TPUG Newsletter

Views and News of Toronto Pet Users Group
c/o John Easton, 258 Lake Promenade, Etobicoke, Ontario, M8W 1B3 phone (416) 251-1511

Fall— 2010

From the President -

Spring and summer have flown by, and fall and winter are at the doorstep. This past summer I re-read **Brian Bagnall**'s book *'On The Edge: The Spectacular Rise And Fall Of Commodore'*. The book is an amazing read and details the history of Commodore computers and Commodore Business Machines. It is an in-depth account of the Commodore years, complete with stories from the makers of the 6502, Commodore 64, Commodore 128, Amiga and dozens of other projects that got off the drawing table and soared, went out the door and failed, or never made it from concept to product. The book is 560 pages long, but once you start reading it you don't want to put it down. A fascinating book, and a great history lesson of the early days of computing.

Reading Bagnall's book always, for some reason, makes me think of three things. Yes, three things came to my mind, and always seem to come to mind when I think of Commodore and they are: what was, what could have been, and, what's next. So in this article, I'll expand slightly on what I mean.

WHAT WAS: When I talk about 'what was', I mean the glory days of Commodore. As outlined in 'On The Edge', Commodore Business Machines was an innovator from the beginning. They introduced the Commodore Personal Electronic Transactor, the PET, in the late 1970's and never looked back. The all-in-one computer was widely popular in schools and businesses around the world. Very few people heard of or conceived of 'home' computers until the VIC-20 was released in 1980. The average person could now afford and use a computer that hooked up to the average television set and in a short time, program them to do many tasks. The Commodore 64 was released in 1982 and the revolution began. The Commodore 64 went on to become the best selling computer of all time with it's incredible sound, video capabilities and games galore to buy and trade. Next came the Amiga with it's incredible graphics. I've said it before and I'll say it again: I think the Amiga was years ahead of it's time and in some ways, still unequalled in computing power.

The 1980's were the boom years for personal computers and for us users of Commodore computers there are many memories of long hours coding games, playing games, using some of the best utility software around (remember Timeworks Sideways? Can't do that on a PC-XT), and (the best part) creating our own utilities. No internet back then, but we got along fine with BBS's, no bloatware, no viruses, and if your computer acts up, just switch it off and on and you're as good as new!

WHAT COULD HAVE BEEN: I'm sure we've all thought about it from time to time, but what if Commodore had survived the bankruptcy of 1994? What kind of a company would they be today? What kind of products would they have made during the 90's and into the 21st century? When Commodore closed shop, it was making products like the CD32, CDTV, Amiga 4000 as well as PC clones like the PC-30/40 and Colt computers. If it had survived past 1994, I think it would have continued with the Amiga line (the Amiga 5000?) but made it better than the standard PC's of the day. As stated, Commodore was always a pioneer in the industry and would most likely have brought the Amiga's incredible multitasking and multimedia capabilities to the forefront.

... continued on page 11 >>>

Member Information

Voice Info
We have discontinued our TPUG phone listing - contact members as listed here at home phones.

Website:	**www.tpug.ca**
e-mail:	**info@tpug.ca**

Membership Rates

NEW ….. $15 per year
see note page 3

Board of Directors

President	Greg VanLaere
Vice President	John Easton
Secretary	Ernie Chorny
Membership Sec.	Ian Colquhoun
Treasurer	Ernie Chorny
Director	Leif Bloomquist
Director	Tom Luff
Director	Ian McIntosh
Director	Joe Palumbo
Director	Tom Williams

Librarians

Head Librarian	Greg VanLaer
Amiga	Ernie Chorny
C128	Tom Luff
CP/M	Ernie Chorny
GEOS	Joe Palumbo
C64	Joe Palumbo
C64 Education	Greg VanLaer
Comal	Ernie Chorny
Plus/4	Ernie Chorny
Vic 20	Ernie Chorny
PET/CBM/SuperPET	John Easton

Support

Mail	Tom Luff
Telephone	John Easton
Disk Orders	Librarians
Member Records	Ian Colquhoun
Meetings	Leif Bloomquist
and	Ernie Chorny
Shows Tom Luff &	Ernie Chorny
Webmaster	Leif Bloomquist

Newsletter

Editor John Easton (416) 251-1511
jeaston@rogers.com

Meeting Schedule

Westside and Amiga West: Third Thursday of the month (except summer) at Alderwood United Church, 44 Delma Drive. Delma Drive is just west of and parallel to Browns Line, south of the Queen Elizabeth Highway, north of Horner Avenue. From the west, exit QEW at Evans Avenue, east on Evans to next stoplight, south on Gair to Delma Drive. From the north or east, follow signs from QEW or Hwy. 427 to Browns Line, exit right to Evans Avenue, turn south on Gair (first stoplight) to Delma.

Contact - **Leif Bloomquist** (416) 737-2328 leif@schemafactor.com
or Ernie Chorny(905) 279-2730 chorny@tamcotec.com

From your Editor …

Another year, another World of Commodore.

Mind you, this year didn't start out so bad … we'd already booked (last year) the venue for this year's event at the same place as last year. Until comments arrived via internet that requests for accommodations on behalf of the WoC met uncomprehending ears.

Well now, what to do? Well, I can only guess, but I'd have to believe that Ian Colquhoun, our intrepid WoC co-ordinator was quick on the mark to approach the next door Holiday Inn with an offer they couldn't refuse. And the rest, as they say, is history. So, here we go again with a sure day (and, if interested, the evening before and the morning after) of unparalled involvement in the World of Commodore.

All that in explanation of why this newsletter wasn't out by early November.

And, carrying on … since our own Joe Quittner insists on producing a new page of tips for almost all monthly TPUG meetings, we're a year or so behind with his output. This month, in an attempt to catch up, you'll find three of his highly educational articles. Hmmm, who knows … sounds like the possibility of an awesome booklet for the future.

And, if this newsletter seems unusually vague in page resolution, I must admit that my trusty HP printer seems to have lost it's fine tuning ability. Please excuse an occasional shift in margains and over-printing. Offhand I suspect a faulty (ink-blocked?) end-of-line sensor, but curiously (obstinately?), just not quite enough yet for total replacement. Tell ya what … if you want the REAL version of this newsletter, check it out on-line in PDF format, hey, you'll even get **colour** where applicable.

Y'r obedient servant
John

TPUG Newsletter is meant to be published somewhat quarterly by the Toronto Pet Users Group (TPUG). TPUG is a volunteer non-profit club dedicated to the service and support of owners and users of Commodore computers.
All rights to material published in TPUG Newsletter are reserved by TPUG, and no material may be reprinted without written permission, except where specifically stated. When reprinting is authorized, please credit TPUG Newsletter, the issue date, and the author. (note - electronic copy *may* be available, please enquire)
Articles, letters, tips, questions, art, etc. are welcome. Send hardcopy or disks "Attn: TPUG Newsletter", or use Internet e-mail.
Advertisements are also welcome. Member's small ads are free. Commercial ads are $100 per page with a $10 minimum.

Notice to new owners of SuperPet and CBM 8296 machines

TPUG has copies of the Waterloo LANGUAGE DISKS (3 in 4040 format) as supplied with the SuperPet on original purchase.

TPUG has the EXECUDESK disk (8050 format) as supplied with the CBM 8296 on original purchase.

These disks are an integral part of the operating systems of the above machines and since Commodore insisted on referring owners of these machines to TPUG for service, we have added these somewhat proprietory (and also virtually unobtainable) disks to our library - all part of the TPUG mandate of service to our members.

We also will attempt to search out copies of original program disks to replace corrupted disks. In this category you will find such programs as VISICALC, WordPro, and PaperClip.

TPUG Annual General Meeting

Take notice that the annual meeting of TPUG will be held in conjunction with the regular January 20th 2011 meeting, only earlier—7:00 p.m. This is your opportunity to discuss club issues and to contribute to the organization as desired.

Typical items covered at an Annual Meeting are, but not limited to: quorum, minutes of the previous meeting, reports on activities within the club, new business, nominations of officers and election of officers.

If you are unable to attend, and wish your proxy vote counted, please ensure that your assigned proxy is in the hands of the Board prior to the meeting. E-mail to the vice-president, *jeaston@rogers.com* will be considered a legitimate means of submitting a proxy to the Board.

Typical Proxy Voting Format —

This will allow ………………………………... to vote on my behalf on all matters at the Annual Meeting of TPUG held 20 January 2011.

Name …………………………………………... TPUG Member # ……………...

Signature ……(difficult via e-mail)……………. Date …………………………..

At our 2009 Annual General Meeting the **annual Membership fee was raised to $15.00**. However, in an effort to pacify the budget-minded, it was decided to allow our members the option of either receiving a newsletter by mail, or on-line in PDF format.

This option allows us to save printing and mailing costs. Should you indicate such a preference on your membership application, these savings will be passed on to those who decline the paper-and-mail route … your membership rate in this case will remain as previous … i.e. $10.00

Remember to indicate this choice on your renewal form … and, should you choose the online newsletter discount, we'll certainly need your current e-mail address.

And note … the membership year now runs from year-end to year-end.

Joe's World 8 ... Markers

EXITS:

The **end** statement in BASIC is final.

The **stop** statement in BASIC can be a final ending plus an indication of the line number in which the stop was asked for. However, using **cont** will continue a BASIC program. This is very useful, for example, in debugging, or making changes before continuing via direct statements (without line numbers), to go somewhere else in a program, or to read/change the value of variables.

SIGNPOSTS:

A subroutine in BASIC always should end with a return statement, but if there are more than one outcome possible, then a signpost can be introduced by putting a number (0-255 only) into a variable, or poking it into a memory location , before the return statement of the subroutine.

```
10  gosub 100 : rem go into the subroutine
starting at line 100
20  if aa=234 then 1000 : rem take an alter-
native path
30  ?"continuing"
40  end
100 zz=1 : rem just anything, for example
101 aa=234 : rem the signpost to the alterna-
tive path
110 return
1000 ?"1000 here!":stop
```

To get back to BASIC from a machine language program, the return **rts** (96) statement is used; but again, if more than one outcome is possible, then a signpost is put in before the **rts** statement. For example, if .X is not used otherwise, then **ldx #5** (162,5) will load 5 into .X (781), and immediately after the return the BASIC program will have an if statement included showing the alternative path if 5 is in 781.

In machine language, the jump statement **jmp** (76) is an important move to another program. For example, 76 123 234 will cause a jump to the address 123+256*234 (=60027) unconditionally.

An indirect jump is available (108). For example, 768 has 139 in it and 769 has 227 in it. Their combination into an absolute address is 139+256*227 (=58251), also, 768/256=3 and 768-256*3=0, so that the indirect jump 108 0 3 will cause 58251 to be the next address. The pair, for example 768/769, is called a vector; it leads to the BASIC routine that prints out the error messages of BASIC (see below). Similarly, 770/771 is a vector that leads to the main BASIC program loop, 772/773 is a vector that produces BASIC's tokens (numbers corresponding to the BASIC commands in English (see below). 774/775 is the vector that leads to the routine that converts a BASIC token (a specified number) into the corresponding command text in English, etc.

There is also a **jsr** op-code (32) in machine language that jumps directly to a subroutine and then comes back after the **rts** (96) op-code that is at the end of the subroutine. There is no indirect jsr equivalent.

To see BASIC's token names (see below) and also error messages, type in direct mode (without the line number)

```
for i=41118 to i+650:?chR(pE(i));:nE
```

BASIC's error listing routine:

```
10 input a:poke 781, a
20 sys 42042:end
Type run and enter a number (1–30)
```

In machine language, the given error number (1-30) is put into .X (781), followed by either the direct jump to the machine language routine starting at 42042 (76 58 164) or, as an alternative, use an indirect jump (108 0 3) via 768/769. This pair of bytes, 768/769, is called a vector.

VECTORS:

A vector is a pair of adjacent memory locations (bytes) that together point to a specific address [peek(i)+256*peek(i+1)]. That pair can be replaced by some other pair of bytes, permitting the user to insert his/her own machine language subroutine. If it is only a diversion, not a substitute, then that inserted user's subroutine at its end should make the machine go to the address that the vector pointed to originally; if it is a substitute then the program may be directed to some other address, or end, or go to BASIC, for examples. Before checking/debugging your new program be sure to save everything first in case you may have to switch off the computer first to continue your work and therefore lose everything that has not been saved on disk or on tape.

TOKENS:

Tokens are numbers that BASIC uses as replacements for the commands in English, also called keywords. Some of them are listed from 41118 (see above), with their last letters in capitals as an indicator to the BASIC routine that that letter is the end of the keyword. These numbers (tokens) start with 128 (end) and continue up to 203 (go) ... for is 129, next is 130, data is 131, rem is 143, etc.

THE BASIC PROGRAM IN MEMORY:

Programs can start anywhere in RAM from 2049 (always after a compulsory zero), with the default start address being 2049. For example:

```
10 rem "TPUG
20 data 12, 34, 56
```

Using the direct mode (without line number) enter this to see how BASIC uses the memory:

```
fO i=2049 to 2078:? stR(i)stR(pE(i));:nE
```

12 8 is the address of the start of the next line, here

12+256*8=2060
10 0 is the line number, here 10+256*0=10
143 is the number of the token, here rem
34 represents the quote character
212 208 213 and 199 are the capital letters TPUG
0 indicates the end of that line. Now, continuing to another line:
29 8 is the address of the next line, here 29+256*8=2077
20 0 is the line number, here 20+256*0=20
131 is the token for data
32 49 50 44 32 are the characters for space, 1, 2, comma, space
51 52 44 32 53 54 are the characters for 3, 4, comma, space, 5, 6
The zero in 2076 indicates the end of that line. Now the next line, if any:
0 0 indicates that there is no further line, it is the end of the BASIC program.

Joe Quittner, TPUG, January 17, 2008

Joe's World 9 ... Pointers

The default memory location (byte) that starts BASIC programs is 2049, but this can be changed by changing the pointer 43/44 content. Similarly, the pointer 55/56 contains the highest location a BASIC program is permitted to use, and it often is changed by the user to protect a program and/or data that are located at higher locations.

Type in this program for example, call it program 1, and then type run:

```
10 ?”No.1":gO15
11 ?“Not to here 11
15 ?”This is 15
```

Without line numbers (in direct mode) type in this program, call it 101; it is used to see where BASIC deposits what. Write down the results, you will need them later.

```
fO i=43 to 56:?i;pE(i):nE
```

To see where the pointers point to, enter this program, call it program 102:

```
fO i=43 to 55 step 2:
?i;i+1;pE(i)+256*pE(i+1):nE
```

Pointer 43/44, pE(43)+256*pE(44) is the start location (byte) of the program area, pointer 45/46 is the start of the variables area, pointer 47/48 is the start of the array area, pointer 49/50 is the start of free RAM (Random Access Memory), pointer 51/52 points to the bottom of the area set aside for string storage, pointer 53/54 is a temporary pointer for string location, and pointer 55/56 shows one above the highest byte that BASIC will use.

MORE THAN ONE BASIC PROGRAMS:

It is possible to load more than one BASIC program within the space 2049 to 40959, but be careful about leaving enough space between them for BASIC to store all constants, variables, arrays and strings while the program is running.

If these pointers are to be changed, then it must be done before entering the program. Unless you have done so already after program 1 (above) was loaded, use program 101 to see, and write down, the contents of bytes 43 to 56.

For example, if not already done so, type in program 1 (see above) normally, with default pointers. Then change byte 44 to wherever you want program 2 to start, and add new for BASIC to change pointers. For this example:

```
pO44,100:new
```

Type in program102 to see the new pointers, then type in this program 2, for example:

```
20 ?“No.2":gO25
21 ?”Not to here 21
25 ?”This is 25
```

Run program 2 then type in program 101 and write down the results for later use.

To get back to program 1, change nearly all bytes 43 to 56 to where they were after running program 1, except that you type 100 (the start of program 2) instead of 160 to prevent program 1 interfering with program 2.

To get back to program 2, again change bytes 43 to 56 to where they were after running program 2.

CHANGING LINE NUMBERS:

Here, for example, is a program to be used as No.2, if the line numbers are to be changed in No.1:

After switching the computer off and back on to start again from the very beginning with default values, type in or load program No.1 normally, then type, for this example,

```
pO44,100:new
```

Type in, or load, the renumber program below as program No.2 and run it. You may wish, as an example, to change the new line numbers of program 1 to start with 1000, and have equal intervals of 10. Note that printers differ; this works for the printer I use. At the end of the renumbering, a list will be shown on the screen (and also printed if the printer is on), showing new and old line numbers. This is important because all jumps, if any, must be changed manually to the new destination line numbers. As a help in this regard, I usually put a colon immediately after the line number in each destination line; often other items must be changed too.

```
43000 rem "renumber by Joe Quittner
43002 close 5:close 7:open 5,4,5:open
```

```
7,4,7:rem for my printer
43004 ?cH(14)cH(144):rem lower case, black
text
43010 fOi=1to10
43011 pR5, cH(27)cH(108)cH(7):rem my
printer's left margin=7
43012 pR5,cH(27)cH(81)cH(77):next rem right
margin=77
43014 dim c(1000,1):rem no more than 1000
line numbers
43016 def fna(i)=i-256*fnb(i):def fnb(i)=
int(i/256)
43018 def fnc(i)=peek(i)+256*peek(i+1)
43020 s=2049:rem this is the starting byte
of program 1
43022 input" New number start, inter-
val";nn,d:nn=nn-d:n=0:rem 1000,10?
43030 :ne=fnc(s):if ne=0 then 43040:rem next
line's starting byte
43032 ln=fnc(s+2):n=n+1:c(n,0)=ln: rem old
line number into array
43034 nn=nn+d:pOs+2,fna(nn):pOs+3,fnb(nn):
c(n,1)=nn:rem new number
43036 s=ne: goto 43030:rem do the next line
43040 :for i=1 to n:print c(i,1)"was"c(i,0:
rem table new-old
43041 pR7,c(i,1)"was"c(i,0)"  ";:next:pR7
43049 end
```

If a program does not start at 2049, and should not start at 2049 when loaded, then save it and load it with 8,1 instead of just 8.

```
save "program name",8,1 or save"@0:program
name",8,1
load "program name",8,1
```

APPENDING A 2nd PROGRAM:

Check carefully first that there are no duplications. Often the easiest way is to simply extend program 1 by typing program 2 into the end of program 1 so as to get a combined program. Then run either from the beginning of program 1 or from wherever program 2 starts. Be careful to save everything before running, in case the computer will stop doing anything and switching off is the only way to reset it to normal, losing everything that has not been saved on disk or tape.

If typing program 2 into the end of program 1 is not an option then type in the first program normally with default pointers. Then type

```
a=pE(45)+256*pE(46)-2:?a
```

This will give you the start byte for the relocated No.2 program. Then

```
pO44,int(a/256):pO43,a-
256*pE(44):clr:sys42291
```

Now type or load No.2. Finally, to start at 2049 and for BASIC to link lines, type

```
pO43,1:pO44,8:clr:sys42991
```

List first before checking or running to see what you've got, in case that you may have to start all over again, to change jumps, etc.

Joe Quittner, TPUG, February 21,2008

Joe's World 10 ... Memory

Changing Memory Contents:

To put the number 123, for example, into byte 20000 type in direct mode (without line number) `pO20000,123`

To see what is in, for example, byte 20000, type
`?pE(20000)`

To see what is in several bytes here is a basic program to do that:

```
10 input a:if a=999 tH end
20 ?pE(a);:gO10
```

To enter 0 to 255 into memory starting at 20000, type:
`fOi=0 to 255:pO20000+i,i:nE`

To see what is in bytes 20005 to 20020, type:
`fOi=20005 to 20020:?pE(i):nE`

To move a whole block, for example, 20010 to 20100, two bytes up, type:
`fOi=20100 to 20010 step-1:pOi+2,pE(i):nE`

To move a whole block, for example 20012 to 20102, down two bytes, type:
`fOi=20012 to 20102:pOi-2,pE(i):nE`

Here is an example program to enter data and/or text into memory starting at 20000. Note that printers differ; this works for the printer I use.

```
0  rem “into memory” by Joe Quittner
10   print cH(144)cH(14)
20   poke 56,78:clr:rem protect 20k+
100  print “     MENU:
101  print “ 1. Type into memory         1000
102  print “ 2. Save memory to disk      3000
103  print “ 3. Load memory from disk    8000
104  print “ 4. Read memory              2000
199  :print “             WHICH? “;:gosub 997
:print g$
201  if g$=“1"then 1000
202  if g$=“2"then 3000
203  if g$=“3"then 8000
204  if g$=“4"then 2000
299  goto 199
997:  poke 204,0:get g$:if g$=“”then 997
998  if peek(207) then 998
999  poke 204,1:return
```

```
1000 print “To end input hold down the CTRL
key while typing e for end
1001 print “To enter an ASCII number press
the f1 key.
1010 n=0:m=20000-1:h$=“”
1012 :goS 997:g=asc(g$+cH(0))
1015 if g=5 then ?:? n ”items, last m is”m:pO
m+1,255:goS 997:gO 100
1016 if g=20 then m=m-1:n=n-1:a=peek(211):if
a>0 then poke 211,a-1
1017 if g=20 then goto 1012
1018 if g=133 then 1030:rem f1 key
1020 :n=n+1:m=m+1:poke m,g:print g$;:goto
1012
1030 :print :input” ASCII Number (0-255)”;g
1032 g$=“”:?”[“+mid$(stR(g),2)+”]”;:goto 1020
2000 rem “ READ MEMORY
2002 gosub 4000
2010 for i=20000 to 256*160-1:a=peek(i):if
a=255 then 2030
2011 if a=13 then print:if p then pR7:goto
2019
2012 if a>31 and a<91 then gosub 2020:goto
2019
2013 if a>192 and a<219 then gosub 2020:goto
2019
2018 if p then print#5,cH(a);
2019 :next:goto 2030
2020 :print cH(a);:if p then print#7,cH(a);
2022 return
2030 :print: if p then print#7
2032 gosub 997:goto 100
3000 rem “ save to disk
3002 gosub 4000:input” FILENAME”;f$:if p then
pR7,f$”:”
3010 n=0:open 8,8,8,”0:”+f$+”,s,w
3012 for i=20000 to 256*160-1:a=peek(i):if
a=255 then 3022
3014 print#8,mid$(stR(a),2):if p then
print#7,stR(a)
3020 n=n+1:print stR(n);:next:stop
3022 :close 8:print n;i-1:if p then print#7
3024 gosub 997:goto 100
4000 rem “ For printer and interface I use:
4010 print “ Printer on? “;:gosub 997:print
g$
4012 p=0:if g$<>“y” then return
4020 p=1:if p1 then return
4021 p1=1:close 5:close 7:open 5,4,5:open
7,4,7
4030 for i-1 to 10
4032 print#5,cH(27)cH(108)cH(5):rem left mar-
gin=5
4034 print#5,cH(27)cH(81)cH(77):rem right
margin=77
4039 print “ Set interface to Junior”:gosub
997:return
8000 rem “ From disk to memory
8002 n=0:m=20000-1:gosub 4000:input “ FILE-
NAME”; f$: if p then print#7,f$”:”
8010 open 8,8,8,”0:”+f$+”,s,r
8012 : input#8,a:s=st:m=m+1:poke m,a:if p
then print#7,a;
8013 n=n+1:print n;a;:if s=0 then 8012
8020 :print n; ”Last m=“m-1:close 8:if p then
print#7,”Last m=“m-1
8022 gosub 997:goto 100
```

BASIC uses consecutive bytes containing not only the program itself, but also enough space for BASIC to deposit constants, variables, arrays and strings that are added to the program space during the running, but machine language programs are not so limited, they can use any available bytes. BASIC space, normally 2049 to 40957, can be reduced, and the remainder then is available for machine language (m/l) programs. There is a large block, 4096 bytes, available for m/l at 49152-53247.

I use disks for storage, so that all bytes in RAM set aside for tape operations may also be used for m/l, such as the 204 bytes 820-1023 and the 23 bytes 2025 to 2047. Also:

On page zero (bytes 0-255) 4 bytes 251-254, and byte 2 are always available for m/l. Tape storage uses on page zero 146, 150, 155-156, 158-159, 165-166, 167 (also used by RS-232), 176-179, 190-192, 195-196. RS-232 uses on page zero 167 (tape use too), 168-171, 180-182, 189, 247-250. In addition, not(?) used, bytes 3-6 are available for m/l on page zero.

On page 1 (bytes 256-511) tape storage uses the 63 bytes 256-318.

To make a lot more m/l space available in RAM, BASIC can be removed completely (maybe only temporarily?), and so can the computer's operating system, but then the user will have to somehow provide substitutes.

Normally the 8192 bytes 40960-49151 of RAM are not used, nor are the 8192 bytes 57344-65535 of RAM. While it is possible to poke into them, peeking is diverted to Read Only Memories (ROM) residing in separate hardware (chips). Byte 1 contains switches that control peeking, whether from normal ROMs or, if desired, from their underlying RAMs.

Before switching either of these from ROM to RAM (it may cause computer hangup!) It is possible to first copy (wait about 30 seconds) into RAM whatever is in ROM, then switch to RAM (after first disabling interrupts?). This makes it possible to then make changes, while no changes to contents can be made while the ROM is in operation.

```
fOi=40960 to 49151:pOi,pE(i):nE
fOi=57344 to 65535:pOi,pE(i):nE
```

To switch from ROM to RAM in range 40960-49151 type:
`pO1,pE(1) and 254`

To switch it back to ROM, type:
`pO1,pE(1) or 1`

by Joe Quittner, TPUG, March 20, 2008

5th Emergency Chicagoland Commodore Convention

By: Ian Colquhoun

The Emergency Chicagoland Commodore Convention was held September 18, 2010 at the Heron Point Building in Lombard, IL. This year ECCC was co-located with the Vintage Computer Festival Midwest 5.0 event which expanded the scope of the brands seen from Commodore to just about anything under the sun from IBM to Apple to HP to Atari and more.

My original plan had been to travel to Chicago with Leif Bloomquist, however a couple of unforeseen problems cropped up and I ended up deciding to just head off on my own while Leif hit the road several hours later. I loaded the car with my travel bag along with two PET 4032s that had been received as part of a donation to TPUG over the summer, hoping to sell them at the expo, and by 9:15am I was heading west on the 401.

The trip was a long one, but pleasant. I had my Sirius satellite radio accompanying the whole way which is nice because you don't have to leave CBC radio on this side of the border. I hit Sarnia just a few minutes past 11am and decided I'd better stop and grab some lunch before heading on. The border crossing was a little slow, but not too bad. The Immigration officer asked where I was headed and I told him that I was attending a vintage computer show in Chicago. He looked in the backseat, saw the PETs and said, "I see that. Those sure are old!"

Google Maps had recommended a route that led from Port Huron, down through downtown Detroit, across to I-96 and SR-14 to Ann Arbor and then to I-94 all the way to Chicago. It turned out that that wasn't the best way because when I hit Detroit, I-94 was down to two lanes and the traffic was terrible. On the way home I avoided Detroit all together and it was a much more peaceful drive.

I stopped briefly mid-afternoon in Marshall, MI for gas and a snack but that was it until the final destination in Lombard.

South-central Michigan is possibly one of the most boring places I have ever driven scenery-wise. There is little of interest along that entire I-94 corridor and as the day pressed on the miles seemed to get slower. Finally however the endless miles of plainness gave way to the city of Gary, IN and for the next better part of an hour and a half I made my way through some of the worst heavy truck traffic I have ever encountered. Gary all the way to Lombard and beyond is solid city, kind of like the way Toronto is from Bowmanville to Milton, only denser. A few stops at toll booths, some attempts at dodging a truck that was billowing a tremendous amount of black smoke and I found myself pulling into the hotel's parking lot at around 5:30pm CDT.

Just after 6pm I headed over to see how the setup was going. ECCC doesn't enjoy the same luxury that World of Commodore does with our essentially unlimited setup time so there was much rushing around. Boxes and plastic bins full of vintage plastic and circuitry was being carted into the Heron Point building. In the basement people were scrambling to ensure they had access to much needed power outlets. By 6:45pm setup time was coming to a close and Jason Compton was busy encouraging everyone to wrap it up, gently at first and then increasing in firmness. Amazingly I think he managed to have everyone out by 7pm when a bunch of us headed to Aurelio's Pizza for some delicious Chicago style pizza!

After dinner I headed back to my room to get the PETs cleaned up and do some keyboard deep-cleaning on them.

In the morning I was up in good time, took advantage of the free breakfast at the hotel, and then headed over to take in the events. I got the PETs unloaded and got my table setup. I had the two PETs and I was also distributing World of Commodore 2010 flyers and TPUG membership brochures.

The expo was taking place in three distinct rooms. The main and largest room was occupied by the Commodore folk. Jim Brain was there selling his wares (JiffyDOS, C64NIC+, etc.) of course, along with many others with the usual treasures to browse. Caricon and his wife(?) were there with their collection of Commodore calculators including a very interesting model that makes use of nixie tubes for the display. The folks from Multiple Classic Computer were there too showing off their rather nifty device that emulates a bunch of different retro gaming systems including a C64.

Over in the other large room (but smaller than the Commodore room) was where the VCF folks were set up. There were tons of machines setup in this room – most of which I had never seen, and some that I'd never even heard of. Jasmaz had an IBM AS/400 system set up and on the network that you could telnet into and play around with. Connected to it was perhaps one of the biggest terminals I had ever laid eyes on. It was physically massive – unbelievably almost as big as the AS/400 itself.

There were numerous Apples set up including a LISA which you don't see every day. Unfortunately I didn't see it powered on so I'm not sure whether it was working or not.

The third room was where the presentation area was setup with a projector and screen. It was also the area claimed by the Atari contingent. Atari 8-bits of all kinds were represented as well as a good number of ST's.

This year there were some rather interesting demos presented:

* **Glenn Holmer** presented **Kipper BASIC** and **BASIC on Bails**, two efforts that bring IP networking to BASIC on your C64. A web server is literally less than five lines of BASIC away.

* **Nate Lawson** presented **ZoomFloppy** (aka xum1541). This is an interface that allows one to connect Commodore serial floppy drives to a PC via USB. One big advantage that it has over past USB efforts is that it also supports floppy drives with parallel connections therefore allowing raw GCR reads with the NIBTOOLS suite. I believe Nate is working in conjunction with Jim Brain to bring this device to market.

* **Payton Byrd** did a presentation on his **CBM-Command** file manager utility for the C64/C128/VIC-20 along with a primer on cross platform development with the cc65 toolset.

* **Dustin Chambers** (Fuzz) and **Ray** (WizardNJ) demoed **Cartograph** and the continuing development of their group's game Crimson Twilight.

Dustin and Ray

* Our own **Leif Bloomquist** demoed his **StereoIn-SID card.** A potentially very capable and interesting card, however there were a few minor technical hurdles that were not quite overcome. Let's hope for a followup at World of Commodore!

There were one or two other presentations that took place, but I was distracted trying to sell a couple of TPUG's PETs so I missed them.

At supper time there was a mass exodus to a local eatery – it may have been called Dapper's West but that doesn't ring a bell with me even though that's the name that appears on Google Maps when I look now. Anyway, it was a fairly typical family restaurant with a staggeringly huge menu. The plate of food that I was served tasted rather good, but the volume of it was crazy. It could easily have served at least two people. Dustin, Ray and I chatted about various things – Canadian vs. American differences and other fascinating topics – then back to the expo venue.

The evening was rather low key and there wasn't anything officially scheduled. People milled around, chatted, hacked, plotted, played games and generally just had fun. I had the opportunity to hook up my electronic bagpipes (brand name DegerPipes) to a C64 via MIDI and literally play the SID as if it were a bagpipe chanter. Useless? Yes. Fun? Lots!

1 am came rather quickly and with it came move-out time. Even though there had been an incredible amount of stuff brought into the expo, it never ceases to amaze me at the efficiency exhibited during move-out. Before you know it, the halls were again empty. Some of us moved over into the hotel lobby and continued visiting and hacking for another couple of hours. I think I packed it in around 3am.

The next morning I managed to hit the road around 10am local time and didn't really feel the worse for wear. Facing the long drive home was a bit daunting, but the traffic in the Chicago area wasn't nearly as bad at that time on a Sunday morning as it had been Friday afternoon. I took a slightly different route back across Michigan to avoid downtown Detroit and found it much more peaceful. I crossed back into Canada at Sarnia just after 6pm and rolled up to my house in Oakville just after 8pm.

All in all it was a great trip and I would recommend it to any of my fellow Commodore enthusiasts! A special thank you goes to Jason Compton for organizing and funding the event year after year. Excellent job!

*Commodore D9060
5 MB hard drive*

World of Commodore 2010

Another year has come and gone and with December approaching it is time for the next instalment of TPUG's World of Commodore show. Continuing from the success of last year's event, we will once again be hosting the show at a hotel. This year **WoC** will be held at:

Holiday Inn Toronto/Mississauga
2125 N Sheridan Way
Mississauga, ON
(905) 855-2000
Toll Free: 1-877-660-8550

For those that attended last year's show at the Admiral Inn, the Holiday Inn is immediately to the east of the Admiral Inn. **Doors open as usual at 10:00** am and events will run throughout the day and late on into the evening.

Tentatively scheduled demos include:

Preserving your precious Commodore manuals and documents on the cheap with an ultra-low-cost DIY book scanner – **Ian Colquhoun**

"Stupid Tricks with KipperBASIC and BASIC on Bails" – **Leif Bloomquist**

All about Arcade Retro Gaming and the birth of the Multiple Classic Computer – **Arcade Retro Gaming**

If you are interested in participating at World of Commodore by presenting a demo then please contact Ian at *icolquhoun@gmail.com*. Demos don't have to be long, nor fancy and no Commodore-related topics are off the table. Topics even marginally Commodore related are welcome!

Confirmed vendors for this year's show so far are:

Toronto PET Users Group
Hardware, software, Library disks, memberships

JP PBM
Super SnapShot carts, new Epyx joysticks, 1581 drive kits, hardware and software for all Commodores

Arcade Retro Gaming
Makers of the Multiple Classic Computer (MCC-216)

The demos are not the only feature of the day. All day long, attendees visit with one another and share ideas. Feel free to bring along your laptop with your various emulator tools, or bring along your whole Commodore rig. Spend the day hacking, gaming, and having a ton of fun.

Anyone that would like to help out with setup on Friday evening or Saturday morning or if you are willing to assist with teardown late Saturday night, please contact Ian at the address above.

Keep an eye on the World of Commodore website at *www.worldofcommodore.ca* for the latest updates on Vendors and Demonstrations and all news related to the show. Join in on the discussion at the TPUG Forums at *www.tpug.ca*.

See you on December 4!

True multimedia and multitasking were just beginning to emerge in the 90's and I think the Amiga would look very much like the modern day personal computers we see today. Commodore was very good at taking something standard (the personal computer) and making it into something better, going one step above and beyond everyone else. I think Commodore would not just make PC's but would make them better, with more options (sort of like Acer computers: well priced, lots of options, well made), built for discerning computer users like us. I can see the commercial now: *man one*: 'I'm having all kinds of trouble with my laptop', *man two*: 'yeah, well my computer doesn't interface well and it's as slow as all get out', *man three*: 'Ha! You guys should have bought a Commodore. I adore my new Commodore 6400-DSL!' (insert incredibly satisfied-looking grin here).

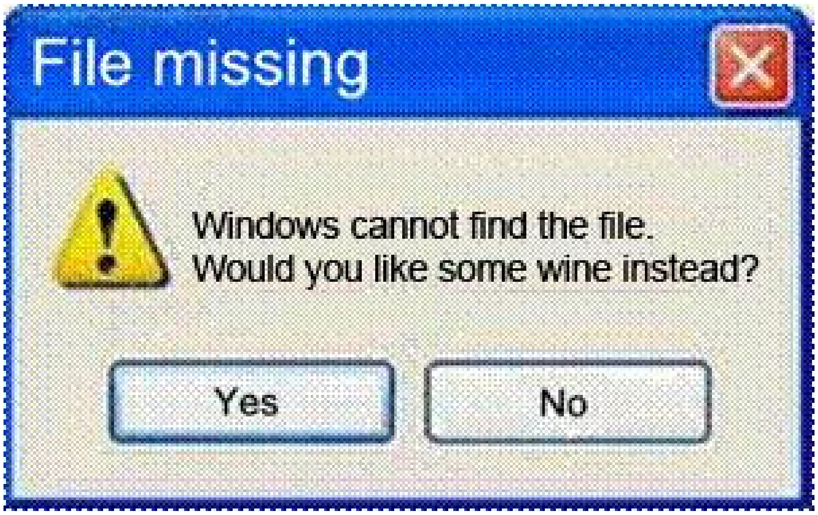
Beyond basic computers, if Commodore had survived I see them making numerous peripherals as well, such as hard drives, optical drives and flat panel monitors. Commodore was actually a pioneer in LCD technology since they made laptop computers. I think if they were around past 1994, we probably would have had colour LCD's monitors by 1996. Who knows? Maybe they would have branched out into non-computer items such as TV's, MP3 players and cell phones.

WHAT'S NEXT: The past several years have seen a tremendous growth in add-on's for our Commodore computers. The Commodore 64 interfaced well with it's peripherals back in 1984, but today we have thumb drives, flash memory and USB. Printers of the past used the standard 25-pin configuration, today it's USB. Disk drives of the past were 5.25-inch floppy drives with the 6-pin cable, today it's half-terabyte SATA drives. We're keeping up with the Commodore (as the old commercial said) but how will our Commodores keep up to the newest hardware? Well, luckily for us there are a number of add-ons and interfaces made so the C64 can talk to the modern world, plus new software to compliment the old. Companies like *Brain Innovations* and *Protovision* make incredible interfaces and software for Commodore products and more are being designed and built all the time.

The future of Commodore computing will be well showcased at this year's World of Commodore show on Saturday Dec 4, 2010. There is a change in the venue for the show this year. This year's event is being held at the Holiday Inn, just down the street from the Admiral Inn (where we held the 2009 show). The location is in Mississauga Ontario and the Holiday Inn is a very short walk from the Admiral Inn. As in past years, there will be a number of vendors and informative demo's that highlight the capabilities of the Commodore. It promises to be an exciting show full of the usual fun and surprises, vendors and demos, rare and common items and a great time for us 'Commodorians' to re-acquaint, discuss and have a good time. Activities will be ongoing during the day that will be sure to answer the question: ***what's next for the Commodore?***

Happy Computing!

Greg Van Laere
TPUG President



Windows cannot find the file.
Would you like some wine instead?

The World Of Commodore

December 4, 2010 from 10AM

At the Holiday Inn which is located on the north side
of the QEW, just west of Erin Mills Parkway
Mississauga, Ontario, Canada

Come see demos, vendors and guest speakers regarding new and innovative products for Commodore computers.

Visit us at: www.worldofcommodore.ca

TORONTO PET USERS GROUP
C/O JOHN EASTON
258 LAKE PROMENADE
ETOBICOKE, ONTARIO M8W 1B3

— FALL 2010 —

We're on the web

www.tpug.ca