H.P.D.C.U.G.V. PRESIDENT'S REPORT

— A.G.M. 16th April, 1985

During the past year there has been much activity with six meetings and two workshops. Several fine presentations were made, notably by Barry Hendy on machine tool automation, Chris Burford on his tablet-weighing system, Gordon Roberts on 3D graphics and Mark Michell, who bravely led the Series 80 Assembler workshop day. The other workshop for the year addressed I/O and transfer of data between machines.

Thanks to all the speakers and to all members who contributed snippets at meetings and articles to "Crosstalk" magazine.

"Crosstalk came alive again after doldrums earlier in the year, mainly through the tenacious efforts of the new editor, HP's John Green. Grateful thanks to John and contributors. Yet if this magazine is to flourish as the User-groups' wider medium of communication, it will require more spontaneous jottings from members. (Not forgetting its Australia-wide distribution).

Another medium that appears under-used by members is our library service provided by Tony Stevens comprising an extensive array of manuals, periodicals and a members' equipment list. Thanks again to Tony for being there.

For the comfortable venue used for five of our meetings, special thanks go to Rory Cox and CSIRO (Geomechanics). Thanks also to HP for tonight, the workshops and one other meeting.

My thanks go to the outgoing committee for giving their time and loyal efforts to keep the ball rolling and their additional efforts to keep "Crosstalk" alive. Bernie O'Shannassy retires after three years, two as inaugural president. Special thanks Bernie, who will be missed.

The new committee will be charged with some additional tasks, namely to make the group more visible to potential new members — that is, a membership drive to augment our "hard core" of about 30 and financial list of about 60 — and to seek out and keep providing what members really need.

— **Chris Simpson**

Crosstalk Prizes announced for 1984

At the recent Annual General Meeting of the Melbourne HP/1000 Users Group, two HP Calculators were awarded by the Group for articles and contributions to Crosstalk for the last 12 months. The calculators were purchased by the Group from DataKey who we wish to thank for the generous discount they applied.

The winners were:
Best Article: Gary Grant of CSIRO Animal Health Laboratories for his article in the Jan./Feb. issue on the LAN at ANAHL.

Draw for non-article contributions: A. J. Stevens of Telecom.

Melbourne HP/1000 Users Group Annual Meeting

The Annual General Meeting of the HP/1000 Users Group, Melbourne, was held at Hewlett-Packard on Wednesday the 13th of March. The "new" office bearers are:

President:	John Gwyther of TUSC, (03) 898 0636.
Secretary:	Chris Emery of Department of Aviation, (03) 667 2328.
Treasurer:	Norm Kay of CSIRO, (03) 544 0633.
Auditor:	Bob Rowlands of CSIRO.
Committee:	Bill Jacobs of TUSC. Keith Haberle of CSIRO.

The meeting listened to a very interesting presentation by Eric Lording on the new INTEGRAL Personal Computer followed by some handson time.

The subscription rate for 1986 was set at $12.00.

CONTENTS

HP Computer Museum
www.hpmuseum.net

For research and education purposes only.

Co-ordinator's Comments

I am pleased to say that the response to the plea in 'STOP PRESS' was heard. Many people answered the call and so the May/June issue is more than new products and coming events.

Although this response was good it does not mean that the next issue will have very much without your continuing contributions.

The not so surprising thing about the response is that the articles are interesting and bound to be of use to others. I am sure there are plenty of interesting applications out there that would make great articles.

It should be remembered that Crosstalk is the Journal of Hewlett-Packard Technical Computer User Groups. These groups are Australia-wide and so it is hoped that users from all over Australia will contribute. In this way we can bring the user groups closer together and enhance a sense of community encouraging the exchange of information.

Crosstalk is in a good position to be used as a medium for this exchange and I invite users to make full use of it. If you don't have the answers then try asking the question, some other reader could then help. If one is looking for a particular piece of equipment or some expertise then Crosstalk could help one find it.

Until next issue.

Regards,
John Green,
HP Melbourne

The Australasian Response Centre

Hewlett-Packard has established the Australasian Response Centre in its Melbourne Office. The Centre provides answers to your software application and system software questions on a range of HP systems including Series 80 Desktop Computers.

A toll-free telephone number provides you with access to the centre from anywhere in Australia.

The telephone numbers are:

(008) 33 9863 Toll free outside Melbourne Zone;
895 2863 within Melbourne Zone.

STOLEN

The following items of computer equipment were recently stolen from Henry Drillich.

One HP86A Serial No. 2230A75316
One 82912 Monitor Option 001,
    Serial No. 2232A52820
One HP9135 A Winchester
One HP9121 S

Should anyone locate any of these items, would you please immediately contact Henry Drillich, (03) 857 6816.

H.P.D.C.U.G.V. A.G.M. highly successful

The recent Annual General Meeting of the H.P.D.C.U.G.V. was a highly successful event. Bruce Marsh, General Manager of Hewlett-Packard Limited explained to members the recent re-organization of HP and the benefits that it will bring to users of HP desktop computers.

Following Bruce's presentation there was an exchange of ideas between the user group members and Bruce. This was an opportunity for members to ask questions of Bruce, regarding HP, and to gain relevant feedback.

After the meeting there was a demonstration of the HP Integral PC, the Think-Jet printer, the Laser Jet printer and data transfer using a Modem.

Unfortunately the turnout of members was not as good as expected which was disappointing in light of the success of the meeting. It is hoped that future meetings attract a greater number of members because of the information to be gained from these meetings.

Overall, the meeting was a tremendous success, enjoyed by all those who attended.

NEW PRODUCTS

HP Series 80 is alive and well

Despite recent rumours, the HP Series 80 product line will not be obsoleted in the near future. The HP-85B and the HP-86B personal computers are selling very well and are in demand. These unique HP Series 80 products are still HP's lowest-priced technical computers and are the choice for technical professionals especially in instrument control and data acquisition applications.

In order to keep HP Series 80 selling well throughout 1985, HP will be announcing in May, 1985 a new extended mass storage plug-in ROM to support new SS-80 disc drives such as the HP 9122D dual disc drive and the HP 9133D Winchester drive. No pre-release EPROM copies of the extended mass storage ROM will be available for either beta testing or early customer needs.

HP 9825 desktop computer to be obsolete May 1, 1985

The HP 9825 has been a very long-lived and successful product, opening the door to the world of small desktop computer systems. Now, however, due to newer and faster technology available in our HP 9000 Series 200, especially the HP 9826, it has reached the point to discontinue the production of the HP 9825. So we are going to obsolete the HP 9825B/T on May 1, 1985.

HP will continue supporting the HP 9825 through the availability of ROMs, interfaces, cables, and upgrade kits for one year after the mainframe goes out of production (May 1, 1986). Technical support will be available for 10 years after mainframe discontinuance.

HP-41 Translator Pac for the HP-71 Handheld Computer

The 82490A HP-41 Translator Pac is a plug-in module for the HP-71 Handheld Computer that allows both an RPN capability very similar to that of the HP-41 Advanced Calculator and also runs HP-41 programs.

The Translator Pac includes the complete HP-41CV programmable function set, plus additional register, flag, and conditional functions from the HP-41CX. While the HP-41CX timer, alarm, and extended memory functions are not supported by the pac, the function set may be extended through user-defined functions.

The Translator Pac may be used to:

- Emulate the HP-41 keyboard operation for calculation.
- Enter the edit HP-41 RPN programs.
- Create and run new RPN programs on the HP-71.
- Create and run new programs in the FOURTH language.
- Create text files.
- Run already translated programs read in from mass storage devices such as the HP 82400A Card Reader, HP 82161A Digital Cassette Drive, or HP 9114A Portable Disc Drive. (Most of over 3,000 translated HP-41 Users' Library programs are available and may be run on an HP-71 with the Translator Pac).
- Transfer programs directly from an HP-41 to an HP-71 via HP-IL.

The 82490A Translator Pac includes the plug-in ROM module, an HP-71 keyboard overlay, and a comprehensive owner's manual. The Translator Pac may be ordered through Computer Supplies Operation.

For further information on the Translator Pac, contact your HP sales rep.

NEW PRODUCTS

NEW MATRIX PRINTERS

Two new heavy-duty line printers from HP feature high-quality, flexible dot-matrix printing at speeds up to 900 lines per minute. The HP 2566A printer at 900 lines per minute (1pm) and the HP 2565A printer at 600 1pm are rugged devices designed for high-speed print applications in data processing, manufacturing and engineering environments.

HP 9000 Series 200 now emulates a variety of terminals

Now you can use your HP 9000 Series 200 computer as an asynchronous alphanumeric terminal to virtually any mainframe manufactured by HP, DEC, Data General, CDC, Prime, and many more.

HP has consolidated three of its Pascal 2.1-based emulators for asynchronous, alphanumeric terminals into one Pascal 3.0-based product. The HP 2392A and VT100 terminal emulator replaces the asynchronous terminal emulator, the HP 2622A terminal emulator, and the VT100 and HP 2622A terminal emulator.

The HP 9000 Series 200 computer retains all of its engineering and computational capabilities while imitating two popular terminals — the HP 2392A and the DEC VT100. The emulator retains the best of the capabilities of its predecessors, with added advantages. These advantages include support on the entire HP 9000 Series 200 (except for Model 226), support on Pascal 3.0 peripherals, and execution directly from Pascal 3.0.

The HP 2392A and VT100 terminal emulator software performs the following tasks:

- Develops and runs programs on a host computer.
- Accesses host peripherals, files, and time-sharing systems.
- Transfers ASCII files between host computer and your system.
- Runs programs designed specifically for HP 2392A and VT100 terminals.

The HP 2392A and VT100 terminal emulator uses a security scheme that locks the software to a specific HP 9000 Series 200 computer using its ID PROM. ID PROM upgrades are available.

P/N	Description
98791B	HP 2392A and VT100 Terminal Emulator

VT100, DEC, VAX™, Datatrieve™, and ALL-IN-1™ are trademarks of Digital Equipment Corporation.

New large-character font cartridge for HP 2934A business printer

Announcing the introduction of a new plug-in character cartridge available for the HP 2934A business printer. The new type style, Orator, similar to the Orator style for the IBM Selectric typewriter, features characters that utilize the full height of the printer's 9 × 12 dot-matrix character cell.

These are approximately 38% taller than a standard character. They are especially useful in generating text for overhead slides, preparation of speeches, or for use when text must be read at a greater-than-normal distance.

This is one of 16 character-font cartridges available for the HP 2934A business printer.

The Orator cartridge has been available through Computer Supplies Operation since January 1, 1985, as P/N 92188K.

HP 2934A offers more features than HP 2932A

The HP 2934A printer offers four more features than the HP 2932A printer. The HP 2934A offers bar codes, large characters, near-letter-quality (NLQ) print, and an optional single-bin sheet feeder.

When considering workstation printers, remember that these printers are built to last up to 10 years, depending upon usage, and that you may have reason to take advantage of these features at some time in the future. You'd be surprised at how many users find applications for the NLQ printing capabilities of the HP 2934A. There is not an upgrade kit for the HP 2932A to make it a 2934A. If you think that you may have a future need for any of these features, consider the HP 2934A instead of the 2932A. You will be impressed with the versatility of your new printer.

DESKTOP FORUM

Electronic Disc Example using an 86A

THE OLD WAY
Do you have a program which loads a large string array and then searches it for an information match? Perhaps this may be a list of names which contains other information characters as a cross reference to a data base or a list of text entries that are referred to by number.

```
e.g    10    DIM A$[25] Array$(600)[25]
       20    A$[1,24] = "FRED" @ A$=A$&" "
       30    ASSIGN#1 TO "ARRAY"
       40    READ#1; Array$()
       50    FOR I = 1 TO 600
       60    IF Array$(I) = A$ THEN 1000
       70    NEXT I
              :
              .
     1000    ! I ve got it.
```

This program uses a lot of RAM and takes a long time to load the array into an 86A from the mass storage device (about 1 minute). This can be particularly annoying when developing the program because the file must be loaded each time the program is run.

In the above example if an element in Array$() is changed then the whole array will need to be PRINTED back to the mass storage device.

THE NEW WAY
If instead an Electronic Disc (ED) file is set up and the data file "ARRAY" arranged as 600 records each of 28 bytes then the file can be copied from the mass storage to ED memory.

```
e.g.2   10    DIM A$[25] B$[25]
        20    CONFIG "ED",":D000",2,32
        30    COPY "ARRAY" TO "ARRAY.ED"
        40    ASSIGN#1 TO "ARRAY.ED"
        50    A$[1,24] = "FRED" @ A$=A$&" "
        60    FOR I = 1 TO 600
        70    READ#1; B$
        80    IF A$=B$ THEN 1000
        90    NEXT I
      1000    ! I ve got it.
```

This program copies the file much faster (in about 7 seconds from a Winchester), but requires an increased search time of 16 seconds compared to 7 seconds. If lines 20 and 30 are executed outside the above program then there is no need to reload the array as in example 1, each time the program is run.

In the example 2 if an element in Array$() is changed then that element need only be written back to the mass storage device.

e.g. 3   300   PRINT#1,I;B$
               where I is the position the original B$ was read from.

In this case at the end of the program the file ARRAY on the ED will need to be copied back to the mass storage device. If only a few elements are corrected in one session then the copy back can be eliminated if a new line is put in example 2.

45   ASSIGN#2 TO "ARRAY"

and a new line is also put in example 3.

305   PRINT#2,I;B$

which ensures the current ED file is updated and that the original file on the mass storage device is maintained correctly. This later point is significant if a power fail occurs as no data will be lost.

CONCLUSION

The new way has proved most successful in a number of programs used by the author and has the special advantage during program development of not having to reload data arrays each time the program is re-run.

A. J. Stevens,
Telecom Australia Research Laboratories,
(03) 541 6532

INCREASING THE SPEED OF LOADING STRING ARRAYS ON AN 86A

There are a number of ways of loading data into a string array in the 86A from a mass storage device. Some of these take much longer than others and are also dependent on the way the file was originally created. The following examples show some of the methods and the tables give the experimental results obtained.

```
10   DIM  Array$(600)[25]
20   ASSIGN#1 TO "ARRAY"

e g 1    100   READ#1,1 @ READ# 1; Array$()

e.g.2    200   READ#1,1
         210   FOR I = 1 TO 600
         220   READ#1; Array$(I)
         230   NEXT I

e.g.3    300   N = 5 @ READ #1,1
         310   FOR I = 1 TO 600 STEP N
         320   READ#1;Array$(I),Array$(I+1),Array$(I+2),Array$(I+3),Array$(I+4)
         330   NEXT I

e.g 4    same as e g.3, but with N = 10 and 10 elements on line 320
```

ACTUAL LOAD TIMES (seconds) Floppy Drive 9130A

File Created as	EXAMPLES			
	1	2	3	4
"ARRAY",600,28	50.0	31.5	31.4	31.4
"ARRAY",70,256	19.7	31.8	31.8	31.7
"ARRAY",1,600*29	18.3	31.3	31.3	30.9

ACTUAL LOAD TIMES (seconds) Winchester 9134A

File Created as	EXAMPLES			
	1	2	3	4
"ARRAY",600,28	41.4	22.2	19.8	19.3
"ARRAY",70,256	14.7	20.7	19.3	18.2
"ARRAY",1,600*29	14.0	19.8	17.3	17.0

These times were obtained by including the following statements in example 1

```
 95   t = TIME
105   DISP TIME - t
```

Similar lines can be used for examples 2 and 3

If lines 100, 220 and 320 are converted into REMARK (!) statements then the execution times for the other statements can be determined.

Example number	1	2	3	4
Time seconds	0.004	1.789	0.380	0.190

These times are insignificant.

Therefore to achieve a fast loading speed care must be exercised in the way the file is created and the method used to load data into the array.

A. J. Stevens,
Telecom Australia
Research Laboratories
(03) 541 6532

SAVING STORAGE SPACE

At our user's meeting of March '83 held at one of our more humble venues — The Windsor Hotel — Chris Simpson gave an excellent talk on "Strings & Things". This talk was subsequently documented in July/Aug. '83 CROSSTALK p. 11-12.

For some years now I have been running a Parish Credit Society using the HP9825 with no mass storage other than the internal tape drive with a capacity of 240K bytes. The membership of the Society is in excess of 1500 and deposits with the Society exceed $500,000.

Each member can have as many as five different types of account —
    SAVINGS ACCOUNT
    LOAN ACCOUNT
    BUDGET ACCOUNT (Bill paying scheme)
    CHRISTMAS CLUB
    FIXED DEPOSITS

All of these accounts have different rates of interest and charges applicable.

The internal RAM on the 9825 is only 24K bytes so you can see the mass storage capacity was stretched to the limit!

By applying the idea of compact number storage, I have been able to significantly expand the capacity of the databank.

The trick is to store numbers converted to the base 256 instead of the normal base 10 numbers. Thus a number in the range 0 to 255 can be stored in one byte, numbers from 0 to 65535 in two bytes, 0 to 16777215 in three bytes and so on.

I adapted this idea to store dates in three bytes by converting the date to the Julian Day and then storing this integer in a string of three bytes. The following listing will enable you to try it directly on an HP86-87 or you can adapt it for use on other desktops.

```
10 REM                    DATES
20 FXD 0


30 OPTION BASE 1
40 DIM A$[3]
50 ! A$[1,3]="   "
```

```
60 CLEAR
70 DISP "ENTER DAY ?"
80 INPUT Day
90 DISP "ENTER MONTH ?"
100 INPUT Month
110 DISP "ENTER YEAR ?"
120 INPUT Year
130 REM
140 REM
150 REM        JULIAN DAY
160 IF Month>2 THEN GOTO 190
170 Month=Month+12
180 Year=Year-1
190 Month=Month+1
200 Julian_day=INT (365.25*Year)-INT (Year/100)+INT
(Year/400)+INT (30.6001*Mont
210   PRINT Julian_day
220 J=Julian_day
230 STORING:
240 REM
250 A$[1,3]=CHR$ (INT (J/65536))&CHR$ (INT (J MOD
65536/256))&CHR$ (J MOD 256)
260 PRINT A$[1,3] ! DATE STORED AS JULIAN NUMBER
CONVERTED TO BASE 256
270                ! AND STORED IN 3 BYTES
280 REM
290 REM
300 RETRIEVING:
310 Julian_day=65536*NUM (A$[1,1])+256*NUM
(A$[2,2])+NUM (A$[3,3])
320   PRINT Julian_day
330 J=Julian_day
340 Year=INT ((J+478164-121.5)/365.2425)
350 Month=INT ((J+478164-INT (365.25*Year)+INT
(Year/100)-INT (Year/400))/30.600
360 Day=J+478164-INT (365.25*Year)+INT
(Year/100)-INT (Year/400)-INT (30.6001*Mo
370 IF Month/3>1 THEN GOTO 400
380 Year=Year-1
390 GOTO 350
400 IF Month<14 THEN GOTO 430
410 Month=Month-12
420 Year=Year+1
430 Month=Month-1
440 REM
450 PRINT "DAY    =",Day
460 PRINT "MONTH  =",Month
470 PRINT "YEAR   =",Year
480 GOTO 70
490 END
```

Using the same principle I was able to store currency in the range +/− $16,000,000.99 in four bytes. The trick here is if the amount of money is negative (e.g.: amount representing a loan balance outstanding), then 100 is added to the fractional part of the number, that is 100 is added to the cents. Then the Integer part of the number is stored in the first three bytes of a string and the Fractional part-the cents, with or without 100 added is stored in the fourth byte.

Again the listing is expanded to enable you to try directly on your desktop —

```
5  !                  STORING MONEY
10 OPTION BASE 1
20 PAGESIZE 24
30 DIM D$[4]
40 CLEAR
50 DISP "ENTER AMOUNT OF MONEY TO BE STORED ?"
60 INPUT M
70 IF M>16000000 THEN BEEP @ PRINT "MONEY TOO LARGE
TO STORE " @ GOTO 50
80 REM
90 MONEY_IN:
100 F=0
110 IF M<0 THEN F=100
120 D=INT (ABS (M))
130 C=FP (ABS (M))*100
140 D$[1,4]=CHR$ (INT (D/65536))&CHR$ (INT (D MOD
65536/256))&CHR$ (D MOD 256)&C
150 PRINT D$[1,4]
160 REM
170 REM
180 REM
190 MONEY_OUT:
200 F=1
210 D=65536*NUM (D$[1,1])+256*NUM (D$[2,2])+NUM
(D$[3,3])
220 C=NUM (D$[4,4]) @ IF C>= 100 THEN C=C-100 @
F=-1
230 D=(D+C/100)*F
240 REM
250 PRINT "DOLLARS IN   = $",M,"   DOLLARS  OUT   =
$",D
260 GOTO 50
270 END
```

I see in the JAN./FEB. '85 CROSSTALK that Henry Drillich has submitted further tricks for compact number storage.

WHO NEEDS MASS STORAGE!

John Hedger,
Spectrometer Services P/L.

HP1000 Users Group — Membership 1985/1986

Name: ....

Company: ....

Address: ....

....

Phone: ....

Telex: ....

Subscription $12 enclosed: Renewal or New

Return to: Norm Kay
HP1000 Users Group
c/- CSIRO
P.O. Box 160
Clayton, 3168

Cheques should be made payable to 'HP1000 Users Group'

H.P.D.C.U.G.V. COMMITTEE MEMBERS
— May 1985

President:	Mr Chris Simpson Simpson Computer Services P/L. 63 Hartington Street, Kew, 3101. Bus: 859 6643, A.H.: 859 6643
Treasurer:	Mr John Hedger Spectrometer Services P/L. Newlands Road, Coburg, 3057 (P.O. Box 135) Bus: 350 1766, A.H.: 459 3413
General Secretary:	Mr Peter Hendy I.G. Hendy Manufacturing 11 Terra Cotta Drive, Blackburn, 3130. Bus: 878 3333, A.H.: 299 2216
Minutes Secretary:	Mr Mark Michell 99 Hotham Street, Collingwood, 3066. Bus: 417 4062, A.H.: 417 2796
Meetings Secretary:	Mr Gordon Roberts TELECOM Radio Construction Branch 14th Floor, 518 Little Bourke Street, Melbourne, 3000. Bus: 606 6063, A.H.: 435 8701
Librarian:	Mr Tony Stevens Telecom Research Laboratories, 770 Blackburn Road, Clayton, 3168. Bus: 541 6532, A.H.: 232 4022
"Crosstalk" Editor:	Mr Henry Drillich 44 Dempster Ave., North Balwyn, 3104. Bus: 857 6816, A.H. 857 6816
H.P. Representative:	Mr John Green Hewlett-Packard Aust. Ltd. 31-41 Joseph Street, Blackburn, 3130. Bus: 895 2895 or 895 2578.

NEXT H.P.D.C.U.G.V. MEETING

The next meeting of the H.P.D.C.U.G.V. will be held on the 30th of May at 4 — 6 pm. at CSIRO Geomechanics Division, Syndal, Vic.

Chris Simpson will have some Series 200 programming manuals to give to those who want them. Henry Drillich will be talking about a new approach to budgeting. HP will provide a demonstration of the HP71B hand-held computer. Please bring your snippets.

MEETING DATES FOR 1985/86 ARE:
30/5/85, 30/7/85, 3/10/85, 26/11/85, 30/1/86, 8/4/86 (A.G.M.).

NOTE: The 3/10/85 is a change from 26/9/85.

OPTIMIZING HP-87, 82901M

DATA TRANSFERS

By creating a revised version of the HP-85 Mass Storage ROM for the HP-87, overall transfer rates for data and programs between the HP-87 and the 82901M disc drive have been increased. However, it is possible to significantly degrade this performance in certain situations. What follows is a brief description of how to obtain optimal performance from your HP-87 and 82901M disc drive.

INTERLEAVE FACTORS

The interleave factor at which a disc is initialized can have on disc performance. Below is a graph of how disc transfer rates vary with interleave factors. The graph shows that the highest average transfer rate occurs at interleave factor six. But it also shows that different applications can benefit from different interleave factors. For example, if an application does mostly reads from the disc, then an interleave of one or 16 could provide significant increases in program execution speed. Refer to the chart to determine which interleave factor should be used. In the majority of cases, interleave six is the optimal choice. And since this is the interleave that the HP-87 initializes discs at, it can usually be left up to the computer to take care of it.

DISC TRANSFER RATES HP87-82901M
10240 BYTES FULL PREC. NUMERICS
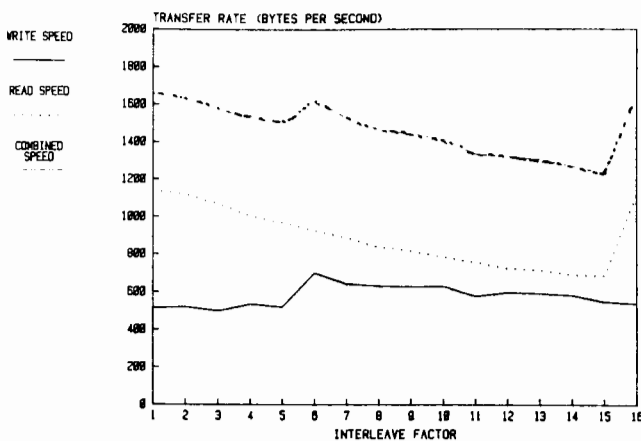
NUMERIC TRANSFERS

Numeric data transfers needn't be worried about. The HP-87 automatically transfers numeric data at the maximum possible rate. However, logical record sizes of 256 bytes will give optimal performance. And don't forget the interleave factor, it always has an affect.

STRING TRANSFERS

Transferring string data is the area most in need of programmer optimization. Because the size of a string can vary (unlike numeric data) the HP-87 will make every effort to insure that no errors occur, such as running over a physical record boundary on the disc. To do this though, the HP-87 has to check every byte as it goes out to insure that this has not happened. This is a very slow process.

To avoid this the programmer must insure that all strings sent to the disc will not cross a physical record boundary. To accomplish this simply ensure that each string sent is 252 bytes long or less. If a string is 252 bytes long, then it and its three bytes of overhead will total 255 bytes, which cannot possibly extend beyond a physical record boundary. Of course, when writing strings in this manner, you must also ensure that each string sent starts at a physical record. If all of these things are done, there should be a speed increase of about 2.5 times.

EXAMPLE

Suppose a string A$ is 10000 characters long. It could be printed to the disc like this:

```
10 PRINT# 1 ; A$
```

But since the string is longer than a physical record, this means that the HP-87 is going to spend time checking for record boundaries every time it ships a byte of information. To prevent it from doing this, you could use a loop like this:

```
10  FOR I = 1 TO 10000 DIV 252
20  PRINT# 1, 1 ; A$[ I*252-251, I*252]
30  NEXT I
40  PRINT# 1 ; A$[10000-(1-1)*252]
```

Line comments:

```
10  FOR I = START OF STRINGS TO # OF 252 BYTE STRINGS
20  PRINT A 252 BYTE STRING
30  DO NEXT 252 BYTE STRING
40  SEND OUT WHAT REMAINS
```

In this example, each string that is sent out is at most 252 bytes. This means that the computer never has to bother with physical record boundaries. This means the computer can transfer the data at maximum speed. Granted this is more complicated than simply saying PRINT# 1 ; A$, but for applications that are disc bound (i.e. they spend most of their time accessing the disc while you sit and wait) this type of programming can significantly reduce your waiting and increase the amount of work done.

— **Access Support,
Hewlett-Packard**

H.P.D.C.U.G.V. "VISIT" TO TELECOM RESEARCH LABS.

Tony Stevens (51 6532) is organising a guided tour around Telecom Research Labs. on Wednesday, 3rd July at 6 pm. The Research Labs. are bristling with HP computers and technical knowhow. Not to mention, how is **OUR** Library? The extent of which may surprise you.

The Research Labs. are at 770 Blackburn Road, Clayton.

EX-PRESIDENT RETIRES

No gold watches or fanfares — just good old-fashioned warm applause was given to Bernie O'Shannassy, who has now retired from the Committee of the H.P.D.C.U.G.V. after three years of invaluable service. It was Bernie who, with his management skills, successfully inaugurated the H.P.D.C.U.G.V., became its first president and drew the interest of over seventy members during those first two years.

Although his current professional position is drawing him away from HP desktop computers, we all hope that he will maintain an interest in the group and we wish him well in his career.

— **C.S.**

Focus 1000

ACCESSING THE SUB-SYSTEM GLOBAL AREA

Gerry Finlay,
Fletcher Jones & Staff Pty. Ltd.,
Warrnambool, Victoria. (055) 62 4011.

The Sub-System Global Area (SSGA) of the RTE operating system is a very useful feature. The method presented here allows access to a table in SSGA, **without the need for Assembler routines,** by using the block data feature of Fortran.

In the examples below, SSGA is used to hold a table of class numbers. Normally, if a program allocates a class number, the class number is lost if the program aborts. But if the class number is kept in SSGA it is never lost.

Firstly, write a Fortran block data subprogram which defines a table of the required size. The following example defines the table ITBLE to have a size of 4 words.

```
BLOCK DATA ITBLE, TABLE IN SSGA
COMMON /ITBLE/ ITBLE(4)
END
```

The relocatable version of the table must be generated into the RTE system and assigned to SSGA by adding the following lines to the on-line generator answer file.

```
* Program input phase.
REL,%ITBLE
* Parameter input phase.
ITBLE,30
```

When the RTE system has been generated with the table ITBLE in SSGA then Fortran programs can be written which use the table. The following example allocates a class number for the program PROGI and stores it in the first word of the table ITBLE.

```
       PROGRAM GETCL () ,GET CLASS

C     Define the table ITBLE in SSGA.
      COMMON /ITBLE/ ITBLE(4)

C     Get a class number.
      ICLAS=0
      CALL EXEC(18, 0, I, O, O, ICLAS)
      ICLAS=IOR(ICLAS, 20000B)
      CALL EXEC(21, ICLAS, I, O)

C     Save the class number in SSGA for PROGI.
      ITBLE(1)=ICLAS

C     Words 2 to 4 of ITBLE are unused.
      END
```

The above program must be loaded with access to SSGA by using the LOADR command OP,SS. If this is not done the program will not load due to the undefined external ITBLE.

Normally the above program would be run from the WELCOM file to initialize ITBLE with the required values at bootup. Then, when the program PROGI is run, it can use the same method to access the table ITBLE to retrieve the class number.

The method presented above has been used successfully on an RTE 4B system with Fortran 4.

Focus 1000

IMAGE DEVELOPMENT TOOLS

by

John Gwyther,
TUSC Computer Systems Pty. Ltd.

INTRODUCTION

HP's IMAGE/1000 database management system (DBMS) has been around now for nearly 10 years, and has found wide acceptance to the point that something like 70% of all HP1000 systems are delivered with IMAGE/1000.

In its heyday, IMAGE was very advanced compared to its competitors. However, since then, it has hardly changed at all. HP has demonstrated very little commitment to IMAGE, other than (belated) efforts to ensure it runs with new operating systems. The outcome of this lack of attention is that the development and maintenance of IMAGE database applications has to date been a tedious, time consuming operation.

Despite these problems, IMAGE is a very handy vehicle for most data-storage applications. At TUSC, we use IMAGE in nearly every application system that we develop. We found that we were repeating certain steps many times in the development cycle, and that errors or omissions in these steps were costing us dearly in time and effort.

Thus, we have put together a series of "tools" to assist the applications developer in his task. These tools range from simple file declaration generators to complete application development systems. Some of these tools have been sourced from overseas, and others have been developed locally. All have valuable roles to play in the life of an IMAGE application.

These tools are:-

DBRELOAD	unload/reload database with changed schemas
IMINC	generate program INCLUDE files describing databases
IMF/1000	interactive repair, analysis and restructuring
SOLUTION	complete applications generator
ASK/1000	full capability report writer
HELP	on-line user HELP facility

Collectively these tools are known as DBTOOLBOX, although they are available individually as well. Let us now look at each of these tools in turn.

DBRELOAD

The purpose of DBRELOAD is to enable a user to unload a database and to reload the data into another database with the same or altered schema definitions. It consists of two programs, DBUNLOAD (which unloads the database into a user file) and DBRELOAD (which reloads the data). With them, you can do such things as:-

* change set and item names.
* change set and item numbers (i.e. changed order of definition).
* change item type, length and location.
* add new items or delete old items.
* seed values into new items.
* change justification and fill values for altered items.
* unload and reload user-selected record sets.

To use DBRELOAD, there are three main steps. First you must unload the database concerned using DBUNLOAD, telling it the database to unload, level 15 password and the file name in which to store that data (FMGR or CI file). Next you alter the schema to reflect the desired changes and run DBDS on it to create the new rootfile and empty datasets. Lastly, you run DBRELOAD to put the data back, specifying the change parameters in a command file.

The changes you can make are as follows:-

1. **name changes** allow you to restore data from set X in the old database to set Y in the new database. In addition, you can restore data from item A in the old dataset to item B in the new dataset. This is very useful for both altering set and item names, as well as "duplicating" datasets.

2. **order changes** allow you to change the order of appearance of items and sets in the schema. This is useful if you wish to add a new master set, or to keep all your data item definitions in alphabetical order.

3. **item definition changes** are the most useful feature of DBRELOAD. If for example you wish to increase a customer name from 32 to 40 characters, DBRELOAD will load the old 32 character item value into the new 40 character item value. If the new length is larger than the old, you can specify leading or trailing padding with user-specified values (normally blanks). If the new length is smaller, you can specify left or right truncation. Other possible uses include 11 to 12 changes, coalescing values into compound items, or breaking compound items apart into discrete items.

4. **adding new items** poses the problem of what value to put into the new item. DBRELOAD allows you a SEED parameter which can be blanks, zeroes or any other string, as well as an existing item value or part of that value.

5. **user-selected records** allow you to select a subset of records to be unloaded. For example, unload all records in all sets with a particular item value.

6. some yet-to-be-implemented features include direct CS80 addressing (reducing the disc storage required for unload), data dictionary interface, more flexible unload criteria and the ability to construct new items.

DBUNLOAD/DBRELOAD run in either RTE-A or RTE-6VM (rev C.83 or later) and support both IMAGE-I and IMAGE-II.

The advantages of DBUNLOAD/DBRELOAD over HP's DBULD/DBLOD are:-

* much more flexible
* supports CI files
* remove restrictions on database changes

The benefit to the applications developer is a more flexible database that can be easily changed or improved without horrendous overheads in time.

IMINC

One of the more tedious tasks facing the applications programmer once he has changed his database is to ensure that his code will cope with these changes. IMINC (short for IMAGE INCLUDE generator) will generate FORTRAN 77 INCLUDE files which fully define in standard notation all the parameters of the database. By following several simple rules for the use of this information, the programmer can ignore the actual values of such things as database name, password, set length, item type, etc.

Whenever the database is changed, all he has to do is to run IMINC against the database, and recompile his programs. Provided the ground rules are followed, he is protected against any nonstructural changes to the database.

The benefit to the programmer is standardised code which is much easier to maintain, plus the removal of those difficult to detect problems that occur when the database definition changes but not all the code does.

As an example, IMINC was run on the database with the following schema:-

```
$CONTROL:LIST,ERRORS=100,ROOT,SET;
BEGIN DATA BASE:AAAAAA:XT:JG;
LEVELS:  1    READ;
         15   WRITE;
ITEMS:   ITEM01,    I1(1,15);
         ITEM02,    X8(1,15);
         ITEM03,    2X6(1,15);
         ITEM04,    R2(1,15);
SETS:

         NAME:      SET01::JG,M;
         ENTRY:     ITEM01(1),
                    ITEM02;
         CAPACITY:  3;
         NAME:      SET02::JG,D;
         ENTRY:     ITEM01(SET01),
                    ITEM02,
                    ITEM03,
                    ITEM04;
         CAPACITY:  5;
END.
```

When IMINC was run on the database root file, it produced this file :-

```
***********************************************************
*                                                         *
* Include file for database:  AAAAAA:XT:JG                *
*                                                         *
* Selected set: ALL SETS                                  *
*                                                         *
* File created by IMINC at   3:56 pm Sun,  14  Apr, 1985  *
*                                                         *
***********************************************************

* Database name and level word parameters for DBOPN

      CHARACTER*22 AAAAAA_name
      PARAMETER   (AAAAAA_name = ' AAAAAA:XT:JG         ')

      CHARACTER*6 AAAAAA_level_word
      PARAMETER   (AAAAAA_level_word = 'WRITE ')

***********************************************************
*                                                         *
*                Set Definitions                          *
*                                                         *
***********************************************************

* Definition of set names

      CHARACTER * 6
      &    SET01_name
      &   ,SET02_name

      PARAMETER (
      &    SET01_name = 'SET01 '
      &   ,SET02_name = 'SET02 '
      &    )

* Definition of set lengths (in words)

      INTEGER * 2
      &    SET01_length

      &   ,SET02_length

      PARAMETER (
      &    SET01_length =     5
      &   ,SET02_length =     13
      &    )

* Record buffer definitions

      INTEGER * 2
      &    SET01_buffer  (SET01_length)
      &   ,SET02_buffer  (SET02_length)

      CHARACTER * (2 * SET01_length)  SET01_char_buffer
      CHARACTER * (2 * SET02_length)  SET02_char_buffer

***********************************************************
*                                                         *
*                Item Definitions                         *
*                                                         *
***********************************************************

* SET01 : Manual master set

      INTEGER * 2     SET01_ITEM01
      CHARACTER *  8  SET01_ITEM02

* SET02 : Detail set

      INTEGER * 2     SET02_ITEM01
      CHARACTER *  8  SET02_ITEM02
      CHARACTER *  6  SET02_ITEM03    (2)
      REAL * 4        SET02_ITEM04

* Definition of item lengths (in words)

      INTEGER * 2
      &    SET01_ITEM01_le
      &   ,SET01_ITEM02_le
      &   ,SET02_ITEM01_le
      &   ,SET02_ITEM02_le
      &   ,SET02_ITEM03_le
      &   ,SET02_ITEM04_le

      PARAMETER (
      &    SET01_ITEM01_le  =    1
      &   ,SET01_ITEM02_le  =    4
      &   ,SET02_ITEM01_le  =    1
      &   ,SET02_ITEM02_le  =    4
      &   ,SET02_ITEM03_le  =    6
      &   ,SET02_ITEM04_le  =    2
      &    )

***********************************************************
*                                                         *
*   Equivalences to link set buffers with item buffers    *
*                                                         *
***********************************************************

      EQUIVALENCE

* SET01 : Manual master set

      &    (SET01_buffer,  SET01_char_buffer)
      &   ,(SET01_buffer  (  1),  SET01_ITEM01 )
      &   ,(SET01_buffer  (  2),  SET01_ITEM02 )
```

```
* SET02 : Detail set

      &   ,(SET02_buffer,  SET02_char_buffer)
      &   ,(SET02_buffer  (  1),  SET02_ITEM01 )
      &   ,(SET02_buffer  (  2),  SET02_ITEM02 )
      &   ,(SET02_buffer  (  6),  SET02_ITEM03 )
      &   ,(SET02_buffer  ( 12),  SET02_ITEM04 )

***********************************************************
*                                                         *
*    Common block declarations of set record buffers      *
*                                                         *
***********************************************************

      COMMON /Common_SET01_/
      &    SET01_buffer

      COMMON /Common_SET02_/
      &    SET02_buffer
```

The programmer uses the name PARAMETER values whenever a set or item or database name is required, and uses the LENGTH parameters whenever an item or buffer has to be moved.

Work is under way to provide PASCAL and HP/C support in IMINC, as well as user-defined "templates" for the output file (which will allow the user to define the format of IMINC's output).

IMF/1000

IMF/1000 (short for IMAGE Maintenance Facility) is an interactive program which operates on existing databases to enable you to alter database parameters **without redefining or reloading the data!**

Using IMF/1000, you perform the following tasks:-

* copy or move the database to another cartridge
* copy or move datasets to another cartridge
* change capacities of any dataset
* detect and repair broken chains
* analyse datasets for synonyms, capacity, etc.
* clean out datasets
* unload and reload datasets to and from tape
* change security passwords and levels
* generate prime numbers for optimum set sizing
* create a DBDS-compatible schema file from the root file
* alter item sizes and locations, set contents, etc.

IMF/1000 is invaluable during application development where you often need to reset your database to rerun test cases, or test the effect of changes in database capacities and so on. It is much faster than doing a DBDS and then a DBLOD, or even a DBRST.

Once the application goes live, IMF/1000 can be a lifesaver as it enables you to quickly recover from database problems such as broken chains, full datasets, poor performance due to too many synonyms, etc. One IMF/1000 user had a CPU failure result in a broken chain in a dataset's free record chain. He was about to embark on a 42-hour database reload when we convinced him to try IMF/1000. In under 2 hours, we had found the chain error and repaired it, saving him an enormous amount of time.

Every IMAGE site is going to face some of these problems at some stage in their life, and you certainly need to have it ready to go when trouble does strike. It's a bit like a parachute; you've got to have it ready for use as it is no good looking for one when the wings come off!

IMF/1000 supports IMAGE-I and IMAGE-II and runs on RTE-6VM and RTE-A.

ASK/1000

HP's QUERY product is perhaps the one part of IMAGE that draws the most flak from users. As a tool for general use, it should be indispensable in all stages of a project. Yet HP have never made the slightest effort to improve the product at all. They even admit that it was developed as an internal development tool that somehow "escaped" into the product. As a result, QUERY is limited to a debugging role during the development stage, and is almost useless as a production report writer.

ASK/1000 is designed as a replacement for both QUERY and the bulk of your application report programs. Operating in the same manner as QUERY, but with vastly simplified syntax, ASK/1000 lets you do such things as:-

* perform multi-set and multi-database finds
* retrieve values from other sets at report time
* generate new "items" using registers
* perform conditional prints in detail lines using IF tests
* allow user programs to reformat report lines at print time

* print item values in header lines
* ensure detail, group and total lines don't cross pages
* sort on date format items (European or USA format)
* call any other program or CI command in-line
* intelligent search methods speed FIND ALL searches
* plus lots more . . .

Perhaps some examples will show you the format and power of ASK/1000. Using a database with master sets CUSTMR (customers) and PRODS (products), we wish to print all orders from a detail set ORDERS, calculating row and column totals per order. The following ASK/1000 command file would do it.

```
FIND ALL ORDERS
*
REPORT
NOSPLITGROUP            ; ensures all groups are on a page
EUROPE                  ; use DD/MM/YY date form
LINES=80                ; define page size
*
S2,ORDER.CUSKEY         ; sort first on customer
S1,ORDER.PRDKEY         ; then product
*
H1,DATE,24              ; print DATE and TIME
* Sort level 2 group lines
G22,ORDER.CUSKEY,10     ; line 1:use key from detail
G21,CUSTMR.NAME,40      ; line 2:get cust. name from master
G21,CUSTMR.ADDRSS,40    ; line 3:get customer address
G11,PRODS.PNAME,60      ; print description from master, using
*                       ; product key from detail set
D1,ORDERS.QTY,70        ; quantity ordered from detail
D1,PRODS.PRICE,80,E1    ; product price from master
R1,LOAD,ORDERS.QTY      ; calculate line price total in R1
R1,MULTIPLY,PRODS.PRICE ; as quantity times price
D1,R1,90,E1
R2,LOAD,ORDERS.QTY      ; calculate qty undelivered in R2
R2,SUBTRACT.ORDERS.QTYDEL ; as qty (ordered) - qty delivered
                        ; set flag F1 to control comment
IF,R2=0,THEN SET F1,ELSE CLEAR F1
D1,"ORDER COMPLETE",100,IF F1
D1,R2,100,IF NOT F1     ; print
*
T1,"PRODUCT ORDERS=",20 ; print total value of product orders
T1,R1,30,E1,TOTAL       ; for this customer
T1,R3,LOAD,R1           ; accumulate in R3 for customer total
*
T2,"CUSTOMER TOTAL=20"  ; print total order value for cust.
T2,R3,30,E1,TOTAL       ; at break point.
END
```

As you can see, ASK/1000 provides considerable freedom in writing reports. If there is something ASK/1000 cannot do, then you can call a user program which is passed all information about the entry being reported plus all registers and flags (64 in all). These user programs are invoked by a statement like:-

    D1,UPROG(4),75

which will call UPROG passing it the value 4. The resultant string from UPROG is printed at column 75.

ASK/1000 will address nearly all your reporting needs without having to write time-consuming and tedious programs for report writing.

I shall discuss the SOLUTION and HELP tools in the next issue.

For more information on these tools, please contact me on (03) 898 0636 or via telex on AA36524 (answerback DCSMMC).

PUZZLE

PLACE

I have four beetles of various sizes, appetites and intellect. Alexander, although not the cleverest, is cleverer than the second fattest. Bonoparte is fatter than the cleverest. Caesar is cleverer than the fattest. Darius is fatter than the third cleverest. The greediest is cleverer than the fattest. The second greediest is fatter than the second cleverest. The third greediest is fatter than the third cleverest. The least greedy is cleverer than the least fat.

Which is fattest, which cleverest, and which greediest?

— **Mark Michell**

PUZZLE SOLUTION

Answer to last issue's puzzle is:
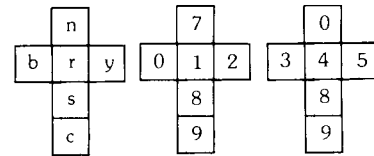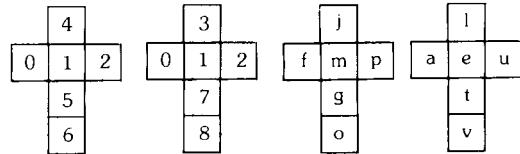
'n' and 'u' are interchangeable.

If anybody is interested, I also have solutions in Italian, Dutch and German — for friends overseas. (Greek and Turkish also).

— **Mark Michell**

COMING EVENTS

4 pm. Thursday, 15th May: Melbourne HP/1000 Users Group Meeting, "Image Processing", CSIRO Division of Atmospheric Physics, Station Street, Aspendale. This is one of the largest HP/1000 sites in Australia. More details by letter to members.

June 17th to 21st: "Productivity 1985", Hilton Hotel, Melbourne.

9 am. Thursday, 12th July: Melbourne HP/1000 Users, UNIX Workshop, HP Blackburn.

? September: Melbourne HP/1000 Users visit to Fletcher Jones, Warrnambool.

4 pm. 14th November: Melbourne HP/1000 Users Meeting, HP Blackburn.

INTERNATIONAL USERS GROUP MEMBERSHIP

At its annual meeting, the Melbourne HP/1000 Users Group announced that it was about to become a full member of the International Association of Hewlett-Packard Computer Users (INTEREX). This will include a copy of the full contributed Library. The Treasurer of the Melbourne Group, Norm Kay, (03) 544 0633, will be the holder of the tape and limited copies of software will be available for members to use, without the need to join INTEREX.

15th May: HP1000 Technical Computer Users Group Meeting, 4.00 p.m., CSIRO Division of Atmospheric Physics, Station Street, Aspendale.

3rd June: 'RTE-A Prog. & Sys. Mgr.' Course, HP Melbourne
'RTE-6VM Sys. Mgr. Course', HP Melbourne

17th — 21st June: "Productivity 1985", Hilton Hotel, Melbourne. Contact HP1000 Users Group.

24th June: 'Getting Started on HP150' Course, HP Melbourne.
'Wordstar on HP150' Course, HP Melbourne.
'Lotus 123 on HP150' Course, HP Melbourne.
'Condor on HP150' Course, HP Melbourne.
'Visicalc on HP150' Course, HP Melbourne.

1st July: 'MS-DOS on HP150' Course, HP Melbourne.
'Assember on HP150' Course, HP Melbourne.
'DSN Link on HP150' Course, HP Melbourne.
'Getting Started on HP150' Course, HP Melbourne.
'Wordstar on HP150' Course, HP Melbourne.
'Lotus 123 on HP150' Course, HP Melbourne.

12th July: Melbourne HP1000 Users Group UNIX Workshop, HP Blackburn.

15th July: 'Introduction to HP1000' Course, HP Melbourne.
'MS-DOS on HP150' Course, HP Melbourne.
'Assember on HP150' Course, HP Melbourne.
'DSN Link on HP150' Course, HP Melbourne.
'Getting Started on HP150' Course, HP Melbourne.
'Wordstar on HP150' Course, HP Melbourne.
'Lotus 123 on HP150' Course, HP Melbourne.

22nd July: 'Fortran 77' Course, HP Melbourne.

29th July: 'MS-DOS on HP150' Course, HP Melbourne.
'Assember on HP150' Course, HP Melbourne.
'DSN-Link on HP150' Course, HP Melbourne.
'Getting Started on HP150' Course, HP Melbourne.
'Wordstar on HP150' Course, HP Melbourne.
'Lotus 123 on HP150' Course, HP Melbourne.

30th July: H.P.D.C.U.G.V. Meeting, 4.00 — 6.30 p.m. at CSIRO, Syndal.

NOTE: If you have requirements for training on any HP products not covered in our formal training programme, please contact Audrey May on (03) 895 2661.

RTE INTERNALS

If anyone is interested in attending an RTE internals course, please contact Brett Hutchinson, Systems Engineer, Melbourne. (03) 895 2661.