, either when the auto shut-down has occurred, or in this case when it has met an OFF command. So it only remains to include OFF in this little routine to ensure that it will carry on with the next command after the Organiser has been switched on by either an ALARM or a DIARY ALARM. I have switched off the actual alarm SOUND to conserve battery power.

```
DELAY:
POKEB $A4,1
OFF
REM Enter name of program you wish to run on this line.
```

Note: If you do not wish to mute the alarm, omit Line 2. If the program you are calling uses BEEP then add a line immediately after the OFF line as follows: POKEB \$A4,0. It is not a good plan to use this program to operate too far ahead, as the OFF command only puts the Organiser into SLEEP MODE, which uses more battery power than a normal switch-off. If you want EXACT timing on your delay, remember that the ALARMS both last for one minute (even if the sound has been muted).

TIMER by Will Chapman

This allows you to input a number of MINUTES and will countdown, display the remaining time, then BEEP when the time has elapsed.

```
TIMER:
LOCAL e%,st$(8),m$(3),s$(3),h%
AT 5,1 :PRINT RIGHT$(DATIM$,8)
AT 1,2 :PRINT "START:",
INPUT h%
e%=h%
h%=h%-1
CLS
st$=RIGHT$(DATIM$<8)
POKEB $20CA,0
POKEB $20C9,0
POKEB $20C8,0
DO
m$=GEN$(PEEKB($20C9)-h%,3)
s$=GEN$(ABS(PEEKB($20CA)-59),2)
IF LEN(s$)=1
s$=":0"+s$
```

```

ELSE
s$=" "+s$
ENDIF
IF LEN(m$)=1
m$="-"+m$
ENDIF
AT 5,1 :PRINT m$+s$
UNTIL m$="-0" AND s$=":00"
POKEB $20CA,VAL(RIGHT$(st$,2))
POKEB $20C9,VAL(MID$(st$,4,2))+e%
POKEB $20C8,VAL(LEFT$(st$,2))
DO
BEEP 200,200
AT 5,1 :PRINT RIGHT$(DATIM$,8)
UNTIL KEY=13
RETURN

```

DATE STORAGE ROUTINES by Carl Burgess

Here are two date storage routines that I have found useful. Each date is stored as a two-byte integer. It only runs from about Nov '85 to somewhere in 1996, but this should be sufficient for most applications.

The routine to output a date string is reasonably fast, so it is possible to use the following construct when entering dates:

```

+-> print date string
|   get key press
|   increment/decrement date number
+<- print odn$:(date number)

```

In the top level procedure you need GLOBAL yn%(10),mn%(12)

```

yn%(1)=366
yn%(2)=731
yn%(3)=1096
yn%(4)=1461
yn%(5)=1827
yn%(6)=2192
yn%(7)=2557
yn%(8)=2922
yn%(9)=3288
yn%(10)=3653
mn%(1)=0
mn%(2)=31
mn%(3)=59
mn%(4)=90
mn%(5)=120
mn%(6)=151
mn%(7)=181
mn%(8)=212
mn%(9)=243
mn%(10)=273
mn%(11)=304
mn%(12)=334

```

ROUTINE TO CALCULATE DATE NUMBER

```

DNM%:(d%,m%,y%)
RETURN yn%(y%-1984)+mn%(m%)*d%-((mn%>2) AND ((y% AND 3)=0))

```

ROUTINE TO RETURN DATE STRING FROM DATE NUMBER

```
ODN$: (d%)
LOCAL a%, b%, c%
DO
a%=a%+1
UNTIL yn%(a%)>=d%
c%=a%+83
b%=d%-yn%(a%-1)
a%=12
WHILE b%<=mn%(a%)-((c% AND 3)=0 AND a%>2)
a%=a%-1
ENDWH
b%=b%-mn%(a%)+(a%>2 AND (c% AND 3)=0)
RETURN=mid$("SUNMONTUEWEDTHUFRISAT"<(3*(d%-7*((d%-1)/7))-2),4)+
" "+GEN$(b%,2)+"
"+MID$("JANFEBMARAPR MAYJUNJULAUGSEPOCTNOVDEC",3*a%-2,4)+" "+
GEN$(c%.2)
```

The following pages are listings of:

TUNE (including data) by SIMON WEBB - if you are as clever as Simon you should be able to write your own data for CREFILE to do a series of brilliant arrangements of BACH's two-part inventions. Simon's "Tune" is vaguely reminiscent of an old German Christmas Carol. The "two-part harmony" is obtained by rapidly alternating from one note to another, (hence the buzzing sound) on the SINGLE channel of the Organiser's BEEP

PWR by STEVE J.T. KNIGHT - a program to calculate WATTS, AMPS, VOLTS and OHMS from any pair of entered values.

COLCODES by STEVE J.I. KNIGHT - this routine will calculate the value of a resistor if its colour codes are entered.

NOTE: Next months programs include a simple TEXT-HANDLER and a routine to COPY-PROTECT any datapak.

```

CREFILE:
LOCAL C%
C%=1
CREATE "A:TUNE",A,A%,B%
DO
  CLS :PRINT "NOTE : "; :INPUT A.A%
  PRINT "NOTE : "; :INPUT A.B%
  APPEND :C%=C%+1
UNTIL C%=129
CLOSE

```

```

196,550,196,550,183,486,183,486,159,429,159,429,183,403,183,403,
196,355,196,355,196,355,196,355,225,403,225,403,225,403,225,403,
257,429,257,429,225,429,196,429,183,486,183,486,196,550,196,550,
225,355,257,355,225,355,196,355,225,355,225,355,225,355,225,355,
196,550,196,550,183,486,183,486,159,429,159,429,183,403,183,403,
196,355,196,355,196,355,196,355,225,403,225,403,225,403,225,403,
257,429,257,429,225,429,196,429,183,486,183,486,196,550,196,550,
225,355,257,355,225,355,196,355,257,550,257,550,257,550,257,550,
159,429,159,429,183,486,183,486,196,550,196,550,225,355,225,355,
257,429,257,429,225,550,196,550,225,355,225,355,196,403,183,403,
159,429,159,429,183,486,183,486,196,550,196,550,225,355,225,355,
257,429,257,429,225,550,196,550,225,355,225,355,225,355,225,355,
196,550,196,550,183,486,183,486,159,429,159,429,183,403,183,403,
196,355,196,355,196,355,196,355,225,403,225,403,225,403,225,403,
257,429,257,429,225,429,196,429,183,486,183,486,196,550,196,550,
225,355,257,355,225,355,196,355,257,550,257,550,257,550,257,550.

```

```

TUNE:
LOCAL A%(128),B%(128),C%,D%
C%=1
OPEN "A:TUNE",A,A%,B%
DO
  A%(C%)=A.A% :B%(C%)=A.B%
  NEXT :C%=C%+1
UNTIL EOF
CLOSE :CLS :PRINT "          TUNE          By Simon Webb" :GET :D%=1
DO
  C%=1
  DO
    BEEP 13,A%(D%) :BEEP 13,B%(D%) :C%=C%+1
  UNTIL C%=10
  D%=D%+1
UNTIL D%=129

```

The first procedure, CREFILE is entered and run. The data in the list can then be entered to create the data file for the tune. Once this data has been entered, CREFILE can be deleted. The second procedure, TUNE can then be entered and run. After a few seconds, a message will appear. On pressing any key, the Psion will then play a tune in two part harmony.

```

COLCODES:
LOCAL bd0%,bd1%,bd2%,bd3%,mult%,tol%,nbnds%,out$(12),bd$(63),mlt$(52),tol$(22)
bd$="BLACK,BROWN,RED,ORANGE,YELLOW,GREEN,BLUE,VIOLET,GREY,WHITE,QUIT"
mlt$="SILVER,GOLD,BLACK,BROWN,RED,ORANGE,YELLOW,GREEN,BLUE"
tol$="BROWN,RED,GOLD,SILVER"
ST:: CLS
out$=""
PRINT"no of bands"
AT 1,2 : INPUT nbnds%
IF nbnds%=0
RETURN
ENDIF
IF nbnds%>4
bd0%=MENU(bd$)-1
BEEP 50,5000
out$=NUM$(bd0%,1)
ENDIF
bd1%=MENU(bd$)-1
BEEP 50,5000
IF bd1%=10 OR bd0%=10
RETURN
ENDIF
bd2%=MENU(bd$)-1
BEEP 50,5000
mult%=MENU(mlt$)
BEEP 50,3000
tol%=MENU(tol$)
BEEP 100,5000
CLS
PRINT"Resistor value"
AT 1,2
IF mult%=1
out$=out$+"R"
ENDIF
IF mult%=4 AND nbnds%>4
out$=out$+"K"
ENDIF
out$=out$+(num$(bd1%,1))
IF mult%=2
out$=out$+"R"
ELSEIF mult%=5
out$=out$+"K"
ENDIF
out$=out$+NUM$(bd2%,1)
IF mult%=3
out$=""
ELSEIF mult%=9
out$=out$+"000K"
ELSEIF mult%=10
out$=out$+"00000K"
ENDIF
PRINT out$
GET
CLS
PRINT"Tolerance"
AT 1,2
IF tol%=1
PRINT"1%"
ELSEIF tol%=2
PRINT"2%"

ELSEIF tol%=3
PRINT"5%"
ELSEIF tol%=4
PRINT"10%"
ENDIF
GET
GOTO ST::

```

```

PWR:
LOCAL cnt%,opt%,w,i,v,r,l%,d$(16)
ST::
v=0.0
i=0.0
r=0.0
w=0.0
cnt%=1
S1:: DO
opt%=MENU("Watts,Amps,Volts,Ohms,Quit")
IF opt%=1
INPUT w
ELSEIF opt%=2
INPUT i
ELSEIF opt%=3
INPUT v
ELSEIF opt%=4
INPUT r
ELSE
RETURN
ENDIF
cnt%=cnt%+1
UNTIL cnt%>2
IF v>0.0
IF r>0.0
i=v/r
w=i*v
ELSEIF i>0.0
w=i*v
r=v/i
ELSEIF w>0.0
i=w/v
r=v/i
ENDIF
ELSEIF i>0.0
IF r>0.0
v=i*r
w=i*v
ELSEIF w>0.0
v=w/i
r=v/i
ENDIF
ELSEIF w>0.0 AND r>0.0
i=SQR(w/r)
v=i*r
ENDIF
CLS
d$=GEN$(v,7)+"V "
l%=LEN(d$)
d$=d$+GEN$(i,15-l%)+ "A"
l%=15-LEN(d$)
IF l%>0
d$=GEN$(v,7+l%)+"V "
l%=LEN(d$)
d$=d$+GEN$(i,15-l%)+ "A"
ENDIF
PRINT d$
AT 1,2
d$=GEN$(r,7)+CHR$(244)+" "
l%=LEN(d$)
d$=d$+GEN$(w,15-l%)+ "W"
IF l%>0
d$=GEN$(r,7+l%)+CHR$(244)+" "
l%=LEN(d$)
d$=d$+GEN$(w,15-l%)+ "W"
ENDIF
PRINT d$
IF UPPER$(GET$)="Q"
RETURN
ENDIF
GOTO ST::

```

IPSO FACTO

NEWSLETTER OF THE INDEPENDENT PSION ORGANISER USER GROUP

VOLUME 1 * * * NUMBER 4 * * * JUNE 1987

Editorial Address:

130 Stapleford Lane, Beeston, Nottingham, NG9 6GB

EDITORIAL - THE SIXTY-FOUR KILOBYTE QUESTION

One of the strong points and at the same time a weakness of the Organiser is the use of EPROMS as the mass storage medium. The datapak system is one of the features which makes the Organiser stand out from the opposition. However, something which has bugged me (and quite a few other members) for some time is the PRICE of datapaks. While I can understand that a datapak contains as rather specialised printed circuit as well as an EPROM, this hardly accounts for the prices charged for datapaks, especially the larger capacity ones. It seems that the prices increase GEOMETRICALLY so that the 128k datapak costs more than the Organiser itself! I have heard that the bigger paks do not sell very well. Maybe this is a chicken and egg situation - if the paks were cheaper perhaps they would sell better.

As some of you already know, there IS a way to buy a 64k datapak for just £29.95 (compared with the list price of £79.75). All that is required is an OXFORD DICTIONARY SPELLING CHECKER. When the label is removed it is possible to FORMAT (erase) the pak, which then becomes a blank 64k datapak!! What is rather worrying is that if Psion can sell a 64k pak FULL OF DATA for £29.95, how come we have to pay £79.95 for a BLANK 64k DATAPAK???

I have had quite a few suggestions for improvements to the Beginners Page VAT programs, so in order to check these out and incorporate the best ones I am delaying publication of the final procedures until I have had chance to check them out thoroughly.

CONTENTS

31 Editorial

34-40 Programs & Procedures

32-33 Hints & Tips

HINTS & TIPS

WARNING

I was a bit premature in my praise of the SERIAL TO PARALLEL CONVERTER (which I bought from Eidersoft). I have since discovered that this little box is not all it is supposed to be. For instance, it will only work properly with a limited range of printers. With many others, including the EPSON range, there are problems ranging from random chars missing to the last character on each line being printed in italics. My previous comments on the converter were based on using it with a BROTHER HR1 DAISY-WHEEL printer. With this printer it behaves quite well UNTIL THE PRINTER BUFFER (4k) IS FULL, then the funny business starts, as above. I approached Eidersoft and the manufacturers (Miracle Systems) and both admitted that they were aware of problems, and, in all fairness offered to refund my money if the adapter didn't work with my printer(s). They do not keep a list of printers with which the adapter works without any problems, but below is my (certainly incomplete) list of printers WITH WHICH IT WILL NOT WORK.

EPSON FX SERIES
EPSON RX SERIES
CANON/KAGA TAXAN PW SERIES
AMSTRAD DMP 2000
MANNESMANN-TALLY

It works perfectly with the TANDY CGP 115!

KEY CLICK

Whilst the Handbook tells you how to switch off the Organisers BEEP (including KEY CLICK) it does not tell you how to ALTER the length of the click, in case, for instance, you are hard of hearing and want a more definite sign that a key has been pressed. The following "one-liner" allows you to lengthen the duration of the click - the default is 1.

CLIK:
POKEB \$20C0,d (where d is the required length in 1/20
secs)

Try lengths of 5 or 10 to hear which you prefer.

NEW OPL "WORDS"

Here is an excellent idea from M.W. NASH. He suggests that we start a library of new OPL words which would then become standard throughout the group and hopes to supply a selection of suitable words in the near future. I will start the ball rolling with one of mine, which I use regularly. First a short explanation. As most of you will know if you declare numeric variables in your procedures, then they are "zeroed" when the procedure is run. This may be unfortunate if you wish to retain these values for future runs. One answer is to use the TEN CALCULATOR MEMORIES which retain their contents until consciously altered. Of course you must be careful not to keep incrementing the contents of these memories unless you mean to do so - hence my little "word":

MZERO:
M0=0
M1=0
M2=0
M3=0
M4=0
M5=0
M6=0
M7=0
M8=0
M9=0
BEEP 200,800

Now let me have YOUR NEW OPL WORDS for the next I.F.

COMPETITION

After the overwhelming response to my little WESTMINSTER CHIMES Competition - see I.F. No 1 - (to date not a single entry) here is another little mind-stretcher. The competition is for the best ONE-LINE PROGRAM for the Organiser. I will be the judge, and there will be a small prize for the winner. The subject is quite open, but the program must be just ONE LINE (not including the procedure title).

COMBINING PROCEDURES

Here's another tip from M.W. NASH. He has found that it is possible to link printer control procedures (for his CGP 115) with a plus sign, Thus GRAPH:+BLUE:+CIR%:(x,y,r)+TEXT: in CALC MODE will invoke the GRAPHIC MODE, SELECT A BLUE PEN, DRAW A CIRCLE OF RADIUS 'r' AT 'x','y'. FINALLY RETURNING TO TEXT MODE. In case this is SERBO-CROAT to any member, the CGP 115 is TANDY's great little colour graphic plotter/printer. At the new price of £49.95 it's well worth having - I have just bought one! The Organiser coupled to this little machine is capable of some dazzling graphics printouts, not to mention the 65 TEXT SIZES. By the way, one or two members have had problems interfacing the Organiser and CGP 115. There are both PARALLEL and SERIAL ports on the CGP 115. As you will have noted above, the PARALLEL port may be used with the Eidersoft/Miracle Converter (and the PSION RS 232 cable, of course). The serial port works quite well if you make yourself a little extension-converter from a 4 PIN DIN PLUG (for the printer end) and a 25 pin male plug (to fit the RS 232 cable end). These should be connected as follows, using 3 core cable:

DIN PLUG		25 PIN PLUG
Pin 1	not used	
Pin 2	to	Pin 4
Pin 3	to	Pin 6
Pin 4	to	Pin 3

SETUP is as follows: BAUD 600, PARITY NONE, 8 DATA, 2 STOP, RTS/CTS

Finally, M.W. Nash has specifically asked me to publish his name, address, etc. as he is anxious to contact as many users as possible. He has also offered to FORMAT DATAPAKS free of charge to any member who sends his pack with return postage:

M.W. Nash, 6 Hazlemere Court, 26 Palace Road, London SW2 3NH
Phones: Home 01 671 8644 Business 01 584 5000 ext3537

PROGRAMS & PROCEDURES

I make no apology for this months P & P which is given over ENTIRELY to a real labour of love by STEVE J.T. KNIGHT. It is an excellent version of PONTOON. I let STEVE explain below.

In the previous issue, you asked for some games. I'm not very good at inventing games, but thought that perhaps PONTOON would be of interest. So I spent several happy evenings programming it. I enclose the result, which is rather long (you said it!! Ed.). I have included some notes about most of the procedures, to try to clarify their purpose etc.

Writing the basic code - dealing hands, calculating scores and so forth - was of course fairly easy. What has taken most time is trying to make thabanker be a bit smart. Of course, he cannot see the player's hand, except for the cards which are twisted. However, this banker has a very good memory, so he knows which cards have been dealt, and so, by inference, which have not yet been dealt. He also keeps an eye on how you are betting - if you suddenly increase your bet, he will try a little harder to beat you. He also works out what sort of scores you stick on, and takes that into account when trying to guess yours hand.

There is much scope for improvement and additions - for instance you cannot split a pair (it would not be too hard to put this in, really). Also, the interface between player and machine is not very smart. I leave this to others to improve if they wish.

To play the game, run PONTOON. It will deal one card and ask for a stake. You have £1500 credit. You cannot stake more than £500 on any one bet. So you enter something between 0 and 500. It will then deal you another card, and prompt you for a bet, stick or twist. Enter the appropriate initial letter. The banker is a bit hard on you, though. If you enter ANYTHING else, it will twist you a card. If you get 21 or you are bust, the banker will look at his own cards, and play as he sees fit. When satisfied, he will stick, and inform you of the outcome. Press EXE and you see what your present balance is. Press EXE again and the game continues. If instead you press Q you leave the game. It's quite fun, and the banker can surprise you with his skill!!

The main program - the one you call. This is straight forward. There are, however, a few lines at the end to print out the variable SPLT. The purpose of this should become clear in the procedure FACTOR% - briefly, it is the figure that the program has las calculated for use in the procedure which gives the mode (or is it median) value of the remaining cards. Reintroducing this in the procedure MODE allows the program to 'learn' over many games.

```

PONTOON:
GLOBAL deck$(52,1)
GLOBAL bankscr%,puntscr%,c%,crd%,ace%,kscr%,bet%,stake%,
lstake%,hide%,buys%
GLOBAL cap%,twists%,kace%,crds%,avbet%,ndls%,totbet,d%,
ts%,as%,ns%
GLOBAL totcor,cor,dls,totpr,avpr,npr%,frst,nn%,splt,err%
cap%=1500
as%=20
DO
KSTAT 3
RESET:
puntscr%=PHAND%:
AT 1,2
PRINT"
bankscr%=BHAND%:
ERRCOR:
SCORE:
GET
CLS
AT 1,2
PRINT"Capital: ";CHR$(6);cap%
KSTAT 1
UNTIL GET$="Q"
CLS
PRINT SPLT
GET

```

A bit long-winded this one, I'm afraid. It is the procedure which deals with the punter's (sorry player's) hand. No splitting of pairs, I'm afraid, but it does nearly everything else. (except checking that you are not playing on credit! Ed.)

```

PHAND%:
LOCAL sev%,nat%,fcrd%,dec$(1),scr%,fac%
KSTAT 1
fac%=FACTOR%:(MODE:)
d%=1
crd%=0
scr%=0
lstake%=500
c%=1
CLS
hide%=1
DO
IF c%=2
AT 1,2
PRINT"Stake: ";CHR$(6)
AT 9,2
STAKE:
AT 1,2
PRINT"
ELSEIF c%>2
AT 1,2
PRINT"Twist,Stick,Buy"
KSTAT 1
dec$=GET$
AT 1,2
PRINT"

```

The next procedure is to try to calculate what cards you are getting. It looks at the remaining cards (remember, it can deduce these because it knows what cards have been dealt so far), and looks for that value of card which has as many higher-scoring cards as there are lower-scoring cards. The statisticians amongst you will surely come up with a better solution!! However, this works quite well. I have tried several solutions to this problem - including simply working out the mean (average) card value. In truth, either method works fairly well, but this should have the edge over a longer period of time.

```

FACTORX:(R)
LOCAL fac%,cn%,v%,n%,nums%(10),t
cn%=1
v%=0
n%=0
DO
IF deck$(cn%)=""
fac%=(cn%-1)/4+1
IF fac%>10
fac%=10
ENDIF
nums%(fac%)=nums%(fac%)+1
v%=v%+1
ENDIF
cn%=cn%+1
UNTIL cn%>52
cn%=1
t=10
n%=0
DO
n%=n%+nums%(cn%)
cn%=cn%+1
IF n%>0
t=FLT(v%)FLT(n%)
ENDIF
UNTIL cn%>10 OR t<=r
RETURN cn%

```

A small program to calculate what value to use in the procedure which works out that value of card at which there are as many higher-scoring cards as there are lower-scoring cards. The performance of predicting the value of the next cards is monitored, and this procedure changes the value of SPLT to try to optimise the calculation. If you want a simpler solution, omit this procedure, and when FACTORX is called, use something like 2.35 instead of MODE:

```

mode:
IF puntscr%>0 AND puntscr%<22
splt=splt-(frst/1000)
ENDIF
IF splt<1.0 OR splt>3.0
splt=2.35
ENDIF
RETURN splt

```

The next procedure is used by BHAND% (the banker's hand) to try to estimate the chances that the next card will bust the hand. You may have ideas on how to improve it!!

```
bust%:(trgt%)
LOCAL bst%,nbst%,cn%,s%
WHILE cn%<52
IF deck$(cn%+1)=""
s%=cn%/4+1
IF s%>10
s%=10
ENDIF
IF s%>trgt%
bst%=bst%+1
ELSE
nbst%=nbst%+1
ENDIF
ENDIF
cn%=cn%+1
ENDWH
IF bst%=0
RETURN 1
ELSEIF nbst%=0
RETURN -10
ENDIF
IF (FLT(bst%)/FL(nbst%))>2.5
RETURN -1
ENDIF
RETURN
```

This procedure is to get in your stake, check it against the last stake you made (you cannot increase your stake over the last one you made in this hand)

```
STAKE:
INPUT stake%
stake%=IABS(stake%)
IF stake%>0 AND stake%>=lstake%
stake%=lstake%
AT 8,2
PRINT stake%
PAUSE(5)
ELSE
lstake%=stake%
ENDIF
bet%=bet%+stake%
cap%=cap%-stake%
```

A tiny procedure to make sure you don't go bust when you don't need to!

```
ACES%:(adsc%)
WHILE (c%=4 OR PEEKB(adsc%)>21) AND ace%>0
POKEB adsc%,PEEKB(adsc%)-10
ace%=ace%-1
ENDWH
```

This sets various things to zero and shuffles the cards if there are insufficient left

```
RESET:
ace%=0
hide%=0
kscr%=0
stake%=0
lstake%=0
bet%=0
twists%=0
c%=1
IF 52-crds%<11
AT 16,2
PRINT"*"
SHUFFLE:
ENDIF
IF bankscr%=40 OR puntscr%=40
CLS
PAUSE(10)
PRINT"Shuffling..."
crds%=0
SHUFFLE:
PAUSE(15)
CLS
ENDIF
buys%=0
kace%=0
```

A procedure to give us a card. As we cannot return more than one value, nor change the values of the actual parameters, we have to pass addresses and POKE them.

```
DEAL:(FACE%,SUIT%)
LOCAL r%,f%,s%
ST::
r%=RAND%:(0,51)
IF deck$(r%+1)<>"1"
deck$(r%+1)="1"
f%=(r%/4)+1
s%=((f%*4)-r%)
s%=ASC(CHR$(s%))
POKEB SUIT%,s%
POKEB FACE%,f%
RETURN
ENDIF
GOTO ST::
```

Again, a simple program to clear out the array which tells us which cards have been dealt.

```
SHUFFLE:
LOCAL cn%
crds%=0
cn%=1
DO
```

```

deck$(cn%)=""
cn%=cn%+1
UNTIL cn%>52

```

This one give us the actual value of the card, and prints the card and suit to the screen, in the position indicated by p% and l%

```

PP
THAND%:(p%,l%)
LOCAL f%,s$(1),f$(2)
DEAL:(ADDR(f%)+1,ADDR(s$))
s$=CHR$(PEEK(ADDR(s$)))
IF F%>10
f$=MID$("JQK",f%-10,1)
ELSEIF f%=1
f$="A"
ELSEIF f%=10
f$="10"
ELSE
f$=CHR$(f%+$30)
ENDIF
at p%,l%
PRINT F$+S$
RETURN f%

```

.....AND THERE'S MORE.

If you have keyed in the previous 6 pages, you deserve a rest, so the remaining procedures will be published in the next newsletter.

A COUPLE OF BLUNDERS FROM PREVIOUS LISTINGS

TIME (page 10)

Lines 10,11 and 12 have the wrong number of parentheses.

They should read:

```

POKEB $20CA,VAL(RIGHT$(st$,2))+VAL(RIGHT$(DATIM$,2))
POKEB $20C9,VAL(MID$(st$,4,2))+VAL(MID$(DATIM$,20,2))
POKEB $20C8,VAL(LEFT$(st$,2))+VAL(MID$(DATIM$,17,2))

```

It's all about counting the PAIRS of parentheses (as most of you already know).

PRDIR (page 18)

Line 24 should be:

```

ELSE LPRINT a$;PEEK(base%+6)-1;" minutes early"

```

IPSO FACTO

NEWSLETTER OF THE INDEPENDENT PSION ORGANISER USER GROUP

VOLUME 1 * * * NUMBER 5 * * * JULY 1987

Editorial Address:

130 Stapleford Lane, Beeston, Nottingham, NG9 6GB

Editorial

The press releases of the new Psion products finally arrived, with all the information on the POCKET SPREADSHEET and COMMS LINK (the latter replacing the old RS 232 cable). There was not quite the same amount of information (in fact NONE) about the new model XP (yes the name is the same!) which has 32k of RAM (22.5k available to the user and the rest reserved as a communications buffer - a great improvement over the old system) The expanded XP model costs the same as the old model - it is not clear at the moment whether Psion will be offering an upgrade service for owners of the old XP. I have been using both the COMMS LINK and the POCKET SPREADSHEET for some weeks now and will review them in a later issue of the newsletter. In the meantime let us hope that the SPREADSHEET will do for the Organiser what the original VISICALC did for APPLE COMPUTERS!

Now a DIRE WARNING. I have heard a rumour that Psion have decided to cut the WRITE line on new issues of the Spell Checker, so that it will be FATAL TO FORMAT THE PAK WITH THE OBJECT OF USING IT AS A 64k DATAPAK. Thats the BAD NEWS - the GOOD NEWS is (so I hear) that Psion are reducing the price of the 128k and 64k datapaks, and possibly the others as well!

CONTENTS

- | | |
|----------------------------|------------------------------|
| 41. Editorial | 42. Organiser Communications |
| 43-46 Text Editor | 47-50 Pontoon |
| 50. The Organiser & Speech | |

Organiser Communications by Peter Johnson

One of the things which attracted me to the Organiser was the possibility of linking it, via the relevant interfaces, to my AMSTRAD PCW; the cable supplied with the Psion's RS232 link is suitable for this and requires no modification. I first tried downloading my address/telephone list and achieved a measure of success by this was not maintained. The following notes are based on my experiences and may be of use to those who'd like to broaden their Organiser's world. Whilst I use a PCW, the principles involved should be the same with any micro; it will be seen that I'm not an expert and I'd be grateful for any help in filling the gaps.

Each computer requires setting up before data can be transmitted. On the AMSTRAD, DEVICE, SETSIO and PIP are required, being available on the system discs supplied with the machine. DEVICE tells it to address the serial port instead of the printer - type DEVICE LST:=SIO <CR>; The DEVICE settings are then shown. For reasons I've yet to discover AUXOUT: and LST: sometimes set themselves to NULL, so if your're having trouble getting through when everything else appears to be satisfactory, check DEVICE. SETSIO sets up the communications parameters - type SETSIO <CR> to see what the current setting are. The following are required for downloading to the Organiser: R9600,T9600,B8,S1,P none,Xon on,H off,I off. To implement any changes enter the relevant keys on one line after SETSIO, e.g. SETSIO B8, XON ON <CR>.

Accessed via the COMMS menu, which becomes available after connecting the RS232 link to the Organiser, SETUP is used to match the parameters of the two machines. On completion, enter EXE followed by RECEIVE, select FILE, enter EXE, accept or change the MAIN filename and then EXE to see BUSY. Now return to the AMSTRAD. Type PIP LST:=filename.ABC and await results. If you've got it right the A> reappears on the AMSTRAD. I've not yet succeeded in getting any indication from the Organiser that it has finished but after pressing CLEAR/ON found my address list happily residing in MAIN as expected.

Hereafter life got difficult. Having got my data to the Organiser, I tried to send it back again. By slowing down the BAUD RATE, I managed to get it on the PCW's screen en clair, but try as I might I couldn't save it to disc. I only got the AMSTRAD out of terminal mode by switching it off, too, whether I used MAIL232, the AMSTRAD COMMS UTILITY, or the Public Domain MEX and UKM7; the instructions for these assume a certain amount of experience so the tyro might be missing something obvious.

Thinking that typing was easier and quicker on the PCW I also tried downloading an Organiser program but with an equal lack of success, and it was an ASCII file and I did remember to tell the Organiser to expect a PROC and not a FILE!

To finish with, an oddity. When the AMSTRAD's serial interface is live, after having set DEVICE and the RS232 LINK is plugged in at both ends, it is impossible to switch off the Organiser - the display blanks but re-appears straight away.

TEXT - A Simple Text-Handler by Mike O'Regan

I have called this program a "text-handler" as it cannot really be called a word-processor, lacking many of the features we have come to expect from other programs for writing text. However, it does work and I have been using it regularly for writing routine correspondence, especially on the bus when I am commuting.

TEXT uses a database format, which means that it has some advantages and disadvantages. The main advantage is that you can write a document in several sessions and your text is secure as data. It also means that you can read back and print as many times as you wish without losing any text. You may retain as separate files any regularly used passages (such as your address, telephone number etc) and include these in your text as you require. The main disadvantage is that editing is restricted to a simple delete of the last line entered, as any attempt to use the Organiser's EDIT and/or UPDATE functions would mean that your line numbering would no longer apply (as EDITed records are added to the end of the file).

TEXT starts off in the usual way for a database, with the CREATE or OPEN options. The difference is that, if you have already CREATED a file with the name you choose, then the file is OPENed, otherwise a new file is CREATED and OPENed. If you are writing a new file, then you are prompted for a line-length (for your PRINTER). The main menu allows the options WRITE, READ, ERASE, PRINT and QUIT.

WRITE displays your line status on the top line of the Organiser's display in the form of a CAPS message (when in CAPS mode) and a line number and character countdown. As you enter your text, on Line 2 of the Organiser, the counter decrements until you have only ten characters of your chosen line length remaining, then it sounds a "bell" on each further character insertion to remind you to press EXE (line feed) before the line is full. You may use the DEL key to correct any mistakes BEFORE you press EXE or ERASE (from the menu if you wish to delete the whole line after pressing EXE). I have also re-defined some of the lesser used characters to allow some commonly used characters which are not available from the keyboard, as follows:

```
/ will give you ?  
* will give you !  
+ will give you &  
> will give you ' (in deference to my surname!)  
$ will give you a "hash" sign which prints as £
```

READ may be used at any time to readback and check the text entered so far.

ERASE deletes the LAST LINE entered.

PRINT has a sub-menu allowing PRINT, SETUP or QUIT. SETUP calls EPSON (published earlier) to allow you to select print-style, page length, etc. PRINT of course prints your complete text, provided your printer is attached and enabled.

QUIT closes the file and takes you back to the Main Menu.

```

text:
GLOBAL fn$(10),dev$(1),l%,sta$(4)
LOCAL o%
o%=MENU("Open,Quit")
IF o%=1
PRINT "Filename:"
INPUT fn$
PRINT "Device:"
INPUT dev$
fn$=dev$+"."+fn$
IF EXIST(fn$)
OPEN fn$,a,line$
l%=m9
GOTO menu::
ELSE
CREATE fn$,a,line$
ENDIF
PRINT "Line length"
INPUT l% :m9=l%
ELSEIF o%=2
STOP
ENDIF
menu::
o%=MENU("Write,Read,Erase,Print,Quit")
IF o%=1
twrite:
ELSEIF o%=2
tread:
ELSEIF o%=3
tera:
ELSEIF o%=4
tprint:
ELSEIF o%=5
CLOSE
RETURN
ENDIF
goto menu::

```

```

tread:
FIRST
DO
VIEW(2,FIX$(POS,0,3)+ " : <"+a.line$+"> ")
IF GET=1
RETURN
ENDIF
NEXT
UNTIL EOF
RETURN

```

```

twrite:
GLOBAL f$(1),rl%,temp$(150),ct%
CURSOR ON
KSTAT 1
DO
temp$=""
rl%=1%
ct%=0
tshow:
DO
f$=GET$
IF ASC(f$)=42
f$=CHR$(33)
ELSEIF ASC(f$)=43
f$=CHR$(38)
ELSEIF ASC(f$)=36
f$=CHR$(35)
ELSEIF ASC(f$)=47
f$=CHR$(63)
ELSEIF ASC(f$)=62
f$=CHR$(39)
ENDIF
IF ASC(f$)>31
rl%=rl%-1 :ct%=ct%+1
temp%=temp%+f$
IF LEN(temp%)>(1%-10)
BEEP 200,800
ENDIF
tshow:
ELSEIF ASC(f$)=1
RETURN
ELSEIF ASC(f$)=8
temp%=LEFT$(temp%,ct%-1)
ct%=ct%-1 :rl%=rl%+1
tshow:
ENDIF
UNTIL ASC(f$)=13
BEEP 200,200
a.line$=temp$
APPEND
UNTIL COUNT=200

```

```

tprint:
LOCAL o%
o%=MENU("Print,Setup,Quit")
IF o%=1
FIRST
DO
LPRINT a.line$
NEXT
UNTIL EOF
ELSEIF o%=2
epson:
ELSEIF o%=3
ENDIF
RETURN

```

```
tshow:
IF PEEKB($7B)=0 OR PEEKB($7B)=64
sta$="CAPS"
ELSE sta$="    "
ENDIF
CLS
AT 1,1 :PRINT sta$;
AT 10,1 :PRINT COUNT+1;":";rl%
AT 1,2 :PRINT RIGHT$(temp$,15)
```

```
tera:
LAST
ERASE
RETURN
```

Note that this program has been deliberately kept short, so that it is practical to run on the CM model. If anyone has any ideas to improve any of the procedures, feel free to write in with your suggestions.

PROGRAMS ON DATAPAK

After next month's newsletter, I am considering making all the programs published so far available on datapak, for the cost of the pack, plus a small handling charge. If anyone is interested in this offer, please write to me so that I can gauge the numbers involved.

....and NOW....the rest of the PONTOON procs.
N.B. the last proc in last months issue has collected a couple of phantom chars during the transfer process - delete the PP on the top line, and the T before HAND%: on the second line.

Used to work out who won, and by how much

```
SCORE:
IF bankscr%<puntschr%
PRINT "You win"
IF puntschr%>21
cap%=cap%+bet%*3
IF puntschr%=50
cap%=cap%+bet%
ENDIF
ELSEIF puntschr%>0
cap%=cap%+bet%*2
ENDIF
ELSEIF bankscr%>=puntschr%
PRINT "You lose"
IF bankscr%>21 AND bankscr%>puntschr%
cap%=cap%-bet%
ENDIF
ENDIF
PAUSE(15)
```

This does a few messy jobs - partly it helps to tell us if you have a natural pontoon i.e. with only two cards. It also works out the real value of the card

```
CRDCALC:(nflg%)
IF crd%=10
d%=d%+4
ELSE
d%=d%+3
ENDIF
IF crd%=1
IF c%<3
POKEB nflg%,PEEKB(nflg%)+10
ENDIF
crd%=11
ace%=ace%+1
ELSEIF crd%>=10
IF c%<3
POKEB nflg%,PEEKB(nflg%)+1
ENDIF
crd%=10
ENDIF
```

Very simple - just returns a random number between s% and e% (in our case 1-52)

```
RAND%:(S%,E%)
```

```

LOCAL r,n%
DO
r=RND
n%=INT(r*100)
UNTIL n%>=s% AND n%<=e%
RETURN n%

```

This procedure is the main line procedure for getting the Banker's hand. The conditions given for the UNTIL at the end of the proc. have been split over two lines to fit the paper width - enter the two lines as one

```

bhand%:
LOCAL bscr%,bnat%,q%
d%=1
crd%=0
ace%=0
ADJSCR:
c%=1
DO
RE::
crd%=hand%:(d%,2)
crds%=crds%+1
CRDCALC:(ADDR(bnat%)+1)
bscr%=bscr%+crd%
ACES%:(ADDR(bscr%)+1)
PAUSE(15)
IF puntscr%=40 AND bnat%<>11 AND c%>1
RETURN -1
ELSEIF bnat%=11
RETURN 40
ESEIF c%=5 AND bscr%<22
RETURN 30
ELSEIF bscr%>21
RETURN 0
ENDIF
c%=c%+1
IF puntscr%=0 AND c%>2
RETURN bscr%
ENDIF
q%=BUST%:(21-bscr%)
IF q%=1
GOTO RE::
ELSEIF q%=-10
RETURN bscr%
ENDIF
IF q% AND bscr%<kscr%-2 AND ace%>0
bscr%=bscr%-10
ace%=ace%-1
q%0
ENDIF
UNTIL (q% AND bscr%>14) OR c%>5 OR (bscr%>=kscr% AND c%>2)
OR (buys%>0 AND bscr%>19)
RETURN bscr%

```

This proc tries to take all the various bits of information available to the banker into account and tries to work out what is the value of your hand. You may notice there is some monitoring of its performance going on (using the proc ERRCOR)

to compensate for trends away from the actual scores you are getting. The longer you play, the better the banker gets at predicting your score!! (A sort of learning program)

```

adjscr:
IF puntscr%>0 AND puntscr%<22
frst=puntscr%-kscr%
ENDIF
IF as%<16
as%=16
ENDIF
IF kscr%<as%
kscr%=as%
ENDIF
ndls%=ndls%+1
IF bet%>(FLT(avbet%)+FLT(avbet%)*.3)
kscr%=kscr%+1
ENDIF
totbet=totbet+bet%
avbet%=INT(totbet/ndls%)
IF puntscr%=0
kscr%=0
ELSEIF kscr%>21
IF kace%>0
WHILE kace%>0 AND kscr%>21
kscr%=kscr%-10
kace%=kace%-1
ENDWH
ELSE
kscr%=21
ENDIF
ENDIF
IF c%=3
npr%=npr%+1
IF avpr<17
avpr=17
ENDIF
kscr%=INT(avpr)
totpr=totpr+puntscr%
avpr=totpr/npr%
ELSE
kscr%=kscr%+INT(cor)
ENDIF
I kscr%<16 AND kscr%<>0
kscr%=16
ELSEIF kscr%>21
kscr%=21
ENDIF
IF puntscr%<as% AND puntscr%>0
ns%=ns%+1
ts%=ts%+puntscr%
as%=ts%/ns%
ENDIF

```

This proc keeps an eye on the difference between average predicted score and average actual score and produces a corrective figure accordingly. Every so often, it forgets all it has learnt, just in case!


```

ERRCOR:
IF puntscr%>0 AND puntscr%<22
err%=puntscr%-kscr%
IF err%<>INT(cor)
dls=dls+1
totcor=totcor+(puntscr%-kscr%)
cor=totcor/dls
ENDIF
ENDIF
IF dls>100
PRINT"*****" :PAUSE(10)
dls=0
totcor=0
cor=0
ENDIF

```

The Organiser & Speech

You may recall me mentioning (on Page 20) that my blind friend, Les Ball was working on a speech system for the Organiser. Well Les has completed his work and the system is now being launched by Les and a local company which specialises in equipment for the visually handicapped. Here are a few notes on the system.

Hardware required.
Organiser II Model XP
Psion RS232 Cable
Dolphin "Mimic" Speech Synthesizer

The Mimic was chosen because of its portability and mains independence. It features a built-in rechargeable ni-cad pack which may be augmented by either another ni-cad pack or ordinary batteries. The system is capable of varying both the speed and pitch of the speech produced. It works in four main modes:

1. Speak text with some punctuation.
2. Speak text with ALL punctuation and symbols.
3. Speak EVERY LETTER and indicate whether upper or lower case.
4. Speak every letter, using PHONETIC ALPHABET, plus punctuation

The software, which Les has developed in machine code, looks at both lines of the Organiser's display, or each line separately (helpful when scrolling down lines of text, for instance). It will speak every key pressed, if required, and will work with ALL the Organiser's built-in features. It is also capable of speaking the status of the hardware, including the type and status of any datapaks fitted.

I can see this system as becoming an indispensable aid (because of its total portability and ease of use) to many blind people who, up to now, have either been tied to a desk-bound system, or have had to miss out on the joys of computing altogether.

Anyone requiring further details should contact:
Les Ball
17 Deepdale Road Tel: (0602 289553
Wollaton
Nottingham
NG8 2FU

IPSO FACTO

NEWSLETTER OF THE INDEPENDENT PSION ORGANISER USER GROUP

VOLUME 1 * * * NUMBER 6 * * * AUGUST 1987

Editorial Address: Mike O'Regan
130 Stapleford Lane, Beeston, Nottingham, NG9 6GB

Editorial

With effect from the next issue of IPSO FACTO the "I" in "IPSO" will be changed to stand for "INTERNATIONAL". This is not because we have renounced our INDEPENDENT status, but to signify our worldwide appeal. We now have members in GERMANY, IRELAND, SWITZERLAND, SOUTH AFRICA, HAWAII, and SARAWAK and I am expecting the first enquiries from the USA soon (after the successful launch of the Organiser in May).

You may have seen a copy of the quarterly Psion house magazine PSION NEWS by now. If you haven't and would like a copy contact Psion, who will be only too pleased to put you on their (free) distribution.

It may also have slipped your notice, but this is the LAST IPSO FACTO of the first series. What this means is that the SIX-MONTH TRIAL PERIOD is now up and it is decision time about whether to continue with the Group. You will probably have gathered from the first paragraph that IPSO is now well established and, with your continued support will carry on into the foreseeable future. I am now beginning to hear from more and more of you, and your letters of appreciation have encouraged me to carry on - even if the old 95-5 rule is still in operation 95% of the work being done by 5% of the members. I would like to take this opportunity to thank all those who have given me practical assistance by submitting material for publication (even if you are still waiting to see your bit in print). There have even been a few cases of people sending me editorial contributions with their membership enquiries.

I have decided to continue membership on a six-monthly basis, at least until the end of our first year (in Feb '88) so I will require you all to renew your six month subscription by filling in the enclosed Renewal Form, adding any further details which I may not have, and finally not forgetting your cheque or postal-order.

Thank you.

CONTENTS

- | | |
|-----------------------|----------------------------|
| 52. Feedback | 53. A Valuable Aid |
| 54. Graphics & Things | 56. Beginners Please |
| 58. Extra | 59. The Pocket Spreadsheet |

FEEDBACK

Some of these snippets have been rather delayed in their publication. I apologise to the authors concerned.

Psion Comms Link

Since my last article on using the Psion as a terminal to E-Mail services, Psion have introduced the Comms Link, an updated and much improved version of their RS232 link. Having recently acquired this link, I thought readers would be interested in the advantages of this when communicating via modem.

TERMINAL mode is visibly improved by the fact that the display now scrolls across until a carriage return is received. The only complaint I would have is that scrolling starts from the left hand side of the screen, and it is very easy to miss the first few letters of a line.

The COMMS menu now has a new function - CAPTURE. This is particularly useful, as it enables the user to capture incoming data, save it, edit it or retransmit it.

The other big improvement on the COMMS LINK is the inclusion of software for the IBM PC (or compatibles). When this is run, and the comms link plugged into the PC RS232, the user can create and open files on the PC, all from the Psion.

Psion are to be congratulated on a much improved product. At the end of my last article, I mentioned that I was hoping to get a small V21 battery or line powered modem for use with the Organiser. Unfortunately I have not had much success in tracking one down so far, so if anyone out there knows where I could obtain one, I would be interested in hearing from them.

Simon Webb Telecom Gold Mailbox 72:MAG80001

Printer Tip from Simon Moss

If you buy a Citizen 40 column printer from Farnell (175-238) for use with the Organiser, you will see from the manual that the data control is DTR (pin 20), which is not used on the Organiser. Don't despair - make up a male to male connector with RTS (pin 4) of the Organiser plug connected to DTR (pin 20) of the printer plug. Select RTS/CTS mode on the set-up procedure and you have a nice little printer for £160.

Random Jottings from John Crook

1. Has anyone any experience interfacing the Organiser to a SPECTRUM via the RS232 port on Interface I?

2. In "Using & Programming the P.O.II on page 120 the example program uses "VAL" as a variable. This is unacceptable as a valid name. In the same book the game TARGET is quite good. However, I saved it to a datapak and it "died". It is so slow that it's not worth playing. Why does the speed drop so much? (Editor's note - John solved this one for himself by writing the following routine to call the TARGET game:

```
TRGT: (name of proc)
TARGET: (calls TARGET game)
```

This has the desired effect, but does anyone know WHY???)

The Organiser - a valuable aid by Toni Fine

I am an occupational therapist at the Royal National Orthopaedic Hospital in Stanmore, Middlesex. Part of my job is to develop the use of computers for assessment and treatment of a wide variety of patients in various settings. The majority of my experience is with Acorn and Amstrad micros, but I have been an owner (though fairly indifferent user) of the original Organiser for quite some time. I found its database facilities particularly useful for quick portable retrieval of information but never got involved in the programming side. All of this changed when I met "Bruce" and this short article is to introduce you to some of the less obvious ways in which the Organiser II is providing useful and functional applications. Apologies in advance to all you experienced programmers - our efforts may seem very naive to you but the end result is a useful program and that, after all, is what we are all aiming for.

Bruce is a 32 year old American photographer who broke his neck in an accident 6 months ago. The level of his spinal cord injury means that he has no lower limb movement and only very weak and limited upper limb, e.g. arm movement and no functional finger movement. Nonetheless he is a dynamic, fascinating man who recently bought himself an Organiser II and asked me to help him with it. It certainly made my Model I look and act like an antique, and after just one session on his, I went out to replace mine!!

The Organiser II is ideal for Bruce for many reasons and we discover more each day. However the initial one was that even with his weak arm movements he could exert sufficient pressure to depress the keys and of course the fact of it being a "mini" keyboard is very useful for him as he has far less distance to have to reach. BUT, how to open the case?? Since Bruce has no functional "grip" in either hand, he cannot pull the cover open and it is totally contrary to his need for independence to have to ask for someone else to open it for him. We did manage to solve this problem by making a rather elegant black leather tight-fitting carry case for the Organiser to sit in and which uses an adjustable length strap from a redundant camera case. Now Bruce wears his Organiser around his neck - this arrangement lets him stabilise it so he can use gross movements to pull open the case and, because it is around his neck, he need not worry about dropping it.

By now you must be thinking, "OK, but what does he DO with it?" At present he's just getting used to it and is entering various names and addresses. However, one more unusual application for the alarms is for him to remind himself at various times of day to do some of the personal care things that paraplegics must do very regularly to ensure good health. And we've just written our first program. As you'll remember, I said Bruce was an American. Thus, when the nurses take his temperature and tell him it's 38 it means nothing to him. So - you guessed it - we wrote a tiny procedure to convert Celsius to Fahrenheit. It was a very useful teaching/learning exercise at the end of it, he has a program that is really useful. I left him today busily starting on a proc to convert feet & inches (useful for him as a photographer to understand distances) and I think there will be no stopping him now. It's very early days for both of us, but watch this space for future developments. Any suggestions or ideas would be gratefully received.

(Editors Note: This was written by our ONLY LADY MEMBER. Any tips you may wish to pass on may be channelled through me)

GRAPHICS & THINGS

You have probably noticed that the Organiser allows a small facility for USER DEFINED GRAPHICS. The first 8 characters in the ASCII set (CHR\$(0) to CHR\$(8), can be defined by the user by POKEing numbers into that part of memory which is reserved for these character definitions. It would have been nice if ALL characters could have been redefined, (perhaps this is possible?) and also if the Organiser display had been continuous instead of split up into 32 8 x 5 blocks. However, within these limitations, the user defined graphics (UDGs for short) can be quite handy. For instance, it is possible to define a set of SUITS (Hearts,Diamonds,Clubs,Spades) for Simon Webb's PONTOON game published in the last two issues. I will show you how on the next page. You may use any of the profusion of UDG generator programs (even the one in the Organiser Handbook). However, with a bit of editing, it is possible to play about with UDGs until they are right. With this in mind I have added a few more "bells & whistles" to the UDG procedure as follows:

1. Allows all EIGHT ROWS to be defined
2. Lets you view all current definitions.

By the way, the handbook still says that UDGs must be re-defined if the Organiser is switched off. This is only true of older machines. The newer Operating Systems retain UDGs in memory until you either re-define them or do a HARD RESET.

The method of defining UDGs on the Organiser is similar to all other BIT-MAPPED GRAPHICS on other computers. Each of the EIGHT BITS (each representing a horizontal row of FIVE DOTS) is defined by allocating it a number between 0 and 31. If you look at the accompanying diagrams you will see the relevance of these numbers - a whole row of FIVE DOTS switched on has the value of 31 (1+2+4+8+16). You will see that each number between 0 and 31 represents a UNIQUE PATTERN. In theory the number of DIFFERENT PATTERNS possible with an 8 x 5 block is 2^{40} (which my Organiser tells me is 1,099,511,627,770 - enough for most purposes!!).

Using UDGs even a measure of simple ANIMATION is possible on the Organiser screen, by quickly alternating two or more UDGs at the same screen location ("PHONE" - reviewed in a previous issue - does this to good effect). A more unusual application is suggested by ALAN WADDINGTON who has discovered that UDGs may be used to provide an IKON MENU. The only limitation to this is that menu choices can then only be made by moving the CURSOR over the item required, as obviously UDGs cannot be accessed from the keyboard. Apart from looking unusual, I cannot think of a practical application for an Ikon driven menu. Can you?

Edited UDG Routine

```
MUDG:
LOCAL char%,count%,byte%(8)
start::
PRINT "0 1 2 3 4 5 6 7"
PRINT CHR$(0),CHR$(1),CHR$(2),CHR$(3),CHR$(4),CHR$(5),CHR$(6),
CHR$(7)
GET
PRINT "Char.No.(0-7)",
INPUT char%
```

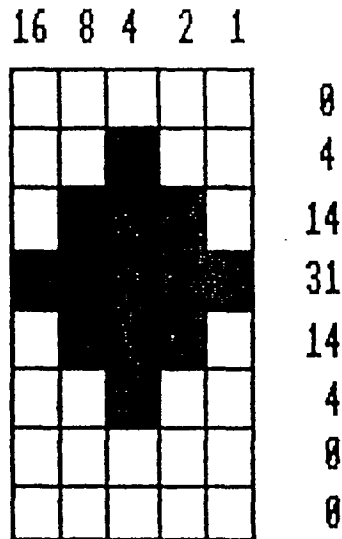
```

IF char%>7
RETURN
ENDIF
count%=1
DO
PRINT "Line",count%
INPUT byte%(count%)
count%=count%+1
UNTIL count%=9
POKEB $180,64+(char% AND 7)*8
count%=1
DO
POKEB $181,byte%(count%)
count%=count%+1
UNTIL count%=9
CLS
GOTO start::

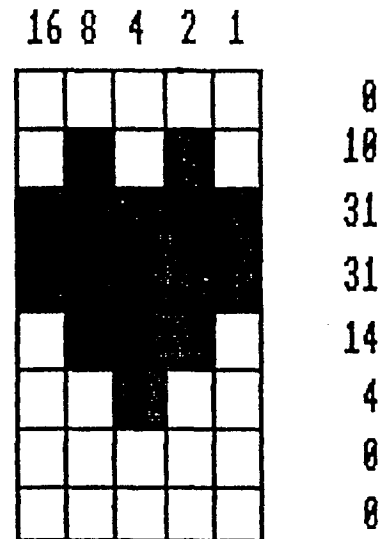
```

Now try defining DIAMONDS,HEARTS,CLUBS and SPADES as CHAR\$(1) to CHAR\$(4) and have a go at your own £ sign (make it CHAR\$(6)) for use with the PONTOON GAME.

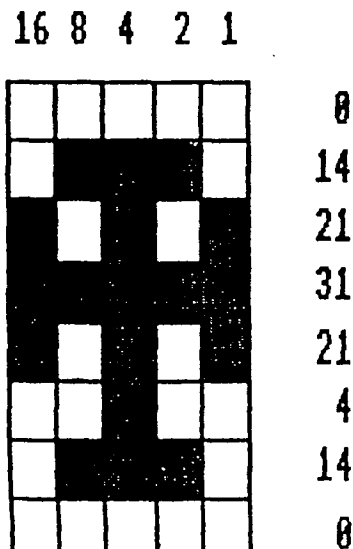
DIAMONDS



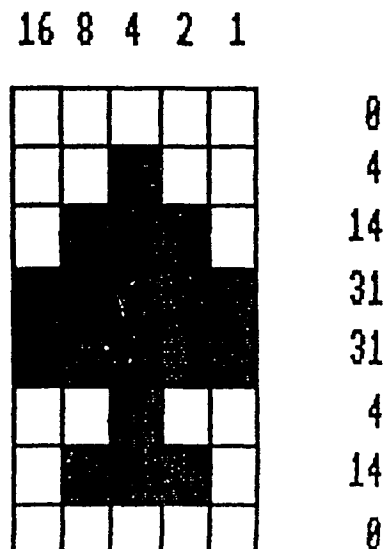
HEARTS



CLUBS



SPADES



BEGINNERS PLEASE

This month I am printing a game called HILO. It is the old number guessing game, where the Organiser selects a RANDOM NUMBER between 1 and 999. You are then invited to guess what the number is and the Organiser will give you "clues" by adjusting the range of choice until you home in on the correct number, when it will tell you how many guesses you took and compare this to the running average.

However, the object of the exercise is not just to key-in the listings, but to study them with these accompanying notes as examples of good programming (I think), which serve to illustrate quite a few features of OPL.

The first point to notice is that the MODULAR nature of the procedures makes the whole thing easy to understand and DEBUG and MODIFY, if necessary. Notice that the main procedure, "hilo", does little more than declare the VARIABLES and keep and show the score. I have given all VARIABLES meaningful names for better understanding. All the NUMERIC VARIABLES are INTEGERS (whole numbers) as signified by the % sign, and are GLOBAL which means that they can be used freely by the procedures which follow. However, as the "yesno" STRING VARIABLE is only required by "hilo" it can be LOCAL. Line 5 of "hilo" may need a little explanation. It is the line where the Organiser selects its "secret" number. As the OPL instruction RND will only produce a (pseudo) RANDOM NUMBER between 0 and 1, we need to "tailor" this to produce a number in the range which suits our game. In this case we require an INTEGER between 1 and 999. The first part of the formula will give us a number between 0 and 998 so the final "1" is added to ensure that the number is in the required range. Lines 13-15 are examples of using the CALCULATOR MEMORIES to good advantage. Note that these DO NOT NEED TO BE DECLARED AS VARIABLES. The reason for using CALC memories is to enable the running average score to be kept quite independent of the game procedures. If I had used ordinary INTEGER VARIABLES these would have been automatically zeroed every time the game was played. The line which just says GET is a way of letting you read the score before moving on. Pressing ANY KEY will let the procedure continue. Notice also (two lines further on) that, if the required input is a single CHARACTER then the string version of GET is all that is required. The UPPER bit is a way of ensuring that it doesn't matter whether "yesno" STRING is UPPER CASE or not. The last point to notice in "hilo" is that it is possible for an OPL procedure to CALL ITSELF, hence the last line.

The single line proc "hdisp" just displays a more or less centred top line of information throughout the game. In "hguess" notice lines 4-10 which remind you quite forcibly not to waste guesses by keying-in a number outside the range. Press EXE (which is the ASCII number 13) to stop the din, save your batteries and carry on with the game. "hhi" and "hlo" adjust the upper and lower limits and make an appropriate BEEP (either high or low) to give you an audible indication of where to guess next.

With a little ingenuity, it should be possible to convert HILO from a NUMBER game to a SOUND game - "GUESS THE PITCH" Why not have a go and learn a bit more in the process!

```

hilo:
GLOBAL x%,guess%,high%,low%,count%
LOCAL yesno$(1)
count%=0
x%=INT(RND*999)+1
low%=1
high%=999
DO
hdisp:
hguess:
UNTIL guess%=x%
BEEP 200,200
m0=m0+count%
m1=m1+1
m2=INT(m0/m1)
PRINT "You took",count%
PRINT "Average",M2
GET
PRINT "Another (Y)",
IF UPPER$(GET$)<>"Y"
RETURN
ENDIF
CLS
hilo:

hdisp:
AT 5,1 : PRINT FIX$(low%,0,-3),"-",FIX$(high%,0,-3)

hguess:
INPUT guess%
count%=count%+1
IF guess%<low% OR guess%>high%
PRINT "Wasted guess"
DO
BEEP 200,50
BEEP 200,400
UNTIL GET=13
RETURN
ENDIF
IF guess%>x%
hhi:
ELSEIF guess%<x%
hlo:
ENDIF

hhi:
high%=guess%
BEEP 200,50

hlo:
low%=guess%
BEEP 200,400

```


BEGINNERS PLEASE EXTRA

Numbering Systems

I once heard of a tribe in Africa where most of the tribesmen, by some odd quirk of heredity, had FIVE FINGERS AND A THUMB on each hand. After wondering for a time if they could have used their peculiarity to become accomplished pianists (which is doubtful), I then thought of the implications on our common numbering systems which would have resulted if the whole human race were similarly endowed. That we would have developed as standard a DUODECIMAL (base 12) system is not as preposterous as it at first appears. Think of our old "imperial" system of weights and measures, with all the odd multiples that this entailed before we were dragged (struggling at times) into the wonderful world of the Systeme Internationale (SI) version of the METRIC SYSTEM, with its connections with DECIMAL calculations. There are some advantages to working to base 12. 12 will divide EXACTLY by 2,3,4 and 6, whereas 10 can only manage 2 and 5. There are still problems when numbers have to be divided by 3, especially on computers!

All this is leading up to the fact that we are now asked to become familiar with numbering systems which use bases other than 10. Computers (at least DIGITAL ones) like to work in BINARY (base 2), sometimes OCTAL (base 8), often in HEXADECIMAL (base 16). Notice that the last two are MULTIPLES of BINARY. The first digital computers were designed to work in DECIMAL and this was responsible for holding them back for a number of years. Digital computers are essentially a collection of switches, each of which can be in one of two states - ON or OFF. Thus a "two-state" numbering system lends itself ideally to this situation - 1 for ON, 0 for OFF. To simplify things quite a lot, if you couple eight switches in PARALLEL, you have the facility to ADDRESS this collection in 256 different ways (use CALC on your Organiser to check this - 2^8). If you look in the back of your Organiser Handbook, (page 194), you will see that the Organiser uses most of the 256 combinations which the 8 BIT PROCESSOR can recognise (although I can't think of any PRACTICAL applications of the Japanese characters, etc. in the last part of the table. However, as you will see in another part of this newsletter, there is a chance to use the first 8 (0 to 7) as USER DEFINED GRAPHICS. You will also notice that the numbers (which I have written in in DECIMAL alongside the UDG blocks for the playing card suits) can also be expressed in HEXADECIMAL. There is not much to be gained, however when the width of the character is restricted to only 5 pixels.

If we are to use HEXADECIMAL (which is quite common as a "shorthand" numbering system for computers) we are in trouble when trying to express numbers using only 0 to 9 of the DECIMAL system. As we don't have a SINGLE DIGIT to express the numbers from 10 to 15, someone had the bright idea of using the first six letters of the alphabet, so we can now count 0 1 2 3 4 5 6 7 8 9 A B C D E F 10 11 . . . etc.

(continued next month)

Up to now I have never even used a spreadsheet, but with the release of Psion's Pocket Spreadsheet for the Organiser, I decided it was about time I found out what they were all about. Admittedly, I had my doubts as to the use of a spreadsheet with so small a display, but had heard so many good reports about it that I took the plunge and purchased one. The manual is extremely good and takes the novice right through from basics using examples you enter into the sheet. The display does take some getting used to, but there are four display modes for showing different information which helps. The spreadsheet functions are very comprehensive, with all the standard mathematical and trigonometric functions, as well as statistical and financial. Probably one of the most powerful features is the ability to interface the spreadsheet to OPL, calling procedures from any cell, and grabbing cells from an OPL procedure. Once a spreadsheet is set up, it can be saved to RAM or datapak, or more interestingly, exported to a spreadsheet on a PC e.g. LOTUS 123. It is exported via the COMMS LINK with the file-server software running on the PC. Once the file is transferred across, it can be retrieved into LOTUS. One anomaly I did notice is that the manual states that the extension for a LOTUS 123 file is .WK1 which is correct. However, when exporting from the Psion, this is displayed as .WR1, and the file is indeed saved on the PC as a .WR1 file, which LOTUS won't read. It is a simple matter to rename the file extension to .WK1, but there is no mention of this in the manual. However, I did manage to get the file exported, and from there I was also able to import files from the PC, again by changing the file extension.

In general, I found the Pocket Spreadsheet to be a well thought out piece of software, which I am already finding useful applications for.

...and there's MORE (on the Pocket Spreadsheet)
from Mike O'Regan

Unlike Simon, I have had considerable experience with spreadsheets and use them practically every day. I found the Pocket Spreadsheet to be quite revolutionary in some ways whilst it has one or two glaring omissions (which could probably have been built in with a little more thought). The manual has quite a few niggling mistakes - for instance the Appendix about the OPL command is quite confusing and it took not a little experimenting to get it to work reasonably well. I find that having experience of spreadsheet operation is quite a help when trying to visualise anything more than the simplest arrangement on the Psion. The "locked" title lines (column A and row 1) are quite handy, but it sometimes takes a little working out on paper if the whole spreadsheet layout cannot be visualised. I know that it is possible to print out intermediate sheets as an aid to design, but it would be nice to be able to print out a sheet consisting of just formulae (which is one of the omissions mentioned above). Another missing feature is the ability to do a GLOBAL BORDER switchoff before printout, as it is disconcerting to have the column letters and row numbers on every printout. One point not covered in the handbook is how to draw horizontal lines to separate various sections of the spreadsheet, but this is soon worked out using = or - (with a leading double-quote of course, otherwise non-alpha entries are

taken as part of a formula and come up as errors). The vertical lines which are printed out as row separators are also "compulsory", and as the files produced by the Pocket Spreadsheet are binary, they cannot be accessed and handled like ordinary files, so the format of the printout cannot be hand-tailored. However, it shouldn't be long before someone cracks the code (a la Diary Printout). Stop press news is that I have just successfully used the Pocket Spreadsheet for doing a VAT return for a friend of mine, who's own computer was on the blink. He was quite impressed at the double-quick time to set up the spreadsheet, enter the figures, and printout everything for his accountant. He would have been even more impressed if the spreadsheet sported a TAB function to help the smooth transition from cell to cell when entering the data, instead of the inevitable cursor keys.

By the way, I made a futile attempt to get a spreadsheet to load directly into a VIEWSHEET screen on the BBC Master. It wouldn't work on IMPORT or EXPORT and I was finally resigned to PRINTing the spreadsheet into the word-processor, VIEW, which it did faultlessly, as usual. I find that it is often more convenient when passing text-files or databases between the Psion and a desk-top to do this in LPRINT format from the Psion. For instance, this lets you pass more information than just the records in a database, such as date, totals etc. and they can be transferred already formatted and ready for printing with the minimum of editing.

DON'T FORGET TO FILL IN YOUR MEMBERSHIP RENEWAL FORM WHICH IS ENCLOSED WITH THIS NEWSLETTER SO THAT YOU GET YOUR SEPTEMBER NEWSLETTER ON TIME!

IPSO FACTO

NEWSLETTER OF THE INTERNATIONAL PSION ORGANISER USER GROUP

VOLUME 1 * * * NUMBER 7 * * * SEPTEMBER 1987

Editorial Address: Mike O'Regan
130 Stapleford Lane, Beeston, Nottingham, NG9 6GB

CONTENTS

61	Editorial	62	Spread the Word etc
63	Genealogy by M. Last	65	Filenames
66	RAMPAKs & DATAPAKs	67	Feedback Galore
69	Postal Arrangements etc	70	Program Page

Editorial

This issue of IPSO FACTO marks the start of the second six-months of IPSO's existence. As I mentioned in the last issue, I am now getting quite a lot of response from members, either in the form of articles and programs or queries on various topics. The one query which stands out, in that it is by far the most frequent one, is the position on the Technical Reference Manual. Ever since Psion reluctantly admitted that they were working on such a publication, there have been rumours and counter-rumours of the document. Psion themselves have been hardly helpful when members have enquired on the subject. However, other members have been privileged to have "draft copies" if they could prove that they had were "serious commercial software developers". In order to help this rather unsatisfactory situation, I have managed to get a copy of the draft Technical Manual from Psion (in the form of IBM compatible discs). As I understand it, I have their tentative approval to release copies of these discs to members, so that they can print their own copies of the (enormous) manual, but I will have to get this approval in writing from Psion before I can do anything further. I will give more information on this subject in the next issue of I.F.

As this is the start of a new period, I have decided to print an enlarged FEEDBACK section, with many of the points which were raised either in letters or in comments on your Membership renewals. If I miss out any, then please draw my attention to the fact, bearing in mind that the answers to quite a number of enquiries have been answered in earlier issues of the newsletter.

For the benefit of the large number of new subscribers (since the excellent review of the Organiser and mention of IPSO in Personal Computer World) I should perhaps clarify the position with regard to membership. I am continuing to accept membership on a SIX-MONTHLY basis, which means that your £6

entitles you to six issues of this newsletter. If you joined during the months of July or August you should have received all SIX issues (March - August) and unless you have paid for the NEXT six (by paying £12 subscription) you will not be entitled to receive further copies. Some new members have already realised this and sent the higher subscription. Older members should have received (and returned) a membership renewal form and are required to pay the £6 subscription. At present a few older members have not yet renewed. I did consider sending these the next issue, assuming they are in the process of renewing, but decided to wait until they actually renew before continuing posting the newsletter. Anyone joining after 1 September will be subscribing to the NEXT six months issues, but may of course get copies of the previous six issues, price £6.

Thanks to all those members who not only promptly renewed but also showed their appreciation of the newsletter so far.

One rather disturbing facet has become apparent as renewals came in. It appears that most of those who have renewed are satisfied with the newsletter in its present form (with a few minor criticisms - for instance "too many games"). Does this mean that those members who do not like the newsletter, or have some serious adverse criticism, are showing this by failing to renew their membership. I suppose I will never know.

SPREAD THE WORD

In spite of Psion's claim to be selling ten to twelve thousand Organiser PER MONTH, I still come across many many people who have not heard of the Organiser, and some of these are in the ranks of professional computer people. Some members are doing a good job as "honorary salesmen" for the Organiser and quite a few have introduced their friends (and customers) to IPSO. LES BALL and myself have been doing our bit in the Nottingham area. So far we have appeared on BBC RADIO NOTTINGHAM, had a half-page spread in the local newspaper (and got the News Editor to join IPSO!) and given lectures and demonstrations to some of the local computer clubs. Perhaps you could do the same in your area.

COMPETITIONS

So far my two attempts to get you all involved in little competitions have met with stony silence. If you can remember that far back, the first one was to write a chiming clock program. One or two members said that they were working on it, but so far - NOTHING! The second one was for the BEST ONE LINE PROGRAM. Again no response. So to start you off I have written a workable one-line WORD PROCESSOR (see below). Plug in your printer and try it out. You will notice that it is possible to actually write with it, with one major drawback - it is necessary to press each key TWICE, that is all except the ESCAPE key (ON/CLEAR), when you have finished writing. Now all you have to do is to edit the program so that it will respond to EACH key-press.

```
writer:  
DO : LPRINT GET$; : UNTIL GET=0
```

A small mystery prize for the best solution.

My main leisure activity is Genealogy. I developed an interest some five years ago, did the necessary research for my direct line and constructed a tree dating back to 1550. That did not satisfy my interest which grew and grew as the bug bit deeper and I am sure that many of you readers will know just what the means. For me it is a fascinating hobby and one which has created introductions to some very nice people who have a similar interest either in genealogy in general or more particularly in the LAST Family. I have interested people who write to me from all over the world now. Some of them I have been able to help, some of them I could not. You see, many of them only wrote after years of frustration in their own researching.

My early researching was a rather haphazard business because I am not very methodical, nor am I logical, preferring to dig here and there to see what I can unearth. It produces results because it means I cover much more ground in less time than I would do if I was more controlled in searching. It also produced some bad errors of provenance which have taken time to sort out.

As I made progress, and that means I collected many more names of persons and places with dates to go with them, so I began to be bogged down with records on paper, collected in a somewhat random fashion and set out in a similar random fashion. I decided to put the data onto a computer, and purchased an Einstein computer less than 2 years ago. Now let me stress that the decision was made in defiance of advice from colleagues and friends who had said that it would be money wasted for me! How wrong they were, because I still consider the purchase of the Einstein to have been one of the most rewarding capital outlays I have ever made. It has been very costly, because my system was bought when prices were much higher than they are today. Too old for wine, women, and song now, not interested in booze, gambling - it might be thought that life is quiet for me. Not so, because computing has opened up great new horizons for me. I get a tremendous thrill from manipulating data, in succeeding in running a new program and being able to help someone who has written to me for help.

I use the computer for handling data, storing it, sorting it, manipulating it. My records are now in a neat and tidy order and of course take up very little space in the house, being stores on 5 inch and 3 inch discs. I do have more than 20,000 names on file, not all of those sorted yet because it is an ongoing job, rather like looking after the Forth Bridge. I doubt I shall ever finish the exercise, because I am constantly collecting new data. So far I have not been able to use the machine to help me cross reference people to form family links to any great extent. True I have done my own line, but I believe that there is great scope for further work yet, once I have found out how to do it.

Until a few weeks ago, I used to collect my data in notebooks, then transfer it into the Einstein, usually in two stages. Firstly into the word-processor which allowed me to then print out hard copy in a nice neat form which I could then

place in a loose-leaf book. Secondly it was entered into the database, Infostar, and more recently dBase II, where I can now manipulate away to my hearts content.

I had seen several ads for Psion and had laid hands on one, but thought it a rather expensive gimmick (one which would obviously appeal to a certain class of person). I wonder if any of you have my feelings of doubts and uncertainty when contemplating these new machines. The adverts are often not quite fair or even honest, and it was not clear that I would be able to use the Psion for what I wanted to do. Sales assistants were unable to help, because they tried to disguise their lack of knowledge with flannel. It did seem that the Psion was set up for intimate relations with IBM and compatible machines, but few people have ever heard of Einsteins.

It was my good fortune that I had a small sum of money come to me unexpectedly, just enough to buy a cheaper Psion. My experiences with Albert (Einstein) had made me realise that, once the initial expense was over, all too often it did no end there and it was quite easy to spend double the sum first thought of. I bought the Psion, and soon doubled the outlay!

I now have the RS 232 link, original version, which I would like to upgrade because the new version looks to be very good, Finance pak (which tells me where my money has gone BUT does not stop me from spending it when perhaps I shouldn't) and I now have the XP version. I consider the money to have been well spent and I am able to do what I had hoped to do, namely use it as a data collection device in my genealogical researches.

I now use Textbase, which I have set up and defined the fields in the same way as I have defined dBase and Infostar. Having entered data in, I can transfer it across to the Einstein and into the bigger database whilst Psion translates as it transfers. Mind you, this is all fairly new, but it has been done successfully with only a few hiccups. Commas have caused problems, and I do sometimes forget to distinguish between upper and lower cases, and I do use both in entering data. There is now a great advantage to me in that I only enter the data once, into the Psion, which has reduced the possibility of errors. It is a more compact and convenient form of storage whilst travelling about the country in my searches and I do have the other several advantages which the Psion offers and which you all know about.

Transferring across to the Einstein was tricky because I could not find out how to re-wire the plug connections. I had bought the Comms lead of course together with a BBC adaptor, because Albert has the same DIN type socket as the Beeb, BUT the wiring was different. The Psion manual did not make it clear just which I had to do and the Einstein manuals were not much help either. Colleagues I asked were not sure what to do either, but eventually I made a decision and persuaded a friend to re-wire the DIN plug for me (I am not much good at these micro-electronic connectors). Then I had to find out what commands to use to start the thing off, and the Psion manual was fairly helpful, once I knew just what it was I had to do in selecting protocols, Baud rates, etc. To anyone versed in

these things, then they would have been quite simple, but to me it is very much a strange new world and one in which I tend to exist in a mental fog. Eventually it all worked.

If anybody needs help in doing what I have done, then please get in touch with me through Mike O'Regan and I shall be pleased to help if I can.

FILENAMES

We all know that the NAME of a procedure or file should not be more than 8 characters - it can certainly be less, if required. However, what is becoming quite common in some of the programs submitted is that we are all using the SAME NAMES - for instance the name UDG (for User Defined Graphics) has been used quite a few times to name procedures which either define pre-set graphics or allow us to define our own. The problem arises when procedures call other procedures with ambiguous names. If the offending programs are commercial ones and in OBJECT CODE only, then it is impossible to edit them so that the duplicate proc names can be amended. I know it is difficult, but we should make a rule that we will try to avoid duplicating proc names and that we should certainly not use OPL WORDS as either proc or file names. If we use the allowed eight characters wisely, it is quite possible to develop a logical system, so that we are unlikely to use the same name twice. A system which I have developed at work may give some indication of what I mean. When I introduced micro-computers to my firm, I decided that, if we were to avoid chaos, I had to insist on a file name system which would be observed by all users. It works like this:

1st character = department (T for Traffic, E for
Engineers etc)
2nd,3rd & 4th chars = main file ref (e.g. PER for
personnel)
5th & 6th chars = type of file (RE for report, SO for
sort, PR for process, etc)
7th & 8th chars = digits 00 to 99

So if I see a filename TPERRE05 I can immediately recognise that it is Traffic, personnel, report number 5.

You will see that I have used a similar idea in my TEXT procedures (IPSO FACTO Pages 44-46) where the secondary procs have all got the prefix T to relate them to TEXT, the main procedure, followed by READ, WRITE, PRINT, SHOW, and ERA (erase)

32K RAMPAK ARRIVES.

As some of you already know, Psion has been working on a RAMPAK for some time. Its emergence has been dogged with bugs, but it seems that they have finally managed to overcome these with the issue of the 32K RAMPAK, price £54.95. With the use of this pak a whole new area is open to Organiser users. For instance it is now possible to design a database with full EDITING facilities, knowing that you can use these without filling up a datapak with erased data, until the dreaded message "PAK FULL" appears, making it necessary to do a file transplant to another pak until this same thing happens. With a big database, this can be a real pain. Another (not quite so obvious) use is in PROCEDURE COMPACTION. Having a RAMPAK installed means that you can now develop quite long procedures in resident RAM, COPY these to your RAMPAK as object only, then COPY them back to resident RAM, if required. This is particularly effective if you wish to use an existing procedure as the basis for a new procedure. In this case you would EDIT the existing procedure, then COPY it to your RAMPAK using the name of the NEW procedure - e.g. COPY A:prog1 to B:(RAMPAK)prog2. The RAMPAK can also be used to store long passages of text, which can of course be erased without recourse to a FORMATTER when the job has been printed. I will personally take advantage of this by writing even more of IPSO FACTO on the Organiser itself. Up to press, I have only done the occasional page on the Organiser, but there is now no reason why the WHOLE NEWSLETTER could not be produced that way. Saving to RAMPAK is FAST, compared to ROM. The next logical step must surely be a 64k or 128k version.

By the way, the instructions supplied with the RAMPAK are careful to point out that it is easy to accidentally wipe out the entire contents of the pak if ANY device is removed or inserted while the Organiser is still switched on. The instructions also point out that the pak behaves differently with different OPERATING SYSTEMS, even including a short procedure which enables you to check out your version, so that there can be no doubt:

```
VERNO:
LOCAL a%
a%=PEEKB($FFE9)
PRINT "Verson:",
PRINT LEFT$(HEX$(a%),1);
PRINT ":";
PRINT RIGHT$(HEX$(a%),1)
GET
```

DATAPAK PRICE REDUCTIONS

You will no doubt be aware that the expected reductions in the price of the two biggest datapaks has been announced (see enclosed Widget leaflet) at:

128k datapak	£99.95
64k datapak	£59.95

Whilst this represents a definite improvement on the old prices, it still looks pretty steep compared to the prices for PAKs announced for the new Cambridge (Sinclair) ZX88

FEEDBACK GALORE!

IAN WALKER has suggested that members might like to form Regional Groups to talk and exchange ideas.

This is an excellent idea, but I am afraid that many members have shown reluctance to allow me to publish their names and addresses, or even disclose the information to other members individually. I asked in a previous newsletter for anyone with objections to contact me and whilst only a small minority actually did express their objection I still feel that I should not use this information without the express permission of the member concerned. A number of commercial enterprises have also approached me to let them have a list of members, which I quite naturally declined. Nevertheless, I still think that regional (and perhaps other) groups could be of great benefit. For those in his area here are IAN's details:

Mr I.J. Walker
24 Grampian Road
Stirling
FK7 9PF
Tel: 0786 78103

PAUL ANTHONY BROWNING would like to contact any other Structural Engineers in the group. (I don't think we have any others, PAUL, but I may be wrong!). He also asked for on INDEX to IPSO FACTO. I have it in mind to produce a fairly detailed index of both subjects and authors at the end of the first volume (Feb '88).

SERGE GERAUD has two queries. First, he is looking for a program to do TIME CALCULATIONS (like the DEGREES, MINUTES and SECONDS on many calculators). The answer to this is probably a program written by member WILL CHAPMAN (mentioned in previous issues) called TCalc. SERGE also asks if anyone have tried to print the new POCKET SPREADSHEET output on 5 inch wide paper. The answer is "yes" - I print out drafts of spreadsheets on the TANDY CGP 115, using the 80 column mode. This gives me a very good idea what a full-size version will look like on A4 paper.

A.M. DEELEY has sent me another reminder that the Psion Mains Adaptor is NOT standard and that the use of any other third party adaptor is likely to damage the Organiser.

JOHN CROOK has successfully linked his Organiser to a SPECTRUM via the Comms Link and a cable he obtained from HIGHBLADE Ltd (Tel. 07683 52560), cost £10. For further details or any other help for Sinclair Owners contact:

John Crook
Pencoed
Sheets Heath Lane
Brookwood
Woking GU24 OEL
Tel: 04867 2391

RICHARD SNOWDEN wants to know if the old RS 232 cable will allow him to connect to IBM compatible machines. The answer is "yes", if you have some kind of Comms program for the IBM (clone). I use a "shareware" program called PROCOMM, which is very effective.

DAVID GRAY takes me to task for "promising to print things in the next newsletter" which do not materialise. Guilty, M'Lud. I did promise to print a DATAPAK PROTECTION PROGRAM earlier and have failed to do so. The reason is that the routine, which I obtained from Psion, is OBJECT ONLY. However, I am able to provide copies to anyone sending me a DATAPAK and return postage. (see also elsewhere in this issue about PAKS and POSTAGE!). A word about the VAT programs which I "started but didn't finish". I have decided that the subject is not quite the thing for the Beginners Page, so have gone on to other things. The new POCKET SPREADSHEET is ideal for VAT purposes, by the way.

DAVIS RIHOY has passed my some valuable information. He has had upgrades for both his Organiser (to 32K) and his RS 232 Link (to Comms Link), cost £50 and £32.50 respectively, for those interested. DAVID also has a query as to whether it is possible to COPY procedures from one device to another from within an OPL procedure and whether this can be made to work with SOURCE and/or OBJECT code. Has anyone any ideas? This one has me stumped.

ALLEN STRAND suggests a national "get together". This has been on my mind since IPSO started. It should be possible to select some central venue, which would suit most of the membership. Let me have your reactions along with suggestions about what we should actually DO if we did get together.

STEPHEN MCGLOIN writes to ask "if possession of one of these machines (the Organiser) normally results in total disruption of the owner's sleep patterns". I am sure we ALL know what you are going through, Stephen. I get some of my best ideas in the middle of the night, and this tendency has increased since I owned the Organiser. When I told this to my wife she just said "So that's where our three children came from!"

It is a pity that the good old TANDY CGP 115 is being discontinued by Tandy. Many members already own this printer and we will be featuring some special pages on it in future issues. I am told that there are still a few around the country in the various Tandy shops, and the pens and paper rolls will continue to be available through Tandy for the foreseeable future. CHRIS REES has recommended the CENTRONICS GLP II printer (and I can endorse that, Ed.). It is cheap, has a reasonable NLQ (near-letter quality) mode and, most important, has both CENTRONICS and RS 232 interfaces.

PERSONAL SMALL AD - FOR SALE Psion Comms Link and POCKET SPREADSHEET, brand new, sensible offers or willing to swap for ATARI ST software/hardware.

Telephone R.McKay on 0408 21870

POSTAL ARRANGEMENTS

A few members have sent me DATAPAKS through the post, either with programs and/or files on them or requesting software. This method is great for letting me have listings of long, involved procedures as I can download them straight to the Beeb for further action. However, one disturbing element is that many are sending PAKs with little or no protection, either in the form of a padded JIFFY BAG or RECORDED DELIVERY. I would urge anyone sending precious datapaks through the post to spend an extra few pence to ensure that the pack doesn't get "lost" in the post. I received a packet recently which had burst apart in the post and it was only a miracle that nothing got lost - the parcel contained leaflets, a 16K datapak and a subscription cheque! I will welcome any programs or letters, etc on datapaks (I can read any letters written with either the Harvester LETTER ORGANISER or our own TEXT program) and will ensure that the paks are returned to you properly packed and RECORDED DELIVERY.

UPPER CASE

Remember my piece in one of the early issues about a padded case for the Organiser. Well many members took advantage of my offer and bought one for £1.50. However, since then I have discovered that there are quite a few suitable cases around which appear to be tailor-made for the Organiser. One of the best is a small CAMERA CASE with end zip-fastener which can be bought from SUPASNAPS (and probably many other camera shops) for the princely sum of £1.95. The best one I have come across so far was given to me by TONE FINE. It is the smallest camera case available from W.H. SMITH at £3.95. This case has a small pocket on the side, just big enough to take a spare battery and about six datapaks. It also has a shoulder strap and a belt strap for those who tote their Organiser like a gun. Finally, there are the special fitted cases which can be obtained from member JOHN NUDING, of MOBY COMPUTERS (see leaflet with Issue 4)

DAMAGE TO YOUR ORGANISER

The use of a suitable case can avoid quite a lot of superficial scratches, etc. However, some members have complained about the labels on the keys wearing away with heavy use. Personally, I have not come across this one, and I would guess that my Organiser gets more use than most. All I can think of is that those complaining of this fault may be hitting the keys with their finger-nails. One fault which I have noticed is that the paint on the metal part around the display will flake off quite easily (I even have a tiny patch on my otherwise pristine XP). One of two mentioned this. Is it widespread?

PROGRAM PAGE

The following program is in BASIC to enable procedures to be written on an AMSTRAD CPC6128 for the Organiser. Over to CARL BURGESS.

My desktop hardware consists of an AMSTRAD CPC6128 + AMSTRAD RS232 SERIAL INTERFACE + text editor on ROM. This in my opinion is much more flexible and easier to use than the PCW8256

Procedures are written using the text editor and saved to disc. On leaving the text editor, a small basic program (see listing) is used to transfer the file down the RS232. The text, basic program and editor can all coexist in RAM at the same time. I have not yet managed to lift text direct from the RAM used by the text editor, and hence have to resort to saving and reading files from disc.

The RS232 link will work unaltered with the AMSTRAD serial interface. However, I cannot get handshaking to work. This means using a slower than normal baud rate.

The basic program serves several functions:

To strip indentations and comments from the text (I use curly brackets before comments in the text file).

To insert a delay between each line of text sent. In the absence of handshaking the Organiser seems to like a short time to "digest" complete lines.

To send the ^Z character at the end of transmission.

Receiving files from the Organiser has presented little problem, using the inbuilt software in the AMSTRAD interface. Listing procedures is a little complex. The simplest solution was to use the !TERMINAL command in the interface and then use a patch to copy all text sent to the screen to the printer.

```
10 MODE 2
20 WINDOW£1,1,80,1,20
30 WINDOW£2,1,80,21,25
40 PRINT £1,"AMSTRAD TO ORGANISER PROGRAM TRANSFER": PRINT
50 !SETTIMEOUT,5000
60 !SETSIO,1200,1200,0 : REM 1200 baud
70 PRINT "Filename:"
80 INPUT a$
90 CLS
100 OPENIN a$
110 WHILE NOT EOF
120 LINE INPUT£9,a$
130 PRINT£1,a$
140 FOR delay=1 TO 500 : NEXT : REM small delay between lines
145 :
150 a%=INSTR(a$,"(") : IF a%>0 THEN a$=LEFT$(a$,a%-1)
160 WHILE RIGHT$(a$<1)=" " : a$=LEFT$(a$,LEN(a$)-1) : WEND
170 WHILE LEFT$(a$,1)=" " : a$=RIGHT$(a$<LEN(a$)-1) : WEND
180 a$=a$+CHR$(13)
190 s%=0 : IF INKEY$=CHR$(26) THEN !OUTCHAR @s%,26 ELSE
!OUTBLOCK,@s%,@a$
210 WEND
220 CLOSEIN
230 s%=0 : !OUTCHAR,@s%,26
240 CLS
250 PRINT "Program terminated"
```

SORRY FOR THE SHORT SPACE DEVOTED TO PROGRAMS. THERE WILL BE MANY MORE PROGRAMS AND PROCEDURES IN FUTURE ISSUES.

IPSO FACTO

NEWSLETTER OF THE INTERNATIONAL PSION ORGANISER USER GROUP

VOLUME 1 * * * NUMBER 8 * * * OCTOBER 1987

Editorial Address: Mike O'Regan
130 Stapleford Lane, Beeston, Nottingham, NG9 6GB
Telecom Gold (Microlink) Postbox MAG11839

CONTENTS

71	Editorial	72	Feedback
73	Communications	76	Progs & Procs

Editorial

As promised last month, there are plenty of procedures this month. I have had many requests from CGP Owners (CGP stands for Colour Graphics Printer - previously available from Tandy and owned by many members). In response I have included a complete set of excellent routines written by member MIKE NASH. Another subject which has brought quite a few enquiries is the problem of connecting the Organiser to other devices, either microcomputers or printers, modems, etc. The two articles by STUART TIRSUN and PETER JOHNSON should help, and I will be featuring other computers and devices in future issues.

Observant members will have noticed in the header of this issue that there has been an addition. I have now obtained an electronic mail (email) box with Microlink as an experiment. In this area I am a complete tyro, so would welcome any advice on the subject. There should be opportunities for me to send mail (even IPSO FACTO?) by email to those members suitably equipped.

Another new departure this month is the release of the first IPSOSOFT package, AUTOSCRIBE. This suite of word-processing programs was developed to fill a gaping hole in the Organiser software field. It was written by myself and ROBERT SNOULTON, after much correspondence (Robert lives in Bournemouth). See enclosed leaflet for more details.

I have had quite a few enquiries about the possibilities of having a members "get-together" and I am making tentative plans for early in the new-year, so as to give plenty of warning to those who would like to attend. At the moment, it looks as if the most convenient day would be either a Saturday or a Sunday. I have started to make tentative enquiries about a location. It just so happens that NOTTINGHAM is a fairly central point for all our membership, being almost equidistant for our northern members, just a bit further for the Scottish

members, but this is balanced by the large number of London and South based members. I am looking at a venue within quick and easy access from the M1. The day should start at about 10 a.m., break for a buffet lunch at mid-day and disperse at around tea time. I have already had a few volunteer speakers and our business members are willing to display and demonstrate their wares for our mutual benefit. What do you, the members, think?

FEEDBACK

PAUL BROWNING would like to know if it is possible to alter the PITCH of the keylick, in addition to its DURATION (see Program Section). Does anyone know the answer.

HAROLD RAMSEIER (Switzerland) sends us details of a pocket modem in reponse to a previous enquiry. It is the MIGENT POCKET MODEM, a miniature 300/1200 bps Hayes compatible modem. Compact in size and battery powered, it looks like an ideal companion for the Organiser. The cost is about \$189 and it is available from:

LOGICSOFT (an MSC International Company)
110 Bi-County Blvd., Dept 1207
Farmingdale, NY 11735 USA

This firm will accept phones orders using VISA etc. on their HOTLINE (1-800-645-3491)

TIM GATHERCOLE has a query. Here is the detail.
". . . when calculating various designs I frequently have to solve angles and side lengths on a right angle triangle. Frequently the shape crops up in the type of designing I do. Rather than calculating it all laboriously on a calculator, I would like to give the Organiser the deails I know and let it calculate the rest. However, the Psion only has radians in its program. Translating into degrees is easy enough, but the problem lies in calculating the inverse trigonometric functions. Basically I don't know how to do it. Has anybody got any ideas? I expect i'm missing something fundamental"

This article is an attempt to explain the use of the RS232 Link to communicate with a (larger) microcomputer. The new Comms Link can be used in the same way. Although it has many more capabilities than the RS232 Link, they are not really necessary for this purpose. I have a Sinclair QL and my examples will use it, but I will try to make this article as general as possible so that it can be used with many different types of computers.

The first problem is to set the interface parameters. The RS232 "standard" has many options, and choosing among them is one of the most difficult tasks for those unfamiliar with data communications. If anyone is interested, I can explain what these options are and how they may be used in another article. For the moment, it is sufficient to say that you should choose the parameters that most simplify the task.

The RS232 or Comms Link must always be used on the Organiser for any type of data communications. The parameters that are chosen for the Link make no difference to the used: a simple protocol is just as easy to use as a complex one. That is usually not true on the other end of the link. The microcomputer can usually handle any protocol the Organiser can produce, and often many that it cannot handle. However, the more complex options on the Organiser may require a special communications program on the micro while simpler options may be handled directly by the device driver portion of the operating system. The latter situation is preferable because the micro can handle the Organiser as flexibly as any other peripheral device. When you use a data communications program the operations on the microcomputer become vastly more complicated because you must constantly use the program to communicate with the Organiser and save data to a temporary file, then you must leave the communications program in order to use another program to manipulate the data. It is much simpler when you can remove the communications program from the entire process.

As an example of this, we will look at the various setup options and see how they relate to the QL. The transmission rate is simple: it should be as fast as both machines can comfortably handle, usually 9600 baud. The next three options - parity, data bits, and stop bits, are generally what the microcomputer like best. the QL prefers NONE, 8 data bits and 1 stop bit. However an IBM PC often prefers EVEN parity, 7 data bits and 1 stop bit. The best choice is the current or normal configuration for your computer. The handshaking protocol is critical because it controls the efficiency of the data transfer. In a direct connection there is little chance for noise to corrupt the data. However we want to have a way for one computer to tell the other to wait when it has special work to do (e.g. writing to a datapak or a floppy disk), otherwise we will have to slow the transmission down to a crawl level so that both computers can always handle special situations. In the case of the QL and, I suspect, most other microcomputers, the RTS/CTS protocol can be handled directly by the serial device driver without a communications program, while XON/XOFF and XMODEM are complicated enough to require the program. I set ECHO to HOST and WIDTH to NONE so that the Organiser does nothing special. Finally, we must set the special transmit and receive characters. You should set TEOF and REOF to what your microcomputer's serial port sends at the end of the file. For the QL that is ASCII 26 (SUB on the Organiser). This is a

common RS232 option for microcomputers. Likewise, TEOL and REOL should be set to whatever your microcomputer recognises as the end of a record when transmitting a file, or the end of a line when transmitting an Organiser program or procedure. For the QL, this is an ASCII 13 (CR on the Organiser, LF elsewhere). By coincidence, these are the same characters used internally by both the QL and the Organiser, so neither needs to do any translating when using the link. Finally the TTRN and RTRN are best used to translate the field separator in the Organiser records to what the DBMS on your microcomputer uses for that purpose. Both the Psion Organiser and Psions's Archive program use an ASCII 9 (TAB character) as the field separator (another coincidence?) so I do not need a translation character. Notice that this character is determined by your DBMS program and not by your serial device driver.

I have carefully chosen the Organiser options so that I can easily access it on the QL. On the QL the Organiser RS232 link can be used by referring to the easy to remember name "ser2hc". On the IBM PC or clones, the proper Organiser set-up options will allow you to refer to it a something like "COM:1". If your operating system supports device redirection then life will be much easier for you. Device redirection means that you can use a device name in a command or program (e.g. "mdv1_game_psion" or "A:GAME.ORG"). Since you do not need a special comms program to mediate between the operating system and the Organiser, the computer can now treat the Organiser like a file. For example a BASIC, ARCHIVE or DBASE III program can read or write directly to and from the Organiser by opening a QL file to "ser2hc" or a PC file to "COM:1". An Organiser program can be sent to the printer on my QL by issuing the command "COPY ser2hc to ser1", because "ser1" is how my machine refers to its printer. On a PC the same thing can be accomplished with a command like "COPY COM1: PRN". A file can be displayed on the QL with the command COPY ser2hc to scr_", and on the PC by a command like "TYPE COM1: ! MORE" where MORE is a DOS filter program. The point is that the microcomputer becomes an Organiser peripheral!

My final comment is that when transmitting data or programs between the Organiser and a microcomputer, you should start the receiving computer first. It will patiently sit and wait for the transmitting computer to start. If the transmitting computer starts first, the receiver will lose some, or all, of the data before you even tell it to start. For example, if I use a text editor on the QL (it is possible to use Psions's QUILL - perhaps another article?) to write an OPL Program and I wish to send it to the Organiser, I must follow this procedure:

1. On the Organiser, enter the COMMS menu and choose RECEIVE.
2. Choose PROCEDURE and enter the program name. The Organiser is now ready to receive the program.
3. On the QL, transmit the program by entering the command: "COPY mdv1_game_psion to ser2hc".

The transfer will proceed automatically without further work on my part. After the Organiser beeps, I can enter the PROGS menu and compile and test my program.

There are many more things to be said about data communication, but I hope this is a good place to begin. I hope you find this a useful article.

MORE ORGANISER COMMS by Peter Johnson (England)

On Page 42 of IPSO FACTO I described my experiences with the RS232 Link, trying to use it to communicate with my Amstrad PCW. Since that piece was written I have discovered how to upload data, both file and procedures from the Organiser to the Amstrad.

The successful transmission combination was found to be 2400 baud, 8 bits, no parity, one stop bit and handshaking off. On the PCW I had no trouble using Amstrad's much-maligned Mail232 to control data reception. The trick was to remember that ALT+STOP ends the transfer and closes the disk file before returning to terminal mode. I still haven't managed to load a procedure to the Organiser. (Perhaps someone else has? Ed.)

Members looking for a daisywheel printer to work with the PCW and Locoscript 2 and for a serial printer to work with the Organiser might find it worthwhile looking at the Silver Reed EX300if typewriter. As well as a 16 cps print speed and the usually extensive typewriting facilities now commonplace in large office machines, it has both parallel and serial interfaces installed. As an added bonus it might be possible to pick one up for a bargain price - I paid £349+VAT for mine when the full price is claimed to be £850+VAT. Be careful though, for W.H.Smith sell the machine without interfaces for £400 inc. With manuals dated 1984 I suspect a replacement model may be on the way.

Both Organiser RS232 Link plug and EX3001F serial socket are female so a lead must be bought or made up. (Blame IBM for introducing a non-standard MALE connection on its PC. Ed). I bought the components from Tandy for £12.05. The outlet on the typewriter already incorporates pin changes called for in the diagram on page 48 of the RS232 Link's manual so making up the lead is quite straightforward.

The Organiser will drive the Silver Reed with both set to 9600 baud, no parity, 7 data bits and 2 stop bits. Transmitting and printing a file gave no problems, but Organiser tabs are ignored and should be translated to carriage returns or a printer character. Long lines should be broken up to stop the typewriter printing off the paper (or use wide paper!). The only thing I couldn't do was to get the Silver Reed to transmit data to the Organiser which is supposed to be possible. Does anyone have any tips?

Oh yes, the machine works a treat with LocoScript 2, too.

If anyone wishes to comment on, or ask questions about, any of the foregoing, I'll be please to hear from them on Leicester (0533) 357268.

PROGS & PROCS

PIN by Ernie Bokkelkamp (South Africa)

This program is very similar to the password program as provided in the manual. The program does not display the password as it is being type in and will clear the display and switch off when the <ON/CLEAR> is pressed. The password must be 5 digits only. Any other characters will be ignored. To be able to start your own private applicaiton, you can insert the procedure name before the RETURN (last line)

```
pin:
LOCAL i%,c%,pin$(5)
ONERR GOTO start::
start::
OFF
CLS
PRINT "Your name"
PRINT "PIN: ";
i%=0
pin$=""
KSTAT 3
CURSOR ON
WHILE i%<5
c%=GET
IF c%=1
GOTO START::
ENDIF
IF c%=8 and i%>0
PRINT CHR$(8);" ";CHR$(8);
i%=i%-1
ELSEIF c%>=0 AND c%<= 9
pin$=pin$+CHR$(c%)
PRINT CHR$(8);
i%=i%+1
ENDIF
ENDWH
CLS
IF pin$<>"12345"
GOTO start::
ENDIF
i%=1
DO
BEEP 15,i%*20
i%=i%+1
UNTIL i%>12
ONERR OFF
REM - - INSERT YOUR ROUTINE HERE IF YOU WISH
RETURN
```

CARPET by R Mackay (Scotland)

This little routine works out the length of rolled material (in this case carpet). It can be easily adapted and/or expanded.

```
carpet:
GLOBAL a,b,c
PRINT "Outer Inches"
```

```

INPUT a
PRINT "Inner Inches"
INPUT b
PRINT "Number of layers"
INPUT c
PRINT (a+b)*(c/2)
GET
PRINT "Feet long"
PRINT (a+b)*(c/2)/12
GET

```

CLICK by PAUL BROWNING (England)

Paul has expanded my original CLIK(page 32). When the program is run the display first shows "Click=0" and when keys are pressed no click will be heard. Pressing the UP cursor key increases the click length, pressing the DOWN key decreases the click length. Pressing Q quits to the main menu. Note that when click=0 all alarms are still active.

```

click:
LOCAL k%,n%
KSTAT 1
a::
AT 1,1
PRINT "Click = ";n%
k%=KEY
IF k%=81 OR k%=113
STOP
ELSEIF k%=3
n%=n%+1
ELSEIF k%=4
n%=n%-1
ENDIF
IF n%<0
n%=0
ELSEIF n%>255
n%=255
ENDIF
POKEB $20C0,n%
GOTO a::

```

PROCEDURES FOR THE TANDY CGP 115

by Mike Nash

The following procedures are intended to be a complete library of CGP procedures. They have rather long names because of the need to group them together in any "dictionary" of new words that the User Group may develop. There is no reason why the names should not be shortened by individuals or a parent calling procedure used. Many of the less experienced members of the group may not be familiar with the use of parameters - they are well worth studying. The main thing to remember is that they can only be run on their own from CALC or called from another procedure. Also remember that if the procedure is to return a numeric or string value, as "qn%:(prmt\$)", and "qstr%:(prmt\$)" the qualifier "\$" or "%" must match the type of value returned. If these strictures are observed, some very useful procedures can be created.

n.b. The REMs can be discarded except for library use.

Example in CALC:

```
tacir%:(qn%("xcoord"),qn%("ycoord"),qn%("radius"))
```

Organiser then prompts for input:

```
"xcoord"?
```

```
"ycoord"?
```

```
"radius" and draws a circle on the CGP115
```

Procedures

```
clik%:(d)
```

```
POKEB $20C0,d : REM adjusts keyclick duration
```

```
tabksp:
```

```
LPRINT CHR$(8) : REM produces backspace on printer
```

```
tarvlf:
```

```
LPRINT CHR$(11) : REM produces linefeed on printer
```

```
tarot:
```

```
LPRINT CHR$(29) : REM rotates pens in text mode
```

```
tatext:
```

```
LPRINT CHR$(17) : REM select text from graphic mode
```

```
tareset:
```

```
LPRINT "A" :REM select text from graphic mode and reset origin  
to left margin
```

```
tacol%:(n)
```

```
LPRINT "C";n :REM parameter "n" = pen positions 0 to 3
```

```
tahome:
```

```
LPRINT "H" :REM returns to origin without drawing
```

```
tadrel%:(x,y)
```

```
LPRINT "J";x;",";y :REM draw from current pen position "x" steps  
sideways and "y" steps up/down
```

```
tamab%:(x,y)
```

```
LPRINT "M";x;",";y :REM move without drawing to "x,y"
```

```
taltyp%:(n)
```

```
LPRINT "L";n :REM select different line types as per manual (0  
to 15)
```

```
tasrtng%:(str$)
```

```
LPRINT "P";str$ :REM print text string in graphic mode
```

```
tasize%:(n)
```

```
LPRINT "S";n : REM choose 0 to 63 text sizes set up in graphic  
mode to alter text mode size
```

```
tarpd%:(n)
```

```
LPRINT "Q";n : REM select 0 to 3 text direction in graphic mode
```

```
tamrel%:(x,y)
```

```
LPRINT "R";x;",";y : REM move pen relative to current pen  
position
```

```
taxisy%:(s,i)
```

```
LPRINT "X";0;s;",";i : REM draw "y" axis as per manual
```

```

taxisx%(s,i9
LPRINT "X";1;s;",",i : REM draw "x" axis as per manual

tadrab%(x,y)
LPRINT "D";x;",",y : REM draw to destination at "x,y" in
relation to origin

tareloc:
LPRINT "I" : REM change origin to current pen position

tacir%:(x,y,r)
LOCAL x3,y3,x1,x2,r1,r2,i
r1=r
r2=r
x1=0
x2=0
IF R<25
i=.1
ENDIF
x3=-x
y3=-y
tamab%:(x,y)
tareloc:
r1=r
r2=r
x2=0
tamrel%:(x2,r1)
x1=0
DO
tadrab%:((r1*SIN(x1)),(r1*COS(x1)))
x1=x1+i
UNTIL x1>(2*PI)
tahome:
tamab%:(x3,y3)
tareloc:
REM draws circle

tagraph:
LPRINT CHR$(18) : REM select graphic mode from text mode

tablack:
LPRINT "CO" : REM selct black pen

tablue:
LPRINT "C1" : REM select blue pen

tagreen:
LPRINT "C2" : REM select green pen

tared:
LPRINT "C3" : REM select red pen

qn:(prmp$)
LOCAL n
CLS
AT 1,1 : PRINT prmp$;CHR$(63)
AT 1,2 : INPUT n
RETURN n

```

Authors note: Needs improvement
- speed is in inverse proportion
to accuracy.

```
qn%:(prmt$)
LOCAL n%
CLS
AT 1,1 :PRINT prmt$;CHR$(63)
AT 1,2 :INPUT n%
RETURN n%
```

```
qstr%:(prmt$)
LOCAL str$(16)
CLS
AT 1,1 :PRINT prmt$;CHR$(63)
AT 1,2 :INPUT str$
RETURN str$
```

To try out these last three routines the LSET: command is perfect. e.g. LSET(qn%:("Baud"),qn%:("Parity"))
Organiser prints "Baud?", etc.

EDITORS NOTES FOR SUBMISSIONS FOR THIS SECTION

I am getting quite a flood of procedures now, some of which are hand-written, some printed. Would software authors please note that, if your submission is longer than just a few lines, I would prefer the whole thing on a DATAPAK (listings as well as source code). I will return your pak by recorded delivery (after erasing it for you if you so wish). This will have the advantage that I will not make any silly typos when tapping in your long listing (quite apart from the time saved!) Also try to keep to the sensible conventions of using LOWER CASE for variable and procedure names and UPPER CASE for OPL words. It is also a good idea to avoid names which clash with existing OPL words (OPL will allow you to do this, but don't!). Try to make your procedure names UNIQUE. I need both hands to count the number of "UDG:"s I have seen so far. Duplicate names can produce some odd unexpected results.

IPSO FACTO

NEWSLETTER OF THE INTERNATIONAL PSION ORGANISER USER GROUP

VOLUME 1 * * * NUMBER 9 * * * NOVEMBER 1987

Editorial Address: Mike O'Regan
130 Stapleford Lane, Beeston, Nottingham, NG9 6GB
Telecom Gold (Microlink) Postbox 72:MAG11839

Editorial

Just over a year ago I bought my first Organiser II. The Organiser was not my first computer, or even my first Pocket Computer, but I can now say that it has probably taught me as much if not more about aspects of computing which I had not previously even considered, such as setting up a working database, communications between quite different computers, etc. I think as Organiser enthusiasts we have all developed a great affection for the little machine which I have seldom encountered with other computers. Letters come in practically every mail backing this up. One or two members who have had problems which involved returning their Organisers to Psion for attention have spoken of "withdrawal symptoms" until their machines were returned.

This month sees the first of a series on Machine Code for the beginner, written by my blind friend LES BALL, who has used machine code extensively on the Organiser to develop his SPEECH facilities. As I am somewhat of a beginner in this area, I am working on the assumption that if I can understand what it is all about, then it should be pitched at the right level for most of our members.

With regard to the Technical Reference Manual, Psion have now provided me with a (very thick) PRINTOUT of the draft manual, which some of you may have seen and possibly obtained a copy of. However, I am still waiting for the files in DISK format so that I can make these available to the members. You will be kept informed.

CONTENTS

81 Editorial
84 Progs & Procs

82 Review - Cubsoft FNKEYS
90 Machine Code for Beginners

IPSO REVIEW - CUBSOFT FNKEYS

Background

For anyone who has used a well-equipped desk-top computer for any kind of large scale data entry or program development, the use of FUNCTION KEYS is a well-known and valued facility. For those who have not had this experience, a short explanation is in order.

The keyboards of many computers have a number of keys (usually either 10 or 16) marked f1, f2 etc. These keys are of enormous value where any kind of repetitive input is concerned, as they can be customised by the user to produce long or short strings of characters at the touch of a single key. Many commercial software packages also make extensive use of function keys for a wide variety of uses. Some keyboards (for example the WYSE 50 TERMINAL which I use in my office) have function keys which feature non-volatile memory, and are thus capable of "remembering" their contents when the terminal is switched on after a power-down.

What CUBSOFT have managed to do is nothing short of a minor miracle. They have devised a function-key system for the Organiser which rivals the facilities on many bigger machines, and all this on a keyboard which is not big enough to sport its own "dedicated" function keys. FNKEYS is one of those (rare) programs which, once you have used it, make you wonder how you managed previously without it!

Basics

FNKEYS is supplied on a datapak, but once the program has been "initialised" it resides in the Organiser's memory and the pak may be removed. If required FNKEYS may be removed by inserting the pak and running the program again, which effectively disables FNKEYS and releases the small section of memory which it had used as a buffer. As FNKEYS has a great many options, it is necessary at first to keep the instructions (or at least the summary sheet) to hand, but you soon get to know your favorite ones and memorise them.

In use

FNKEYS uses the MODE key on the Organiser extensively, and so it has an option to redefine this key (known as the HOT KEY) so that it will not seriously clash with other normal uses of the MODE key (e.g. with POCKET SPREADSHEET, or AUTOSCRIBE). The other "main" keys are SHIFT, and the 4 CURSOR (arrow) keys. It is quite ingenious how many separate facilities have been crammed into this program, and this review can only hint at some of the applications and give you a taste of what is possible. I will illustrate one of my own uses as an example:

As many of you know, I use my Organiser and AUTOSCRIBE for writing much of my Group correspondence. I write letters using the filename "alet" and I also keep a file of the IPSO Letterhead under the filename "bhead". This last file is on the same pak as AUTOSCRIBE as it is continually in use. I also need a file copy of each letter. Using FNKEYS, I have "taught" the Organiser to do the following sequence at the touch of JUST TWO KEYS!

1. Check that the COMMS LINK is enabled.
2. Load my letterhead file
3. Get the date from the system clock
4. Insert it in the correct position
5. Load my letter file
6. Set the printer up to a pagelength of 70
7. Print the whole letter

8. Print an extra copy for the file
9. Quit AUTOSCRIBE and switch off the Organiser (if required)

All that is required to run the above sequence is <MODE> <L>!

To show exactly how much was involved setting this up, here is the sequence of operations using FNKEYS:

1. <MODE> <RIGHT ARROW> (starts recording)
2. Go through the required sequence MANUALLY, being very careful not to make any mistakes (because FNKEYS records EVERY keypress)
3. <MODE> <RIGHT ARROW> (to end recording)
4. <MODE> <LEFT ARROW> L

That simple procedure is all that is required to allocate the whole sequence to the L key!

And that's just for starters. FNKEY has also features which help OPL program writers by providing not only 2 key versions of the most popular OPL words, but also ensuring that you use the correct syntax with DO..UNTIL, WHILE..ENDWH, and IF..ELSEIF . ENDIF. Other facilities include a set of readymade maths functions, a DATE facility (previously mentioned), a full set of punctuation - not normally available from the keyboard), etc.

I mentioned at the beginning of this review that many desk-top computers had a number of function keys provided. By using SHIFT and CONTROL combinations it is possible on these machines to have at your disposal up to 40 (with 10 key machines) or 72 (with 18 key machines) effective function keys. Using FNKEYS a staggering 312 function keys are available at any one time! This is made possible by separating the memory buffer into 6 discrete "BANKS" and using the ALPHA UPPER and LOWER CASE keys, hence $26 \times 2 \times 6 = 312$.

FNKEYS has been subject to almost continuous modification and improvement by its author, Mike Leigh, since its inception and announcement through IPSO. The review version was supplied with only a draft copy of the (sizeable) Instruction Leaflet. I am sure that this system will take its place among the Organiser best-sellers as a valuable addition to an already remarkable machine.

As I go to press there is now news of yet another addition to the arsenal of FNKEYS features, namely CUT & PASTE, a valuable tool for use with, for instance, text editing and word-processing programs. I have not had chance to fully explore this, and one or two of the other recent additions, but will report on these at later date.

As you will have noticed from the leaflets enclosed with the last newsletter, FNKEYS is available from:

CUBSOFT
6 Okeover Road
Salford
M7 0JX

Price £34.95

PROGS & PROCS

MZAP from B Saunders

This is a handy utility for program developers. It is a routine to display the Organiser's memory in a handy form.

The screen displays:

1. The address in hex (top left)
2. The eight bytes of ASCII translation, non-printable characters being denoted by a dot (top right)
3. The contiguous eight bytes in hex (bottom).

The following controls are used:

CURSOR RIGHT	- Increase eight bytes
CURSOR LEFT	- Decrease eight bytes
CURSOR UP	- Increase one page (256 bytes) if held increase 4k
CURSOR DOWN	- Decrease one page (256 bytes) if held decrease 4k
EXE	- Reset address to \$8000 (initial value)
CLEAR	- Exit

The integer-real translations at the start are because the HEX\$ function expects 16 bit signed integers.

Please note it is not possible to PEEK (or POKE) the actual locations below \$40 using the OPL commands. This area contains hardware control registers for things like datapack selection, timers and the COMMS LINK.

```
mzap:
REM MZAP 0.2 (c) BS 270787
LOCAL l%, n,v%,a%
n=$8000
DO
IF n>32760
n=-32768
ENDIF
IF n<-32768
n=32760
ENDIF
IF (n>-1 AND n<16)
PRINT "0";
ENDIF
IF (n>-1 AND n<256)
PRINT "0";
ENDIF
IF (n>-1 AND n<4096)
PRINT "0";
ENDIF
PRINT HEX$(n);
AT 9,1
l%=0
DO
v%=PEEK(n+l%)
IF v%<32 OR v%>126
PRINT ".";
ELSE
PRINT CHR$(v%);
ENDIF
l%=l%+1
```

```

UNTIL l%=8
l%=0
DO
v%=PEEKb(n+l%)
IF v%<16
PRINT "0";
ENDIF
PRINT HEX$(v%);
l%=l%+1
UNTIL l%=8
a%=GET
AT 1,1
IF a%=3
n=n+256
ELSEIF a%=4
n=n-256
ELSEIF a%=5
n=n-8
ELSEIF a%=6
n=n+8
ENDIF
IF KEY=3
n=n+4096
ENDIF
IF KEY=4
n=n-4096
ENDIF
IF a%=13
n=$8000
ENDIF
UNTIL a%=1

```

I had quite a few replies with programs to provide Tim Gathercole with the required facilities for trig calc in Degrees. Here is the most complete set. My thanks to those members who took the trouble to write in with solutions.

Trig Calculations in Degrees by Paul Browning.

```

sin:(x)
GLOBAL s
s=SIN(RAD(x))
RETURN s

```

```

cos:(x)
GLOBAL c
c=COS(RAD(x))
RETURN c

```

```

tan:(x)
GLOBAL t
t=TAN(RAD(x))
RETURN t

```

```

asin:(x)
GLOBAL a
IF x=1
A=90
RETURN a
ELSEIF x=-1
A=-90
RETURN a
ELSE
a=ATAN:(x/sqr(1-(x*x)))
RETURN a
ENDIF

```

```

acos:(x)
GLOBAL a
IF x=0
a=90
ELSE
a=atan:(SQR(1-(x*x))/x)
ENDIF
IF x<0
a=a+180
ENDIF
RETURN a

```

```

atan:(x)
GLOBAL a
a=DEG(ATAN(x))
RETURN a

```

These functions are used either as subroutines of an OPL procedure, or from CALC mode in a similar way to the preprogrammed functions. The distinction between degrees and radians being made by the inclusion or omission of the colon (:). For example `acos:(.9)` in CALC mode returns 25.84 and `cos:(25.84)` returns 0.90, but `COS(25.84)` operates the standard, preprogrammed function and returns 0.76.

Paul also states: I am aware that some of the names of these procedures clash with OPL words but the existence of the : is a very positive distinction, and I would argue that any change of the names of these procedures would not be an improvement. I would also suggest that DEGREE is the more commonly used angle measure in this country and therefore the colon will be used more often than it is omitted. Tim Gathercole obviously like to think in degrees, and I must say that the radian is about as foreign to me as the KIP-FOOT. By the way, if anybody is interested, I also have a conversion program between KIP-FEET and KILONEWTON-METRES.

I'm afraid all this is SERBO-CROAT to me (Ed.)

NOTEPAD

by Adrian Pegg

This program is, in essence, a format for data storage. It has ten fields available to the user, automatically date-stamps every entry, and permits the use of "keywords" in the first two fields. The keyword <URGENT> (available by entering 'u' (or 'U') in either of the first two fields) can be called instantly from the main menu to page through data classified as <URGENT>. Keywords have been defined in this way to enable the FIND command to be used in the form `FIND: <R` which would page through entries classified <REMINDER>, and so on. Keywords are fully redefinable by the user, of course. LIST will page through all entries, and, in this mode as well as FIND, pressing DEL gives you the option to Delete an entry. Pressing MODE provides an option to EDIT same, (although keyword entry is not available in EDIT mode) ADD simply date stamps another record and accepts input from the keyboard (with EXE pressed at the end of each line) until EXE is pressed or the ten fields are filled. There is no reason, but the way, why there cannot be up to 15 fields (the 16th taking care of the date-stamping) should that be required. If NOTEPAD is placed in the system menu it will be readily at hand to record expenses, temporarily required phone numbers (not needed in Datapak), things to do, people to call etc.

```

NOTEPAD:
GLOBAL opt%, i%, in$(255), f$(50), c%, g$(1), k$(1)
IF NOT EXIST ("a:pad")
CREATE "a:pad", A, d$, a1$, a2$, a3$, a4$, a5$, a6$, a7$, a8$, a9$, a10$
ELSE OPEN "a:pad", A, d$, a1$, a2$, a3$, a4$, a5$, a6$, a7$, a8$, a9$, a10$
ENDIF
start::
CLS
opt%=MENU("FIND, ADD, LIST, URGENT, OFF")
IF opt%=0
CLOSE
RETURN
ELSEIF opt%=5
OFF
GOTO start::
ELSEIF opt%=3
f$=""
GOTO urgent::
ELSEIF opt%=4
f$="<URGENT>"
GOTO urgent::
ELSEIF opt%=1
CLS
PRINT "FIND:";
INPUT f$
IF f$=""
GOTO start::
ENDIF
urgent::
FIRST
DO
IF FIND(f$)
c%=DISP(-1, "")
REM *End Search*
IF c%=1
GOTO start::
ENDIF
REM *Delete*
IF c%=8
PRINT CHR$(15); "DELETE Y/N";
g$=GET$
IF LOWER$(g$)="y"
ERASE : BACK
ENDIF
REM *Edit*
ELSEIF c%=2
CLS
PRINT "Edit Y/N:"
g$=LOWERS$(CHR$(VIEW(2, a.d$)))
IF g$="y"
CLS : PRINT "Editing:"
EDIT a.d$
EDIT a.a1$
IF a.a2$<>""
EDIT a.a2$
ENDIF
ENDIF

```

```

IF a.a3$<>""
EDIT a.a3$
ENDIF
IF a.a4$<>""
EDIT a.a4$
ENDIF
IF a.a5$<>""
EDIT a.a5$
ENDIF
IF a.a6$<>""
EDIT a.a6$
ENDIF
IF a.a7$<>""
EDIT a.a7$
ENDIF
IF a.a8$<>""
EDIT a.a8$
ENDIF
IF a.a9$<>""
EDIT a.a9$
ENDIF
IF a.a10$<>""
EDIT a.a10$
ENDIF
UPDATE
ENDIF
GOTO start::
ENDIF
ENDIF
NEXT
UNTIL EOF
CLS : PRINT "SEARCH COMPLETED Press any key"
GET$
GOTO start::
ELSEIF opt%=2
LAST : NEXT
CLS
a.d$=LEFT$(DATIM$,16)
PRINT "ADD:";
INPUT in$
IF in$=""
BACK
GOTO start::
ELSE keywd:
a.a1$=in$
ENDIF
INPUT in$
IF in$=""
GOTO end::
ELSE keywd:
a.a2$=in$
ENDIF
INPUT a.a3$
IF a.a3$=""
GOTO end::
ENDIF

```

```

INPUT a.a4$
IF a.a4$=""
GOTO end::
ENDIF
INPUT a.a5$
IF a.a5$=""
GOTO end::
ENDIF
INPUT a.a6$
IF a.a6$=""
GOTO end::
ENDIF
INPUT a.a7$
IF a.a7$=""
GOTO end::
ENDIF
INPUT a.a8$
IF a.a8$=""
GOTO end::
ENDIF
INPUT a.a9$
IF a.a9$=""
GOTO end::
ENDIF
INPUT a.a10$
end::
APPEND
ENDIF
GOTO start::
RETURN

```

```

KEYWD:
IF LEN(in$)>1
RETURN
ENDIF
k$=LOWER$(in$)
IF k$="u"
in$="<URGENT>"
ELSEIF k$="r"
in$="<REMINDER>"
ELSEIF k$="m"
in$="<MILEAGE>"
ELSEIF k$="e"
in$="<EXPENSES>"
ELSEIF k$="t"
in$="<TIME SHEET>"
ELSEIF k$="c"
in$="<CALL>"
ELSEIF k$="w"
in$="<WRITE TO>"
ENDIF
PRINT CHR$(14);in$
RETURN

```


The aim of this article is to establish the level of interest in the use of machine code on the Organiser. There are two aspects to this: How many people who have little or no knowledge of machine code would be interested in extending the flexibility of OPL? and, would those of you who understand how to use machine code would be willing to contribute by passing on hints, tips, and particularly routines which you feel would be of interest and could, with your agreement, be used as examples. For now let's start at the beginning with a few definitions of words you may have seen in the Organiser Manuals.

Memory

There are two kinds of memory used in the Organiser:

ROM,
or READ ONLY MEMORY, which you cannot change and contains the system software, which consists of whole sequences of instructions which tell the processor what to do when your program gives certain commands. We can also use these sequences of instructions directly from machine code and these are known as ROM CALLS.

and

RAM (or Random Access Memory) - this is memory we can write to as well as read from, which means the contents can be changed at any time.

EPROM - this is a form of programmable ROM (READ ONLY MEMORY), which can be written to but the contents of this memory cannot be changed in the same way that we change the contents of RAM. This is the form of memory used in the Organiser Paks, and when we wish to re-use this memory we have to put the pak under an ultra-violet lamp to clear it. This, of course, clears the whole pak, and not just individual sections or locations as we do with RAM. This is important to understand because if we erase a file or procedure on a Pak it is no longer accessible to the machine under ordinary circumstances, although the information can be recovered provided the pak has not been subject to erasure by an ultra-violet lamp before we wish to recover it.

BIT - as its name suggests, this refers to one BIT (binary digit) of information, in digital form, whether 0 or 1, or in terms of a switch OFF (0) or ON (1).

BYTE - a BYTE is 8 BITS of information, again the equivalent of 8 switches, which may be turned on or off in any combination we wish. We number these BITS 0 to 7, with 0 being the LOWEST ORDER or LEAST SIGNIFICANT BIT. With eight BITS there are a possible 256 combinations, or in terms of numbers from 0 to 255.

CPU - or Central Processing Unit. This is the unit which controls the system and where all arithmetic processing is carried out. The processor in the Organiser is an HITACHI HD 6303 8 BIT micro-processor. The significance of the 8 BIT is that this is the size of the number which can be processed at any given instant, and our information is therefore transferred from one place to another internally 8 BITS or 1 BYTE at a time.

The set of wires or LINES along which this information passes is known as the DATA BUS and the wires are, like the BITS, numbered 0 to 7. The number of BITS we transfer at a time determines the way our memory is organised. Each location in memory is 8 BITS or 1 BYTE wide and the size of our memory denotes the number of locations we have available. Obviously these locations need to be numbered so that the processor knows where to look for information. The number of a location is known as its ADDRESS. Internally the processor has a number of wires or lines which it uses to "address" the memory. For now let us think in terms of 65536 locations of memory being available. To address this number of locations we would need 16 lines available. We can of course organise this memory in any way we wish, but since the processor need information to tell it how to interpret instructions and give it facilities to interface with the outside world, clearly some of this memory is not available to us. In the Organiser 32k of memory is taken up by the internal software of the machine and this is put on ROM. In theory this should leave us with 32k of memory available, which could be RAM. In practice the machine also needs some RAM for its own use, since it also needs memory it can write to as well as read from. This is known as SCRATCHPAD. The scratchpad in the Organiser contains large numbers of system pointers which tell the processor where to look for information it needs, for example how much memory is left, how much memory to allocate for certain tasks it needs to perform, where to look for application software, the length and the location of the information allied to that software that the user has put in. From our point of view, this system is particularly useful, because if we wish to redirect the processor to perform some other task before it completes its own little job, we can do so. This is known as SOFTWARE INTERFACING. Whatever memory the processor does not use is now available to us and obviously dependent on the size of memory chip used by Psion we could currently have somewhere around 22k of memory available. Of course on the Organiser the pak system means we have much more memory available, but this is now addressed in the way I have described and it would not be appropriate at this point to look at the way the Organiser addresses paks.

(to be continued)

IPSO FACTO

NEWSLETTER OF THE INTERNATIONAL PSIOM ORGANISER USER GROUP

VOLUME 1 * * * NUMBER 10 * * * DECEMBER 1987

Editorial Address: Mike O'Regan
130 Stapleford Lane, Beeston, Nottingham, NG9 6BB
Telecom Gold (Microlink) Postbox 72:MAG11839

EDITORIAL

May I take this opportunity to wish all members a Happy and "Organised" Christmas, with your stockings full of RAMPAKS, COMMS LINKS and other goodies.

Preparations are going ahead slowly for our "get-together" in the new year (before Easter, I hope). From correspondence I have had from interested members, it looks as if it will be a Saturday or Sunday in either January or February. I have made arrangements that we can book a large room in a local club and lay on a buffet luncheon. The get-together will probably start at about 10 a.m. and run until about 5 p.m. As soon as a date is settled I will let you all know. There will be room for those members who are dealers to display their wares and even demonstrate or speak to us all if required. In order to give me some idea of the number of interested members, would you please let me know, with an indication of whether a Saturday or Sunday is preferred.

Membership is expanding to the extent that I expect we will have some 300 members by the end of our first year's existence. Only about 24 did not renew after the first six-month period. We now have members in HAWAII, SARAWAK, JORDAN, ISRAEL, SOUTH AFRICA, GERMANY, SWITZERLAND, SWEDEN, and DENMARK in addition to many in ENGLAND, IRELAND, SCOTLAND and WALES. I am going to approach the distributors in the USA about some advertising "over there"!

Contents

93 Machine Code Part II
96 Feedback

94 Operating Systems
98 Progs & Procs

As well as this memory described and that of the paks, there are other small areas of memory internal to the processor, which we will use in machine code. These are known as REGISTERS, which are temporary storage areas. In the 6303 processor there are 7 such registers:

Register A)	also known as ACCUMULATORS
Register B)	
Register D		A and B registers combined
Register X		
Register SP		Stack Pointer
Register PC		Program Counter
Register CC		Condition Code

Register A is particularly important because all the arithmetic processing is carried out using this register. Registers A and B are 8 bits (or 1 byte) wide and combined as register D are 16 bits (or 2 bytes) wide.

Register X is 16 bits wide.

Register CC is 8 bits and contains what are known as FLAGS. In other words, the condition of each bit will tell the processor whether certain things have happened, for example whether the result of a certain process has produced 0 (zero) or if adding two numbers together will give us a number greater than 255 or larger than 8 bits.

Register PC keeps track of the sequence of operations when a machine code program is running

There is one other area of temporary memory used by the processor which is known as the STACK. This will require more information later, but for now imagine this as a pile of bytes of information. If we wish to look at that information, we must take off the pile what is on the top, in other words the last thing which is on must be taken off first.

Register SP does exactly what its name implies, by pointing to the position in the stack which is currently accessed.

There are techniques for manipulating the stack, and these, and the way that the stack is organised in memory will be discussed later.

All these registers will be examined in greater detail later in this series.

(to be continued)

ORGANISER OPERATING SYSTEMS

I have had an increasing number of enquiries about the different Organiser Operating Systems, and their peculiarities, so below I am printing all the known information about ALL operating systems used since the introduction of the Organiser II. You may check your version number by using the procedure published earlier (VERNO:). These notes apply to either the CM, XP or LA (32k XP).

Version 2.3 and earlier - not released in any quantity. Any machines with this operating system should be returned to Psion for update.

Version 2.4 (released 12 May 1985)

Enhancements:

1. Devices booted when the Organiser is cold booted.

Bugs:

1. If the install code of a device alters the menu, a crash may ensue on a cold boot.
2. Re-booting a machine with more than one device installed may cause a crash.
3. An attempt to translate the illegal statment "11SIN" results in a crash.
4. If a field is assigned to that is beyond those already assigned to before, it may write the data over the diary or operating system variables. This can, in rare cases, result in a crash. The fix is to assign (a null string or zero) to the last field after creating or opening a file and after getting an END OF FILE condition. (The END OF FILE condition is when EOF is non-zero. This means that all values of the fileds have been zeroed out)
5. If the diary is backed over midnight and then brought forward, using the arrow keys, any diary entries seem to have disappeared.
6. If RAM is full, editing a procedure can give OUT OF MEMORY and leave the screen corrupted.
7. If an error is encountered when adding to or subtracting from a calculator memory (MO-M9) then that memory is corrupted.
8. Floating point AND produces unreliable results.
9. VIEW(1,"") does not work properly
10. VIEW of a 255 character string doesn't scroll
11. Strings longer than 255 can be declared (e.g. 256 gives a zero length string).
12. Declaring arrays which overflow memory size can cause machine to crash at run-time. (e.g. LOCAL S\$(255,1000) or LOCAL a(10000).
13. INPUT a.a\$ artificially limits the length input to 252 minus the record size (as found from RECSIZE). It should allow 254 characters.
14. If the machine is turned off in the minute before an alarm is due, it won't go off for 34 minutes.
15. If packs are accessed at the same time as the buzzer is used (e.g. from a key click) the botton byte of the pack can, very rarely, be blown to zero.

Version 2.6 (released on 15 October 1986)

Enhancements.

1. Checksum on RAM pack reader.
2. Battery checking improved.
3. Can call OPL programs on device D (top slot Ed.)
4. INFO calculates FREE memory as percentage free of the total memory less the operating system.
5. OPL string comparison become case dependent.
6. UDGs preserved when the Organiser is turned off.
7. Intelligent, faster NEXT - works on one pack at a time.
8. A POSITION to an illegal place, e.g. past the end of file, used to leave the position alone. It now positions to the last or first record as appropriate.

Bugs.

1. If after a successful CLOSE another CLOSE is done when no files are open it can crash the system.
2. It is possible, but unlikely, to get spurious alarms when loading devices.
3. If the ON/CLEAR is pressed at the same moment as an alarm, it may be missed.
4. When a translator error is detected, some memory may be lost until the language is run.
5. Using MENU with an item with more than 17 letters causes an infinite loop.
6. When on a null diary entry, deleting it causes the next diary entry to be deleted.
7. TRAP DELETE "a:"+xxx\$: can cause a crash if there is an error in xxx\$:
8. If logical name D is used then it can cause OPEN to fail with error number zero when it should have succeeded. An error of zero is not reported as an error at the top level. It is safest not to use logical name D.
9. POS200 ONLY. The menu cell is grown every time the machine is booted, so memory eventually fills up. Solution is to cold boot the machine.

Introduced Bugs.

1. Sizing 8, 16, & 32k packs can fail even when the pack is good.
2. Erasing records on RAMPACKS followed by APPEND may corrupt the PACK. Can be solved by avoiding using UPDATE and doing FIRST after using ERASE.
3. Copying Mk I packs fails with PACK CHANGED error.

Version 3.1 (released on 30 January 1987)

Enhancements

1. If a file is OPENed/CREATed/DELETed without a specified device it takes the last device used (as opposed to a random device).
2. Reports CHR\$(256) and negative arguments to LEFT\$, RIGHT\$, and MID\$ as errors
3. Intelligent NEXT improved - works with 1 file on each device.
4. LA split device loader.
5. 128k RAM packs supported.
6. Faster deleting on RAMPACKS.

FEEDBACK

Q. DAVID RIHOY asks "How do you (or other readers of the newsletter) organise your files and procedures on datapaks? This is not as silly as it sounds - one reason why I have not made as much use of the Organiser as I might is that I am not sure how best to keep diaries, files, and procedures. I realise that one person's Organiser is probably another person's dis-Organiser, but is it better to have one diary or two (personal and work) or more, in RAM, on RAMPAK, or on DATAPAK, etc? Presumably utility procedures e.g. BEEP ON/OFF should be on every PAK, but what about e.g. CALC procedures, especially if they interact with a non-copyable PSION pak?"

A. This is indeed a vexed problem for the user who wishes to make the most of all facilities offered by the Organiser. When I first started to work with the Organiser, I thought that the 3 devices A: B: and C: would be adequate for all purposes, but I soon realised that it would be great to have YET ANOTHER DEVICE (say another slot - D:?). If you are running, say, POCKET SPREADSHEET in B: with a DATAPAK (or preferably RAMPAK) in C: for the SPREADSHEET files created, there is not much option to have access to other procedures. The answer is NOT to put ALL your applications procs on one big DATAPAK (64k or 128k) as you will find that, because of the way that DATAPAKs are accessed, some of your procs will run so slowly as to be almost unuseable. After many trials, I have found that 16k paks are a good compromise to hold regularly used procs, in my case AUTOSCRIBE, FMAN, and a suite of procs to handle the IPSO Group database. The latter files are held on 2 RAMPAKS; (the second one as a security backup) as the information on the file is added to (or amended) almost daily. I keep only one diary (not extensive) in RAM, and must admit to being caught out once or twice when having to do a hard reset occasionally to escape from some diabolical TRAP or other fatal error. How do other members tackle this?

FOR SALE

Organiser II (CM) still under warranty £70.00

Telephone C.PASCOE on 0323 847142

PAUL BROWNING writes:

One thing I find increasingly annoying about the Organiser is the limitations of the standard CALCulator and the difficulties which arise in trying to enhance the CALC mode using the OPL language. In theory, it should be possible to store the calculation which is keyed in in CALC mode in a string variable and so to be able to store files of previous calculations in their "un-evaluated" form. CALC mode, however, limits the user to just storing the answer, which is an incredible shame! What is REALLY needed is a program written in machine code which will "evaluate" a string using the software which must already exist in ROM and return the answer to a numerical variable. I have already written such a program in OPL starting from first principles, but it does not use the ROM software, it's too slow, and, to be honest, I'm not sure that it gives the correct answer in every case! There MUST be an easier and better way, but I don't know how. Just to show what I mean, a word called EVAL could be created as part of the OPL language, so that a program like this could be written.

```

imposibl:
LOCAL x$(255),y
x$="2.56*25,23**4.5/25+(45.23+2.5163)"
y=EVAL(x$)
PRINT y

```

The routine, if it were possible to write it, would display the number 2.084596032E+05, and of course we could store x\$ in a file for later amendment or recall. It's so frustrating, not to be able to do this. Can you, or anybody else help?

JOHN SPILLETT writes:

My immediate problem is to link my Organiser to an OLYMPIA CARRERA TYPEWRITER. The interface for the typewriter is PARALLEL whereas the Comms Link/RS 232 on the Organiser is SERIAL. The Centronics box is £100. Do you know of anybody who has successfully linked the two machines. Also has any member got a surplus/unwanted Comms Link for sale? (I presume the one previously advertised has been sold Ed.)

BATTERY POWER CONSUMPTION

Several members have written in pointing out an error in information previously published about the power consumption when the Organiser obeys an OFF command. They all confirm something which I had already verified - that the SLEEP mode on the Organiser USES EXACTLY THE SAME POWER as a normal switch off from the top-level menu. This is good news for anyone using any procedure which contains the OPL command OFF, as there is no need to fear that the battery will expire while the Organiser is waiting to be "awakened" by the next switch-on. Thanks to those members who pointed out this error.

COMPETITION WINNER

The winning entry in my little competition to improve my one line word-processor is as follows:

```

t:
DO
LPRINT GET$
UNTIL PEEKB($20B0)=1 OR (PEEKB($75)=0 AND PEEKB($74)=0)

```

The fantastic prize (an Organiser padded case) goes to Steve J.T. Knight. Sorry about the delay, but I was underwhelmed by the almost total lack of interest shown. Thanks to the other entrants for their efforts. It'll be a cold day before we have any more competitions (I can vouch for that!)

PROGS & PROCS

A mixed bag of procs this month, starting off with a contraversial one from ERNIE BOKKELKAMP (South Africa) to remove the OFF command from the top level menu. I would recommend that you SAVE any precious data you may have in RAM before trying this one. Not that it will crash deliberately, but you never know where you will go if you get a POKE wrong.

The next one, DIRFIND from DAVID MITCHELL-GOGAY (Horsted Keynes) enables diary details to be displayed in chronological sequence based upon the SEARCH parameters entered, which are in the form DDMMYY. The search can be limited to a particular day of the month, any particular month, any particular year, combined in any permutation. To select every occurrence of a day or month or year zeros are entered. Thus 001100 displays all entries for November in the diary. 010087 displays all entries for the first of the month in 1987.

As the program stands the last selected entry is displayed before the program terminates.

Finally, as it's Christmas, a little game called MEMORY, from GLYN POLLINGTON (Woolacombe), to help you recover after an overdose of Christmas Pud!

A REMINDER FOR SUBMISSIONS TO PROCS & PROGS

Whilst I recognise that not all members have access to printers, may I appeal for future contributors to submit any material for this section in the form of a PRINTOUT from the actual working program, or, better still, on a DATAPAK. For anyone who is rather afraid of sending me a pak by post, I am prepared to pay return postage (recorded delivery) and even ERASE the pak for you before returning it, if you so wish. I have a number of interesting programs which have already been submitted, either written out (some practically illegible) or listed in such a way as to be uncopyable. Please remember that I can only handle the shortest routines, if I have to retype them and try them out in my Organiser. Unfortunately, I have been unable to acknowledge all material submitted, so to those who are still waiting to see their programs printed my apologies.

AUTOEXEC ???

According to the PSION manual pages 65 and 66 you can not remove the OFF item from the main menu and you can not replace a standard item with your own OPL procedure. This is not true. The procedure OFFOPL modifies the main menu item OFF to call a procedure called OFF. This is done by changing a two byte address field in the main menu to zero, which indicates to the organiser that a procedure has to be called. Normally the rom address of the standard item will be in this field.

First you have to write a procedure called OFF which has to execute the OFF command (Otherwise you will not be able to switch your Organiser off). Here is an example:

```
OFF:
CLS :OFF :PRINT "Organiser II";CHR$(15);"Model XP"; :GET
```

You can run this program after you have translated it, but it will not run from the main menu. For this you have to enter and run the procedure OFFOPL. When you run the procedure it will display an address which you should keep for later, in case you would like to remove the OFF procedure from the main menu.

```
OFFOPL:
LOCAL Adr%,Char%,Last%
Adr%=PEEKW($2002) :Char%=0
DO
  IF PEEKB(Adr%)=0                : REM examine length byte
    BREAK                        : REM if zero then end of menu
  ENDIF
  Last%=Adr%                      : REM keep last address examined
  Adr%=Adr%+PEEKB(Adr%)+3        : REM increase address pointer
UNTIL 0
IF PEEKB(Last%)=3                : REM Check for correct length
  IF (PEEKB(Last%+1)=%0 AND PEEKB(Last%+2)=%F AND PEEKB(Last%+3)=%F)
    PRINT CHR$(12);"Please Note";CHR$(15);"OFF ADR: ";PEEKW(Last%+4) :GET
    PRINT "Change "+CHR$(3F)+" : ";
    IF GET$="Y"
      POKEW Last%+4,0
    ELSE
      PRINT CHR$(10);"Reset "+CHR$(3F)+" : ";
      IF GET$="Y"
        PRINT CHR$(10);"Address: ";CHR$(3F);
        INPUT Adr%
        PRINT CHR$(10);"Sure: ";Adr%;CHR$(3F);
        IF GET$="Y"
          POKEW Last%+4,Adr%
        ENDIF
      ENDIF
    ENDIF
  ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
```

All prompts must be acknowledged with an uppercase Y.

After running this procedure the OFF procedure will be called instead of the normal OFF menu item. This gives you the chance to insert a procedure to called everytime the machine is switched on. I use it to start my password routine and to scan the A:MAIN file for any records marked \$RMD, which are reminders.

Another possibility is to store an identifier record on your data packs which can then be displayed, to show which packs are mounted.

```

|dirfind:
LOCAL b%,i$(6),d%,m%,y%,dy%,mn%,yr%,mm%,hh%,l%,t$(255,1),tc%,ts$(100)
b%=PEEKW($2004)
KSTAT 4 :CLS :PRINT "Enter DDMMYY"
INPUT i$
d%=VAL(LEFT$(i$,2))
m%=VAL(MID$(i$,3,2))
y%=VAL(RIGHT$(i$,2))
A::
l%=PEEK(b%)
B::
IF l%=0
  CLS :PRINT "  END OF DIARY" :GET
  RETURN
ENDIF
dy%=(PEEK(b%+3)+1)
mn%=(PEEK(b%+2)+1)
yr%=(PEEK(b%+1))
mm%=(PEEK(b%+5))
hh%=(PEEK(b%+4))
b%=b%+7
IF y%<yr% AND y%<>0
  l%=0
  GOTO B::
ENDIF
IF (y%=yr% AND m%<mn% .AND m%<>0)
  l%=0
  GOTO B::
ENDIF
IF (y%=yr% AND m%=mn% AND d%<dy% AND d%<>0)
  l%=0
  GOTO B::
ENDIF
IF (d%=dy% OR d%=0) AND (m%=mn% OR m%=0) AND (y%=yr% OR y%=0)
CLS :PRINT dy%;" / ";mn%;" / ";yr%;" " ;hh%;" ":" ;
IF mm%=0
  PRINT "00"
ELSE PRINT "30"
ENDIF
tc%=1
DO
  t$(tc%)=CHR$(PEEK(b%))
  tc%=tc%+1 :l%=l%-1 :b%=b%+1
UNTIL l%=0
l%=tc% :tc%=1 :ts$=""
DO
  ts$=ts$+t$(tc%) :tc%=tc%+1
UNTIL tc%=1%
VIEW (2,ts$)
GOTO A::
ENDIF
DO
  l%=l%-1 :b%=b%+1
UNTIL l%=0
GOTO A::

```

```

memory:
GLOBAL lev%,a%,b%,c%,d%,x%
GLOBAL p$(16),q$(16),r$(16)
LOCAL a$(1)
PRINT " MEMORY GAME"
START::
PRINT " LEVEL 1-9"
KSTAT 3
a$=GET$
IF a$=CHR$(1) :RETURN :ENDIF
lev%=VAL(a$)+9
KSTAT 1
SETUP:
PRINT "CONGRATULATIONS";CHR$(EE);
PRINT " ANOTHER GAME";CHR$(63)
a$=GET$
IF a$<>"Y" :RETURN :ENDIF
CLS
a$=""
p$=""
q$=""
r$=""
GOTO START::

setup:
LOCAL a
b%=1
c%=2
x%=1
CLS
DO
a=RND*26
a%=INT(a)+65
p$=p$+CHR$(a%)
x%=x%+1
UNTIL x%=17
DO
q$=MID$(p$,b%,1)
r$=LEFT$(p$,b%)
d%=ASC(q$)
BEEP 200,((d%-60)*20)
PRINT q$;
PAUSE 18-b%
q$=""
b%=b%+1
UNTIL b%=c%
b%=1
CLS
INKEY:
c%=c%+1
UNTIL c%=lev%

inkey:
LOCAL a$(16),b$(1),t$(16),e%
DO
b$=GET$
e%=ASC(b$)
PRINT b$;
BEEP 200,((e%-60)*20)
a$=a$+b$
t$=LEFT$(r$, (LEN(a$)))
IF a$<>t$ :GOTO END:: :ENDIF
UNTIL LEN(a$)=LEN(r$)
CLS
a$=""
PAUSE 10
RETURN
END::
CLS
PRINT "Sorry, wrong"
PRINT r$
GET
CLS
PRINT "ANOTHER TRY";CHR$(63)
b$=GET$
IF b$="Y"
p$=""
q$=""
r$=""
SETUP::
ENDIF
STOP

```

IPSO FACTO

NEWSLETTER OF THE INTERNATIONAL PSION ORGANISER USER GROUP

VOLUME 1 * * * NUMBER 11 * * * JANUARY 1988

Editorial Address: Mike O'Regan
130 Stapleford Lane, Beeston, Nottingham, NG9 6GB
Telecom Gold (Microlink) Postbox 72:MAG11839

Editorial

With this issue of the newsletter I welcome our first member on the mainland of the United States, Mr Paul Lichtman from California. Let's hope that he spreads the Organiser "gospel" so that we get a few more members from America. While I am on the subject of membership numbers, it looks as if IPSO is about to have a considerable boost after our mention in PSION NEWS (I am told, although I haven't yet seen a copy). The end of our first year (March-February) should see the membership at more than 300 for the first time.

For all those who have enquired and been most patient, I can say that I now have a copy of the Technical Reference Manual ON DISK. This is IBM format and I have managed to make an excellent printout from the files. Psion have agreed that I can release "bits" of the manual to members, on the strict understanding that they will NOT ANSWER ANY QUERIES RAISED. Nor will they be responsible for any errors. Any members who are interested should phone me (or write) for further details.

With the next newsletter, I hope to give a definite date for our "get-together". The consensus at the moment is for a SATURDAY (some members expressing a wish to bring their wives along for a wild shopping spree in Nottingham, while we conduct the serious business - apologies to our lady members). The venue I have selected has a good car park and a regular bus service to the city centre. We will probably assemble between 9-30 and 10 a.m. for coffee, have a buffet lunch, and wind up the proceedings at 4 to 5 p.m.

Contents

102	Editorial	103	IPSO REVIEW
104	File Store Structure	107	Progs & Procs

IPSO REVIEW
THE FINGER ORGANISER (Harvester Information Systems Ltd)
Reviewed by Toni Fine

As a confirmed and devoted Organiser user, I must confess to being more than a little skeptical about Harvester's FINGER ORGANISER. After all, although the Psion keyboard takes a bit of familiarisation, I was managing to "hunt and peck" quite adequately and didn't see the need for a typing tutor. However, Harvester kindly supplied me with a review model since in my job as an Occupational Therapist I thought it might be a useful tool in assessing a patient's ability to use the Organiser at all, and of course, when it arrived, I tried it myself. To my surprise, I found it quite addictive and, more importantly, feel that it is indeed useful as means of improving AND maintaining improvement in Organiser speed, dexterity, and accuracy.

Basically, the package consists of a datapak containing a series of graded exercises and a small manual with charts of recommended finger placements. The exercises range from Level 1, which has a target speed of 20 words per minute, to Level 5, which is 40 words per minute, and each level has many exercises within it. On the top screen line letters or words are presented and the user must "echo" them as accurately and as quickly as possible on the line below. There are two variables related to success: speed and number of mistakes, and upon the completion of each exercise you are told your success or lack of it on both scales. If the exercise is completed within the target time and with no more than the allowed number of mistakes, then the user proceeds to the next exercise. If not, he must repeat the exercise until he does satisfy those criteria.

It is surprisingly addictive and I was amazed at how inordinately pleased I felt with myself when I did achieve the correct time and how I actually resorted to "cheating" i.e. using the "wrong" fingers when I got stuck and had to repeat a particularly difficult exercise several times. Since the sentences are quite amusing, one is eager to progress.

Because the sound cues can be turned on or off, it is easy and pleasant to practice in public on trains, tubes, planes, and quite absorbing, but the thing that pleases me most is that, after a few weeks' regular use, the familiarity that one acquires really does remain and the general speed and dexterity is still evident months later, despite only occasional exercise. The element of a game, i.e. trying to beat the machine, is quite motivating and the actual practical use for future Organiser skill make it a package I recommend.

File Store Structure by David Chastney-Parr

What follows is the structure of the files and how they are stored in the Organiser RAM. All data files have a header of the following format.

File separator.
Data file identifier.
File name of 8 bytes padded with spaces
File identifier \$90 to \$FE

So for the MAIN file; the header will be:

\$09 - Separator
\$81 - Data file identifier
\$4D 41 49 4E 20 20 20 20 - Name
\$90 - MAIN file identifier

Other data files, when they are added are given the first available file identifier starting with \$91. The data records are non-contiguous and are added to the end of the files store area. When records or files are ammended or deleted the remaining files are moved down, overwriting the discarded data.

The format of each data record is a single byte indicating the record length, followed by a single byte corresponding to the file identifier, followed by the actual record. Fields within the record are separated by a \$09 byte. The end of the RAM file store area is indicated by a two \$FF bytes. These can easily be located, using a program such as MEMORY: on page 9, or MZAP: on page 84, by first saving a known record such as "END" in the A:MAIN file.

The final bytes will then be
\$03 - Record length
\$90 - MAIN file identifier
\$45 4E 44 - Actual record
\$FF FF - File area terminator

Other file types, are saved DIARY files, PROCEDURE files and COMMS SETUP files. They all have similar file headers which are followed immediately by their associated information.

File separator.
File identifier.
 where \$82 = Saved DIARY file.
 \$83 = PROCEDURE file.
 \$84 = COMMS SETUP file.
File name of 8 bytes padded with spaces
\$00 02 80
Size of file information, 2 bytes
Then follows the actual file information

The format of a saved DIARY file is the same as the current DIARY.

The format of a PROCEDURE file is the translated code, (if saved), followed by the OPL source, (if saved).

The format of a saved COMMS SETUP file is always the same, with a fixed length. For a COMMS LINK file the length is 27 bytes.

Byte 1 is the BAUD rate where

\$00 = 50 \$02 = 110 \$04 = 300 \$06 = 1200 \$08 = 4800
\$01 = 75 \$03 = 150 \$05 = 600 \$07 = 2400 \$09 = 9600 baud

Byte 2 is the PARITY where

\$00 = NONE \$02 = EVEN \$04 = SPACE
\$01 = ODD \$03 = MARK

Byte 3 is the number of BITS where

\$00 = 7 \$01 = 8

Byte 4 is the number of STOP bits where

\$00 = 1 \$01 = 2

Byte 5 is the HANDshake where

\$00 = NONE \$04 = DTR
\$01 = XON \$05 = XON/DTR
\$02 = RTS \$06 = RTS/DTR
\$03 = XON/RTS \$07 = ALL

Byte 6 is the PROTOCOL where

\$00 = NONE \$01 = XMODEM \$02 = PSION

Byte 7 is ECHO where

\$00 = HOST \$01 = LOCAL

Byte 8 is the WIDTH, the number of characters sent before a carriage return is sent. It can be set to any value upto 250.

\$00 = no carriage return sent
\$FA = maximum

Byte 9 is TIMEOUT where

\$00 = no timeout
\$FF = maximum

The final bytes in the file are in sets of three characters where the first byte of each group indicates the number of characters used in the group; either zero, one, or two.

Bytes 10 to 12 are for REOL.

Bytes 13 to 15 are for REOF.

Bytes 16 to 18 are for RTRN.

Bytes 19 to 21 are for TEOL.

Bytes 22 to 24 are for TEOF.

Bytes 25 to 27 are for TTRN.

The Psion Top level menu is pointed to by \$2000/1. The start of the current DIARY is pointed to by \$2004/5. The start of the 10 bytes before the file store area is pointed to by \$2006/7.

Each procedure name in the Top level menu is preceded by the name length and followed by the entry point. The entry points are different for the various models but the structure is the same. Your own procedure names inserted into the top level menu have two \$00 bytes after the name.

The standard set are:-

	32XP	CM
\$04 FIND	\$E639	\$E326
\$04 SAVE	\$E636	\$E323
\$05 DIARY	\$A63D	\$A5FA
\$04 CALC	\$CD06	\$CB66
\$04 PROG	\$C6CE	\$C52E
\$05 ERASE	\$E632	\$E31F
\$04 TIME	\$E88F	\$E55C
\$04 INFO	\$E757	\$Eg4b
\$05 ALARM	\$A266	\$A223
\$04 COPY	\$E3B9	\$E0A6
\$05 RESET	\$E43D	\$E12A
\$03 OFF	\$80FC	\$80FC
\$00	end of MENU.	

If the COMMS LINK is inserted and initialised, then the entry COMMS, normally goes between RESET and OFF with an entry address of \$0B2D. The Concise Oxford Spelling Checker is automatically inserted as DICT just before OFF and has an entry address of \$0446. Interestingly, if these names are inserted elsewhere within the top level menu and an attempt is made to action them; then you get, for example, the message "MISSING PROC DICT". Pressing the ON/CLEAR button twice also replaces the name in its correct place, without having to delete and insert. If menu changed within the Psion top level menu then the whole of the file store area and the current DIARY is moved as well.

Current DIARY entries are stored in chronological order and come immediately after the TOP level menu. The end of the current DIARY is flagged with a single \$00 byte. There is then a gap of 10 bytes before the start of the file store area.

Machine Code Series

I must apologise for a short "break in transmission" in Les Ball's Machine Code Series. This is caused by the necessity to clear the printing of the whole of the Organiser's Microprocessor Instruction Set, which we hope to obtain from Hitachi in time for the next newsletter.

Programs & Procedures

RMENU by Richard Marsh

Function: Similar to MENU but allows list of options to start at varying places according to the value of parameter sn%. Could be used in any situation where it is convenient to start the menu list in varying places, e.g. for date editing or input where pressing EXE will take you to the next field.

Restrictions: None over and above those of MENU. Syntax is the same with ',' acting as item separator and no restriction on variable item lengths.

Parameters: sn% - position of start item in menu list.
s\$ - menu list
Returns: m% - menu item number selected
Errors: reasonable error-proof - sn% outside range of possibilities will default to starting menu with first item

```
RMENU:(sn%,m$)
LOCAL m%,lm%,n%,s%,ls%,sp%
lm%=LEN(n$)
DO
  n%=n%+1
  s%=s%+ls%
  ls%=s%
  IF n%=sn%
    sp%=s%
  ENDIF
  s%=LOC(RIGHT$(m$,lm%-s%),",")
UNTIL s%=0
IF sp%=0 OR sn%<2
  m%=MENU(m$)
ELSE
  m%=MENU(RIGHT$(m$,lm%-sp%)+", "+LEFT$(m$,sp%-1))
  IF m%>0
    m%=m%+sn%-1
    IF m%>n%
      m%=m%-n%
    ENDIF
  ENDIF
ENDIF
RETURN m%
```

ROCKWELL by Simon Bates

This little game has practical value (keyboard speed skills) as well as being an entertainment. It could be a valuable follow up to a session on FINGER ORGANISER (see review section)

```
ROCKWELL:
GLOBAL inc%,le%,c$(1),text$(16),move%,a$(1)
DO
CLS
KSTAT 3
PRINT "Enter level"
```

```

PRINT "(1-9)"
a$=GET$
le%=LOC("123456789",a$)
UNTIL le%
KSTAT 1
DO
inc%=le%*9-1
TEXT:
UNTIL LEN(text$)=5
start::
text:
a$=KEY$
IF a$=LEFT$(text$<1)
move%=move%+1
IF LEN(text$)=1
CLS
PRINT " Well done, in"
PRINT " ";move%;" moves"
a$=GET$
IF a$=CHR$(13)
rockwell:
ENDIF
STOP
ENDIF
text$=RIGHT$(text$,LEN(text$)-1)
CLS
PRINT " moves:";move%
AT 17-LEN(text$),2
PRINT text$
ENDIF
GOTO start::

text:
inc%=inc%+1
IF inc%=9*le%
inc%=0
c$=CHR$(RND*26+65)
text$=text$+c$
BEEP 100,LEN(text$)*20
CLS
PRINT " moves:";move%
AT 17-LEN(text$),2
PRINT text$
IF LEN(text$)=16
CLS
PRINT "After ";move%;" moves"
PRINT "you have FAILED!"
a$=GET$
IF a$=CHR$(13)
rockwell:
ENDIF
STOP
ENDIF
ENDIF
RETURN

```

DATAPAK JUNGLE by Ernie Bokkelkamp

I thought that owning more than one datapak would be a luxury, seeing the price of the units, but this has changed very quickly and I am now the owner of two datapaks and one program pak.

This can cause some problems when you try to run procedures which require a particular datapak to be mounted to be able to access data files.

The procedure MARKPACK is used to write an identification file called "PACKID" into RAM or datapak and the procedure MOUNT is used to find a datapak or if not available to prompt the user to mount it in any slot.

The MOUNT procedure can be used in any of your procedures as shown in the example.

```
MARKPACK:
LOCAL d$(1),f$(10),id$(6),c%
PRINT CHR$(12):"Markpack"
DO
AT 1,2
PRINT "Device: ";
TRAP INPUT d$
IF ERR
RETURN
ENDIF
UNTIL (d$>="A" AND d$<="C")
IF EXIST(d$+":PACKID")
OPEN d$+":PACKID",A,t$
AT 1,2
PRINT CHR$(15);d$+": "+a.t$
CLOSE
GET
ELSE
PRINT CHR$(12);"NO PACKID"
AT 1,2
PRINT "ID: ";
TPAP INPUT id$
IF ERR
RETURN
ENDIF
CREATE d$+":PACKID",a,t$
a.t$=id$
APPEND
CLOSE
ENDIF
```

```

MOUNT:(id$)
LOCAL d%,s$(6)
start::
PRINT "Busy...";
d%=%a
DO
IF EXIST(CHR$(d%)+":PACKID",a,t$)
s%=a.t$
CLOSE
IF s%=id$
RETURN d%
ENDIF
ENDIF
d%=d%+1
UNTIL d%>%c
PRINT CHR$(12);"Mount request"
PRINT CHR$(15);"Pack: ";d$;
BEEP 100,100
IF GET=1
STOP
ENDIF
CLS
GOTO start::

```

```

EXAMPLE:
LOCAL d%,f$(10)
d%=mount:("RAM")      request pack "RAM" to be mounted
IF d%=0 :STOP :ENDIF  if ESCAPE
f%=CHR$(d%)+":MAIN"   else create file name
OPEN f$,a,text$
CLOSE
RETURN

```

MORE NOTES ON PROGRAM SUBMISSIONS

When I asked for program/procedure submissions to be preferably on datapak, I didn't realise that so few people actually owned datapaks, let alone have one spare to use for submitting contributions to this newsletter. So, whilst such submissions will still be more than welcome - especially if the programs are long and involved - I want to make it clear that I am still prepared to accept material on other media. I own a BBC Master 128 and have reasonable access to IBM clones, so I am able to read floppy discs prepared on these machines. Failing that, a good clear PRINTOUT is quite acceptable, but please try to print on A4 paper, with at least a ONE INCH MARGIN on the LEFT EDGE of the paper. I must apologise for the fact that one or two pages of recent newsletters managed to sneak through with practically no left-hand margin, and some of the text was obscured by the punched holes. If anyone has had trouble with this please let me know, and I will replace the affected sheets.

IPSO FACTO

NEWSLETTER OF THE INTERNATIONAL PSION ORGANISER USER GROUP

VOLUME 1 * * * NUMBER 12 * * * FEBRUARY 1988

Editorial Address: Mike O'Regan
130 Stapleford Lane, Beeston, Nottingham, NG9 6GB
Telecom Gold (Microlink) Postbox 72:MAG11839

Editorial

This issue is, of course, the last of Volume 1, marking as it does the end of the first year of IPSO. Before I say anything else, I would like to take this opportunity to thank you all for your support during the formative months and to say a special "thank you" to the following members who have contributed material for this newsletter:

Les BALL, Simon BATES, Ernie BOKKELKAMP, A BOULDEN, Paul BROWNING, Carl BURGESS, Will CHAPMAN, David CHASTNEY-PARR, John CROOK, Toni FINE, Peter JOHNSON, Steve J.T. KNIGHT, Neville LAST, R MACKAY, Richard MARSH, David MITCHELL-GOGAY, Simon MOSS, Mike NASH, Mark NEEDHAM, Adrian PEGG, Glyn POLLINGTON, Harold RAMSEIER, Alan RICHEY, B SAUNDERS, Barry THOMAS, Stuart TIRSUN, Ian WALKER, and Simon WEBB. I hope I haven't missed anyone and, at the same time would like to thank other members whose contributions have not yet been published.

In the next Volume I will be featuring a new series of articles for COMPLETE BEGINNERS (by popular request!), which will feature hints and tips on making the most of the Organiser's built-in facilities, as well as simple OPL programming. There has been a good response to Les BALL's machine-code series, and Les tells me that he hopes to feature some very useful routines such as DIRECTORY PRINTOUTS, PRINTING CALCULATOR, accessing top-level Menu items from within OPL procedures etc.

Following the mention of IPSO in PSION NEWS, I am having a sudden influx of new members (between 25 and 50 PER WEEK! at the moment) so we can look forward to lots of interesting and useful articles and procedures in future. Keep 'em coming!

More news of IPSOMEET (our get-together in April) next month.

Finally, don't forget your MEMBERSHIP RENEWAL on the enclosed form.

Contents

112 Progs & Procs 115 Machine Code - Part III

117 Hitachi Processor Instruction Set

Programs & Procedures

NWEEK by A. Boulden

This routine for calculating the day of the week involves working out the number of days elapsed since the start of the Julian Calendar (BC 4713). I think it is an improvement on the date storage routines listed in Issue 3 of IPSO FACTO as it will work for any date from The Great Flood to Eternity!

```

NWEEK:
LOCAL dy%,mh%,yr%,a,b,c,d,f,d$(21)
d$="SunMonTueWedThuFriSat"
PRINT"Day ";:INPUT dy%
PRINT"Month ";:INPUT mh%
PRINT"Year ";:INPUT yr%
a=1+INT((mh%-15)/12)
b=INTF(1461*(yr%+4800+a)/4)
c=INT(367*(mh%-2-12*a)/12)
d=(yr%+4900+a)/100
f=dy%-2447095+b+c-INT(3*INT(d)/4)
CLS
PRINT dy%;"-";mh%;"-";yr%
PRINT MID$(d$, (1+f-INTF(f/7)*7)*3-2,3)
GET
RETURN
    
```

PDIARY by A. Boulden

I have modified and improved (well I think so!) the various Diary printout procedures that have featured in IPSO FACTO and Psion's own newsletter. It prints the month by name as well as by date. I've deliberately left out the year = I can't plan more than one ahead! I also think I've ironed out a few of the bugs that appeared in earlier versions.

```

PDIARY:
local i,r,u,v,g,h,b%,l%,c%,e%,m$(36),d$(21)
m$="JanFebMarAprMayJunJulAugSepOctNovDec"
d$="SunMonTueWedThuFriSat"
b%=peekw($2004)
PRINT "Print to end of month <1-12> ";
TRAP INPUT e%
IF ERR > 0 :RETURN :ENDIF
a::
l%=PEEKb(b%)
IF l%=0
end::
CLS
PRINT " End of Diary"
GET
RETURN
ENDIF
IF PEEKb(b%+2)<>e%
i=1900+PEEKb(b%+1)
r=1+INT(((PEEKb(b%+2)+1)-15)/12)
u=INTF(1461*(i+4800+r)/4)
v=INT(367*((PEEKb(b%+2)+1)-2-12*r)/12)
g=(i+4900+r)/100
h=(PEEKb(b%+3)+1)-2447095+u+v-INT(3*INT(g)/4)
h=1+h-INT(h/7)*7
LPRINT mid$(D$, (h*3)-2,3);" ";
    
```

```

IF PEEKB(b%+3)+1<10 :LPRINT "0"; :ENDIF
LPRINT (PEEKB(b%+3)+1);" ";
LPRINT MID$(m$,((PEEKB(b%+2)+1)*3)-2,3);" ";
IF PEEKB(b%+4)<10 :LPRINT "0"; :ENDIF
LPRINT PEEKB(b%+4);";";
IF PEEKB(b%+5)=0
LPRINT "00 ";
ELSE
LPRINT "30 ";
ENDIF
c%=PEEKB(b%)
b%=b%+7
LPRINT " ";
DO
LPRINT CHR$(PEEKB(b%));
c%=c%-1
b%=b%+1
UNTIL c%=0
LPRINT
GOTO a::
ENDIF
GOTO end::

```

Example Printout:

```

Sat 23 Jan 14:30 TAKE CAT TO VETS
Mon 08 Feb 15:00 CAR SERVICE
Sat 19 Mar 13:30 LUNCH JOHN
Wed 13 Apr 09:30 DENTIST

```

RENAME: and POUNDS: by Alan Richey

Both the procedures that follow were written to solve apparent drawbacks in the ORGANISER software. I suspect there are proper solutions. if so I would be grateful for the answer. But if not, I offer the following solutions.

Problem 1. After writing a procedure there did not seem to be a way of changing the name.

RENAME

```

LOCAL P%,X%,A$(16),B$(8)
CLS
PRINT"Change Filename From - ";
INPUT A$                               : Get present Filename
IF A$="" :RETURN                       : Quit if just 'EXE' is pressed
ENDIF
A$=LEFT$(A$+"",8)                      : Expand to 8 characters
PRINT" To - ";
INPUT B$                                : Do the same for new name
IF B$="" :RETURN
ENDIF
B$=LEFT$(B$+"",8)
P%=$2400                               : Start memory address

```



```

DO
X%=0 : Wordlength counter to zero
DO
IF PEEKB(P%+X%)<>ASC(MID$(A$,X%+1,1)) : Keep going till name found
GOTO a::
ENDIF
X%=X%+1 : Check all 8 characters of name
UNTIL X%=8
X%=0
DO
POKEB P%+X%,ASC(MID$(B$,X%+1,1)) : Replace with new name
X%=X%+1
UNTIL X%=8
CLS
BEEP 200,5000
PRINT"Renamed",B$;"Press SPACE" : Print result
GET
RETURN : and quit
a::
P%=P%+1 : Next memory address
X%=0 : Reset wordlength counter
IF KEY=81:RETURN : Quit if 'Q' pressed
ENDIF : during procedure execution
UNTIL P%=$4000 : Or $6000/$8000
CLS
PRINT"FILE NOT FOUND Press space key" : Print failure
BEEP 500,2000
GET
RETURN : and return

```

When I was writing an Accounts program I could find no way of printing any value between 0 and +/-9999, right justified, to 2 decimal places AND including a '£' sign. All the OPL commands that would give me 2 decimal places also required the length of the string to be specified, which would obviously vary with the value. My solution is as follows:

```

POUNDS

POUNDS:(P$,V)
LOCAL Q,Q$(9)
POKEB $180,64:POKEB $181,6 : Define '£' as in manual
POKEB $181,9:POKEB $181,9
POKEB $181,12:POKEB $181,8
POKEB $181,24:POKEB $181,31
POKEB $181,0
Q=ABS(V)
Q$=CHR$(0)+FIX$(Q,2,LOG(Q)+4)
IF V<0:Q$="-"+Q$
ELSE Q$=" "+Q$
ENDIF
PRINT P$+REPT$(" ",16-LEN(Q$+P$))+Q$
RETURN

```

Procedure called with 'P\$' as the Account name and 'V' as the value, ie. POUNDS:("Current",123.45)
 (Note that it has a slight 'bug' in that it cannot cope with absolute values of less than 1)

Ideally we should now take a detailed look at the INSTRUCTION SET of the 6303. Unfortunately this would take a good deal of time and space. We are therefore reproducing the instruction set as contained in the 6303 Data Sheet issued by Hitachi.

To those who have experience of machine code on other processors, this will be self-explanatory, but here are a few notes for beginners. The heading on the left-hand side of the page tells us the kind of operation the instruction is going to perform i.e. arithmetic, logical, data manipulation etc. The first heading of the table is PNEUMONICS, a series of letters designed to help us remember the instruction and the format in which we would enter this into an Assembler. The next five columns, which each have three headings, are termed addressing modes. These describe the way in which the operation is handled. The modes are Immediate, Direct, Indexed, Extended and Implied.

Let us suppose we wish to put a number into a particular place. Let's deal with the simplest form of this i.e. a byte. If we wish to load a byte, let us say into the A Register. Look up the instruction LDA A, which means Load Accumulator A. Under the addressing mode Immediate you will see the HEX number 86. This is the code which tells the processor that the number immediately following it is to be put into the Accumulator A (or A Register). Therefore if we wish to put the number 1 into the A Register, the instructions would be: 86 1. It may be that we do not wish to put a number into this register immediately, but want to transfer a number contained in another memory location to that register. If that number is lower than 255 decimal (FF hex) our four-figure address would be 00FF maximum. This portion of memory is known as ZERO PAGE. To make it easier, the processor handles this in what is known as the Direct Mode. We do not, therefore, need to put in 00FF, or, say, 0010. As long as the processor has been alerted to the fact that we are dealing with zero page or direct mode we need only put in the relevant byte (FF or 10 or whatever the zero page location is). So LDA A Direct code would be 96 followed by our zero page location. This would transfer the contents of address xx zero page into the A Register. Note that the contents of the address are put into the register, not the number we have specified, which would have to be done through the Immediate mode.

Of course we may wish to transfer the contents of a higher address to the A Register, let us say 4000 HEX. There are two ways we can do this. If we know we are going to use location 4000, for example, we can use the Extended Mode. LDA Extended is B6 followed by 4000, which gives us a 3 byte instruction. This transfers the contents of location 4000 HEX into the A Register.

There is another way of doing the same thing, by using the Indexed Mode. Here we need to point the X Register at the address 4000. We can do this by loading the Index Register immediately with 4000. If you look at the Instruction Sheet, you will find under LDX Immediate "CE", so LDX Immediate 4000 would be CE 4000. The number 4000 is now in the Index Register. We can now instruct the processor to load into Register A the contents of the memory held in the Index Register, plus a given number called a DISPLACEMENT. Therefore to load the contents of Address 4000 into Register A our displacement would be 0, so LDA A Indexed (Opcode A6) followed by our displacement, in this case 0. Note that the displacement is only one byte, so we have a maximum of FF as a displacement. In other words, if we wish to load the contents of 40FF into the A Register, then A6 FF would do this.

The last addressing mode is Implied. These are 1 byte instructions which have immediate effect. In other words we do not need to add any other instruction to this. If, for example, we wish to clear the A Register, or, if you wish, put a value of 0 into the A Register, then the Implied Mode 4F would do this. To clear the B Register, Implied 5F. The next column of the table describes the action of the instruction and should be reasonably easy to follow.

The last column describes the effects upon the FLAGS, (or Condition Code Register) and may be more difficult to understand, but for now we can leave this and look more closely at these as we use routines.

In each of the addressing mode columns there are 2 other divisions containing numbers. In the column headed by a "tilde" (~) we are given the number of MACHINE CYLES taken to perform the operation. In the column headed by a "hash" we are given the number of BYTES for the instruction. The number of machine cycles is not particularly important for most applications, but where high speed is needed then we would need to take this into account.

For those who really want to get "into" machine code Hitachi produce two books on the 6303, one of which devotes a whole page to each instruction. Another useful book, if you can still get it, is "6800 Assembly Language Programming" by Leventhal. However, fortunately, I have been unable to obtain a copy of this for myself.

It would be useful what kind of routines people have not been able to perform with OPL and are hoping that they can do with machine code. It would also be useful if anyone has routines or ideas if they could send them to me directly, but please say if you do not want these routines published.

(to be continued)

Editor's Note

For obvious reasons, it has been necessary to use four pages of the newsletter to reproduce the Hitachi Instruction Set, without which the above article (and the ones which follow) would be incomprehensible. However, for those of a nervous disposition who are not yet at the stage of machine code, I must apologise for the space I have allocated and hope that it will subsequently be all worth while.

Many thanks to Hitachi Electronic Corp for giving permission to reproduce the Instruction Set (see pages 117 to 120)

Table 14 Accumulator, Memory Manipulation Instructions

Operations	Mnemonic	Addressing Modes															Boolean/ Arithmetic Operation	Condition Code Register					
		IMMED			DIRECT			INDEX			EXTEND			IMPLIED				S	O	Z	V	C	
		OP	~	#	OP	~	#	OP	~	#	OP	~	#	OP	~	#		H	I	N	Z	V	C
Add	ADDA	8B	2	2	9B	3	2	AB	4	2	8B	4	3				$A + M \rightarrow A$	1	0	1	1	1	1
	ADDB	CB	2	2	DB	3	2	EB	4	2	FB	4	3				$B + M \rightarrow B$	1	0	1	1	1	1
Add Double	ADDD	C3	3	3	D3	4	2	E3	5	2	F3	5	3				$A : B + M : M + 1 \rightarrow A : B$	0	0	1	1	1	1
Add Accumulators	ABA													1B	1	1	$A + B \rightarrow A$	1	0	1	1	1	1
Add With Carry	ADCA	89	2	2	99	3	2	A9	4	2	B9	4	3				$A + M + C \rightarrow A$	1	0	1	1	1	1
	ADCB	C9	2	2	D9	3	2	E9	4	2	F9	4	3				$B + M + C \rightarrow B$	1	0	1	1	1	1
AND	ANDA	84	2	2	94	3	2	A4	4	2	B4	4	3				$A \cdot M \rightarrow A$	0	0	1	1	R	0
	ANOB	C4	2	2	D4	3	2	E4	4	2	F4	4	3				$B \cdot M \rightarrow B$	0	0	1	1	R	0
Bit Test	BIT A	85	2	2	95	3	2	A5	4	2	B5	4	3				$A \cdot M$	0	0	1	1	R	0
	BIT B	C5	2	2	D5	3	2	E5	4	2	F5	4	3				$B \cdot M$	0	0	1	1	R	0
Clear	CLR							6F	5	2	7F	5	3				$00 \rightarrow M$	0	0	R	S	R	R
	CLRA													4F	1	1	$00 \rightarrow A$	0	0	R	S	R	R
	CLRB													5F	1	1	$00 \rightarrow B$	0	0	R	S	R	R
Compare	CMPA	81	2	2	91	3	2	A1	4	2	B1	4	3				$A - M$	0	0	1	1	1	1
	CMPB	C1	2	2	D1	3	2	E1	4	2	F1	4	3				$B - M$	0	0	1	1	1	1
Compare Accumulators	CBA													11	1	1	$A - B$	0	0	1	1	1	1
Complement, 1's	COM							63	6	2	73	6	3				$\bar{M} \rightarrow M$	0	0	1	1	R	S
	COMA													43	1	1	$\bar{A} \rightarrow A$	0	0	1	1	R	S
	COMB													53	1	1	$\bar{B} \rightarrow B$	0	0	1	1	R	S
Complement, 2's (Negate)	NEG							60	6	2	70	6	3				$00 - M \rightarrow M$	0	0	1	1	①	②
	NEGA													40	1	1	$00 - A \rightarrow A$	0	0	1	1	①	②
	NEGB													50	1	1	$00 - B \rightarrow B$	0	0	1	1	①	②
Decimal Adjust, A	DAA													19	2	1	Converts binary add of BCD characters into BCD format	0	0	1	1	1	③
Decrement	DEC							6A	6	2	7A	6	3				$M - 1 \rightarrow M$	0	0	1	1	④	0
	DECA													4A	1	1	$A - 1 \rightarrow A$	0	0	1	1	④	0
	DECB													5A	1	1	$B - 1 \rightarrow B$	0	0	1	1	④	0
Exclusive OR	EORA	88	2	2	98	3	2	A8	4	2	B8	4	3				$A \oplus M \rightarrow A$	0	0	1	1	R	0
	EORB	C8	2	2	D8	3	2	E8	4	2	F8	4	3				$B \oplus M \rightarrow B$	0	0	1	1	R	0
Increment	INC							6C	6	2	7C	6	3				$M + 1 \rightarrow M$	0	0	1	1	⑤	0
	INCA													4C	1	1	$A + 1 \rightarrow A$	0	0	1	1	⑤	0
	INCB													5C	1	1	$B + 1 \rightarrow B$	0	0	1	1	⑤	0
Load Accumulator	LDAA	86	2	2	96	3	2	A6	4	2	B6	4	3				$M \rightarrow A$	0	0	1	1	R	0
	LDAB	C6	2	2	D6	3	2	E6	4	2	F6	4	3				$M \rightarrow B$	0	0	1	1	R	0
Load Double Accumulator	LDD	CC	3	3	DC	4	2	EC	5	2	FC	5	3				$M + 1 - B, M \rightarrow A$	0	0	1	1	R	0
Multiply Unsigned	MUL													3D	7	1	$A \times B \rightarrow A : B$	0	0	0	0	0	⑥
OR, Inclusive	ORAA	8A	2	2	9A	3	2	AA	4	2	BA	4	3				$A + M \rightarrow A$	0	0	1	1	R	0
	ORAB	CA	2	2	DA	3	2	EA	4	2	FA	4	3				$B + M \rightarrow B$	0	0	1	1	R	0
Push Data	PSHA													36	4	1	$A - M_{SP}, SP - 1 \rightarrow SP$	0	0	0	0	0	0
	PSHB													37	4	1	$B - M_{SP}, SP - 1 \rightarrow SP$	0	0	0	0	0	0
Pull Data	PULA													32	3	1	$SP + 1 \rightarrow SP, M_{SP} \rightarrow A$	0	0	0	0	0	0
	PULB													33	3	1	$SP + 1 \rightarrow SP, M_{SP} \rightarrow B$	0	0	0	0	0	0
Rotate Left	ROL							69	6	2	79	6	3				M	0	0	1	1	⑦	1
	ROLA													49	1	1	A	0	0	1	1	⑦	1
	ROLB													59	1	1	B	0	0	1	1	⑦	1
Rotate Right	ROR							66	6	2	76	6	3				M	0	0	1	1	⑧	1
	RORA													46	1	1	A	0	0	1	1	⑧	1
	RORB													56	1	1	B	0	0	1	1	⑧	1

(Note) Condition Code Register will be explained in Note of Table 17.

(continued)

Table 14 Accumulator, Memory Manipulation Instructions

Operations	Mnemonic	Addressing Modes												Boolean/ Arithmetic Operation	Condition Code Register																				
		IMMED			DIRECT			INDEX			EXTEND				IMPLIED			5	4	3	2	1	0												
		OP	~	#	OP	~	#	OP	~	#	OP	~	#		OP	~	#	H	I	N	Z	V	C												
Shift Left Arithmetic	ASL							68	6	2	78	6	3											•	•	•	•	•	•						
	ASLA													48	1	1	A		•	•	•	•	•	•											
	ASLB													58	1	1	B		•	•	•	•	•	•											
Double Shift Left, Arithmetic	ASLD													05	1	1	C		•	•	•	•	•	•											
Shift Right Arithmetic	ASR							67	6	2	77	6	3											•	•	•	•	•	•						
	ASRA													47	1	1	A		•	•	•	•	•	•											
	ASRB													57	1	1	B		•	•	•	•	•	•											
Shift Right Logical	LSR							64	6	2	74	6	3											•	•	R	•	•	•						
	LSRA													44	1	1	A		•	•	R	•	•	•											
	LSRB													54	1	1	B		•	•	R	•	•	•											
Double Shift Right Logical	LSRD													04	1	1	C		•	•	R	•	•	•											
Store Accumulator	STAA							97	3	2	A7	4	2	B7	4	3												•	•	•	•	•	R	•	
	STAB							D7	3	2	E7	4	2	F7	4	3													•	•	•	•	•	R	•
Store Double Accumulator	STD							D0	4	2	E0	5	2	F0	5	3													•	•	•	•	•	R	•
Subtract	SUBA	80	2	2	90	3	2	A0	4	2	B0	4	3																•	•	•	•	•	•	•
	SUBB	C0	2	2	D0	3	2	E0	4	2	F0	4	3																•	•	•	•	•	•	•
Double Subtract	SUBD	B3	3	3	93	4	2	A3	5	2	B3	5	3																•	•	•	•	•	•	•
Subtract Accumulators	SBA													10	1	1													•	•	•	•	•	•	•
Subtract With Carry	SBCA	B2	2	2	92	3	2	A2	4	2	B2	4	3																•	•	•	•	•	•	•
	SBCB	C2	2	2	D2	3	2	E2	4	2	F2	4	3																•	•	•	•	•	•	•
Transfer Accumulators	TAB													16	1	1													•	•	•	•	•	R	•
	TBA													17	1	1													•	•	•	•	•	R	•
Test Zero or Minus	TST							60	4	2	70	4	3																•	•	•	•	•	R	R
	TSTA													40	1	1													•	•	•	•	•	R	R
	TSTB													50	1	1													•	•	•	•	•	R	R
And Immediate	AIM							71	6	3	61	7	3																•	•	•	•	•	R	•
OR Immediate	OIM							72	6	3	62	7	3																•	•	•	•	•	R	•
EOR Immediate	EIM							75	6	3	65	7	3																•	•	•	•	•	R	•
Test Immediate	TIM							78	4	3	68	5	3																•	•	•	•	•	R	•

(Note) Condition Code Register will be explained in Note of Table 17.

Additional Instruction

In addition to the HD6801 instruction set, the HD63701X0 prepares the following new instructions.

AIM..... (M)·(IMM) → (M)

Executes "AND" operation to immediate data and the memory contents and stores its result in the memory.

OIM..... (M) + (IMM) → (M)

Executes "OR" operation to immediate data and the memory contents and stores its result in the memory.

EIM..... (M) ⊕ (IMM) → (M)

Executes "EOR" operation to immediate data and the memory contents and stores its result in the memory.

TIM..... (M) · (IMM)

Executes "AND" operation to immediate data and changes the relative flag of the condition code register.

These are 3-byte instructions; the first byte is op code, the second immediate data and the third address modifier.

XGDX..... (ACCD) → (IX)

Exchanges the contents of accumulator and the index register.

SLP

Goes to the sleep mode. Refer to "LOW POWER DISSIPATION MODE" for more details of the sleep mode.

Table 15 Index Register, Stack Manipulation Instructions

Pointer Operations	Mnemonic	Addressing Modes										Boolean/ Arithmetic Operation	Condition Code Register											
		IMMED		DIRECT		INDEX		EXTEND		IMPLIED			5	4	3	2	1	0						
		OP	~ #	OP	~ #	OP	~ #	OP	~ #	OP	~ #								H	I	N	Z	V	C
Compare Index Reg	CPX	8C	3 3	9C	4 2	AC	5 2	BC	5 3									X - M:M+1	•	•	:	:	:	:
Decrement Index Reg	DEX											09	1 1					X - 1 -> X	•	•	•	•	•	•
Decrement Stack Pntr	DES											34	1 1					SP - 1 -> SP	•	•	•	•	•	•
Increment Index Reg	INX											08	1 1					X + 1 -> X	•	•	•	•	•	•
Increment Stack Pntr	INS											31	1 1					SP + 1 -> SP	•	•	•	•	•	•
Load Index Reg	LDX	CE	3 3	DE	4 2	EE	5 2	FE	5 3									M -> X _H , (M+1) -> X _L	•	•	Ⓢ	:	:	•
Load Stack Pntr	LDS	8E	3 3	9E	4 2	AE	5 2	BE	5 3									M -> SP _H , (M+1) -> SP _L	•	•	Ⓢ	:	:	•
Store Index Reg	STX			DF	4 2	EF	5 2	FF	5 3									X _H -> M, X _L -> (M+1)	•	•	Ⓢ	:	:	•
Store Stack Pntr	STS			9F	4 2	AF	5 2	BF	5 3									SP _H -> M, SP _L -> (M+1)	•	•	Ⓢ	:	:	•
Index Reg -> Stack Pntr	TXS											35	1 1					X - 1 -> SP	•	•	•	•	•	•
Stack Pntr -> Index Reg	TSX											30	1 1					SP + 1 -> X	•	•	•	•	•	•
Add	ABX											3A	1 1					B + X -> X	•	•	•	•	•	•
Push Data	PSHX											3C	5 1					X _L -> M _{sp} , SP - 1 -> SP X _H -> M _{sp} , SP - 1 -> SP	•	•	•	•	•	•
Pull Data	PULX											38	4 1					SP + 1 -> SP, M _{sp} -> X _H SP + 1 -> SP, M _{sp} -> X _L	•	•	•	•	•	•
Exchange	XGDX											18	2 1					ACCD -> IX	•	•	•	•	•	•

(Note) Condition Code Register will be explained in Note of Table 17.

Table 16 Jump, Branch Instruction

Operations	Mnemonic	Addressing Modes										Branch Test	Condition Code Register											
		RELATIVE		DIRECT		INDEX		EXTEND		IMPLIED			5	4	3	2	1	0						
		OP	~ #	OP	~ #	OP	~ #	OP	~ #	OP	~ #								H	I	N	Z	V	C
Branch Always	BRA	20	3 2															None	•	•	•	•	•	•
Branch Never	BRN	21	3 2															None	•	•	•	•	•	•
Branch If Carry Clear	BCC	24	3 2															C = 0	•	•	•	•	•	•
Branch If Carry Set	BCS	25	3 2															C = 1	•	•	•	•	•	•
Branch If = Zero	BEQ	27	3 2															Z = 1	•	•	•	•	•	•
Branch If > Zero	BGE	2C	3 2															N ⊕ V = 0	•	•	•	•	•	•
Branch If > Zero	BGT	2E	3 2															Z + (N ⊕ V) = 0	•	•	•	•	•	•
Branch If Higher	BHI	22	3 2															C + Z = 0	•	•	•	•	•	•
Branch If < Zero	BLE	2F	3 2															Z + (N ⊕ V) = 1	•	•	•	•	•	•
Branch If Lower Or Same	BLS	23	3 2															C + Z = 1	•	•	•	•	•	•
Branch If < Zero	BLT	2D	3 2															N ⊕ V = 1	•	•	•	•	•	•
Branch If Minus	BMI	28	3 2															N = 1	•	•	•	•	•	•
Branch If Not Equal Zero	BNE	26	3 2															Z = 0	•	•	•	•	•	•
Branch If Overflow Clear	BVC	28	3 2															V = 0	•	•	•	•	•	•
Branch If Overflow Set	BVS	29	3 2															V = 1	•	•	•	•	•	•
Branch If Plus	BPL	2A	3 2															N = 0	•	•	•	•	•	•
Branch To Subroutine	BSR	8D	5 2																•	•	•	•	•	•
Jump	JMP					6E	3 2	7E	3 3										•	•	•	•	•	•
Jump To Subroutine	JSR			9D	5 2	AD	5 2	8D	6 3										•	•	•	•	•	•
No Operation	NOP											01	1 1					Advances Prog. Cntr. Only	•	•	•	•	•	•
Return From Interrupt	RTI											3B	10 1						•	•	•	•	•	•
Return From Subroutine	RTS											39	5 1						•	•	•	•	•	•
Software Interrupt	SWI											3F	12 1						•	•	•	•	•	•
Wait for Interrupt*	WAI											3E	9 1						•	•	•	•	•	•
Sleep	SLP											1A	4 1						•	•	•	•	•	•

(Note) * WAI puts R/W high; Address Bus goes to FFFF; Data Bus goes to the three state. Condition Code Register will be explained in Note of Table 17.

Table 17 Condition Code Register Manipulation Instructions

Operations	Mnemonic	Addressing Modes			Boolean Operation	Condition Code Register											
		IMPLIED				5	4	J	2	1	0						
		OP	-	#								H	I	N	Z	V	C
Clear Carry	CLC	0C	1	1	0 → C	•	•	•	•	•	•	•	•	R			
Clear Interrupt Mask	CLI	0E	1	1	0 → I	•	R	•	•	•	•	•	•	•			
Clear Overflow	CLV	0A	1	1	0 → V	•	•	•	•	•	•	R	•	•			
Set Carry	SEC	0D	1	1	1 → C	•	•	•	•	•	•	•	•	S			
Set Interrupt Mask	SEI	0F	1	1	1 → I	•	S	•	•	•	•	•	•	•			
Set Overflow	SEV	0B	1	1	1 → V	•	•	•	•	•	•	S	•	•			
Accumulator A → CCR	TAP	06	1	1	A → CCR	⑩						•	•	•	•	•	•
CCR → Accumulator A	TPA	07	1	1	CCR → A	•	•	•	•	•	•	•	•	•	•		

LEGEND

- OP Operation Code (Hexadecimal)
- ~ Number of MCU Cycles
- M_{SP} Contents of memory location pointed to by Stack Pointer
- # Number of Program Bytes
- + Arithmetic Plus
- Arithmetic Minus
- Boolean AND
- + Boolean Inclusive OR
- ⊕ Boolean Exclusive OR
- M Complement of M
- Transfer into
- 0 Bit = Zero
- 00 Byte = Zero

CONDITION CODE SYMBOLS

- H Half-carry from bit 3 to bit 4
- I Interrupt mask
- N Negative (sign bit)
- Z Zero (byte)
- V Overflow, 2's complement
- C Carry/Borrow from/to bit 7
- R Reset Always
- S Set Always
- ‡ Set if true after test or clear
- Not Affected

(Note) Condition Code Register Notes: (Bit set if test is true and cleared otherwise)

- ① (Bit V) Test: Result = 10000000?
- ② (Bit C) Test: Result ≠ 00000000?
- ③ (Bit C) Test: BCD Character of high-order byte greater than 10? (Not cleared if previously set)
- ④ (Bit V) Test: Operand = 10000000 prior to execution?
- ⑤ (Bit V) Test: Operand = 01111111 prior to execution?
- ⑥ (Bit V) Test: Set equal to N ⊕ C = 1 after the execution of instructions
- ⑦ (Bit N) Test: Result less than zero? (Bit 15=1)
- ⑧ (All Bit) Load Condition Code Register from Stack.
- ⑨ (Bit I) Set when interrupt occurs. If previously set, a Non-Maskable Interrupt is required to exist the wait state.
- ⑩ (All Bit) Set according to the contents of Accumulator A.
- ⑪ (Bit C) Result of Multiplication Bit 7=1? (ACCB)

Table 18 OP-Code Map

OP CODE					ACC A	ACC B	IND	EXT DIR	ACCA or SP				ACCB or X						
	HI	0000	0001	0010	0011	0100	0101	0110	0111	IMM	DIR	IND	EXT	IMM	DIR	IND	EXT		
LO	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F			
0000	0	SBA	BRA	TSX	NEG				SUB				0						
0001	1	NOP	CBA	BRN	INS	AIM				CMP				1					
0010	2	/		BHI	PULA	OIM				SBC				2					
0011	3	/		BLS	PULB	COM				SUBD				ADD					
0100	4	LSRD	/		BCC	DES	LSR				AND				4				
0101	5	ASLD	/		BCS	TXS	EIM				BIT				5				
0110	6	TAP	TAB	BNE	PSHA	ROR				LDA				6					
0111	7	TPA	TBA	BEQ	PSHB	ASR				STA				STA					
1000	8	INX	XGDX	BVC	PULX	ASL				EOR				8					
1001	9	DEX	DAA	BVS	RTS	ROL				ADC				9					
1010	A	CLV	SLP	BPL	ABX	DEC				ORA				A					
1011	B	SEV	ABA	BMI	RTI	TIM				ADD				B					
1100	C	CLC	/		BGE	PSHX	INC				CPX				LDD				
1101	D	SEC	/		BLT	MUL	TST				BSR	JSR				STD			
1110	E	CLI	/		BGT	WAI	JMP				LDS				LDX				
1111	F	SEI	/		BLE	SWI	CLR				STS				STX				
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F		

UNDEFINED OP CODE

* Only each instructions of AIM, OIM, EIM, TIM

Amstrad comms program. . .70
 AUTOEXEC, program. . .99
 AUTOSCRIBE, word-processor. . .71
 Battery consumption. . .97
 Beep Beep. . .14
 Beep parameters. . .23
 Beginners page. . . .2, 12, 24, 56-57, 58
 Bits & Pieces. . .23
 Blunders. . .40
 Books. . . .3, 13
 CARPET, program. . .76
 Cases, Organiser. . .3, 69
 CLICK, program. . .77
 CLIK, program. . .32
 Clock, getting at. . .10, 15
 COLCODES, program. . .29
 Combining procedures. .33
 Comms & RS 232 link. . .73-74
 Comms, modem. . .22
 Comms, Organiser - PCW. .42, 75
 Competitions. . .33, 62, 97
 Contributors, list. . .111
 Damage to Organiser. .69
 Datapak formatting. . .4
 Data storage. . .26
 Date string. . .27
 DELAY, program. . .25
 Design, Organiser. . .24
 Diary printout. . .18
 DIRFIND, program. . .100
 Dolphin speech synthesiser. . .20
 Event timing. . .5
 Feedback. . .52, 67, 72, 96
 Files, Psion. . .23, 65, 104-106
 Finger Organiser Review. . .103
 FNKEYS review. . .82-83
 Format for submissions. .80, 110
 Genealogy & the Organiser. .63-65
 Graphics, UGGs. . .54
 Handbook, Organiser. . .3
 HILO, program. . .57
 Hints & Tips. . .32
 HITACHI processor instruction set. . .117-120
 Ipsomeet. . .68, 71, 92, 102
 KEYBOARD, program. . .15
 LABEL, program. . .14
 Letters. . .4
 Machine code. . .81, 90-91, 93, 115-116
 Mains adaptor. . . .4, 67
 MARKPACK, program. . .109-110
 MEMORY, program. . . .8, 101
 MUDG, program. . .54-55
 MZAP, program. . .84-85
 NOTEPAD, program. . .86-89
 N WEEK, program. . .112
 Occupational therapy & Organiser. . .53
 OFFOPL, program. . .99
 Operating systems, Organiser. . .94-95
 OPL additions. . .32
 PDIARY, program. . .112
 PDIR, program. . .18

Phone review. . .21
PIN, program. . .76
Pocket Modems. .72
Pocket Spreadsheet. . .59-60
Pokes, main Organiser. . .16
PONTOON, program. . .34-40, 47-50
Postal arrangements. . .69
POUNDS, program. . .114
Price reductions. . .66
Printers. . .19
Printer plotter CGP 115. . .33, 68, 77-80
Printer adaptors. . .4, 32
Printer utilities. . .8
Programs on datapak. . .46
PWR, program. . .30
Rampak, 32k. . .66
Regional Groups. . .67
RENAME, program. . .113
Reviews. . .5, 17, 21, 82, 103
RMENU, program. . .107
ROCKWELL, program. . .107-109
SECS, program. . .15
Spectrum Interface. . .67
Speech & the Organiser. . 50
Spell checker erasure. . .31, 41
Storage, datapak. . .3
Technical Reference Manual. . .13, 81, 102
TEXT, program. . .43-46
Textbase, Eidersoft. . .5
TIME, program. . .10
TIMER, program. . .25
TONE, program. . .14
TRIG CALCULATIONS, program. . .85-86
TUNE, program. . .27
Widget Review. . .17
Word-processor, one-line. . .62
WRITER, program. . .62