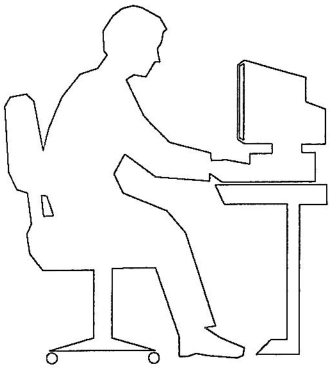
Texas Instruments Brisbane User Group (TIBUG)



BUG BYTES

June/July 1997

Editor's Note

Dennis Remmer
dennis@dstc.edu.au

Didn't I just do one of these? :)

It's June already and there's a chill in the Brisbane air, but that's OK.

The next meeting will be held at Chas Bagley's house, 2 Dakar Rd, Algester, on Friday night 30th May, starting at 7:30p.m.

This issue there are heaps of hints and tips to get the most out of the V9T9 TI emulator, as well as some technical information on TI bitmap mode. I've included a complete solution to the Scott Adams adventure Return to Pirate's Isle, and some neat on-line Adventure resources.

An excellent article on Disk Drive technology from a recent Micropendium is reproduced, and lastly I've found some weird and wonderful (and very funny) bits and pieces for your enjoyment.

Best Regards...

V9T9 Tricks and Tips

Brian Tristam Williams
POLAR@global.co.za

(Editor: These are various postings to the TI Mailing List and Newsgroup, which should be of general interest.)

Win95: 'MS-DOS Prompt' Context-Menu Addition

You load up Windows Explorer, and look at your V9t9 directory. You have the directory visible in Explorer, but you'd like to be there to do something in DOS, like run one of the PC programs in the UTILS directory. The default way would be to click on **Start -> Programs -> MS-DOS Prompt**. You'd then be thrown into Windows' default directory. To get to V9t9, you'd have to remember where V9t9 was (on your drive), then type something like:

```
CD \APPS\UTILS\V9T9
```

...to get there. OK, easy enough, but this can get really tedious to do over and over again. The solution: How would you like to be able to right-click on any directory in Windows Explorer's left-hand-pane, then click on MS-DOS Prompt, and be dropped off in the directory of your choice?

Here's how to do it:

- In Windows Explorer, go to the menu and click on **'View'** -> **'Options'**. Then go to the File Types tab.
- Scroll down to the registered File-Type named **'Folder'**, click on it, then click the **'Edit...'** button.
- You will see the **'Edit File Type'** dialogue. Click on the **'New...'** button. The **'New Action'** dialogue pops up.
- In the **'Action'** field, enter **"MS-DOS Prompt"**.
- In the **'Application'** field, you type **"C:\Win95\command.com"** *(Note, however, that Win95 is the name of MY Windows 95 directory - you will need to replace this with your own directory's name - usually "WINDOWS".)*
- Click on the **'OK'** button to close the **'New Action'** dialogue, close the **'Edit File Type'** dialogue using the **'Close'** button, then close the **'Options'** dialogue, using the **'Close'** button.

From now on, you should be able to go to the directory of your choice by right-clicking on it (in Windows Explorer's left-panel) and clicking on 'MS-DOS Prompt'. You can switch this prompt

CONTENTS

between a window and full-screen using Alt-Enter.

To take this a bit further: If you would like to run a DOS command, such as DIR, before going to the command prompt, do this:

- Back in the **'Application'** field of the **'New Action'** dialogue, instead of the **"C:\Win95\command.com"** typed in earlier, type in: **"C:\Win95\command.com /K DIR"** *(remembering to use the name of your Windows directory instead of 'Win95').*

This will run a DIR of the directory you specified, then turn the command prompt over to you. Using this technique, you could even run whole batch files before you enter the MS-DOS prompt, by substituting "DIR" with the batch-file's name. (The /K switch instructs Command.Com to turn control over to you when it is done executing the DIR command. You could use the /C switch, which tells it to simply run the command and end the program. For a complete list of switches, type in **"COMMAND /?"** at the DOS prompt, in your Windows directory.)

Win95: 'TI Directory' Context-Menu Addition

You look at V9t9's DISK directory in Windows Explorer. You see several TI files, but you have no idea whether they're DV/80, PROGRAM, INT/VAR, or whatever. What you would normally need to do is run the MS-DOS Prompt (made easier with the last solution) in the V9t9 directory, and type `UTILS\TIDIR DISK`, in order to get a directory of which TI files are available, how many sectors they are, and, especially important, what TYPE of files you have there. A hassle.

This next step enables you to right-click on the V9t9 FIAD directory (such as DISK), then click on **'TI Directory'** in the context menu in order to get a complete directory, in .TXT format, in Windows' Notepad application. This is how you do it:

- Firstly, you need to create a batch file, named TICAT.BAT, and save it to a directory in your PATH, such as **'C:\Windows\Command'**.

- To do this, open the Notepad program, and type in the following batch program:

```
@echo off

C:\APPS\UTILS\V9T9\utils\tidir /L %1 /OT > C:\WIN95\temp\ticat.txt

notepad C:\WIN95\temp\ticat.txt

erase C:\WIN95\temp\ticat.txt
```

- with the following modifications: Replace **"C:\APPS\UTILS\V9T9"** with the name of the directory in which your copy of V9t9 is housed.

- Replace **"C:\WIN95"** with the name of your Windows 95 directory. So, if your copy of V9t9 is in the directory **"C:\V9T9\V6.0"** and your Windows directory is **"C:\WINDOWS"**, the program would actually look like this:

```
@echo off

C:\V9T9\V6.0\utils\tidir /L %1 /OT > C:\WINDOWS\temp\ticat.txt

notepad C:\WINDOWS\temp\ticat.txt

erase C:\WINDOWS\temp\ticat.txt
```

- Save this file in a directory such as **"C:\WINDOWS\COMMAND"**, by going to Notepad's menu, clicking on **'File' -> 'Save As...'** and typing in **"C:\WINDOWS\COMMAND"** in the **'File name:'** field of the 'Save As' dialogue.

- Now that you have that down, you need to point to it from your Explorer context-menu. Do it like this: In Windows Explorer, go to the menu and click on **'View' -> 'Options'**. Then go to the **'File Types'** tab.

- Scroll down to the registered FileType named **'Folder'**, click on it, then click the **'Edit...'** button. You will see the **'Edit File Type'** dialogue.

- Click on the **'New...'** button. The **'New Action'** dialogue pops up. In the **'Action'** field, enter **"TI Directory"**.

- In the **'Application'** field, you type `"C:\WINDOWS\COMMAND\TICAT.BAT"`

(Note, however, that WINDOWS is the default Windows 95 directory - if yours differs from this, you will need to replace this with your own directory's name.)

- Click on the **'OK'** button to close the **'New Action'** dialogue, close the **'Edit File Type'** dialogue using the **'Close'** button, then close the **'Options'** dialogue, using the **'Close'** button.

Okay, that's it. If everything went according to plan, you should be able to right-click on a V9t9 FIAD directory, and 'TI Directory' should appear in the context menu. Clicking on this should start Notepad, which will show you the directory of TI Files, which might look something like this:

```
Directoryof C:\DOSAPPS\UTILS\V9T9\DISK

Filename   Secs   Type      Len  Bytes
DRAWYUV    3      PROGRAM          748
HEXDUMP    3      PROGRAM          618
QUIKPROG   1      PROGRAM          162
CODES      6      PROGRAM          482
A          6      DIS/VAR   80     516
```

...and can be saved or edited at will, or you can simply close the program. The file that is created is erased before you even start reading it, so don't worry about files building up on your hard drive. Obviously, using this technique in another directory which has nothing to do with V9t9 won't yield very interesting results.

Win95: Reading DV/80 Files with Notepad

Here's one for you: you've just used the last solution to get a directory of a V9t9 FIAD, and you see a DV/80 file, named (for argument's sake) "MYFILE" you're interested in reading. The first option would be to start V9t9 and load up a text-editor, and look at the file. However, you might want to look at the file in an 80 (or more) -column editor such as Notepad. If you simply opened the file with Notepad, it would look corrupted, due to the different methods that the TI and PC use for saving text files. So, usually, you would have to go into V9t9 under the MS-DOS Prompt (made easier with the first tip), and type in `"UTILS\TI2TXT DISK\MYFILE DISK"`. This would convert the file "MYFILE" to the DOS-compatible text-file "MYFILE.TXT". Then you'd have to exit the DOS

Prompt, then start up Notepad, load up the file, read it, then close Notepad and go back and delete the file. And all you wanted to do was take a look at it! Wouldn't it be nice if you could look at such a file with Notepad, with the correct formatting, after two double-clicks?

First, you need to create a file named `DV80READ.BAT`, which looks like this:

```
C:\APPS\UTILS\V9T9\utils\ti2txt %1 .
NOTEPAD %1.txt
erase %1.txt
```

making the following change: replace **"C:\APPS\UTILS\V9T9"** with the name of YOUR V9t9 directory. Create this file by starting Notepad, and typing it in. After you've typed in this program, save it to a directory which is in your MS-DOS PATH, such as **"C:\WINDOWS\COMMAND"**. Do this by going to Notepad's menu, then clicking on **'File' -> 'Save As...'**, and typing `"C:\WINDOWS\COMMAND\DV80READ.BAT"`, then closing Notepad.

OK, you've done that, but how do you get it to run when you choose the DV/80 file you need to view?

- Well, first you need to make DV80READ.BAT appear in your **'Open With'** dialogue box. This is the dialogue that pops up when, in Windows Explorer, you double-click on a file with an extension that is not associated with any application.
- In order to do this, open notepad, type in a word or two, then save the file as **"C:\1.!"** This will put the file **"1.!"** in your root directory, for now.
- Close Notepad, then go into Windows Explorer, and double-click on this file.
- You will get the **'Open With'** dialogue, most likely. Click the **'Other...'** button, then find the directory of DV80READ.BAT by double-clicking on **"Windows"**, then **"Command"**, then "`dv80read.bat`".
- This will select this file and close the **'Open With...'** file-selection dialogue box. Close the **'Open With'** dialogue box by clicking the **'Close'** button. You will then get an error message. Click the **'No'** button, and close Notepad.
- You can now delete the "**1.!**" file - it is no longer necessary.

Now you can go to a V9t9 FIAD directory such as DISK, double-click on a known DV/80 file, and it should pop-up in Notepad. You can then save this file anywhere else, in DOS text-file format.

There is one instance when this method won't work, though: If your DV80 file has any characters which DOS wouldn't accept, such as a file named "FW/DOCS", which could not contain the '/' character. V9t9 will change the '/' character to one the PC would accept, and this would confuse this little batch file. Also, if your filename is longer than eight characters long, it won't work.

V9T9 - Soundblaster DMA Usage

Tyler D. Vantighem
tvantigh@freenet.calgary.ab.ca

To everyone on how to get the +SBDMA option to work.

If you dig deep enough into the docs you come across that you need the CT-VOICE.DRV to use the +SBDMA option. For this you need a real sound blaster card with the proper drivers.

To see if you have the proper drivers check in your sound blaster directory SB??\DRV (?? for specific SB card) for the CT-VOICE.DRV. If you see two files called CTMMSYS.SYS and CTSB??.SYS (?? for specific SB card) a little more work is need to setup your system for V9T9.

If you have the required files the following modifications are need to be made to your CONFIG.SYS and AUTOEXEC.BAT files:

<u>CONFIG.SYS</u>

Found just CT-VOICE.DRV
- nothing to do

Found CT-VOICE, CTMMSYS.SYS and CTSB??.SYS?
- add:

```
DEVICE=C:\SB??\DRV\CTSB??.SYS /UNIT=0 /BLASTER=A:220 I:5 D:1 H:5
DEVICE=C:\SB??\DRV\CTMMSYS.SYS
```

- (A:220 = Your SB Port Address)
  (I:5 = Your SB Interrupt Address)
  (D:1 = SB Low DMA Address)
  (H:5 = SB High DMA Address [SB16 and up])

- if you have EMM386 installed you can load them into the UMB area by using DEVICEHIGH.

<u>AUTOEXEC.BAT</u>

- add:

```
SET SOUND=C:\SB??
SET BLASTER=A220 I5 D1 H5 P330 E620 T6
```

- (A220 = Your SB Port Address)
  (I5 = Your SB Interrupt Address)
  (D1 = SB Low DMA Address)
  (H5 = SB High DMA Address [SB16 and up])
  (P330 = MIDI Port Address (optnl))
  (E620 = SB32+ WT Synth address)
  (T: Type of SB Card:
  1=SBv1.0
  3=SBv2.0
  2=SBPROv1.0
  4=SBPPROv2.0
  6=SB16/SB32(SB64??))

- add:

```
SET MIDI=SYNTH:1 MAP:6 MODE:0
```

(works for SB16/SB32 not sure for SB/SBPRO)

- for SB16/SB32 add:

```
C:\SB??\DIAGNOSE /S
C:\SB??\MIXERSET /P /Q
```

(could also be SB16SET)

- for SB32 add:

```
C:\SB??\AWEUTIL /S
```

Note: SB32 options are for SBAWE32 also and maybe SBAWE64 users also.

I hope this helps everyone out.

SCSI — MFM — RLL — WHAT?

TIers keep up with changes in storage technology

By MICHAEL J MAKSIMIK

This article first appeared in the newsletter of the Chicago TI User Group. —Ed.

The incredible leaps in technology and demands for newer, faster hardware have become the new military-industrial complex. Old technology falls to let new technology grow and flourish. It is a good thing that our community has adjusted to this and has accepted some of the newer technologies in disk storage. But what is new, what is old, and what is the difference?

Hard disks, as you know, were in existence for years before microcomputers used them. Huge MFM drives on the IBM System/36 (the size of car tires) were used in the late 1970's at incredible cost by today's standards. In fact, some model 5365s (about the size of a dishwasher) are still in operation. Yes, they still run, and run on 220 volts! That much current is needed to keep the disk drives spinning. Those drives emit gravity when they spin!

Eight-inch hard disks were created using a standard which had existed in the form factor of the 8-inch floppy drive. It was an industry standard and microcomputer makers began seeing this as an affordable mode of storage. The standard was FM, Frequency Modulation, also used in FM radio. FM was based on two signals — one signal was a timing signal, a "clock," and the second signal was the data signal. By reading both signals we could store blocks of data onto the magnetic surface, using a magnetic "north" for a zero and a magnetic "south" for a one. The clock would be active at the center of every data cell, so we could "sample" the data line at precise intervals, thereby getting our data. FM was the standard for storage for many years. Single density floppy I/O is still done using FM encoding.

Later, we developed MFM, which embedded the clock signal within the data stream. Ones and zeroes were arranged in such a way that the clock could be extracted from the data stream itself, which also meant we could use the extra space on the drive for data (the *extra* space was used in FM encoding by the clock signal) thereby giving birth to "double" density. In reality, there isn't any doubling of density. In fact, the exact same number of flux transitions occur on double-density media. The difference is that only data is stored, no clock signal.

RLL encoding came later. RLL stands for "Run Length Limited," a technique which solved neighboring bit-cell corruption problems inherent in MFM storage, and at the same time increasing capacity of storage to about 1.5 times the normal capacity of MFM. RLL drives are still used today, but are disappearing in favor of more advanced encoding methods. We can write data into a space using multiple phases which allows 2, 4, 8, or even 16 bits in the same space that 1 bit would have taken. This technology allows a single platter to store what would have taken eight or more platters to store earlier.

NEW ENCODING TYPES

We are also seeing the emergence of new encoding and media types. Up to a few years ago, we just saw magnetic media, either with flexible media, as in floppy disks using a contact-type head, or rigid platter-type media with a flying head. Modifications of this method are used in tape drives (tapes can be streaming or blocked and accessed like floppy drives). I was even discussing with my brother, Tom, about using a VCR to back-up hard disk drives (this was actually used several years ago). In the VCR, the head rotates and the data is recorded in candy-stripe style diagonally across the width of the slow-moving tape. This method allows a high volume of data in a small area.

Lately we have seen new types — optical, magneto-optical, and RAID technology. With optical media, the bit cells are sampled using sensitive LED laser pick-ups which read microscopic pits on the surface of a smooth aluminum/polymer disc. When writing the data, a powerful laser is used. To read the data, a weak laser is used.

In magneto-optical, the media is still read using a magnetic head, but it is much smaller and more sensitive than normal read/write heads. To write data, the media is heated using an optical laser beam. This melts the surface of the media slightly, which allows the magnetic pattern being written to be "frozen" into the surface when the beam stops. I saw the inspiration for this when doing some research about earth plate tectonics (ocean surface). The earth's magnetic field was "frozen" in ferro-magnetic ores deposited on the volcanic rifts of the Pacific Ocean floor over thousands of years. This clearly showed

1997 TI FAIRS

APRIL

Fest West '97, April 5, San Jose Civic Auditorium, San Jose, California. Contact Fest West '97 c/o Don O'Neil, 3297 Woody Lane, San Jose, CA 95132, or call (408) 934-0352.

MAY

Multi Users Group Conference, May 23-24, Ohio State University, Lima Campus. Contact Charles Good, P.O. 1ox 447, Venedocia, OH 45894. Phone (419) 667-3131. Preferred e-mail address good.6@osu.edu.

This TI event listing is a permanent feature of MICROpendium. User groups and others planning events for TI/Geneve users may send information for inclusion in this standing column. Send information to MICROpendium Fairs, P.O. Box 1343, Round Rock, TX 78680.

DISK DRIVES —

(Continued from Page 29)

reversals of the earth's magnetic field many times. While these molten ores were solidified, they recorded the current state of earth's magnetic field. We can replicate this in a magneto-optical drive, with more practical results.

"Floptical" media is optical-guided media, where a beam is used to guide the read/write head in smaller increments thereby allowing more storage on the media. As we machine parts smaller and more precisely, we will be storing voluminous amounts of data in the tiniest of spaces. There is even research into storing data in crystals. By using lasers within crystals, we can record an electron-excited state within a unit cell of a crystal and, later, we can read it because the unit cell would change the polarization of light as it reflects. And since there are billions of unit cells in a crystal the size of a dime, we can readily imagine the amount of storage of this newly emerging technology.

Well, we know the various media types and their encoding strategies, but how are they connected to the computer?

COMMON INTERFACE TYPES

The interface types we have seen most frequently over the past few years are: ST506, ST412, ESDI, IDE, and SCSI.

ST506 and ST412 are similar to the Shugart floppy drive interface. The Shugart interface was a standard 5.25-inch drive interface (still used today) to allow control of up to four drives, FM or MFM, with clocking controlled by the media itself (hard-sectored) or the data (soft-sectored).

Hard-sectored disks have an index hole for every location where a sector should start. The index hole would trigger the start of bit cell sampling and that would control timing and synchronization of the disk media.

Soft sectoring used a special bit pattern which surrounded each sector, and that bit pattern would tell the disk controller where a sector began and where it ended. While a Shugart interface was fine for floppy disks (the signal to the head was single ended, that is, only one line was needed to read the bit cell and a logic level determined the value of the bit cell), we needed something better for hard disks.

Hard-sectored disks have an index hole for every location where a sector should start. The index hole would trigger the start of bit cell sampling and that would control timing and synchronization of the disk media.

Since hard disks moved data much faster than floppy disks, we had differential I/O for the reading and writing. In differential, we use a current loop to transmit the data to and from the read/write head. Current loop was immune to the problem of induction of signal logic, which affected single ended lines. ST506 and ST412 are very similar in this respect and, for the most part, are interchangeable. However, ST412 was used with 8-inch hard disk drives, while ST506 was not. ST506 is what we used for our Myarc HFDC controller. Many RLL and MFM drives use this format for system interface.

ESDI was a standard which was implemented in the IBM PS/2 and some clones. The drives were 5.25-inch (some 3.5-inch also) and it was a higher speed interface — typically it could handle enhanced RLL encoding methods and transfer speeds at double what MFM rates were. However, the interface was not compatible with ST506 or ST412. It was processor specific to PS/2 models and was not general enough for the rest of the computing world to adopt.

Integrated Drive Electronics (IDE) is popular for MS-DOS machines, and is specific to the architecture of Intel-based processor boards. IDE drives used a command-based structure, combining an intelligent 16-bit parallel interface with a disk drive. By placing a computer on board the drive itself, the system could relieve itself of the burdens of disk controlling and just issue commands to read/write, etc, and it would let the drive do all the dirty work. IDE now allows for other devices (such as CD-ROM and tape drives) to be attached to the Intel-based processors. However, because the standard of IDE is closely tied to signal timing and bus behavior of Intel-based processors, it is difficult to implement on other platforms (including our own 99xx-based processors). The IDE standard has been expanded and, through the use of replicate interfaces, you can attach multiple devices onto an IDE bus. Still, many IDE adapters on PCs are limited to two drives on the bus.

SCSI IS VERSATILE

SCSI was developed in the mid-1980s as a peripheral interface for small computers (hence the name Small Computer Systems Interface). Through commands and handshaking, it could be used to exchange data between peripherals, between host computers, or a combination of both. Peripherals can communicate with each other (as in an unattended tape backup of a hard disk) or in a RAID disk unit (where multiple disk drives maintain small copies of each other, thereby providing data redundancy). Hosts can exchange data with each other if they were connected on a SCSI bus using high speed transfers, and it can provide a flexible interface for parallel processing.

SCSI today has emerged into new implementations. The first, SCSI-1, was used in 1986 to provide disk systems for Apple Macintosh computers, file servers, and high-end computers. Later, SCSI-2 was introduced as a way to refine the command structure of the interface, and to provide more standard commands on compliant devices. It also introduced faster timing methods and defined more device types. SCSI-3 is already out and defines wide data interchange for high transfer rates (40 megabits per second).

SCSI is implemented as an 8-bit parallel interface. There are many more control signals used than in a printer interface, and it is bidirectional, that is, data can be sent and received. Also, the data bit lines are also device select lines, and there are different negotiation phases used to provide an orderly flow of data between devices.

There are also device priorities, and each device can itself control eight sub-devices, called logical units. A single SCSI bus can control 64 devices, including the host device and it's locally attached logical units. SCSI allows connection of disk drives, hard disks, removable media drives, printers, tape drives, network adapters, and other devices.

SCSI hard disks can use MFM, RLL or other encoding methods. Yes, you can have an MFM drive with a SCSI interface instead of an ST506 interface. Most SCSI drives are using RLL or multiphase storage encoding, few are using MFM (exept the older drives you can find at computer fairs). SCSI tape drives come in many varieties (streaming, rack, helical scan, reel-to-reel, 3490 cartridge, quarter-inch cartridge, etc.), but the nice thing is that they all share a common command set which makes them all somewhat interchangeable as far as data storage and retrieval is concerned.

SCSI is used to allow printers to get font information quickly from the host while leaving the parallel port free for print data. It is also used for CD-ROM drives and the abundance of cartridge-type media now available (Syquest, ZIP drives, EZ 135 drive, etc.).

SCSI offers several advantages over other media interfaces:

- It is non-computerspecific. This way, it is not tied to any chipset or manufacturer, so it can be easily implemented on any computer platform.
- It uses a high level protocol for data exchange, making your data very secure.
- SCSI devices are intelligent, often controlled by their own microprocessor. This relieves much of the burden of media I/O to the device's onboard hardware, so your computer can do more interesting things. It also means much faster transfer rates.

Our computers (the TI99/4A and the Myarc 9640) can use media types from FM diskettes to SCSI hard disks. We have adjusted to accept the newest media types, inexpensively. SCSI is used in IBM AS/400 midranges and mainframes. It is used in Apple computers and IBM computers, and now it is used with our computers as well. For those who can get them larger than 40 megabytes, the Myarc HFDC can control up to 3 MFM hard disks, and it is the only controller which can control high density floppy drives.

Few MFM hard disks survive in the open market nowadays, but they are still out there and they offer inexpensive storage. MFM drives are a bargain at computer fairs. If you aren't concerned about getting a used drive, or about size (MFM drives are hard to get in sizes larger than 40 megabytes) you can go for that storage media. The HFDC is still for sale and is an adequate system. It offers complete compatibility with the TI and the Geneve.

The SCSI market is expanding wide open, and you have many choices as far as media is concerned. The Western Horizon Technology SCSI controller for the TI/Geneve works well and is easy to hook up. A single cable bus connects all drives together and drives are available in fixed or removable format. In fact, I use the EZ-135 drive. Each of its cartridges yields 127 megabytes of formatted data storage. It is fast (as fast as any SCSI-2 buffered drive) and is cheap for the storage size (the drive is about $120, the cartridges run about $20). The SCSI bus can be shared between the TI and the Geneve, so you can use both computers on the bus simultaneously, sharing drives and media. SCSI has a great future ahead of it because it is ready for the faster processors on the market.

What size should I buy? Well, for now, our file system supports 31 sectors worth of bitmap, at a maximum of 16 sectors per allocation unit. This translates into 248 megabytes total storage per drive. This should be plenty for our uses since programs on the TI/Geneve are compact and we don't use a lot of space for program data.

On PCs, the story is different. The operating systems are graphically based, and many use virtual memory storage. This increases disk requirements on PC's to values higher than 248 megabytes, so we will see many hard disks in the 1 gigabyte or higher range, and fewer in the 248-800 megabyte range.

If you use a TI or Geneve, it really is not a good idea to invest in a very large drive. Save the money and get a cartridge drive, and just buy lots of cartridges. First of all, your storage requirements on the TI will not exceed 127 megabytes at a time. Trust me, you don't have that much data. If you decided to store the contents of software libraries on hard disk, well just separate them into 127-megabyte segments and store these groups on separate cartridges. Rarely will you need to access it all at once.

Storage media seems to be going in two directions: SCSI and IDE (or E-IDE, enhanced-IDE interface). SCSI is clearly at an advantage, because it is supported across the entire computer industry and is already available in 16- and 32-bit versions for wider bus requirements. E-IDE is suitable for the PC segment of the market but it will always be tied to that market. The market determines a final, single standard eventually, and E-IDE will be the next dinosaur. The market determined a standard for PCs and left all other machines in the dust. I am glad that the bandwagon we hopped on was the right one.

Competition produces cassette check program

Competition Computer has produced a new cassette-based checkbook manager for the TI99/4A. Kyle Crichton of the company explains that TI's Checkbook Manager was disk-based, and that the company responded to the needs of cassette-only users.

The program sells for $5, with a $1 shipping and handling charge. Contact Competition Computer at 350 Marcella Way, Millbrae, CA 94030, information (415) 697-1108, orders 1-800-471-1600.

Crichton also says he expects the company, which produces an IBM-compatible CD-ROM, to introduce some new products, both hardware and software, at Fest West in April.

Bitmap Mode Demystified!

Tom Bagget / Jeff White

tommyb@tapscan.com / jhwhit01@ulkyvx.louisville.edu

(When I first started studying the 9918A a year or so ago, I searched high and low for detailed information on it. I ultimately found a very enlightening message on Delphi from Jeff White that was originally dated Sept. 10, '93. I thought I would pass this along to you, since he definitively covered every aspect of the 9918A, MUCH better than I've seen it covered anywhere else.)

<excerpt follows>

First, the E/A manual is wrong, but the 1 page addendum explains many errors in it. Included is the correction for bit-map mode register settings.

```
Pattern descriptor table:

at >0000-17FF, r4 = >03
at >2000-37FF, r4 = >07

Color table:

at >0000-17FF, r3 = >7F
at >2000-37FF, r3 = >FF
```

"Now why did TI do this?" you may ask. The simple explanation is that the bits in the registers correspond to video RAM address bits. Bit 2 (bit 5 in TI's big-endian numbering scheme) of r4 corresponds to the most significant bit of the video RAM address. Bit 7 (or TI's bit 0) of r3 corresponds to the most significant bit of the video RAM address as well.

The value of the pattern descriptor table base address is the value in r4 of the meaningful bits shifted by 11 (or multiplied by 2048). The color table base address is the value in r3 of the meaningful bits shifted by 6 (or multiplied by 64).

For bitmap mode, the meaningful bit of r4 gives a value of 0 or 4. Multiply by 2048 and you get 0 or 8192 (>0000 or >2000), respectively. The meaningful bit of r3 gives a value of 0 or 128. Multiply by 64 and you get 0 or 8192 (>0000 or >2000), respectively.

Designing the TMS9918A (and V9938/ V9958) this way simplifies the routing of the register values to address lines on the die. The other option would require routing different bits of r3 and r4 as the most significant bit of the video RAM address.

The "unused" bits of r3 and r4 less significant than the used bits should be set to 1. This is so that the address lines these less significant bits correspond to do not have their values masked to 0.

When you put >00 into r4, you were making A13, A12, and A11 of the pattern descriptor address always 0. This meant when the VDP was fetching >0000-07FF, it fetched the first 2048 bytes. Also, when the VDP was fetching >0800-0FFF, it was fetching >0000-07FF. Likewise with >1000-17FF.

For the 9918A, the video address bits correspond to register bits as follows:

```
r3 = A13 A12 A11 A10 A9 A8  A7  A6

r4 = 0   0   0   0   0 A13 A12 A11
```

Now you can do some weird things if you know that when you set one of the "unmeaningful" bits of r3 or r4 during bitmap mode to 0 you are in fact ANDing the corresponding address lines when said registers are used for a display fetch with 0, or effectively masking them to 0. So, you can cut the color table to 2048 bytes by setting A12 and A11 of r3 to 0, or putting the values >1F or >9F into it depending on where you want the base address to be.

Using this undocumented functionality is what our good friend Eric Lafortune did in "Rock Runner" to get great graphic effects. I have also used this functionality to write a 64-column terminal emulator. With a full 6K pattern descriptor table, and 2K color table, there is 8K of video RAM left. This is enough to hold 8 screen image tables which can be quickly flipped for a scroll effect, with only the patterns for lines 7, 15, and 23 (new line) needing to be re-drawn in the pattern descriptor table.

Of course, I could cut the color table down to 1K by masking an additional address bit (A10) to 0. In fact, I can cut it down to 64 bytes by making all of A12 to A6 bits in r3 to 0. Load all 64 bytes with the same two color code, such as >1F, and the entire screen will be black and white. However, just defining those 64 bytes is actually just defining the colors for 8 pixel lines of each 8 characters.

Isn't this all very interesting? You can have a complete monochrome bitmap display with 6K for patterns, 64 bytes for colors, and 768 bytes for pattern position (screen image table). With the fast scroll, I was using in my 64x24 column terminal emulator eight separate screen image tables which must be put on 1K boundaries non-conflicting with pattern descriptor table or color table.

The pattern descriptor table is put at >0000-17FF by loading r4 with >03. The color table is put at >2000-203F by loading r3 with >80. Two screen image tables are located at >1800-1AFF (r2=>06) and >1C00-1EFF (r2=>07). Six more can be located at >2400-26FF (r2=>09), >2800-2AFF (r2=>0A), >2C00-2EFF (r2=>BB), >3000-32FF (r2=>0C), >3400-36FF (r2=>0D), and >3800-3AFF (r2=>0E).

To do the scroll takes some fairly complex programming, but setting up the eight screen image tables is a big and easy part of it. The first screen image table has >00-FF repeated three times. The second screen image table has >20-FF,>00-1F repeated three times. The third screen image table has >40-FF,>00-3F repeated three times. See the pattern? The eighth and last screen image table has >E0-FF,>00-DF repeated three times.

The tricky part, and the part that slows the code down, is re-drawing the 8th, 16th, and 24th lines after a screen image table page flip. This is because the first line in the middle third of the screen must be scrolled to the last line of the top third of the screen. The first line of the last third of the screen must be scrolled to the last line middle third of the screen. The last line of the last third of the screen must be drawn over with the new line.

Of course, I have written complex code to handle all this special scrolling. The 64 columns looks quite good with the right font. There is sufficient memory left of the 16K video RAM for disk I/O buffers and limited sprites. A TE can do without sprites (none in text mode TE's).

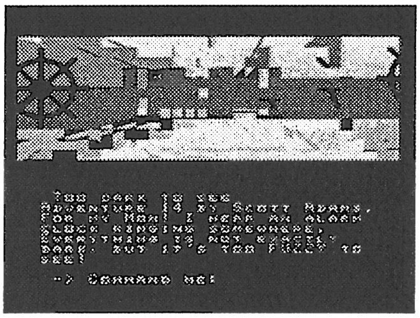
There would be 256 bytes after each of the first seven alternate screen image tables. After the color table would be 960 contiguous bytes. After the eighth alternate screen image table would be 1280 contiguous bytes. A CALL FILES(2) equivalent would have room in that 1280 bytes to handle disk I/O.

How to Solve Scott Adams Adventure#14: Return to Pirate's Isle

Dennis Remmer

dennis@dstc.edu.au

I wrote this many years ago and found the file! Thought I'd share the solution to this difficult adventure. Obviously you shouldn't read it if you don't want to know... At the end of the article are some interesting notes about an Internet site for Adventurers.



To make things easy, I'll go location by location with a description of each.

1. The BUNK - the bunk in the cabin has two levels. The TOP has 2 items, a *DIAMOND WATCH* and a face mask. The BUNK has a mattress which, when lifted, holds a book which is self explanatory. Move the mattress and reveal a *DIAMOND RING*.

2. The SHIP's CABIN is filled with goodies. The fan can be stopped by first looking at the ceiling to find a button, opening the porthole, and pressing the button. Keep looking out the porthole until the floodlight becomes dim, and immediately press the button again. The fan will have stopped, but you'll have enough power to start the engine later. Examine the fan first and then unscrew the fan blade with the screwdriver you find in the ENGINE ROOM on top of the ENGINE. Unscrew the frame of the painting and examine said painting a few times to see what you have revealed. All of this activity should give you (amongst other things) a *DIAMOND BROOCH* (from the fan) and a *REMBRANDT PAINTING* underneath the original painting.

3. Outside the porthole is a WALKWAY on the ship. Trying to go through the porthole will be difficult, so before you attempt it, drop all items in the cabin (including your glasses). On the walkway you will be disoriented without your glasses so try squinting. This will reveal an oilskin raincoat which comes in handy later. Go into the sea, and then go to the boat again, which will put you on the TOP DECK. Go below, store the raincoat, wear your glasses and get the fan blade.

4. Make your way onto shore and dig on the BEACH BY THE HILL. With the rum in your possession, climb the hill and jump up from the LEDGE. At the TOP OF A SMALL HILL, examine the sleeping pirate twice and then give him the rum. Wake him and he'll head off to work. In the crack you'll have to use the knowledge you gained in Pirate Adventure to make your way around. In case you don't have it, here it is: go crack, go toolshed, get hammer, go north, go crack. Jump ledge and go back to the ship's cabin with your goodies. You'll have a *GOLD EARRING* and a box (don't open it yet) containing *RARE STAMPS*.

5. Having dropped off your stuff, go back to the BEACH BY THE HILL, and head east from there. On the ROCKY BEACH pick up a rock. Go for a swim, and remember your sequence for swimming: hold breath, swim down, do your stuff, swim up. In this case what you have to do underwater is feel the silt, swim back up and put your glasses back on. If you examine what you now have, you'll reveal *DOUBLOONS*, and the pumice rock. Examine the rock to reveal algae (something Funori going on here). Examine the algae to reveal a *PIECE OF AMBER*. Head back to your cabin.

6. Drop off what you have. Follow these instructions: get mask, make glue from the algae, remove lenses from your glasses, glue lenses onto your mask. A snazzy mask for swimming underwater!

7. Make you way onto the DOCK and into the SEA. To the east should lie the dock. Spit into your mask to prevent it fogging up. Hold breath, swim down, swim east, swim up. You're under the dock. Climb the beam and you're ON A BEAM UNDER THE DOCK. You'll find a *SILVER DOLLAR* and a *DIAMOND PIN* on the beam. Go back to the ship (remember your swimming sequence) and drop off your stuff.

8. Sailing the boat is tricky and takes precise timing. Get the map. Go down and crank the engine. Go up and sail the boat.

9. Night has fallen. Go down and stop the engine immediately. If you don't have more than 18 litres of fuel, you're in trouble. You are now at sea with Pirate's Isle off in the distance. Under the sea here is a sunken ship which contains a *JEWELLED CHEST*. Follow these instructions and you should get back in one piece: drop map, press button, go up (presuming you're in the cabin), so sea, hold breath, swim down, go ship, get chest, swim up, go boat, go down, press button, drop chest, get map, go down, crank engine, go up, sail ship.

10. Hopefully you made it. Now is the time to find your last treasure and the place to store them. Enter the sea from the deck, swim down, west, and up. Feel the boat, swim into the opening and enter the underside of the ship. You've found the SMUGGLER'S HOLD. From here drop everything before entering the NARROW CRAWLWAY. Go into the HIDDEN SMUGGLER'S HOLD and put that infernal alarm clock to sleep. Here is a sign with a purpose! Take the sign back with you to the SMUGGLER's HOLD (the one with the pool). Drop the sign here, and you've got a place to store your treasures.

11. Put your mask back on, and enter the pool. Swim down and feel the silt, to reveal an oyster. Take it and make your way over the boat to the underside of the dock. Swim down from here and look east to see one of the wooden pilings on the ship's hull. Examine it to reveal a

snail. Swim up and open the oyster with the snail. A *PEARL* ! Head back to the cabin.

12. Time to store your treasures. If you swim with the painting unprotected it will ruin, so waterproof it with the oilskin raincoat. Grab what you can and make your way back to the SMUGGLER's HOLD. Drop them and go back for more. Once you've dropped everything in the Hold, you should open the box to reveal the stamps, and you can open the chest for the last treasure, a *RARE BOOK* which you may read at your leisure. Lastly, unwrap the painting to store your 13th treasure. Type SCORE!

Well Done!

Note that an excellent on-line resource for Adventure aficionados can be found at
`http://www2.rz.hu-berlin.de/inside/angl/people/pdd/advent.html`

This page has all sorts of links, including one to the home page of Scott Adams himself! Also available is a PC/Mac/Amiga public domain Scott Adams Adventure interpreter (ScottFree), source code, and data files. Solutions as well. Great stuff. The page has stuff other than Scott Adams resources, including Infocom resources, adventures playable on-line (!), and other adventure systems.

Weird Theories!

Omni Magazine

(Editor: Somebody emailed me this very funny list of the latest scientific theories...)

Results of a contest for "theories" sponsored by Omni magazine:

GRAND PRIZE WINNER

When a cat is dropped, it always lands on its feet, and when toast is dropped, it always lands with the buttered side facing down. I propose to strap buttered toast to the back of a cat; the two will hover, spinning inches above the ground. With a giant buttered cat array, a high-speed monorail could easily link New York with Chicago.

RUNNER-UP #1:

If an infinite number of rednecks riding in an infinite number of pickup trucks fire an infinite number of shotgun rounds at an infinite number of highway signs, they will eventually produce all the world's great literary works in Braille.

RUNNER-UP #2:

Why Yawning Is Contagious: You yawn to equalize the pressure on your eardrums. This pressure change outside your eardrums unbalances other people's ear pressures, so they must yawn to even it out.

RUNNER-UP #3:

Communist China is technologically underdeveloped because they have no alphabet and therefore cannot use acronyms to communicate ideas at a faster rate.

RUNNER-UP #4:

The earth may spin faster on its axis due to deforestation. Just as a figure skater's rate of spin increases when the arms are brought in close to the body, the cutting of tall trees may cause our planet to spin dangerously fast.

HONORABLE MENTION:

The quantity of consonants in the English language is constant. If omitted in one place, they turn up in another. When a Bostonian "pahks" his "cah," the lost r's migrate southwest, causing a Texan to "warsh" his car and invest in "erl" wells.

True Signs

(Editor: More amusing email from out there...)

Signs that have really been posted.

At a Santa Fe gas station: "We will sell gasoline to anyone in a glass container."\

In a New York restaurant: "Customers who consider our waitresses uncivil ought to see the manager."

On the wall of a Baltimore estate: "Trespassers will be prosecuted to the full extent of the law. --Sisters of Mercy"

On a long-established New Mexico dry cleaners: "38 years on the same spot."

In a Los Angeles dance hall: "Good clean dancing every night but Sunday."

In a Florida maternity ward: "No children allowed."

In a New York drugstore: "We dispense with accuracy."

In the offices of a loan company: "Ask about our plans for owning your home."

In a New York medical building: "Mental Health Prevention Center"

On a New York convalescent home: "For the sick and tired of the Episcopal Church."

On a Maine shop: "Our motto is to give our customers the lowest possible prices and workmanship."

At a number of military bases: "Restricted to unauthorized personnel."

On a display of "I love you only" Valentine cards: "Now available in multi-packs."

In the window of a Kentucky appliance store: "Don't kill your wife. Let our washing machine do the dirty work."

In a funeral parlour: "Ask about our layaway plan."

In a clothing store: "Wonderful bargains for men with 16 and 17 necks."

In a Tacoma, Washington men's clothing store: "15 men's wool suits, $10. They won't last an hour!"

On a shopping mall marquee: "Archery Tournament -- Ears pierced"

Outside a country shop: "We buy junk and sell antiques."

In the window of an Oregon store: "Why go elsewhere and be cheated when you can come here?"

In a Maine restaurant: "Open 7 days a week and weekends."

On a radiator repair garage: "Best place to take a leak."

In the vestry of a New England church: "Will the last person to leave please see that the perpetual light is extinguished."

In a Pennsylvania cemetery: "Persons are prohibited from picking flowers from any but their own graves."

On a roller coaster: "Watch your head."

On the grounds of a public school: "No trespassing without permission."

On a Tennessee highway: "When this sign is under water, this road is impassable."

Similarly, in front of a New Hampshire car wash: "If you can't read this, it's time to wash your car."

And apparently, somewhere in England in an open field otherwise untouched by human presence, there is a sign that says, "Do not throw stones at this sign."

TI Brisbane User Group

Committee

President:	Rex Jones 3396 1662
Vice-President:	John Campbell 3351 3107
Secretary:	Garry/Col Christensen 3888 4857 / 3284 7783
Treasurer:	Val Jones 3396 1662
Editor:	Dennis Remmer 3870 5710 dennis@dstc.edu.au
Disk Librarian:	John Reynolds 3357 9758
Tape Librarian:	Bruce Campbell 3351 3107
Book Librarian:	Trevor Campbell 3351 3107
Module Librarian:	Bruce Campbell 3351 3107

Contributions to TIBUG are invited from both members and non-members. Articles for inclusion in the succeeding bi-monthly newsletter are required at least two weeks before the monthly meeting and may be included in that newsletter at the discretion of the Editor.

Most original articles by members of TIBUG in this newsletter are on available on disk and are available to other User Groups on request.

Submissions of articles, reviews, comments and letters from members is encouraged, however the Editor asks that those submitting keep the following in mind:

Submissions should be about the TI Community in particular, computers in general, or of sufficient general interest. The preferred media is computer file, preferably in ASCII (Text) or Microsoft-Word compatible format, submitted on MacIntosh or IBM-compatible floppy disk or via Electronic Mail to the Editor. Handwritten submissions are acceptable but please remember that they have to be retyped. Other submissions, such as typed, printed or photocopied are welcome but must of reproducible quality.

Submissions are best sent directly to the Editor:

Dennis Remmer - TIBUG Editor
PO Box 30, Toowong QLD 4066
(Email: dennis@dstc.edu.au)

TI Brisbane User Group
PO Box 3051
Clontarf MDC, QLD 4019
Australia