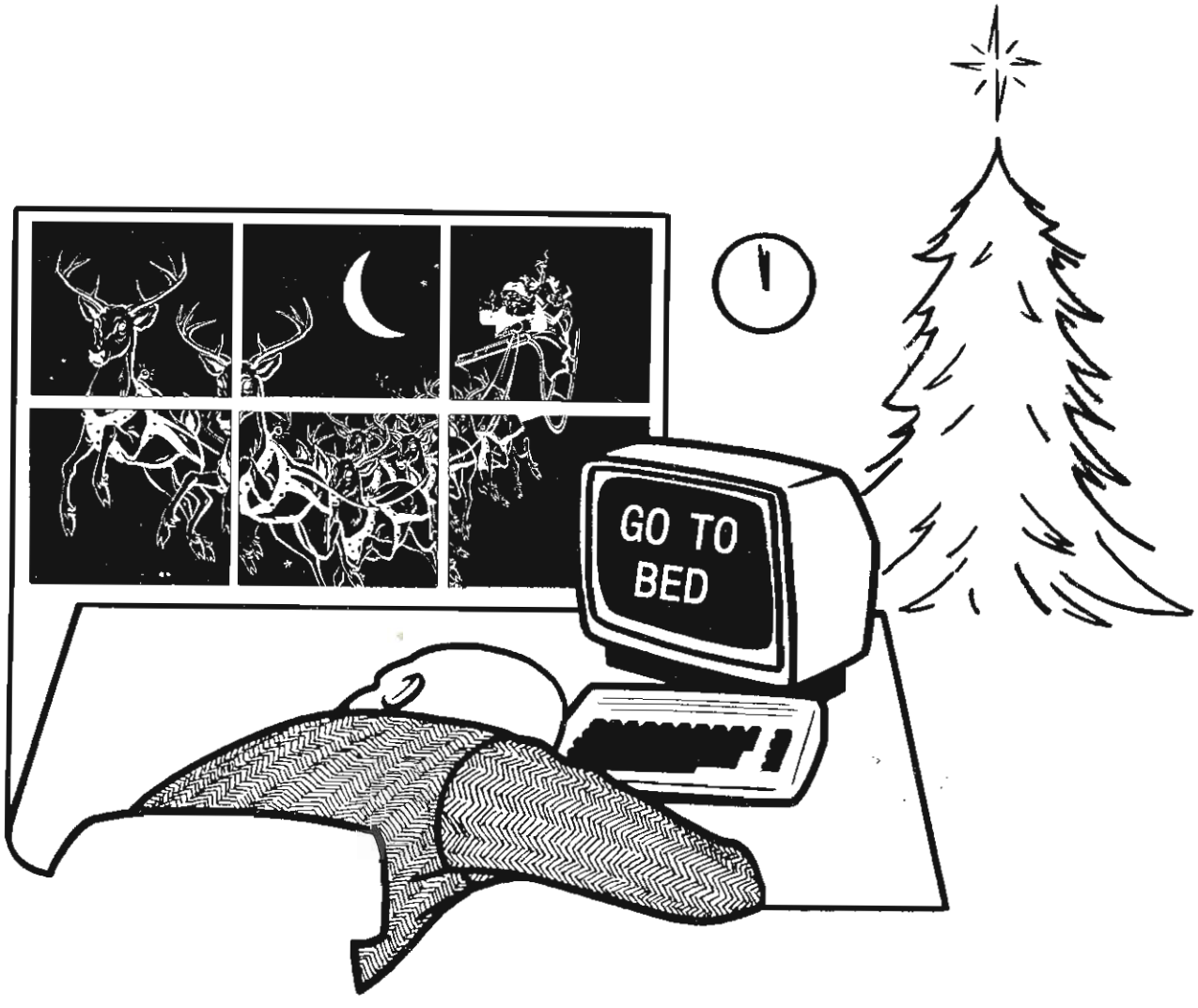
$2.00 US

ISSUE 29

NOVEMBER-DECEMBER 1985

Midnite Software Gazette

The First Independent U.S. Magazine for users of Commodore brand computers.



Published by: Micro-PACE Computers Inc.
Robert Wolter
Editor-in-Chief: Jim Oldfield Jr.
Editor: Tim Sickbert
Art Direction: Art Lewis Kimball
Assc. Editors: Jim & Ellen Strasma
Robert Baker
Dr. Richard Immers
Art Lewis Kimball
Subscriptions: 1-800-362-9653
One Year Price: $23 US $28 CDN
$33 S $43 AIR
Questions, back issues, Problems: 1-217-356-1885
BBS Number: 1-217-356-8056
(300 BAUD)
Address for Subs. And Reviews: Post Office Box 1747
Champaign, IL. 61820
Other Correspondence: 1510 N. Neil St.
Champaign, IL. 61820
Issue 29 , 1985 Nov/Dec

C128 NOTES
By: Robert W. Baker

Even with the expanded memory capacity of the C128, Basic programs are still limited to 64K. However, you now have 64K for the program and another 64K for data, both less whatever space is needed by the system. In actuality, you'll find a little more than 58,000 bytes free in program space and 64,000 bytes free in data space when you first power on or reset the system.

The program limitations exist due to the internal format of Basic programs. Remember that each Basic line is stored as a two byte link, a two byte program line number, the tokenized program line, and a single byte of zero that acts as an end of line flag. The two byte link is an address pointer that indicates the start of the next program line which has not been changed from previous versions of Basic. Since it remains as a two byte value, this limits program space to 64K, the maximum address a two byte value can specify.

When a Basic program line is tokenized, each Basic reserved word is converted to a special one or two byte value referred to as it's token. Note that I said one or TWO byte value. In Basic 7.0 many of the new keywords get converted to two byte values, unlike previous versions of Basic that used all one byte tokens. Two special token values, 206 and 254 decimal, indicate two byte tokens with the next byte indicating the exact keyword they represent together.

Anyone that's used Basic 4.0 on the older PET and CBM systems is in for a surprise. The familiar Basic 4.0 disk commands from those systems are now included in Basic 7.0, but the internal tokens used for these commands are different. You cannot load a program written on a Basic 4.0 machine that uses these commands and expect it to run on the C128. It just will not work. The commands will appear as different Basic 7.0 commands and normerally result in syntax errors when executed.

Other items to watch out for when transferring older Basic programs (including those for the C64 or C64 mode on the C128) to the C128:

1) PEEKs, POKEs, SYS and WAIT commands will usually not work for obvious reasons.
2) The subroutine address used by the USR command is now moved to 4633 and 4634 decimal, but then even the subroutine being called by this command will also have to be moved or fixed as well.
3) The FRE command now uses the parameter from the bank command, with zero used to find the free program space and one used to find the free data space. Older programs that used the FRE(0) command to check free memory space will probably have to be changed to FRE(1) to check data space on the C128.
4) LIST now lists the program lines and the program resumes operation instead of stopping and returning to immediate mode as with older Basics.
5) In addition to TI, TI$ and ST; the following reserved variables have been added and cannot be used as normal variables: DS, DS$, ER, EL. You may have to change some variable names in older programs to get them to operate properly.

As with any new machine, exercise caution when writing new

programs. There are bound to be a few bugs in the new Basic. Therefore, use common sense and make frequent backups when working on a new program. A word to the wise, a few seconds trouble may save hours of work.

I have found one problem in Basic so far: The RENUMBER command has a serious problem that could wipe out an entire program in memory. If an ON..-GOTO... or ON..GOSUB... command has a line number that doesn't exist in the program, the remainder of the program following that line will be destroyed when the program is renumbered. To protect yourself against disaster, save a copy of the program on disk before renumbering the program lines.

If you come across any other problems in the new Basic or even the new 1571 drive, drop me a note with as much detail as possible about the problem. If you can make a short example that illustrates the problem, so much the better. I'm willing to act as a clearing house to document known problems in the new systems if they begin to show up. In that way we can let everyone know about problem areas as they are found and how to work around them.

C128/1571 HINTS

By: Robert W. Baker

Here are a few quick programming hints on how you can determine the kind of system you're running on (C64 vs C128), the disk drive type (1541 vs 1571), and what mode the drive is operating in (if a 1571).

To find the system type you can try the following:

```
  Bank Select
20 POKE 65280,0
30 IF PEEK(65280)<>0 THEN
  SY=64: GOTO 100 : REM C64
  System
40 IF PEEK(215)=128 THEN SZ=80
  : REM 80 Column C128
100 REM ... main program ...
```

The poke in line 10 insures the C128 bank select is properly set for other peeks and pokes to work as expected without using the Basic 7.0 BANK command (since it won't work on a C64). This insures the program runs on all systems in any mode. The C128 does not set the BANK selection to the default value when a program is started--the program must set it before doing any peeks, pokes, etc.

The poke and peek in lines 20-30 then determine if the system is a C128 in 128 mode by modifying an unused location in upper RAM on a C128. On a C64, this location holds a non-zero value within the ROMs and cannot be modified.

If the system is a C128, then the peek in line 40 can be used to see if 40 or 80 column mode is being used. This test can be deleted if you really don't care. Otherwise, this test should be used instead of testing the 40/80 column switch directly. The switch position could have been changed since the system was powered on or reset, thus returning a false value. Basic only checks the switch once whenever the system is reset. If the switch is changed after that, it will not have any effect on the display size.

To easily find the disk drive type and mode of operation, you can try the following:

```
10 OPEN 15,8,15
```

```
20 DM=0: PRINT#15,"U0>M9"
30 INPUT#15,EN
40  IF  EN<>31  THEN  DT=0:  GOTO
  100 : REM 1541 Drive
50 DT=1: PRINT#15,"U0>H0"
60 INPUT#15,EN
70  IF  EN=31  THEN  DM=1  :  REM
  1571 Drive in DS Mode
100 REM ... main program ...
```

All drives but the 1571 ignore the "U0>M9" command in line 20 and return a zero error code, while the 1571 drive returns an error code of 31. Remember that the 1571 uses the "U0>Mx" mode select commands, so the nine is seen as an invalid parameter.

If the drive is found to be a 1571, the "U0>H0" head select command in line 50 can then be used to see if the drive is in single sided mode. This command is invalid when the drive is in double sided mode and an error code of 31 would be returned. In single sided mode, the normal head is selected and a zero error code is returned.

After this short routine runs, the variables DT and DM can be tested to determine the Drive Type and Drive Mode:

DT = 0 for 1541
= 1 for 1571
DM = 0 for single sided mode
= 1 for double sided mode

Of course these routines can be compacted by combining lines and you can use only what you need, they're only given as examples of what can be done. The commands will work with or without a diskette installed in the drive and will not change the current mode on a 1571 drive. Once the current drive type and mode are determined, the U0 commands can be used to change the drive as needed

1571 NOTES

By: Robert W. Baker

The 1571 drive has a long list of new features that make it extremely flexible but possibly confusing. From Basic, in C64 or C128 mode on the C128 system, the 1571 drive can be in either single or double sided mode. Single sided mode is identical in format to the 1541, while the double sided mode is somewhat compatible. We'll talk more about the double sided format some other time.

When the C128 is in C128 mode, the 1571 drive will normerally be in double sided mode unless placed in single sided mode under software control. When the C128 is in C64 mode, the 1571 drive can be in either mode -

- When powered up in C128 mode and GO 64 is entered, the drive stays in double sided mode.
- When powered up while holding the Commodore key, the drive goes into single sided mode when the computer goes to C64 mode.
- When the 1571 drive is used with a real C64, VIC-20, etc. the drive will normerally operate in double sided mode.
- In all cases, however, the mode of the 1571 drive can be changed under software control.

The U0 command for the Commodore disk drives has been expanded to include several new features:

"U0>M0" - selects 1541, single

sided mode
"U0>M1" - selects 1571, double sided mode
"U0>H0" - selects head 0 in 1541 mode (normal)
"U0>H1" - selects head 1 in 1541 mode (special)
"U0>"+CHR$(x) - changes the device number to x where x = 8 to 32

These commands are sent via the drive command and error channel, secondary channel 15:

```
OPEN15,8,15
PRINT#15,"U0>M0"
```

Additional burst commands are also supported via the U0 command structure but should be avoided unless you really know what you're doing. These commands allow changing the number of retries on disk operations, numerious non-standard disk format selections, plus the new high speed transfers and additional status information about the drive and disk formats.

Besides the software command to change the device number, there's now two tiny switches accessable through an opening in the rear panel of the 1571 that allow easy selection of the desired device number. There's no need to open the case and cut traces for permanent device number changes.

In both of the normal modes of operation, the 1571 drive can usually only store a maximum of 144 files per disk. The disk directory and internal buffers have not been expanded over and beyond the 1541, only the overall storage capacity has been increased from 664 to 1328 blocks free. Since the internal buffer space is unchanged, you still can only open two sequential or one relative file at the same time without getting into trouble. Also, relative files are still limited to 167,132 bytes or 65,535 records maximum per file.

The "U0>H1" command, mentioned earlier, allows selecting the second side of a normal single sided disk, and then formatting and using that side like another single sided disk on the 1571. In effect you'll have two disks in one, there will be two directories and you could store 288 files on one disk instead of only 144. However: a disk formatted like this can only be used on a 1571; there's no indication that the second side is being used; the disk cannot be used if flipped in a drive; and a 1541 cannot use the second side in any way.

1571 DOUBLE SIDED FORMAT

By: Robert W. Baker

When the 1571 disk drive is operating in the single sided mode, the drive will format diskettes with the exact same format, Block Availability Map (BAM), and directory as used on the 1541. When operating in double sided mode, the 1571 will format BOTH SIDES of the diskette using a similar format but with a few new entries in the BAM and an alternate BAM block on the second side of the disk. The second side is a mirror image of the first, with the same number of tracks and the same number of sectors within each relative track. The accompanying tables give more details on the new layout.

The state of byte three of the usual BAM in track 18, sector 0 indicates the format of the diskette. If this byte is zero, it

indicates a single sided disk. When this byte is one, the disk is a double sided disk formatted on the 1571. On either format disk, the remainder of the BAM is identical except for the last 35 bytes. Also, the directory is unchanged, starting at track 18, sector 1, with the same format and size.

On a double sided disk, the additional block availability information for the second side of the disk is stored in the last 35 bytes of the BAM. Each byte indicates the number of free blocks in each corresponding track on the second side. The information on the exact blocks that are free on the second side is stored in a new table located in track 53, sector 0. Be careful, the new table starts in the very first byte of the block. Since this block is not linked to any other block on the disk, there's no need for the pointer usually found in the first two bytes.

Note that the original block availability information was stored in a contiguous area with four bytes per track. The added information for the second side of the disk is split with the first byte of the original format placed in a table at the end of the old BAM. There are 35 bytes representing the 35 tracks on the second side of the disk. The remaining three bytes of the original format are then placed in the new table in track 53, sector 0. With three bytes per track, this table takes up the first 105 bytes of the block.

With the compatible formats, the older single sided drives can still read and display the disk and directory information from a double sided disk. The flag in byte three of the BAM is ignored by these drives, and the remaining information is identical in format plus all on the first side of the disk. However, since the single sided disk doens't know anything about the added block availability information at the end of the BAM, the number of blocks free will be incorrect when it's displayed. Always remember that the older single sided drives will treat all disks as single sided disks.

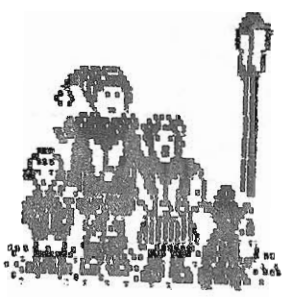
Programs and files from the first side of a double sided disk can be loaded or read on a single sided drive, as long as the file doesn't cross over to the second side of the disk. If you attempt to access any file that utilizes tracks 36 to 70, you'll receive a DOS error #66, illegal track and sector. Unfortunately, displaying the directory will not give you any indication as to what side a file uses. Until new utilities are available you'll have to access the file to find out.

When the 1571 drive is operating in double sided mode, it uses the flag in byte three of the BAM to know how to handle a particular disk. Single and double sided disks are each handled correctly and directories indicate the proper number of free blocks. However, if the 1571 drive is in the 1541 single sided mode, it will behave just like the 1541 and ignore the second side of double sided disks.

The above facts are important to keep in mind. If you format a disk and save files when the drive is in double sided mode, and then go to C64 mode on the C128 system, you may have problems. The drive will switch to single sided mode if you reset the system and hold the

Commodore key instead of typing GO 64. With the drive now in single sided mode you'll no longer be able to access the second side of the disk. Watch what you're doing!

One other parting comment concerns something I've noted on my prototype 1571 drive that I haven't been able to verify on the final production units. It has to do with the allocation of the control blocks. Track 18, sectors 0 and 1 are always allocated while the remainder of track 18 is only allocated as it is needed for the directory. For some apparent reason, the entire track 53 is allocated even though only the first sector is needed for the expanded BAM. The other 18 sectors in that track are normally wasted and unavailable. I still have to check this one out on the final version of the drive, but I've been told by Commodore that the format has not been changed.



MIDNITE REVIEW FORM

PROGRAM: _____
Author: _____
Price: _____ Media _____
Type/Application: _____
For computer: _____
Company: _____
Required Equip: _____
Optional Equip: _____
Protected? __ How? _____
Warranty? _____
Similar to: _____
Compatible with: _____

In 250 to 500 words, please describe the program, tell what you liked, what you did not like, what standard features are/are not implemented, and who, if anybody, should buy it. Then, considering how well it works, its price, and compatibility, state whether it is NOT RECOMMENDED, AVERAGE, RECOMMENDED, or HIGHLY RECOMMENDED. Finally, include your name, address, and phone number.

MIDNITE SOFWARE GAZETTE will pay $10 per review published, at the time of publication, or, upon request, credit $10 to your subscription.

Please submit completed reviews to:
MIDNITE SOFTWARE GAZETTE
PO BOX 1747
CHAMPAIGN, IL 61820
or upload it to the midnite base of STARSHIP MPC at (217) 356-8056 (24 hours, Punter)
Be timely, be detailed, but be CONCISE!

1571 BLOCK AVAILABILITY MAP (BAM) FORMAT

Track 18, Sector 0

Byte	Contents	Definition
0-1	18,01	Track/sector of first directory block
2	65	ASCII 'A' indicating 4040/1541 format
3	0	Indicates single sided format
	1	Indicates double sided format
4-143		Bit map of available blocks on first side, tracks 1-35 Entries are 4 bytes per track: byte 0 = #avail sectors in track byte 1 = bit map for sectors 0-7 byte 2 = bit map for sectors 8-15 byte 3 = bit map for sectors 16-20 in bit maps, 1 = available (unused) 0 = unavailable (used)
144-159		Diskette name padded with shifted spaces
160-161	160	Shifted spaces
162-163		Diskette ID
164	160	Shifted space
165-166	50-65	ASCII '2A' representing DOS version (2) and format type (A=4040/1541)
167-170	160	Shifted spaces
171-220	0	Nulls
221-255		Number of available blocks on second side, tracks 36-70 (1571 mode only) Entries are 1 byte per track, indicating the number of availble sectors in each track

---

ALTERNATE 1571 BLOCK AVAILABILITY MAP (BAM) FORMAT

Track 53, Sector 0

Byte	Contents	Definition
0-104		Bit map of available blocks on second side, tracks 36-70 Entries are 3 bytes per track: byte 0 = bit map for sectors 0-7 byte 1 = bit map for sectors 8-15 byte 2 = bit map for sectors 16-20 in bit map, 1 = available (unused) 0 = unavailable (used)
105-255		unused

1571 BLOCK DISTRIBUTION BY TRACK

Track#	Sector Range	Total# Sectors	
1 to 17	0 to 20	17 * 21 = 357	(side one)
18 to 24	0 to 18	7 * 19 = 133	
25 to 30	0 to 17	6 * 18 = 108	
31 to 35	0 to 16	5 * 17 = 85	
36 to 52	0 to 20	17 * 21 = 357	(side two)
53 to 59	0 to 18	7 * 19 = 133	
60 to 65	0 to 17	6 * 18 = 108	
66 to 70	0 to 16	5 * 17 = 85	
		1366 Total	

Mail all subscriptions, requests, and reviews to:
MIDNITE SOFTWARE GAZETTE
P.O. BOX 1747
CHAMPAIGN, IL 61820

Please mail all other correspondence to:
MIDNITE SOFTWARE GAZETTE
1510 N. NEIL ST.
CHAMPAIGN, IL 61820