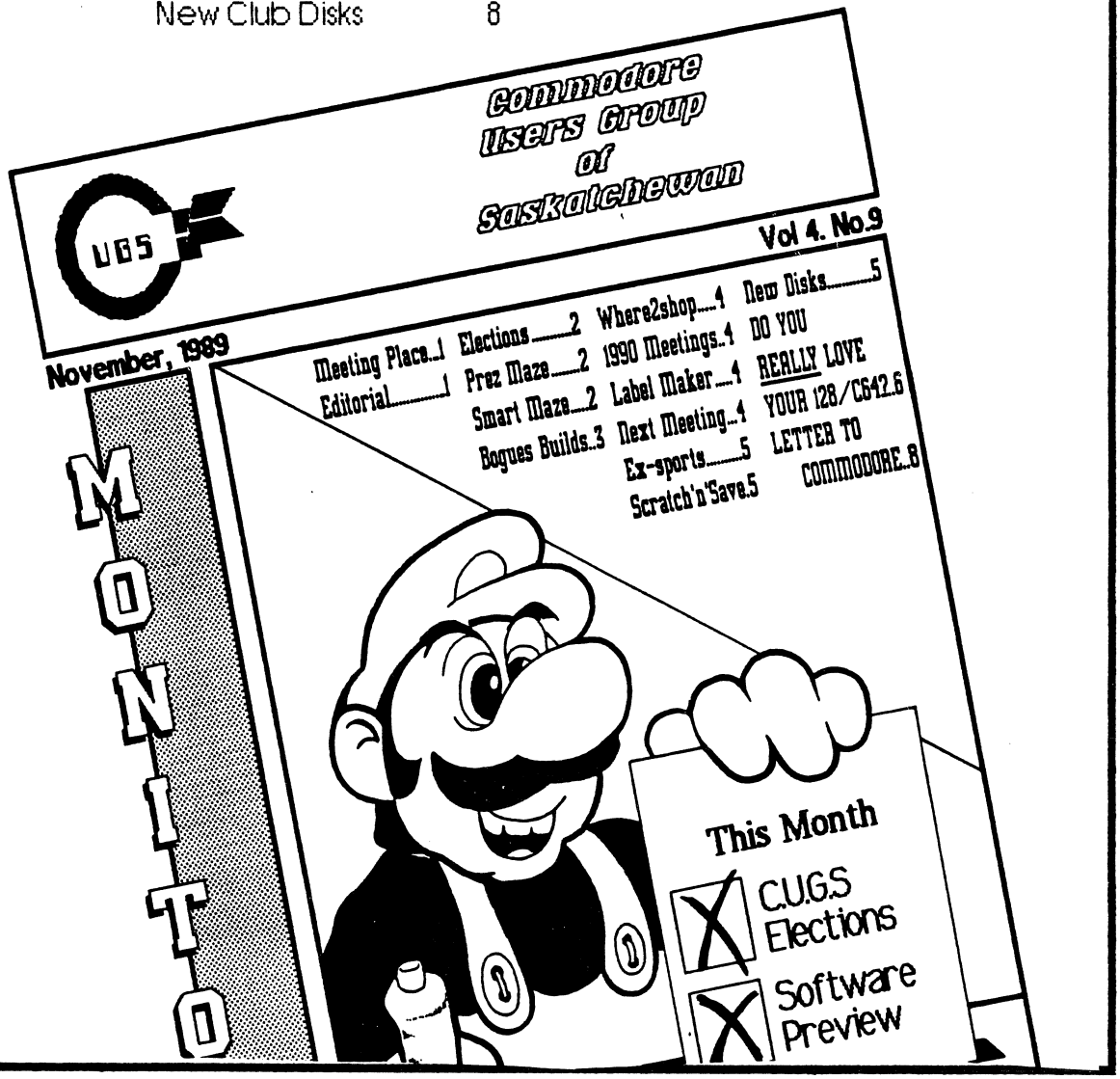
Commodore Users Group of Saskatchewan

May, 1990 **Vol 5. No. 5**

MONITOR

What's Inside



BBS List for Regina, Saskatchewan, Area Code 306, For April, 1990.

Name of System	Phone #	Baud	BBS Program Used	Sysop's Name	Code
Bar Room BBS	584-7145	2400	DeusCBCS v0.1c	Rod Snaith	I
Billionaire's Boys Club	586-9571	1200	Wildcat Ver 1.03	Jason Howorko	I,G
Bit Bucket	352-3236	2400	Fido v12h	Bart Ritchie	I
Buccanner's Den	352-2477	1200	Opus v1.03b	Ryan Cannon	I
Computer Way, The	757-1210	2400	GT Power Beta #14	Volker Polan	I
CUGS BBS	586-1189	1200	AABBS v11.3	Richard Maze	C
Cyberlink BBS	789-4378	2400	RemoteAccess v0.02	Matthew Ornawka	I
Datapac 300	565-0111	300	Westbridge	none	
Datapac 1200	565-0181	1200	Westbridge	none	
Datapac 2400	565-6000	2400	Westbridge	none	
Diddly Squat BBS	586-4136	1200	RemoteAccess v0.02	David Shivak	I
Double Check	525-0807	2400	Paragon v2.04g	Randy Coghill	P
Double Q Access	584-2916	1200	Bruce 9000	Robert Patterson	I,G
Excalibur	949-8605	2400	EBBS Ver. 4.5	Yves Desjardins	C
Fernando's Retreat	585-0298	9600	Opus v1.03b	Colin Campbell	I,G
Forbidden Zone	586-0794	2400	GTPower v15.01	Greg Armstrong	I
FrINgE.	543-7935	1200	Minibaud v 4.0	John Alexander	A,*
Ganshirt-at-home	543-1949	1200	Fido v12h	Ken Ganshirt	I,G,M
Impossible Mission	569-9705	1200	ALBBS	Kevin Hoffman	C
Lab BBS, The	525-8620	2400	QuickBBS Ver. 2.61	Yong Lim	I,G
Magic Fountain	586-2692	9600	Opus v1.03b	Scott Wilson	I
Micro City I	584-0747	2400	MCBBS	Ron Ware	A,G
Micro City II	584-0748	2400	MCBBS	Ron Ware	A,G
Missing Link, The	522-4001	9600	GT Power 15.01	Stephen Crawford	G,M
Myth Drannor	569-3183	1200	C-Base 2.0	Rob Addie	C,D
Polestar	586-1551	9600	RemoteAccess V0.02a	Bryce Eckstein	I
Pool Hall, The	586-0922	9600	PCBoard Ver. 14.0	Rodger Linka	I,G,M
Pool Hall, The	586-8490	2400	PCBoard Ver. 14.0	Rodger Linka	I,G
Punisher's Pain Empoureum	543-9128	1200	KGB BBS v5.0A	Gilles Archer	C,D
Ratt II	949-6105	1200	BBS Express	Keith Grill	I
Regina FIDO 1	777-4493	9600	Fido v12h	Ken Ganshirt	I
Regina FIDO 2	569-0271	2400	Fido v12h	Ken Ganshirt	I
Shadowland	789-8989	2400	Home Made	Bob Hamilton	I
Star Traiders Inc	545-0259	2400	Opus v1.10	Robert Gunther	I,G
Tee Wun Kay	779-1237	2400	Opus v1.10	Garry Ehman	I
Turbo Plus BBS	949-8880	2400	Home Made	Jim Nickel	I
U of R 2400	585-5216	2400	Deckserver Cluster	Develswitch Sal	*
Unibase 1200	789-0709	1200	Unix	Leigh Calnek	
Unibase 2400	789-0715	2400	Unix	Leigh Calnek	
Welfare BBS	586-3665	1200	Minibaud v2.2	Dave Guerrero	C,G

Codes: I=IBM board A=Apple board C=Commodore board P=Amiga board
G=Games D=Temp. down *=7,E,1 settings M=MNP

ALL BULLETIN BOARDS run at 8,N,1 modem settings unless otherwise stated.

Obligatory Stuff

CUGS MAILING ADDRESS:

CUGS
143 Birchwood Cres.
Regina, Sask.
S4S 5S3

CUGS BBS - (306) 586-1189

President	Barry Bircher	359 1925
Vice President	Richard Maze	586 3291
Treasurer	Real Charron	545 7681
Editor	Jarrett Currie	757 2391
Asst Editor	Shaun Hase	584 3371
Librarian	Keith Kasha	359 1748
Asst Librarian	Steve Bogues	949 1378
Members at Large	Ken Danyliczuk	545 8644
	Harry Chong	789 2142
	Earl Brown	543 2868
	Gord Williams	543 8373
	Joe Gomes	789 8174

If you have any questions about CUGS please feel free to contact any of the above executive members.

The **Monitor** is published monthly by the COMMODORE USERS' GROUP OF SASKATCHEWAN (CUGS), Regina, Sask., Canada. CUGS meetings are held at 7 pm the FIRST WEDNESDAY of every month (unless otherwise noted) in the North-West Leisure Centre, corner of Rochdale Boulevard and Arnason Street. Next CUGS meeting:

Wednesday, June 6

Anyone interested in computing, especially on the C64, 128 or 64C, is welcome to attend any meeting. Out of town members are also welcome, but may be charged a small ($5.00) mailing fee for newsletters. Members are encouraged to submit public domain software for inclusion in the CUGS DISK LIBRARY. These programs are made available to members. Any member is entitled to purchase DISKS from our public domain library for a nominal fee. Programs are 'freeware', from computer magazines, or the public domain. Individual members are responsible for deleting any program that he/she is not entitled to by law (you must be the owner of the magazine in which a particular program was printed). To the best of our knowledge, all such programs are identified in their listings. Please let us know if you find otherwise. Contact our club Librarian, Keith Kasha.

CUGS is a non-profit organization comprised of C64, 64C, C128, and 128D users interested in sharing ideas, programs, knowledge, problems and solutions with each other. The more members participate, the better the variety of benefits. Membership dues are pro-rated, based on a January to December year.

Editorial

by

Jarrett Currie

Athough there are many types of computer users, and many have tried to typecast them, I want to put them into three broad categories for sake of discussion.

There are those practical enough that they purchase a computer as if it were a kitchen appliance. These hopelessly planned and organized souls have read all the articles and heard all the salesmen and developed a plan to purchase their computer. Financial resources and software purchasing out of the way before the machine's acquistion, these unfortunates are doomed to life with a bored and starving computer.

The second, though less systematic than the first, have marvelled at the growing electronic toys and while he couldn't exactly explain why, has decided that he too must have one. Easily swayed by the bright lights sand surrealistic sounds, he purchases the first one with a name he has heard before. Although Barnum had described this type of computer purchaser more eloquently than I ever could, it is not above me to observe that while he is doublessly pleased with his machine, he has still alot of following to do. He purchases software and hardware as quickly as his friends do and lacks any sense of adventure in being the first on his block to own a title.

The last type is the true computer zealot. He is fully aware of the fact that while his computer may never earn its keep financially, it is a great way to spend the night! His hardware and software spending is erratic at best and longs to own a diskfull of programs all lacking directions. The computer for him is an adventure: sometimes into the pits of frustration, sometimes to the hills of ekation, and always to the plains of pleasure. He seeks out, actively, the best in software, and has gone so far as to write it when it could not be found. The country's borders are no barrier, for he seeks sustenance for his computer internationally, if necessary. And, most of all, he shares his expertise with all those who will listen.

As you sit and read this article, place yourself in one of these broad categories and attempt to realize what the Commodore community is missing if you don't place yourself in the third category. Owning a computer can be as exciting as owning a blender if all you do is punch in numbers.

But, there is more. Lots more. But you will never appreciate what your computer can do for you if you don't actively participate in aquiring that knowledge.

What better place to start exercising your new skills than with our club? If you have always wanted to try telecommunicating, but were a little shy on the other boards, try the club's! If you have ever wanted to program but there seemed to be too many hurtles, then by all means give one of our experts a call. And, if you are thinking about buying a new software package, just mention its name at a club meeting, and see what response you get.

Learning to succeed with your computer is not like learning to program your microwave, and it has a great deal more potential. With some enthusiasm on your part, you too can be a proud computer owner and a valuable member to our computer club.

Network News
Finding BASIC Strings

by Bob k7 Kober

There is the POINTER command in the C128's BASIC 7.0, that can be used to locate any string's descriptor, and from it, the location of the string in memory. The BASIC 2.0 in the C64, however, doesn't have such a command, so finding the strings is a little more difficult. Herein, is an explanation of how and where strings are stored, and a program in BASIC and ML that will actually find them.

Where string data is stored

Strings in the C64 can be stored in two areas. They can be located in the actual BASIC text area, or in the string storage area, located at the top of the BASIC program area.

If you have a line in your program like this:

```
10  A$="hello":B$=" there"
```

both the "hello" and the "there" would be located in the BASIC text area. If you were now to have a line following like this:

```
20 C$=A$+B$
```

the data in C$, "hello there", would be stored in the string storage area at the top of BASIC memory.

String Descriptors

In order for your program to find these strings, it uses a "descriptor" which is located in the variable storage area located immediately following the end of the BASIC program.

The start of this variable storage area is pointed to by the pointer in VARTAB, located at 45-46 ($2D-$2E), and the end of this area by the pointer in ARYTAB, located at 47-48 ($2F-$30).

The string descriptor is 7 bytes long, and contains the variable name in the first two bytes, the string length in the third byte, and the location of the actual string data in the forth and fifth byte, in normal LObyte/HIbyte order. The last two bytes are unused, and contain zeros. (Note that the string length is contained in one byte, and this explains why a string can be no longer than 255 bytes.)

To identify the descriptor as a string, the second byte of the variable name has BIT 7 set. If the string variable is B$, then the first byte is 66 ($42) for the B, and the second is 128 ($80), a zero byte with BIT 7 set.

If the string variable were BK$, then the first byte would still be 66 ($42), but the second byte would be 75 ($4B), for the K, PLUS the 128 ($80), BIT 7 set, or 203 ($CB).

Setting the 7th BIT of these two bytes in different combinations, designates different kinds of variables. But, for his article, we are concerned only with strings.

Finding the Descriptors

Since these descriptors contain the address of the string data, all we have to do is find the descriptor. Here's how we do that.

Let's say for this example, we use BK$ as the string we want. (Well, I did write this article!). Search the variable storage area between the locations pointed to in VARTAB pointer, and the ARYTAB pointer, looking for BK$ which will be two consecutive bytes containing the value 66 ($42) and 203 ($CB).

Once we find these, the byte immediately following the 203 contains the string length, and the two after that, the string's address in memory.

Just multiply the second of these by 256, and add it to the first, and we have the address. ie: if these two bytes contained 33 and 191, the string would start at 191*256+33 or 48899, and go upward from there.

Here's BASIC

Okay, now you know the theory behind finding the strings, here's the actual BASIC program that will do it.

```
10   BK$="HELLO ":GOSUB 30
20   BK$=BK$+"THERE":GOSUB 30:END
30   S=PEEK(46)*256+PEEK(45)
40   E=PEEK(48)*256+PEEK(47)
50   FOR I=S TO E:A=PEEK(I)
60   IF A=66 THEN X=1:GOTO 80
70   IF X THEN IF A=203 THEN J=I+1:I=E
80   NEXT:ON-(J=0)GOTO 130:L=PEEK(J)
90   AD=PEEK(J+2)*256+PEEK(J+1)
100  PRINT:PRINT"STRING LENGTH IS"L
110  PRINT"STRING LOCATION IS"AD
120  RETURN
130  PRINT"VARIABLE BK$ NOT FOUND"
```

This routine will return the lengths and addresses of 2 strings. "HELLO" which is in the BASIC text area, and "HELLO THERE", which is in the string storage area in high memory.

Here's ML

Here it is in monitor format. Notice it uses several of the built in BASIC ROM routines

```
0334   20 FD AE    JSR  $AEFD
0337   20 9E AD    JSR  $AD9E
033A   20 A3 B6    JSR  $B6A3
033D   85 02       STA  $02
033F   98          TYA
0340   20 CD BD    JSR  $BDCD
0343   A9 2D       LDA  #$2D
0345   20 D2 FF    JSR  $FFD2
0348   A9 00       LDA  #$00
034A   A6 02       LDX  $02
034C   20 CD BD    JSR  $BDCD
034F   60          RTS
```

Here is the same thing in the form of a BASIC loader.

```
10   FOR D=828 TO D+27
20   READ Y:POKE D,Y:NEXT
30   DATA 32,253,174,32,158,173
40   DATA 32,163,182,133,2,152
50   DATA 32,205,189,169,45
60   DATA 32,210,255,169,0,166,2
70   DATA 32,205,189,96
```

This routine can be used in two ways:

```
BK$="This is a test"
SYS 828,BK$
```

SYS828,"This is a test"

The return will be location and length of the string in this format XXXXX-L where XXXXX is the address of the first character of the string, and L is the length of the string.

It may not ever be necessary for you to find a string's location in normal programming, but should you ever need to, now you know how it is done.

From the Desk of the PRESIDENT

As you all have probably heard, the club as a whole is putting on a library expansion blitz. We are purchasing several disks full of programs for our machines to eat. Almost every area will receive a few calories. As I mentioned last month, the club is presently opening up a new library GEOS section. We have ordered several disks and, as of yet, have had not heard or seen of them. I'll give them a couple of days more to cough up.

After reading several other clubs' newsletters and getting to know some of these clubs better, C.U.G.S. is par to above par with any other club. With Commodore 8-bit computers, slowly but surely going to the way-side, user groups and the owners of these machine are surely going to get better acquainted.

Even though the manufacturer will eventually drop the computer, as they all will do at one time or another, the C-128 and especially the C-64 will be around for many years to come. With our increasing information ring slowly growing, you can be sure that the club is getting involved with other users nationwide. Any gossip or tidbit of information will be gleaned off of these groups along with our already established ring of articles in the Commodore magazines.

I have been a Commodore 128 user now for several years. I can stand and be proud of what our machines can do, will do and have done. At this point in history, there has been NO computer, bar NONE, that has lasted as long as the 64 has. It is now estimated that there are 9 million 64 and 128 computers out there with a very strong base of 64/128 software developed and still being developed for it. IBM would have loved to have that kind of base with the failed PCjr. At one point, there were more 64's than there were IBM compatibles.

The point here is that the machines were developed for the home user in mind and was and is very successful. In using both the 64/128 and an IBM at work, I often wonder how IBM ever got it's reputation in the home environment. I can see the advantages in business, but as a home computer?

The club is doing rather well. There is always room for improvement, however. During a recent executive meeting, it appeared to me that the executive are a very dedicated group of users who enjoy learning and teaching others how to use their machine. They are a group with a vision. A vision with a future. They are dedicated in preserving a classic computer by increasing a library so YOU, the member, are able to have a source of programs and help for years to come.

You can help the club help you by lending a hand, giving a program to the library, helping others, telling others of C.U.G.S. writing an article for the newsletter, being there at the meeting, asking questions, and just using the club.

AGENDA

May, 1990

Disk and disk drive (A physical aspect)
by Barry Bircher

Desktop Publishing (Generic)
by Ken Danylczuk

Draw -- 5 1/4 diskette cleaner kit
donated by Software Supermarket

HINTS AND TRICKS

by
Shaun Hase

This first hint is, I think, familiar to everyone. I have two 1571's and most of the 64 software I have only uses one drive, so one drive gets more use. I suppose this is good, but I thought that it might be better to use my second drive once in a while. To change drive 8 to drive 9 and vice versa, I came up with the following program, that can be entered in immediate mode, to do so, without turning any drive off.

```
10 OPEN15,8,15,"U0>"+CHR$(10):CLOSE15
20 OPEN15,9,15,"U0>"+CHR$(8):CLOSE15
30 OPEN15,10,15,"U0>"+CHR$(9):CLOSE15
40 NEW
```

The program first sets drive 8 to drive 10, sets drive 9 to drive 8 and then sets drive 10 to drive 9. In short, without any physical changes, the drives have had their device numbers switched. Now, for someone who has a 1581 disk drive and would like to use it as drive 8, this little program is perfect. To clear this set-up, you could run the program again, turn the drives off, or reset the computer (not RUN/STOP - RESTORE).

The second hint deals with printing columns of text with Paperclip III. Paperclip III can not actually print columns, but with the use of two commands and a bit of forethought, printing columns is no problem. The format directive used to do this is <>ap#, where <> is the checkmark, ap is the command to do alternative page printing and # is whether you want all pages (0), odd pages (1) or even pages (2) printed. I'm not sure if older versions of Paperclip have this directive. Once you've got the margins all set up to print on half of the page, insert the alternative page printing directive somewhere above the printing of any text. Next, set it to print the odd pages of your document. Once it is done that, re-insert the paper the same way it was. One of two things can be done now. You can either adjust your margins, or use the <>of# directive to shift the output to the right without affecting the margin settings. Then, change the alternative page printing directive to print even pages and print the document again. You should have two columns of text on each page. I have found that using the elite pitch (12 cpi) works best (you get more characters per line). You could even use this technique to print out a simple newsletter.

GEONEWS
from QLink

RUN Magazine is pleased to [announce] an agreement with Berkeley Softworks to market the long-awaited geoBasic program to GEOS users. Developed by the GEOS pros at Berkeley Softworks, geoBasic lets GEOS users program their own applications in BASIC and take advantage of the ease of use of GEOS. RUN, which obtained the North American distribution rights, plans to begin shipping this product in mid-June of this year.

According to RUN publisher Steve Robbins, RUN seized the opportunity to bring to market a product that many GEOS users have been clamoring for. "We know that thousands of GEOS users have been waiting a long time for this product, and we're pleased to be in a position to deliver geoBasic to the GEOS user community. We're certain that users will feel it's been worth the wait," said Robbins.

GeoBasic supports all of the features - pull-down menus, information boxes, icons and mouse pointer for easy point-and-click operations - that make GEOS so easy to use. Also, with geoBasic users can create programs that use icons, menus, sprites and dialog boxes. RUN Editor-in-Chief Dennis Brisson noted that "GEOS users are accustomed to high-quality products to use with GEOS; geoBasic follows that tradition." He states that RUN has assigned some of the top GEOS programmers to generate sample applications to show you what can be done with geoBasic.

GeoBasic is a 40-column program that runs in C-64 mode on the 64 or 128 (preliminary testing of the product, however, indicates that geoBasic will also work in 128 40-column mode.) GeoBasic includes a text editor for entering and editing programs, menu editor, bitmap editor, icon editor and dialog box editor. It includes up to 104 new commands. It supports color and sound, text windows, drawing commands and mouse support, as well as structured loops, subroutines, mathematical functions and access to machine language.

The disk, which comes with a complete operations manual, will sell for $39.95. See ad in May issue of RUN for ordering information. It is important to note that any questions regarding geoBasic should be directed to RUN, not Berkeley Softworks.

EXECUTIVE MEETING

Minutes

April 9, 1990

1) A letter and newsletter from the "WEST BANK" users group was received by the club.

2) Duplication of users group newsletter for the executive was approved. Ken to be responsible.

3) Monitor to be mailed after contact made by a users group. Honorary membership to users group corresponding with CUGS. APPROVED -- Barry to provide the names and addresses to Real.

4) Barry purchased GeoWorld disks #1-5 and #23.

5) CUGS to start a GEOS library. Earl to look after it. Disks to be GEOS-ready.

6) It was decided that the club should be able to publish the Monitor using club-owned hardware and software. To this end, a commitee was struck. The members of the commitee are: Jarrett, Shaun and Keith. They are to look into the hardware and software needed, availablity, and cost. They will report back to the executive at the next meeting.

7) The inventory list of all CUGS assets (hardware and software) is to be updated for the next meeting. The information is to be given to Real for compilation. Phone him the information or leave a message on the board.

8) The future of CUGS BBS was discussed. To be reviewed at a later date.

9) CUGS mall demonstration discussed. Maybe in September or December.

10) Hall rental for the fall. Request submitted. Will not know until after June 25.

11) Executive individual photo reqested by Barry. They will be digitized for the Monitor.

12) Next executive meeting May 7, 1990 at Barry's, time 7:00pm.

EXPERTS LIST

The people below have agreed to let their names be listed as "experts" in some aspect of C64/128 computing. If you've a question, these brave volunteers can likely answer it, or help you find an answer that works. If you have a skill at some computing process, consider listing yourself with our other volunteers.

Wordprocessing

Paperclip III	Shaun Hase	584 3371
Paperclip (to version E)	Richard Maze	586 3291
Paperclip (to version E)	Jarrett Currie	757 2391
Paperclip (any version)	Ken Danylczuk	545 8644
Pocket Writer 2 & 3	Yves Desjardins	949 8526

Spreadsheets

Multiplan	Richard Maze	586 3291
Pocket Planner	Barry Bircher	359 1925
Better Working SS	Ken Danylczuk	545 8644

Databases

Pocket Filer	Barry Bircher	359 1925
Oracle (Consultant)	Ken Danylczuk	545 8644

Communication

Pro128Term	Barry Bircher	359 1925
Pro128Term	Jarrett Currie	757 2391
Library files	Barry Bircher	359 1925

Music/Sound

(Most)	Ken Danylczuk	545 8644

Languages

Forth	Ken Danylczuk	545 8644
Pascal	Ken Danylczuk	545 8644
ML (machine language)	Ken Danylczuk	545 8644
ML (machine language)	Barry Bircher	359 1925
BASIC (general)	Richard Maze	586 3291
BASIC 7.0 (graphics)	Shaun Hase	584 3371
BASIC (2.0-7.0, files)	Ken Danylczuk	545 8644

Graphics

Print Shop/Master	Ken Danylczuk	545 8644
Koala Painter/Printer	Ken Danylczuk	545 8644

Hardware

Disk Drive Maintenance	Ken Danylczuk	545 8644

GEOS

GEOS 64	Jarrett Currie	757 2391
GEOS 128	Barry Bircher	359 1925

General

Super Snapshot (3, 4, 5)	Yves Desjardins	949 8526

C.U.G.S
would like to thank
Software Supermarket
for their donation of a
disk cleaner
for the
prize draw

Floating Point Numbers

By Bob (k7) Kober

A Floating Point (FP) number is the product of a mantissa multiplied by a power of two. In Commodore BASIC, FP numbers occupy 5 bytes, with the power of two contained in the "exponent" (1st) byte, and the mantissa in the last four bytes.

Briefly, a mantissa is a sum of NEGATIVE powers of 2, just as a binary number is the sum of POSITIVE powers of two. This will be explained further later.

The exponent byte actually contains the true exponent value with a 129 ($81) offset added, so that it can represent both positive and negative numbers. Therefore if the exponent byte contains a $84, then the true exponent is $84-$81 or $03. Two is raised to the power of 3, and thus the mantissa is multiplied by 8. Note that a zero in the exponent byte indicates a FP number of zero, regardless of what the mantissa holds. This makes it easy to zero any FP number.

The four-byte mantissa (32 bits) can hold any number between 1.0 and 1.99999999, and is arranged with the most significant byte, (MSB) first. BIT #7, of the MSB, is also the most significant BIT, and must has a value of 2^0, or 1. Bit #0 is the least significant BIT, and has a value of 2^-3, or 0.00000001.

BASIC "normalizes" its numbers so that the value in the mantissa is always equal to, or greater than 1. This is to prevent leading zeros, which would reduce the accuracy by pushing the least significant bits out of the mantissa. It does this by multiplying the mantissa by 2, until all the leading zeros are gone. Then the exponent is reduced by 1 for every "times 2" that was required.

Now, since the normalized number is always 1 or greater, BIT #7 of the mantissa's MSB will ALWAYS be set. This allows it to be used as the sign bit. When signed math is required, this BIT is checked, and if found set, the number is considered negative. If signed math is NOT required, this BIT is always assumed to be set.

Here is the procedure used to convert a decimal number to its Floating Point equivalent.

Let's convert PI (3.141592654) to FP.

1) Change the number so that it is in the range of 1 to 1.99999999, by multiplying or dividing by two, keeping track of the number of times required. Since the number is now too high, we will divide by two.

   3.141592654 / 2 = 1.570796327, so it only took 1 operation. Therefore we will ADD 1 to the exponent byte. 129 + 1 = 130. Now we have the first digit of our FP number, $82 (130).

2) Divide this scaled down number by two, so the mantissa's MSB has a value of 2 to the power of -1, rather than a 2 to the power of 0 This is the intermediate mantissa, .7853981635. This sort of "un-normalizes" it so it can be further evaluated.

3) Now, we multiply this new number, .7853981635, by 256, giving us 201.061929900. The 1st mantissa digit, is the integer of this number, 201 or $C9. However, since our number is positive, we clear BIT #7, and get $49. This then becomes the second digit in our FP number.

4) We now take the fractional part of the above number, .061929900, multiply it by 256, giving us 15.854054400. The integer of this $0F (15)is the third digit in our FP number.

5) Repeating step 4 using the fractional part .854054400, we get an integer of 218, and a fraction of .637926400. We now have the fourth digit in our FP number, $DA (218).

6) Once more we multiply the fraction left over by 256, giving us 163.30915840. Since this is the last digit, we round up if the fraction is greater than .5. Since .3 is less than .5, we'll keep 163, the last digit in our FP number $A3 (163). Note that this rounding of the last digit is the cause of the minute errors found in some math calculations performed in BASIC.

FINALLY, the Floating Point equivalent of the decimal 3.141592654 is this string of bytes:

$82 $49 $0F $DA $A3.

Here is the procedure in reverse - converting to decimal from Floating Point.

Starting with $82 $49 $0F $DA $A3

1) Starting with the exponent byte, we get the true power by subtracting the offset, $81 (129). $82-$81=$01. Therefore, the mantissa will be multiplied by 2 to the power of 1, or 2.

2) BIT #7 of the next byte is NOT set, so the number is positive. Now that we know that, we have to SET BIT #7 to do the remaining calculations. Therefore $49 is

now $C9, and our mantissa is $C9 $0F $DA $A3.

3) Starting with the LSB, we divide this byte, ($A3) 163 by 256, and get .63671875.

4) Now, add this to the next byte, and get 218.63671875. Divide this by 256, and get .85404968.

5) Continuing, adding the next byte $0F (15) gives us 15.85404968. Dividing this by 256 returns .061929881.

6) Once more, adding the remaining byte $C9 (201), = 201.061929881. Dividing this by 256 gives .7853981635.

7) Now, we have our intermediate mantissa, .7853981635. To get the true mantissa, (remember we have to "normalize" it) we multiply this by 2 to raise the mantissa's MSB to the value of 2 to the power of 0, or 1. Now our true normalized mantissa is 1.570796327

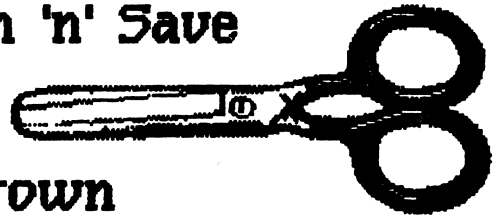
8) Finally, to get the decimal number, we multiply the true mantissa by the exponent value 2, as calculated in step 1 above, and get 3.141592654.

So, now we have gone from Floating Point designation, back to a decimal number.

NOTE: In the above, we use 5 bytes to store the FP number. This is the format used when the number is in a variable. When it is in the Floating Point Accumulator, it is stored in 6 bytes, and BIT #7 of this sixth byte is now the sign BIT, rather than BIT #7 of the MSB.

Scratch 'n' Save

by



Earl Brown

Last month (April), was my birthday. And my dear wife gave me a CD player for a gift. About four years ago, we bought a stereo color TV and about three years ago a stereo VHS recorder. As far as the TV and video recorder are concerned, there was really no big surprises to me as how far computerization has evolved in these two mediums. Perhaps because there was a lot written and advertised on TV's and video recorders and I examined them on a continuous basis in retail stores. How to store your scanning channels, how to set a sleep time, and how to program the clock and recording times, were things I was expecting in these entertainment devices. But for a CD player, I was totally caught off guard. I hadn't looked into the where-for-thou of CD players. I was expecting something that just played CD disks in the same manner as a record player played 7" and 12" records. I did assume there would be a PLAY button and a STOP button and a method of selecting a 3" or a 5" Compact Disk. But never in my wildest dreams did I realize that you could:

1) Randomly play the selections.
2) Play one selection at a time.
3) Fast forward or reverse full selections.
4) Fast forward or reverse through partial selections.
5) Program in any order all or part of the CD selections.
6) Fade in or out of any selection.
7) See the playing time, or balance of time of a selection or CD.
8) Pause a selection for any period of time
9) Program time length to fit standard sizes of audio cassettes.

And perhaps as I go along, I may even find more things that I could do with this CD player. Granted, if you don't record audio cassettes, all these features will mean diddley to the average listener. But if you do, then you welcome with open arms all of these features, and are wishing for a few more. The CD player is certainly sofisticatedly computerized (a mouthful). And there is no question about hearing the flawless reproduction of a completely digitized musical selection (DDD). A little departure from my regular column.

CUGS has just received ten new sides of GEOS programs that will be added to our library as early as next month. Barry Bircher is the one who ordered them and he assures me that there are some dandy GEOS programs amongst them. If you are a GEOS owner, put a few bucks aside, as these disks will be well worth the money.

I have prepared two disks for this month's meeting. First, as always, the May, 1998 issue of Gazette programs (listing appears elsewhere in the Monitor) and the third Graphic disk for the 128 Library (#RC). Among other programs this disk contains the RUNPAINT program along with the Japanese 1351 mouse fix for this particular program (originally published in Run) for those of you who own this magazine. This program also appears on a graphic disk in our 64 library. It works in either mode. Pick the disk with the programs you would like the most. See you next month.

Next Month

June Agenda

Spreadsheets
by: Barry Bircher

Garage sale and rap session

Draw -- $30 gift certificate
donated by THE DUNCANS

Commodore User Group of Saskatchewan

143 Birchwood Crescent
Regina, Saskatchewan
S4S 5S3

Registration Form

C.U.G.S. is a non-profit orgianization comprised of C64, 64C, C128 and C128D users that exists solely for its membership. If you would like to renew your membership, or start a new one, please take a moment the complete the following registration. The charge for a 1-year membership is only $10.00, plus $5.00 if you want the newsletter mailed to you.

Name: _____

Street: _____ City: _____ Prov: _____

Postal Code: _____ Phone Number: _____

Would you like the **Monitor**, the club newsletter, mailed to you? (There is a $5.00 mailing charge per year.) Yes _____ No _____

Participation Contest

I would like to remind C.U.G.S. members that the **Monitor** is still running the Participation Contest.

For each article you get published in the **Monitor**, you will receive another chance to win the prize to be awarded in December.

And for the first article, you will receive 1 free public domain disk, of your choosing, from our club library.

Here's how it stands so far:

Name	# of Entries
Shaun Hase	4
Richard Maze	4
Barry Bircher	3
Earl Brown	1
Ken Danylczuk	1
Yves Desjardins	1

NEW CLUB DISKS

CUGS GAZETTE MAY #90

screen store
screen store/128
alpha lock
dvorak keyboard
numeric keypad
interrupt prog
vdc monitor+/128
monitor+ rel/128
custom cursor
minimap
minimap demo
race ace
isolation
megasqueeze
sheerluck

128 GRAPHICS 3 #RC

animator/128
runpaint boot
spriteshell/128
color editor/128
color demo
revolver demo
vdc graphics/128
clock demo.vdc
paint thin.vdc
worm demo.vdc
colorplot/128
gr demo/128
surfaces/3-d/128
r.p. mouse bug
screendesign/128
fracscapes128
cellular1
seq read(128/80