

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

Spring— 2010

From the President -

Spring is here! After the long winter, people are coming out of hibernation and thinking about that long overdue project that must get done. Time to start thinking about the Commodore projects that need to be completed. Hmmm, will it be microIEC drive case that I've been working on at a snails pace? Or rebuilding that RTC memory expander cartridge? Or perhaps another stab at fixing that Commodore 64 that continues to stump me with the cartridge port problem, but otherwise functions very well? Perhaps some spring cleaning is in order? (Remember: Commodore keep, IBM toss). Well, whatever your project is, spring is a great time to tinker with something old and start something new!

The Toronto PET Users Group is a large club of people that have a vast amount of talents that they bring to make TPUG great. It is a diverse community that strives for a common purpose: to enjoy the computing experience of the Commodore 64. Each member brings skills to the table that contributes to the clubs success. You may excel at programming machine language code, or demoing a new Commodore product, or master the fine art of soldering. Your skills matter to TPUG.

Many of the skills that I speak of were well represented at the World of Commodore 2009. As President I want to thank the many volunteers that helped make the World of Commodore show a success this past year. Your efforts were appreciated! I arrived at World of Commodore on Friday night for the usual moving of sale items from our storage locker to the hotel. This was the first year at the Admiral Hotel, so we first had to figure out how the room would be laid out for the

show. After moving a few tables and shuffling a few chairs, we finally got it set up just in time for the typical late-evening chat.

World of Commodore started promptly on Saturday morning, and right away (as usual) there was a lineup at the door, an early indicator of the shows success. I was the lead-off demo for the event, and the demo was for our much anticipated release of the TPUG library CD. The library CD contains thousands of D64 images of the entire TPUG library going back as far as 1979, now preserved for good. Pretty much all the systems are represented on the disk, from the PET to the C64 to the Amiga. Much of the software on the library CD was completely unique to TPUG and many of the programs that went into the library were contributions from all over the world. A large portion of these programs came from many TPUG members in the Toronto area, so the CD contains lots of Canadian content (something I forgot to mention in my demo).

Jim Brain also did a demo of ROM-el, JiffyDos and other add-on devices for the Commodore 64 and VIC-20. Jim had a table at WOC where he was selling, showing, and soldering the latest inventions from Brain Innovations. I'm typically a 'stock Commodore' person but this year I decided to make the leap into the high-end of Commodore computing by purchasing a microIEC device. The microIEC allows for mass storage of programs and D64 images. I've been toying around with it ever since WOC.

The day also included demos by Glen Holmer (networking in GEOS), Leif Bloomquist (the

... 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) 221-2328 leif@schemafactor.com
or **Ernie Chorny**(905) 279-2730 chorny@tamcotec.com

From your Editor ...

Well, another year ... another World of Commodore. This year proved to be exceptional thanks to the diligent work of Golan Klinger and Ian Colquhoun in locating our new venue. The Admiral Hotel in nearby Oakville proved to be an excellent location for our requirements.

This occasion also featured the introduction of the much anticipated CD of the entire (well, that is still under discussion ... rumors abound about the possibility of yet more treasures) **TPUG Library**. Our president, Greg VanLaere was the main man behind this endeavour and, as he outlined the efforts and detail required in his introductory talk at WoC, (detailed spread-sheets, un-cooperative disks and ancient contact lists), one has to be amazed that it came together at all.

In this issue, we also feature original articles from our own Joe Quittner, — we'll never keep up with his monthly contributions to our meetings, but at least we've combined two in an attempt to catch up. Leif Bloomquist carries on with reports on his exploits in the Commodore community, and we've included a report from one of our WoC presenters, Chiron Bramberger, the author of PetSyth, a music keyboard application for the PET. Thanks to all our contributors, without you this newsletter wouldn't exist (like ... this is YOUR newsletter, not mine).

Carry on the good work 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 proprietary (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.

Do you have your TPUG CD Library disk yet? Just \$20.00 (plus any applicable taxes)



TPUG
Toronto PET Users Group
Complete Disk Library
2009

Amiga — C64 — VIC20 — COMAL
— PET — SuperPet — GEOS — B128
— C128 — CP/M —
Educational Software

Plus Catalogs, Emulators, Tools,
and Historical Photos

www.tpug.ca

*And, if I'd been thinking, I'd have insisted on a more reproducible disk label ... fine green lettering on black just doesn't cut it ... sorry about that. **John***

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 on-line newsletter discount, we'll certainly need your current e-mail address.

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

INSTANT 1581 DRIVE KIT
(Just add a standard PC floppy drive)

\$49.95

Includes upper and lower shell with logic board & faceplate, a serial cord and power supply box.

- Power Supply only (1581/41-II) \$24.95
- Upper case only \$ 7.95
- Lower case only \$ 6.95
- 1581 Logic Board only \$19.95
- Serial Cable only \$ 8.95

OPTIONAL:

- 1581 JIFFYDOS ROM add \$32.95
- plus 10% shipping (15% USA)

Taxes are extra for Ontario and Canada residents (PST/GST)

Mail cheque or Money Order to:
JP PBM Products by Mail
Box 60515, Jane/Wilson P/O
Downsview, ON, M3L 1B0

Note: Dealers and User Groups Welcome

JP. PBM Products by Mail is the NEW Manufacturer of Super Snapshot Cartridge V5.22 - NOW SHIPPING

*CURRENT Commodore Club MEMBERS SAVE \$5 MORE off the regular price before freight and taxes.

Mail Cheque/M.O. to:	SSv5.22 Cartridge	\$ 74.95
JP PBM Products by Mail	*C= Club members (-\$5)	\$ _____
BOX 60515 JANE/WILSON P/O	32K RAM add \$19	+\$ _____
DOWNSVIEW, ONTARIO		
CANADA M3L 1B0	subtotal	\$ _____
	+10% Freight (15% USA)	\$ _____

All Prices Are Cdn. Funds	Subtotal	\$ _____
US Funds at par	Ontario Res. add 8% PST+	\$ _____
Send \$2 for a catalogue	Canada Res. add 5% GST+	\$ _____
on disk (1541 format)	TOTAL (CDN. FUNDS)	\$ _____

Visit JPPBM.COM for full product listings and PayPal ordering information.

Joe's World – 6 Status Register

Memory location (byte) 783 is the Status Register called .P and it has, just as every other byte, 8 bits that serve here as indicators and are called 'flags'. For example, the Carry Flag (C) has value 1; (peek(783)and1) results in 0 if C is clear, or 1 if set. The Zero Flag (Z) has value 2; (peek(783)and2) results in 0 if Z is clear, or 2 if set; and the Negative Flag (N) has value 128, with (peek(783)and128) resulting in 0 if N is clear, or 128 if set. These flags are very important in machine language operations.

Machine language operations are very fast, and here is a program to show just how much faster they are than BASIC while counting and storing a given value into a given byte (memory location).

```
rem" Speed difference
2 print chr$(14) chr$(144)
10 rem" Example: Loading 123 into memory location
251 20x256=5120 times
20 print "Please wait about 45 seconds!
21 ti$="000000":for i=1 to 5120:poke 251,123:next:t=ti
22 print t " jiffies to load 123 into 251 5120 times, using
BASIC
30 for i=820 to 1023:read a:if a>255 then 40
31 poke i,a: next: stop
32 data 169, 123, 162, 0, 160, 0, 133, 251, 232, 208, 251
33 data 200,192, 21, 208, 246, 96
40 ti$="000000":sys 820:t=ti
41 print t " jiffies to load 123 into 251 5120 times via Ma-
chine Language": print
42 print " There are a total of 2 to the power of 16 =
65536 memory locations. "
43 print "and each 256 is said to be a page, 0 to 255. ";
44 print "251, being below 256, is on page zero. ":print
50 print "Operations involving page zero are faster than
locations 256 - 65535.
51 print " Now comparing the speeds of loading 123 into
12345, 5120 times:
100 rem " 123 into 12345, 5120 times using BASIC
110 print " Please WAIT about 57 seconds!
120 ti$="000000":for i=1 to 5120: poke 12345,123: next:
t=ti
121 print "It took " t " jiffies to load 123 into 12345 via
BASIC 5120 times.
130 c=12345:b=int(c/256):a=c-256*b: print
a+"b"*256="a+b*256:rem 57,48
133 data 169, 123, 162, 0, 160, 0, 141, 57, 48
134 data 232, 208, 250
135 data 200, 192, 21, 208, 245, 96
140 poke 12345,0:poke 251,0:poke 780,21:ti$="000000":
sys837:t=ti
141 print t" jiffies to count 5120 into 12345 via Machine
Language":print
200 rem " Just counting to 10000; first the BASIC speed:
201 print " Please wait about 2 minutes.
```

```
202 ti$="000000":i=0
203 i=i+1
205 if i>10000 then t=ti:print t " jiffies to count to 10000
via BASIC.":goto 210
206 goto 203
210 c=10000:b=int(c/256):a=c-256*b: rem a=16, b=39
211 print "Machine language: we get .X to go to 256 39 ";
212 print "times, and then count 16 more.
220 data 162,0, 160,0, 232, 208,253
221 data 200, 192,40, 208,248
222 data 232, 224,17, 208,251, 96, 999
230 ti$="000000":sys 855:t=ti
231 print "Using machine language, it only took "t" jiffies"
```

Notes:

Line 2: Character string 14 (ch\$(14)) sets upper-case/lowercase mode (not upper case/graphics), and ch\$(144) sets the text colour to black.

Line 21: The elapsed time can be measured in jiffies; there are 60 jiffies in each second. The reserved variable ti is used by BASIC to count jiffies. Setting ti\$ to all zeros also sets ti to zero.

Lines 30 and 31: load the machine language data statements into whatever is needed in memory starting at 820.

Lines 32 and 33 contain the machine language to store the number 123 into memory location 251 5120 times. 169 is the op-code to load the number following the op-code, called an operand, here 123, into the accumulator ((.A) in memory location 780) and also, maybe, set a flag in the status register ((.P) at 783). This depends on the operand number getting loaded into .A, here 123. If the number is zero, then the zero flag (Z) is set, meaning that bit one (value 2) of 783 is set (to 1). If the operand is greater than 128 (the maximum content of bytes is 255) then the negative flag (N), bit 7 of 783, value 128, is set. 123 is neither, and bits 1 and 7 of .P are both clear, zero.

The next op-code is 162 (ldx), and its operand is zero. 162 (stx) loads the operand into register .X (location 781) and, if required, sets the Z or N flag; the operand here being zero, the Z flag will be set, while the N flag will be clear.

Next, (sty) 160 0, loads zero into .Y (location 782), and again the Z flag is set.

Next, (sta) 133 251 stores into 251 whatever is in .A (here 123) for the first time. 133 never sets any flags.

Next, (inx) 232, which needs no operand, increments .X (adds 1 to its value, but after 255 comes to zero again) and then, depending on the incremented value of .X may set either the Z or N flags, or neither. Here we now have 1 in .X with the Z and N flags both being clear (=0).

The next op-code, (bne = branch if not equal or not zero)

208 has one very important operand. Because in this example the Z flag is clear there will be a branch, and the operand will determine where to. We want to go back to op-code 133 from the 208 where we are now. Counting back in the program, starting with 254, we have 208=254, 232=253, 251=252, 133, the target = 251. Therefore we set the operand of 208 to 251 to complete the loop back to 133. Again and again the content of .A will be stored into 251 until, after the 256th time, 251 has been incremented to zero and therefore the Z flag has been set. That changes 208 to no branch, to (iny) 200 which has no operand, but may set either the Z or the N flag, depending on the content of .Y.

Here .Y is now 1, but we need it to go to 21 at which point .X has already counted 20x256=5120 times as required in this example. Op-code 192 (cpy) compares .Y with the content of the operand of 192, here 21, and sets depending on the content of .Y (the N, Z, or C (carry) flags). As soon as .Y reaches 21, the Z flag is set and the next op-code, 208, no longer branches back to 133, but carries on to the next op-code, 96, which returns to BASIC and re-enables the keyboard. The operand of the 208 is arrived at by counting back to 133 from 254 as before: 208=254, 21=253, 192=252, 200=251, 251=250, 208=249, 231=248, 251=247 and the target 133=246.

By Joe Quittner, TPUG, November 15, 2007

Joe's World ... 7 ... Functions and Comparisons

FUNCTIONS:

Functions are simple equations that are calculated by BASIC, but they can be combined. A phantom variable name is used in a defined function, i for example, and this i is on its own, will not clash with the other i used by BASIC.

Often in machine language operations there is a need to calculate the two operands (numbers following the op-code number) that together point to a byte (memory location) that is above 255, on pages 1 to 255, or for use in indirect operations.

I often use these three functions:

```
10 def fna(i)=i-256*fmb(i)
12 def fmb(i)=int(i/256)
14 def fnc(i)=peek(i)+256*peek(i+1)
20 input "memory location (byte) (end=999)";b: if b=999
then stop
22 print fna(b); fmb(b); fnc(b):goto 20
```

fnc(43) shows the beginning of the BASIC program.
fnc(45) shows the beginning of the BASIC variables area.
fnc(47) shows the beginning of the BASIC array variables area.
fnc(49) shows the beginning of the free RAM area
fnc(51) shows the bottom of the string storage area.
fnc(53) shows the temporary string pointer location.
fnc(55) shows the highest byte +1 of the consecutive RAM (variable)

COMPARISONS:

< is smaller than
> is greater than
= is equal to
<= is smaller or equal to
>= is greater or equal to
<> is not
and BASIC's response is 0 if NO, -1 if YES

In BASIC comparisons we must distinguish between strings and numerical items. Comparing, for example, a with b is very easy

```
10 print "123<456; 123=456; 123>456; 123<=456"
12 input"a,b";a,b
14 print a<b; a>b; a<=b; a<>b
```

To compare in BASIC the value of numerical strings, use val(a\$) instead of a\$. This is important in sorting for example.

Machine language comparisons use "flags", actually bits of .P, which is memory location (byte) 783

Here is an example program:

```
0 rem "comparisons by J. Quittner
2 print cH(14)cH(144): rem "U/L case, Black characters
8 print " Non-numerical strings; 0 means NO, -1 means YES.
10 a$="apples": o$="oranges": b$="apples"
20 print "Given that a$="a$", b$=also "b$", and o$="o$"
40 print "a$<o$ a$=b$,equal?";a$=o$; a$=b$
42 print "a$<>o$ a$<>b$ not equal?"; a$<o$; a$<>b$
44 print "a$>o$ a$>b$ greater than?";a$>o$; a$>b$
46 print "a$<o$ a$<b$ smaller than?";a$<o$; a$<b$
48 print "a$<=o$ a$<=b$ less or equal?";a$<=o$; a$<=b$
50 print "a$>=o$ a$>=b$ greater or equal?";a$>=o$;
a$>=b$
52:
134 print: print " MORE STRING COMPARISONS - Nu-
merical Strings - ";
136 print "using val(a$): again 0 means NO, -1 means YES."
138 n1$= "12345":n2$= "16":print "Given that n1$="n1$"
and n2$="n2$"
140 print "Are n1$ and n2$ equal"; val(n1$)=val(n2$)
142 print "Are n1$ and n2$ not qual?";val(n1$)<>val(n2$)
144 print "Is n1$ greater than n2$?";val(n1$)>val(n2$)
146 print "Is n1$ less than n2$?";val(n1$)<val(n2$)
148 print "Is n1$ less than or equal to n2$";
```

```

val(n1$)<=val(n2$)
150 print "Is n1$ greater than or equal to n2$?";
val(n1$)>=val(n2$)
200 print: print: print " Now machine language comparisons:"
210 for i=820 to 1023:read a: if a>255 then 220
211 poke i,a: next: stop
220 input " Load .X with which number (0-255, end=999)";b:if b>255 then stop
222 poke 781,b: rem " Number b into .X, the X Register (781)
224 print " We want to compare the number "b" with 123; lower? equal? or higher."
226 print "Op-code 224 (cpx) compares the value of .X with the number ";
228 print "that follows 224, here 123 as an example.":sys
820:rem "execute
230 data 224, 123, 96, 999: rem "96 returns (rts) it to BASIC
232 print "The number in .X, " peek(781) "is ";
233 c=peek(783) and 131
234 if c=128 then print "lower than 123"
235 if c=3 then print "equal to 123"
236 if c=1 or c=129 then print "higher than 123"
240 print:
goto 220

```

We are here only concerned in these comparisons with the N flag (negative, =128), the Z flag (zero,=2), and the C flag (carry,=1). Because 128+2+1=131 we and 131 with peek(783) in line 233 in order to filter out the values of all other flags.

Usually, after a comparison in machine language, there is one of these branching operations:

bcc =144=	branch if the Carry flag is clear
bcs =176=	branch if the Carry flag is set
beq =240=	branch if equal, or the value is zero
bne =208=	branch if not equal, or if the value is not zero
bmi = 48=	branch if the value is minus, greater than 127 but less than 256
bpl = 16=	branch if the value is plus, less than 128
bvc = 80=	branch if the overflow flag is clear
bvs =112=	branch if the overflow flag is set

By the way, these operation codes are available to change the value of flags, but be careful! It is advisable to first save everything before there is any change of either the D or I flags, just in case there is a "hangup". It is quite safe, however, to poke 783,0 which clears ALL of the flags.

clc = 24=	clear the Carry flag (C)
sec = 56=	set the Carry flag (C)
cld =216=	clear the Decimal flag (D)
sed =248=	set the Decimal flag (D)
cli = 88=	clear the Interrupt flag (I)
sei =120=	set the Interrupt flag (I)

The value of the flags is C=1, Z=2, I=4, D=8, B(break)=16, V=64 and N=128. 32 is not used, but is usually set.

By Joe Quittner, TPUG, December 20, 2007

Twitter on the VIC-20 at the Personal Computer Museum

Like it, love it, or dismiss it as a fad, Twitter is a hot buzzword these days and is becoming ubiquitous. It's even made history in some ways - playing a part in organizing protests during the 2009 Iranian elections, for instance.

On February 20th, 2010 I traveled to the Personal Computer Museum in Brantford, Ontario to witness another small piece of history at the intersection between Twitter and Commodore. Syd Bolton and his team at the Museum have achieved what many would have thought was impossible: Using a Commodore VIC-20 as a Twitter client for sending and receiving messages.

The first "tweet" was supposed to occur at 10 AM, I arrived about 15 minutes early. The museum was already



jam-packed: About 40 curiosity seekers, retrocomputing enthusiasts and reporters filled the museum's lower level, crammed around the pivotal VIC-20 in one corner. Syd loaded the software from a cassette tape (very retro) and ran it.

So at 10 AM, Matt Ruiss, from local radio station FM 92 sent the historic "tweet", while Syd watched over his shoulder. Cameras flashed, people following the action on Twitter (@vintagepc) confirmed that the message went through, and there was polite applause.

Syd then encouraged the crowd to come and try the system out themselves, using their own twitter accounts. I browsed the other museum exhibits while the excitement died down, and then gave the system a try myself.



self use a similar approach for my Telnet Commodore 64 BBS, so I'll stay out of that debate. However, there is a true standalone Twitter client for the Commodore 64 with RR-Net compatible network cartridge, so in theory it could be possible in the VIC as well.

Syd plans to soon release the software they developed at <http://www.tweetver.com/>, so everyone can try the software out and draw their own conclusions.

And the Museum appears to have succeeded in its goals of pulling off a brilliant publicity stunt that put itself and Commodore computers back in the limelight, at least for a couple of days.

Leif Bloomquist



The first thing I noticed was a Commodore VIC-1011A sticking out the back of the VIC. That's an RS232 serial interface cartridge. So I had a suspicion on how this magic feat was accomplished - possibly a modern PC lurking in the back providing the heavy lifting? Several attendees were reaching the same conclusion so we asked Syd about it. He was somewhat coy in his response, but it appears that, yes, there is a Windows PC running the NET Twitter API in the background to take the messages from the VIC and forward them to the Twitter network.

Purists will debate whether the VIC-20 or the PC was actually doing the "Twittering". I my-



Crazy Canuck Charts Course for California (and Commodore)....

In March 2010 I traveled to Silicon Valley (Santa Clara, California) for EclipseCon. Eclipse is a Java-based software development environment comprising an integrated development environment (IDE) and an extensible plug-in system, considered by many to be the state-of-the-art in open source software development. Eclipse is used extensively at NASA, the Canadian Space Agency, and thousands of companies around the world. In the neighborhood by the conference center were the headquarters for Google, Yahoo, Cisco, Citrix...

On the flip side, it's no surprise that there is a strong interest in retro-computing in Silicon Valley as well, with the extensive computer history in the area. As luck would have it, my trip coincided with a monthly FCUG (Fresno Commodore User's Group) meeting, so I arranged to attend.

Fresno Commodore User's Group



Robert Bernardo, FCUG's president, picked me up at the San Jose International Airport on Saturday night and we drove to Fresno. We had a spirited discussion during the drive about cars, 80s movies, Commodore, and the new Steampunk Commodore 64, which Robert had recently commissioned.

On Sunday morning, FCUG treasurer Dick Estel picked me up from my hotel and we proceeded to the FCUG meeting - at Serrano's Mexican Restaurant! They often meet at the nearby Pizza Pit, but Serrano's has the advantage of being quieter and has a small meeting room at one side of the restaurant.

We set up a C128 and VIC-20 on one side, ordered our food (it's hard to find mole sauce this good in Canada!), and the meeting started. Dick Estel gave a brief overview of their club's finances as we ate, and then we moved to the technical portion of the meeting.

One of the highlights was the grand opening of a brand-new, still sealed Canadian Tire Program Pack for the VIC-20. These cassettes were included free with every VIC-20 sold at Canadian Tire back in the day.

Now, in Canada we find Canadian Tire to be ubiquitous, whereas our American friends hadn't heard of the store before and found the concept of an automotive store selling computers rather humorous. So a bit of explanation was in order on this Canadian institution. I just wished I had some Canadian Tire money to show them.

I was intimately familiar with the Canadian Tire Program Pack, as I myself received one with the VIC 20 I bought with my paper route money over 20 years ago. Fortunately, the tape worked perfectly and we tried out each of the games in turn. It was a great nostalgia trip for me, and a novelty for the FCUG members.

We also had a look (but not a live demo) of the VIC-20 Hamtext cart, and tried out the Super Student educational software "suite" for the C64 and the text-adventure Wolfman.

To wrap up, I gave the group an extensive tour of the TPUG Library CD and its contents, which resulted in three sales!

Fresno Autorama

After the FCUG meeting, Robert included me in his other retro interest: Classic hot rods! We went to the Blackie Gejeian's 51st Annual Autorama at the Fresno Convention Center, and oohed and ahhed at the custom airbrush jobs and crazy car dashboards loaded with video screens. From there I took the bus back to Silicon Valley and attended EclipseCon.





Weird Stuff Warehouse

Another lucky break was that EclipseCon was held a short taxi ride from Weird Stuff Warehouse (www.weirdstuff.com) - Silicon Valley's home for retro-computing, surplus computer parts, and all kinds of technological detritus. Most TPUG members are aware of

Active Surplus in Toronto - this was somewhat similar...but about 10 times the size.

I took a lunch break and made the trip to the Warehouse. They had a few items of Atari software and hardware on display and for sale near the entrance, but no Commodore. The store manager said they do get Commodore gear in the warehouse on occasion, but it gets bought up quickly (by FCUG members, I suspect). However, I found a decent large laptop case - large enough to use for my SX-64!

Plans for Next Trip

The conference was over quickly and I had no time to do any further digital archaeology. Next time I'm in the Valley I'll have to visit the Computer History Museum, where Jack Tramiel made a rare appearance back in 2007 at the 25th anniversary of the Commodore 64, and possibly the Intel Museum and the Tech Museum of Innovation.

-Leif Bloomquist

Disk Library ... queries

This From: "Jim Buchan" <jbuchan@sympatico.ca>
To: <info@tpug.ca>
Sent: Saturday, December 05, 2009 4:39 PM
Subject: TPUG Compete Disk Library

Good afternoon.

Just came back after attending your great show this year. Upon discovering the fact that you had the Complete TPUG library on CD, I snatched one up as this is something I've been waiting for... specifically because of one program to which I have been looking for since '85 for the Vic-20 (Bush Trail). A simple slalom game, but was one of my favorites, and does not show up anywhere on the net (funet / zimmers, etc).

I referenced the old 1985 paper copy of the library file and cross-checked it with your PDF file ('VIC 20 Library 300dpi.pdf') which has it listed in the Old Library section (see screenshot below), but after looking through all of the 'v' directories on the CD, I could not locate the file. I'm likely looking in the wrong place, and if you let me know the location to where this file is located, I'd appreciate it, as this is the one file I've been searching decades for.

Much obliged,
Jim Buchan

From: "Greg Van Laere" <gregvanlaere@sympatico.ca>
To: <jbuchan@sympatico.ca>
Subject: RE: TPUG Compete Disk Library
Date: Monday, December 07, 2009 9:15 PM

Hi Jim,

The disk image is located in the 'VIC20' folder, sub-folder 'V-More Utilities'. The disk image you need is (v)v1.d64. If you launch the VIC-20 VICE emulator (located in folder 'XTRAS', sub-folders 'VICE\WinVICE-2.1' there's an application called 'xvic' which is the VICE emulator), then select from the 'File, Attach Disk Image' dropdown menu, you can navigate to the 'V-More Utilities' folder and select the (v)v1.d64 image.

I did the above and typed LOAD "BUSH TRAIL",8 <enter> and it loaded ok. It displayed the following screen: "Ride your snowmobile out to your cottage. Hit any key to continue".

Hope this helps, and have fun!

Greg

PetSynth

Chiron Bramberger

Sound on the PET



The sound on a Commodore PET is generated using the VIA 6522 chip using two memory mapped addresses. The first is the serial I/O shift register, which converts a parallel byte value into a serial binary output as high and low signals on the CB2 line of the user port of the PET, as well as the piezoelectric speaker built into all but the earliest of PET's.

For example, the basic command...

```
POKE 59466,85
```

...would load the shift register with a value that results in 01010101. This would cause the attached speaker to move in and out at a regular time interval, and produce a sound. If you were to look at this sound on an oscilloscope it would be a 50% pulse square wave.

The pitch is determined by the rate at which this value is written out to the CB2, and this is set on the VIA as well. First, the VIA's shift register must be placed into a "free running" condition. This causes the shift register to output the bits "endwise" and to do so over and over again. Once this is done, the timing can be set by writing a value into 59464 of the VIA, such that a larger number results in a longer time interval and thus a lower pitch.

In short, you setup the chip for sound, write the waveform shape, and set the pitch, and you've got a tone being output continuously by the chip without the CPU doing any of the work. This is similar to a dedicated synthesizer chip, like the Commodore 64's SID chip, where the chip takes care of producing the sound while the rest of the computer is free to do other things.

PetSynth produces its basic sounds using this method, which is the way sound is produced in any PET application or game. The program scans for a new key-press using the kbhit() command (part of the cc65 C-compiler) and if there's a new key-press, it grabs the key value and sets the note. After a certain period of time, it turns off the sound, as there is yet no convenient method of detecting a key release style event. This is how **PetSynth** was able to be quickly and easily programmed to be played as a musical keyboard in real-time.

However, the unique effects produced by **PetSynth** are a result of several tricks using the VIA.

The first is the waveform shape, which changes the texture of the sound. By writing other values to this register, one can create tones with added harmonies, or by effectively changing the duty-cycle of the pulse width. This is similar to a draw-bar organ, where you're adding extra overtones. With 8 bits to work with, the variety of 256 waveforms covers a larger number of possibilities than the standard pulse width mode of a typical square wave generator. Most square wave generators have a roughly 4 modes, 25%, 50%, and 75% respectively. **PetSynth** allows you to cycle through all 256 possible values of this register. In the last version as of this writing, on the keypad of the PET 4032, the 1 and 3 keys change this value up and down respectively, while the 2 key set the default value of 15.



The other effect is the vibrato, which is done by changing the value of the current pitch up and down, by adding and subtracting 1 from whatever pitch is currently selected (by what note is being played). However, PetSynth can produce strange sounds by setting this value higher or lower than 1, and in the extremes causes sound effect style sounds more than a musical note. This is adjusted with the - and 0 keys increasing and decreasing this value. The . key resets it back to 1.

The distortion effect works by simply turning the sound on and off again. It is turned on and off with the 8 key at a default value of 1. Adjusting this value doesn't have much impact on the level of distortion as currently implemented.

Finally, the noise effect is created by randomly and rapidly changing the waveform value. The lower pitched notes tend to sound more bass like and the higher pitched notes tend to sound higher as well, but not predictably from note to note. In general, I've used the lower notes as bass drum notes, middle as snare, and higher as high hats. This is turned on and off with the = key.

This is a quick overview of sound generation on the PET, and how **PetSynth** takes advantage of this to turn the PET into a real-time keyboard playable synthesizer. In the future, I hope to incorporate more effects, different sounds, and other interesting features so that the PET

can enjoy a musical place alongside the Amiga, Commodore 64 and Vic-20 family of computers.

PetSynth is released under the GPL open source license, using the cc65 compiler. The compiled and source code versions of the software can be downloaded at the **PetSynth.org** website.

Ed. Note ... the photos are from the Music Demo at the 2009 World of Commodore. Rob Adlers (C=64) in the foreground, Chiron Bramberger (PET) in the middle and Leif Bloomquist (VIC 20) in the distance ... yes, all playing at once.



... from page 1

future of networking for the Commodore 64), Dustin Chambers (the Cartograph software), Ian Colquhoun (A solid state hard drive for the Amiga), and the sweet sound of Commodore generated music was provided by Rob Adlers (using MSSIAH64), along with Leif Bloomquist (using VIC-MIDI on the VIC-20) and Chiron Bramberger (using PetSynth on the Commodore PET). Jason Compton gave a talk on his experiences with the Commodore with one of the best demo titles I've ever heard: 'I lost my hair to Commodore, a chronicler's tale, 1993-1998'.

Commodore Business Machines was a trendsetter back in the 70's and 80's. Many of the computational devices that Commodore produced were ahead of their time and quite unique. This is why we decided to include a Commodore 'Wall of Fame' in the show this year. We set up numerous working systems along one wall and invited users to interact with the rarest of the vintage computers. I marveled well at the 8296 PET (the Por-

sche PET as it's sometimes called). We had one of these in my school library. I can remember there seemed to always be someone sitting at it, either doing a book report that they would eventually send to the disk drive (or that clunky printer), or enjoying a game of Space Attack surrounded by a small (but quiet) crowd. Lots of vintage memories and fun was had at World of Commodore 2009!

To start off 2010 right, we held our AGM (Annual General Meeting) on Jan 21. Congratulations go to John Easton, Ian Colquhoun, Ernie Chorny and Ian McIntosh who were re-elected as directors. Many questions were raised and answered. The meeting was well attended and the evening set the direction for the club for the next year. Yes, this was democracy in action.

What a great club we have! We have a proud history, a diverse group, and the skills and the talents to make 2010 a banner year for TPUG!

Greg Van Laere
TPUG President

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

— SPRING 2010 —



We're on the web

www.tpug.ca

