SORCERER'S APPRENTICE

INTERNATIONAL COMPUTER USERS NEWSLETTER

PAGE 109

VOLUME 4

NUMBER 6

Sep. 15, 1982

IN THIS ISSUE -

Random I/O

By Don Gottwald

As all of you already know - Ralph LaFlamme has gone on to more important matters and will not be able to continue as Editor. Yours truly has been elected again to fill the void until we can arrange for someone else to take over. I have already relinquished many of my duties to other people - you'll no doubt hear from them in the near future.

The reason for the dot matrix printing is, Ralph's computer is down again and he was therefore unable to print this issue. If his computer is repaired in time for the next issue - we will again have the nice print. I've decided to run this issue in full print, without reduction to assure better readability. The next issue will again be a double issue, so we can get caught up and stay that way for next year.

I would like to thank Mike Patterson of California for keying in some of the articles in this issue, despite the fact that he was moving at the time. Thanks are also in order for all the people who have contributed articles in the last few weeks. Please don't be upset if your article did not make this issue - it'll probably be in the next one. We still need more articles so we can have enough available for editing, which will enable us to be on time with each issue.

Many of you have responded very positively to my last article. We will continue to provide the newsletter as long as the interest is there.

Challenge Systems Co. of Richardson Texas, indicated that three companies are currently negotiating for the manufacturing rights to the Sorcerer. A company in India, and a company in Peru will probably be manufacturing the Sorcerer in the near future. The third negotiations are with the designer of the Sorcerer II for the rights in North America. All the people involved appeared to be very confident that an agreement can be reached very soon. I will keep you posted.

Jerry Rude, 13730 W. Park Dr., Magalia, CA 95054 wants to contact people who are interested in software dealing with Electrical Engineering, as well as Civil and Mechanical Engineering. He's also interested in finding a supplier of TTY parts in the Northern California area.

Jerry Chapin of 74 Garfield Ave., Colonia, NJ 07067 represents a growing group of Sorcerer owners living in Northern New Jersey, all with a 48k Sorcerer. They would like to establish contact with other Sorcerer users and programmers to exchange information and to try to overcome the software shortage for the Sorcerer.

A SPEECH SYNTHESIZER FOR THE EXIDY SORCERER

Using the Votrax SC-01

by Dave Trzcinski
15516 Centralia
Redford, Mich. 48239
(313-534 2993)

Soon after obtaining a used Sorcerer and becoming familiar with it's operation, I decided it was time to begin working on a peripheral device. First I wanted to build something that I could use later to improve my machine language programming skills. Second it had to be reasonably simple to interface, since this was my first try. Finally I wanted it work with the parallel output port. A local electronics parts supplier had a listing in his catalog for a single chip speech synthesizer. At the time I hadn't heard of the Votrax SC-01 but I did know Votrax was involved in early research of electronic speech synthesis. The idea of interfacing a speech synthesizer with the Sorcerer seemed like such an interesting project that I drove over to the supplier and got all the data he had on the SC-01.

The Votrax SC-01 synthesizer is a single 22 pin CMOS chip that synthesizes the sounds of human speech by producing sounds called "phonemes" (said foneems). Phonemes are the fundamental sounds that we use in producing speech. The SC-01 contains the coding required to produce 64 of these sounds. Also on the list are codes for 2 no sound wait words, and 1 stop word.

From the block diagram (see fig. 1, pg.113), it looked like interfacing the SC-01 with my Sorcerer would be easy. First the device will accept the phoneme coding from 6 of the 8 available output port data lines. The 2 remaining high bits can be used for optional voiced inflection. As expected there were a set of lines for data handshake. One of these lines was used to strobe the data into a holding register, the second was used to return the handshake (to the computer) for busy or next phoneme conditions.

Figure (2, pg. 113) shows the data strobe and acknowledge/request handshake sequence. When the A/R line is high data may be placed on the bus. When the SC-01 receives a strobe pulse, the data byte is locked into it's data register and the A/R line flips low. This condition is maintained until the phoneme output has been completed. It then returns high, ready for the next code. Hopefully I could figure out how to match these requirements with the lines available from the output port.

To my delight, the chip had a built in RC clock, meaning I didn't need a special xtal. The simplest setup used a resistor and capacitor for timing. The audio output arrangement of 3 lines looked strange at first, but one figure in the data sheet diagrammed simple connections to an audio amplifier chip. Only one obvious problem remained. A single supply voltage was required, but the minimum was 9 volts instead of the 5 volts available from the output port. Well for now I guess I could steal power from the 12 volt serial port (and it was going so smoothly).

After 3 evenings wire wrapping and unwrapping, adding and subtracting logic chips, the circuit shown in fig. (3, pg. 113) resulted. Data is fed to the SC-01 through a DP8304 bus driver. The driver doesn't effect the data, but provides insurance in case something happens to the cable connecting the computer and the synthesizer. The audio output feeds an LM386 through a 20 K level control pot. With an 8 ohm speaker, the volume produced is loud enough to be heard several rooms away. One shots (#1) and (#2) take the down going true signal (not data available) from the Sorcerer and convert it to the up going (strobe) input required. One shot (#1) is required to provide the delay needed to allow the phoneme data to reach the register in the SC-01 before it is held by the strobe pulse from one shot (#2). When the phoneme has been completely sounded the Request line then goes high. This fires one shot (#3) which produces the down going true pulse to reset the data avail-

cont'd on page 112

able flip flop in the Sorcerer. The software monitors this change and begins sending the next phoneme code. Bypass capacitors are provided to help keep each power voltage clean. The 39 k resistor from the SC-01 A/R line to ground, provides a pull down for the one shot. I tried a voltage divider and a series drop resistor arrangement here (since I am driving 5 volt logic with 12 volts), but this worked the best.

I built my synthesizer on a small piece of perf. board to which I epoxied a 30 pin connector edge from a scrap PC board. I try to standardize on connector pin assignments, because after I get an S-100 box built, I'll just plug the board in.

Program #1 shows a Basic listing for saying the alphabet. Lines (2) thru (9) are for editing and printing purposes. Beginning at 10, decimal data is read in. It is then placed on the output port (output 255,A). Statement 40 (WAIT 254,64,191) causes the sorcerer to monitor the data accepted line for a reset. If you stop program execution while a phoneme is in progress the phoneme will not stop, typing (OUT255,63) <CR> will turn off the sound.

For demonstrations I run program #2. This program uses the synthesizer to describe my system to those present.

I paid $55 dollars for my Votrax SC-01 (see note 1), which at first seems like a lot for a chip, But for the ease of installation and programming the price isn't that bad. If requested, Westland Electronics will provide a dictionary filled with word to phoneme code conversions. I found the dictionary very helpful for encoding words that weren't even listed.

Program #1:

```
5   REM--THIS IS "ABC"
6   REM--SPEAK ABC'S
10  READ L
20  IF L=63 THEN RESTORE
30  OUT 255,L
40  WAIT 254,64,191
50  GOTO 10
55  REM-ONE LTR/STATEMENT
60  DATA 6,33,41,62
70  DATA 14,60,41,62
80  DATA 31,60,41,62
90  DATA 30,60,41,62
100 DATA 60,41,62
110 DATA 2,1,29,62
120 DATA 30,26,60,41,62
130 DATA 62,62
140 DATA 6,33,41,42,16,62
150 DATA 21,0,9,41,62
160 DATA 30,26,0,6,33,41,62
170 DATA 25,0,6,33,41,62
180 DATA 2,0,35,24,62
190 DATA 2,1,12,62
200 DATA 2,1,13,62
210 DATA 52,53,55,62
220 DATA 37,60,41,62
230 DATA 62,62
240 DATA 25,34,54,55,55,62
250 DATA 21,49,58,62
260 DATA 2,1,31,62
270 DATA 42,60,33,41,62
280 DATA 34,54,55,55,62
290 DATA 15,60,33,41,62
300 DATA 30,50,14,35,24,34,
    54,55,62
310 DATA 2,1,25,3,31,62
320 DATA 45,21,0,9,41,62
330 DATA 18,60,41,62
340 DATA 62,62,62,62,62,62
350 DATA 31,60,41,62
360 DATA 62,62
370 DATA 21,0,9,41,62
380 DATA 13,22,53,55,62
390 DATA 12,21,41,62
400 DATA 6,33,41,62
410 DATA 14,60,41,62
420 DATA 31,60,41,18
430 DATA 62,62,62,62,62,62,
    63
```

Program #2:

```
5   REM-THIS IS "VOICE"
6   REM-VOICE DESCRIBES THE
      48K SYSTEM
10  READ L
20  IF L=63 THEN RESTORE
30  OUT 255,L
40  WAIT 254,64,191
50  GOTO 10
55  REM--ONE WORD/STATEMENT
60  DATA 27,2,35,24,35,53,
      55,62
65  DATA 30,32,33,15,62,62,
      62
70  DATA 57,11,31,62
80  DATA 11,9,18,62
90  DATA 41,52,52,43,62
100 DATA 29,43,2,0,13,30,24
      41,62
110 DATA 2,0,28,3,18,11,9,
      30,41,62
120 DATA 31,53,52,43,31,58,
      58,3,58,62
122 DATA 3,3
125 DATA 25,50,12,37,34,54,
      55,42,58,62
140 DATA 31,37,60,33,25,10,
      20,62
150 DATA 42,54,55,55,62
160 DATA 34,54,55,55,62,62
165 DATA 62,62,62
170 DATA 12,21,0,41,62
180 DATA 57,61,42,31,62
190 DATA 21,49,58,62
200 DATA 25,35,35,13,42,6,
      33,41,13,30,62
210 DATA 11,9,13,62
220 DATA 29,52,52,43,42,41,
      5,5,41,42,62,25,0,
      6,33,41,62
225 DATA 62
230 DATA 14,35,8,41,42,31,
      62
```

cont'd on page 135

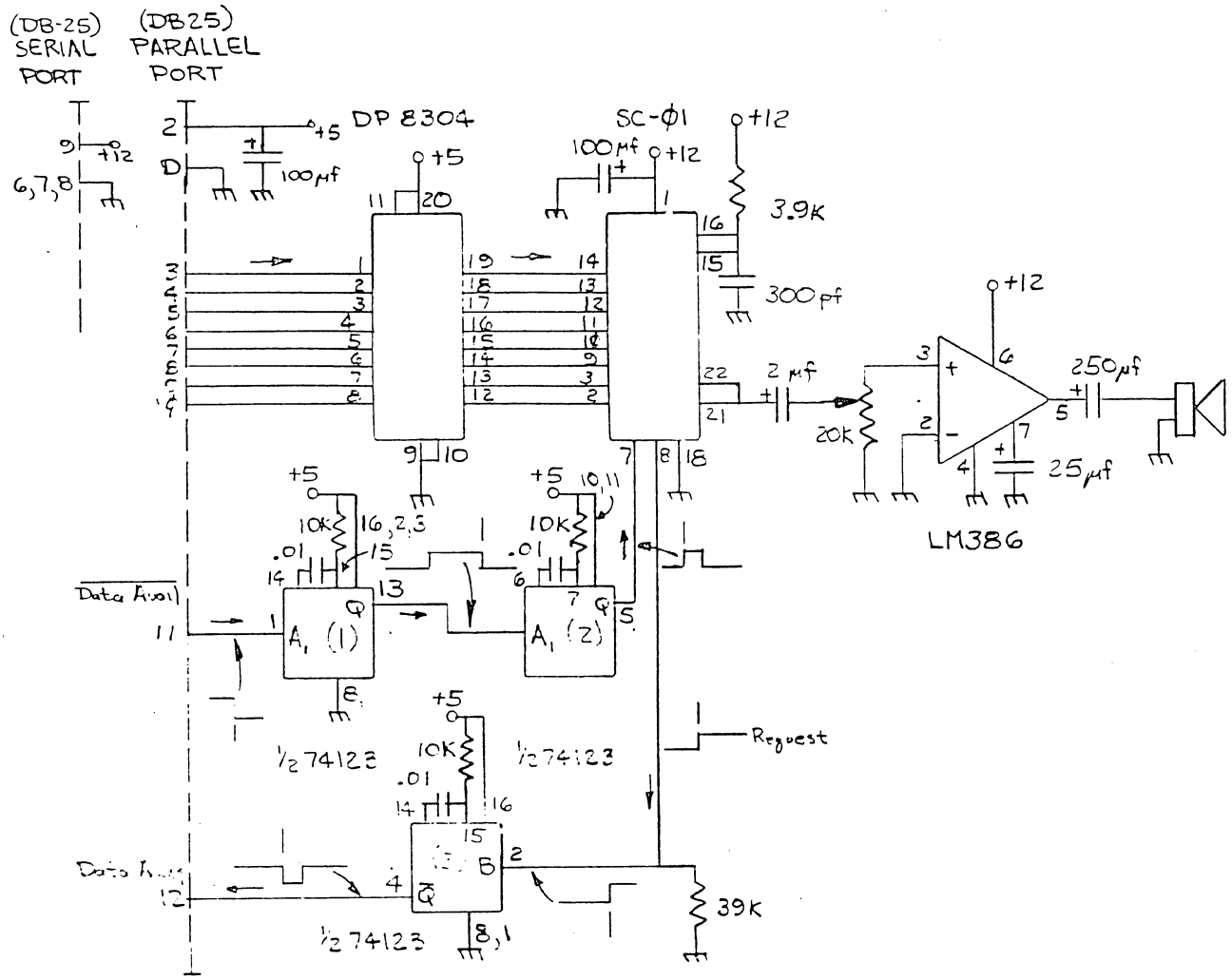

Figure 3

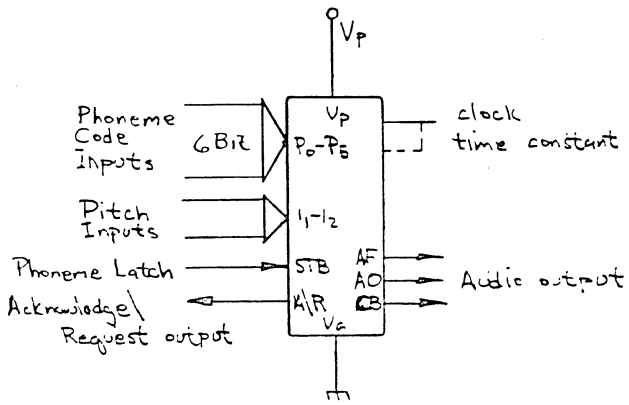

Figure 1

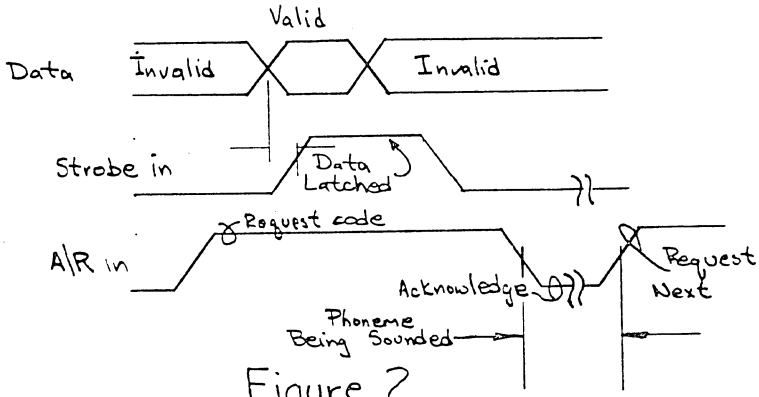

Figure 2

CPYNSTAR:

A Utility to Convert Northstar Disks to Vista Format

By Larry Conklin
105 Riverglen Road
Liverpool, N. Y.
(315)652-3564

When I was ready to add a disk drive to my Sorcerer, Exidy was not yet selling a disk system of their own. A note in the system manual recommended the Northstar S-100 controller. Since I intended to add the S-100 expansion chassis to my system anyway, the Northstar system looked quite attractive. The operating system is very good, and is well supported by third party software vendors. The other alternative was the Vista disk system, running under CP/M. At that time Vista had not announced their Sorcerer compatible controller, but they were offering a customized version of CP/M that was ready to run on the machine. The combination of CP/M and the larger disk capacity (204.8 K vs. 179.2 K) pursuaded me that Vista had more to offer, and that is the system that I ended up buying. I have been very well pleased with the choice.

As anyone, who has ever read a Lifeboat catalog can attest, there are dozens of 5" disk formats, none of which is compatible with any of the others. Unfortunately, the Vista disk format is not well supported. Very few software vendors make their products available on Vista compatible disks. I was surrounded by tempting CP/M software, but had no way to jump the "media gap". I was beginning to think I would have been better off with the Northstar system after all. I could have purchased CP/M for that system and, in contrast to the Vista format, almost everybody that sells CP/M software supports the Northstar disk format. It has come as close as any to becoming a standard format for 5" drives.

In reading through the technical manual for the Vista controller, I discovered a statement that it was "similar to the Northstar disk controller". It occurred to me that if they were similar enough, you ought to be able to persuade the Vista controller to read a Northstar disk. As it turns out, you can. Figuring out how to do it turned out to be harder than I had expected. CPYNSTAR is the result of that effort. It will copy the entire contents of a Northstar double density disk to a Vista disk. As written two drives are required, but since the copy is made one track at a time, the program could be modified for use on a single drive system.

The first step was to find out exactly what the respective disk formats were, and how they differed. A friend who had a Horizon system provided me with information from the Northstar documentation that described the disk format. I had to dig the corresponding information for the Vista format out of the listings they provided for their CP/M BDOS. Figure 1 compares the formats.

Northstar	Vista
Sector hole	Sector hole
32 bytes of zeros	24 bytes of zeros
FB (start of	01
FB data bytes)	55 (start of data byte)
512 data bytes	track no.
CRC check byte	sector no.
	512 data bytes
	CRC check byte
	AA
	24 bytes of FF

Figure 1

In addition to the differences that are apparent in the figure, the CRC check bytes are calculated differently. On a Northstar disk each data byte is excluxive OR'ed to the CRC byte and the result is rotated left. On a Vista disk the data byte is added to the CRC byte and the result is rotated right.

Having determined what the respective recording formats were,the next step was to figure out how to read a Northstar disk with the Vista controller. To do that, you have to understand how the Vista controller gets in sync with the serial data stream coming in off the disk. After a sector hole is encountered, the controller uses the string of zeros to get in phase with the

cont'd on page 116

GOODBYE F.D.S.

	DIGITRIO	EXIDY FDS
NO S100 NEEDED	yes	yes
8" DRIVES	yes	no
5" DRIVES	35,40,77,&80	77 track
DOUBLE SIDED DRIVES	yes	no
5" CAPACITY (SS)	350K	308K
MAX NO OF DRIVES	4	3
COMPLETE SINGLE USER FACILITY	yes	no
HARDWARE DOC	yes	none
USER GUIDE	40 pages	10 pages
TPA SIZE (MKII)	40K	39K
WARRANTY	12 months	3 months

$425

includes:

*controller
*CP/M 2.2
*Sorcerer cable
*extra utilities
*documentation
*shipping costs

VISA

DEALER ENQUIRIES SOLICITED

Digitrio's new disk controller is the most versatile way to attach floppy disk drives to the Sorcerer. First released in June 1982 the Digitrio contrcller is now being used by satisfied customers throughout Australia.

For further information contact Dr. D. Trussell at:

DIGITRIO, BOX 4553, MELBOURNE, VIC 3001, AUSTRALIA

* * * * * * * * * * * * * * * * * * *

TERCENTENNIAL TECHNICAL
Video and Data Communications
Technical Services

* * * * * * * * * * * * * * * * * * *

For quality Service on the following equipment:

Exidy Sorcerer Computers - Expansion Interface & Box
Radio Shack...all models - Zenith/Heath - Atari
Leedex, Sanyo, Electrohome.....and other brand monitors
Micropolis, Vista, Shugart, Percom, MPI...and other drives
Modems, Dot Matrix Printers.....most brands
Memory upgrades and new Monitor 1.1 PROM's available.

Services available on most home entertainment electronics:

Solid State Television, foreign and domestic brands.
Video Recorders---Video Cameras---Video Monitors
Stereo Receivers---Audio Recorders---Record Changers
Video Switchers---Time/Date Generators---Video Disc Systems
Video Games: Atari and others.

For above services (Tech. lic. #8228), or consultation, call or write:

Jack MacGrath
TERCENTENNIAL TECHNICAL
P.O. Box 5
70 Tercentennial Drive
Billerica MA 01821
Phone, after 6:00 pm EDT, (617)667-8272

data. As the data is received it is shifted bit by bit through a register in the controller. The controller is looking for the first 1 to appear in what would be the carry bit of the register. When the 1 is seen, the controller knows that the start of the data byte is in the register, and that each 8-bit group that follows will represent another data byte. Apparently the Northstar controller uses a different scheme for recognizing the start of the data since the all important synchronization byte required by the Vista controller is missing. As a consequence, when the Vista controller attempts to read a Northstar disk, the first non-zero bit seen as the "carry" bit is the MSB of the first start of data byte (FB). At this point bits 7-1 of the data register contain bits 6-0 of the first start of data byte and bit 0 contains bit 7 of the second start of data byte. The controller thinks it is in sync with the data however every subsequent byte will be "over-shifted" by one bit.

The solution to this dilemma is to read each byte from the controller register and then rotate it right through the carry bit. The result of this operation is to shift bits 6-0 down into their proper position in the data byte. Bit 0, as read from the controller register, is the MSB of the byte that follows in the data stream. The rotate through carry operation captures this bit and saves it for next time. The bit that is rotated into bit 7 of the data byte from the carry bit, is the MSB of the current data byte which was saved from the preceding byte read from the controller. Figure 2 illustrates what is going on.

The only byte that cannot be recovered in this manner is the initial start of data byte. Since we already know what we expect to see in this byte when the controller sync's up, this isn't a problem. The program reads each 512 byte sector into a buffer, correcting for the "phase error" as it goes. The CRC check byte is then computed from the contents of the buffer and compared with the value read from the disk. If all is well, the buffer is added to the contents of BIGBUF until ten sectors have been accumulated. The contents of BIGBUF are then written on the output disk in Vista format.

Aside from the "phase correction" logic just described, the rest of the program is straight-forward. However, I did make calls to low-level routines in Vista's disk driver software (SEEK, WSEC, SELDSK, and MHOME). If your installation of Vista's CP/M is configured for a different memory size than mine (54 K) you will have to change the four calling addresses by redefining the corresponding equates. Unless Vista has released a new version of their BDOS, you should be able to compute the new addresses by biasing the entry points by the difference in memory size. In any case, I used Vista's names for the entry points and you should be able to find them in the VISTA BDOS listing.

The program is self prompting and very easy to use. You are asked for an offset value and whether the input disk is in Lifeboat CP/M format. As far as I am aware, everyone who sells CP/M software on Northstar disks uses the same logical to physical sector mapping scheme. The offset value determines where each sector will be located on the output disk, relative to where it was on the input disk. The offset is in terms of logical (512 byte) sector numbers. The CP/M directory on a Vista disk is located on sector 6 of track 1, but on a Northstar disk, it is in the first sector of track 2. Thus the required offset for a CP/M conversion is -4. The flexibility provided by being able to specify an offset, and to enable or disable the logical to physical sector mapping allows the program to be used to do a straight conversion. I have written another utility that will convert Northstar DOS files on a Vista format disk to CP/M files, but that is another story.

There are a few vendors that only supply their programs on single density disks. There is no way to directly convert these, since the Vista controller is totally unable to read them. If you can find someone running CP/M on a Horizon he can use the PIP utility to convert the disk to double density for you, and you're in business. I hope this utility is as useful to other Vista users as it has been to me. I have purchased several CP/M programs on Northstar disks and converted them with no trouble at all. None of them would have been accessible to me without this routine.

*EXT MOFF and *MEM OFF Modifications for Sorcerer I

by Eric Moyer

For those users of disk drives with the Sorcerer I who want to or have already expanded their computers to support 48K internally (See the March 1, 1981 issue of Sorcerer's Apprentice, pp. 32-33) and want to be able to use the disk drive without the bootstrap interfering with other memory, here is a modification which will disable the RAM at the location of the bootstrap. With this modification, the boot program is positioned at the top of RAM (BF00-BFFF hex for 48K or DF00-DFFF hex if you also choose to build the RAMPAC modification). As some may remember, it was necessary to cut the trace from the Expansion Bus connector J3 pin #45 (see Exidy tech note number 2) to enable the disk drive to function correctly. This is the pin which the FDS or DDS sends the negative-logic (hereafter referred to as not-) EXT MOFF signal to the Sorcerer on to tell it to turn off the RAM at the bootstrap location. With the addition of three gates and some jumpers, the Sorcerer I can correctly interpret this signal and disable the proper RAM. The total cost for this modification is about 50 cents for a 74LS08. (See diagram below).

The OR gate labelled 10A is the existing OR gate on the circuit board, the three others are the additional ones needed for this modification. The single 74LS00 gate is one left over from the 48K modification (see page 118 of this issue) while the 74LS08 is the only additional part needed. The pin numbers near the gates correspond to the IC pin numbers, the 74LS08 can be added to a small piece of perforated PC board. The not-ROMPRE line is connected to pin 23 of the ROMPAC socket; while the not-EXT MOFF line is connected directly to pin 45 of the Expansion Bus connector J3.

NOTE: The numbering for the ROMPAC socket is INCORRECT. Be sure to use the numbering on the ROMPAC itself in the position it would be installed in the computer. Also note the two locations where traces must be cut. Use a sharp X-acto knife or equiv. to make two cuts about 1/16" apart, and then remove the cut out portion of the trace. For these modifications it may be necessary to remove the main PC board; do so carefully.

If you have not yet done the 48K mod, you can replace two of the 74LS00 gates (ones with pin # 8 connected to pin #'s 12 and 13) with a single 74LS08 gate, which puts the total parts count at 2 chips instead of 3.

When the modification is complete and you turn on your Sorcerer and disk drive, you should get a message saying that the top of RAM is BEFF for 48K or DEFF for 56K. Notice that this is 256 bytes lower than before, the memory location is now being used by the disk drive's bootstrap PROM, and the 256 bytes of RAM are now inactive.

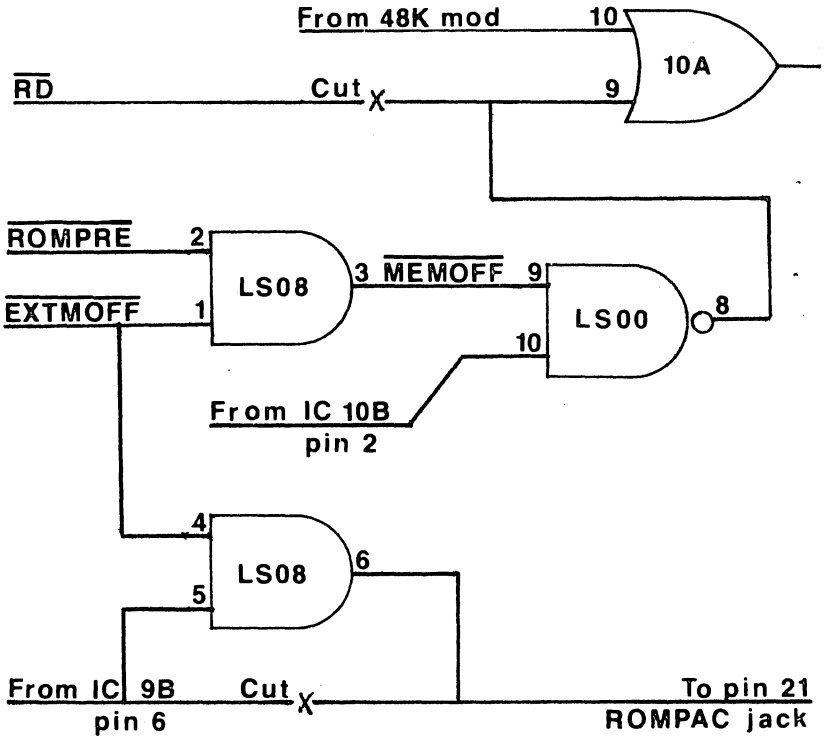

Diagram 1

* * * * * * * * * * *

Converting the Sorcerer to 56K the Easy Way

by Bob Alexander
4675 Holycon Circle
San Jose, Ca 95136

(with Model I mods and notes by Eric Moyer)

I had been looking for a way to add more memory to the Sorcerer for some time, since many applications required it, as well as several languages I was interested in learning. So what follows is a description of a method that will get you going with a minimum of fuss and cash outlay.

I had a spare EPROM PAC laying around, and decided to use the new Hitachi 6116-3 2K static RAMs which have been available for some time now at a relatively modest cost as well as being pin compatible with the 2716 EPROM. If you use the standard Exidy ROMPAC, don't dispair, as the modification will work equally well for either.

If trying to figure out the jumpers on the card keeps you awake at night mumbling to yourself, as it does to me, you're in good company! Don't worry - we don't use 'em! All wiring is point to point to keep changes on different versions of the card to a minimum.

EPROM PAC modifications:

1. Cut the traces (if present) to pins 26 and 27 at the connector. This removes +12V and -5V from the card.

2. Locate the bypass capacitors that are on these two lines, and remove them. You should wind up with only one electrolytic cap and four ceramic caps (all on the +5V line only) when finished.

3. Cut any jumpers in the pad areas associated with pins 19, 20, and 21.

4. Check the connections to the above pins to insure they're not shorted to each other, and that the individual pins are connected to all four sockets.

5. Jumper pin 21 of the sockets to pin 28 of the connector - this is the WRITE enable signal.

6. Jumper pin 20 of the sockets to ground.

7. Make sure pin 19 of the sockets goes to pin 20 of the connector. If not, wire it in.

8. Plug in the 6116 static RAMs. The only precaution here is to make sure the static RAM speed is as fast as your system RAM. I ran into a problem with this.

NOTE: On some of the older ROM PACS pin numbers 28 and 30 on the edge connector are connected together (actually it is one BIG pin). For that PAC to work correctly with the RAMPAC modifications, it is necessary to cut these apart. Using a sharp knife make two lengthwise slits a third of the way in from either side of the big pin, and scrape out all the trace in between the two slits. This should result in the big pin being converted into two normal sized pins, with a gap of bare board the width of a pin trace in between them. This modification is necessary on ALL EPROM/ROM pacs, NOT just the one with the static RAM in it. (E.M.)

This completes the mods necessary to the EPROM PAC.

Model I modifications:

The only modifications necessary for the main PC board (for non-disk users) is to cut one trace and install one jumper.

1. Locate IC 9B (a 74LS21). Cut the trace connecting to pin 3 (yes, its on the bottom).

2. Run a jumper from pin 3 of RAM chip IC 13A (this is the WR signal, there may be a closer place to pick it up, but I used this) to pin 28 of the ROMPAC connector.

NOTE: The numbering of the ROMPAC connector is WRONG. Refer to the numbering on a properly inserted ROMPAC itself.

That's it for the Model I mods. For disk users, see the EXT MOFF modification, and the controller board reconfiguration for the Model II below.

Model II modifications:

There are two areas on the main board that need to be changed and one switch on the

cont'd on page 119

disk-controller card.

1. Locate IC 9A...it's the 74LS08 near the main RAM. There is a spare positive AND gate at pins 4, 5, and 6 that we will use.

2. Run a jumper from 9A-5 to 2F-1 (EXT MOFF -low true)

3. Locate and cut the trace going to 10A-8 (ROMRD15)

4. Jumper 10A-8 to 9A-4

5. Jumper 9A-6 to the trace you cut from 10A-8 which goes to pin 21 of the ROMPAC connector.

6. Locate E18 and E19 and jumper them together. These are near the ROMPAC connector and may be hard to read. If so, just wire 6H-5 (WRITE - low true) to pin 28 of the ROMPAC connector.

7. The next step will require that you open up the disk drive cabinet and locate the 8 pole DIP switch near the left-rear corner of the controller card. Set switch 3 'on' and make sure all others are 'off'. This is the boot address switch. If you own one of the older controllers without the switch, make the following mods to the controller card.

A. Cut 5D-12 and 13 (74LS04)
B. Cut 3F-7 (74LS241)
C. Jumper 3F-5 to 3D-3
D. Jumper 5D-13 to 3F-7
E. Jumper 5D-12 to 3D-2

That's it! No software changes should be necessary. Although the changes here are specific to the Mod I 'B' and the Mod II 'C-2' boards, they should be applicable to all others as well.

Also, the Word Processor Pac will not work now, but the ROMPAC BASIC and Dev Pac should if you issue a 'GO C000' upon cold-booting the system. The new disk boot address is now at DF00.

I haven't included any mods for the Micropolis system, since I am unfamiliar with its operation.

Total cost will run between $30-60 unless you need to purchase an extra EPROM card, which was, at last check, around $50 from Exidy, but since I never use the ROMPAC BASIC any longer, I decided to use that card. At any rate, I haven't seen add-on memory for less than $135 or so, which means you'll save $25-100, even if you need to buy the extra card.

IN THE PUBLIC DOMAIN

by Bruce Blakeslee
CP/M - Micropolis Librarian
906 Crestwood Road - West
Westfield, New Jersey 07090
(201)-233-3185 (Evenings)

This, the third installment, will focus on the SIG/M User's Group. The Special Interest Group for CP/M software was formed about 18 months ago by the ACGNJ (Amateur Computer Group of New Jersey) because it was felt that a new outlet for public domain software was necessary. The CP/MUG had become somewhat inactive, rarely releasing new disks of software, and it was felt that another voice was necessary. In the 18 months they have been in existence, 69 disks of public domain software have been released. With 4 to 8 new disks a month this is a very active and dynamic group eager to publish good software. The following is a listing of the software available on their disks. I am able to provide a much more extensive catalog of CP/MUG and SIG/M software on disk if this article whets your interest. Note the end of this article for particulars on ordering software.

```
**********************************
*      Notations Used Below      *
*                                *
*  1 = A Volume Most Will Want   *
*  2 = Of Interest From The      *
*      Point of Software         *
*      Development.              *
*  3 = Of Limited Interest       *
**********************************
```

Volume 1	Original 350 Point Adventure (8080)	(1)
Volume 2	Adventure Source Code (Fortran)	(3)
Volume 3	Expanded Adventure - 500 points	(1)
Volume 4	Miscellaneous CP/M Utilities	(1)
Volume 5	8080/8085 Memory, ICOM Disk Diag.	(3)
Volume 6	6502 Simulator for Z80	(2)
Volume 7	RBBS and CP/M Utilities	(2)
Volume 8	PASCAL and Communications Utilities	(3)
Volume 9	CP/M Utilities	(1)
Volume 10.....	Utilities, Games, Disassembler	(1)
Volume 11.....	Expanded Adventure for Z80	(1)
Volume 12.....	Cross Reference index of CP/MUG Volumes 1-47 and SIG/M Vol.'s 1-11.	(3)
Volume 13.....	HELP system, FIGFORTH 1.1	(1)
Volume 14.....	HELP system	(1)
Volume 15.....	Catalogs of CP/MUG Volumes 1-47 and SIG/MUG Volumes 1-3.	(3)
Volume 16.....	CP/M Utilities & Communications	(1)
Volume 17.....	CP/M Utilities	(1)
Volume 18.....	CP/M Utilities	(1)
Volume 19.....	Pascal Z User's Group Volume 1	(1)
Volume 20.....	Pascal Z User's Group Volume 2	(1)
Volume 21.....	Pascal Z User's Group Volume 3	(1)
Volume 22.....	Pascal Z User's Group Volume 4	(1)
Volume 23.....	Pascal Z User's Group Volume 5	(1)
Volume 24.....	Pascal Z User's Group Volume 6	(1)
Volume 25.....	Pascal Z User's Group Volume 7	(1)
Volume 26.....	Games, Mail Label	(1)
Volume 27.....	Advanced Terminal Systems-Confidence Tests.	(3)
Volume 28.....	Games, Billing System, Home Remote Control Programs	(1)
Volume 29.....	Various DOC, READ-ME, and Abstracts	(3)
Volume 30.....	More of the above	(3)
Volume 31\		
\		
Through	The Yale Catalog of Bright Stars	(3)
/		
Volume 38/		
Volume 39.....	Language Analyzer	(1)
Volume 40.....	Misc. System Support Utilities	(2)
Volume 41.....	Cromemco CDOS-CP/M System Support	(3)
Volume 42.....	SD Sales Hard Disk Support	(2)

cont'd on page 121

Volume 43.....	CP/M-86 BIOS, TINCMP Compiler	(2)
Volume 44.....	CP/M Utilities	(1)
Volume 45.....	Dungeon - Original Fortran Code	(3)
Volume 46.....	More Dungeon	(3)
Volume 47.....	More Dungeon	(3)
Volume 48.....	CP/M and Communication Utilities	(1)
Volume 49.....	Accounts Payable/Receivable and Master DataBase System	(1)
Volume 50.....	DTC Hard Disk Bios, Voice Synthesis System	(1)
Volume 51.....	1981 Information Catalog	(3)
Volume 52.....	Updated Disk Cataloging System	(1)
Volume 53.....	Updated SAM76 Language	(1)
Volume 54.....	Z80 CCP Replacement	(1)
Volume 55.....	Documentation Addendum #3	(3)
Volume 56.....	Musicraft Music System	(1)
Volume 57.....	More Musicraft	(1)
Volume 58.....	More Musicraft	(1)
Volume 59.....	PISTOL (FORTH like) Language	(1)
Volume 60.....	CP/M Utilities	(1)
Volume 61.....	Dan's Information Management System	(1)
Volume 62.....	Pascal Z User's Group Volume 10	(1)
Volume 63.....	Pascal Z User's Group Volume 11	(1)
Volume 64.....	Pascal Z User's Group Volume 12	(1)
Volume 65.....	Updated Bulletin Board System Support, Utilities.	(1)
Volume 66.....	Updated Bulletin Board System	(1)
Volume 67.....	Updated Bulletin Board System	(1)
Volume 68.....	CP/M Utilities	(1)
Volume 69.....	Pascal Z User's Group Volume 13	(1)

As you can see there is a lot here. In the future SIG/M is supposed to release tiny ADA as featured in S-100 Microsystems among other things. What can I say, order the Catalog disk from me and get a more detailed picture of what's in these disks. In future months I will announce additional volumes of the CP/MUG and the SIG/M Users Group. I will also be discussing individual volumes which I feel are of significant interest to Sorcerer Users. (Some of the languages and utility disks fall into this category.) I would also like to publish the remarks of others about pieces of public domain software you have used and liked or found poorly written. This is how we move ahead and progress. I would now like to announce the first in a series, I hope, of Sorcerer User Group disks. These disks are made up of software, Sorcerer users have either written themselves or significantly rewritten to run on the Sorcerer. It is all Public Domain. As I have indicated before, If you contribute to the Sorcerer Users Group I will send you a disk of software of your choice. (One disk for each disk or partial disk contributed.) I hope this will encourage many of you to contribute.

The Sorcerer User's Group (SUG)

SUG Volume 01 ...Software discussed by Brian Lewis in Volume 4, #3. This includes, MSPEED, ASCIIFY, LIFE, MODEM7X, etc.

SUG Volume 02 ...TIP, FIX, and BASIC PAC programs. This includes 56K-MOD, 8304PROB.TIP, MX-80.TIP, SERIAL.DOC, MX80PLOT.BAS, etc.

SUG Volume 03 ...PAC BASIC programs. These are tu be used by systems which can run both PAC BASIC and the disk system together. This includes, STARTREK, 5 Astronomy programs, BOWLS5, CHKR5 (Checkers), etc.

SUG Volume 04 ...More TIP, FIX, MOD, and DOC files as well as a program for the Hayes Smartmodem.

```
**********************************************
*                   PRICES                   *
*                                            *
* 1. $3.00 + postage if you supply the       *
*      media already formatted.              *
*                                            *
* 2. $8.00 + postage if I supply the media   *
*                                            *
**********************************************
```

C/80 REVIEW

by Robert Hageman

C, a compiler language developed at Bell Laboratories by Dennis M. Ritchie and Brian W. Kernighan, runs faster than BASIC and allows faster-easier program development than assembler. C/80, a subset of standard C (here read BDS/C), was written by Walt Bilofsky of The Software Toolworks, based on Ron Cain's Small C compiler as published in Dr. Dobb's Journal. C/80 for the Exidy Sorcerer is distributed by Triangle Systems at a price of $49.00.

As one might expect of a subset, C/80 does not support some of the nifty features to be found in the full language. Then too, C/80 does not support the "nifty ?" price tag to be found on the full language (try Whitesmiths' C Compiler @$600). The most obvious feature missing from C/80 is floating point math, which is the most costly to develop.

I found C a promising language for file manipulation, the best covered examples in what I read. I know, C has become the second language on RCPMs for public domain programs; utilities such as SQueeze, USQueeze, and TYPESQ were all written in C.

The language is designed for structured programming. The rules that control syntax also lead one to write orderly and readable programs. This makes debugging less of a chore as the program can be written in modules and each module debugged separately. C/80 allows the use of machine language subroutines. Routines may be included in the C program either as a part of the source file, or as a separate file you tell the compiler to include. This means you can have special keyboard drivers, port drivers, etc. included in any program you need them in. Indeed, other C programs can be included as subroutines in a new program.

The "User's Guide" supplied with C/80 is just that - a guide. It will serve the purpose of informing the programmer already familiar with C of the differences between standard C and C/80. It is not and was never intended an in depth manual for learning the language. The guide does try to steer you to some good references; The C Programming Language, An Introduction to the C Programming Language, Dr. Dobbs Journal, Version 1.5 C/80 Manual, and Structured Design. The manual was written by Walt Bilofsky, the writer of C/80.

I found C/80 was not directly compatible with BDS/C programs. The problem arises from those missing features which BDS/C takes for granted. This is not to say BDS/C programs can not be run on C/80, but rather that the programs will require editing and rewriting to make them work.

I would recommend C/80 to those:

1. wanting to try a new language without losing an arm and a leg.
2. wanting a language easier than assembler but with most of the speed of machine code.
3. wanting a faster language than BASIC but not too much harder to learn.

I would not recommend it to those:

1. knowing their interests are in the BDS/C programs to be found on some RCPMs.
2. knowing they NEED floating point math for the programs they intend to write.
3. who have worked in a larger C compiler and are therefore comfortable with the features.

* * * * * * * * * * * * *

EDITORS NOTE:

Triangle Systems is no longer supporting the Exidy Sorcerer with any kind of software. If any of you have sent money to them and have not received any merchandise, or if you received defective merchandise, we suggest you contact Wim Plaat of Triangle Systems at the following address:

Triangle Systems
1690 West Lane Avenue
Columbus, OH 43221
Tel. (614)486-3527

The telephone is normally answered by an automatic answering machine and callbacks are very seldom made. We strongly suggest that you use certified mail in any dealings with them. Should you receive no satisfaction via this method - contact the Attorney General of the State of Ohio. DO NOT send original documentation; have a copy notarized!

SORCERER TELECOMMUNICATIONS SYSTEM

A review by Tom Bassett

Remote computer systems, CBBS's, and data-base systems have proliferated of late. All that software, there for the taking! For the Sorcerer user without a CP/M disk system, though, it's been "look but don't touch"; read the mail but don't download a file 'cause your terminal program can't handle it.

Enter STS

The Sorcerer Telecommunications System Version 2.1 was written by Jonathan E. Burnett to make a cassette-based Sorcerer with a serial modem behave like a CP/M system running a version of Ward Christiansen's MODEM program. With STS, files can be downloaded and uploaded. The program is written in Z-80 assembly language, and will run on any Sorcerer (16K or more of memory is desirable). The STS program is being sold by the Sorcerer's Apprentice for $30, which buys the tape and the documentation.

I have a CP/M disk system with a serial modem (a Novation D-Cat) and I've been running a version of the MODEM program for some time. I was furnished a copy of the STS program to test and review. I shut off the disk drives, booted the Sorcerer monitor, and gave STS a good run. Here's what happened.

Documentation

First, of course, I read the documentation supplied with the tape. There's a table of contents, which is handy for later reference. The documentation reads almost like a "walk-through" instruction; once or twice through it and the user has a good grasp of the system.

The tape loaded on the first try at 1200 baud (a 300 baud save is furnished also). I LOG'd the tape, and STS came up and running with a nice "communication parameter" selector.

Parameters

The parameters are pre-set for the most common situation, but the user can change any parameter by merely typing the number of that parameter as shown on the selector menu. Typing the number "toggles" the selection, and the current selection of each parameter is indicated by a reverse video display. Typing the number 2 several times will toggle the baud rate back and forth between 300 and 1200, for example. The parameters that may be selected are (1) full or half duplex; (2) baud rate 300 or 1200; (3) 1 or 2 stop bits; (4) 7- or 8-bit words; (5) even, odd, or no parity; (6) auto linefeed on or off; (7) echo mode on or off; (8) control code display on or off; and (9) error display on or off.

Menu-Driven

The parameters were pre-set the way I wanted them, so after playing with the selector for a while I hit return and STS displayed a master menu. The menu allows seven different functions. Once in a selected function the user can return to the master menu by typing the escape key. Very nice; in fact, it's a better setup than most of the MODEM versions I've used.

The master menu selections available are (1) set parameters (as described above); (2) terminal mode; (3) save to tape; (4) load from tape; (5) block file transfer; (6) ASCII file transfer; and (7) directory display.

I typed a 2 to get into the terminal mode, and dialed the number of the Sorcerer remote system in Detroit. The remote signed on, booted its CP/M system, and STS behaved exactly the way I hoped it would; that is, it gave me full control of the remote just as an intelligent terminal should. From the remote CP/M A> prompt I typed SD (a sorted director program) and got a directory of the A: disk. Typing SD B: brought up a directory of the B: disk. Calling in the MINICBBS system, I dumped a few messages. Everything worked just fine.

Block File Transfer

Now it was time to test the block file transfer mode. While still in the terminal mode I typed XMODEM S FILENAME.EXT. This called the XMODEM program in the remote system, with instructions to S(end) the named file. XMODEM signed on, and I hit the escape key to go back to the master menu of STS. The menu appeared instantly, and I typed a 5 to go to the block file transfer mode. A sub-menu offered me the choice of S(end) or R(eceive). I typed an R, and the

cont'd on page 124

screen showed me the available RAM for storage of the file being received, and the corresponding number of CP/M disk sectors available.

Why show the number of CP/M sectors available? Simply because XMODEM, when it signs on in the S(end) mode, displays that information for the file being sent, and in fact the file is sent in sector-sized blocks. Showing the sector count available to STS is a nice touch, as it allows the user to make a direct comparison to the available RAM in his system. If it's evident that the available RAM is not enough to hold the file being downloaded, the user can escape back to the master menu, enter the terminal mode, and abort the file transfer, thus saving connect time (which, of course, saves money if it's a toll call).

As the file was being downloaded from the remote system, STS displayed the sector being transferred, and the error count. I received the file with no errors until I deliberately knocked on the telephone handset with a pencil to introduce some garbage into the reception. STS recovered nicely, retransmitting the offending block.

Data Verification

STS assumes that the sending program will use the CRC data verification method. However, if after 10 seconds STS has not received a CRC confirmation signal it will automatically switch to the checksum data verification method, and display a message to that effect. A nice touch.

STS allows 10 errors or timeouts on any one sector being transmitted. If the errors exceed 10, STS automatically aborts the block transfer mode and returns to the master menu.

When the block file transfer was completed, STS announced that fact and returned to the master menu. Typing a 7 displayed the STS file directory, showing the file just downloaded and the beginning and ending memory addresses of the file. STS can hold up to nine files; the memory available for them is equal to the Sorcerer memory size minus 6919 bytes (the size of the STS program).

Uploading

Back to the terminal mode (type a 2 from the master menu). The next test was for transmission of a file from my system to the remote. I decided to retransmit the file I had just downloaded. While in the terminal mode I typed XMODEM FILENAME.EXT. This loaded the remote modem program in the receive mode. Typing escape got me back to the master menu, and typing a 5 again (block transfer mode) brought up the sub-menu. I typed an S to indicate I was sending a file, and STS asked for the file number to send. Since I had only the one file in memory I typed a 1 and we were off and running. The file transferred without error, and, when finished, STS went back to the master menu. I typed a 2 to get back into the terminal mode, and, when the remote showed me the A> prompt, I typed BYE to terminate the session with the remote.

Tape Save

Now that I had a file in memory, I wanted to save it to tape. I typed a 3 from the master menu (save to tape), and was shown a directory display listing all the files in memory and the number of each. STS asked me to type the number of the file I wanted to save (I typed a 1) and the usual 5-character file name. Then I selected the 1200 baud option, started the tape recorder in the record mode, and hit the return key. The file was written to tape, and STS asked if I wanted to make another copy. I answered N(o), and STS then asked if I wanted to dump more files. I answered N again, and the master menu reappeared.

Of course, saving a file to tape is not as fast as saving a file to disk. But in all other aspects the STS program runs as fast as most versions of the MODEM program, and is really easier to use. The master menu, and the interactive prompts, are more "user-friendly" than the command lines required in the MODEM program.

Tape Load

I tested the tape load option (4 on the master menu) by loading the program I had just saved. This demonstrated the memory-clear feature of STS. On the first tape load attempt following a tape save or any file transmission, STS will assume that you have either saved all the files in memory or have transmitted them all, and no longer need them. The program will automatically purge all

cont'd on page 125

files, making the whole memory available to load new files. On subsequent tape loads (with no intervening tape saves or file transmissions) STS will load the additional tape files without purging memory, up to a maximum of nine files or memory capacity.

This automatic purge feature is good, but it is necessary that the user understand it. A file could be lost if the correct sequence of loads, transmissions, and saves is not followed. Perhaps a cautionary "are you sure?" message would be useful here.

Now that we have saved a downloaded file to tape, what do we do with it? Well, first of all we have to know where in memory it should be loaded. The file is saved to tape, and will be reloaded, at the memory addresses at which it resides after being downloaded. If the file is a program, this will probably not be the run address. So after we reload the program it must be moved with the monitor to the correct address and resaved on tape. Likewise, a text file may have to be moved to the correct load address for the word processor pac. The STS documentation does not mention any of this, and it probably should, although anyone familiar with the Sorcerer will be familiar with these requirements.

ASCII Files

STS has another feature that's really frosting on the cake. The CP/M MODEM programs are designed to transfer binary and text files in the block mode, as is STS. But STS also incorporates a file transfer option that works in a mode similar to that employed by the PLINK CP/M public domain programs. In this mode, 7-bit ASCII files are transferred without data verification. This allows STS to operate with remote systems having little or no transfer software at the remote end. The documentation explains the protocols necessary, and gives step-by-step procedures. This mode will allow "capture" of interactive communications with the host computer, such as the dialog that appears on your screen when in contact with a CBBS system. Most of the CP/M MODEM programs don't have this feature, although the latest versions (MODEM7, for example) have incorporated it.

STS allows you to "mix" received files; that is, you might receive a file in the block mode, then receive another in the ASCII mode, then a third in the block mode, etc.

The STS documentation provides directions for customizing STS so that the program LOGs in the desired configuration. This makes it unnecessary to select the desired communication parameters at the beginning of each session.

The documentation also provides a listing of key-press / hex code / character display or function. And, for the benefit of the first-time user of a communications system, the documentation has a glossary of communications terms.

It's Good!

STS is a major programming effort, and is light-years ahead of any other communications software available for a cassette-based Sorcerer. I like it; it's easy to learn and use. Some of the communications software available for CP/M disk systems could take a lesson in user-friendliness from STS.

* * * * * * * * * *
* *
* FOR SALE! *
* ========= *
* *
* * * * * * * * * *

Sorcerer I w/ 32k RAM and BASIC ROMPAC, $500.00. Len Crane, 87070 Dukhobar Rd., Eugene, OR 97402. (503)485-3516, 10-6

* * * * * * * * * *

= = = = = = = = = =

* * * * * * * * * *

Sorcerer I w/32k RAM, B&W monitor, programs...$650.00 or best offer. Leland Womack, 14567 Kittery Street, Poway, CA 92064, (714)486-1388.

* * * * * * * * * *

Several requests have been received for the address and telephone number of Micromint (mentioned in the article on Speech Synthesis, SA Vol. 4, issue #3, page 64):

Micromint Inc.
917 Midway
Woodmere, NY 11598
Tel. (516)374-6793
(800)645-3479

* * * * * * * * * *

```
                ;*********************************************
                ;* CPYNSTAR - Utility to copy all 35 tracks  *
                ;* from a Northstar double density disk on   *
                ;* drive A to a second disk on drive B, in   *
                ;* Vista format. Unformatted sectors on the  *
                ;* source disk are output as zero-filled     *
                ;* sectors.                                  *
                ;*      Version 2.0 - 3/21/81                *
                ;*********************************************

                        .PABS           ;GENERATE ABSOLUTE OBJECT MODEL
0100                    .LOC  100H      ;ORIGINED AT 100H FOR COM FILE
0100  C3 0115           JMP   START     ;JUMP AROUND SCRATCH AREA
                ;
                ;PROGRAM VARIABLES AND EQUATES
                ;
C93D            SEEK=0C93DH             ;SEEK & TRACK SECTOR (VOS)
C8F1            WSEC=0C8F1H             ;WRITE IN VISTA FORMAT (VOS)
C777            SELDSK=0C777H           ;SELECT DRIVE (VOS)
C747            MHOME=0C747H            ;HOME THE HEAD (VOS)
DB50            INDAT=0DB50H            ;DATA INPUT ADDRESS
D900            REBOOT=0D900H           ;REENTRY INTO CP/M
E23D            ASCHEX=0E23DH           ;CONVERT ASCII TO BINARY (EXIDY)
0005            CPMIO=5                 ;CP/M I/O ENTRY ADDRESS
0D0A            CRLF=0D0AH              ;CARRIAGE RETURN/LINE FEED
000C            ERASE=0CH               ;CLEAR SCREEN, HOME CURSOR
0103  00        TRACK:  .BYTE  0        ;CURRENT TRACK NUMBER
0104  00        SECTOR: .BYTE  0        ;CURRENT SECTOR NUMBER
0105  0000      BIAS:   .WORD  0        ;ABSOLUTE SECTOR BIAS
0107  00        SKWFLG: .BYTE  0        ;SET IF SKEWED READ
0108  00        CRTRY:  .BYTE  0        ;RETRY COUNTER
0109  00        CRC:    .BYTE  0        ;CRC CHECK BYTE
010A  00        ERRCD:  .BYTE  0        ;ERROR CODE
010B  0000      BIGPTR  .WORD  0        ;TRACK BUFFER POINTER
010D  06        CMDBUF: .BYTE  6        ;COMMAND ENTRY BUFFER
010E  00                .BYTE  0        ;CP/M RETURNS CHARACTER COUNT HERE
010F                    .BLKB  6
                ;
                ;PRINT OPERATOR INSTRUCTIONS AND WAIT FOR CARRIAGE RETURN
                ;
0115  0E02      START:  MVI    C,2      ;2=WRITE A CHARACTER
0117  11 000C           LXI    D,ERASE  ;FORM FEED CHAR
011A  CD 0005           CALL   CPMIO    ;SEND TO CRT
011D  0E09              MVI    C,9      ;9=PRINT A BUFFER
011F  11 032A           LXI    D,MSG1   ;OPERATOR PROMPT MESSAGE
0122  CD 0005           CALL   CPMIO    ;SEND TO CRT
0125  CD 0317           CALL   WAIT     ;WAIT FOR CARRIAGE RETURN
0128  0E09      INPT:   MVI    C,9
012A  11 03DF           LXI    D,MSG3   ;REQUEST BIAS FROM OPERATOR
012D  CD 0005           CALL   CPMIO
0130  CD 02EC           CALL   INPUT    ;WAIT FOR OPERATOR ENTRY
0133  3A 010E           LDA    CMDBUF+1 ;GET NO. CHARACTERS ENTERED
0136  FE05              CPI    5        ;TEST FOR TOO MANY
0138  30EE              JRNC   INPT     ;TRY AGAIN IF SO
013A  21 010F           LXI    H,CMDBUF+2;POINTS TO ENTRY
013D  7E                MOV    A,M
013E  FE2D              PCI    "-"      ;WAS ENTRY NEGATIVE?
0140  2001              JRNZ   IPT1     ;IF SO CONTINUE
0142  23                INX    H        ;IF NOT ADVANCE POINTER PAST "-"
0143  CD E23D   IPT1:   CALL   ASCHEX   ;CONVERT ENTRY TO BINARY --> (DE)
0146  3A 010F           LDA    CMDBUF+2 ;GET FIRST CHARACTER AGAIN
0149  FE2D              CPI    "-"      ;TEST FOR MINUS
014B  2006              JRNZ   IPT2     ;BRANCH IF NOT
014D  16FF              MVI    D,0FFH   ;ELSE NEGATE (DE)
014F  7B                MOV    A,E
0150  ED44              NEG
0152  5F                MOV    E,A
0153  ED53 0105 IPT2:   SDED   BIAS     ;SAVE BIAS VALUE
0157  0E09      IPT5:   MVI    C,9      ;NOW ASK FOR SKEW/NO SKEW
0159  11 0402           LXI    D,MSG4   ;SET MESSAGE POINTER
015C  CD 0005           CALL   CPMIO    ;PRINT THE PROMPT MESSAGE
015F  CD 02EC           CALL   INPUT    ;WAIT FOR RESPONSE
0162  0600              MVI    B,0
0164  3A,010F           LDA    CMDBUF+2 ;PICK UP ENTRY CHAR
0167  FE59              CPI    "Y"
0169  2806              JRZ    IPT3
016B  FE4E              CPI    "N"
```

cont'd on page 127

```
016D  2803            JRZ   IPT4
016F  18E6            JMPR  IPT5      ;ENTRY WAS NFG, TRY AGAIN
0171  04        IPT3: INR   B         ;SET SKEW FLAG FOR "Y"
0172  78        IPT4: MOV   A,B       ;IF "N", DON'T
0173  32,0107          STA   SKWFLG    ;THEN SAVE IT
                ;
                ;TRACK LOOP - COPY 10 SECTORS FROM EACH OF 35 TRACKS INTO
                ;             MEMORY ('BIGBUF'), THEN COPY BUFFER ON OUTPUT
                ;             DISK
                ;
0176  3E00            MVI   A,0
0178  32 0103         STA   TRACK     ;INITIALIZE TRACK POINTER
017B  0E00            MVI   C,0
017D  CD C777         CALL  SELDSK    ;SELECT DRIVE A
0180  CD C747         CALL  MHOME     ;HOME THE HEAD
0183  0E01            MVI   C,1
0185  CD C777         CALL  SELDSK    ;SELECT DRIVE B
0188  CD C747         CALL  MHOME     ;HOME THE HEAD
                ;
                ;SET UP FOR TRACK READ
                ;
018B  21 045E   TLOOP: LXI   H,BIGBUF ;BUFFER START ADDRESS
018E  22 010B         SHLD  BIGPTR    ;INITIALIZE BUFFER POINTER
0191  0E00            MVI   C,0       ;0 SELECTS DRIVE A
0193  CD C777         CALL  SELDSK    ;DRIVE SELECT ROUTINE
0196  3E00            MVI   A,0
0198  32 0104         STA   SECTOR    ;SET SECTOR POINTER
                ;
                ;READ 10 SECTORS
                ;
019B  3A 0103         LDA   TRACK
019E  47              MOV   B,A
019F  0E00            MVI   C,0       ;POINTS TO SECTOR 0, THIS TRACK
01A1  CD C93D         CALL  SEEK      ;DUMMY SEEK FOR TIMING
01A4  CD 0204   RLOOP: CALL  READIN   ;READ NORTHSTAR SECT.
01A7  A7              ANA   A         ;A-REG RETURNS STATUS
01A8  C4 0307         CNZ   ZERBLK    ;ERROR IF NOT ZERO
01AB  3A 0104         LDA   SECTOR    ;GET SECTOR POINTER
01AE  3C              INR   A         ;INCREMENT
01AF  32 0104         STA   SECTOR    ;AND STORE
01B2  2A 010B         LHLD  BIGPTR    ;GET BUFFER POINTER
01B5  01 0200         LXI   B,512     ;AND SECTOR SIZE
01B8  09              DAD   B         ;ADD THEM
01B9  22 010B         SHLD  BIGPTR    ;AND UPDATE POINTER
01BC  FE0A            CPI   10        ;ALL SECTORS READ?
01BE  20E4            JRNZ  RLOOP     ;LOOP TIL 10 READ
                ;
                ;SET UP FOR WRITING
                ;
01C0  21 045E         LXI   H,BIGBUF  ;ADDRESS OF BUFFER
01C3  22 010B         SHLD  BIGPTR    ;RESET BUFFER POINTER
01C6  0E01            MVI   C,1       ;1 SELECTS DRIVE B
01C8  CD C777         CALL  SELDSK    ;DRIVE SELECT ROUTINE
01CB  3E00            MVI   A,0
01CD  32 0104         STA   SECTOR    ;RESET SECTOR POINTER
                ;
                ;WRITE 10 SECTORS
                ;
01D0  CD 0298   WLOOP: CALL  WRITEV   ;WRITE SECTOR IN VISTA FORMAT
01D3  A7              ANA   A         ;A-REG RETURNS STATUS
01D4  280A            JRZ   NOERR     ;ZERO IF NO ERROR
01D6  0E09            MVI   C,9       ;BUT IF WRITE PROTECTED
01D8  11 042E         LXI   D,EMSG1   ;SET ERROR MESSAGE POINTER
01DB  CD 0005         CALL  CPMIO     ;AND PRINT THE MESSAGE
01DE  1816            JMPR  DONE      ;AND QUIT
01E0  3A 0104   NOERR: LDA   SECTOR    ;GET SECTOR POINTER
01E3  3C              INR   A         ;INCREMENT
01E4  32 0104         STA   SECTOR    ;AND SAVE
01E7  FE0A            CPI   10        ;ALL SECTORS WRITTEN?
01E9  20E5            JRNZ  WLOOP     ;LOOP TIL 10 WRITTEN
01EB  3A 0103         LDA   TRACK     ;GET TRACK POINTER
01EE  3C              INR   A         ;INCREMENT
01EF  32 0103         STA   TRACK     ;AND STORE
01F2  FE23            CPI   35        ;ALL TRACKS COPIED?
01F4  2095            JRNZ  TLOOP     ;CON TILL 35 COPIED
                ;
                ;PRINT OPERATOR PROMPT TO RELOAD SYSTEM DISK, WAIT FOR 'RETURN'
                ;THEN REBOOT CP/M.
```

cont'd on page 128

```
01F6  0E09      DONE:   MVI     C,9         ;9=PRINT A BUFFER
01F8  11 0397           LXI     D,MSG2
01FB  CD 0005           CALL    CPMIO       ;PRINT OPERATOR PROMPT
01FE  CD 0317           CALL    WAIT        ;WAIT FOR 'RETURN'
0201  C3 D900           JMP     REBOOT      ;RETURN TO CP/M
                        ;
                        ;READN - READ 1 SECTOR FROM NORTHSTAR DISK INTO 512 WORD BLOCK
                        ;        STARTING AT 'BIGPTR'. RETURNS STATUS CODE IN A. IF
                        ;        READ WAS OK, A=0.
                        ;
0204  2A 010B   READN:  LHLD    BIGPTR      ;PICK UP BUFFER POINTER
0207  DD21 023C         LXI     X,SKWTBL    ;SET POINTER TO SKEW TABLE
020B  3A 0103           LDA     TRACK
020E  47                MOV     B,A         ;SET TRACK NUMBER
020F  3A,0107           LDA     SKWFLG      ;PICK UP SKEW FLAG
0212  A7                ANA     A           ;TEST FLAG STATE
0213  3A 0104           LDA     SECTOR
0216  2806              JRZ     RN1         ;IF FLAG WAS 0, USE SECTOR NO
0218  32 021D           STA     RN2+2       ;ELSE SET SKWTBL INDEX POINTER
021B  DD7E00    RN2:    MOV     A,0(X)      ;AND USE VALUE FOUND
021E  4F        RN1:    MOV     C,A         ;AS SECTOR NUMBER
021F  3E0A              MVI     A,10        ;SET RETRY COUNT TO 10
0221  3D        GSCT2:  DCR     A
0222  32 0108           STA     CRTRY
0225  E5                PUSH    H           ;SAVE BUFFER POINTER
0226  C5                PUSH    B           ;SAVE TRACK AND SECTOR NO.
0227  CD 0246           CALL    LONE        ;READ A SECTOR
022A  C1                POP     B           ;RETRIEVE TRACK AND SECTOR
022B  D1                POP     D           ;RETRIEVE BUFFER POINTER
022C  C8                RZ                  ;RETURN IF GOOD READ
022D  EB                XCHG                ;PUT POINTER BACK IN HL
022E  32 010A           STA     ERRCD       ;SAVE ERROR CODE
0231  3A 0108           LDA     CRTRY       ;GET RETRY COUNT
0234  B7                ORA     A           ;IS IT ZERO?
0235  20EA              JRNZ    GSCT2       ;IF NOT, TRY AGAIN
0237  3A 010A           LDA     ERRCD       ;ELSE, GET ERROR CODE
023A  B7                ORA     A           ;SET UP ZERO FLAG
023B  C9                RET
                        ;
                        ;SKWTBL - TABLE OF LOGICAL-PHYSICAL SECTOR NUMBER ASSOCIATIONS
                        ;         USED BY LIFEBOAT. EACH ENTRY IS THE PHYSICAL SECTOR
                        ;         NUMBER TO READ FOR THE LOGICAL SECTOR NUMBER, USED AS
AN                      ;         INDEX INTO THE TABLE.
                        ;
023C  00        SKWTBL: .BYTE   0
023D  05                .BYTE   5
023E  01                .BYTE   1
023F  06                .BYTE   6
0240  02                .BYTE   2
0241  07                .BYTE   7
0242  03                .BYTE   3
0243  08                .BYTE   8
0244  04                .BYTE   4
0245  09                .BYTE   9
                        ;
                        ;
                        ;LOAD ONE NORTHSTAR SECTOR
                        ;       B=TRACK
                        ;       C=SECTOR
                        ;       RETURNS NON-ZERO STATUS IF ERROR
                        ;
                        ;
0246  CD C93D   LONE:   CALL    SEEK        ;START MOTOR, FIND TRACK/SECTOR
0249  C0                RNZ                 ;RETURN IF ERROR
024A  37                STC                 ;PRESET MSB FOR 1'ST BYTE
024B  11 DB50           LXI     D,INDAT     ;SET DISK READ ADDRESS
024E  1A                LDAX    D           ;GET FIRST BYTE
024F  1F                RAR                 ;ROTATE BACK INTO 'PHASE'
0250  F5                PUSH    PSW         ;AND SAVE LSB OF NEXT BYTE IN CY
0251  FEFB              CPI     0FBH        ;SYNC CHARACTER?
0253  203F              JRNZ    SYNERR      ;EXIT IF NOT SYNC CHAR
0255  F1                POP     PSW         ;RESTORE MSB TO CY
0256  1A                LDAX    D           ;SECOND BYTE
0257  1F                RAR
0258  F5                PUSH    PSW
0259  FEFB              CPI     0FBH        ;SYNC CHARACTER?
025B  2037              JRNZ    SYNERR      ;EXIT IF NOT
025D  F1                POP     PSW
```

cont'd on page 129

```
025E  01 0000          LXI   B,0       ;CLEAR B & C
                       ;
                       ;READ FIRST HALF OF SECTOR INTO BUFFER
                       ;
0261  1A       RLUP1:  LDAX  D         ;GET NEXT BYTE
0262  1F               RAR
0263  77               MOV   M,A       ;SAVE IT
0264  23               INX   H         ;ADVANCE BUFFER POINTER
0265  05               DCR   B         ;DECREMENT BYTE COUNT
0266  20F9             JRNZ  RLUP1     ;LOOP TIL 256 BYTES READ
                       ;
                       ;READ SECOND HALF OF SECTOR INTO BUFFER
                       ;
0268  1A       RLUP2:  LDAX  D         ;GET A BYTE
0269  1F               RAR
026A  77               MOV   M,A       ;MOVE TO BUFER
026B  23               INX   H         ;ADVANCE BUFFER POINTER
026C  05               DCR   B         ;DECREMENT BYTE COUNT
026D  20F9             JRNZ  RLUP2     ;LOOP TIL 256 BYTES READ
                       ;
                       ;COMPUTE CRC FROM DATA IN BUFFER
                       ;
026F  1A               LDAX  D         ;READ CRC
0270  1F               RAR
0271  32 0109        0  STA   CRC       ;SAVE IT
0274  01 0000          LXI   B,0       ;CLEAR B & C
0277  2A 010B          LHLD  BIGPTR    ;RESET BUFFER POINTER
027A  7E       CRC1:   MOV   A,M       ;GET A BYTE
027B  A9               XRA   C         ;X-OR WITH CRC
027C  07               RLC             ;ROTATE LEFT 1
027D  4F               MOV   C,A       ;AND SAVE NEW CRC
027E  23               INX   H         ;ADVANCE BUFFER POINTER
027F  05               DCR   B         ;AND DECREMENT BYTE COUNT
0280  20F8             JRNZ  CRC1      ;LOOP TIL 256 BYTES COUNTED
0282  7E       CRC2:   MOV   A,M       ;REPEAT FOR SND HALF OF BLOCK
0283  A9               XRA   C
0284  07               RLC
0285  4F               MOV   C,A
0286  23               INX   H
0287  05               DCR   B
0288  20F8             JRNZ  CRC2
028A  3A 0109          LDA   CRC       ;GET CRC READ FROM DISK
028D  B9               CMP   C         ;COMPARE WITH COMPUTED VALUE
028E  3E53             MVI   A,"S"     ;ERROR CODE
0290  C0               RNZ             ;RETURN IF ERROR
0291  3E00             MVI   A,0       ;ZERO FOR GOOD READ
0293  C9               RET
                       ;
                       ;EXIT PROCEDURE FOR BAD SYNC BYTES
                       ;
0294  F1       SYNERR: POP   PSW       ;CLEAN UP STACK
0295  3E56             MVI   A,"V"     ;SYNC ERROR CODE
0297  C9               RET
                       ;
                       ;
                       ;WRITEV - WRITE ONE SECTOR ON DISK IN VISTA FORMAT FROM 512
                       ;         WORD BLOCK STARTING AT 'BIGPTR'. RETURNS 'BIGPTR'
                       ;         UPDATED TO START OF NEXT BLOCK, AND STATUS CODE IN A.
                       ;         IF DISK IS WRITE PROTECTED, A CONTAINS "P".
                       ;
                       ;
0298  3A 0103  WRITEV: LDA   TRACK
029B  47               MOV   B,A       ;SET TRACK NUMBER
029C  3A 0104          LDA   SECTOR
029F  4F               MOV   C,A       ;SET SECTOR NUMBER
02A0  CD 02BE          CALL  OFFSET    ;USE BIAS TO COMPUTE OFFSET
02A3  C5               PUSH  B
02A4  CD C93D          CALL  SEEK      ;DUMMY SEEK, FOR TIMING
02A7  C1               POP   B
02A8  2A 010B          LHLD  BIGPTR    ;PICK UP BUFFER POINTER
02AB  CD CBF1          CALL  WSEC      ;CALL VISTA WRITE ROUTINE
02AE  2A 010B          LHLD  BIGPTR    ;BUFFER POINTER
02B1  01 0200          LXI   B,512     ;NUMBER OF BYTES WRITTEN
02B4  09               DAD   B         ;UPDATE POINTER
02B5  22 010B          SHLD  BIGPTR    ;AND SAVE
02B8  FE50             CPI   "P"       ;TEST FOR WRITE PROTECTED
02BA  C8               RZ              ;RETURN IF SO
02BB  3E00             MVI   A,0       ;ELSE
```

cont'd on page 130

```
02BD  C9              RET             ;RETURN WITH A=0
                      ;
                      ;
                      ;OFFSET - GIVEN A TRACK NUMBER IN (B) AND A SECTOR NUMBER IN
                      ;         (C) COMPUTES THE ABSOLUTE SECTOR NUMBER, THEN ADDS IN
                      ;         THE CONTENTS OF 'BIAS'. THIS RESULT IS THEN CONVERTED
                      ;         BACK TO TRACK AND SECTOR AND RETURNED IN (B) & (C).
                      ;
02BE  C5        OFFSET: PUSH  B         ;SAVE TRACK & SECTOR
02BF  48                MOV   C,B
02C0  0600              MVI   B,0       ;NOW (BC)=TRK
02C2  C5                PUSH  B
02C3  E1                POP   H         ;COPY TRK INTO (HL)
02C4  09                DAD   B         ;(HL)=2*TRK
02C5  29                DAD   H         ;(HL)=4*TRK
02C6  29                DAD   H         ;(HL)=8*TRK
02C7  09                DAD   B         ;(HL)=9*TRK
02C8  09                DAD   B         ;(HL)=10*TRK
02C9  C1                POP   B         ;GET ORIGINAL ARGUMENTS BACK
02CA  0600              MVI   B,0       ;NOW (BC)=SECTOR
02CC  09                DAD   B         ;(HL)*(10*TRK)+SECTOR=ABS. SECTOR
02CD  ED4B 0105         LBCD  BIAS
02D1  09                DAD   B         ;NOW ADD IN BIAS
02D2  CB7C              BIT   7,H       ;TEST SIGN OF RESULT
02D4  2804              JRZ   OS1       ;BRANCH IF RESULT POSITIVE
02D6  2600              MVI   H,0       ;ELSE, IF IT WAS NEGATIVE
02D8  2E00              MVI   L,0       ;FORCE TO 0
02DA  0600      OS1:    MVI   B,0       ;CLEAR QUOTIENT
02DC  0E00              MVI   C,0       ;AND REMAINDER
02DE  11 000A           LXI   D,10      ;DIVISOR
02E1  AF                XRA   A
02E2  ED52      DIV1:   DSBC  D         ;SUBTRACT FROM (HL)
02E4  3803              JRC   DIV2      ;DONE IF RESULT GOES NEGATIVE
02E6  04                INR   B         ;ELSE, INCREMENT QUOTIENT
02E7  18F9              JMPR  DIV1      ;AND CONTINUE
02E9  19        DIV2:   DAD   D         ;ADD BACK TO GET REMAINDER
02EA  4D                MOV   C,L       ;MOVE TO C
02EB  C9                RET             ;AND RETURN
                      ;
                      ;
                      ;INPUT - WAIT FOR OPERATOR INPUT INTO CMDBUF
                      ;
02EC  3E00      INPUT:  MVI   A,0
02EE  32 010E           STA   CMDBUF+1  ;RESET CHARACTER COUNT
02F1  21 010F           LXI   H,CMDBUF+2;SET UP TO CLEAR CDBUF
02F4  11 0110           LXI   D,CMDBUF+3
02F7  01 0005           LXI   B,5
02FA  3620              MVI   M,20H     ;ASCII BLANK
02FC  EDB0              LDIR            ;CLEAR CMDBUF
02FE  0E0A              MVI   C,10      ;10=BUFFERED READ
0300  11 010D           LXI   D,CMDBUF
0303  CD 0005           CALL  CPMIO     ;WAIT FOR INPUT, TERMINATED BY <CR>
0306  C9                RET
                      ;
                      ;ZERBLK - WRITE 512 ZEROS INTO BLOCK, STARTING AT 'BIGPTR'.
                      ;
0307  2A 010B   ZERBLK: LHLD  BIGPTR    ;GET ADDRESS OF BLOCK
030A  ED5B 010B         LDED  BIGPTR
030E  13                INX   D
030F  01 01FF           LXI   B,511     ;NUMBER TO WRITE - 1
0312  3600              MVI   M,0       ;ZERO FIRST BYTE
0314  EDB0              LDIR            ;BOOT COPY TILL (BC)=0
0316  C9                RET
                      ;
                      ;WAIT - WAIT FOR CARRIAGE RETURN FROM KEYBOARD THEN RETURN
                      ;
0317  0E0B      WAIT:   MVI   C,11      ;11=TEST CONSOLE STATUS
0319  CD 0005           CALL  CPMIO
031C  CB47              BIT   0,A       ;TEST FOR CHARACTER PRESENT
031E  28F7              JRZ   WAIT      ;LOOP TIL CHARACTER RECEIVED
0320  0E01              MVI   C,1       ;1=READ A CHARACTER
0322  CD 005            CALL  CPMIO     ;GO GET IT
0325  FE0D              CPI   0DH       ;WAS IT CARRIAGE RETURN?
0327  20EE              JRNZ  WAIT      ;WAIT FOR CARRIAGE RETURN
0329  C9                RET
                      ;
                      ;
```

cont'd on page 131

```
                ;PROMPT AND WARNING MESSAGES
                ;
032A  0D0A      MSG1:   .WORD  CRLF
032C  4C6F6164204E      .ASCII "Load Northstar disk (source) in drive A."
0354  0D0A              .WORD  CRLF,CRLF
0358  4C6F61642062      .ASCII "Load blank disc in drive B."
0373  0D0A              .WORD  CRLF,CRLF
0377  507265737320      .ASCII "Press 'RETURN' to continue."
0392  0D0A              .WORD  CRLF,CRLF
0396  24                .BYTE  "$"
0397  0D0A      MSG2:   .WORD  CRLF
0399  4C6F61642043      .ASCII "Load CP/M system disk in drive A,"
03BA  0D0A              .WORD  CRLF
03BC  7468656E2070      .ASCII "then press 'RETURN' to reboot."
03DA  0D0A              .WORD  CRLF,CRLF
03DE  24                .BYTE  "$"
03DF  0D0A      MSG3:   .WORD  CRLF
03E1  456E74657220      .ASCII "Enter absolute sector offset $"
0402  0D0A      MSG4:   .WORD  CRLF
0404  497320736F75      .ASCII "Is source disk in Lifeboat CP/M format ? $"
042E  0D0A      EMSG1:  .WORD  CRLF,CRLF
0432  4552524F5220      .ASCII "ERROR - Output disk is write protected."
0459  0D0A              .WORD  CRLF,CRLF
045D  24                .BYTE  "$"
                ;
                ;TRACK BUFFER STARTS HERE (10 SECTORS = 5120 BYTES)
                ;
045E            BIGBUF=
185E            EBUFF=BIGBUF+5120
                      .END
```

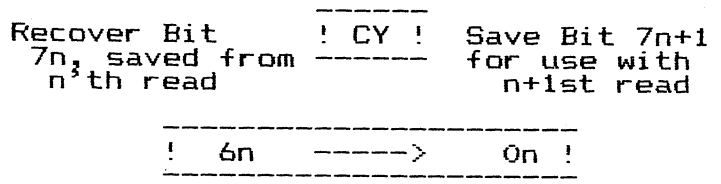

Figure 2

* *

TRS-80 Level II Cassette BASIC Update

By Robert Lansdale

My article 'TRS-80 Level II Cassette Basic' in the April 1982 of Sorcerer's Apprentice had some severe bugs in it.

What you read was my personal documentation for TRS-80 BASIC, TRS-80 Disk BASIC and Rom-Pac Disk BASIC. It was sent into the magazine by a friend who had the 'old' package. The documentation has since been upgraded to a more 'user' oriented paper describing all aspects of the conversion. The article didn't mention where to get the package. It may be obtained from me for a fee of $25.00 payable in U.S. funds.

Robert Lansdale Jr.
18 Ashfield Drive,
Etobicoke, Ontario.
M9C-4T6, Canada.
(416)781-8788

For your $25, you will receive the new updated package which contains:

- TRS-80 Level II BASIC for 32k (or greater) Sorcerer (cassette version)

- TRS-80 Level II Disk BASIC for 48k Sorcerer & CP/M.

- Example TRS-80 program using TRS-80 BASIC.

- 12 pages of documentation for above.

- Sorcerer Rom-Pac BASIC relocated to 8000H and Disk I/O added to interface to CP/M. (48k or more needed). This is useful to DISCUS 2D owners who can no longer use the Rom-Pac because of memory conflict between the Pac and the disk boot ROM.

Please indicate if you have:

1) Printer capable of printing an ASCII file

2) A 8" single density disk or Double density 8" DISCUS 2D or DISCUS 2+2 (please send 8" disk if you satisfy condition number two).

Pac-BASIC relocated to 8000H with the CP/M disk accessing command is useful to Sorcerer users who have a disk controller occupying anywhere from C000-DFFF (such as Morrow DISCUS users) therefore not being able to insert the BASIC Rom-Pac. The Pac-BASIC requires a 48k Sorcerer with CP/M.

The TRS-80 basic will run virtually any TRS-80 Level II Model I BASIC program without ANY modifications. Machine language games can be converted without too much trouble. Such an example was the Sub-logic FS1 3-d flight simulator. The conversion process is now described in the TRS-80 documenation.

Please note the following bugs in the article:

1) In the TRS-80 interpreter changes section, most of the changes and jumps are no longer valid for the TRS-80 Disk BASIC but are still ok for the cassette version.

2) In the 'CHART' on page 53, the numbers '1 2 4 8 16' should be moved to the left 7 spaces so that they are aligned under their proper row. Also the numbers down the right side of the chart are the column number of the keys and have nothing to do with the keyboard characters.

3) On page 54, left column in emphasized mode the title'PAC BASIC TO TRS-80 DISK BASIC' should read 'PAC-BASIC AND TRS-80 DISK BASIC DOCUMENTATION'. Pac-BASIC and TRS-80 BASIC are two entirely different programs and do not interact as the title may suggest.

4) Page 54, second column. Under the heading 'TRS-80 BASIC' in emphasized print is:

'Radio Shack BASIC has been modified so the PAC BASIC rou tines will be able to reside inside the BASIC itself' should in fact be:

'Radio Shack BASIC has been modified so the CP/M Disk I/O routines will be able to reside inside the BASIC itself'

5) Most people thought that you could load Rom-Pac programs into TRS-80 BASIC. This can be done, but the tokens ARE NOT COMPATIBLE and the program will look like garbage when listed.

These programs have been extensively used for well over two years and no flaws have shown up as of yet. Please inform me if any problems occur.

South Valley Electronics

2110 E Walsh Avenue
Santa Clara, CA 95056
(408)727-0906

The GREAT Exidy Sale!

Description	Quantity	Price
48k Sorcerer Model II	150	$ 125.00
32k Sorcerer Model I	50	$ 50.00
Disk Controller Board SS	40	$ 150.00
Keyboards	45	$ 50.00
Tops	400	$ 5.00
Bottoms	70	$ 5.00
DDS cases with hardware	200	$ 25.00
FDS Cabinet, top, bottom, back	40	$ 15.00
Cassette Interface Boards	100	$ 15.00
Mod II Power Supply Boards	n/a	$ 10.00

NOTE: All items AS IS, NOT GUARANTEED, and SUBJECT TO PRIOR SALE!

CALL ahead to see if a particular item is available. The list above is NOT all inclusive, so if you don't see it on this list, we may still have it in stock.

Send check or moneyorder. California residents add applicable saes tax. Overseas orders make drafts for US funds payable to Wells Fargo Bank, (Oak Mead) Park Stage Stop, 2930 Bowers Avenue, Santa Clara, CA 95056, Account #052454.

The GREAT Exidy Sale!

SORCERER'S APPRENTICE
P. O. Box 33
Madison Heights, MI 48071 USA

1983 Renewal Application

NAME: _____

ADDRESS: _____

CITY: _____

STATE/PROVINCE: _____

POSTAL CODE: _____

COUNTRY: _____

TELEPHONE: Home---------- Bus:----------

If you do NOT wish the above released, please sign here:

_____

Would you be willing to contribute to the newsletter? Yes No

If YES, in what category: (check all that apply) ---- Articles, ---- Programs, ---- Surveys, ---- Advisor, ---- Editor, Hardware, ---- Software, ---- Librarian: ---- Cassette, ---- Soft Sector Exidy Disks, ---- Micropolis Mod II Disks, Vista 10 Sector Hard Disks, ---- Other: (Please specify)

_____

* * * * * * * * * * * * * * * * * *

The following information is optional and will NOT be released.

EQUIPMENT: Sorcerer I II RAM: 8k 16k 32k 48k >48k

Expansion Unit: _____ Disk System: _____

Printer: _____ DOS: _____

Peripherals & Software: _____

_____

_____

Age Group: <25 26-35 36-45 46-55 >55; Occupation: _____

AMOUNT ENCLOSED: $----------

Volume 5:

---- Bulk Mail (USA only) $18.00; ---- First Class Mail (USA, Canada & Mexico) $24.00; ---- Air Mail $32.00

BACKISSUES ----- USA THIRD CLASS MAIL:

VOLUME 4: ---- $18.00; VOLUME 3: ---- $15.00; VOLUME 2: ---- $12.00; VOLUME 1: ---- $12.00

*** ADD $3.00 FOR FIRST CLASS MAIL IN USA, CANADA AND MEXICO; ADD $6.00 FOR AIR MAIL ***

Make checks or money orders (US Funds drawn on US banks) payable to: SORCERER'S APPRENTICE.

✲ ✲ ✲ ✲ ✲ ✲ ✲ ✲ ✲ ✲ ✲ ✲ ✲ ✲ ✲ ✲ ✲

Getting the %MEMOFF working with Micropolis Mod II. Or, how to read the directions and still blow it.

By Don Myklebust
19710 Guthrie
Strathmore, CA 93267
(209) 568-1389

Finally got a Model 2 Sorcerer built up a couple of months ago. For several reasons, I wanted badly to have the disk boot usable at BC00. It didn't work until last Friday. Here's why.

Got the BC00 boot working with the phone help of Steve Day at South Valley Electronics. Don't remember just what he said that tipped me off, but the main thing is that if you do exactly what the Tech Note says and DON'T THINK ABOUT IT, it'll work just fine. The only thing different is the IC numbering. The note is for the mother board unit as is the Model 2 manual schematic.

For starters, a number of people have the idea that the black Sorcerer 2 manual is wrong in the edge connector (J3) numbering on sheet 7. They say that the numbers are 1 off such that pin 45 on the schematic is actually pin 46.

The second part of this is that the S100 card is numbered unconventionally (true) at J3 and that you should assume that it's wrong (not particularly) and mentally turn it around when you do your jumpering, etc. (not true). This stems from the fact that J3 is even-numbered on the front of the card instead of starting with pin 1 on the front as is more usual. What they would have you do is put the jumper on pin 45 on the front of the board after mentally reversing the connector numbering. What this actually does is put the jumper on pin 46 of the Sorcerer.

To sum up:

1. The Sorcerer 2 manual schematic sheet 7 is correct in its J3 pin numbering.

2. Also correct in pin numbering is the sheet labeled Expansion Chassis Mother Board'. Of course, the IC numbering is different than for the plug-in card.

3. Tech Note 3 is for the mother board S100 unit. IC 1A called out in the note is 6D on the plug-in board.

4. You really do end up connecting Sorcerer J3 pin 45 to S100 J3 pin 46.

5. One could say that the way J3 was numbered on the plug in board was a goof. OK, but at least they kept the same convention on the mother board which means that (except for IC layout) the documentation for one is good for the other.

6. The tech note should go something like:
A. On S100 jumper S100 bus pin 21 to 1A pin 11 (6D pin 11 on older plug in card).

B. Jumper 1A pin 9 (6D pin 9 on card) to J3 pin 46.

✲ ✲ ✲ ✲ ✲ ✲ ✲ ✲ ✲ ✲

continued from page 112

```
240 DATA 50,35,15,62
250 DATA 43,5,13,30,50,49,
         12,62
260 DATA 47,0,25,3,31,2,0,
         31,62
270 DATA 12,2,0,12,58,41,62
280 DATA 62,62,62
290 DATA 21,0,9,41,62
300 DATA 47,0,12,62
310 DATA 31,37,60,33,25,10,
         20,62
320 DATA 42,54,55,55,62
330 DATA 34,54,55,55,62
340 DATA 45,11,9,57,62
350 DATA 6,33,41,62
360 DATA 15,52,42,43,47,31,
         62
370 DATA 2,1,31,3,31,60,41,
         3,30,47,0,17,3
380 DATA 52,53,55,3,45,50,
         49,13,62
390 DATA 31,37,60,41,42,16,
         62
400 DATA 31,2,13,57,11,31,
         50,0,41,18,58,62
410 DATA 62,62,62,62,62,63
```

NOTE: Westland Electronics stocks the Votrax SC-01, along with a complete catalog of electronic components.

Westland Electronics
37387 Ford Rd.
Westland, Michigan 48185
in Mich. (313) 728-0650
outside Mich. (800) 521-0664

The Sorcerer's Apprentice Newsletter is published eight times per year by the Sorcerer's Apprentice Users Group. A Computer Bulletin Board Service is available, courtesy of Robert Hageman. To access the CBBS just call (313)535-9186.

Membership Rates are $18.00/Volume (Bulk Mail USA only). $24.00/Volume USA, Canada and Mexico (First Class Mail). All others $32.00/Volume (Air Mail).

Backissues are available as follows:

Aresco Volume I, Issues 4&5 only (30 copies only)...$ 3.00 ea.
SA Volume I, issues 1-7.....(45 copies only)...$12.00/Vol.
SA Volume II, issues 1-5.....(45 copies only)...$12.00/Vol.
SA Volume III, issues 1-8.....$15.00/Vol.

(Prices include Third Class postage in USA only. First Class Mail for USA, Canada and Mexico add $3.00/Volume. All others add $6.00/Volume for Air Mail).

C/PMUG and SIGM disks are available from Bruce Blakeslee, 906 Crestwood Rd., Westfield, NJ 07090, on Micropolis Mod II format only. Send a formatted disk and $3.00 plus $1.50 for mailing, or send $8.00 plus $1.50 for mailing, for any disk. Make checks or moneyorder payable to Bruce Blakeslee.

STS, the Sorcerer Telecommunications Software on cassette, is available from Jonathan Burnett, 5422 Missouri Ave., Jacksonville, FL 32205, for $30.00 plus $1.50 for mailing. Make checks or moneyorder payable to Jonathan Burnett.

PAPS, the Personal Accounts Payable System is available from Sorcerer's Apprentice, P.O. Box 33, Madison Heights, MI 48071 for $50.00 plus $1.50 for mailing. You'll receive three diskettes (Micropolis Mod II format only). Make checks or moneyorder payable to Sorcerer's Apprentice.

All members are encouraged to submit articles, programs and/or newsworthy items for publication. Material may be submitted on hardcopy, Wordprocessor files on cassette (please record both at 300 and 1200 baud), Micropolis Mod II diskette (Wordstar or Spellbinder), or uploaded to the CBBS (313-535-9186). When sending magnetic media, be sure to wrap it in foil to prevent degradation of the information while in transit. Whenever possible, also include a hardcopy. Cassette and Disks will be returned, hardcopy only if requested and accompanied by a SASE.

Commercial advertising rates available upon request. Classified advertising - send $1.00/35 column line or part-line.

To contact us by telephone, call Don Gottwald (313)286-9265, Ralph LaFlamme (318)856-4954, or leave a message on the CBBS (313)535-9186.

SEND ALL CORRESPONDENCE TO:

Sorcerer's Apprentice
P. O. Box 33
Madison Heights, MI 48071
USA

BULK RATE
U.S. POSTAGE
PAID
WYANDOTTE
MICHIGAN 48192
PERMIT NO. 23

RETURN AND FORWARDING POSTAGE GUARANTEED
ADDRESS CORRECTION REQUESTED