```
 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  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   S  S   S  S
 SSSS    S   S  S     S     S      S   S   S    SSSS   SSS   SSSS
```

THE SYM-1 USERS' GROUP NEWSLETTER

ISSUE NUMBER 3 - MAY/JUNE 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.



Editor/Publisher: H. R. 'Lux' Luxenberg
Business/Circulation: Jean Luxenberg
Associate Editor: Thomas Gettys

SUBSCRIPTION RATES:

USA/Canada $9.00 for a volume of 6 issues; overseas $12.50. Make checks payable in US dollars to "SYM Users' Group," P. O. Box 315, Chico, CA 95927, Telephone (916) 895-8751.

## FROM THE EDITOR

This issue, as promised, is heavily devoted to computer music and graphics. First, though, let us point with pride to our 'new look'; please observe the right justified text. No more sloppy ragged right margins. We'll tell you later how it was done, and tell you how you, too, can make a high-class word processor out of your SYM-1.

As for graphics, we will present examples of both oscilloscope and KTM-2/80 programs. And for the music, we will concentrate mainly on the D/A (DAC or Digital-to Analog Converter) approach, although other methods will be described. We developed a number of music and graphics programs for our KIM several years ago. These were hand assembled, patched for the SYM, relocated, modified to include TSTAT so we would not have to hit RST to get out of an infinite loop, etc. There is no source code for them. We therefore will publish them in disassembled form, and refer you to the original articles for the comments.

One of our graduate students developed some scope graphics for the KIM as part of a graduate project. These included a PONG game, a Bouncing Ball simulation, a Raster Graphic Display, and a Vector Display which showed five lines of five alphanumerics (sort of a crude typewriter). All but PONG, which depends on the KIM keypad logic, have been "transcribed" for the SYM. We are now fixing up a simple-minded two-axis laser deflection system involving mirrors and speakers to produce wall sized laser graphics. Unfortunately, progress is slow.

Happy reading, hardware put-togethering, programming, and then, watching and hearing your SYM perform!



## SYM WORD PROCESSOR

This issue was "set" with an early version of Carl Moser's new SYM Word Processor (SWP). We were sent a preliminary version for testing and debugging. We reported the minor bugs back to Carl, and suggested some new features to be added. The improved version, SWP-1, is now available on cassette. No printed manual is provided, but, with the fully commented source code and a supplied example of a text file showing its use, the cassette material explains itself. The cassette contains three copies each of the complete source code, a version stripped of all comments, and a sample text file. The "stripped" source code will permit a quick assembly (without the need for .CT). SWP-1 does not split words, that is, it will not hyphenate for you. In wide columns this is not a major problem. In narrow columns like this one, you may want to do as we have done. If the wide spaces between words are objectionable, a few iterations of a manual hyphenization process will fix things up, as we have done here. As is our established policy, we will fully support this product with improvements, corrections, suggestions for better use, etc. SWP-1 is actually easier to use for text editing than RAE-1 alone, since there is no need to try to equalize the lines. SWP-1 puts all of the lines into one long string. After generating the text, additional lines are inserted to indicate paragraph endings, margin changes, etc. Wouldn't you like to be the first kid on your block to have a really fine, "up-to-date", truly modern, word processor? Send for yours, TODAY! See back page for ordering information.

## THE KTM-2/80

When I first saw the list price of the KTM-2/80, I thought it was a lot of money to put out for a keyboard and a handful of chips. That was before I saw what came for the money. Now, I think that it is the most cost effective terminal available, and that the price is unbelievably low for what you get. The -80 has TWO microprocessors, a 6502 and a 6507, two VIA's, 2 K of RAM and 12 K of ROM! It is a truly professional stand-alone terminal (capable of 9600 Baud) and I use it on our local timeshare system (at only 300 Baud, however). The graphics capabilities, which are actually 160 by 48 (not 80 by 24), are an added bonus. Surplus monitors are available for around $50, and a cabinet maker can make a case for under $50. Where else can you get a terminal with all of the KTM-2/80 features for $550?

I am actually beginning to think of the SYM-1 as an accessory to the KTM-2/80, converting it to a fully intelligent terminal, rather than the other way around. I have even suggested to Synertek Systems Corporation that they consider an enhanced KTM board with sockets for MON, RAE, and BAS, and 8K of RAM (giving up the hex pad and the 7-segment displays). The world's penultimate Single Board Computer! Add a single expansion board with prom burner, disk controller, and 24 K of RAM, and, with all of the fine SYM software becoming available, Synertek would have a really powerful, low-cost, super development system. Judging by the letters we receive, many of our readers are well on their way to assembling such dream systems, on their own, but not packaged as "neatly" as could be.

Many have written and called about upgrading their 40 column KTM-2 to the 80 column capability. Some bad news and some good news. If you have the early model KTM-2 (prior to S.N. 0733) it cannot be done. For the newer model, Synertek will release a conversion kit, available through the Users' Group. Wisely, Synertek is waiting until a detailed technical manual describing the conversion procedure is available. The conversion manual is being prepared by an experienced SYM-1 user, Bob Myers, 109 Fire Lane, North Cape May, NJ 08204, (609) 522-7781, x 250. Contact Bob directly for availability information; hopefully we can announce the availability of the kits and manual in our next issue.

## CASSETTE RECORDER TIPS

Our long lasting problem with unreliable cassette readback has been solved, and in a very simple manner indeed! We can now read almost any tape sent to us at any setting of the volume control above a minimum threshold. We replaced the 0.22 ufd capacitor at C16 (now on all new production, and sent with the MON 1.1 replacement kit) with the original 0.01 value. The lower value blocks out low frequency hum, flutter, and wow. We have made the change on eight of our local SYM-1s and recommended it to others, who have called concerning cassette problems, then called back to report that the fix also worked on their systems.

The SYM-1 cassette subsystem operates at 1420 Baud. That can easily be doubled, even with inexpensive recorders, by replacing the values of TAPET1, TAPET2, and HSBDRY with one-half their default values. The speed can also be tripled, or quadrupled, but at 4X (5.7 KBaud) the high frequency response of the recorder itself becomes the limiting factor. We first became aware of this capability of SYM when, on the same day, we received a "unreadable" tape (which sounded rather high-pitched) from one subscriber, and a letter from another explaining how to increase the baud rate. We found that the unreadable tape had actually been sent (unintentionally) at 2840 Baud, but was easily readable with the proper parameter values. Try the higher rates; they do save time.

A number of readers who have had problems with cassette read reliability have sent in their own "fixes", some of them requiring "heroic" measures. If the fix described here does not work for you, you might want to try the one proposed by Jay Sinnett, elsewhere in this issue. Don't be satisfied with less than nearly 100% reliability from the cassette interface. It is capable of very high reliability. Since we added our fix every cassette read failure was definitely linked to a tape defect at a specific location on the tape. Once you are sure that a particular cassette is free of "glitches" you can expect 100% readback.

One final note on reading commercially available KIM-1 format tapes which include either the top of page zero, the top of page one, or the KIM-1 System RAM at $1780+ (if you have no RAM there yourself)! Use the ID = $FF option to read in the data elsewhere, as, for example, with .L1 FF,0200. Incidentally, MON 1.1 allows you to specify the value of KMBDRY at $A631. We wonder, and will probably experiment soon, whether changing the default value to the proper choice will permit reading KIM/HYPERTAPE formats?????

## ATTENTION NEW ZEALAND SUBSCRIBERS

My colleague, Dr. Gary Sitton, Professor of Computer Science, California State University, Chico, will be in residence at the University of Canterbury, Christchurch, New Zealand, May 24-July 15. His areas of interest include Date Base Management and Operating Systems. He would enjoy meeting with any or all of you.



## HARDWARE MODIFICATION FOR BETTER TAPE RELIABILITY

Jay C. Sinnett

U.S. Environmental Protection Agency
Environmental Research Laboratory
South Ferry Road
Narragansett, RI 02882

The first cassette recorder I tried with my SYM for data recording worked extremely well. The volume and tone control settings were entirely noncritical, and I never failed to read a tape correctly. However, when I got RAE-1, I purchased two new recorders of a different make. These recorders proved to be extremely sensitive to slight changes in volume and tone controls. Even using different brands of tape was impossible without resetting the controls. Fortunately, I was able to use an oscilloscope and the Synertek tape diagnostic programs to completely solve the problem with a hardware modification. If you have had this kind of problem, you may find this suggested hardware modification useful. If your tape recorder is reliable and easy to use, don't make any changes!

The designations left, right, etc. refer to the board when oriented so that the printing on it reads normally.

1. Carefully unsolder the right-hand ends of both CR28 and CR29.
2. Bend CR29 toward the top edge of the board, so that the body of the diode extends by the left-hand end of CR28 and R93. Bend the free lead of CR29 so it touches or wraps around the left-hand lead of R94 and solder it there (ground).
3. Bend CR28 in the same direction so that it lies above CR29. Solder its free lead to the left-hand end of R95 (+5V).

Before I made this modification, I had one extremely narrow range of workable volume settings just above the threshold of detection of Sync. After this change, my volume control could be set anywhere above threshold without problems.

The reason this works is that when an audio cassette player plays back a digital waveform, the amplitude of positive-going and negative-going peaks are not always equal or even constant, but change according to the timing. When the signal input to an unmodified SYM exceeds 1.4V peak-to-peak, the diodes CR28 and CR29 conduct, causing C16 to build up a charge on each peak. This charge in turn modifies the zero crossing time, destroying the integrity of the data. The new placement of the diodes allows a signal swing of 6.4V peak-to-peak before the diodes conduct to protect the LM311 comparator.

For the hardware purist or person who has not yet installed his hardware modification which came with the Monitor update, I also recommend adding a bit of hysteresis to the new circuit to avoid noise on low-level signals. This may not be necessary in many cases.

1. Change R94 and R95 to 100Ω resistors (supplied in the Synertek kit).
2. Remove R87 and R126.
3. Change R96 to 100kΩ (you supply).
4. Install a 2.2k resistor from the right-hand end of R94 to the hole where the right-hand end of R126 was (you supply).
5. Install the R97 (1k) and C16 (0.22μf) as instructed in the new monitor kit.

## IN THE NEXT ISSUE

*A comparision of all known (to me) ways of expanding SYM-1.
*A discussion of "cheap" video terminals, and inexpensive printers.
*A description of Frank Winters' TOPS (Tape OPerating System), with nearly all the convenience of a DOS, at much slower speed, but much lower cost.
*And, of course, more programs!

## RAE NOTES

RAE NOTES No. 2 has been mailed to subscribers. No. 2 contains a full description of the disk vectors and flags built into RAE-1, and illustrates their use with the full source code listing of Tom Gettys' RAE/FODS Linking Patch. No. 2 listed six absolutely safe page zero locations completely untouched by BAS, RAE, FODS, or MON. Mailed with No. 2 was an annotated copy of Technical Note 101SSC, February 1980, "Adding Motor Control for a Second Cassette Recorder to SYM-1".

Also mailed with No. 2 was a USER PATCH FOR RAE-1 submitted by Jean Cyr, a portion of which is being published in this issue. As more of RAE-1 users begin to disassemble RAE's object code and probe into its inner workings, we can expect more enhancements to be provided. One of our readers has promised to provide a patch to suppress the // at the end of .PR. Note that SWP-1, Moser's SYM Word Processor, already does this, and the form-feed operation in SWP-1 will force the ending ">" to the top of the next page. No. 3 will include the long promised page zero/page one memory maps, and will describe the use of the Printer Control Vector built into the >HArdcopy Set command.

Please make the following correction to the RAE-1 Reference Data Card included with No. 1: In the section "Recovery from Accidental Clear" replace PR 9999 with PR /.

## A SORTING PATCH FOR RAE

Jean M. Cyr, 29 Greenboro Crescent, Ottawa, Ontario, Canada, K1T 1W5, submitted a very nice program called USER PATCH FOR RAE-1. It provides a better interface to a TTY, and has other nice features. The complete, fully commented, version is being sent to RAE NOTES subscribers. Published here is an abbreviated version of that portion of his program which permits the printing of an alphabetically sorted Label File. He has not yet found a way to suppress the printing of the unsorted file. Can anyone help him? It might also be nice to provide another patch to permit the printing of a numerically sorted Label File.

```
>ASSEMBLE LIST

                0010 ;SORTING PATCH FOR RAE-1
                0020 ;PORTION OF USER PATCH FOR RAE-1
                0025 ;
                0030 ;JEAN M. CYR
                0040 ;29 GREENBORO CRESCENT
                0050 ;OTTAWA,ONTARIO
                0060 ;CANADA K1T 1W5
                0070 ;
                0071 ;Editor's Note: To save space
                0072 ;in the listing, printing of
                0073 ;the Macro Expansions was sup-
                0074 ;pressed. These can be found in
                0075 ;the object code verification
                0076 ;below
                0077 ;
                0085 LBLSIZ     .DE $500
                0200 LBL        .DE $0104
                0210 BUF        .DE $00C8
                0325 SCRN       .DE $FE
                0330 SCRC       .DE $FC
                0350 DUMMY      .DE 0
                0460 !!!MW      .MD (FROM TO)
                0465            LOAD (FROM)
                0470            STORE (TO)
                0475            LOAD (FROM+1)
                0480            STORE (TO+1)
                0485            .ME
                0490 ;
                0495 !!!MT      .MD (FROM TO)
                0500            LDY $0
                0505 ...MT1     LDA (FROM),Y
                0510            STA (TO),Y
                0515            BMI ...MT3
                0520 ...MT2     INY
                0525            BNE ...MT1
                0530 ...MT3     CPY $2
                0535            BCC ...MT2
                0540            .ME
                0545 ;
                0550 !!!STORE   .MD (ADR)
                0555            SET DUMMY = ADR
                0560            IFM DUMMY
                0565            SET DUMMY = $100
                0570            ***
                0575            IFP $FF-DUMMY
                0580            STA *ADR
                0585            ***
                0590            IFP DUMMY-$100
                0595            STA ADR
                0600            ***
                0605            .ME
                0610 ;
                0615 !!!LOAD    .MD (ADR)
                0620            SET DUMMY=ADR
                0625            IFM DUMMY
                0635            ***
                0630            SET DUMMY=$100
                0640            IFP $FF-DUMMY
                0645            LDA *ADR
                0650            ***
                0655            IFP DUMMY-$100
                0660            LDA ADR
                0665            ***
                0670            .ME
                0690            .EC
                0695            .BA $1F71
                0700            .OS
1F71- 4C 03 B0  0710 USEREXIT   JMP $B003
                0950 SORT       .DE $1F74
                0955 SORTLBLS   MW (LBL SCRN)
                0960 NEXTLBL    MW (SCRN SCRC)
1F88- A0 02     0965            LDY #2
1F8A- B1 FC     0970 NEXTCHAR   LDA (SCRC),Y
1F8C- 30 03     0975            BMI COMPSTRING
1F8E- C8        0980            INY
1F8F- D0 F9     0985            BNE NEXTCHAR
1F91- 20 F3 1F  0990 COMPSTRING JSR ADRNEXT
1F94- A0 02     0995            LDY #2
1F96- B1 FE     1000            LDA (SCRN),Y
1F98- F0 D7     1005            BEQ USEREXIT
1F9A- B1 FC     1010 COMPCHAR   LDA (SCRC),Y
1F9C- 51 FE     1015            EOR (SCRN),Y
1F9E- 30 0B     1020            BMI EOS
1FA0- B1 FE     1025            LDA (SCRN),Y
1FA2- D1 FC     1030            CMP (SCRC),Y
1FA4- 90 1B     1035            BCC XCHANGE
1FA6- D0 D6     1040            BNE NEXTLBL
1FA8- C8        1045            INY
1FA9- D0 EF     1050            BNE COMPCHAR
1FAB- B1 FE     1055 EOS        LDA (SCRN),Y
1FAD- 10 0A     1060            BPL EOSC
1FAF- 29 7F     1065            AND #$7F
1FB1- D1 FC     1070            CMP (SCRC),Y
1FB3- F0 0C     1075            BEQ XCHANGE
1FB5- 90 0A     1080 HIGHLOW    BCC XCHANGE
1FB7- B0 C5     1085            BCS NEXTLBL
1FB9- 09 80     1090 EOSC       ORA #$80
1FBB- D1 FC     1095            CMP (SCRC),Y
1FBD- F0 BF     1100            BEQ NEXTLBL
1FBF- D0 F4     1105            BNE HIGHLOW
                1110 XCHANGE    MT (SCRC BUF)
                1115            MT (SCRN SCRC)
1FDF- 20 F3 1F  1120            JSR ADRNEXT
                1125            MT (BUF SCRN)
1FF1- B0 81     1130            BCS SORTLBLS
                1135 ;
1FF3- 98        1140 ADRNEXT    TYA
1FF4- 38        1145            SEC
1FF5- 65 FC     1150            ADC *SCRC
1FF7- 85 FE     1155            STA *SCRN
1FF9- A5 FD     1160            LDA *SCRC+1
1FFB- 69 00     1165            ADC #0
1FFD- 85 FF     1170            STA *SCRN+1
1FFF- 60        1175            RTS
                1180
                1185            .EN
```

After the unsorted Label File is listed, enter >RUn SORT; then, after the Warm Start re-entry message and prompt, enter >LAbels, to get a listing of the alphabetically sorted Label File.

```
1F70 00 4C 03 B0 AD 04 01 85,36
1F78 FE AD 05 01 85 FF AD FE,16
1F80 00 85 FC AD FF 00 85 FD,C5
1F88 A0 02 B1 FC 30 03 C8 D0,DF
1F90 F9 20 F3 1F A0 02 B1 FE,5B
1F98 F0 D7 B1 FC 51 FE 30 0B,59
1FA0 B1 FE D1 FC 90 1B D0 D6,26
1FA8 C8 D0 EF B1 FE 10 0A 29,9F
1FB0 7F D1 FC F0 0C 90 0A B0,31
1FB8 C5 09 80 D1 FC F0 BF D0,CB
1FC0 F4 A0 00 B1 FC 91 C8 30,95
1FC8 03 C8 D0 F7 C0 02 90 F9,72
1FD0 A0 00 B1 FE 91 FC 30 03,81
1FD8 C8 D0 F7 C0 02 90 F9 20,7B
1FE0 F3 1F A0 00 B1 C8 91 FE,35
1FE8 30 03 C8 D0 F7 C0 02 90,49
1FF0 F9 B0 81 98 38 65 FC 85,29
1FF8 FE A5 FD 69 00 85 FF 60,16
 5316
.
```

## SOME GAMES (AND MORE) FOR THE SYM-1 WITH KTM-2/80

Many readers have asked, "Game programs, please?"; nearly as many have said, "No games, thank you!". I think we can please both groups of readers with the programs we shall describe, because, while I incline towards the "no game" group, myself, I did find these particular games fascinating. The story begins with my receiving a program listing, in BASIC, from Jack Gieryic, for publication. Not wishing to publish a program without testing it first, even though I know the author well from having read many of his published articles, I asked Jack if he would mind sending me a cassette dump, in place of the listing. The thought of spending many hours keying in and debugging a BASIC listing is not my idea of a great time. Well, Jack sent six program packages on cassette: three games, two utilities, and a graphics demonstration package (GDP-1). GDP-1 is published here.

All six require 4 K of RAM and a KTM-2/80 (no, the programs will not convert easily to the 40 column KTM-2). Jack's skill with graphics is impressive. Jack calls his product line JACK BUILT PROGRAMS. No. 1 is a one-person game, DEPTH CHARGE, which requires a three dimensional search, and presents a simulated sonar-type display. Nos. 2 and 3 are two-person games. Tom Gettys would rather play against the computer, but I rather like the idea of having a human companion around to share the pleasures of the computer with. No. 2 is the well-known OTHELLO, which I had never played before, but learned quickly enough. No. 3 is an adaption of the old TV Game Show CONCENTRATION, again well implemented by Jack.

My favorite, because it was not a game requiring personal competition, but provides entertainment, was No. 4, the Graphics Demonstration Package, which also includes an example of Computer Assisted Instruction (CAI). It asks you to enter your name, then asks you to make a selection from a "menu" (see listing). "The Square Story" is a teaching program. "Football Field" is a drawing of a football field. The others are dynamic graphic shows. What Martin Gardner has said about music (see elsewhere in this issue) applies equally well to art. To paraphrase him, Art (with a capital A) and music, to be interesting, must consist of the proper mixture of the "expected" and the "unexpected". The purely random (incoherent) patterns are dull, as are the totally regular (coherent) ones. "Ink Spots" illustrate the principle well. The patterns are reminiscent of the Rorschach (Ink Spot) Personality Test, except that the bilateral symmetry is missing (must ask Jack to include that feature in an updated version).

No. 5, PLOT, is a multiple mathematical graph drawing utility, and No. 6, BAR, is a very versatile Bar Chart (vertical bars) drawing utility. If you have the KTM-2/80 you will enjoy these programs; if you have the money to spend on "luxury" items, like the KTM-2/80, you probably don't have the time to key in long programs. Fortunately, all of the JACK BUILT PROGRAMS are available on cassette. See the back page of this issue for ordering information. A prelinary version of the GRAPHICS DEMONSTRATION PACKAGE is printed here for your information. It is definitely convertible to 40 columns. See what I meant about keying in a long BASIC program?

```
1 E=27:S=124:LIM=2000:TH=32:GOTO100
2 PRINTCHR$(E)+"=";:RETURN
3 PRINTCHR$(E)+"R":RETURN
4 PRINTCHR$(E)+"G":RETURN
5 PRINTCHR$(E)+CHR$(114);:RETURN
6 PRINTCHR$(E)+CHR$(103);:RETURN
7 GOSUB2:PRINTCHR$(Y+TH)+CHR$(X+TH)+CHR$(S):RETURN
8 FORY=YSTOYS+YL:GOSUB7:NEXT:RETURN
9 FORX=XSTOXS+XL:GOSUB7:NEXT:RETURN
```



```
10 PRINTCHR$(E)+"H"+CHR$(E)+"J":FORA=1TO5:NEXT:RETURN
11 X=INT(77*RND(1)):Y=INT(23*RND(1)):GOSUB7:RETURN
12 GOSUB5:GOSUB6:S=124:RETURN
13 GOSUB10:GOSUB3:GOSUB4:RETURN
14 YL=INT(21*RND(1)):IFYL<3THEN14
15 RETURN
16 GOSUB3:GOSUB4:GOSUB20:GOSUB25:RETURN
17 FORA=1TO5000:NEXT:RETURN
18 FORA=1TO2000:NEXT:RETURN
19 S=63+INT(64*RND(1)):RETURN
20 XS=INT((79-XL)*RND(1)):YS=INT((21-YL)*RND(1)):RETURN
21 X=XS:GOSUB8:Y=YS:GOSUB9:RETURN
22 X=XS:GOSUB8:Y=YS+YL:GOSUB9:RETURN
23 Y=YS:GOSUB9:X=XS+XL:GOSUB8:RETURN
24 Y=YS+YL:GOSUB9:X=XS+XL:GOSUB8:RETURN
25 GOSUB22:GOSUB23:GOSUB5:GOSUB6:RETURN
26 Y=YS+YL:FORX=XSTOXS+XL:GOSUB7:Y=Y-1:NEXT:RETURN
27 Y=YS:FORX=XSTOXS+XL:GOSUB7:Y=Y+1:NEXT:RETURN
28 PRINTCHR$(Y+TH)+CHR$(X+TH);A:RETURN
100 GOSUB10:GOSUB2:PRINT"(*HI. I AM YOUR COMPUTER.  I WOULD LIKE TO "
102 GOSUB2:PRINT")*KNOW WHO YOU ARE.  PLEASE TYPE YOUR NAME"
104 GOSUB2:PRINT"*#AND THEN HIT THE KEY MARKED RETURN.
106 GOSUB2:PRINT"-4";:INPUT"";N$:GOSUB10
108 GOSUB2:PRINT"!&HERE IS A LIST OF THINGS I CAN DO FOR YOU ";N$;"."
110 GOSUB2:PRINT"$&TYPE THE NUMBER OF YOUR CHOICE AND THEN HIT "
112 GOSUB2:PRINT"$&THE RETURN KEY.  I'M WAITING FOR YOU, ";N$;"."
114 GOSUB2:PRINT"&-1  THE SQUARE STORY":GOSUB2:PRINT"'-2  RECTANGLES"
116 GOSUB2:PRINT"(-3  TRIANGLES":GOSUB2:PRINT")-4  DIAMONDS"
118 GOSUB2:PRINT"*-5  RANDOM":GOSUB2:PRINT"+-6  RANDOM GRAPHICS""
120 GOSUB2:PRINT",-7  INVERSE RANDOM GRAPHICS"
122 GOSUB2:PRINT"--8  INK SPOTS":GOSUB2:PRINT".-9  RANDOM INK SPOTS"
124 GOSUB2:PRINT"/-10 FOOTBALL FIELD"
135 PRINT"":INPUT"YOUR CHOICE IS ";B:GOSUB10
137 IFB<1THEN108
139 IFB>10THEN108
150 ONBGOSUB1000,2000,900,200,700,800,800,400,400,500
152 GOSUB17:GOSUB10:GOTO108
199 END
200 GOSUB3:GOSUB4:FORK=1TO10:GOSUB14:XL=YL:GOSUB20:GOSUB19
205 YL=1+INT(YL/2):XL=YL:GOSUB26:YS=YS+YL:GOSUB27:XS=XS+XL:YS=YS-YL
210 GOSUB27:YS=YS+YL:GOSUB26:NEXTK:GOSUB5:GOSUB6:RETURN
300 GOSUB14:XL=2*YL:GOSUB16:RETURN
400 GOSUB3:GOSUB4:GOSUB19:X=40:Y=12
402 FORA=1TO3:A(A-1)=A-2:B(A-1)=A-2:NEXT
410 FORK=1TO500:IFB=9THENGOSUB19
412 A=INT(3*RND(1)):IFA=3THEN412
414 L=INT(3*RND(1)):IFL=3THEN414
416 IFA(A)<>0THEN440
417 IFB(L)=0THEN412
440 X=X+A(A):IFX<2THENX=77
442 IFX>77THENX=2
444 Y=Y+B(L):IFY=-1THENY=22
446 IFY=23THENY=0
448 GOSUB7:X=X+A(A):GOSUB7:NEXT:GOSUB5:GOSUB6:RETURN
500 S=97:XS=10:YS=10:B=10:GOSUB3:GOSUB4:FORX=XS+4TOXS+48STEP4:GOSUB590
512 NEXT:S=126:FORX=XSTOXS+3:GOSUB590:NEXT:FORX=XS+44TOXS+47:GOSUB590
515 NEXT:S=113:Y=YS-1:FORX=XSTOXS+47:GOSUB7:NEXT:GOSUB5:S=103:X=XS-1
565 GOSUB590:X=XS+43:GOSUB590:S=119:Y=YS+B+1:FORX=XSTOXS+47:GOSUB7:NEXT
572 GOSUB5:GOSUB6:Y=YS-2:A=0:FORX=XS+2TOXS+22STEP4:GOSUB2:GOSUB28:A=A+1
0
575 NEXT:A=50:FORX=XTOXS+42STEP4:GOSUB2:A=A-10:GOSUB28:NEXT:RETURN
590 FORY=YSTOYS+B:GOSUB7:NEXTY:RETURN
600 GOSUB14:XL=1+INT(75*RND(1)):GOSUB16:RETURN
700 GOSUB10:IFB=5THENGOSUB3
```

```
 710 S=63+INT(64*RND(1)):GOSUB4:FORA=1TO2000:X=INT(77*RND(1))
 715 Y=INT(23*RND(1)):GOSUB7:NEXT:GOSUB5:GOSUB6:RETURN
 800 GOSUB13:IFB=6THENGOSUB5
 810 FORA=1TO2000:S=63+INT(64*RND(1)):GOSUB11:NEXT:GOSUB12:RETURN
 900 GOSUB3:GOSUB4:FORK=1TO10:GOSUB14:XL=YL:GOSUB20:GOSUB19
 905 B=INT(5*RND(1)):IFB=5THEN905
 910 IFB<1THEN905
 915 ONBGOSUB21,22,23,24
 920 ONBGOSUB26,27,27,26
 925 NEXTK:GOSUB5:GOSUB6:RETURN
 1000 GOSUB2:PRINT"**A SQUARE IS A SPECIAL CASE OF A PARALLELOGRAM.  ALL
 *
 1010 GOSUB2:PRINT"$*FOUR SIDES ARE EQUAL IN LENGTH AND ALL FOUR ANGLES
ARE *
 1020 GOSUB2:PRINT"%*RIGHT ANGLES (90 DEGREES).  I WILL NOW DRAW AN EXAM
PLE *
 1030 GOSUB2:PRINT"&*FOR YOU ";N$;".":GOSUB17
 1040 S=124:YL=12:XL=24:YS=8:XS=3:GOSUB3:GOSUB4:GOSUB25:GOSUB17
 1043 GOSUB2:PRINT")ATHE SMALL SQUARE IN THE CORNER"
 1044 GOSUB2:PRINT"*AMEANS THIS IS A RIGHT ANGLE.":GOSUB18
 1045 GOSUB3:GOSUB4:GOSUB2:PRINT")%"+CHR$(97):GOSUB2:PRINT")$"+CHR$(113)
 1046 GOSUB5:GOSUB6:GOSUB17
 1048 GOSUB2:PRINT",ALOOK WHERE THE ARROW IS POINTING."
 1050 GOSUB18:GOSUB4:GOSUB2:PRINT"*%"+CHR$(103)
 1052 GOSUB3:GOSUB2:PRINT")&"+CHR$(113)+CHR$(113)
 1053 GOSUB2:PRINT"*&"+CHR$(92):GOSUB2:PRINT"+'"+CHR$(92):GOSUB5:GOSUB6:
GOSUB17
 1054 GOSUB2:PRINT".AI WILL NOW DRAW SOME SQUARES FOR YOU, ";N$;"."
 1056 GOSUB17:FORL=1TO10:GOSUB10:GOSUB19:GOSUB300:GOSUB18:NEXT
 1060 FORL=1TO10:GOSUB19:GOSUB300:NEXT:RETURN
 2000 S=63+INT(64*RND(1)):FORL=1TO10:GOSUB600:NEXT:RETURN
OK
```

Here is what a partial RUN looks like on a printing terminal. The "=" sign (which followed a non-printing "ESC") signals the KTM-2 that the following two characters are absolute Y,X cursor coordinates. The "HJ" seems to be a residue from the screen-clear operation.

```
=(*HI. I AM YOUR COMPUTER.  I WOULD LIKE TO
=)*KNOW WHO YOU ARE.  PLEASE TYPE YOUR NAME
=**AND THEN HIT THE KEY MARKED RETURN.
=-4
   HJ
=!&HERE IS A LIST OF THINGS I CAN DO FOR YOU LUX.
=$&TYPE THE NUMBER OF YOUR CHOICE AND THEN HIT
=$&THE RETURN KEY.  I'M WAITING FOR YOU, LUX.
=&-1  THE SQUARE STORY
='-2  RECTANGLES
=(-3  TRIANGLES
=)-4  DIAMONDS
=*-5  RANDOM
=+-6  RANDOM GRAPHICS
=,-7  INVERSE RANDOM GRAPHICS
=--8  INK SPOTS
=.-9  RANDOM INK SPOTS
=/-10 FOOTBALL FIELD

YOUR CHOICE IS
              HJ
=$*A SQUARE IS A SPECIAL CASE OF A PARALLELOGRAM.  ALL
=$*FOUR SIDES ARE EQUAL IN LENGTH AND ALL FOUR ANGLES ARE
=%*RIGHT ANGLES (90 DEGREES).  I WILL NOW DRAW AN EXAMPLE
=&*FOR YOU LUX.
```



## MICRO TECHNOLOGY UNLIMITED SOFTWARE FOR THE SYM-1

Micro Technology Unlimited has, for many years, marketed an 8 Bit DAC Board, K-1002, for music generation, and the 8K RAM Visible Memory Board, K-1008, for high resolution graphics. These are available from MTU, together with excellent manuals, K-1002-1L, and K-1008-1L, respectively, written for the KIM-1. The two manuals, together with SYM-1 supplements, and the 8 Bit DAC Board may also be obtained through the SYM-1 Users' Group. The SYM-1 Supplement to the K-1002-1L Manual, "8 Bit Digital Music Software", is now available, and the SYM-1 Supplement to the K-1008-1L Manual, "Graphic/Text Subroutines and Demonstrations", will be available 1 June 1980. In addition, the Users' Group will have available SYM-1 readable object code, on cassettes, for each of these items, relocated to avoid any pages 0 and 1 conflicts. MTU has arranged for the Users' Group to adapt, debug, market, and support the SYM-1 versions of their software products.

## HARDWARE RECOMMENDATION

One of the problems with a "component" system like SYM, as opposed to a "packaged" system like the Apple II, is where to plug in all of the power cords. There's the power supply, the monitor, the recorder power supply, the scope, the modem, the printer, the soldering iron, etc. To make things even worse, we have two systems up and running, and the dual floppy disk system is temporarily (perhaps indefinitely!) using its own pair of power supplies. I can't even begin to count the number of power cords. A more serious problem, however, was the tendency of the oscilloscope to completely "crash" the system whenever it (the scope) was turned on or off. Thus the scope had to be turned on first, and left running as long as the system was in use.

Both problems were solved with products of Electronic Specialists, Inc., 171 South Main Street, Natick, MA 01760 (write for their catalog). Their Isolator ISO-2, at $55, provides two groups of three 3-prong sockets, each group filter-isolated from the other, and from the power line; their ISO-1 (same price) provides only 3 sockets but these are isolated from each other. You can get either with a 15 A circuit breaker for $62, or a circuit breaker and switch/pilot light for $67. Their ISO-3, more expensive, is similar to the ISO-1, but provides heavier filtering, for more severe noise environments. My assembly of power cords is now much neater, and things no longer interact when switched on or off.

## WHITE AND BROWN MUSIC

Martin Gardner, in the Mathematical Games section of Scientific American, April, 1978, has some interesting words to say about computer generated music. By this he means music actually "composed" by the computer:

> "It is commonplace in musical criticism to say that we enjoy good music because it offers a mixture of order and surprise. How could it be otherwise?"

He defines "white" music as being completely random, i.e., complete surprise, and "brown" music as being a mixture of order and surprise. An example of complete order is the simple musical scale repeated over and over. Both white music and the scales are dull. He offers several examples of brown music, one of which is called 1/f music. These sound surprisingly "good". When I first read the article, I programmed the examples for my KIM. Unfortunately the listings have been lost. Mr. Gardner describes the process for generating brown music so well, that you should have no trouble writing the program yourself, either in Assembly or BASIC. You will not need a DAC system, even the simplest timed loop, or VIA timer, square wave generator will be adequate for the purpose. You should have much fun with this one!

## HI-DENSITY PLOTTING WITH THE KTM-2

by: Bill Gowans
254 Old Orchard Grove
Toronto,Ontario M5M2E5
(416) 488-3456

### DESCRIPTION

This routine effectively quadruples the KTM-2/80 graphics density by mapping a virtual 48X160 screen onto the real 24X80 screen. This allows 7,680 individual points to be controlled and tested,giving the KTM-2 a respectable graphics capability for most applications.The routine was written to interface with the KTM-2/80 and BAS-1,however only minor changes are needed for KTM-2/40 or Assembler interface.In addition,the general technique used can be applied to other video terminals having capabilities similar to the KTM-2.

The quad density is achieved by creating and maintaining an internal memory map of the KTM-2 screen.Each of the 1,920 (24X80) character positions is considered as consisting of 4 separate elements (pixels). Thus we can have 16 possible combinations of the 4 pixels.The KTM-2 character set contains graphic characters for each of the 16 pixel combinations,all that is needed is a way to select the proper one.Since there are 4 pixels,we can assign a 4-bit code with each bit representing a particular pixel.This gives us a series of 4-bit codes with a range from 0-15 which can be used to index a table containing the correct code to display the graphic character required.Setting or resetting a pixel merely involves turning the appropriate bit on or off in the 4-bit code and using the resulting value to access the new graphic character.

The use of a 4-bit code also allows us to compress the 1,920 character map into 960 bytes by combining two 4-bit (Nybble) codes into one Byte. This complicates the code slightly but the resultant saving in memory is well worth it.To simplify the accessing of the proper screen map byte a table of pointers was created (RTAB) to allow direct indexing to the correct row.This in conjunction with the column allow us to access the map bytes without having to perform multiply operations.(Note-if you have a KTM-2/40,the RTAB entry increment can be changed from +40 to +20 and the 'BSS' following the label 'MAP:' can be reduced to 479)

One problem in using the 16 graphic characters for pixel display is that they can not all be displayed in the same mode (some require normal mode while others require reverse mode).The solution to this was to allocate one bit in the Pixel Map Table (CHAR) entry to indicate the mode that the KTM-2 had to be in for proper display.The rightmost bit was used for this purpose (0=normal,1=reverse) leaving the leftmost 7 bits to code the graphic character.An internal mode indicator (MODE) is used to keep track of the KTM-2's current mode (0=normal,-1=reverse).When the mode bit and mode indicator differ,the KTM-2 mode is changed prior to displaying the character.
Total memory required is 1241 bytes for the KTM-2/80 version and 761 bytes for the KTM-2/40.This allows both plot and tris routines to be used in a 4K system with approximately 2500 bytes left for BAS-1 use.



### FUNCTIONS

Four functions are provided by this routine:

CLEAR - This clears the KTM-2 screen and the internal screen map.The mode indicator is reset to normal mode (zero).

SET - The referenced pixel will be turned 'ON' in the internal map and the appropriate graphic character displayed.

RESET - The referenced pixel will be turned 'OFF' in the internal map and the appropriate graphic character displayed.

TEST - The referenced pixel in the internal map will be tested and a value returned representing its state (0='OFF',1='ON').

'CLEAR' requires no parameters while the other three calls require that a Virtual Row (0-47) be passed in the A-register and a Virtual Column (0-159) passed in the Y-register (Note-for KTM-2/40 the Virtual Column can only be from 0-79).This would seem to be a problem as the BAS-1 'USR' function only allows one parameter to be passed in the A-Y register pair (others can be passed on the stack).We can slip two parameters past BAS-1 for the price of one if we structure our call as follows:

```
USR(A,256*R+C)
```

where: A = Address of Routine
R = Virtual Row (0-47)
C = Virtual Column (0-159) **(0-79) for KTM-2/40**

Multiplying by 256 effectively shifts the Virtual Row into the A-register while the Virtual Column remains in the Y-register.
If the 4 entry point addresses (CLEAR,SET,TEST and RESET) are equated to the variables C,S,T,R,and the Virtual Row/Column to the variables Y and X then the 4 calls can be illustrated as follows:

```
CLEAR -     Q = USR(C,0)

SET   -     Q = USR(S,256*Y+X)

RESET -     Q = USR(R,256*Y+X)

TEST  -     Q = USR(T,256*Y+X)
```

**Note-to use an Assembler interface,the 'JMP BSRET' must be replaced with 'RTS'.

### USAGE

- Prior to beginning a plot,the 'CLEAR' function should be invoked and the KTM-2 placed in Graphics/Normal mode.

- Your program should not change the KTM-2 mode (Normal/Reverse) as it will cause unpredictable results on the plot.

- After plotting has been completed your program must reset the KTM-2 mode to whatever is required as the final state is unpredictable.

```
                        1 ;*****************************************************
                        2 ;*                                                   *
                        3 ;*     HI-DENSITY PLOT ROUTINE FOR THE KTM-2         *
                        4 ;*                                                   *
                        5 ;*                BY : BILL GOWANS                   *
                        6 ;*                                                   *
                        7 ;*****************************************************
                        8         ORG   $19EE
                        9 ;*****************************************************
                       10 ;*                ZERO PAGE WORK LOCATIONS           *
                       11 ;*****************************************************
                       12 ZWORK:  EPZ   $FE
                       13 RPTR:   EPZ   $EE
                       14 ;*****************************************************
                       15 ;*                  PROGRAM VARIABLES                *
                       16 ;*****************************************************
                       17 ROW:    BSS   1
                       18 COL:    BSS   1
                       19 FLAG:   BSS   1
                       20 MODE:   BSS   1
                       21 CINDX:  BSS   1
                       22 ;*****************************************************
                       23 ;*                  EXTERNAL ROUTINES                *
                       24 ;*****************************************************
                       25 SEND:   EQU   $8A47
                       26 BSRET:  EQU   $D14C
                       27 WPON:   EQU   $8B9C
                       28 WPOFF:  EQU   $8B86
                       29 CLRM:   EQU   $8723
                       30 ;*****************************************************
                       31 ;*                  PIXEL MASK TABLE                 *
                       32 ;*****************************************************
                       33 MASK:   EQU   *
19F3:01 02 04          34         BYTE  $01,$02,$04,$08
19F6:08
19F7:10 20 40          35         BYTE  $10,$20,$40,$80
19FA:80
                       36 ;*****************************************************
                       37 ;*              SCREEN ROW POINTER TABLE             *
                       38 ;*****************************************************
                       39 RTAB:   EQU   *
19FB:2B 1A             40         WORD  MAP,MAP+40,MAP+80,MAP+120
19FD:53 1A
19FF:7B 1A
1A01:A3 1A
1A03:CB 1A             41         WORD  MAP+160,MAP+200,MAP+240,MAP+280
1A05:F3 1A
1A07:1B 1B
1A09:43 1B
1A0B:6B 1B             42         WORD  MAP+320,MAP+360,MAP+400,MAP+440
1A0D:93 1B
1A0F:BB 1B
1A11:E3 1B
1A13:0B 1C             43         WORD  MAP+480,MAP+520,MAP+560,MAP+600
1A15:33 1C
1A17:5B 1C
1A19:83 1C
1A1B:AB 1C             44         WORD  MAP+640,MAP+680,MAP+720,MAP+760
1A1D:D3 1C
1A1F:FB 1C
```



```
1A21:23 1D
1A23:4B 1D             45         WORD  MAP+800,MAP+840,MAP+880,MAP+920
1A25:73 1D
1A27:9B 1D
1A29:C3 1D
                       46 ;*****************************************************
                       47 ;*                    SCREEN MAP                     *
                       48 ;*****************************************************
                       49 MAP:    EQU   *
                       50         BSS   959
                       51 MAPE:   BSS   1
                       52 ;*****************************************************
                       53 ;*              PIXEL CHARACTER MAP TABLE            *
                       54 ;*****************************************************
                       55 CHAR:   EQU   *
1DEB:C1 99 97          56         BYTE  $C1,$99,$97,$E8,$95,$C9,$BD,$92
1DEE:E8 95 C9
1DF1:BD 92
1DF3:93 BC C8          57         BYTE  $93,$BC,$C8,$94,$E9,$96,$98,$F9
1DF6:94 E9 96
1DF9:98 F9
                       58 ;*****************************************************
                       59 ;*                    MAIN PROGRAM                   *
                       60 ;*                                                   *
                       61 ;*         THERE ARE 4 ENTRY POINTS IN THE           *
                       62 ;*         PROGRAM:                                  *
                       63 ;*                                                   *
                       64 ;*                                                   *
                       65 ;*         "CLEAR" - CLEARS THE KTM-2 AND            *
                       66 ;*                   INTERNAL SCREEN MAP.            *
                       67 ;*                                                   *
                       68 ;*         "SET" - TURNS ON THE REFERENCED           *
                       69 ;*                 PIXEL.                            *
                       70 ;*                                                   *
                       71 ;*         "RESET" - TURNS OFF THE PIXEL             *
                       72 ;*                                                   *
                       73 ;*         "TEST" - TESTS STATE OF PIXEL             *
                       74 ;*                  AND RETURNS VALUE                *
                       75 ;*                  (0=OFF,1=ON)                     *
                       76 ;*****************************************************
1DFB:A9 0C             77 CLEAR:  LDA   #$0C       ;LOAD SCREEN CLEAR CHAR
1DFD:20 47 8A          78         JSR   SEND       ;SEND IT OUT TO KTM-2
1E00:20 86 8B          79         JSR   WPOFF      ;TURN OFF WRITE PROTECT
1E03:A9 2B             80         LDA   #<MAP      ;SETUP THE
1E05:85 FE             81         STA   ZWORK      ; LOW AND
1E07:A9 1A             82         LDA   #>MAP      ;  HIGH ADDR
1E09:85 FF             83         STA   ZWORK+1    ;   IN MONITOR
1E0B:A9 EA             84         LDA   #<MAPE     ;    AND THEN
1E0D:8D 4A A6          85         STA   $A64A      ;     CLEAR MAP
1E10:A9 1D             86         LDA   #>MAPE     ;      AREA TO
1E12:8D 4B A6          87         STA   $A64B      ;       ALL ZEROS
1E15:A9 00             88         LDA   #$00       ;RESET MODE FLAG
1E17:8D F1 19          89         STA   MODE       ; TO INDICATE NORMAL MODE
1E1A:20 23 87          90         JSR   CLRM       ;USE MONITOR ROUTINE TO CLEAR
1E1D:4C 9C 8B          91         JMP   WPON       ;TURN WRITE PROTECT BACK ON
1E20:A2 00             92 RESET:  LDX   #$00       ;FLAG(0) = RESET PIXEL
1E22:F0 06             93         BEQ   PLOT       ;JUMP TO MAIN ROUTINE
1E24:A2 80             94 TEST:   LDX   #$80       ;FLAG(-) = TEST PIXEL
1E26:30 02             95         BMI   PLOT       ;JUMP TO MAIN ROUTINE
1E28:A2 40             96 SET:    LDX   #$40       ;FLAG(40) = SET PIXEL
```