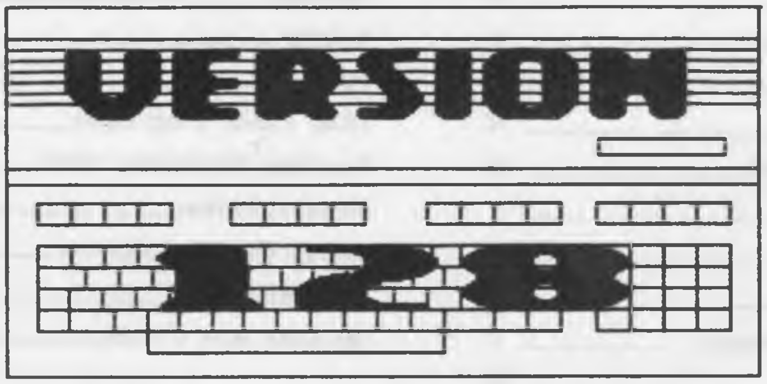
COLOR 64's



COPYRIGHT 1993
By, Adam Fanello
4822 Larwin Ave.
Cypress, CA 90630

Nature Reserve BBS
714-828-7296

Your Registraion Number: 103

Date Purchased: 7-21-94

(Add the pages of this manual to the end of your Color 64 manual.)

TABLE OF CONTENTS

THANKS!

Special thanks go to:

John Pinson of Inner Circle, whom first recognized my SysOping abilities. Additional thanks for the publicity he provided during the development of Version 128.

Rodger Hyatt of Swamp Land, whom was the first brave Color 64 SysOp to let me take a crack at programming on his board.

Jeff Terhoeve of Over Board, for winning the prize of me turning his board into a Version 128 guinea-pig.

Most of all, my thanks to YOU for purchasing Version 128!

OVERVIEW OF INSTALLATION

Congratulations! You are the owner of the ultimate upgrade to your Color 64 system! Version 128 will allow you to run your BBS in C128 mode, without going back to stock Color 64 overlays! A program called MALT will interpret each overlay and create a conversion file for it.

Version 128 was designed to work with Color 64 v7.3. If you have an older version, contact Fred Ogle for an update. I wrote V128 to be as compatible with Color 64 as possible.

Color 64 uses the check-mark character in many of its file names. Anywhere you see the check-mark in this manual (✓), press SHIFT @ on your computer to produce the character.

Converting your Color 64 v7.3 BBS to V128 is not a complicated process... however, it does take several hours. DON'T start this until you have plenty of time in order to finish. You may even want to spend a couple days on it! The following is a step by step guide to the conversion process. Details can be found later within this documentation. Read this ENTIRE manual BEFORE starting! It's short and wont take long.

A) Back-up your program files drive! This includes your overlays, bbs parameters, boot files, ML files, etc etc....

B) Copy your Version 128 disks. Put the originals in a safe place.

C) Boot your computer to c128 mode. All programs run in c128 mode unless stated otherwise.

D) Run MALT and interpret ALL of your overlays. Refer to the section entitled, MALT.

E) Merge each ALT file created by MALT to its corresponding overlay. Refer to the section entitled VMERGE.

F) Merge any additional V128 ALT files on the V128 disk to their corresponding overlays. These include "v128init.alt" for your "✓bbs.ini*" overlay. "v128m-ul.alt" for any overlay which may contain multi-upload.

G) If you run Network, refer to the section of this manual entitled NETWORK. If you run Plus Term, refer to the section of this manual entitled PLUS TERM.

H) If you run the games menu 'Mod Menu1', then merge "v128mod menu.alt" into it. If you run the "✓bbs.ict*" overlay, then merge "v128ict.alt" into it.

I) Delete your old "✓bbs.ml*", "✓bbs.pun*", "✓bbs.xmo*" and "✓bbs.setup*" files, and copy the new ones over from the Version 128 disk.

J) Run "✓bbs.setup", or "✓bbs.setup+nw" for Network users. Even if you do not wish to change anything, you MUST run setup and save a new ✓bbs.parms file!

K) If you are running RAMDOS, then refer to the section of this manual entitled RAMDOS to setup your new ramboot file.

L) Boot your BBS by RUNing the file "bbs". (Or RUN "ramboot" if you need to load your overlays into the RAMDisk.) The screen width you are in when you run "bbs", is the screen width that the BBS will run in.

HARDWARE COMPATIBILITIES

To run V128, you need a Commodore 128 computer, either the older flat model, or the c128d. A 5 1/4 inch drive is needed to read the disks that V128 comes on. An RGB (80 column) monitor is recommended, but not required. All modems, disk drives, hard drives, and RAM devices that work with Color 64 can also be used with V128. However, some storage devices have special requirements in order for them to work in C128 mode.

LT. KERNAL HARD DRIVE SYSTEM

In order to use your Commodore 128 in 128 mode with the Lt. Kernal, an additional C128 Adapter Board is required. This can be ordered from Xetec for about $30 by calling 913-827-0685.

INCONTROL DATA CHIEF

The ICT hard drive can be connected via the serial port where it can use the C128's extremely fast burst mode.

IEEE DRIVES

You need an IEEE interface that will allow you to use your Commodore 128 in 128 mode. Quicksilver 128 is the one such interface.

CREATIVE MICRO DESIGNS

Compatible with all CMD devices. The CMD HD works very fast in both serial and parallel (RAMLink) mode. You may want to buy a copy of Jiffy-Dos that works on the C128 in 128 mode. One can be bought from CMD (1-800-638-3263) for about $78.

RAMDOS 128

Version 128 has been programmed to work with RAMDOS 128, which is included with V128. Read the section on RAMDOS 128 for details.

LOADING BASIC PROGRAMS

Anytime you load a BASIC program, use either of the forms:

```
        LOAD "filename",8              or              DLOAD "filename",u8
```

Do NOT load BASIC programs with commands such as:

```
        LOAD "filename",8,1            or              BLOAD "filename",u8
```

I say this, because the C128 and C64 store their BASIC programs in different areas of memory, and the C128 actually his *two* places where it stores programs. Using the LOAD command with the number after the device number, or the BASIC 7.0 BLOAD command, results in the loading of a file to the same memory address to which it was saved. This won't work if the program wasn't saved from the same area of memory that you are trying to load it into. I strongly recommend that you get in a habit of using the correct load commands. Many people use the binary form of load, figuring it will work for all files. In this case, it doesn't.

MALT

MALT stands for Make ALTerer file. It is a stand alone (runs separate from the BBS) program that will interpret any Color 64 mod or overlay, and create an 'ALT'. This ALT file is then merged into the mod or overlay so that it can run under the V128 system.

MALT is easy to use. When you run MALT, first thing you need to do is tell it where it can find the mod or overlay it is to interpret. Then enter the name of the the file to be interpreted. Wild-card characters ARE allowed here. Also, you may instead enter a dollar sign ($), optionally followed by a search pattern. (DON'T include a colon after the dollar sign.) MALT will then read the disk directory and allow you to choose which file(s) to convert. In this way, you could choose to convert as much as an entire disk at a time. A complete log of all files converted, and any errors or warnings, is written to a SEQ file named "malt log". Be sure to always review this file before doing any merging. Next, enter the name of the file that the alterations are to be written to. The next question asks you for the *direction* of the translation. MALT can transfer either from Color 64 to Version 128, or Version 128 to Color 64. The computer will display the file name and line number it is working on during the conversion.

The actual MALT process takes from a few seconds, to a few minutes; depending on the size of the file being converted. Press ESC to abort.

MALT is quite accurate. I've tried to put every common alteration needed into it. Although rare, there are times when MALT fails to make all the conversions needed. For these cases, I and a few other SysOps have released additional ALT files to be used. You'll find several on your V128 disk.

MALT will interpret any RND() statements it can safely figure can to V128's True Random Numbers function, USR(). In converting from V128 to C64, it will change any USR() functions to their equivalent RND() commands. See the section in this manual entitled TRUE RANDOM NUMBERS. If you don't program, don't worry about this, it's automatic.

Notes:
MALT is not compatible with Jiffy-Dos or Warpspeed. They must be disabled. They DO however work with the rest of the BBS.
ALT files are 'mergables'. They are not meant to be run or have other files merged into them. The lines in an ALT file are not necessarily in numeric order!

MALT v3.1 has been compiled to Machine Language with Becker Basic, (c) Abacus Software.

RESERVED VARIABLES

BASIC 7.0 reserves four variables for its own use. These are ER, EL, ERR$(), DS, and DS$. You may NOT use the variables EL, ERR$(), DS, or DS$ for anything other than what is specified on page 328 of the C128 SYSTEM GUIDE. I have managed to trick BASIC into letting us use ER, EL, and DS. ERR$() isn't a problem because besides for this one, BASIC only uses the first two letters of a variable name anyway. Unfortunately, I was unable to pull DS$ from BASIC's clutches, so your board cannot use this variable. MALT will give you a warning if it finds DS$ in anything it is converting. This doesn't happen too often, but if it does, you'll have to change it to a different variable, or get a hold of an ALT file made for that particular mod or game.

ADDITIONAL LINES

When MALT creates an ALT file for a complete overlay, it automatically creates four new lines. An explanation of the purpose of each of these lines follows...

Lines 4 and 999:
These two lines deal mainly with error trapping. Line 4 stores the name of the current overlay into memory. If a BASIC error occurs, line 999 is called. This line will display on your screen, the line in which the error occurred. It then loads the "✓bbs.trap" overlay, which displays the error both to you, and the person on-line, then returns to the main menu.

Lines 298 and 299:
These lines are effective only when the BBS is run in 40 columns. GOSUBing to line 298 will place the computer in 2 mhz (fast mode) and set the screen output temporarily to the 80 column screen. Call this line before routines you want to speed up, and don't mind the 40 column screen being blanked out. A GOSUB to line 299 will return the output to the 40 column screen, and the computer speed back to it's previous speed. If the BBS is set to run in 80 columns, then these two lines will have no effect.

MERGING FILES

Any good BASIC merge utility for the c128 will work. However, do NOT try to use a c64 mode program (like BAID) to do your V128 merging! The c64 can not make sense of any BASIC 7.0 (c128's version) commands. I have written, and included a utility called "vmerge" which will work. It's rather easy to use. Simply load"vmerge",#,1 or bload"vmerge",u# (where # is the drive number) to ready vmerge. Load the overlay into memory, then type

```
sys5555,#:"filename
```

to merge in "filename". (Where # is again the drive number.) A quote can be added on the end, but it's not needed. If you find another C128 BASIC merge utility that you like better, by all means use it!

NETWORK WITH V128

Version 128 comes with Network 128 v1.26. This version is of course fully compatible with Network 64 v1.24 and v1.26. If you are already running Network and are converting your old Color 64 overlays to V128, do not try and convert your old Network overlays! Instead, replace your old "✓bbs.nw1*" and "✓bbs.nw2*" overlays with the new V128 Network overlays. If you where already running Network 64 v1.26, then that is all there is to it. If you where running an older version, or you are installing Network into your system for the first time, read and follow the instructions in the "READ ME" file on the Network disk that came with V128.

PLUS TERM

With permission from it's original author, Sam Lewit, a V128 version of Plus Term is included on your disk. If you are upgrading your system from Color 64 and are already running Plus Term on it, simply replace your old overlay with the new one on the V128 disk. Otherwise, unlibrary the file named "term128.sprt.lbr" and follow the installation instructions in it.

RAMDOS

Ramdos 128 V4.5 is included with Version 128. This version works with REU's of 128k to 8 megs! Because of the design of Ramdos 128, neither SPEEDY nor RAPID work properly with it. RAPID will automatically disable itself if RAMDos is detected.

There are actually two versions of RamBoot on your disk. One only works well with REU's of 512k (or less), with one source disk, and requires you to program it with the name of every file it is to copy. This version is described below. The second version, RamBoot 2, allows for multiple source directories, using wildcard search file matching to choose files. It also can bypass an error with RAMDOS when dealing with REU's of MORE than 512k. The program and detailed documentation for this version is on your first V128 disk as "ramboot2.lbr".

Below is the instructions for using the first (original) version of RamBoot:

Copy all your overlays, including "✓bbs.trap", to a boot disk. Also copy over the files: "bbs", "✓bbs.pun+", "✓bbs.xmo+", "✓bbs.ml? v128.*", and "ramdos128.bin4.5".

Now load the file "ramboot", and enter:

```
LIST -89
```

On line 10, you will see:

```
10 DATA 8,0,i0, 15,0,i0
```

Alter the first part of this line if necessary to indicate the device number, drive number, and initialization command of the disk that contains your overlays to be copied to the RAMDisk. The second half of the line is the device number, drive number, and command for the RAMDisk. Changing that information will alter what device and drive numbers your RAMDisk becomes. On lines 20 through 48 (add lines where needed), type in the EXACT overlay names of each overlay you wish to have copied to the RAMDisk. Do NOT use wild-cards in the file names. Lines 50 through 88, type in the file names of all ML files to be copied to the RAMDisk. You probably wont need to add anything here. Now save 'ramboot' back to disk.

Run "ramboot" to boot the bbs. To restore the old contents of the RAMDisk after resetting, run "ramreboot" instead.

VARIABLE KILLER

Along with the memory addresses being different, in order for a DIM Kill to work on the c128, all string variables to be killed must be cleared. Failure to do this will result in your computer going senile![1] In order to take care of this for you, a set of ML routines takes the place of the old DIM Killer. These routines will automatically clear any string variables before they are killed; preventing a variable scramble. As far as converting goes, there isn't any need to worry about the DIM Killer, MALT handles it.

When writing your own games or mods using a DIM Killer, the new code to do so is to precede the DIMs with a 'SYSC(36)+10' and proceed with 'SYSC(36)+13'. To continue with the format of the old DIM Killer, the code would look as follows:

```
  52000 SYSC(36)+10:GOSUB52020
  52010 SYSC(36)+13:RETURN
  52020 beginning of your program
```

When MALT converts V128 Dim Killers to c64 mode, the V128 SYS command Dim Killers are converted to the c64 PEEK and POKE Dim Killers. The variable 'K9' is used for the c64's Dim Killer variable. It's a good idea to check and make sure this isn't conflicting with anything in the game. (V128 does not use any such BASIC variable. It's all handled in the ML.)

Version 128 also supports what is called a Variable Killer. This routine will kill NON-ARRAY variables along with the array variables killed by a normal Dim Killer. By doing this, no-longer in use variables are removed from memory, making more room for the contents of existing variables (such as messages), and makes the system run a bit faster because there is less variables for the computer to search through.

To use, replace the old Dim killer calls, sysc(36)+10 and sysc(36)+13, with the Var Killer calls of sysc(36)+32 and sysc(36)+35 respectively. The Variable Killer routine automatically does a Dim Kill also. You *may* use both together though, it won't hurt anything, but it's redundant.

Be careful with this! If you arn't comfortable with doing some programming on your board, then don't bother with this. It saves memory and makes the BBS run slightly faster, but if used in the wrong places, it can cause problems. For example, some games set a variable which is used to flag that the current caller has already placed this call. Don't Variable Kill these games! When in doubt, *don't!* When to use it and when to not can be tricky question, so let me explain just how the routine works, so that you will have a better idea of how to use it.
The sysc(36)+32 makes the computer remember the *number* of variables it currently knows. This includes both array and non-array variables (as the old dim killer is built in). The game or utility now does it's thing, modifying old variables, and most likely, creating new ones. After the game or utility has finished executing, a sysc(36)+35 will make the computer 'forget' any NEW variables it has added to memory, so that the number of variables in memory is the same as it was before the game or utility started. It does NOT however, forget any CHANGES to variables it already knew.

Due to the complexity of determining if a Variable Killer can be safely used, MALT does not make use if it when converting Color 64 games and utilities to V128. All Color 64 Dim Killers are converted strictly to V128 Dim Killers.

1. Variable scrambles are caused by a problem with a DIM killer. Scrambles have a delayed reaction. It won't show itself until after the problematic routine... sometimes it may not even be the same caller on-line at the time. A variable scramble is easy to recognize, as many string variables (hold characters) and arrays (bundle of numbers) will turn into a bunch of nonsense! If you ever see this problem (not likely so long as you use MALT), check your DIM Killers.

TRUE RANDOM NUMBERS

BASIC's random number generator is based on a mathmatical formula! As a result, it's far from random. this is a problem with all computers though, and is commonly just accepted.

The Commodore 64 and 128 however, has SID! The Sound Interface Device's voice number three has the ability to generate *truly* random noise. V128 taps into this hidden resource to give your BBS truly random numbers.

To use this in your mods and games, simply replace:

```
                INT(RND(0)*n)        with            USR(n)
```

Where n, in both the RND and USR functions, is the highest integer number generated, plus one. In other words, they will return a number between 0 and n-1. You don't have to worry about the details though, it is just a direct replacement. Also remember, the 0 within the RND() function is a dummy value. It can be any number or variable.

If all this is starting to make you regret buying V128, hold on! You do NOT have to go through all your overlays changing the RND's! You MAY change it in some mods and games for truly random selection and play, if you wish.

Two things to watch out for: Avoid adding USR() right after sound is played. Because True Random numbers uses the sound chip to work, any sound being played when it is called will be cut off. Second, the USR() function is limited to returning integer values between 0 and 255. If a larger number is needed, you'll have to stay with the original RND() function.

MALT will automatically make this change for you when converting files.

RAPID

Version 128 has the ability to run at 1.2mhz on the 40 column screen. This gives a 20% speed increase without the screen blanking. This can be done because 20% of the time the computer is updating a screen which is outside the parameter of your monitor screen. During this time, RAPID kicks your computer up to 2mhz. During the other 80% of the time, the computer is updating the actual 40 column screen you see, and RAPID moves back down to 1mhz. This combined effect results in 1.2mhz. This effect also often results in flickering on your screen during disk access. The amount of flicker depends on the type of disk drive. If you find this annoying, replace line 105 in "bbs" and "reboot" with the following line to disable RAPID.

```
105 POKE 4095,0
```

RAPID only goes into effect if you run your board in 40 columns. 80 column systems will run constantly in 2mhz. Also, when RAPID is active, the BASIC command FAST will still put the computer in full 2mhz and blank the screen. The SLOW command will return to 1.2mhz.

Rapid does NOT work with RamDos, and will be automatically disabled. Some have also reported problems with it and CMD cartridge devices. If you run one with your system in 40 columns and have trouble, disable RAPID.

I highly recommend an 80 column monitor. The 0.8mhz more does make a big difference, and many V128 mods only work in 80 columns.

CONTROL KEYS

Version 128 added several new CONTROL-key commands. The following is a complete listing of these new keys, along with the old ones from Color 64. Most of these commands can be entered either from the keyboard while online, or in the output text in a SEQ file or program overlay.

CONTROL-A: Turn rainbow on.
CONTROL-B: Change background color if followed by a color key, otherwise turn underlining on for c128 systems in 80c.
CONTROL-C: Cycle to previous system color.
CONTROL-D: Cycle to next system color.
CONTROL-F: Turn character flashing off.
CONTROL-G: Produce a bell sound.
CONTROL-K: Copy character entered at current column position on the last line entered. (For line entry only.)
CONTROL-O: Turn character flashing on for c128 users in 80c.
CONTROL-P: Abort current operation.
CONTROL-S: Pause (Same as HOME key.)
CONTROL-U: Delete back to the beginning of the line.
CONTROL-V: Repeat line being entered. (To check for line noise.)
CONTROL-W: Delete back to last space. (Word delete.)
CONTROL-X: Delete back to the beginning of the line.
CONTROL-Z: Turn rainbow off.

SYSTEM COLORS

In Color 64, the function keys are used in program code to turn rainbow on and off, and to cycle to the next system color. The character codes produced by the function keys on the c64 will still have the same effect on V128. However, due to the fact that the function keys are programmable in BASIC 7.0, an alternate method is necessary. The following is a list of alternative key combinations to produce the same effects as the c64 function keys. (The old c64 equivalents are in parentheses.)

CONTROL-A: Turns rainbow mode on. (F1)
CONTROL-Z: Turns rainbow mode off. (F3)
CONTROL-D: Cycle to next system color. (F5)
CONTROL-C: Cycle to previous system color. (None)

These codes are also available for use on the board in the message base.

If you wish to disable the programmable function keys when programming in BASIC 7.0, enter:
POKE 828,183
To enable the programmable function keys, enter:
POKE 828,173

ALT-KEY COMMANDS

Version 128 has a few sysop commands that are activated by ALT key combinations. First, from the wait-for-caller screen, holding the ALT key alone for a second will bring up 'snapshot' screens. These screens are duplicates of both the 40 and 80 column screens from when the last BASIC error occurred. This allows you to see what was going on on the screen at the time of the error. Press any key to return to the wait-for-caller screen.

V128 has added a couple new features to use while in CHAT mode. Both these default to an active mode when you boot the board, and can be toggled off and back on while in chat mode.

ALT C Toggle Auto-Color Changing. With this feature active, anything you type will be shown in your first system color (as defined in bbs.setup) and anything the user types will appear in the second system color. This feature helps both you and the user keep track of who said what.

ALT G Produce a bell sound on YOUR end only. Used to adjust the volume on your monitor without disturbing the current caller.

ALT W Toggle automatic Word-Wrap in chat.

ALT 0 Clear the Local Buffer. See the section of this manual entitled LOCAL BUFFER.

ALT + Open Local Buffer. See the section entitled LOCAL BUFFER.

ALT - Close Local Buffer. See the section entitled LOCAL BUFFER

ALT @ Dump the Local Buffer to screen. See the section entitled LOCAL BUFFER.

From anywhere within the BBS, you may enter ALT #, where # is a number between 1 and 8. These are your programmable function keys! They are defined using the BASIC-7.0 KEY command. You can read about this command on page 269 of your Commodore 128 System Guide. You can put KEY commands to define your programmable keys in the BBS and REBOOT programs, or within your overlays themselves! The BBS and REBOOT files included have the ALT-1 key programmed as an example. There is one limitation to keep in mind when making your keys, each key can be no longer than 20 characters long. Anything after 20 characters will be chopped off.

Note: ALT key combinations are entered by pressing and HOLDING the ALT key, while tapping the letter or number for the command.

LOCAL BUFFER

With the Local Buffer, you can buffer text from anywhere in the bbs, just as you would if calling from a terminal program. You can also dump the buffer anywhere in the BBS (usually a message editor). A command from DOS will allow you to load and save to and from the buffer.

From anywhere in the BBS, press ALT + to open the buffer, ALT - to close the buffer, ALT 0 (that's a zero) to clear the buffer, and ALT @ to dump the buffer at the current location. You can abort a buffer dump by pressing ALT -.

If you wish to be able to load to and save from the buffer, simply merge the file "buffer-dos.mrg" from your V128 mods disk into your DOS overlay. If you run a separate DOS overlay, then merge into "√bbs.dos*". Otherwise, merge into "√bbs.xfe*".

To access the Local Buffer options, go to DOS from the BBS. (Press F7 twice from the wait-for-caller screen, or usually the command is '>' from the main menu.) At the prompt, enter **buff**:

```
8> buff
```

A short menu will appear giving you the options to Load or Save, as well as reminding you of the ALT key commands. This all works rather straight forward.

The Local Buffer is stored in the same space as the BASIC overlays. So, the smaller your BASIC overlay, the more buffer space you have available. Anytime the board loads a new overlay, there is the possibility that the new overlay loaded in will overwrite the end of the buffer. This possibility is minimized by having the buffer start at the top of memory and work down, while the overlays are loaded from the bottom, up. You can figure out the amount of buffer space you have by subtracting the overlay size (in blocks) from the number 190. Lastly, don't worry about your overlays becoming corrupt. The buffer will automatically close if it gets close to running into the overlay in memory, even if you switch overlays with the buffer still open.

ANSI

Version 128 has the incredible ability to translate Commodore color, graphics, and cursor movement into the most similar ANSI code. The results are surprisingly (and sometimes humorously) similar to the original PETSCII picture, animation, or plain text. Just to make sure you know what your users are experiencing, a custom ANSI character set is loaded in when an ANSI caller logs on! The only difference between what you see and what the caller sees is the colors. You will continue to see Commodore colors, because that's what your monitor supports... the caller will see the eight colors that standard ANSI supports.

These ANSI abilities are already built into the V128 ML, all you need is the ability for users to select it. This is handled by the Gfx Type mod. To install, unlibrary the file "gfx type.lbr" from the V128 MODS disk and follow the instructions within it.

Some Color 64 on-line games check to see if the user is in Commodore Gfx mode, or ASCII, and will not allow an ASCII user to play. Many of these games can be played in ANSI. The file named "need c/g or ansi" on your V128 disk is an example line of how to change these games to allow EITHER Commodore Gfx users or ANSI users to play. In many cases, all you'll need to do it change the line number and merge it in.

SPEEDY

Speedy increases text file display by approximately two times. (And at 2mhz, this will quadruple your speed over Color 64!) Speedy is actually the normal file read routine, plus PAUSER, rewritten COMPLETELY in machine language! Unfortunately, SPEEDY runs into some trouble with RAMDos and other RAM devices... users won't be able to pause or abort! So I don't recommend you use Speedy with these RAM devices (REU, RAMLink, RAMDrive).

SPEEDY is already built-in to the ML. All you need to do is tell BASIC to use it! Merge "speedy-msg" into your "✓bbs.msg*" overlay, and "speedy-others" into every other overlay in which you want faster text file display. (ALL of them?!)

PAUSER

Pauser is automatic screen pausing, that won't interfere with graphics or text animation! Users have the choice of having pauser active or not, and their choice is saved to their account file so that it remains set properly from one call to the next. The merge files are as follows:

"autopause-ini" - merge into "✓bbs.ini*" and, if you are using it, into "✓bbs.pwm*".
"autopause-msg" - merge into "✓bbs.msg*". This merge isn't needed if you are also using SPEEDY.
"autopause-others" - merge into all other overlays you want autopausing in. This merge isn't needed if you are are also using SPEEDY.
"autopause-set" - merge in wherever it fits, and point either a spare command or a sub-menu option at it. This is where users may choose if they want Pauser active for them or not.

MCI V128

MCI stands for Message Command Interpretor. This system has been popular on C-Net and Image boards for many years. Attempts at similar things have been attempted for Color 64 before, but they have been very limited and slow. MCI V128 is built in to the ML, allowing for quick use without the limitations of BASIC. Unlike other BBS system which use MCI, V128's MCI is not used as the primary method of I/O for the system. Rather, it functions more as a 'toy' for the callers.

To install MCI, merge "mci-init.mrg" into your "✓bbs.ini*" overlay, and "mci-msg.mrg" into your "✓bbs.msg*", "✓bbs.xfe*", and if you have it, merge "mci-dos.mrg" into "✓bbs.dos*". Then copy "@help mci" to your help files drive, put it on the help files menu, and rename it.

CHAT MODE MESSAGES

Version 128 allows for three customizable sets of messages when entering and exiting chat mode. To use, RUN the program on the first disk named "✓chat ed".

First think you will see are prompts asking you for the device and drive number, as well as drive command, of the disk with your "✓bbs.ml0 v128.*" file. Enter all the correct data. Next, you are prompted for the update version number of your ML (the last two digits on the ML files).

After a few seconds, you will be able to edit the chat mode entry and exit messages. Enter RETURN or use the UP and DOWN cursor keys to move between the messages. The editor is pretty straight forward, just play with it.

When you're done editing, press the ESC key. From there you can choose to resume editing, save your changes, or abort without altering the old chat messages.

Notes:
No message may be more than 40 characters long. This counts any CONTROL characters, color changes, and the following characters that Chat Ed adds automatically:
To the beginning of each chat mode entry message, a CLR (clear screen) CONTROL-Z (rainbow off), CONTROL-D (cycle next system color), and RETURN is added.
On both the chat mode entry and exit, a CONTROL-D and a couple RETURNS are added on the end.

CUSTOM CHARACTER SETS

Version 128 allows for the use of a custom character set on the 80 column screen. This feature DOES NOT take away any of your free memory. When the c128 is turned on, the 80 column character set is copied from ROM to a section of RAM that can only be accessed by the 80 column video chip. Your custom character set it simply copied on top of the old character set.

The custom character set files are to be on the same disk as your overlays. If your using a RAMDisk, be sure that the custom character set files are copied over. The custom character set files are to be named "✓charset1" (containing the lower/upper case character set) and "✓charset2" (containing the upper/graphics character set). If you do not wish to use a custom character set, simply exclude both files from your disk. A sample set of ✓charset files is included on your V128 MODS disk.

PICTURE MAKER

Picture Maker is a full screen text/graphics picture editor, allowing free cursoring about the screen. Picture Maker ONLY works in local mode, it would just be too slow on-line. Picture Maker is also limited to pictures of only 24 lines in length.

To install, merge the file "pic-dos.mrg" into whichever overlay your DOS is in. What I mean by that is that, if you have a "√bbs.dos*" overlay, then merge "pic-dos.mrg" into there. Otherwise, merge "pic-dos.mrg" into your "√bbs.xfe*" overlay. With that done, copy the "√bbs.pic+" and "√bbs.pic-ml v2.0" files from the V128 MODS disk, to your overlays disk (and add it to "ramboot" if your using it).

Now boot your BBS and go to DOS. Enter the backarrow key (under the ESC key) and press RETURN.

Commands:
T - Text mode. Lets you type and draw on the screen. You may use CURSOR, TAB, and ESC key combinations (see page 370 of the Commodore 128 System Guide). Tapping HOME twice resets the current window back to the full screen. Press CONTROL-P to exit.
L - Load a file to screen. It will ask for the drive specs, and then for the destination. Move the cursor to the top-left corner for the picture.
S - Save the screen to a file. The computer will ask for the top-left and bottom-right coordinates and the file specs. If the file already exists, it will be replaced.
4 - Switch to the 40 column screen.
8 - Switch to the 80 column screen.
C - Change between the lower/upper case character set, and the upper/graphics character set.
Q - Quit
SHIFT CLR/HOME - Clear the screen.

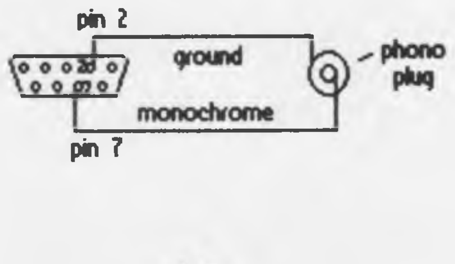
An independently running (outside the BBS system) version of Picture Maker 128 is also included. This version is Public Domain, so you can have artistic users make pictures and menus for you in both 40 and 80 columns.

NOTES ON 80 COLUMNS

Displaying 80 Columns On A Composite Monitor:

The c128 80 column RGBI video can be viewed on a composite video (40 column) monitor! On a monochrome composite monitor, the picture sharpness is just as good as an RGBI monitor. However, on a color composite monitor (such as the 1702) the picture isn't very good... in fact it takes quite a bit of getting used to. It will also always be monochrome, even from a color monitor.

A special cable is required to make this connection. One can be bought from some stores which carry Commodore hardware, or through mail-order. Or, you can build your own cable according to the following schematic:



Colors On The 80 Column Screen:

Several colors are different on the 80 column screen than they are on the 40 column screen. You can experiment and find most of these differences yourself. However, one color- medium grey, can be an annoyance. On some RGBI monitors, medium grey is SO dark, you will need to turn your brightness up to see it. (The 1902 doesn't have this problem.) On a composite monitor which has been jury-rigged for 80 columns in the above described method, medium grey will be BLACK!

GENERAL NOTES

During file transfers, the data appears on the 40 column screen only. If you run in 80 columns and wish to view the data, you must switch to the 40 column screen.

MALT can NOT properly convert overlays that have been crunched. Hope you have backups!

"√bbs.ini*" no longer has to be the largest overlay. The only limitation is that no overlay can be more than 190 blocks.

Time normally freezes in chat-mode. This is usually preferable, but at times you may want to deduct the time in chat instead. Pressing F4 to exit chat mode will result in the user's time being deducted.

If you ever decide to sell your copy of V128, you will need to write me a *signed* letter stating to whom you have sold it to. If you don't, the new owner will not receive any technical support or updates.

UPDATES and MODS

Updates and new mods are available for download on my BBS:
Nature Reserve BBS 714-828-7296
When you log on, remind me (Handle: Ant) that you are a Version 128 owner, so I can give you the correct access.

You may also receive updates by sending me a disk in a self-addressed stamped mailer. Include your registration number with your request and *sign* it.

This manual was written for Version 128.306. You can find documentation for any updates since the printing of this manual by running the *first* file on the *first* side of the *first* disk which came with V128. (This should be the *first* thing you do after reading this manual!)