```
 SSSS   S   S  S     S     SSSS   S   S  S   S   sSSS   SSS   SSSS
S    S  S   S  SS   SS     S   S  S   S  S   S  S      S   S  S    S
S       S   S  S S S S     S   S  S   S  S   S  S      S   S
 SSSS    SSS   S  S  S SSS SSSS   SSSSS   SSS    SSSS    S    SSSS
     S   S     S     S     S      S   S   S        S   S        S
S    S   S     S     S     S      S   S   S    S   S   S    S   S
 SSSS    S     S     S     S      S   S   S     SSSS   SSS   SSSS
```

THE SYM-1 USERS' GROUP NEWSLETTER

ISSUES NUMBERS 5 & 6 - SEPTEMBER/OCTOBER/NOVEMBER/DECEMBER 1980

SYM-PHYSIS is a bimonthly publication of the SYM Users' Group, P. O. Box 315, Chico, CA. 95927. SYM-PHYSIS and the SYM Users' Group (SUG) are in no way associated with Synertek Systems Corporation (SSC), and SSC has no responsibility for the contents of SYM-PHYSIS. SYM is a registered trademark of SSC. SYM-PHYSIS, from the Greek, means the state of growing together, to make grow, to bring forth.

We welcome for publication all articles dealing with any aspect of the SYM-1, and its very close relatives. Authors retain all commercial copyrights. Portions of SYM-PHYSIS may be reproduced by clubs and educational institutions, and adaptations of programs for other computers may be freely published, with full credit given and complimentary copies provided to SYM-PHYSIS and the original author(s). Please include a self-addressed stamped envelope with all correspondence.

Editor/Publisher: H. R. "Lux" Luxenberg
Publisher/Manager: Jean Luxenberg
Associate Editors: Thomas Gettys, Jack Brown

SUBSCRIPTION RATES:

USA/Canada $9.00 for Issues Numbers 1 through 6; overseas airmail $12.50, payable in US dollars to "SYM Users' Group," P. O. Box 315, Chico, CA 95927, Telephone (916) 895-8751.

The Introductory Issue, Number 0, dated September 1979, is not part of the first year's subscription, but is available separately, postpaid first class or air mail, for $1.50 in USA/Canada, $2.00 elsewhere.

COMMENTS AND REMARKS

This double issue marks the end of our first year of publication as a bimonthly. Many of our readers requested that we go monthly....No way!!!! Our own inclination was to drop the newsletter entirely, and to retire to a subtropical paradise somewhere, far from a source of the necessary power to operate a SYM.

Computer newsletters seem to have short (but merry?) lives. Eric Rehnke's KIM-1/6502 USER NOTES did not quite make it through its third year. Eric is now with Rockwell International, editing their in-house newsletter, INTERACT, for the AIM-65. We read it regularly, for any ideas which we can possibly asSYMilate into our own systems. VIPER, the newsletter for the RCA COSMAC VIP (Versatile Interface Processor, an exellent 1802-based single board computer) lasted only two years. The Pet Gazette also had a short life span. Fortunately, COMPUTE. is carrying on with the mission of these newsletters.

Well, we're not yet ready to quit, though we do feel mighty tired at times, so we'll try for another year. To ease the frequency of the pain

somewhat, while possibly increasing its intensity accordingly, we will go quarterly. Insted of 24 pages six times a year, SYM-PHYSIS will be 36 pages, four times a year. Any advertising materials will be on extra pages, and no editorial material will be sacrificed.

We appreciate the many letters and telephone calls which we have received, thanking us for the services provided by the Users' Group. These kind words do make our efforts seem worthwhile, and provide the incentive to continue. We are now only about a month behind in our correspondence, however, for which we offer our apologies, and two or three months late on RAE NOTES No. 3. At least, we are able to keep up with your telephoned questions!

Incidentally, if anyone is preparing an index (for personal use) in RAE format, on cassette, of the material in Issues 0 through 6, we would be pleased to publish it in Issue 7; we're sure other readers would also appreciate it.

SYM AS A PERSONAL COMPUTER & THE EDUCATIONAL/ACADEMIC MARKET

SYM-1, like many other single board computers, was intended to serve primarily as a simple, versatile, relatively inexpensive, system test and evaluation vehicle, and as a "learning system" for the particular family of chips sourced by its manufacturer.

As was the case with several other single board computers, users quickly realized that SYM could form the basis for a powerful "personal" computer; personal, not only in the sense that it is personally affordable, but because it can be "personalized" to match the user's needs, and personality. As of this writing, there are about 20,000 SYMs out there, and we have about 1000 subscribers. Many of our subscribers have more than one SYM, and many non-subscribers read borrowed copies. SYM-PHYSIS thus reaches at least 5% of the SYM owners, and perhaps as many as 10%. Many owners are using their SYMs in far more sophisticated ways than could ever have been forseen. Their SYMs are being expanded into full-blown systems (no two alike) for unique applications, and with, of course, mostly non-Synertek add-ons.

The only major accessories or expansion products marketed by Synertek Systems Corporation which are priced at the personal computerists' price level are BAS-1, RAE-1, and the KTM-2s (now, also the -3s). As is the case with Apple, Pet, TRS-80, etc., numerous entrepreneurs are providing SYM compatible accessories and expansions at prices far lower than the "original source" can profitably meet at low volume. Synertek Systems sees its major market as the large volume, OEM/Industrial user, not the educational/academic or personal system user, and has oriented its marketing and customer support services accordingly, as many SYM owners are discovering. Where then, are the non-OEM/Industrial users to find the technical (both software and hardware) support for their systems?

The optimum support that a computer manufacturer can provide for its educational/academic and personal system customers is to support a users' group, which can then provide a vehicle for "self-support." We feel that Synertek Systems is providing us (as the Users' Group) all of the support (moral, and technical) we need, and that this support will continue to increase as we continue to prove that we, in turn, are passing this support on to SYM users. In short, we believe that SSC is providing excellent, and an increasing amount, of user support, through their support of the SYM Users' Group.

We feel that the educational/academic market for systems built around the SYM-1 : BAS-1 : RAE-1 : KTM-2/80 nucleus is far greater than

(continued on page 5/6-24)

VERY CHEAP EPROM ERASER

In BYTE, Vol.2, No. 1, January 1977, many years ago, Lawrence Burbey described how to 'Build the "Coffee Can Special" EROM Eraser.' We finally were able to check out Mr Burbey's suggestions last week, and can suggest an even cheaper approach; skip the coffee can!

Nearly four weeks after we asked our local appliance dealer to order some for us, we received three of the Sylvania 4 W Germicidal Lamps, Type B. These lamps are less than 1 1/2 inches in diameter, and require an adaptor (their base is like that of an outdoor Christmas tree lamp, known as "Intermediate") to fit into a standard lamp socket. We obtained two wall mounting lamp sockets and wired them in series, put the Type B in one, and a 25 W lamp in the other. The 25 W lamp glowed brightly, but the Type B did not. When we switched to a 40 W lamp the Type B lamp began to glow after a few seconds of warm-up time. The series lamp is required for current limiting, otherwise the Type B will go "poof" and probably spread mercury vapor all around. The 40 W seems right, so I didn't move up to a 60. The Type B ran very cool, so I placed a "loaded" (incorrectly) 2716 EPROM directly on top of the lamp. After 10 minutes the EPROM read all FF's when checked out in the SYM.

No, I didn't use the coffee can recommended by Burbee, nor did I use the pair of bread-baking tins recommended (for eye-safety) by L. B. Golter in 'Build a Low-Cost EPROM Eraser,' BYTE, April, 1980. The corrugated paper in which the Type B lamp was wrapped had the following warning:

> The eyes and skin should be protected from the direct rays of this germicidal lamp. Ordinary window glass, opaque materials, or clothing are adequate protection.

I used a piece of cardboard for protection. A second warning appears on the wrapper:

> This lamp emits some ultraviolet wavelengths which produce ozone in the air. Where the sharp odor of ozone is detected, personal exposure should be avoided as inhalation for repeated periods or over a half hour at a time may cause respiratory irritation.

Yes, there was a minor ozone odor within an inch or two of the lamp, but none at a distance of one foot, so I see no problem here. I have no data on lamp life, but I have written Sylvania for a spec sheet on the Type B. They are supposed to last years in clothes driers (as "germicidal" lamps), however. And one more "fact": U. S. postage stamps fluoresce beautifully in the "short-wave" ultraviolet radiation emitted by this type of lamp, but not under longer wave, so-called "black light" radiation.

MORE ON THE TYPE "B" LAMP

We now have more information on the Type "B" Lamp recommended above as a "cheap" EPROM Eraser. Type B refers to the type of glass used; this type of glass transmits ozone generating radiation at 185 nm (1 mw for the 4 W lamp), as well as "erasing" radiation at 254 nm (100 mw for the 4 W lamp). The ozone itself has no odor; the odor commonly attributed to ozone is actually that of the nitrous oxide it forms in combination with atmospheric nitrogen. The Type B lamp will neutralize objectionable odors, such as stale tobacco smoke, or mildew, etc., and, from what I can gather, if you cannot smell the nitrous oxide, the ozone has reverted to oxygen, and presents no hazard.

The specification sheet for the lamp states that the glass used in ordinary eyeglasses provides adequate eye protection, and that the skin can stand continuous radiation at an irradiance of 0.1 uW/sq.cm. Since the 4 W lamp produces an irradiance of 1.2 uW/sq.cm. at one meter, it would be "skin-safe" at 3 meters.

Anyway, I am now convinced that suitable eye/skin protection is provided by the cardboard tube from a toilet tissue roll, which fits nicely over the lamp. The tube can be cut to be slightly longer than the lamp, and a piece of conductive foam in which the EPROM to be erased is inserted can serve as the "lid." What could be cheaper? And deodorize the room at the same time, if the air is circulating?

The lamp is rated at 0.35 A at 10.5 V (I measured 10.5 VAC when in series with the 40 W incandescent lamp), and has a rated life of 4000 hours.

CASSETTE INTERFACE COMMENTS

Only two of our purchasers of cassette software have not been able to read our cassettes. We sent these two subscribers new cassettes, plus a cassette with a 6 minute synch signal on one side, and a repeated sequence of one page blocks on the other side, to give them plenty of time to adjust volume controls. This must have done the job, since we have not heard further from them.

We received a review copy of 6502 FORTH from Eric Rehnke, and could not read it. He replaced it with a KIM format tape, which we did read. Also, we have had problems reading Jack Brown's tapes. We tried six recorders and three SYMs with no luck. One of our subscribers in Wales will be visiting us this month, and he asked us to purchase a number of items for him including two Sanyo 1530A recorders. We purchased for him instead two Model 1540A recorders. These are a better buy, since they have EJECT, and include the AC Adaptors.

While checking these out I tried Jack Brown's tapes on these machines. Through serendipity (the oscilloscope was connected to the right point and the read plug didn't fit right in the earphone jack), I discovered that the input wave shape was much more nearly symmetrical, when the plug was in only part way, leaving the internal speaker connected. I was able to read Jack's tapes then, but the noise was deafening. I will look into this matter further, and report next issue. In the meantime I am sending Jack one of my recorders to ensure getting a readable copy of the source code for SYM-FORTH. (Since my writing the above, Jack has switched from his "high-quality" recorder to one of Radio Shack's cheapest models, and I now read him just fine. Seems that price correlates inversely with reliability!)

On the other hand, I received a cassette today from Stephen Cole of Hants, England at 2800 Baud, twice normal SYM speed, which I read easily. We have always thought of the KIM format as a fallback if the SYM format doesn't work, but the KIM speed takes 21 times as long. Cole's tape made me realize that we could also send the SYM format at half or quarter speed, if necessary. In computing SYM vs KIM format speeds note that each SYM byte requires 9 bits SYM and 18 bits KIM.

RECOMMENDED READING

The two bimonthlies, COMPUTE., which deals with 6502-based System Computers, and compute II., which covered the 6502- and 1802-based Single Board Computers, are merging into a single monthly, COMPUTE. This is welcome news, indeed, since, as you will see elsewhere in this issue, SYM owners can benefit from programs for, and information about Apple (and Pet and Atari, and OSI, and the others). Address, page 1-8.

ENHANCING THE SYM-1 I/O CAPABILITIES AT (ALMOST) NO COST

Perhaps you, too, may have wondered why Synertek "spoiled" one of the two 8-bit parallel ports in VIA #1, at $A000, by assigning PB 6 to ON BOARD CASSETTE IN. The reason was to provide "compatibility" with KIM-1, which lacked PB 6 for a very good reason.

KIM used a pair of 6530's to provide its I/O-TIMER capabilities, and to hold the 2 K KIM (Keyboard Interface Monitor) firmware. The 6530, which Rockwell calls the RRIOT (for ROM-RAM-I/O-TIMER), is very much like the 6532, which Rockwell calls the RIOT (for obvious reasons), except for also including a 1 K ROM. In the 6530, PB6 was sacrificed to free its pin to be used as one of the ROM addressing lines. Synertek "replaced" the pair of 6530s on the KIM with one 6522 (VIA) and one 6532 (RIOT), and put the 4 K VIM (Versatile Interface Monitor, now called SUPERMON) into a single 4 K ROM. Since KIM lacked PB 6, the SYM designers felt free to dedicate PB 6 to one of the system functions.

With my KIM I used an 8-bit digital to analog converter (DAC) on the A Port for music generation, and then added a 6-bit DAC on the B Port. I was then able to generate 64 by 64 vector graphics on an oscilloscope. I didn't even use the full 8-bit port because of memory poverty (only 5 K RAM!).

With one of my SYMs, there is 32 K of memory for handling vector graphics. While detailed vector graphics on an oscilloscope would present flicker problems, my intention is to deflect a large screen laser display, where the flicker would be part of the ambience. Furthermore, for direct film recording with the laser beam, flicker is again not a problem.

Thus, I want 256 by 256 laser graphics (4 K of memory and pixels) and the two 8-bit DACs. Also, see elsewhere in this issue, I want two 8 bit DACs for my new Stereo Music Synthesizer software package from Micro Technology, Unlimited (MTU). For these, and many other reasons, I have long been looking for, and and finally found, a simple way to recover the use of PB 6.

I recently received, almost as a gift, a used dual trace oscilloscope, which I have been using to "study" the performance of the cassette interface (some SYMs still have problems in this area). Meanwhile the scope helped me to find a very simple way to restore the full use of PB 6 without interference with the cassette function.

PB 6 is available on the Expansion Connector as pin E-X (AUD TEST). You can bring a wire from there (or from any of a number of places on SYM) to near the Application Connector. Since you are unlikely to need all of the signals on the A connector you can replace any one of them with PB 6. Now for the simple "fix". By biasing the inverting input (pin 3) of the LM 311 comparator (U26) very slightly negative with respect to the non-inverting input, the output of the comparator (which connects directly to PB 6) will "float" high when there is no cassette input; otherwise it will "read" the cassette input values properly. This biasing is easily, and very cheaply, done by connecting a one megohm resistor from pin 3 of U 26 to ground. This may not be the optimum value of resistance but it does work well. And, now, PB 6 is free for use as an output when not inputting tape. Have not tried using it as a direct input, because I don't need it, but I see no reason why active low signals (inactive during cassette input) could not be wire-ored to PB 6 as inputs. I do plan to use Port B indirectly as an input though, and either the unused CA 1 or CB 2, in analog to digital conversions, as soon as I work out the details.

ENHANCING THE SYM-1 TIMER CAPABILITIES AT NO COST

And now, let's examine the timer situation. KIM used two timers, one in each 6530. These timers are of the 6532 (not 6522) type. One was dedicated for system use, the other totally free. Many KIM programs were written using the interrupt capabilities of the 6530/6532 timer. These may be adapted directly to SYM. Contrary to the misinformation on the top of page 4-20 of the SYM-1 Reference Manual, all 32 of the 6532 addresses are available (although many of these are redundant). In fact, one address listed as being "N/A" is actually used by MON 1.1! See lines 0256 and 0257 of the cassette interface portion of MON 1.1.

The four Write Timer addresses at $A41C-$A41F Enable Interrupt. The four Write Timer addresses at $A414-$A417 Disable Interrupt. Read Timer at $A404 Disables Interrupt. Read Timer at $A40C Enables Interrupt). So go ahead and wire up the IRQ output of the 6532, and take advantage of the timer interrupt and the PA 7 edge detect interrupt capabilities, if you need them.

Incidentally, while MON 1.0 used the 6522 timer in its cassette software, MON 1.1 uses the Divide by 8 feature of the 6532. Since the count of this clock is compared with HSBDRY for the 0/1 decision, it is apparent that HSBDRY is the number of 8 (not 5 !) microsecond intervals. And, now, the default value for HSBDRY ($46 = 70 DEC) is actually 560 (not 350) microseconds. To get an actual value of 350 usec, HSBDRY would have to be 44 DEC = $2C. This is far too low. Even to read the worst tapes I receive, I need never go below $39.

It was Gene Zumchak, of Niagara Micro Design, Inc., 1700 Niagara St., Buffalo, NY 14207, who pointed out to me the 8 vs 5 usec error in the description of the new (MON 1.1) SYM High Speed Tape Format. Gene writes an excellent column for COMPUTE., and has developed some excellent KIM/SYM products, including an S-100 expansion bus.

TODAY SYM-FORTH, TOMORROW SYM-PASCAL!

I am not given to being overly enthusiastic about computer languages. I have not yet jumped on the Pascal bandwagon, although my academic colleagues wish to banish BASIC totally from the curriculum, and replace it with Pascal. Right now, however, my enthusiasm for FORTH is growing more rapidly with each use. I am willing to predict that the special issue on FORTH of BYTE Magazine, Vol. 5, No. 8, August 1980, will do more to spread the popularity of FORTH than all that has been written on it up to that issue, and that FORTH may even overtake PASCAL in "popularity."

My first impression of FORTH was similar to my first impression of the H-P calculators, both based on an aversion to having to store in my own head-memory a knowledge of what both I and the calculator had put on the stack. The Reverse Polish Notation (RPN) didn't bother me, however, and I soon realized that never did I have to keep track of more than a few items at once.

Why all the enthusiasm? While preparing for this issue, we had the opportunity to learn and use nearly all of the "higher level" languages available for SYM. These and their origins and sources will be reviewed elsewhere in this issue. FORTH has been placed in the public domain and numerous versions are available, as can be realized at once by scanning the ads in the referenced issue of BYTE. Eric Rehnke kindly lent me a review copy of 6502 FORTH, together with a source listing and extensive documentation, and examples of its use. I would, and do recommend his version highly, because of the updating service and extensive documentation he is providing, for any 6502 system other than the SYM.

Fortunately, for SYM owners, Jack Brown (of Brown's Basic Enhancements) has just finalized a version of FORTH, especially tailored to fit the SYM. It is called, naturally, SYM-FORTH! It is tailored to fully utilize all of the features present in the SYM monitor. Jack has provided additional FORTH "Screens", for those SYMmers with KTM-2s, to enhance the editing capability within the basic FORTH package. He has also sent us FORTH Screens to provide interesting graphics on the MTU Visible Memory. His SYM-FORTH manual and the sample Screens provide a self-contained instruction package, bringing you up to a reasonable proficiency level within a few hours.

Incidentally, the 6502 Assembler built into SYM-FORTH occupies less than 1 K of RAM. Up to now, Bob Denison's 2 K Symbolic Assembler, was the most compact Symbolic Assembler available (it is still the only "free-standing" assembler which can be useful on a 4 K SYM).

While I am in love with RAE, and feel very friendly toward BASIC (it being so widely spoken, though despised by the programming elite), if I could have only one higher level language on my SYM, I think I would chose FORTH. Why? Because my main applications are word processing (I bet that I could write a great word processor in FORTH), graphics, and audio (voice and music). FORTH seems, to me, to be a "natural" for these latter two applications.
Jack is now working on a SYM-Pascal! Having observed both the quality of Jack's work, and the speed at which he works, we should be able to have a good Pascal on our SYMs very soon.

MODIFIED EPROM BURNER

Below is the Wiring List for Giersic's Modification of Hobart's EPROM Programmer. See page 5/6 - 27 for more information.

EPROM PIN	AA CONNECTOR	SIGNAL	6522 NAME
1	6	ADDR 7	2PB7
2	H	ADDR 6	2PB6
3	7	ADDR 5	2PB5
4	J	ADDR 4	2PB4
5	8	ADDR 3	2PB3
6	K	ADDR 2	2PB2
7	9	ADDR 1	2PB1
8	L	ADDR 0	2PB0
9	D	DATA 0	2PA0
10	3	DATA 1	2PA1
11	C	DATA 2	2PA2
12	1	GROUND (ALSO FOR 25VOLT SUPPLY)	
13	12	DATA 3	2PA3
14	N	DATA 4	2PA4
15	11	DATA 5	2PA5
16	M	DATA 6	2PA6
17	10	DATA 7	2PA7
18	4	(NOT) CE/PGM	2CA2
19	15	ADDR 10	3PB2
20	5	(NOT) OE	2CB2
21	--	+25 VOLTS	(.1 CAP TO GROUND)
22	T	ADDR 9	3PB1
23	16	ADDR 8	3PB0
24	A	+5 VOLTS	(.47 CAP TO GROUND)

ON POWER SUPPLIES, EXPANSION BUFFERING, AND SYSTEM RELIABILITY

Our working habits are such that we jump from one task to another, wake up in the middle of the night with a great idea, or break for a nap or a thought session in the hot tub. Thus, there is always something "important" in both of our office SYMs that we would hate to lose (we do have backup at all times, however, but it's nicer never to need it, like insurance). Thus our systems are left on (except for the video monitors) at all times, except when we leave town for one reason or another, and we want our SYMs ready at all times, and to have a much better memory than our own.

There was a period when our systems behaved rather flaky at times. (Note to our overseas subscribers, and American, as well: "Flaky" is American slang, but very appropriate here. It means "consisting of flakes," but one of the dictionary definitions of flake is "a small loose mass or bit (sic)"; also, flake is a synonym for chip. Flake derives from the Norwegian word flak, which means disk. Thus flaky is legitimate computerese jargon.) At any rate, loose bits, or "flakes" were giving us problems. We suspected bad chips, poor system design, whatever; anything but the real causes.

The real problems turned out to be not with the digital design, but in the power supply area. We found that, when we were using power supplies at over 75% of their specified current capacity, we began to have loose bit problems, that disappeared when we cooled the power supplies with a fan. Apparently, a power supply operated near its specified limits should not be installed in a confined space without either forced air cooling, or being mounted on a heat conducting surface. Our solution was to split the loads between several power supplies, operating each one at well below rated capacity.

Next, we had long leads from the power supplies to the various subsystems. Because the wire sizes (chosen randomly) and the current drains differed, we measured differences in both ground and +5 levels between, for example, the SYM and the disk controller (mounted with the disks) as high as 0.5 V. When we installed a pair of 12-gauge leads from the power supply to the system, and shorter leads to the subsystems, all of our problems vanished. Our SYMs are now far more reliable than even I am.

And now for possible buffer problems: On our two main systems we do NOT buffer the expansion "bus", and we fill the 4 K "gap" at $1000-$1FFF with the (unbuffered) 4 K Blalock expansion board. On another system at school we are using a buffered motherboard which has sockets to hold 4 K of RAM to fill this gap. This board has two sets of three-state, two-way buffers in series on the data bus to the 4 K of RAM. We noticed peculiarities in accessing data from this RAM block which we could only attribute to problems with the buffers. Rather than replacing these buffer chips, which, incidentally, are quite expensive, and hard to locate, we removed them from their sockets and replaced them with wire jumpers. This cured the problem.

The problem was not in reading data from this block, but whenever a STA instruction to this block was performed elsewhere, the instruction forced a break instruction to follow. While I don't fully understand the problem, it is possible for the sudden current surges caused by a number of fast acting three-state buffers changing state simultaneously to interfere with the control bus signals.

While this problem was local to us, several readers have reported having similar problems with other buffered expansion devices for the SYM. These apparently work well with KIM and AIM, but give problems with some, not all SYMs. One of our readers still has not solved his problems; we have asked him to keep us posted on his progress.

VIDEO MONITORS AND POWER SUPPLIES

We hadn't intended to market SYM hardware; we started to do so mainly as a courtesy to our overseas subscribers, who find it very expensive to purchase multiple money orders in small amounts each. On some items we buy for resale, we are given a reasonable discount for purchases in modest quantities. On others the discount is trivial, or even zero, for small quantities. One example of the latter is a line of power supplies, advertised in the popular computer magazines at $24.95 for the 3 A model. The price to us as an OEM is also $24.95 in lots of up to 24.

We are beginning an OEM activity, assembling systems which include a SYM-1 with 4 K RAM, BAS-1, and RAE-1/2 installed, a KTM-2/80, a Leedex Video Monitor, a Power-One 5 V, 6 A power supply, a Sanyo 1540A cassette recorder, all interconnecting cables, and the SWP-1 and Brown Basic Enhancement Package. The future addition of a Blalock 4 K Memory Expansion Board and a printer will provide at least one class of users (some of my non-technical college professor colleagues) with reasonable word processing and Computer Assisted Instruction capabilities at a far lower cost than available in any other system. They will have to provide their own enclosure.

As a result of this we will have in stock the Leedex Video Monitors, and the Power-One 3 A and 6 A (both with OVP) power supplies. An OEM presumably makes his profit on the "system integration" part of his job, and on the software sales, rather than on the hardware markup. The hardware components are not really purchased for individual resale.

If you wish, however, you may purchase Leedex Monitors and Power-One power supplies from us, for MORE than you would have to pay elsewhere. These items are too heavy to justify overseas airmail costs, so we do not recommend them to overseas subscribers. With the Power One you will be getting parts list and instructions for adding unregulated +11 and +22 and regulated +12 V if you wish. With the Leedex Monitor, you wll be getting the assurance that the Leedex has been personally checked out by us for at least an hour in one of our working systems, prior to our reshipment. Incidentally the Leedex and Power-One items have been selected by Synertek Systems Corporation as components of their own Micro Development Tool (MDT) Systems, and have excellent reliability records.

HUGH CAMPBELL'S APPLE TAPE LOADER

The program on page 5/6-22 is printed as a disassembly of the object code cassette dump sent us by Mr. Campbell. It is easily relocatable any whole number of pages by changing the 13 values of $3E. It adds the following command to MON 1.1, if its starting address is patched to the Unrecognized Command Vector (URCVEC), with .SD 3E00;A66D(cr):

.A xxxx,yyyy(cr)

This command loads a (machine language) Apple tape from xxxx to yyyy. The user is reminded to observe all copyright restrictions concerning making backup, duplicate, or modified, copies of purchased software.

A PROGRAM JUST LOADED WITH GOODIES

The following program is one that probably none of you will want in its entirety, yet we publish it completely because it shows explicitly how to do what so many of you have asked: "How can I write my own power-on routine?"

In addition, the program is loaded with many nice features, literally too numerous to mention, although we'll cover as many as possible.

First, read Mr. Cole's letter, which he sent in the form of initial comments in the source code; this will describe the hardware complement he is using. We are not familiar with the Ithaca Audio V.D.U., although it is a very popular "rider" on the S-100 bus. We will say more elsewhere on the Beta Computer 32 K Dynamic RAM Card.

Since the source code was so lightly commented (we're glad because it does save printing space!), we felt obliged to add a few, near the end.

The program occupies 2 K, from $F000 to $F7FF. There is still unused EPROM from $F5D9 to $F7FB for additions. $F800 to $FFFF is still available for echoing system RAM. It is left as "an exercise for the student" to argue the merits of this approach. Mr. Cole has given up VIA #3 and its associated functions, to be able to assign its 1 K address space to the V.D.U. The VIA chip could be given another address, of course, and its functions preserved, if desired, by adding the calls in this EPROM. But when was the last time you called on VIA #3, except through ACCESS? Note that JSR ACCESS will write over several bytes in the V.D.U. Notice the elegantly human-factored approach in the log-on message and in the clock setting routine.

In summary, whether you are a beginner or a pro, whether you skim through this program or study it in great detail, you have got to learn something new and useful from it.

```
PRINT 0000 0370
 0010  ;STEPHEN .E. COLE M.I.P.R.E. T(eng). C.E.I.
 0020  ;  70,SYDNEY ROAD
 0030  ;     GOSPORT,
 0040  ;       HANTS,
 0050  ;         PO12 1PL
 0060  ;21st AUGUST 1980
 0070  ;Dear Lux:
 0080  ;This program sits at F000 and resets on power up.
 0090  ;The V.D.U is the ITHACA AUDIO S-100 card Type IA-1100 and is
 0100  ;at location $AC00 to $AFFF.
 0110  ;The ram is the beta computers 32k dynamic ram card.
 0120  ;The ram is allocated from $1000 to $7FFF and $9000 to $9FFF;
 0130  ;the latter for patches both rae and basic
 0140  ;I'm very pleased with progress of SYM-PHYSIS.
 0150  ;and feel the articles cover a wide range.  Auther Rechards
 0160  ;has been thinking about AA-Connector Bus but thinks as I do
 0170  ;that it would take up one of the ports to control the tri-state
 0180  ;buffers, which would probably be the best way to control it.
 0190  ;At the moment I am working on a auto trace for BASIC to
 0200  ;be patched like Jack Brown's editor.  Have you any advice
 0210  ;on how to do this?
 0220  ;Disks at this moment are not possible due to cash flow.
 0230  ;I would like to get in contact with any other sym-physis
 0240  ;members in the British Isles to organise a get together.
 0250  ;I work for the British Broadcasting Corporation
 0260  ;as an engineer on operations and maintance at B.B.C. T.V. (SOUTH)
 0270  ;in Southampton.
 0280  ;Also the program below reads the keyboard port to provide a slow
 0290  ;down routine, and the CR on the hex key pad acts as the break key
 0300  ;which is extended outside the box on to the ascii keyboard,
 0310  ;which is a full ascii type with numeric pad and cursor control keys.
 0320  ;My printer is a tractor feed 132 col variable paper 8" to 15" wide.
 0330  ;It is made by a firm in Germany, RENA; have you ever heard of them?
 0340  ;It is a 5 by 7 dot printer and prints all 35 dots at once
 0350  ;at 132 characters a second.
 0360  ;That's all for now; hope to hear from you soon.
 0370  ;
```

```
STEPHEN COLE'S POWER-ON
   UTILITY PROGRAM


0380             .OS
0390             .LS
0400             .BA $F000
0410 ;       VDU CONTROL PATCH
0415 ;      AND CURSOR ADDRESSING
0420 ;
0430 IER         .DE $A00E
0440 IFR         .DE $A00D
0450 PCR         .DE $A00C
0460 ACR         .DE $A00B
0470 T1LL        .DE $A006
0480 T1HC        .DE $A005
0490 COUNTCLOCK  .DE $A612
0500 SECS        .DE $A613
0510 MIN         .DE $A614
0520 HOUR        .DE $A615
0530 DDRA        .DE $A003
0540 ORA         .DE $A00F
0550 CURID       .DE $A600
0560 CHAR1       .DE $A601
0570 LINENO      .DE $A602
0580 KBCHAR      .DE $A603
0590 PRFLAG      .DE $A604
0600 COUNT       .DE $A605
0610 LINBUF      .DE $9F70
0620 TECHO       .DE $A653
0630 INVEC       .DE $A660
0640 OUTVEC      .DE $A663
0650 BEEP        .DE $8972
0660 IER1        .DE $A80E
0670 IFR1        .DE $A80D
0680 PCR1        .DE $A80C
0690 ACR1        .DE $A80B
0700 DDRA1       .DE $A803
0710 DDRB1       .DE $A802
0720 ORA1        .DE $A801
0730 ORB1        .DE $A800
0740 RAM         .DE $A600
0750 MAP         .DE $A606
0760 LA          .DE $A60A
0770 SA          .DE $A60E
0780 BASIC       .DE $C000
0790 BASICWARM   .DE $9000
0800 RAE         .DE $B000
0810 RAEWARM     .DE $B003
0820 STATUSVEC   .DE $896A
0830 UNRECVEC    .DE $81D1
0840 DISSCANVEC  .DE $8906
0850 EXEVEC      .DE $887E
0860 TRACEVEC    .DE $80C0
0870 USERBR      .DE $804A
0880 IRQVEC      .DE $800F
0890 NMI         .DE $809B
0900 OUTBYT      .DE $82FA
0910 INBYTE      .DE $81D9
0920 OUTCHR      .DE $8A47
0930 ;      MEMORY MAP VDU
0940 ;      1 K BLOCK
0950 ;      START ADDRESS $AC00
0960 ;
F000- A2 FF      0970 RESET       LDX #$FF
F002- 9A         0980             TXS
F003- A9 CC      0990             LDA #$CC
                 0995 ;TURN OFF POR AND TAPE UNIT
F005- 8D 0C A0   1000             STA $A00C
F008- A9 04      1010             LDA #4
F00A- 48         1020             PHA
F00B- 28         1030             PLP
F00C- A2 7F      1040             LDX #$7F
F00E- BD CB F4   1050 DEFAULT     LDA DEFAUBLK,X
F011- 9D 00 A6   1060             STA RAM,X
F014- CA         1070             DEX
F015- 10 F7      1080             BPL DEFAULT
F017- A9 FF      1090             LDA #$FF
F019- 8D 02 A8   1100             STA DDRB1
F01C- 8D 03 A0   1110             STA DDRA
F01F- 20 72 89   1120             JSR BEEP
F022- A9 00      1130             LDA #$00
F024- 8D 05 A6   1140             STA COUNT
F027- 20 3D F0   1150             JSR VDU
F02A- 20 3D F0   1160             JSR VDU
F02D- A2 00      1170             LDX #$00
F02F- BD 4B F5   1180 MESS        LDA MESSAGE,X
F032- 20 47 8A   1190             JSR OUTCHR
F035- E8         1200             INX
F036- E0 8E      1210             CPX #$8E
F038- D0 F5      1220             BNE MESS
F03A- 4C 00 80   1230             JMP $8000
F03D- 8D 01 A6   1240 VDU         STA CHAR1
F040- A9 80      1250             LDA #$80
F042- CD 01 A8   1260             CMP ORA1
F045- F0 0E      1270             BEQ GO
F047- A2 20      1280             LDX #$20
F049- A0 FF      1290 DELAY2      LDY #$FF
F04B- 88         1300 DELAY       DEY
F04C- C0 00      1310             CPY #00
F04E- D0 FB      1320             BNE DELAY
F050- CA         1330             DEX
F051- E0 00      1340             CPX #$00
F053- D0 F4      1350             BNE DELAY2
F055- A9 00      1360 GO          LDA #$00
F057- CD 04 A6   1370             CMP PRFLAG
F05A- F0 03      1380             BEQ VDU1
F05C- 20 11 F3   1390             JSR PRINT
F05F- AD 01 A6   1400 VDU1        LDA CHAR1
F062- AE 00 A6   1410             LDX CURID
F065- 38         1420             SEC
F066- C9 20      1430             CMP #$20
F068- B0 03      1440             BCS TB
F06A- 4C 9A F1   1450             JMP CTRL
F06D- C9 7F      1460 TB          CMP #$7F
F06F- D0 05      1470             BNE TA
F071- A9 20      1480             LDA #$20
F073- 4C F8 F2   1490             JMP CLOSE
F076- 20 06 A6   1500 TA          JSR MAP
F079- E8         1510             INX
F07A- E0 40      1520             CPX #$40
F07C- B0 13      1530             BCS NEWLINE
F07E- 8E 00 A6   1540 CUR         STX CURID
F081- 20 0A A6   1550 PA          JSR LA
F084- 20 0E A6   1560             JSR SA
F087- 18         1570             CLC
F088- 69 80      1580             ADC #$80
F08A- 20 06 A6   1590             JSR MAP
F08D- AD 01 A6   1600 FINISH      LDA CHAR1
F090- 60         1610             RTS
F091- AC 02 A6   1620 NEWLINE     LDY LINENO
F094- C0 0F      1630             CPY #$0F
F096- F0 0C      1640             BEQ SCROLL
F098- C8         1650             INY
F099- 8C 02 A6   1660             STY LINENO
F09C- A2 00      1670             LDX #$00
F09E- 20 17 F1   1680             JSR LINE
F0A1- 4C 7E F0   1690             JMP CUR
F0A4- A9 40      1700 SCROLL      LDA #$40
F0A6- 8D 0B A6   1710             STA LA+1
F0A9- A9 AC      1720             LDA #$AC
F0AB- 8D 0C A6   1730             STA LA+2
F0AE- 8D 10 A6   1740             STA SA+2
F0B1- A9 00      1750             LDA #$00
F0B3- 8D 0F A6   1760             STA SA+1
F0B6- A9 C0      1770 AGAIN       LDA #$C0
F0B8- CD 0F A6   1780             CMP SA+1
F0BB- D0 22      1790             BNE NOT
F0BD- A9 AF      1800             LDA #$AF
F0BF- CD 10 A6   1810             CMP SA+2
F0C2- D0 1B      1820             BNE NOT
F0C4- A9 C0      1830             LDA #$C0
F0C6- 20 8A F2   1840             JSR SUB1
F0C9- A9 AF      1850             LDA #$AF
F0CB- 20 94 F2   1860             JSR SUB2
F0CE- A2 00      1870             LDX #$00
F0D0- A9 20      1880             LDA #$20
F0D2- 20 0E A6   1890 YES         JSR SA
F0D5- E8         1900             INX
F0D6- E0 40      1910             CPX #$40
F0D8- D0 F8      1920             BNE YES
F0DA- A2 00      1930             LDX #$00
F0DC- 4C 7E F0   1940             JMP CUR
F0DF- A2 00      1950 NOT         LDX #$00
F0E1- 20 0A A6   1960 MOREVDU     JSR LA
F0E4- 20 0E A6   1970             JSR SA
F0E7- E8         1980             INX
F0E8- E0 40      1990             CPX #$40
F0EA- D0 F5      2000             BNE MOREVDU
F0EC- 18         2010             CLC
F0ED- A9 40      2020             LDA #$40
F0EF- 6D 0B A6   2030             ADC LA+1
F0F2- 8D 0B A6   2040             STA LA+1
F0F5- 90 09      2050             BCC AGAIN1
F0F7- 18         2060             CLC
F0F8- A9 01      2070             LDA #$01
F0FA- 6D 0C A6   2080             ADC LA+2
F0FD- 8D 0C A6   2090             STA LA+2
F100- 18         2100 AGAIN1      CLC
F101- A9 40      2110             LDA #$40
F103- 6D 0F A6   2120             ADC SA+1
F106- 8D 0F A6   2130             STA SA+1
F109- 90 AB      2140             BCC AGAIN
F10B- 18         2150             CLC
F10C- A9 01      2160             LDA #$01
F10E- 6D 10 A6   2170             ADC SA+2
F111- 8D 10 A6   2180             STA SA+2
F114- 4C B6 F0   2190             JMP AGAIN
F117- 18         2200 LINE        CLC
```

```
F118- A9 40     2210          LDA #$40
F11A- 6D 07 A6  2220          ADC MAP+1
F11D- 20 8A F2  2230          JSR SUB1
F120- 90 09     2240          BCC NO
F122- 18        2250          CLC
F123- A9 01     2260          LDA #$01
F125- 6D 08 A6  2270          ADC MAP+2
F128- 20 94 F2  2280          JSR SUB2
F12B- 18        2290 NO       CLC
F12C- 60        2300          RTS
F12D- 38        2310 LU1      SEC
F12E- AD 07 A6  2320          LDA MAP+1
F131- E9 40     2330          SBC #$40
F133- 20 8A F2  2340          JSR SUB1
F136- B0 09     2350          BCS LU2
F138- 38        2360          SEC
F139- AD 08 A6  2370          LDA MAP+2
F13C- E9 01     2380          SBC #$01
F13E- 20 94 F2  2390          JSR SUB2
F141- 60        2400 LU2      RTS
F142- 20 F7 F1  2410 LU       JSR CLCUR
F145- 20 0E A6  2420          JSR SA
F148- 20 2D F1  2430          JSR LU1
F14B- AC 02 A6  2440          LDY LINENO
F14E- 88        2450          DEY
F14F- C0 FF     2460          CPY #$FF
F151- D0 0C     2470          BNE LU3
F153- A0 0F     2480          LDY #$0F
F155- A9 AF     2490          LDA #$AF
F157- 20 94 F2  2500          JSR SUB2
F15A- A9 C0     2510          LDA #$C0
F15C- 20 8A F2  2520          JSR SUB1
F15F- 8C 02 A6  2530 LU3      STY LINENO
F162- 4C 7E F0  2540          JMP CUR
F165- 20 F7 F1  2550 OPEN     JSR CLCUR
F168- 20 0E A6  2560          JSR SA
F16B- 4C E4 F2  2570          JMP OPENSUB
F16E- A2 3F     2580 CLINE    LDX #$3F
F170- A9 20     2590          LDA #$20
F172- CA        2600 CLINE1   DEX
F173- 20 0E A6  2610          JSR SA
F176- D0 FA     2620          BNE CLINE1
F178- 4C 7E F0  2630          JMP CUR
F17B- 20 F7 F1  2640 START    JSR CLCUR
F17E- 20 0E A6  2650          JSR SA
F181- A2 00     2660          LDX #$00
F183- 4C 7E F0  2670          JMP CUR
F186- A9 C3     2680 PR1      LDA #%11000011
F188- 8D 0C A8  2690          STA PCR1
F18B- A0 FF     2700          LDY #$FF
F18D- 88        2710 Y7       DEY
F18E- C0 00     2720          CPY #$00
F190- D0 FB     2730          BNE Y7
F192- A9 E3     2740          LDA #%11100011
F194- 8D 0C A8  2750          STA PCR1
F197- 4C CD F2  2760          JMP PR2
F19A- C9 0D     2770 CTRL     CMP #$0D
F19C- F0 63     2780          BEQ CR
F19E- C9 01     2790          CMP #$01
F1A0- F0 D9     2800          BEQ START
F1A2- C9 11     2810          CMP #$11
F1A4- F0 BF     2820          BEQ OPEN
F1A6- C9 0E     2830          CMP #$0E
```

```
F1A8- F0 C4     2840          BEQ CLINE
F1AA- C9 10     2850          CMP #$10
F1AC- F0 D8     2860          BEQ PR1
F1AE- C9 0B     2870          CMP #$0B
F1B0- D0 03     2880          BNE CARRYON
F1B2- 4C 42 F1  2890          JMP LU
F1B5- C9 00     2900 CARRYON  CMP #$00
F1B7- F0 26     2910          BEQ CLEAR
F1B9- C9 09     2920          CMP #$09
F1BB- F0 71     2930          BEQ TAB
F1BD- C9 08     2940          CMP #$08
F1BF- F0 79     2950          BEQ BS
F1C1- C9 1D     2960          CMP #$1D
F1C3- F0 45     2970          BEQ HOME
F1C5- C9 07     2980          CMP #$07
F1C7- F0 6B     2990          BEQ BELL
F1C9- C9 03     3000 CTRLC    CMP #$03
F1CB- D0 0A     3010          BNE RET
F1CD- A9 CC     3020          LDA #$CC
F1CF- CD 0C A0  3030          CMP $A00C
F1D2- F0 06     3040          BEQ ON
F1D4- 8D 0C A0  3050          STA $A00C
F1D7- 4C 8D F0  3060 RET      JMP FINISH
F1DA- A9 EC     3070 ON       LDA #$EC
F1DC- 8D 0C A0  3080          STA $A00C
                3090 ;
                3100 ;
F1DF- A9 20     3110 CLEAR    LDA #$20
F1E1- A2 00     3120          LDX #$00
F1E3- 9D 00 AC  3130 CL       STA $AC00,X
F1E6- 9D 00 AD  3140          STA $AD00,X
F1E9- 9D 00 AE  3150          STA $AE00,X
F1EC- 9D 00 AF  3160          STA $AF00,X
F1EF- E8        3170          INX
F1F0- E0 00     3180          CPX #$00
F1F2- D0 EF     3190          BNE CL
F1F4- 4C 0A F2  3200          JMP HOME
                3210 ;
                3220 ;
F1F7- AE 00 A6  3230 CLCUR    LDX CURID
F1FA- 20 0A A6  3240          JSR LA
F1FD- 18        3250          CLC
F1FE- 69 80     3260          ADC #$80
F200- 60        3270          RTS
                3280 ;
                3290 ;
                3300 ;
F201- 20 F7 F1  3310 CR       JSR CLCUR
F204- 20 0E A6  3320          JSR SA
F207- 4C 91 F0  3330          JMP NEWLINE
                3340 ;
                3350 ;
F20A- 20 F7 F1  3360 HOME     JSR CLCUR
F20D- 20 06 A6  3370          JSR MAP
F210- A9 00     3380          LDA #$00
F212- AA        3390          TAX
F213- A8        3400          TAY
F214- 8D 00 A6  3410          STA CURID
F217- 8D 02 A6  3420          STA LINENO
F21A- 20 8A F2  3430          JSR SUB1
F21D- A9 AC     3440          LDA #$AC
F21F- 20 94 F2  3450          JSR SUB2
F222- 20 0A A6  3460          JSR LA
```



```
F225- 18        3470          CLC
F226- 69 80     3480          ADC #$80
F228- 20 06 A6  3490          JSR MAP
F22B- 4C 8D F0  3500          JMP FINISH
                3510 ;
                3520 ;
F22E- 20 F7 F1  3530 TAB      JSR CLCUR
F231- 4C 76 F0  3540          JMP TA
                3550 ;
                3560 ;
F234- 20 72 89  3570 BELL     JSR BEEP
F237- 4C 8D F0  3580          JMP FINISH
                3590 ;
                3600 ;
F23A- 20 F7 F1  3610 BS       JSR CLCUR
F23D- 20 0E A6  3620          JSR SA
F240- CA        3630          DEX
F241- E0 FF     3640          CPX #$FF
F243- F0 03     3650          BEQ UL
F245- 4C 7E F0  3660          JMP CUR
F248- A2 3F     3670 UL       LDX #$3F
F24A- AC 02 A6  3680          LDY LINENO
F24D- C0 00     3690          CPY #$00
F24F- F0 22     3700          BEQ BOTTOM
F251- 88        3710          DEY
F252- 8E 00 A6  3720          STX CURID
F255- 8C 02 A6  3730          STY LINENO
F258- AD 07 A6  3740 TOP      LDA MAP+1
F25B- 38        3750          SEC
F25C- E9 40     3760          SBC #$40
F25E- 20 8A F2  3770          JSR SUB1
F261- 90 03     3780          BCC TOP1
F263- 4C 7E F0  3790          JMP CUR
F266- CE 08 A6  3800 TOP1     DEC MAP+2
F269- CE 10 A6  3810          DEC SA+2
F26C- CE 0C A6  3820          DEC LA+2
F26F- 18        3830          CLC
F270- 4C 7E F0  3840          JMP CUR
F273- A2 3F     3850 BOTTOM   LDX #$3F
F275- A0 0F     3860          LDY #$0F
F277- 8C 02 A6  3870          STY LINENO
F27A- 8E 00 A6  3880          STX CURID
F27D- A9 00     3890          LDA #$00
F27F- 20 8A F2  3900          JSR SUB1
F282- A9 AF     3910          LDA #$AF
F284- 20 94 F2  3920          JSR SUB2
F287- 4C 7E F0  3930          JMP CUR
F28A- 8D 07 A6  3940 SUB1     STA MAP+1
F28D- 8D 0F A6  3950          STA SA+1
F290- 8D 0B A6  3960          STA LA+1
F293- 60        3970          RTS
F294- 8D 08 A6  3980 SUB2     STA MAP+2
F297- 8D 10 A6  3990          STA SA+2
F29A- 8D 0C A6  4000          STA LA+2
F29D- 60        4010          RTS
                4020 ;
                4030 ;
                4040 ;  KEY BOARD SCAN ROUTINE
F29E- A9 01     4050 KBSCAN   LDA #%00000001
F2A0- 8D 0B A8  4060          STA ACR1
F2A3- 8D 0C A8  4070          STA PCR1
F2A6- A9 02     4080          LDA #%00000010
F2A8- 2C 0D A8  4090 KBWAIT   BIT IFR1
```

```
F2AB- F0 FB      4100            BEQ KBWAIT
F2AD- AD 01 A8   4110            LDA ORA1
F2B0- 18         4120            CLC
F2B1- 69 80      4130            ADC #$80
F2B3- 8D 03 A6   4140            STA KBCHAR
F2B6- AD 0D A8   4150            LDA IFR1
F2B9- 8D 0D A8   4160            STA IFR1
F2BC- A9 80      4170            LDA #%10000000
F2BE- 2C 53 A6   4180            BIT TECHO
F2C1- F0 06      4190            BEQ NOECHO
F2C3- AD 03 A6   4200            LDA KBCHAR
F2C6- 20 3D F0   4210            JSR VDU
F2C9- AD 03 A6   4220 NOECHO     LDA KBCHAR
F2CC- 60         4230            RTS
F2CD- A9 00      4240 PR2        LDA #$00
F2CF- CD 04 A6   4250            CMP PRFLAG
F2D2- F0 07      4260            BEQ PR3
F2D4- 8D 04 A6   4270            STA PRFLAG
F2D7- AD 01 A6   4280            LDA CHAR1
F2DA- 60         4290            RTS
F2DB- A9 01      4300 PR3        LDA #$01
F2DD- 8D 04 A6   4310            STA PRFLAG
F2E0- AD 01 A6   4320            LDA CHAR1
F2E3- 60         4330            RTS
F2E4- A8         4340 OPENSUB    TAY
F2E5- E8         4350            INX
F2E6- 20 0A A6   4360            JSR LA
F2E9- 8D 01 A6   4370            STA CHAR1
F2EC- 98         4380            TYA
F2ED- 20 0E A6   4390            JSR SA
F2F0- AD 01 A6   4400            LDA CHAR1
F2F3- E0 40      4410            CPX #$40
F2F5- D0 ED      4420            BNE OPENSUB
F2F7- 60         4430            RTS
F2F8- A2 3F      4440 CLOSE      LDX #$3F
F2FA- A8         4450 CLOSESUB   TAY
F2FB- CA         4460            DEX
F2FC- 20 0A A6   4470            JSR LA
F2FF- 8D 01 A6   4480            STA CHAR1
F302- 98         4490            TYA
F303- 20 0E A6   4500            JSR SA
F306- AD 01 A6   4510            LDA CHAR1
F309- EC 00 A6   4520            CPX CURID
F30C- D0 EC      4530            BNE CLOSESUB
F30E- 4C 81 F0   4540            JMP PA
                 4550 ;
F311- AD 01 A6   4560 PRINT      LDA CHAR1
F314- C9 0A      4570            CMP #$0A
F316- F0 17      4580            BEQ LFP
F318- C9 0D      4590            CMP #$0D
F31A- F0 4C      4600            BEQ PRINTOUT
F31C- C9 0C      4610            CMP #$0C
F31E- F0 24      4620            BEQ FFP
F320- C9 0B      4630            CMP #$0B
F322- F0 32      4640            BEQ VTP
F324- AE 05 A6   4650            LDX COUNT
F327- 9D 70 9F   4660            STA LINBUF,X
F32A- E8         4670            INX
F32B- 8E 05 A6   4680            STX COUNT
F32E- 60         4690            RTS
F32F- A9 04      4700 LFP        LDA #%00000100
F331- 8D 0F A0   4710            STA ORA
F334- A0 00      4720            LDY #$00
```



```
F336- C8         4730 Y2         INY
F337- C0 80      4740            CPY #$80
F339- D0 FB      4750            BNE Y2
F33B- A9 00      4760            LDA #$00
F33D- 8D 0F A0   4770            STA ORA
F340- 8D 05 A6   4780            STA COUNT
F343- 60         4790            RTS
F344- A9 08      4800 FFP        LDA #%00001000
F346- 8D 0F A0   4810            STA ORA
F349- A0 00      4820            LDY #$00
F34B- C8         4830 Y3         INY
F34C- C0 80      4840            CPY #$80
F34E- D0 FB      4850            BNE Y3
F350- A9 00      4860            LDA #$00
F352- 8D 0F A0   4870            STA ORA
F355- 60         4880            RTS
F356- A9 10      4890 VTP        LDA #%00010000
F358- 8D 0F A0   4900            STA ORA
F35B- A0 00      4910            LDY #$00
F35D- C8         4920 Y1         INY
F35E- C0 80      4930            CPY #$80
F360- D0 FB      4940            BNE Y1
F362- A9 00      4950            LDA #$00
F364- 8D 0F A0   4960            STA ORA
F367- 60         4970            RTS
F368- A2 00      4980 PRINTOUT   LDX #$00
F36A- AD 0D A8   4990 STROBE     LDA IFR1
F36D- 8D 0D A8   5000            STA IFR1
F370- A9 C7      5010            LDA #%11000111
F372- 8D 0C A8   5020            STA PCR1
F375- A0 FF      5030            LDY #$FF
F377- 88         5040 Y6         DEY
F378- C0 00      5050            CPY #$00
F37A- D0 FB      5060            BNE Y6
F37C- A9 E7      5070            LDA #%11100111
F37E- 8D 0C A8   5080            STA PCR1
F381- 18         5090            CLC
F382- AD 0D A8   5100 TEST       LDA IFR1
F385- 6A         5110            ROR A
F386- 90 E2      5120            BCC STROBE
F388- BD 70 9F   5130            LDA LINBUF,X
F38B- 8D 00 A8   5140            STA ORB1
F38E- EC 05 A6   5150 CO1        CPX COUNT
F391- F0 04      5160            BEQ CRP1
F393- E8         5170            INX
F394- 4C 6A F3   5180            JMP STROBE
F397- A9 02      5190 CRP1       LDA #%00000010
F399- 8D 0F A0   5200            STA ORA
F39C- C8         5210 Y          INY
F39D- C0 80      5220            CPY #$80
F39F- D0 FB      5230            BNE Y
F3A1- A9 20      5240            LDA #$20
F3A3- 8D 00 A8   5250            STA ORB1
F3A6- 4C 2F F3   5260            JMP LFP
F3A9- A9 14      5270 CLOCKSTART LDA #$14
F3AB- 8D 12 A6   5280            STA COUNTCLOCK
F3AE- 8D 0B A0   5290            STA ACR
F3B1- A9 C0      5300            LDA #$C0
F3B3- 8D 0E A0   5310            STA IER
F3B6- A9 F0      5320            LDA #$F0
F3B8- 8D 06 A0   5330            STA T1LL
F3BB- A9 C2      5340            LDA #$C2
F3BD- 8D 05 A0   5350            STA T1HC
```

```
F3C0- 60         5360            RTS
F3C1- 08         5370 CLOCKIRQ   PHP
F3C2- 48         5380            PHA
F3C3- F8         5390            SED
F3C4- A9 F0      5400            LDA #$F0
F3C6- 8D 06 A0   5410            STA T1LL
F3C9- A9 C2      5420            LDA #$C2
F3CB- 8D 05 A0   5430            STA T1HC
F3CE- CE 12 A6   5440            DEC COUNTCLOCK
F3D1- D0 3B      5450            BNE EXIT
F3D3- A9 14      5460            LDA #$14
F3D5- 8D 12 A6   5470            STA COUNTCLOCK
F3D8- A9 01      5480            LDA #$01
F3DA- 18         5490            CLC
F3DB- 6D 13 A6   5500            ADC SECS
F3DE- 8D 13 A6   5510            STA SECS
F3E1- C9 60      5520            CMP #$60
F3E3- D0 29      5530            BNE EXIT
F3E5- A9 00      5540            LDA #$00
F3E7- 8D 13 A6   5550            STA SECS
F3EA- A9 01      5560            LDA #$01
F3EC- 18         5570            CLC
F3ED- 6D 14 A6   5580            ADC MIN
F3F0- 8D 14 A6   5590            STA MIN
F3F3- C9 60      5600            CMP #$60
F3F5- D0 17      5610            BNE EXIT
F3F7- A9 00      5620            LDA #$00
F3F9- 8D 14 A6   5630            STA MIN
F3FC- A9 01      5640            LDA #$01
F3FE- 18         5650            CLC
F3FF- 6D 15 A6   5660            ADC HOUR
F402- 8D 15 A6   5670            STA HOUR
F405- C9 24      5680            CMP #$24
F407- D0 05      5690            BNE EXIT
F409- A9 00      5700            LDA #$00
F40B- 8D 15 A6   5710            STA HOUR
F40E- 68         5720 EXIT       PLA
F40F- 28         5730            PLP
F410- 40         5740            RTI
F411- A2 00      5750 CLOCKSET   LDX #$00
F413- BD 82 F4   5760 HR         LDA HRMESS,X
F416- 20 47 8A   5770            JSR OUTCHR
F419- E8         5780            INX
F41A- E0 0D      5790            CPX #13
F41C- D0 F5      5800            BNE HR
F41E- 20 D9 81   5810            JSR INBYTE
F421- 8D 15 A6   5820            STA HOUR
F424- A2 00      5830            LDX #$00
F426- BD 8F F4   5840 MI         LDA MINMESS,X
F429- 20 47 8A   5850            JSR OUTCHR
F42C- E8         5860            INX
F42D- E0 0F      5870            CPX #15
F42F- D0 F5      5880            BNE MI
F431- 20 D9 81   5890            JSR INBYTE
F434- 8D 14 A6   5900            STA MIN
F437- A2 00      5910            LDX #$00
F439- BD 9E F4   5920 SE         LDA SECSMESS,X
F43C- 20 47 8A   5930            JSR OUTCHR
F43F- E8         5940            INX
F440- E0 0F      5950            CPX #15
F442- D0 F5      5960            BNE SE
F444- 20 D9 81   5970            JSR INBYTE
F447- 8D 13 A6   5980            STA SECS
```

```
F44A- A9 00     5990            LDA #$00
F44C- 8D 12 A6  6000            STA COUNTCLOCK
F44F- AA        6010            TAX
F450- BD AD F4  6020 GOCLOCK    LDA GOMESS,X
F453- 20 47 8A  6030            JSR OUTCHR
F456- E8        6040            INX
F457- E0 1E     6050            CPX #30
F459- D0 F5     6060            BNE GOCLOCK
F45B- 20 D9 81  6070 BACK1      JSR INBYTE
F45E- C9 0D     6080            CMP #$0D
F460- D0 F9     6090            BNE BACK1
F462- 4C A9 F3  6100            JMP CLOCKSTART
F465- AD 15 A6  6110 CLOCKCHECK LDA HOUR
F468- 20 FA 82  6120            JSR OUTBYT
F46B- A9 2E     6130            LDA #'.
F46D- 20 47 8A  6140            JSR OUTCHR
F470- AD 14 A6  6150            LDA MIN
F473- 20 FA 82  6160            JSR OUTBYT
F476- A9 20     6170            LDA #$20
F478- 20 47 8A  6180            JSR OUTCHR
F47B- AD 13 A6  6190            LDA SECS
F47E- 20 FA 82  6200            JSR OUTBYT
F481- 60        6210            RTS
F482- 0D 45 4E  6220 HRMESS     .BY $0D 'ENTER HOURS '
F485- 54 45 52
F488- 20 48 4F
F48B- 55 52 53
F48E- 20
F48F- 0D 45 4E  6230 MINMESS    .BY $0D 'ENTER MINUTES '
F492- 54 45 52
F495- 20 4D 49
F498- 4E 55 54
F49B- 45 53 20
F49E- 0D 45 4E  6240 SECSMESS   .BY $0D 'ENTER SECONDS '
F4A1- 54 45 52
F4A4- 20 53 45
F4A7- 43 4F 4E
F4AA- 44 53 20
F4AD- 0D 54 4F  6250 GOMESS     .BY $0D 'TO START TYPE CR AT TIME SET '
F4B0- 20 53 54
F4B3- 41 52 54
F4B6- 20 54 59
F4B9- 50 45 20
F4BC- 43 52 20
F4BF- 41 54 20
F4C2- 54 49 4D
F4C5- 45 20 53
F4C8- 45 54 20
F4CB- 00 00 00  6260 DEFAUBLK   .BY 00 00 00 00 00 00 $9D $00 $AC
F4CE- 00 00 00
F4D1- 9D 00 AC
F4D4- 60 BD 00  6270            .BY $60 $BD $00 $AC $60
F4D7- AC 60
F4D9- 9D 00 AC  6280            .BY $9D $00 $AC $60 00 00 00 00 00
F4DC- 60 00 00
F4DF- 00 00 00
F4E2- 00 00 00  6290            .BY 00 00  00  00 00 00 00 00 00
F4E5- 00 00 00
F4E8- 00 00 00
                6300 ;COMMENTS ADDED BY LUX FROM THIS POINT ON:
                6310 ;COMPARE THE ENTRIES FROM THIS POINT ON WITH THE
                6320 ;MON 1.1 DEFAULT TABLE ON PAGES 4-21 AND 4-22 OF
                6330 ;THE SYM-1 REFERENCE MANUAL, BEGINNING AT $A620.
                6340 ;THESE ARE HIS "J" COMMANDS FROM MON (SEE BELOW FOR DEFS):
```



```
F4EB- 00 C0     6350            .SE BASIC
F4ED- 00 90     6360            .SE BASICWARM
F4EF- 00 B0     6370            .SE RAE
F4F1- 03 B0     6380            .SE RAEWARM
F4F3- 11 F4     6390            .SI CLOCKSET
F4F5- 65 F4     6400            .SI CLOCKCHECK
                6410 ;HE CAN RESET WITH .J6 AT THE KEYBOARD
                6420 ;IF IN MON, BUT NOT AFTER A SYSTEM CRASH.
F4F7- 00 F0     6430            .SI RESET
                6440 ;HE SETS HIS CASSETTE DEFAULTS TO 2800 BAUD!!!!!
F4F9- 00 02 04  6450            .BY $00 $02 $04 $2C $23 $00
F4FC- 2C 23 00
F4FF- 00 1A 00  6460            .BY $00 $1A $00 $00 $00 $00
F502- 00 00 00
F505- 00 00 2D  6470            .BY $00 $00 $2D $00 $00 $00 $00
F508- 00 00 00
F50B- 00
F50C- 00 6D 6E  6480            .BY $00 $6D $6E $86 $06 $00
F50F- 86 06 00
F512- 00 00 00  6490            .BY $00 $00 $00 $00 $00 $00 $00
F515- 00 00 00
F518- 00
F519- 00 00 01  6500            .BY $00 $00 $01 $4C $00 $80
F51C- 4C 00 80
F51F- B0 00 00  6510            .BY $B0 $00 $00 $00 $10 $00 $F0
F522- 00 10 00
F525- F0
F526- FF 00 00  6520            .BY $FF $00 $00 $00 $00 $4C
F529- 00 00 4C
                6530 ;NOTICE THE USE OF THE .SI AND .SE PSEUDO OPCODES HERE:
F52C- 9E F2     6540            .SI KBSCAN
F52E- 4C        6550            .BY $4C
F52F- 3D F0     6560            .SI VDU
F531- 4C        6570            .BY $4C
F532- 6A 89     6580            .SE STATUSVEC
F534- 4C        6590            .BY $4C
F535- D1 81     6600            .SE UNRECVEC
F537- 4C        6610            .BY $4C
F538- D1 81     6620            .SE UNRECVEC
F53A- 4C        6630            .BY $4C
F53B- 06 89     6640            .SE DISSCANVEC
F53D- 7E 88     6650            .SE EXEVEC
F53F- C0 80     6660            .SE TRACEVEC
F541- 4A 80     6670            .SE USERBR
F543- C1 F3     6680            .SI CLOCKIRQ
                6690 ;HE USES MON 1.1 IRQVEC AND NMI(VEC), BUT HIS OWN RESET.
F545- 9B 80     6700            .SE NMI
F547- 00 F0     6710            .SI RESET
F549- 0F 80     6720            .SE IRQVEC
                6730 ;HERE IS HIS AUTO LOG-ON MESSAGE WHICH PRESENTS A
                6740 ;"MENU" OF THE JUMP TABLE:
F54B- 0D 53 2E  6750 MESSAGE    .BY $0D 'S.E.COLE SYS-1  AUG 1980' $0D
F54E- 45 2E 43
F551- 4F 4C 45
F554- 20 53 59
F557- 53 2D 31
F55A- 20 20 41
F55D- 55 47 20
F560- 31 39 38
F563- 30 0D
F565- 20 4A 55  6760            .BY ' JUMP TABLE' $0D
F568- 4D 50 20
F56B- 54 41 42
F56E- 4C 45 0D
```

```
F571- 30 20 42  6770            .BY '0 BASIC' $0D '1 WARM START BASIC' $0D
F574- 41 53 49
F577- 43 0D 31
F57A- 20 57 41
F57D- 52 4D 20
F580- 53 54 41
F583- 52 54 20
F586- 42 41 53
F589- 49 43 0D
F58C- 32 20 52  6780            .BY '2 RAE' $0D '3 WARM START RAE' $0D
F58F- 41 45 0D
F592- 33 20 57
F595- 41 52 4D
F598- 20 53 54
F59B- 41 52 54
F59E- 20 52 41
F5A1- 45 0D
F5A3- 34 20 53  6790            .BY '4 SET CLOCK' $0D '5 PRINT CLOCK' $0D
F5A6- 45 54 20
F5A9- 43 4C 4F
F5AC- 43 4B 0D
F5AF- 35 20 50
F5B2- 52 49 4E
F5B5- 54 20 43
F5B8- 4C 4F 43
F5BB- 4B 0D
F5BD- 36 20 52  6800            .BY '6 RESET SYS' $0D
F5C0- 45 53 45
F5C3- 54 20 53
F5C6- 59 53 0D
F5C9- 37 20 47  6810            .BY '7 GOTO HEX 0200' $0D
F5CC- 4F 54 4F
F5CF- 20 48 45
F5D2- 58 20 30
F5D5- 32 30 30
F5D8- 0D
                6820 ;HERE'S WHERE THE ACTUAL RESET VECTOR COMES FROM:
                6830            .BA $F7FC
F7FC- 00 F0     6840            .SI RESET
                6850            .EN
```

READING APPLE TAPES

Published below are portions of two letters from Hugh Campbell, one of our many Australian subscribers.

We concur with Mr. Campbell's remark about Microchess at the current state-of-the-art. But, for its time, Peter Jennings's 1.1 K Microchess for the unaugmented KIM was a real tour de force. It seemed a near miracle to be able to have that neat little assembly of electronic parts tell us, on its little display, where it wished us to move its pieces. We converted our Microchess from KIM to SYM many months back, and it was just as much fun as on the KIM, in spite of our increased sophistication (both SYM and us!).

For awhile, we considered making arrangements to market our SYM version, until we found that one was already available from the 6502 Program Exchange. Very regretably, this version makes use of a terminal, so that, even though it will actually draw the board for you, on request, the old charm is gone. We decided against recommending that version when we found that, if we were in check, and made no attempt to protect our king, Microchess also ignored the check, and turned its attention elsewhere!

Sargon is well known to us by reputation, and one of our colleagues reports that Fastgammon provides a worthy opponent. So that you can replay and "post-mortem" a game, Fastgammon gives you the option to repeat the previous sequence of dice-throws with each restart. We publish below a copy of Mr. Campbells Apple Loader Program, so that you can purchase some of the Apple (machine language only) programs on cassette for your use on SYM.

We have not been able to try the Apple Loader Program, ourselves, as we are still waiting for delivery of the Apple Tape "Slide Show", which we ordered. We ordered this tape because of Dave P. Kemp's (developer of the SP-1 "Speak & Spell" interface for the SYM) article "Slide Show for the SYM", in MICRO, Issue Number 25, June 1980. In this article Mr. Kemp shows how to, not only read machine language dumps of the Apple graphic displays into the SYM, but how to convert them from Apple display format into MTU's Visible Memory format in the process.

We will be sending copies of this issue to Hayden Books and to Quality Software, suggesting that they consider marketing SYM cassette versions of their programs. Quality Software, according to Mr. Campbell, does not attempt to "protect" their software. Hayden does "protect" their Apple software, but as you can see from Mr. Campbell's letter, such protection can be "broken".

19 Brushy Creek Road
Lenah Valley
Tasmania 7008
Australia

17 August 1980

Dear Lux

Thank you for your note about my SYM-1 modifications to Apple II games.

Yes - you can load Apple machine code cassettes into a SYM by using a modification of the Apple cassette load from their monitor. By a stroke of luck or something, the speed of the tapes is almost right. I have to slow down my Superscope recorder by about a quarter of a turn of the speed control. It was one of those "I wonder if's..." that worked. It even worked with the MON - 1.0 that I was using when I found how to read Apple tapes.

Loading into SYM-1 helps to break security codes. When Mon 1.0 or 1.1 meets an Apple monitor address it returns to monitor without destroying the code, as the Apple does on RESET.

I have the code for loading Apple tapes, together with working modifications of:

Hayden's Sargon and Sargon II
Personal's Microchess and Checker King
Quality's Fastgammon

All are modified to run on a SYM-1 and a 40 character KTM-2, with primitive graphics. You really need a separate board for chess; the other two are OK on the VDU.

I have not relocated any of the programs, since I have only the code that I disassembled myself to work from; and plenty of RAM besides. The programs use the followins RAM.

SARGON :- 800-2CFF
SARGON II :- E00-2FFF
FASTGAMMON :- 800-20D0
CHECKER KING :- 3D0-1FFF

I have not listed Microchess. It is not worth trying when the Sargons are available.

FASTGAMMON is a good one to try first, because that does not need any code-breaking. It just needs the translation from Apple I/O to SYM I/O. None of my modifications is very tidy or elegant. I simply changed what I needed and left the rest of the Apple code as it was. I have added something to the original in most cases. For instance, in the Sargon that I have sent to you, the ten move opening 'books' are not in the Apple version.

Would you like to help me to do something with the Apple loader and games? Distance and time are both a bit much for negotiating from here in Australia.

I have sent you the first SARGON with instructions so that you can see what I am talking about. I actually sent that off to Hayden about a year ago, but heard nothing from them. I suspect that they could not read my tape made with MON 1.0. Things should be a good bit better now

You can see what I think of SWP-1. What do you think of Sargon?

Yours faithfully

Dear Lux

Thank you for your note dated 7 September. I look forward to hearing whether Hayden Book will cooperate.

The other side of this tape contains my loader for Apple tapes. As you will see, it is only 196 bytes, and so it is loaded almost before you know it. It is located at 3E00-3EC4, but it relocates easily by changing the six or so 3E bytes. To use the program in its present location, enter SD 3E00;A66D and return. Then enter A xxxx,yyyy and return. xxxx and yyyy are the start and end addresses on the Apple instructions for loading. I have to fiddle a bit with speed and tone. Most tapes load with both controls set about where they are on the diagram with the tape. Some don't need a change at all. Sometimes there is a glitch right at the beginning of a tape which causes an ER FF. You can overcome this by seeing where the code begins on the tape, and saving the carriage return till that point. A good load returns to monitor in the usual way.

As for translating the Apple programs to SYM-1. Well, that is easy sometimes, and sometimes not. With an Apple memory map such as the one in MICRO recently, you can find the Apple I/O instructions, and fiddle SYM ones into their place. The remaining instructions with Apple monitor addresses (C0xx and Fxxx) you either NOP or use in some way. Several days later, you can play an Apple game on a SYM.

Anti-copying codes are a problem. It depends how much trouble has been taken. The good thing is that, since SYM does not recognise Apple monitor instructions, it does not delete or change code, and so all the bootstraps and so on are still on your tape. If you are lucky, the tape has some recognisable ASCII on it somewhere. You can generally count on 4C xx xx at 0800. But Sargon II and Microchess 2, for instance, EOR with a hidden code before beginning the game. Sargon II also collects four addresses from odd places ranging from page zero to 5FFF, tells you the wrong code for the EOR, and deletes part of the bootstrap anyway. It is all part of the challenge.

Also on the tape is a SYM version of Quality Software's Fastgammon. It is the only Apple tape I have found that makes no attempt at all to confuse the user. QS are also in California.

Regards

Hugh Campbell

HUGH CAMPBELL'S APPLE TAPE LOADER PROGRAM: See preceding letters, and short article on page 5/6-9 for information on using this program.

```
3E00-  AD 57 A6    LDA   A657
3E03-  C9 41       CMP   #41
3E05-  D0 07       BNE   3E0E
3E07-  AD 49 A6    LDA   A649
3E0A-  C9 02       CMP   #02
3E0C-  F0 03       BEQ   3E11
3E0E-  4C D1 81    JMP   81D1
3E11-  AD 4D A6    LDA   A64D
3E14-  85 3D       STA   3D
3E16-  AD 4C A6    LDA   A64C
3E19-  85 3C       STA   3C
3E1B-  AD 4B A6    LDA   A64B
3E1E-  85 3F       STA   3F
3E20-  AD 4A A6    LDA   A64A
3E23-  85 3E       STA   3E
3E25-  4C 7B 3E    JMP   3E7B
3E28-  A5 3C       LDA   3C
3E2A-  C5 3E       CMP   3E
3E2C-  A5 3D       LDA   3D
3E2E-  E5 3F       SBC   3F
3E30-  E6 3C       INC   3C
3E32-  D0 02       BNE   3E36
3E34-  E6 3D       INC   3D
3E36-  60          RTS
3E37-  A0 4B       LDY   #4B
3E39-  20 49 3E    JSR   3E49
3E3C-  D0 F9       BNE   3E37
3E3E-  69 FE       ADC   #FE
3E40-  B0 F5       BCS   3E37
3E42-  A0 21       LDY   #21
3E44-  20 49 3E    JSR   3E49
3E47-  C8          INY
3E48-  C8          INY
3E49-  88          DEY
3E4A-  D0 FD       BNE   3E49
3E4C-  90 05       BCC   3E53
3E4E-  A0 32       LDY   #32
3E50-  88          DEY
3E51-  D0 FD       BNE   3E50
3E53-  AC FF FF    LDY   FFFF
3E56-  A0 2C       LDY   #2C
3E58-  CA          DEX
3E59-  60          RTS
3E5A-  A2 08       LDX   #08
3E5C-  48          PHA
3E5D-  20 69 3E    JSR   3E69
3E60-  68          PLA
3E61-  2A          ROL   A
3E62-  A0 3A       LDY   #3A
3E64-  CA          DEX
3E65-  D0 F5       BNE   3E5C
3E67-  60          RTS
3E68-  00          BRK
3E69-  20 6C 3E    JSR   3E6C
3E6C-  88          DEY
3E6D-  AD 00 A0    LDA   A000
3E70-  29 40       AND   #40
3E72-  C5 2F       CMP   2F
3E74-  F0 F6       BEQ   3E6C
3E76-  85 2F       STA   2F
3E78-  C0 80       CPY   #80
3E7A-  60          RTS
3E7B-  AD 02 A0    LDA   A002
3E7E-  29 BF       AND   #BF
3E80-  8D 02 A0    STA   A002
3E83-  A9 00       LDA   #00
3E85-  8D 0B A0    STA   A00B
3E88-  20 69 3E    JSR   3E69
3E8B-  A9 16       LDA   #16
3E8D-  20 37 3E    JSR   3E37
3E90-  85 2E       STA   2E
3E92-  20 69 3E    JSR   3E69
3E95-  A0 21       LDY   #21
3E97-  20 6C 3E    JSR   3E6C
3E9A-  B0 F9       BCS   3E95
3E9C-  20 6C 3E    JSR   3E6C
3E9F-  A0 37       LDY   #37
3EA1-  20 5A 3E    JSR   3E5A
3EA4-  81 3C       STA   (3C,X)
3EA6-  45 2E       EOR   2E
3EA8-  85 2E       STA   2E
3EAA-  20 28 3E    JSR   3E28
3EAD-  A0 31       LDY   #31
3EAF-  90 F0       BCC   3EA1
3EB1-  20 5A 3E    JSR   3E5A
3EB4-  C5 2E       CMP   2E
3EB6-  F0 0A       BEQ   3EC2
3EB8-  A9 07       LDA   #07
3EBA-  20 47 8A    JSR   8A47
3EBD-  A9 FF       LDA   #FF
3EBF-  4C D1 81    JMP   81D1
3EC2-  4C 03 80    JMP   8003
```

```
3E00 AD 57 A6 C9 41 D0 07 AD,38
3E08 49 A6 C9 02 F0 03 4C D1,02
3E10 81 AD 4D A6 85 3D AD 4C,DE
3E18 A6 85 3C AD 4B A6 85 3F,A7
3E20 AD 4A A6 85 3E 4C 7B 3E,0C
3E28 A5 3C C5 3E A5 3D E5 3F,F6
3E30 E6 3C D0 02 E6 3D 60 A0,0D
3E38 4B 20 49 3E D0 F9 69 FE,2F
3E40 B0 F5 A0 21 20 49 3E C8,04
3E48 C8 88 D0 FD 90 05 A0 32,88
3E50 88 D0 FD AC FF FF A0 2C,53
3E58 CA 60 A2 08 48 20 69 3E,36
3E60 68 2A A0 3A CA D0 F5 60,91
3E68 00 20 6C 3E 88 AD 00 A0,30
3E70 29 40 C5 2F F0 F6 85 2F,27
3E78 C0 80 60 AD 02 A0 29 BF,FE
3E80 8D 02 A0 A9 00 8D 0B A0,0E
3E88 20 69 3E A9 16 20 37 3E,29
3E90 85 2E 20 69 3E A0 21 20,84
3E98 6C 3E B0 F9 20 6C 3E A0,41
3EA0 37 20 5A 3E 81 3C 45 2E,60
3EA8 85 2E 20 28 3E A0 31 90,FA
3EB0 F0 20 5A 3E C5 2E F0 0A,8F
3EB8 A9 07 20 47 8A A9 FF 4C,24
3EC0 D1 81 4C 03 80,45
 5845
```

H. R. LUXENBERG 8/29/80
EDITOR: SYM-PHYSIS ,SYM-1 USERS' GROUP NEWSLETTER
P. O. BOX 315 CHICO,CA 95927

LUX,
JUST A SHORT CASSETTE TO EXPRESS MY APPRECIATION OF SYMPHYSIS.
YOUR MATERIAL HAS GREATLY HASTENED MY EVOLUTION OUT OF SIMPLESYMDOM.
I AM COMPOSING THIS FILE USING MY SYM, A SURPLUS XEROX KEYBOARD FROM CALIFORNIA DIGITAL, AND PAIA ELECTRONICS' TVT 6-5/8 (CHEAP VIDEO).
I REMEMBER SEEING IN ISSUE 3 OF SYM-PHYSIS A PROMISE TO DISCUSS 'CHEAP' VIDEO TERMINALS, WHICH I WAITED ANXIOUSLY FOR, BUT DID NOT SEE.
I HAVE JUST ADAPTED CHEAP VIDEO TO THE SYM-1 AND THOUGHT OTHERS MIGHT BE INTERSTED IN MY EXPERIENCE.

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

CHEAP VIDEO FOR THE SYM

JOHN MATTOX
102 NW 27 TERRACE
GAINESVILLE FL 32607
(904) 378-6768

PAIA ELECTRONICS SELLS A BARE BONES VIDEO DRIVER KIT (TVT6-5/8) FOR $43, WHICH INCLUDES DON LANCASTER'S CHEAP VIDEO COOKBOOK. TVT CONSISTS OF SEVEN INTEGRATED CIRCUITS; AN UPPER CASE CHARACTER GENERATOR IS SUPPLIED WITH THE KIT. THE IMPLEMENTATION OF AN UPPER-LOWER CASE GENERATOR IS DISCUSSED.

TVT IS DESIGNED FOR THE KIM. THE DESIGN REQUIRES THE DISPLAY MEMORY DATA BUS TO BE BUFFERED (TO PROVIDE DIRECT ACCESS BY TVT WHILE THE HOST MICROPROCESSOR EXECUTES A SUBROUTINE WHICH INCREMENTS THE ADDRESS BUS). THIS MEANS THAT ADDITIONAL OFF-BOARD MEMORY IS REQUIRED WITH TRI-STATE DRIVERS IN BOTH DIRECTIONS FOR THE SYM.

TVT USES THE 4 HIGHEST ADDRESS LINES AS DISPLAY INSTRUCTIONS. THIS MEANS THAT IF ONE USES THE INSTRUCTION DECODER PROM SUPPLIED WITH THE KIT, ADDRESS SPACE $6000-$DFFF MUST BE RESERVED FOR TVT. THIS IS CLEARLY UNACCEPTABLE. FOR $3 PAIA ELECTRONICS WILL TRADE THE PROM FOR ONE PROGRAMMED TO USE MEMORY SPACE $2000 THROUGH $9FFF. THE MONITOR ENABLE JUMPER MUST BE REPLACED WITH OFF BOARD LOGIC USING A 6522 VIA TO SWITCH BETWEEN ENABLING THE MONITOR AND ENABLING THE TVT. ADDITIONAL EXPANSIONS WILL ALSO NEED TO BE DESELECTABLE.

IN ADDITION TO USING 50% OF THE ADDRESS FIELD, TVT REQUIRES UP TO 95% OF THE CPU TIME (THERE AIN'T NO FREE LUNCH). ALSO, FOR DISPLAYS WITH MORE THAN 40 COLUMNS, THE HORIZONTAL FREQUENCY OF THE CRT MONITOR MUST BE REDUCED.

MY CURRENT IMPLEMENTATION IS TO DISPLAY WHILE WAITING FOR KEYBOARD INPUT (WHICH IS TIME OTHERWISE WASTED). WITH A 1 K DISPLAY MEMORY, I AM USING A 64X16 DISPLAY. I AM BUILDING AN 8 K DISPLAY MEMORY WHICH WILL ALLOW A 80X24 DISPLAY AND 256X256 BLACK & WHITE GRAPHICS.

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

I INCLUDE HARDCOPY BECAUSE I AM NOT CERTAIN OF MY RECORDER. YOU MAY PUBLISH WHAT YOU WISH OF IT. IF I HAD IT TO DO OVER, I WOULD HAVE INSTEAD PURCHASED KTM-2/80.

SINCERELY,

JOHN

John: Thanks for the valuable writeup. Many of our readers will also find it helpful. Your recorder writes fine, reads easily! - Lux.

Dr. H. R. Luxenberg
Editor SYM-PHYSIS
P. O. Box 315
Chico, California 95927

August 14, 1980

Dear Dr. Luxenberg:

I have been using Jack Brown's 'Super TCP' and 'Ultrarenumber' along with a modified version of Tom Gettys' 'Merge/Delete' and find them to be a marvelous enhancement to my SYM. I've put them together with a graphics package for MTU's K-1008 board, a tape verify and tape directory segment (after the one of Jack Gierysic), and linkage to a printer, all on a pair of 2716 EPROMs (stacked, of course) with the result that I have a very nice overall operating system for the SYM. My primary use of the system right now is to write programs for lecture demonstration in Physics classes using the TV graphics.

In using Ultrarenumber I discovered that if it finds reference to a nonexistent line number following GOTO, GOSUB, etc. it gives it the number 65535 which is an illegal line number in SYM-1 BASIC. The This means that you cannot run a renumbered program until all the illegal references are individually corrected. The largest acceptable line number in SYM-1 BASIC is 63999 which is F9FF in hex. Therefore I changed lines 3220-3230 in Ultra-renumber as follows:

```
3220   STA *FACTO-1
3222   LDA #$F9
3230   STA *FACTO
```

Now when the above circumstance occurs, it recomputes the references to the nonexistent lines as 63999. References to this line number can then be trapped out by using a statement such as:

```
63999 GOTO____
```

which takes the program to some appropriate point. I have not found a situation in which I could not get a satisfactory renumbering job.

I certainly enjoy SYM-PHYSIS and find it generally very helpful. I think it is clearly the best publication going for "Symmers". I have found the work of Jack Brown and Tom Gettys very instructive, especially in understanding how some of the BASIC routines work. Keep up the good work.

Sincerely,

James G. Pengra
Department of Physics
Whitman College
Walla Walla, Washington 99362

SYM AS A PERSONAL COMPUTER & THE EDUCATIONAL/ACADEMIC MARKET

(continued from page 5/6-2)

Synertek appreciates, and are exploring the possibilities of OEMing a package built around this nucleus, with attractive discounts to individual faculty and students to encourage personal systems as well. We consider working engineers just breaking-in to the microcomputer field as being in the 'student' category, for discount purposes, too.

We have purveyed several such systems locally, and would be interested in hearing your suggestions and comments about this class of systems.

HOW TO ADD MORE VIAS EASILY

The SYM-1 dedicates a full 4 K of address space ($A000-$AFFF) to the 6532 and the three on-board 6522s, 1 K to each. The 6532 needs 128 addresses for its RAM, and 32 addresses for its I/O-TIMERS, but the 6522s each need only 16 addresses of their assigned 1 K blocks.

When the problem arose of adding the HDE Disk Controller, which has its own on-board 6522, with full address decoding, we assigned it the addresses $A88X (16 bytes only), and temporarily removed the (user supplied) VIA #2 in U28, so that there would be no address conflicts. We could have broken the 1 K block at $A800-$ABFF in half, and assigned $AA00-$ABFF to any added VIAs, but decided to use only page $A8 for all I/O, since the upper half of this one page alone would allow the addition of eight more VIAs (with full decoding).

The first two purchasers of the HDE SYM-FODS came up with two very different hardware implementations of the simple logic to keep VIA #2 out of the upper halves of the four pages to which it has access. These are both presented elsewhere in this issue. It is interesting to note that, not only did each chose a different logical equation to implement, each selected a different technology. One chose the IC approach, adding a chip; the other chose DTL (Diode Transistor Logic) because it (the transistor) was already there.

As of now we have not added the additional VIAs we thought we would want. One reason is that we were able to "recover" the use of PB 6 on VIA #1. The second reason is that we would like to put all of our "extra" ports at the Auxilliary Application Connector, and not load our unbuffered Address and Data Busses at the Expansion Connector down any further, even though we have had no problems as yet. In fact, we just installed the Color-Mate color graphics system on the (unbuffered) expansion bus (addresses $9000-$9FFF) with no problems.

Since we fully intend to install the AY-3-8910 Programmable Sound Generator (one of these days!), we plan to make use of its "free" built-in pair of 8-bit I/O ports, which can be used independently of the sound generation function, to handle the control functions we wish. One of the ports could serve as the origin of a two-way data bus to a number of other VIAs, and the second port could provide the chip select and resister select functions. Obviously we have not thought this through in detail as yet, because we are still not yet certain of our requirements.

MORE FROM JOHN GIERYIC ABOUT JACK BUILT PROGRAMS

We told Jack that we didn't think there would be a great market for games for the SYM-1/KTM-2 system. This turned out to be true; most of our readers seem to want mostly "utilities." The demand for SWP-1, and BBE-1 (our new name for the Brown Basic Enhancements Package, given because the Purchasing Office people who place orders with us feel uncomfortable unless the product has an ordering number as well as a name), has been high; the Jack Built Programs have not shaken the earth with fast movement.

Well, Jack has generated some PLOT Utilities for BASIC, which work with the (built-in) Gowan Double Density Plot and Tris Patch machine language programs. These additions to the Jack-Built Programs line may be ordered through the Users' Group. The best way to describe these programs is to reprint the instruction sheet; this we do below. He also provided some bus exterminators and a well human-engineered BASIC EPROM Programmmer Program. We will therefore turn the next few pages over to Jack. But first, a few more editorial comments (to keep our editorial license current):

Jack is an extremely good BASIC programmer, and is using RAE to prepare his manuscripts for publication. We're glad about the latter, since it is so easy to "SWP" his manuscripts into camera-ready copy. His PLOT programs mix BASIC with MLC (Machine Language Code), which is, of course, generated by RAE. We hope we can persuade Jack to redo one of his PLOT programs in SYM-FORTH, and report back to us a (not necessarily objective) comparision of the two approaches.

We are curious to hear what this Jack will have to say on FORTH versus BASIC, because our other Jack (Jack Brown, that is), also very well versed in BASIC, has sent us some graphics programs (for the MTU Visible Memory), and the classical "Towers of Hanoi" as a beautiful graphics demonstration on the KTM-2/80, written in FORTH, with its own built-in Tris Patch, and MLC portions, compiled with FORTH'S built-in ASSEMBLER Vocabulary. We find, as only "occasional" programmers (meaning only when we have to, because we couldn't con someone else into doing it for us!), that the FORTH program is actually easier to follow than a BASIC one, in spite of many more years of BASIC experience.

from JACK BUILT PROGRAMS

These three programs all require 8K of RAM, a KTM-2/80 and BAS-1. Enter BASIC with 6400 bytes free. Each program consists of a BASIC program part and a machine language part (19F0 thru 1FFF) containing Bill Gowans' plot package from SYM-PHYSIS Issue #3 plus the tris functions. Two loads are required in BASIC. The first loads the machine language portion and the second loads the BASIC program. Once this is done the user can go to one of the other two programs by loading only the BASIC part of that other program. The machine language portion is identical for all three programs.

```
**********************
*                    *
* DUAL Y-AXIS PLOT   *
*                    *
**********************
```

This package produces a plot of two equations of the form Y = F(X) over the same user-specified x range, while using a different user-specified y range for each equation. This concept is similar to the dual trace oscilloscope. It allows the user to view two widely differing plots superimposed on the same "piece of graph paper". Two equations are already in memory. Enterins the followins values in response to the program prompts will sive a "feel" for the program: -1, 1, 0, 720, 0, 720.

```
**********************
*                    *
* 4 QUADRANT PLOT    *
*                    *
**********************
```

Up to four independent equations of the form Y = F(X) are plotted on a four quadrant grid which utilizes the entire monitor screen. The user specifies the number of equations to plot, the maximum positive x value and the maximum positive y value. The program plots the equation(s) on a 160(h) by 48(v) "dot" grid. Enter the followins values in response to the program prompts: 2, 1, 720. This will produce four complete cycles of a sine wave and another 4 cycles of a cosine wave.

```
**************
*            *
* POLAR PLOT *
*            *
**************
```

Up to four independent equations of the form R=F(TH) are plotted on a polar grid which uses the entire monitor screen. The scaling is adjusted such that the plot of R = 5 does appear as a circle on the monitor. The program prompts the user for the maximum R value, the range of THETA (minimum and maximum), and the increment value. For example, if the range of THETA is 10 to 30 degrees with an increment of 4, then the program would plot a point every 4 degrees between 10 and 30. A minimum THETA of 0 cannot be entered due to error checking by the program. Try entering the following values to the program prompts: 4, 6, 1, 360, 4. This will show what the program can do. Now change line 20 to read R=TH and enter RUN. Enter the following values to the prompts: 1, 27, 1, 1440, 6. This should result in an impressive spiral.

JACK BUILT PROGRAM BUGS

Two of my programs have minor buss. I want to thank Don Full and Cap Teisen for bringing these to my attention. Lux already has the fixes incorporated into his source. To the many who have already purchased the two programs, check your source. If the following fixes are missing, then add them. Do not write over your original tape. Copy the corrected source to a new tape. If the new tape checks out OK, then, and only then, should you rewrite your original tape.

BAR GRAPH

Symptoms: Very short bars (less than 8 scan lines tall) are actually 8 scan lines taller than they should be. With this fix these short bars will be the correct length. If the bar is too short to appear as even a single scan line, that bar on the graph will be blank. That's the way it should be.

Fix: Insert the following line: 526 IFC=0THENQ=1:GOTO532

PLOT

Symptoms: Under certain conditions, a single point does not appear to "follow" the plot. Instead it appears at a point somewhere below its real position and the plot has a "hole" (missing plot point) at this position. This hole will appear at the very top line of the graph. The bug is a very common one; a relational operator in an IF statement did not include the "equal to" case.

Fix: Insert an "=" in line 153 so it reads as follows:

```
153 IFY>=YLTHENB=39:GOTO160
```

MORE ON THE EPROMMER

Dear Lux and all SYMmers:

I read with extreme interest Joe Hobart's article "An EPROM Programmer for the SYM-1" in SYM-PHYSIS Issue #4. His idea was elegant and, best of all, cheap! As a complement to his simple hardware design, I have written a very comprehensive software package which gives the user a great deal of power when programming the 2516 or 2716 EPROM.

I reassigned the ports on the 6522's such that NO modifications are necessary to the SYM. Simply wire the programmer per the directions in the enclosed article and you're ready to RUN. This allows you to use the four buffered bits on Port AC00 for other applications on the "AA BUS".

For you lucky users who own a KTM-2 or KTM-2/80, my program will utilize cursor positioning to make the man-machine interface a bit more pleasing. For you other terminal owners, all data and displays will be left justified on your screen.

An additional feature of this program is the ability to turn off all power to the EPROM when the EPROM is not being accessed. This enables the user to change EPROMs without turning off the SYM. While the option list is being displayed and the program is waiting for the user's selection, all port bits are logic 0. If the user installs a double pole single throw switch on the +5 volts to EPROM pin 24, and the +25 volts to EPROM pin 21, the user can turn this switch off and remove/replace/insert EPROMs. This switch need be turned on only for options 1,2,3,4 and 7.

Wouldn't it be nice if this switch could be automated? Well, I've provided such a signal on the AA connector pin S (3PB3). This pin will go high when the EPROM is accessed (options 1, 2, 3, 4, and 7). This signal can be used as a means to control a relay which would replace the manual dpst switch.

There is yet another method of implementing this automatic control of the +5 and +25 voltages. In my case, I used two SIGMA relays (part no. 191TE1A1-5S). These are dual-in-line packaged reed relays (14 pin DIP) with an internal suppression diode. Each relay has a single pole single throw switch. One relay is used to switch the +5 volts and the other to switch the +25 volts. The signal from pin S (3PB3) drives an inverter (7404), which, in turn, is used to sink current on the +5 volt control relay coil. Another signal is provided on the Y pin (3PB4) to drive another inverter, which is used to sink current on the +25 volt control relay coil.

This use of one of the buffered bits on the port at AC00 requires a change on the SYM. Buffer B4 (lower lefthand of the SYM board) should have point A jumpered to point 3 (refer to Figure 4-5a in your SYM Reference Manual) and point B jumpered to point 18. Also, the .47 capacitor to EPROM pin 24 is repositioned so it is always connected from ground to +5 volts. My schematic for this method is summed up below.

SYMcerely,

Jack Gieryic

From JACK BUILT PROGRAMS

```
********************************
*                              *
* 2516/2716 EPROM PROGRAMMER *
*                              *
********************************
```

Hardware requirements:

SYM-1
SYM BASIC
7K RAM
Terminal with at least 16 lines and
at least 32 characters per line

This program presents the user with a list of 8 options from which to choose. When the selected option is completed the program will ask the user if he/she wants to go back to the option display. If the response is YES, then the option list will again be presented. All addresses and limits can be entered in either decimal or hex. Hex numbers are simply preceded by the letter H, e.g., H2F5.

OPTIONS

- 1 - PROGRAM EPROM
- 2 - COMPARE EPROM TO MEMORY
- 3 - VERIFY EPROM IS CLEARED
- 4 - DISPLAY EPROM MEMORY
- 5 - DISPLAY MEMORY
- 6 - ENTER MEMORY DATA
- 7 - READ EPROM TO MEMORY
- 8 - MEMORY MOVE

OPTION 1 - This option is used to program the EPROM. It prompts the user for the EPROM starting address, the number of bytes to program, and the data starting address in memory. The EPROM starting address plus the number of bytes cannot extend beyond the 2K of the EPROM. If it does, the program will again prompt the user for all of the data. This option allows the user to program any number of bytes, anywhere within the EPROM, from any area of the SYM memory, without disturbing the remaining locations in the EPROM. After the programing is completed this option will verify the data just written, and any errors will be displayed in the following format:

ERROR "EPROM address" "EPROM data" "memory address" "memory data"

If the requested parameters were entered in decimal then the error data is displayed in decimal. If the parameters were entered in hex then the error data will be displayed in hex.

OPTION 2 - This option will compare any portion of the EPROM to memory. The option requests 3 parameters, as does option 1. Any errors are displayed as in option 1.

OPTION 3 - This option will verify that any part of the EPROM contains hex FF. It will request a starting address and the number of bytes to check. Any locations not containing FF will be printed, along with the data found in those locations. It is wise to verify if the EPROM is cleared, prior to programming, as this could save a lot of time.

OPTION 4 - This option is used to view the contents of any number of contiguous bytes in the EPROM without bringing the data into memory. The option asks for the starting address and the number of bytes to display. The data is displayed in hex regardless of the format used to enter the address and byte count (decimal or hex).

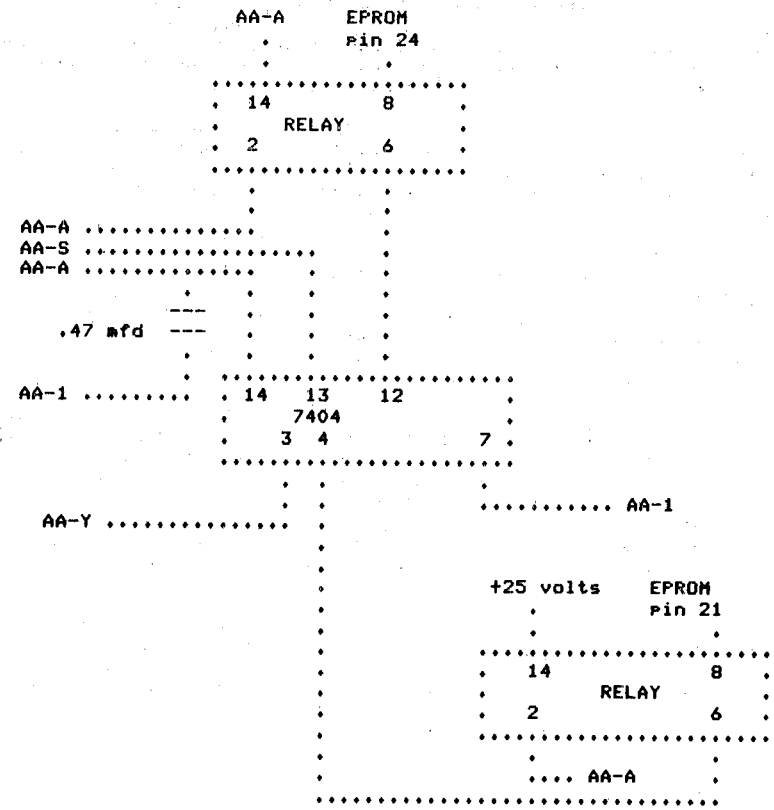
OPTION 5 - This option is used to view the buffer before burning it into the EPROM. Any area within memory may be viewed, and there is no limit on the number of bytes to display. The display format is identical to option 4.

OPTION 6 - This option will allow the user to hand-construct a buffer in memory for transfer to the EPROM via option 1. This option will also enable the user to change data before burning it into an EPROM. This option requests a starting address (decimal or hex), and then displays the address (decimal or hex), followed by the data (hex only). A dash appears after the data and the user now enters new hex data to change that location, or the same data as is displayed, to skip over that location. The next address and its data are now displayed, and so on, until the user enters the letters END, instead of data. This terminates the option.

OPTION 7 - This option allows the user to transfer any or all of an EPROM's data into the SYM's memory. This allows the user to change the data via option 6, if desired, and then burn a new EPROM, via option 1. The option requests the EPROM's starting address, the number of bytes to read in, and the beginning of the buffer in the SYM's memory. Make sure your memory buffer is large enough!

OPTION 8 - This option will allow the user to move any size block of memory into another area of memory. With this option, the user can move data from an EPROM already being used in his/her system into RAM, alter it via option 6, and then burn a new EPROM. All this without touching the EPROM being used in the system!

Note: Options 6, 7 and 8 will not allow the user to use page 0, page 1, or the memory occupied by the program. This prevents self-destruction. This program resides in the first 5K of memory.



Above is the schematic for Jack's Automatic EPROM Programmer Switcher. His revised wiring list for the EPROM Programmer is on page 5/6 -7. Please note that the BASIC listings for the EPROM PROGRAMMER and for the PLOT programs are NOT published in this issue! Only the instructions for these five programs are given here, so that you can decide whether or not they will be useful enough to purchase in cassette form from JACK BUILT PROGRAMS.

PRODUCT RECOMMENDATIONS

We have tested a number of new products since Issue No. 4, and can recommend all but one of them (that one will not be listed here). Some of these we will be distributing. Please note that we don't recommend because we are distributing. On the other hand, if we can recommend a product, we may try to get distribution rights, if possible.

The ColorMate

The easiest way to describe this newest product from MicroMate, P. O. Box 50111, Indianapolis, IN 46256, is to reprint the following extract from their brochure:

> ColorMate brings the flexibility of color video display to KIM, SYM and AIM microcomputer systems. Designed around the Motorola 6847 video display generator, ColorMate offers nine modes of operation, ranging from alphanumeric to full graphic. A 12-bit word format in the alphanumeric/semigraphic modes provides capability to mix alphanumeric with semigraphic characters. Two pages of video memory are implemented in the alphanumeric/semigraphic modes, providing added flexibility in many applications. Selection of the page to be displayed requires only a write to the ColorMate control register. The alphanumeric display format is 16 rows by 32 columns. Semigraphic modes provide colorful applications on a 48 by 64 element grid. Full graphics resolution ranges to 192 by 128.
>
> The ColorMate is memory mapped, and requires a 4K byte memory space. Decoding to 4K boundaries is provided onboard. It interfaces to the Expansion (E) connector of the KIM, SYM and AIM microcomputers. The single 5 volt supply is provided via the interface. The PC board is a standard 4½ by 6½ inch format, with plated through holes and gold contacts. The ColorMate may be configured to drive a standard TV (ch. 3 or 4).
>
> In addition to a complete description of the various modes of operation,the ColorMate User Manual includes example software for a color video terminal driver (ViewMate), a Fourier synthesis program, and various graphics plot subroutines.

ColorMate is available directly from MicroMate, either as the PC board and manual alone, for $50, or fully assembled and tested, but without the necessary IC's, at $95. In either case, you will have to hunt up the IC's somewhere; you can most easily obtain the complete set of IC's from MicroMate, for $125 additional. Add 2% to all prices, for shipping and handling. NOTE: The ColorMate will NOT work with PAL or SECAM color TV!!!!

While the maximum full graphics mode resolution of 128 H x 192 V is less than that of the Atari (320 H x 192 V), the Apple II (280 H x 192 V), or the TRS-80 Color Computer (256 H x 192 V), the color capability is still quite impressive, and can be very useful and effective. The ColorMate requires 4 K of address space (ours is at $9000-$9FFF), with 3 K of video mapped 2114 RAM on board (the remaining 1 K is not memory mapped, but is used as the "address" for the control register). We use a B/W monitor switched between the KTM-2/80 and the MTU Visible Memory, with a Radio Shack Coax Switch, for most purposes. The color graphics coexist simultaneously on a color TV set. The ColorMate has its own on-board RF Modulator, so that connection to the TV is through a 75 ohm to 300 ohm switching adaptor to the VHF antenna terminals on the set. When the TV is not available for SYM use, SYM has an extra 3 K of RAM with which to play!

Several subscribers have already written, or phoned, to let us know they are using, and are pleased with, ColorMate. Dick Turpin has provided plenty of software in the manual with which to get started, and will be supporting a User Group. The source code is in RAE format, and we are making arrangements with Dick to provide ColorMate Graphics software in cassette form. The board uses what we are calling the "Reverse KIM" pinout. See the next recommendation for installation suggestions.

The Quest Expansion Board

As many of you have discovered, not all 44-contact SYM/KIM/AIM expansion boards are compatible. With the exception of contacts E-16, -17, -18, -19, -20, and -X, 38 out of 44 of the SYM, KIM, and AIM expansion "Pin-out" assignments are identical. One group of expansion boards (MTU, for example) has, except for E-2, and E-3 (in addition to those listed above), the identical contact assignments. For this class of boards, any "mother-board" must have what we are calling the "KIM-1" Bus. All connector contacts, except for the ones listed above, are wired in parallel with the contacts on a connector into which the SYM is "plugged."

MOS Technology, producers of the KIM-1, devised, and marketed, for a very short time only, the KIM-4 Motherboard. All of the lettered, and two of the numbered, contacts were shifted by one position with respect to the KIM-1 Bus, and some signals were dropped, and others added. The Computerist, Hudson Digital Electronics, and RNB Enterprises, among others, support the KIM-4 Bus structure, with varying degrees of fidelity.

Other expansion boards are available, most notably the ColorMate and the Beta Computer Devices' 32 K Memory Board, which use the "Reverse KIM" Bus. These are designed to plug directly into a connector which "extends" and replaces the edge contacts on the SYM board. Alternately, a connector may be "reversed" and its solder-eyes or solder-tails soldered directly to the edge contacts on the expansion board, and the board/connector combination mounted directly onto the edge contacts of the SYM.

What's a fella to do, if, like me, he wants to add an HDE Disk Controller (KIM-4 Bus), ColorMate (Reverse KIM Bus), and an MTU Visible Memory Board (KIM-1 Bus) to the same SYM? We found that the 44-contact Expansion Board which was developed by Quest Electronics, P. O. Box 4430, Santa Clara, CA 95054, for use with their Super Elf (an excellent, RCA 1802 based, single board computer) System, when fitted with an extra, reverse mounted, 44-contact connector, fills the bill admirably. It can be fitted to the SYM straight out, or at a right angle. If fitted at a right angle, the three solder tail sockets can be mounted on either side of the board, to project either forward or backward. There are three rows of installation holes for each connector, so that the connectors can be installed in either of two positions, to provide either direct or reversed KIM-1 Bus.

There is plenty of room between the connectors to cut the appropriate traces, and the unused rows of installation holes make it easy to insert the necessary jumpers to convert one or more of the connectors to accept a KIM-4 board, e. g., the HDE Disk Controller. These boards can also be used on the Applications Connector, to mount a pair of DACs for stereo music or vector graphics, for example, and also on the Auxilliary Applications Connector, for all kinds of goodies. If you make your own application boards, you wire the contacts to match the application connector. If you use commercially available boards, such as MTU's DACs, you must, of course, cut and jumper the traces on the expansion boards to match.

While we are on the subject of busses, we should mention the SYM compatible "S-44" bus, and series of memory and other cards available from Kathryn Atwood Enterprises, P.O. Box 5203, Orange, CA 92667. This bus is ingeniously arranged so that no damage will be done if the boards are inserted backwards, and, in fact, many of the boards will work either way! This has to be the ultimate in "idiot-proofing" equipment.

The Beta Computer Devices 32 K Dynamic RAM Board

The SYM is expandable to 4 K of RAM on-board, and the Blalock Memory Board provides an easy expansion to 8 K. As many of our readers have discovered, with BASIC and/or RAE in ROM, an 8 K RAM SYM makes a very respectable system. Additional utilities are very easily put in EPROM.

The ColorMate provides its own 3 K of RAM, if you want color graphics. If you want to add a disk system, you will need at least 8 K more of RAM, but once you have the disk system up, you will want more and more RAM, for all of the things the disk system will let you do.

One of our SYMs has been expanded to 32 K, by adding MTU's 16 K dynamic RAM, and MTU's 8 K Visible Memory (also dynamic). The 8 K Visible Memory provides the bonus of high resolution black and white graphics. The disadvantages include the necessity for adding additional voltages to the power supply, unregulated +8 V and +16 V, and the need to provide a card case to hold the added cards. The graphics capability more than makes up for these two minor disadvantages.

For a long time, we have been looking for a 32 K expansion board which would permit us to build a really portable, two piece system, which could travel with us, under an airplane seat, if necessary. One piece would be our 9" Sanyo Monitor, in a protective case; the other, about the size and shape of an attache case, would hold the SYM, the KTM, the power supply, and the cassette recorder.

Stephen Cole, whose letter appears elswhere in this issue, is using the Beta Computer Devices Model 6502DM Memory Board. At the same time we received Stephen's letter, another reader wrote in, asking our opinion of the board. Not wishing to recommend an item we had not yet tried, we ordered one from Beta, at 1230 West Collins, Orange, CA 92668. Here are our comments:

The board requires only a single +5 V (at 0.8 A max) supply. There is no heat generating regulator on-board, and the board generates its own regulated -5 V and +12 V. The board is 4" x 6" and has edge fingers designed for the S-44 Bus (see elsewhere for some comments on this). The board has holes for mounting a (supplied) right angled 44 contact connector to fit the Reverse KIM bus. This means that the connector fits directly on the Expansion Connector edge fingers of the SYM. The connector can be fitted to the memory board in two ways, so the the board may be mounted either extended out and away from the SYM, or "folded" neatly and compactly below it. The latter mounting style is what we will use for our portable system. We will probably bring out a second connector, so that a disk controller card can be added for home use. We might then bring along our disk system as a third unit, if we travel by private auto.

The board is dynamic RAM only. It contains no ROM sockets, additional VIAs, or EPROM programming system. We find that, for the SYM, these "omissions" are of no consequence. The SYM is "loaded" with its own on-board VIAs, and the Hobart EPROM Programmer works off the Auxilliary Application Connector. All 32 K of RAM may be freely assigned in 4 K blocks, anywhere in empty memory space, with absolutely no constraints. This means that, if you have the 4 K RAM sockets on board filled, you will have 36 K of RAM available. The extra 4 K can be used to fill the gap at $9000-$9FFF; this would be a good location for all of the MON/BAS utilities you are using. The only free memory now available is the 2 K block at $F000-$F7FF, or, if you inhibit the System RAM "echo" at $F800-$FFF, the whole top 4 K block. One of the four on-board ROM sockets can be freed to hold one or two (piggy-backed) EPROMS. If you inhibit the echo, the top six bytes in EPROM must contain "fixed" IRQ and NMI vectors. You will have to give up the flexibility provided by IRQVEC and NMIVEC in System RAM.

To summarize, we like the Beta Board, and recommend it highly, if you need no additional graphics capability beyond that provided by the KTM-2/80 (160 x 48). For our fully portable system we are willing to accept this restriction.

HUEY II

Many readers have asked about adding a floating-point arithmetic package to SYM, which can be called from assembly language programs, or "patched" to the various "tiny" languages, e. g., tiny-c, tiny basic, tiny pilot, etc.

One approach, if you have BAS-1 resident, is to call on its subroutines, as required. We have not done this, but refer you to an article by R. M. Mottola, "MEAN 14: A Pseudo-Machine Floating Point Processor for the Apple II", in MICRO No. 28, September, 1980. The name "MEAN 14" is a parody of the name "SWEET 16", for the pseudo-machine 16-bit processor package in the Apple II. MEAN 14 can be adapted to SYM by replacing the Applesoft subroutine call addresses by the corresponding BAS-1 addresses. We have not done this, mostly for lack of time, but also because the real problem is to provide a free-standing floating point package for use without BAS-1.

The real answer is provided by Don Rindsberg's "HUEY II", available from the 6502 Program Exchange (address elsewhere in this issue). We have long been fond of the original HUEY, but did not recommend it earlier because we knew that the new version was in the works, and because the old edition of the manual was incomplete, requiring the user to locate, somewhere, a copy of the December 1977 Kilobaud! The new manual is now a self-contained document.

Huey II may be used alone to make the SYM-1 act like a Reverse Polish calculator, or its subroutines may be called from other high or low level languages. We recommend it highly, even if you don't ever use it. We suggest that you study its structure and its numerical algorithms, particularly if you are new to programming and computational methods.

FOCAL, FAST FOCAL, XPLO, AND TEC 65

FOCAL (FORmula CALculator) is a close relative of BASIC, with a number of elegant enhancements, originally developed by DEC for the PDP systems. TEC 65 is a 6502 version of the Text Editor, TECO. XPLO is similar to Pascal and C, all three being descendants from ALGOL.

All are available, in SYM cassette format (we tested them all), from the 6502 Program Exchange, 2920 Moana, Reno, NV 89509. Please write them (our contact there is Dave Marsh), for information on memory requirements, prices, and additional supporting software availability. Source code listings are available, unfortunately not in RAE format (hence not on SYM readable cassette), for all of these languages, so that you can easily adapt them to your own system configurations.

If you are deeper into software than into hardware, all ot these are worth owning for study and comparision purposes. Each has its own unique set of good and bad features. If hardware is your major area of interest, you should know that all languages are "equivalent" in the sense that any language can be made to do any job, although some may be more convenient or/and faster than others, in certain applications.

Our own feeling about languages, based purely on personal experience, and, of course, personal bias, is that, in the microprocessor environment, FORTH might have the edge in speed and convenience, particularly for control applications. Of course others, with equal or greater experience, have their own personal biases. The only honest recommendation that can be made here, or anywhere, is to get and try them all, and come to your own conclusion.

tiny-c

C is a 'Pascal-like' language developed by the Bell Telephone Laboratories. It seems to have a reasonably wide usage, but nowhere near the mass popularity of Pascal. tiny-C is an integer only version, available from tiny-c associates, P.O. Box 269, Holmdel, NJ, 07733, as "A Home Computing Software System".

We tried tiny-c, and liked it, and found the manual and documentation to be of outstandingly high, truly professional quality. The SYM-1 readable cassette version presents a few problems in reading and organization to produce a "Load and Go" cassette. We have reported the results of our tests, and our suggested fixes, back to tiny-c associates (tca), and assume that the fixes will be incorporated into future versions. We hope to make arrangements to market the SYM version for tca, but it is too early to report further on this at present.

There is neither the time nor the space to describe tiny-c here, and we are not reporting on prices, because of a rise in overseas mailing costs early next year, and because the price of a new text on tiny-c has not yet been announced. We suggest you write tca directly for any additional information.

MISCELLANEA

The reason that the back of each sheet of SYM-PHYSIS may seem to be "upside-down" with respect to the front side is so that the issues may be punched along the LONG edge for insertion into a three hole binder. If the binder is then turned 90 degrees clockwise, the pages are then all right-side up. This is not our own idea; we borrowed it from the KIM-1/6502 USER NOTES, because we liked their format. The issues are not pre-punched for you because we are doing our best to keep all costs down. Besides, about twenty percent of our subscribers are in "metric" countries, and we are not sure of the standards for the metric three hole punch.

We decided against carrying a "Beginners' Column" in each issue, because SYM-PHYSIS is sold only by the volume, not by the issue. By the time a new reader has finished Issue No. 2, and read some of the recommended books and articles he is no longer a beginner. A number of new subscribers originally called us nearly every day, when they first set up their SYMs, to ask very elementary questions, to which they could have found the answers in the Reference Manual (admittedly, that's not always easy!). Their calls became less frequent, and their questions became much more sophisticated and challenging, as the weeks went by. Now they call or write only to report on some new or exciting application or expansion.

We will be teaching a weekend course at the University of California at Davis (about 20 miles west of Sacramento) on December 5-7, 1980, on "Microprocessor Fundamentals." The $475 fee includes a "free" SYM. If you already have all of the SYMs you need, the fee is reduced accordingly. Please write or phone Garrett Jones, University Extension, University of California, Davis, CA 95616, (916) 752-2177. We plan to offer one or more similar courses at Cal State Chico, next spring. Please write Prof. O. S. Madrigal, Department of Computer Science, California State University, Chico, CA 95929, for additional information.

John R. Robertson, of Portland, Oregon, advised us of a company in Hong Kong that makes enclosures for the SYM and the KTM-2s. We have written them concerning possible import arrangements. We will report further in the next issue. If you can't wait till then, contact us after mid-December.

Tom Evans, WA6WTA, 20501 Hatteras St., Woodland Hills, CA 91367, would like to hear from any hams doing RTTY with the SYM.

Norrito Giorgio, 1200 Levin Ave., Mt. View, CA 94040, is interfacing an Exatron Stringy Floppy to his SYM.

Bruce Thompson, Cornell University, has been using four SYMs in remote stations to sense the geomagnetic and geoelectric fields and record them on cassettes. These run unattended for three days on 6 V car batteries. The timing and synchronization are critical so he runs them with an external oscillator which is buried in the ground to reduce diurnal rate variations. Since the sample periods are between .001 and 10 secords, he has T1 feeding T2 via PB7 connected to PB6 externally in order to get the range necessary. He says that the SYMs operate with ne'er a problem despite the 100 degree temperatures and 100% humidity.

Shahrakh Ghaffari, Chemistry Department, Oregon State University, Corvallis, OR 97331, sent us a note describing how to transfer BASIC programs from KIM to SYM. Those readers with both KIM BASIC and SYM BASIC may wish to contact him for the technique.

John Blalock asked us to mention that the prices for the W7AAY 4 K RAM Board, and the W7AAY RAE-1/2 ROM Board are now $8.00 plus a 15 cent (why isn't there a cent sign on an ASCII keyboard?) self-addressed, stamped envelope, and $16.00 postpaid, in the USA. Please order directly from him, P. O. Box 39356, Phoenix, AZ 85069. As a courtesy to our foreign subscribers, and for the convenience of those ordering other items from us at the same time, we will keep a small stock of both on hand at all times. Overseas, please add postage costs for one ounce, and three ounces, respectively.

PILOT is an extremely easy-to-learn CAI (Computer Assisted Instruction) Language, which younsters can learn to use nearly as soon as they are able to read and write. It has been placed in the public domain, and a number of "Tiny" Pilot versions are available for microcomputer systems use. Recent issues of MICRO contain a number of articles on 6502 Tiny Pilots, beginning with a SYM version by Nick Vrtis. Nick sent us a cassette version of a SYM Tiny Pilot, together with a new instruction manual, greatly enhanced and improved over the version originally published. We hope that Nick decides to market his new version. It will be of great value to those with young SYMmers in their households.

SUPER-SYMS?

At least one of our readers is working along the following lines:

1. Relocate SUPERMON to $F000-$FFFF
2. Reassign the I/O, etc., to $E000-$EFFF
3. Obtain from Carl Moser a relocated RAE at $C000-$DFFF (same as BASIC)
4. Use a second Beta Memory Board with only 16 K of RAM to fill $A000-$DFFF
5. Use a Blalock Memory Board to free 4 K from the first Beta Memory Board to be assigned to $8000-$8FFF
6. Call BASIC, RAE, FORTH, Pascal, and all other higher level languages in from disk as needed

This approach will provide a 56 K RAM/4 K ROM SYM system. To provide more RAM, memory bank switching is the next step.

One of our Computer Science graduate students is adding a Z-80 board, similar, in function, to that made for the Apple II by Microsoft, so that he can run CP/M and UCSD Pascal on his SYM. His system expansion bus will be S-100. We'll keep you posted on this one!

TAKE A 'BREAK'

Here are some 'trivia' on the BRK instruction and the B status bit in the status register. While BRK is usually considered a one byte operation in the 6502, in some ways it actually is a two byte instruction, with the second byte being ignored. If you look up the specs on BRK in the 6502 Programming Manual, you will see that BRK causes the Program Counter to advance by 2.

It is important in programming for KIM, if you wish to continue after a BRK stop, to put in a 'dummy' byte. In the SYM, the 'saved' value of the Program Counter is decremented by 1 in the SAVINT subroutine, so that a dummy byte is not needed. If you have both a KIM and a SYM, it would not hurt to follow each BRK with a NOP to make the programs more transportable.

You will learn a lot about the 6502 interrupt capabilities, and see that BRK, in effect, generates an IRQ in software, by studying the monitor interrupt subroutines from $800F to $80AC. You will see how MON makes returning from a BRK identical to returning from an interrupt. During the BRK, you may examine and, if you wish, modify memory and/or registers. Whether you got to this point by NMI, IRQ, BRK, or USRENT, reentry is via the zero parameter .G, through $83F3, which eventually returns you to the 'interrupted' program with an RTI, after restoring the departing conditions. Very elegant programming here. And we have not even discussed how program trace (for all non-MON instructions) is implemented through NMI! The trace program is also worth studying, to see how NMI is used.

IRQ and BRK treat the Program Counter differently. BRK increments by 2, since its instruction has been completed. IRQ does not increment, since it occurs just before the next instruction is to start. Otherwise, with but one minor, but important, exception, the 6502 handles IRQ and BRK nearly the same way. For both, three bytes, PCH, PCL, and P (the status register, i.e., flags) are popped on the stack. During IRQ the B flag is popped as a zero. Any other transfer of P to the stack, as with PHP, for example, or with BRK, pops B as a one. As was pointed out by a reader (can't remember whom; will credit him in the next issue, if he reminds me) the only place where this bit of trivia is documented is in Table 4-5 of the Reference Manual, where it is explicitly stated that PHP sets B to 1.

The 'expansion' bit (bit 5) also pops, and is therefore pulled, as a 1. Has anyone found a use for this bit, or some other way to set it (it clears on RST)? The overflow bit (bit 6) can be set from the outside world, if desired, by a negative going edge at the S.O. (also called RD) input (pin 38) on the 6502. This is brought out at E-5. Has anyone out there made use of this input?

HOW TO MAKE SYM EXECUTE YOUR COMMANDS

We have automated our cassette production by giving SYM the necessary commands to LOD selected files from disk and .S2 them to tape. This is done by making extensive use of SUPERMON's .E (Execute) command. For those of you who are not familiar with the .E command, we give two simple examples:

EXAMPLE 1: Suppose you wish to dump multiple copies of a program from $0200 to $0347 to cassette with ID = 01. Using either .M or .D (we prefer .D), and the ':' for ASCII input feature, enter the following sequence at, say, $0100:


```
.D 0100
0100 :S :2 :0 :1 :, :0 :2 :0
0108 :0 :, :0 :3 :4 :7 0D :E
0110 :0 :1 :0 :0 0D 00
```

The 0D is ASCII Hex for CR; the 00 is the terminator for any .E sequence. After entering the above, start your recorder going, hit RETURN, enter .E 0100, and hit RETURN again. Stop the recorder, and stop the recording with RST when you have made enough copies. You can obviously modify the sequence to dump selected blocks a fixed number of times. Before we put our Disk Bootstrap into EPROM, we had it on cassette. We used this technique to fill the full lengths of both sides of the cassette tape with the Bootstrap, so that we never had to waste time rewinding, or look to see which side was up.

EXAMPLE 2: A cassette save and load cannot be made over the top of page 0 (the cassette loader pointer is at FE,FF). Save and load over the top of page 1 is not a good idea, since it clobbers the stack. Thus, programs which require initialization in pages 0 and 1 should contain their own initialization subroutines. Two alternatives are: 1) Dump the program in three sections, and 2) Dump the program in one section, with the pages 0 & 1 blocks in higher memory, and move these blocks down with .B's before the .G .

The first alternative is 'automated' by writing a loading sequence including :L :2 0D :L :2 0D :L :2 0D 00 , and dumping it to casette preceding the three sections of the program. .L2 in the loading sequence, .E to its address, and it will bootstrap in the three sections of the program.

The second alternative is 'automated' by appending a move-and-go 'Execute' sequence to the program, which contains the necessary :B's to do the moving, and the :G to start the program. After the tape is read in, start the program, not with a .G to its starting address, but with a .E to the starting address of the 'Execute' sequence. Don't forget to terminate the sequence with "00", and to use "0D" for CR.

Now that you have seen examples of how to use .E, study its source code carefully, and you may discover how to extend its capabilities greatly, by writing your own version, and setting EXEVEC at $A672 to point to your version.

Note that .E can accept up to three parameters, but that the version in SUPERMON uses only one. Your version can use the other parameters to set vectors, print out messages, call subroutines, etc. Note that EXEVEC normally points to RIN, and that Execute essentially replaces INVEC(+1) with RIN, so that 'inputs' will come from RAM (or ROM, if you wish). You can do something similar with OUTVEC to steer outputs to RAM rather than to the terminal. You may now wish to reexamine the MERGE/DELETE Program for SYM BASIC on page 1-13 to see how this was done by Tom Gettys.

DISKS AND TAPES AND GRAPHICS AND APPLES

As mentioned elsewhere in this issue, the SYM can be taught to read Apple II generated tapes. In fact, on the 'other' SYM (the 'fun-one'), right now, there is a high resolution (B/W) picture of 'Hopalong' Cassidy, with an excellently simulated, quasi-halftone, gray scale. This was transfered to the SYM from one of the school's Apples, by dumping to cassette the memory-mapped image. The Apple II memory mapping is not, as we mathematicians say, a continuous one-to-one transformation, but the cassette read-in program makes the proper transformation. The Apple tape sounds different from a SYM tape; when we get the time, we'll compute and compare the data rates.

This little experiment was done to check out the read-in program, while waiting for two Apple Disks to be delivered to us by an extremely slow Apple Dealer, from whom we ordered them, over a month ago. These disks contain two "Magic Lantern" graphics packages for the Apple. When they arrive, we will read them into the Apple, dump the "frames" onto cassette, read these frames into the SYM, and dump them onto cassette (later onto disk, when our graphics SYM gets its own disk drives), for future enjoyment on the SYM. We paid the asking price for these disks and will be using them only on one system. The intermediate copies will be solely for the purpose of making the machine-readable medium readable on our machine.

The thought naturally occurs: Why not bypass the double cassette transfer, and fix up things so that our SYM can directly read the Apple Disks? We realize that the Disk II System is copyrighted, but then, so too, is the Apple's Cassette Firmware; yet a variant of this was published in the open literature. If we buy Apple Software to use on our SYM, it is obviously to Apple's advantage to let us be able to read the purchased software. Suppose we buy software written for the Apple by others; Apple likes to have others market software for the Apple, since the existence of such software helps to sell Apples. In our case, however, we would be using a modification of Apple's disk software, but with no advantage to Apple. This is rather a sticky problem area, no?

I'm not sure any one can really advise on the new (1978) copyright law, because there have been very few test cases on the principle of "fair use" of copyrighted material. Fair use implies non-commercial use, but only in ways which do not injure potential sales by the owner of copyright. Commercial use, i.e., piracy, would be easy to prosecute. It would be very difficult to prove that "wholesale" copying for distribution to club members, say, hurts the sales of a product, because the club members might not buy, if they had to pay.

Our own standpoint on the use of Apple software for the SYM is that we will buy such Apple (or Pet, or OSI, or whatever) software, for our personal use, if a) we want it, and b) we can figure out a way to read the purchased media with the SYM, and modify the software, as required, for SYM use. This in spite of, and possibly because of, the fact that some vendors of Apple disk software take advantage of the Apple Monitor and the DOS capability to cause the disk record to "self-destruct", if the purchaser should attempt to make a back-up copy.

Thus, in the very near future, we shall try to teach our old SYM a few new tricks, such as how to read an Apple disk.

REMARKS RE RAE-1

Carl Moser has provided us with a listing of the source code for RAE-1, so that we can more easily answer your RAE questions, and so that RAE NOTES NO. 3 can be more definitive about certain points. There is only one real "bug" that we have discovered in RAE. The pseudo opcode .EJ (essentially a "form-feed") falls one line short of the 66 lines necessary for an 11 inch form.

Many of you have objected to the "//" and the ">" prompt at the end of a manuscript. These can be suppressed by pointing OUTVEC to a patch which watches for these characters, and replaces them with nulls, before calling OUTCHR. If you want a (single) slash to appear as part of the text, your patch should store slashes each time they are sent out for printing, and wait for the following character to appear. If the following character is a second slash, suppress both; otherwise print both. If you wish the > to appear as a prompt on the CRT, but not on the hard copy, your patch should examine the "Hard Copy Flag" at $011F.

COMMENTS ON SWP-1

We have modified our own version of SWP-1 (with Tom Gettys' help) to suppress the word "PAGE" (if you want the word, put it in as part of the title or footer, as we often do, in lower case), and the leading zeroes in the page numbers.

Tom has also added the ability to continue a manuscript from either tape or disk. We are now editing Jack Brown's SYM-FORTH Manual with SWP-1, and it is still like magic to us to see SYM print out a 75 page manual with no human intervention, after the initial call to SWP.

We are not too happy with SWP's lack of a simple way to "TAB". The current way is to force spaces with a sequence of "up-arrows". Be careful with the use of spaces before and after up-arrows (best not to use them) since SWP "kills" spaces before them, and "transfers" all spaces after them, to the right-hand end of the line. SWP collapses all strings of spaces to a single space, except after a ".", where it prints a double space, to mark the end of a sentence. If you only want one space after a ".", use an up-arrow instead of a space. We leave it to the user to figure out how to get SWP to put two spaces between the "?" at the end of a sentence, and the first word of the following sentence.

We are studying the source code of RAE to see how it handles the tabs (only in steps of eight columns) to see if this method may be incorporated, or improved on for SWP. When this is done we will issue a SWP-2. To keep the faith with owners of SWP-1, the price of SWP-2 will be increased by the same amount that SWP-1 owners will be asked to send in to cover the costs of printing and mailing a listing of the enhancements to their current version. OK?

RECOMMENDED READING

Ever since we first begain using our KIM-1, we have looked forward to reading each of Professor Marvin L. De Jong's articles, first in the KIM-1/6502 User Notes, then in MICRO and in COMPUTE. We have long considered him to be "Mr. 6502", or, rather, "Prof. 6502". We were, then, very pleased to hear of his new book on the 6502. We are pleased, too, that Bob Peck agreed to review the book for all of us. We will add only one comment to Bob's review: If we could have only one book to go with our SYM, this would be the one.

PROGRAMMING AND INTERFACING THE 6502
with Experiments, Marvin DeJong
Howard W. Sams, 1980, $13.95

Robert A. Peck
DATAPATH, P.O. BOX 2231
Sunnyvale, CA 94087

I have been teaching assembly language programming for the past year at a local engineering college. As part of this teaching experience, I have collected quite a large number of books on the 6502 and other processors to try to determine the best approach to teaching assembly language programming. This book appears to have taken a different approach than any of the others I have seen.

Almost all other microprocessor books take the idea of introducing all of the addressing modes, then introduce the whole instruction set next. Then show subroutines, I/O devices, interrupts and so on. This information may occupy, as it does in many texts, as little as one chapter in the book with the rest dedicated in some way to applications of one kind or another. As an instructor, I have tried in the past to follow the outline of the texts I have used in this way but have come to realize this is a lot of data to throw at a student in one blob. I came away from earlier experiences believing that these other texts could serve the student as reference material once the student had been explained the techniques of this

type of programming but there was no text available which could lead the beginning student through assembly language programming without tossing everything at the student at once, making the concept itself even more difficult to understand.

Marvin has placed an understanding of the nature of the instruction stream as the prime goal of his book. He introduces the instruction set a few related instructions at a time. Each group is shown with as few basic addressing modes as needed for understanding of the function itself. Additional addressing modes are only shown as the progression of the instruction set explanation requires.

By a careful selection of the order in which the instructions are discussed, he is able to begin illustrating the functions of each by means of typical assembly language programs from the second chapter onward. By this means he is allowing the student immediate familiarity with the techniques the student will be using to generate his own programs.

Each grouping is then thoroughly discussed and illustrated by example and experiments which the student can do on a SYM or other 6502 unit. After the entire instruction set has been fully examined in this manner, he includes a section on hardware interfacing to the 6502 processor, continuing the emphasis on the programming aspects of this interface.

For anyone with an interest in learning the 6502 assembly language programming, with or without an instructor, I recommend this book highly. As an invitation to make your own judgement on the merits of this book, look at the table of contents of this book alongside that of any other 6502 programming book available today. You will see that it is more intensively dedicated to exactly the business at hand, that of showing the techniques of assembly language programming than the others. It does not include some of the filler material which is 'nice' to have at times but for a student to be able to judge what is filler and what is really necessary makes this book well worth its price in that it is all of value.

I intend to continue to use this as a text for the hardware and the programming courses in which I have future occasion to be involved. Based on Marvin's approach and the completeness of this text, I only hope that if someday I am called on to teach the Z-80, Z-8000, the 68000, or the 8086, I would hope that, by that time, Mr. DeJong might have had the opportunity to produce a text of this quality for this set of processors as well.

A BASIC WORD PROCESSING SYSTEM

Here are two BASIC programs submitted by KIN-PING KWOK, 22 Tung Choi St., Tat Ming Bldg., 10/F, Flat A, Kowloon, Hong Kong. We have not had the opportunity to test them, but the explanation of the programs, which appears following the programs, shows the right justified text which they produce. For those users who prefer BASIC to Assembly Language, or who, for any other reason, do not have RAE-1 installed, and thus cannot use SWP-1, these programs will provide a word processing capability.

```
10 ST=&"1000":DIM X(2):X(2)=8190+ST
20 GOTO 200
100 CH=PEEK(CS):IF CH=0 THEN NU=0:GOTO 200
110 IF CH=13 AND NU<>0 THEN CN=CA:GOTO 200
120 IF CH=13 THEN NU=-1:PRINT
130 DI=USR(&"8AA0",CH*256)
140 CS=CS+1:NU=NU+1:CA=CA+1:IF NU<400 THEN 100
150 CN=CA
200 CH=INT(USR(&"8A58",&"D14B",0)/256) AND 127
```



```
210 IF CH<32 THEN 500
220 POKE ST+CA,CH
230 CA=CA+1
240 IF CA>8000 THEN PRINT CHR$(7);
250 IF CA>8190 THEN CA=8190:PRINT CHR$(8);
260 GOTO 200
500 IF CH>7 AND CH<14 THEN ON CH-7 GOTO 1000,1050,1100,1150,200,1200
510 IF CH=5 THEN 2000
520 IF CH=17 THEN IF NU=0 THEN 200
525 IF CH=17 THEN NU=0:CA=CN:GOTO 100
530 IF CH=6 THEN X(0)=CA+1+ST:X(1)=X(0)-1:CN=CN+1:CS=CS+1:GOTO 550
540 IF CH<>2 THEN 200
545 X(0)=CA+ST:X(1)=X(0)+1:CN=CN-1:CS=CS-1
550 FOR DI=0 TO 2
570 CH=INT(X(DI)/256)
580 POKE 42574-2*DI,X(DI)-CH*256:POKE 42575-2*DI,CH
590 NEXT
595 DI=USR(&"8740",0)
600 X(0)=CN-CA:IF X(0)<1 THEN 200
610 FOR CH=0 TO X(0)-1
620 DI=USR(&"8AA0",256*PEEK(CA+ST+CH))
630 NEXT
635 PRINT " "+CHR$(8);:POKE 25,0
640 PRINT CHR$(27)+"+";:X(1)=INT((960-X(0))/40)
650 PRINT CHR$(32+X(1))+" ";
653 DI=960-X(0)-X(1)*40:IF DI=0 THEN 200
655 FOR CH=0 TO DI-1:PRINT CHR$(9);:NEXT
660 GOTO 200
1000 CA=CA-1:IF CA<0 THEN PRINT CHR$(9);:CA=0
1010 GOTO 240
1050 CA=CA+1:GOTO 240
1100 CA=CA+40:GOTO 240
1150 CA=CA-40:IF CA<0 THEN PRINT CHR$(10);:CA=0
1160 GOTO 240
1200 PRINT:POKE ST+CA,CH:GOTO 230
2000 IF NU=0 AND CA>0 THEN POKE ST+CA,0
2005 PRINT
2010 INPUT "COMMAND ";IN$
2020 IF IN$<>"L" THEN 2010
2030 CA=0:CS=&"1000":NU=0:GOTO 100
OK
```

```
10 N=70
11 L=10
12 A=0:A$="":B$="":B=0:CS=&"1000"
13 I=1:C$=A$:C=0:P=0:Z=0:T=2
18 X8=59:X9=63:X0=13:X1=32:X3=43:X4=48:X5=33:X6=41:X7=58:E$="  "
30 FOR A=LEN(A$)+I TO N
40 B=PEEK(CS):A$=A$+CHR$(B):CS=CS+I:IF B=Z THEN END
45 IF B=X0 THEN C$="":GOTO 500
47 NEXT
48 A=A-I:B=ASC(MID$(A$,A,I))
50 IF(B>X3ANDB<X4)ORB=X7ORB=X8ORB=X9ORB=X5ORB=X6THENP=A:GOTO 80
60 IF B=X1 THEN P=A-I:GOTO 80
70 GOTO 48
80 BL=Z:B1=Z:C$=""
90 B=PEEK(CS):C$=C$+CHR$(B):B1=B1+I:CS=CS+I:IF B=Z THEN END
96 IF B=X1 OR (B>X3 AND B<X4) OR B=X5 OR B=X6 THEN 110
100 IF B<>X7 AND B<>X8 AND B<>X9 THEN BL=I:GOTO 90
110 IF B=X1 AND BL=Z THEN P=N:GOTO 90
120 IF P=N THEN 500
130 D$=A$
140 C=N-P
145 A=I
150 B$=MID$(A$,A,I):IF B$="." OR B$="," OR B$=";" THEN 160
155 IF B$<>":" AND B$<>"!" AND B$<>"?" THEN 180
```

```
160 IFMID$(A$,A+I,T)<>E$THENA$=LEFT$(A$,A)+" "+MID$(A$,A+I):C=C-I:P=P+I
163 IF C=Z THEN 170
165 IFMID$(A$,A+I,T)<>E$THENA$=LEFT$(A$,A)+" "+MID$(A$,A+I):C=C-I:P=P+I
170 IF C=Z THEN C$=MID$(A$,N+I)+C$:A$=LEFT$(A$,N):GOTO 500
180 A=A+I:IF A<P THEN 150
200 A=I
210 IF MID$(A$,A,I)<>" " THEN A=A+I:GOTO 250
220 IFMID$(A$,A,T)<>"  "THEN A$=LEFT$(A$,A)+" "+MID$(A$,A+I):C=C-I:P=P+I
230 IF C=Z THEN 170
240 A=A+I:IF MID$(A$,A,I)=" " THEN 240
250 IF A<P THEN 210
260 A$=D$
270 IF B1=I THEN C$=RIGHT$(A$,T)+C$:A$=LEFT$(A$,N-T)+"-":C=I:GOTO 145
280 C$=RIGHT$(A$,I)+C$:A$=LEFT$(A$,N-I)+"-"
500 PRINT TAB(5)+A$
510 A$=C$
520 IF LEFT$(A$,I)=" " OR LEFT$(A$,I)=CHR$(X0)THEN A$=MID$(A$,T):GOTO520
530 GOTO 30
OK
RUN
```

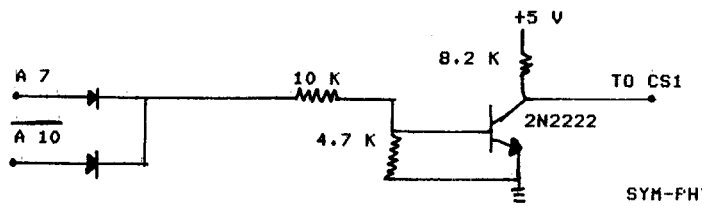
The first program is for input and edit. It store ASCII start from $1000. You can use cursor control to edit the input passage. Type CTL-E to end the input. If you type L to reply COMMAND, the passage will list for you to edit. It stops when it meet a CR or zero or it had list 400 characters. Use cursor control and CTL-F and CTL-B to edit, insert and delete. Varaible N in the second program is the number of characters per line. Since the programs are written in BASIC, the speed of the program is not very fast. You can change them into machine code programs to increase the speed and that is not very difficult. At last, the programs are written for KTM-2/40, change 40 to 80 in line 640,653 in the first program for KTM-2/80.

HOW TO 'REWIRE' THE VIA AT $A800

As mentioned elswhere in this issue, additional VIAs may be addressed in the $A800 block by inhibiting the enabling of U28 when address bit A7 is high.

Lev Barshay, of Nestle, elected to cut the existing $\overline{AA8}$ trace between pin 6 of U10 and pin 23 of U28 (VIA #2). Note that the latter is the $\overline{CS2}$ input, which is active low. The new input to pin 23 is obtained by "or"-ing A7 and $\overline{AA8}$ (active low) in 1/4 of an open collector 74LS32, and using the existing R60 as the pull-up resistor. Note that if this is done properly, VIA $\overline{\#3}$ at $AC00 is also inhibited, since its pin 23 shares the same AA8 and pull-up resistor. Lev mounted the 74LS32 by cementing it, pins up, to his SYM, near U24. He obtained $\overline{AA8}$ from pin 6 of U10, and A7 from pin 1 of U20, although other convenient sources of these signals could have been used. The output of the "or" circuit goes to the "low" end of R60.

Darrell Johansen, of Sense Modular, decided to work with pin 24 of U28. This is CS1 for VIA #2. He cut the existing trace to pin 24, and used one of the four buffers in the lower left corner of SYM to fashion the following "nor" circuit. Note that he now uses $\overline{A10}$, not A10, as previously, because of the inversion provided by the "nor".



THOUGHTS ON SMALL SYSTEMS AND MONITORS [MOSTLY THE SYM-1]

The title above is not ours; rather it is the title of an article by H. T. Gordon, in Dr. Dobb's Journal of Computer Calisthenics and Orthodontia, Number 48, September (?), 1980.

We personally read each issue of Kilobaud Microcomputing, MICRO, BYTE, and COMPUTE., and call to your attention any SYM relevant material. Tom Gettys reads many of the other computer magazines, and called the above mentioned article to our attention. Some of Professor Gordon's thoughts are worth quoting here; we shall do so, but recommend that you read the entire article, if you can:

> "The concept that has always appealed to me is more that of the "hi-fi" enthusiasts, where a system consists of several interacting but independently-replaceable components, from many competing sources. True competition, however abhorrent to industrial giants, makes possible a maximum of progressive change at the minimum cost to society. This eccentric, unbundled-components point of view enabled me to resist the lure of the increasingly powerful packaged systems that have entered the market in recent years. However flexible, they must be designed for some least-common-denominator purpose and tend to allocate large resources to things like BASIC interpreters (for me a turn-off). They are like low-cost smoothly-paved roads leading where everyone else wants to go."
>
> "Even the minority of programmers who have read the ROMS created by someone else do not fully appreciate the problem until they toy with the idea of creating their own. It's the chasm between critic and artist, or rather between a builder of sand-castles and a sculptor in marble."

Professor Gordon goes on to discuss monitor ROMS, and terminals, and related topics. His article inspires me to disassemble the KTM-2/80 ROM to see what kinds of enhancements could be added by replacing the ROM with my own EPROM. He closes with the following thought:

> "A brief afterthought on FORTH. To me, FORTH has been the most tantalizing of the existing HLLs. Perhaps the word is infuriating, since the FORTH enthusiasts - like the Rosicrucians or the initiates to the ancient Eleusynian mystery - won't tell you what it is. In comparison, my own much more miniscule programs come with a surfeit of explanatory comment (tinseled with allusions). When I glanced at a listing of FORTH, its most striking quality was the virtual absence of comment. Adam Osborne recently observed (InfoWorld 2(8):7, 1980) that the success of an HLL depends less on its intrinsic merit than on how hard it's pushed. Whatever the demerits of BASIC - and they are legion - being unexplained is not one of them. Dozens of books expound it in great detail, and some are brilliant. Where is the book that describes how FORTH works, from the ground up, in a painstakingly detailed, translucent and vivid way?"

By HLL, Dr. Gordon means higher level languages. We also very much like his use of the term "machine-linguists" for machine language programmers. Our answer to his closing question is Jack Brown's Manual for SYM-FORTH. Jack Brown's FORTH programs are more heavily commented than most of the BASIC programs we have seen lately. We will send Professor Gordon a preliminary copy of the SYM-FORTH Manual for his review and comments.

MORE ON TEXT EDITORS AND WORD PROCESSORS

We had just this morning delivered the camera-ready copy for a 44-page issue of SYM-PHYSIS to our printer, when we received a cassette in the mail from Frank Winter, whose article on TOPS we published in Issue No. 4. (See pages 3-25 and 4-12.) After looking over his word processing program on cassette, in RAE format, we decided to "stop the presses", and publish his letter and these comments. He is therefore responsible for this issue containing an extra four pages more than we had planned.

We like SYM mainly because of its RAE and SUPERMON firmware, and we use RAE as our text editor, because it is there!!! With SWP-1 added, our requirements for a word processor are fully satisfied. That doesn't stop us from examining and appreciating others, however.

We realize that many of you have no need for an assembler, so you have no need for the "A" part of RAE (Resident Assembler Editor). What other options are open, if you just need an Editor? If you have BAS-1, you can use BASIC for your word processing needs. That is why the BASIC word processing program by Mr. Kwok appears in this issue.

But what do you do for a word processor, if you don't have either BASIC or RAE? Very simple; just get a copy of TEC 65 from the 6502 Program Exchange (address is elsewhere in this issue). TEC 65 is a really fine text editor. We publish Frank's letter to show you how TEC 65 can be greatly extended. We will send a listing of the source code to Dave Marsh of the Program Exchange. Perhaps we can work out a three way arrangement to provide our readers with Frank's Enhanced SYM Version of TEC 65. Are any of you interested? Here is Frank's letter:

Dear Lux,

I enclose a copy of some modifications to the TEC 65 text editor, which I understand you purchased some time ago.

I was quite impressed with the capabilites of this language, but found the lack of formatting a problem. This arises when you alter the original text, and you still want a neat printout.

The enclosed program was specifically written to link up with TOPS (by the way thanks for publishing my comments). I don't think that it would be very difficult to change it to suit your disk operating system.

The following commands are available when you activate the formatter:

\T10\ sets the top margin to 10 lines
\B10\ sets the bottom margin to 10 lines
\L5\ sets the left margin to 5 character spaces
\R58\ sets the line length to 58 characters
\P\ sets the page length to 60 lines
\I10\ breaks and indents 10 characters
\E1\ ejects TWO pages. \E\ ejects to the end of the current page
\M1\ breaks and sets line spacing to two (ie one blank between lines)
\S2\ breaks and spaces two lines regardless of the invoked linespacing

When the formatter is not activated the text is printed exactly as it resides in the buffer.

My doctoral thesis is entering its final stage, and I plan to submit it

by February 1981. My next project is a text for Operations Research, which will emphasise the use of personal computers.

I hope you are over the major hurdles of getting your book published, and I look forward to the next issue of SYM-PHYSIS. In the meantime, kind regards from down under. It is getting bloody hot now - I really should get my office air conditioned!

Frank....

[Editor's note: Winter is enjoying (!) summer in Australia right now!]

MYSTERY PROGRAM

Here is a BASIC program which looks interesting, and quite useful. We don't know who submitted it, because it somehow got separated from its transmittal letter. We vaguely remember writing the author, to ask if he would mind resubmitting on cassette, because we were too busy to key it in for test and reproduction. Our filing system is such that the original letter has been misfiled forever, and we don't recall ever getting a cassette. If the author will let us know, we'll give him full credit in the next issue. Incidentally, here is a good example of incomplete documentation. Every program you write should be "signed and dated" in a comment line (and even copyrighted, maybe).

NOTE: This program must be used with BBE-1 (Brown's BASIC Enhancements).

```
100 CLEAR
110 PRINTCHR$(27)+"E"
120 GOSUB 680:DIM A$(50):DIM N$(50):DIM P$(50)
130 PRINTCHR$(27)+"E"
140 GOSUB 680:PRINTTAB(10);"**MENU**":PRINT:PRINT
150 PRINT"TO BUILD A FILE TYPE 1"
160 PRINT"TO SEE FILE-TYPE 2"
170 PRINT"TO SEE INDIVIDUAL NAME TYPE 3"
180 PRINT"TO CORRECT-TYPE 4"
190 PRINT"TO SAVE FILE-TYPE 5"
200 PRINT"TO GET FILE FROM TAPE TYPE 6"
210 INPUT Q:ON Q GOTO 220,320,350,430,580,640
220 INPUT"WHEN READY HIT RET (TO CLOSE THE FILE TYPE END FOR NAME)";X
230 FOR I=1 TO 50:PRINTCHR$(27)+"E":GOSUB 680:PRINT"ENT NAME"
240 PRINT"HIT 'RETURN' KEY";:INPUT N$(I)
250 IF N$(I)="END" THEN P1=I:GOTO 300
260 INPUT"ENT ADDR";A$(I)
270 INPUT"ENT PHONE #";P$(I)
280 IF FRE(X$)<100 GOTO 300
290 NEXT
300 PRINT"FILE CLOSED--":INPUT"TO SEE MENU,HIT 'RETURN'";X
310 GOTO 130
320 PRINTCHR$(27)+"E":GOSUB 680
330 FOR I=1 TO P1:PRINTI,TAB(7)N$(I),A$(I),P$(I):NEXT
340 INPUT"TO SEE MENU HIT 'RETURN'";X:GOTO 130
350 PRINTCHR$(27)+"E":FOR E=1 TO 10:NEXT:INPUT"ENT NAME";N$
360 FOR I=1 TO P1:IF N$(I)=N$ THEN 390
370 NEXT
380 PRINT"NAME NOT IN FILE":GOTO 400
390 PRINTN$(I),A$(I),P$(I)
400 PRINT:PRINT"FOR CONT. TYPE 1, TO STOP TYPE 0";:INPUT X
410 IF X=1 GOTO 350
420 GOTO 130
430 PRINTCHR$(27)+"E":GOSUB 680
440 PRINT"ENTER THE LINE'S NAME TO BE CHANGE":INPUT N$
450 FOR I=1 TO P1:IF N$=N$(I) GOTO 480
```

```
 460 NEXT
 470 PRINT"NOT IN FILE":GOTO 550
 480 PRINT"ENTER CORRECTED INFO."
 490 INPUT N$(I),A$(I),P$(I)
 500 FOR T=1 TO P1
 510 IF T=I THEN T=T+1
 520 IF N$(I)=N$(T) THEN PRINT"EXIST ON LINE";T:PRINT
 530 NEXT
 540 PRINT:PRINT: PRINT"THE LINE NOW IS:":PRINTN$(I),A$(I),P$(I)
 550 INPUT"FOR CONT. TYPE 1, TO STOP TYPE 0";X
 560 IF X=1 GOTO 430
 570 GOTO 130
 580 PRINTCHR$(27)+"E":GOSUB 680
 590 INPUT"MAKE PREP. FOR CASSETTE, WHEN READY HIT RETURN";X
 600 PRINT"COPYING"
 610 .SAVEV 1
 620 PRINT"DONE"
 630 INPUT"TO SEE MENU, HIT RETURN";X:GOTO 130
 640 PRINTCHR$(27)+"E":GOSUB 680:INPUT"WHEN READY, HIT RETURN";X
 650 PRINT"LOADING DATA"
 660 .LOADV 1
 670 PRINT"DATA LOADED":INPUT"TO SEE MENU, HIT RETURN";X:GOTO 130
 680 REM *** TIME DELAY RUTINE FOR CLEAR SCREEN***
 690 FOR E=1 TO 10:NEXT:RETURN
OK
```

MORE ON SOUNDS AND MUSIC

We see from the 1981 Radio Shack Catalog, not only that TI's SN76477 has gone up in price, but more importantly, that TI has introduced a new sound generation chip, the SN76488, more amenable to computer control, at $6.99 (RS 276-1766). If any of you try it, please let the rest of us know your results.

We used to think that the MTU Advanced Music Package was the greatest thing in the computer music business since the invention of the square wave. But, not any more! Now we feel that the MTU Advanced Real-Time Music Synthesis Techniques Package has taken its place. We suggest you read Hal Chamberlin's article (with the above title) in BYTE, April 1980, and, if this interests you, to send for the stereo audio demonstration cassette, which we have available. The MTU package, with full source code listing, and three demonstration scores, is available from the Users' Group in SYM readable format. The program is memory intensive. Only one of the three demos will work in a 8 K system; the other two require 16 K and 32 K, respectively.

KTM-2 TO KTM-2/80 CONVERSION

Bob Myers called today, just in time to get into this extra page, to tell us that he is now ready to start shipping the KTM-2 Upgrade Kit. The Kit includes two Synertek ROMS, full instructions, and 'artwork' to show where to make the trace and jumper modifications. The cost for the Kit is $65, plus shipping charges. You will need to buy sockets, a pair of 2114s, and a pair of other ICs, in addition to the parts supplied with the kit.

Bob asks us to advise those who have written and received no answer from him, that, while he was on an extended business trip, his office was moved from one building to the next, and that many of his papers got "lost" during the short haul. Please write him again with any questions, or to place your order. His address is on page 4-23.

>

AND SOME WORDS ABOUT WORDS

If you have interfaced TI's "Speak & Spell" to your SYM, you will be interested in the Phoneme Software Package, available from S.PEE.K uP SOFTWARE, 6710 Forest Bend, San Antonio, TX, 78240.

MORE ON DISK SYSTEMS

Quite a few of our readers are beginning to add HDE File Oriented Disk Systems (FODS) to their SYMS. We (Tom Gettys and I) provided Dick Grabowsky of Hudson Digital Electronics, Inc. with Version 1.0 of our SYM-FODS software package. Since that time SYM-FODS has been extended considerably. The major extensions have included adding .CT (name) to RAE-1 to permit Continue on Disk, and .DISK (filename) to SWP-1 to permit concatenating discrete files into one long sequentially page numbered document (fortunately, just in time to handle Jack Brown's very thorough (and very long) SYM-FORTH Manual).

SYM-FORTH has been designed from the first to work with a simulated (cassette/RAM) Disk System, and can easily be patched to "for-real" Disk System. We are making arrangements with HDE to provide SYM-FODS users with all extensions on a timely, non-profit basis.

CHECKOUT TIME

May we wish all of you the appropriate Season's Greetings, and a Happy New Year, even though a little early? Issue No. 7 will reach all re-subscribers in late January or early February. Meanwhile, we will concentrate first on answering the pile of old letters, some nearly a month old. They come in at the rate of 2-3 per day, so there must be perhaps 70 or so. Next, RAE NOTES NO. 3.

Teachers glory vicariously in the achievements of their students, in the same way that parents do with their offspring. Two years ago, in our KIM days, Steve Crescenti developed the software for a laser graphics system, tested on an oscilloscope. During the past year, Tom Gettys developed the foundations for a SUPERMON Extension Package, and Paul Close implemented a Voice Recognition System (12 word vocabulary, cooperative speaker, 95% recognition). These tasks were parts of their Master's Degree projects. Two have gone into industry; Tom is teaching at Cal State, Chico.

This current year, Hamid Kahansi is well into his project of Apple II/SYM-1 Complete Information Interchange (ASCII?), involving cassette, disk, and RS-232 subsystems. Peggy Leung, our first woman student to become interested in micros, is using the SYM as an intelligent controller interface between an H-P System and a surplus incremental plotter. Also, several students, both men and women, in our Industrial Technology major, with strong electronics backgrounds, are beginning to think micro-digital! During the next year there should be many interesting results to report to you.

We know that we ourselves will not be able to advance the state-of-the-art of Computer Music, Computer Speech, Pattern Recognition, and Image Processing significantly; we hope that some of our students may. We do hope to become very skilled in FORTH, because of our feeling that FORTH will enable us to accomplish our system design goals more rapidly than any other programming tool (to us languages are merely tools for communication).

We will close with the same words that Jack Brown used in a recent letter to us: "May the FORTH be with you!"