

circulation 1060

YEP, YOU GUESSED IT, WE MOVED AGAIN! This time to a "permanent" P.O. Box.

Please note our new address-

KIM-1/6502 USER NOTES
P.O. Box 33077
North Royalton, Ohio 44133

new phone number also! but not yet known.

RENEWAL TIME IS HERE!!!!!!

Your response has been so gratifying that I've decided to go for 6 more!

When extending your subscription, please mark RENEWAL on the envelope and your check.

The new rates for #7 - #12 are: For U.S. & Canada - \$5.00 (includes 1st class postage)
International - \$10.00 (includes air mail postage and envelope).

ARTICLE CONTRIBUTORS PLEASE NOTE.....To alleviate possible typographical errors, please submit typed originals, single-spaced on white bond with 8 inch wide columns.

CALCULATOR INTERFACE information:

I neglected to mention in issue #4 that all keystroke data entries (starting at \$0300) should be preceded by two (2) CA/CE commands (\$B4) to properly initialize the calc. chip. EXAMPLE: suppose you wanted to add 3 and 6---- at address \$0300 you'd enter B4 B4 31 12 61 62 FF.

FROM THE FACTORY!

Arnie Karush, Commodore Business Machine Co., (new owners of MOS Technology), has passed along the following interesting bits of info-

- Production is stopped on the KIM-2, and 3 memory boards. These boards will still be available on a special order, cash in advance basis. A new memory board will be introduced around August at a lower cost per byte than KIM 2 or 3.
- The KIM-4 motherboard production is also halted awaiting some design changes and will be re-introduced around August.
- The KIM-5 ROM board and the ROM set (Assembler, Editor, Mathpack, etc.) are being postponed indefinitely.
- COMPUTER STORES-The Commodore Business Machine Company wants more computer stores to stock KIM-1's, so they have announced a better markup margin for dealers at smaller quantities than before. (Check with C.B.M. for more details).
- Around 7500 KIM's are purportedly in the field at this time and Commodore states that they are working on increasing KIM production to meet demand. (I guess they just can't make 'em fast enough!)

P. 1

A LOW-COST RTTY TERMINAL UNIT (send and receive) was featured in the May '77 issue of Z2 Magazine. It utilizes a Digital Group cassette interface board and looks like a very reasonable approach to bridging the gap from single-sideband gear to your computer for not too many bucks. Anyone working on an RTTY program for KIM?

KIM-1/6502 SOFTWARE

Got the latest flyer from 6502 Program Exchange (2920 Moana, Reno, Nev., 89509). They say that their FOCAL (FCL-65) package is now available for KIM, TIM or any 6502 system. The flyer goes on to say that FCL-65 takes a little over 4K of memory, comes on paper tape, and that the complete source listing is available. The EXCHANGE also listed several games and a Scientific Math Package for FCL-65. They want \$.50 for their complete program list, and I can recommend them.

ARESCO (314 Second Ave., Haddon Hts., N.J. 08035) lists several programs available for KIM, TIM, etc.. on paper tape or KIM cassette.

The flyer lists FOCAL (\$40) a 2.5K resident assembler (\$30) and XPL6 (a COMPILER) for (\$40). According to ARESCO, all program packages include complete source listings as well as object code and user manuals. They want \$2.00 for a complete information package.

MICRO-SOFTWARE sent me a card announcing immediate availability of an MOS compatible assembler/editor which resides in just over 2K. They say that it is available on KIM cassette and KIM or TIM paper tape starting at address \$2000. The pricing information was a bit confusing so I'd suggest contacting them for more info:

MICRO-SOFTWARE SPECIALISTS, P.O. Box 3292, E.T. Station, Commerce, TEX 75428

To all user's of MICROCHESS, Please note the change of address to MICROCHESS (KIM-1), 27 Firstbrooke Road, Toronto, Ontario, Canada, M9E 2L2. Copies of MICROCHESS are still available at \$10.00 from the above address. For paper tape add \$1.00, for cassette add \$3.00.

WOW!!! LONG LIVE THE 6502!!!!!!!!!!

MORE KIM STUFF!

Gary Mayhak sent along a very neat LED display cover for KIM. It's a red plexiglass piece that fits over and around the displays, makes them easier to read, and dresses up KIM in process.

If you'd like to spiff up your KIM, send Gary an SASE and \$2.00 for one (or \$3.00 for 2). His address is 1347 Turret Drive, San Jose, Ca. 95131. I'd suggest sending him a self addressed stamped cassette box so there's no chance of the cover being re-configured in the mail.

"I have interfaced a T.I. 5050M PRINTING CALCULATOR to my KIM-1. The printer is 10 column numeric only but price is \$90.00. If any of your subscribers are interested, please have them contact me." David G. Rainey, 103 Roosevelt St., Grants, New Mexico 87020. Send S.A.S.E.

VIDEO DISPLAY MODULE & KIM EXPANSION application notes are available from Riverside Electronics (see ad in this issue). Four of the application notes (MVM-1,2,3,4) concern hardware & software considerations for their MVM-1024 video display module, and one application note (KIM1-1) outlines design ideas for KIM memory & I/O expansion. (This one's particularly useful). If you're looking for a memory-mapped video display module, the MVM-1024 deserves a look-see. These 5 application notes (MVM-1,2,3,4 & KIM1-1) are available from Riverside for \$1.00 (to cover postage). If you just want application note KIM1-1, it's free. They also have a package of software listings for KIM to drive the MVM-1024, available for \$3.00 (KIM -2). These application notes make interesting reading.

KIM-1 software and hardware new product announcements have often been found in ON-LINE, a classified ad newsletter dedicated to the computer hobbyist. It's published every three weeks and subscription rates are 18 issues/\$3.75, 36 issues/\$7.00 (for N. America). ON-LINE, Dave Beetle, Publisher, 24695 Santa Cruz Hwy., Los Gatos, CAL 95030

Jim Butterfield
14 Brooklyn Avenue
Toronto Ontario
M4M 2K5 Canada
September 1976

KIM-1 PROGRAM: BAGELS
(MASTER MIND/JOTTO)

Background

This game of guessing a "secret word" has appeared in many forms. BAGELS has appeared on many time-sharing systems and pocket calculators, usually as a digit-guessing game. JOTTO follows similar rules, but is concerned with guessing a five-letter (English) word. Recently, a game called MASTER MIND has been commercially marketed; the objective is to guess colours.

Starting the Program.

Load the program, and start at address 200 (AD 0 2 0 0).

The Play.

The computer has chosen four letters, all of which are A, B, C, D, E, or F. Letters may be repeated - for example, the computer's "secret" combination might be CACF or BBBB.

You get ten guesses. Each time you guess, the computer will tell you two things: how many letters are exactly correct (the right letter in the right place); and how many letters are correct, but in the wrong position.

For example, if the computer's secret combination is GEFB, and you guess BAFD, the two numbers will be 1 and 1 (the F matches exactly; the B matches but in the wrong place). These numbers will show on the right hand side of the display; the code you entered will appear on the left.

Make a note of your guesses and the computer's response. With a little mental work, you should be able to break the code exactly in seven or eight words. A correct guess will produce a response of 4 - 0. If you don't guess right in ten moves, the computer will give you the answer.

After a correct guess, or after the computer tells you the answer, it will start a new game (with a new secret code) the instant you touch a new key.

```

;          LINKAGES TO KIM MONITOR
;
KEYIN  = $1F40
GETKEY = $1F6A
TABLE  = $1FE7
PADD   = $1741
SBD    = $1742
SAD    = $1740
;
;          WORK AREAS
;
0000    SECRET  **+4  computer's secret code
0004    WINDOW  **+6  display window
000A    INPUT  **+4  player's input area
000E    EXACT   **+1  # of exact matches
000F    MATCH  **+1  # of other matches
0010    POINTR  **+1  digit being input
0011    MOD    **+1  divisor/delay flag
0012    RND    **+6  random number series
0018    COUNT  **+1  number of guesses left
```


BAGELS

```
027B E6 0E      INC EXACT
027D 8A          TXA      destroy input
027E 95 0A      NOTEX   STA INPUT,X
0280 A5 07      LDA WINDOW+3 has fourth digit arrived?
0282 F0 31      BEQ BUTT   ...no
0284 A0 03      LDY #3    ...yes, calculate matches
0286 B9 0A 00 STEP LDA INPUT,Y for each digit:
0289 29 18      AND #18   ..has it already been
028B F0 12      BEQ ON    matched?
028D B9 00 00   LDA SECRET,Y
0290 A2 03      LDX #3    if not, test
0292 D5 0A      LOOK    CMP INPUT,X ...against input
0294 F0 05      BEQ GOT
0296 CA          DEX
0297 10 F9      BPL LOOK
0299 30 04      BMI ON
029B E6 0F      GOT     INC MATCH increment counter
029D 16 0A      ASL INPUT,X and destroy input
029F 88          ON     DEY
02A0 10 E4      BPL STEP
02A2 A2 01      LDX #1    display counts
02A4 B4 0E      TRANS   LDY EXACT,X
02A6 B9 E7 1F   LDA TABLE,Y
02A9 95 08      STA WINDOW+4,X
02AB CA          DEX
02AC 10 F6      BPL TRANS
02AE 20 CE 02 DELAY JSR SHOW long pause for debounce
02B1 E6 0F      INC MATCH
02B3 D0 F9      BNE DELAY
02B5 20 CE 02 BUTT JSR SHOW wait for key release
02B8 D0 FB      BNE BUTT
02BA F0 9F      BEQ WAIT

;
; TEN GUESSES MADE - SHOW ANSWER
;
02BC A2 03      FINISH LDX #3
02BE B4 00      PIN2   LDY SECRET,X
02C0 B9 E7 1F   LDA TABLE,Y
02C3 95 04      STA WINDOW,X
02C5 CA          DEX
02C6 10 F6      BPL FIN2
02C8 A9 E3      LDA #E3   'square' flag
02CA 85 08      STA WINDOW+4
02CC D0 E0      BNE DELAY unconditional jump

;
; SUBROUTINE TO DISPLAY
; AND TEST KEYBOARD
;
02CE A0 13      SHOW   LDY #13
02D0 A2 05      LDX #5
02D2 A9 7F      LDA #7F
02D4 8D 41 17   STA PADD
02D7 B5 04      LITE   LDA WINDOW,X
02D9 8D 40 17   STA SAD
02DC 8C 42 17   STY SBD
02DF E6 11      POZ    INC MOD    pause loop
02E1 D0 FC      BNE POZ
02E3 88          DEY
02E4 88          DEY
02E5 CA          DEX
02E6 10 EF      BPL LITE
02E8 20 40 1F   JSR KEYIN
02EB 60          RTS
END
```

BAGELS

Program notes:

1. Program enforces a pause of about 4 seconds after displaying counts or answer. This guards against display being 'missed' due to bounce, hasty keying.
2. After count displayed, or at end of game(s), user can blank display, if desired, by pressing GO or any numeric key. Game operation is not affected, but user may feel it 'separates' games better.
3. When a digit from the user's guess is matched, it is destroyed so that it will not be matched again. There are two significantly different types of 'destruction', however (at 27D and 29D); the test at label STEP is sensitive to which one is used.

Here's an excellent example of using KIM to check itself... from Lewis Edwards Jr.
 1451 Hamilton Ave
 Trenton, NJ 08629

"PLL SET" PROGRAM

Having trouble loading from tape, especially on "SUPERTAPE"? Suspect the PLL adjustment might be off, but were afraid to adjust it, or didn't have a meter or scope handy? Use this program and KIM's built in hardware to make the adjustment. Hold the tip of the plug you plug into the tape recorder's earphone jack to applications pin #14 and adjust the control for 0's or combinations of 7's and L's on the display. "L" means the PLL TEST line is low and "7" means it's high. The program generates a signal that alternates slightly below and slightly above the one generated by KIM at 1A6B. The regular tape input channel is utilized and decoded to control the display.

1780 A9 07	BEAN LDA #07	Set the input
1782 8D 42 17	STA SBD	
1785 A9 01	LDA #01	and output ports
1787 8D 01 17	SFA PA0	
178A 85 E1	STA E1	Initialize the toggle
178C A9 7F	LDA #7F	
178E 8D 41 17	STA PADD	Open display channels
1791 A2 09	MORE LDX #09	Start with the first
1793 A0 07	LDY #07	digit Light top & right
1795 2C 42-17	BIT SBD	if PLL output
1798 30 02	BMI SEGS	is high
179A A0 38	LDY #38	otherwise left & bottom
179C 8C 40 17	SEGS STY SAD	Turn on the segments
179F 8E 42 17	STX SBD	and the digit
17A2 2C 47 17	DELA BIT CLKRDI	Half cycle done?
17A5 10 FB	BPL DELA	No, wait for time up
17A7 E6 E2	INC E2	Count the cycles
17A9 30 04	BMI LOTO	128 1/2 cycles, send low tone
17AB A9 91	HITO LDA #91	128 1/2 cycles, send hi tone
17AD D0 03	BNE CLK1	
17AF A9 93	LOTO LDA #93	
17E1 EA	NOP	Equalize the branches
17B2 8D 44 17	CLK1 STA CLK1T	Set the clock
17B5 A9 01	LDA #01	
17B7 45 E1	EOR E1	Flip the toggle register
17B9 85 E1	STA E1	
17BB 8D 00 17	STA PA0	Toggle the output port
17BE E8	INX	
17BF E8	INX	Next display digit
17C0 E0 15	CPX #15	Last one?
17C2 D0 CF	BNE NEXT	No, do next
17C4 FG CB	BEQ MORE	Yes, do more

p. 3

Tom Wear
 380 Beldaire Ct
 Punta Gorda, FL
 33950

ADDING MEMORY TO KIM

Would you like to add 4K starting at location ~~0400~~ without address line drivers and without changing U4 to 74LS145? Maybe you can, or if you already have— Pass the word.

Prompted by a query from Wm. Dial, I pulled the drivers to my memory board and jumpered the lines at the socket. The system was then cycled continuously on a memory test program for two hours without an error. The load on KIM was 32 2102's from three different sources, and a TWT which added one TTL 'LS' input load to each address line.

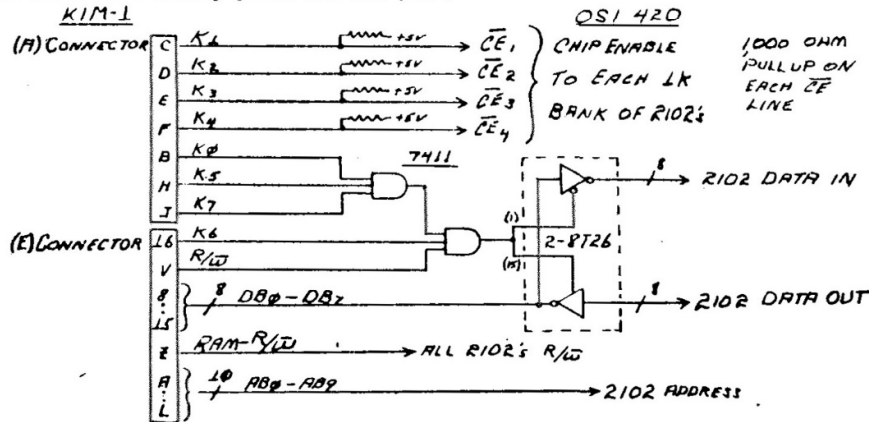
The discussion and drawings that follow will describe:

1. BARE BONES - The memory suggested by the test conducted.
2. ~~0400~~ & ~~2000~~ PLUS - My current 4K and its decoding.
3. ON THE BACKPLANE - Full decoding for KIM.
4. TEST PROGRAM - My effort towards a complete memory test.

The OSI 420 Memory Board (but none of the OSI decode method) and the OSI 480 Backplane are used. However, what is shown is applicable to other available PC boards. Some of this will seem a short cut to the complexities of KIM expansion in comparison to OSI or MOS Technology approach (and it is), but there is NO short cut to good sockets in every position and a well managed power supply.

1. BARE BONES

Success or non-success may depend greatly on the individual differences of the 6502 on each KIM board, the 2102's used, and most particularly, on electrical noise environment (do not skimp on at least .01 uf and preferably .1 uf disk capacitors along that 5-volt power distribution bus). In addition to 2102's, two 8T26 data buffers and a 7411 for control of the 8T26 are used. Pad D of the 420 board is a likely place for the 7411.

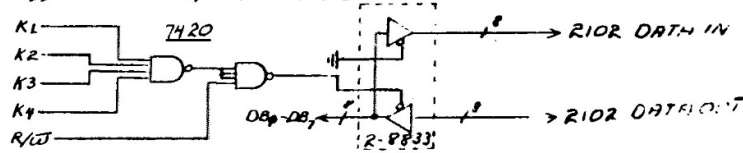


Operation: The 7411 AND gates control the 8T26 data buffers such that if any of K5, K6, or K7 are low, the 8T26 puts no signal on the data lines to KIM. When K5, K6, K7 are all high the 8T26 direction is controlled by R/W from KIM to read or write to memory selected by any of K1, K2, K3, or K4.

The check out should proceed initially with only 1K of memory installed to aid distinguishing potential inadequate drive from KIM, from other irregularities. It seems extremely unlikely that any KIM would not drive at least 1K of new memory.

Once peeking and poking succeeds via the KIM keyboard, a long cycling run with a memory test program is handy to search for those rare events or to gain confidence that there are none.

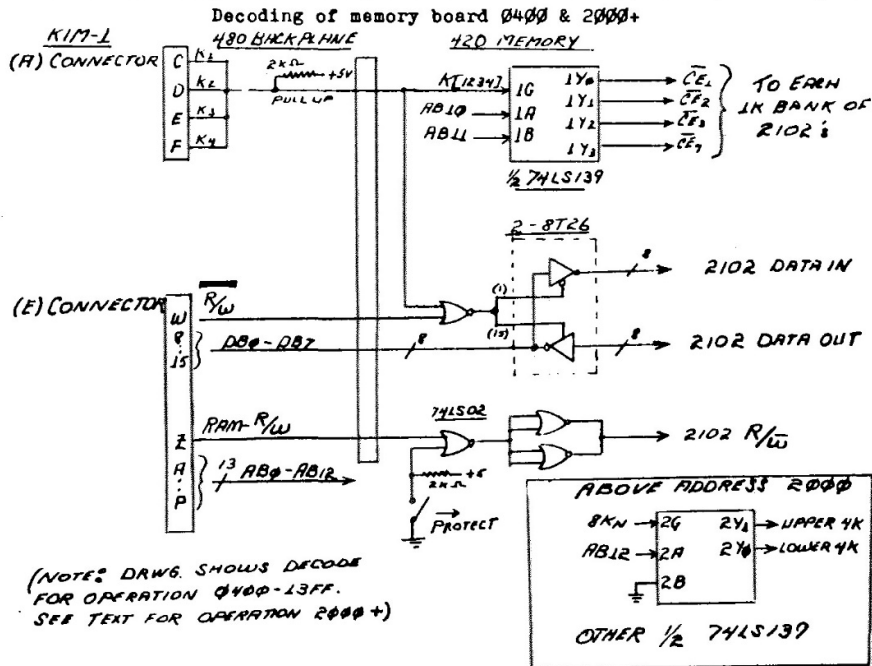
If you have selected some other memory board for you addition, like SWTP which uses the 8833 data buffer, then use this:



If for whatever reason address drivers are to be added, I would suggest that the installation effort be part of your planning for future expansion and not on the memory board. However, OSI in their Application Note #5 did describe a scheme of installing two 7417's as drivers on the 420 board.

2. 0400 & 2000 PLUS

For operation above address 2000 obviously what has been shown so far will not work. Further decoding of AB15 thru AB10 is needed plus a solution to KIM U4 74145 loading on AB12, AB11 and AB10. I chose to install a 74LS145. Pulling a DIP with proper tools is a simple operation; without can be a nightmare. If you feel shy may I suggest a visit to the friendly TV repairman--he should have an innate curiosity about microprocessors and their application to TV games.

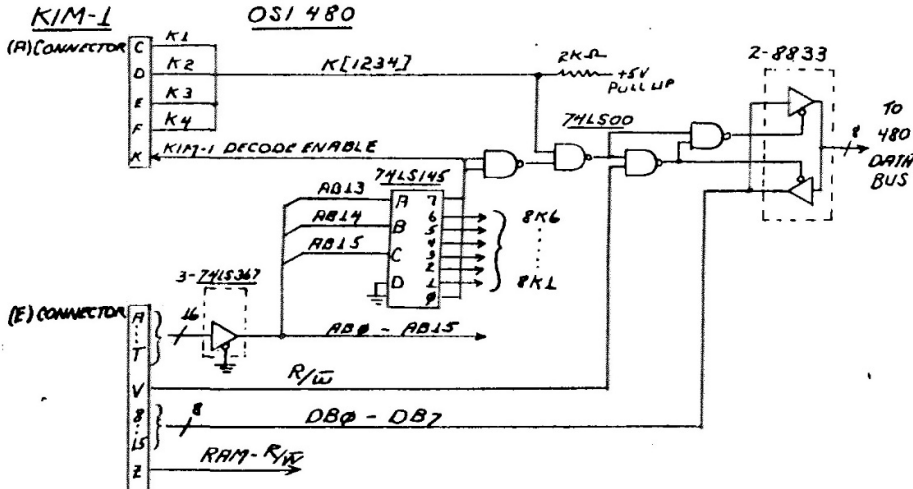


The NOR gate was necessary for the control of the 8T26. Note the negated (inverse) R/W. Also the NOR gates in the RAM-R/W line are not essential but were free. It is not good practice to parallel normal TTL gates, however, where they are on the same substrate, generally, no problems arise. Wired as shown and without the 480 Backplane, direct connection to KIM-1 would provide 0400 to 13FF operation.

With expansion decoding per KIM Manual, page 74, 8K1 could replace K(1234) at 1G of the 74LS139 and the memory would operate 2000-2FFF but then repeat 3000-3FFF. If for some reason two such 4K memories are to be operated in the same 8K block, then the other half of the 74LS139 can be used to resolve the repeat. Input to 2G would identify the block and outputs 2Y0 or 2Y1 to 1G would select upper or lower 4K. If you are adding 8K of 2102's on one board for operation above 2000 then the 74LS139 should be replaced with 74LS138. My planning suggests leaving this 4K design at 0400 to 13FF, and waiting for some of those new 4, 8, or 16K-bit chips to reach more palatable pricing.

3. ON THE BACKPLANE

Decoding AB15, AB14 and AB13 as described in the KIM Manual is about as good as any considering the restrictions imposed by the KIM ROM address locations. The approach shown here commits the 8K7 space to the interrupt vectors only. (I will try to skip by on 56K) The data buffer is needed only for isolation in DMA operations or maybe for the full 56K.



The 8833 data buffer was selected to preclude inverting data to TWT which has its memory accessible. On the OS1 480 Backplane, pads in area B13 to B25 will accommodate the 74LS145 and 74LS00. In the area B1 to B13 the traces were peeled, stick-on pads placed and drilled for installation of the two 8833's. The Area near B25 was designed for 7417's as drivers and therefore only 14-pin pads. I preferred tri-state so drilled extra holes and installed 74LS367's.

OSI - Ohio Scientific Instruments, 11679 Hayden St., Hiram, Ohio 44234

SWTP - Southwest Technical Products Corp., Box 32040, San Antonio, TX 78284

4. TEST PROGRAM

This program takes about five seconds per 1K of memory but I believe it is thorough. Each location must hold contents while all other under test are changed. All possible combinations of contents are used. The program is self-cycling and at the end of each cycle the display flashes the total cycles accumulated. An error causes a stop and the display will show the address and contents of the error location. The programing has been selected for speed and any improvements are welcomed. Load 0000 with BEGIN ADH and 0001 with END ADH, then enter program at 0002.

```

00 IX      -BEGIN      20 E6 23      INC Z BASE2
01 IX      -END        2E E6 28      INC Z BASE3
02 14 00    ENTRY     LDY Z BEGIN  30 04 28      CPY Z BASE3
04 04 17    STY Z BASE1  32 00 ED      BCS ②
06 04 23    STY Z BASE2  34 69 01      ADC # 01      C-0
08 04 28    STY Z BASE3  36 D0 12      BNE ④
0A A4 01    LDY Z END    38 E6 F9      INC Z INH
0C A9 00    LDA # 00     3A A9 FF      LDA # FF
0E 05 F9    STA Z INH    3C 8D 07 17   STA TRMT
10 05 FA    STA Z POINTL  3F 20 1F 17   JSR SCAND ;dapl
12 05 FB    STA K POINTH  42 AD 07 17   LDA TRMT
14 AA      TAX        45 10 F8      BPL ③
15 9D 00 IX ① STA BASE1,X ;clear 47 A9 00      LDA # 00
18 E8      INX        49 AA      TAX
19 D0 FA    BNE ①        4A A4 00      ④ LDY Z BEGIN
1B E6 17    INC Z BASE1  4C 04 23      STY Z BASE2
1D 04 17 CH 17  CFY Z BASE1  4E 04 28      STY Z BASE3
1F 00 FA    BCS ①        50 A4 01      LDY Z END
21 DD 00 IX ② CMP BASE2,X ;check 52 D0 0D      BNE ②      always
24 D0 2E    BNE ERROR    54 A5 23      ERROR LDA Z BASE2
26 FE 00 IX INC BASE3,X  56 85 FB      STA Z POINTH
29 E8      INX        58 86 FA      STY Z POINTL
2A D0 F5    BNE ②        5A 4C 4F 1C   JMP START

```

Note that the program begins and ends on a page boundary. For example, set BEGIN to 02 and END to 03, and the test will be conducted from 0200 to 03FF.

Want to enhance TINY BASIC? Here's a way..... from Don Box, 1250 White Oak Dr.,
Cookeville, Tenn. 38501

A few quick words to let you know I have Tom Pittman's TINY BASIC running on my KIM-1 & 2. I am using an old model 15 (5-level) teletype and had to do software code conversion (will furnish the routines if anybody wants; send S.A.S.E.).

TINY BASIC has a USR function to call user written sub-routines. Included are two short routines to allow the simulation of a subscripted variable.

STORE SUBSCRIPTED VARIABLE		RECALL SUBSCRIPTED VARIABLE	
STORE	TYA ; Y on entry has subscript	RECALL	TYA
	ASL-A ; double because saving 2		ASL-A
	TAY ; bytes		TAY
	LDA B4 ; pick up first byte (LSD)		LDA ARRAY,Y
	STA ARRAY,Y ; store abs,Y		PHA
	INY ; count		INY
	LDA B5 ; get second byte (MSD)		LDA ARRAY,Y
	STA ARRAY,Y ; store it		TAY ;return MSD
	RTS ; return		in Y
			PLA ;LSD in A
			RTS ;back to TINY

location B4, B5 is TINY BASIC's variable Z

ARRAY is location where data can be stored (in my case 0200)

p. 5

```

700 LET I=1
710 PRINT "VAR1,VAR2"
720 INPUT A,B
730 LET Z=A
740 LET Z=USR(STORE,I*2-2)
750 LET Z=B
760 LET Z=USR(STORE,I*2-1)
770 LET I=I+1
780 IF I<=max no. GO TO 720
790 END
.
.
SUBROUTINE
TO SEARCH TABLE FOR T=VAR1 AND IF FOUND RETURN VAR2 in J
.
.
900 LET J=1
910 IF T<=USR(RECALL,J*2-2) GO TO 960
920 LET J=J+1
930 IF J<=max no. GO TO 910
940 REM ERROR RETURN HERE
950 RETURN
960 LET J=USR(RECALL,J*2-1)
970 RETURN
.
.

```

note: 740 & 760 are dummy; LET's to force a call to the store routine.

max no. = number of pairs of values to be stored

note: value is returned as the value of the USR function

where: STORE = decimal equivalent of address where the store routine is located

RECALL = decimal equivalent of address where the recall routine is located.

TINY BASIC programs can easily be stored and loaded from cassette tape. Location 0020, 0021 contains the starting address and 0024, 0025 will have the ending address. Set up for normal tape dump (using KIM's dump @ 1800) and write down the contents of 0024 and 0025. To reload, use KIM's tape loader (@ 1873), then reset 0024 and 0025. Warm start TINY BASIC and you're off and running.

LOCAL USER GROUPS getting started-

Somerville, N.J. area-
Frank Raymond
574 Auten Rd. #4C
Somerville, N.J. 08876 Phone 215-8743644

Philadelphia, Pa. area-
Ron Kushnier
3108 Addison Ct.
Cornwell Hts., Pa. 19020 Phone 215-757-9057

Phoenix Arizona area-
Karl Lunt
1561 W. Peoria Ave.
Phoenix, Ariz. 85029

Keep the rest of us up to date on your local KIM group activities!

.....

p. 6

MVM 1024 MICROPROCESSOR VIDEO MODULE

THE KIM-1 COMES ALIVE WITH A VIDEO DISPLAY. At last there's a sophisticated display that interfaces easily to the KIM-1

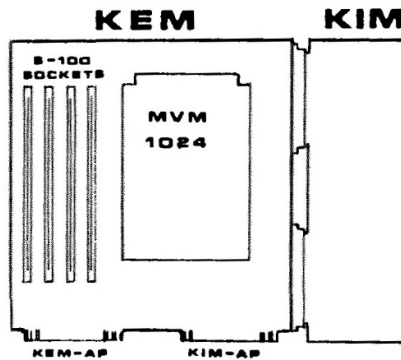
- *** 1K bytes on-board, cursor addressed RAM
- *** Read/writable true blinking cursor
- *** 16 rows of 64 characters, upper/lower case ASCII
- *** Reversed video characters, full screen video reverse
- *** Tangent block characters for block graphics and Op Art
- *** Byte parallel display, whole screen can be written in milliseconds
- *** Text editing and monitor functions available in 1K x 8 KIMBUG EPROM

**KIM-1
GOES
VIDEO**

Please write for further information. We support our products.

KEM KIM-1 EXPANSION MODULE

Have you ever envied S-100 Bus users? ENVY NO MORE. Be the envy with your KIM-1 and KEM. Use low-cost S-100 bus RAM and other S-100 accessories with your KIM-1. Add the best display available. It's all yours with the KEM.



- * Buffered data, address, and controls for the S-100 bus
- * Comes with two S-100 connectors, room for two more, expandable
- * Connector and interface for MVM-1024 Display
- * Connector KIM-1 appl. ports, audio etc.
- * KEM application connector, ASCII keyboard interface
- * Space for four 1K x 8 2708 EPROMs (These are down to \$30 now and getting even cheaper. Maybe \$25 by year's end.)

Riverside
ELECTRONIC DESIGN INC.

1700 NIAGARA STREET BUFFALO NEW YORK 14207

PHONE 716 875-7070

Here are some more real-time clock subroutines to add to C. H. Parsons program in the last issue. Simply replace the no-ops starting at \$03DB with the proper subroutine calls. By the way, to start the clock, initialize the correct time in the zero page time registers and start the program at \$03C0. If the clock is running, start at \$03C9. By now you have an idea of the potential of an open-ended real time clock in your machine. Some further work by Parsons resulted in a temperature sensor interface that ties into the clock routines. These will be presented in future issues. By the way, if $\frac{1}{2}$ sec. interrupts start playing havoc with a fully expanded real time clock, and you don't want to install a clock chip, simply use a 60 HZ. power line conditioning circuit and a divide-by-60 counter arrangement to give you 1 sec. interrupts.

All routines were written by C. H. Parsons

Two Tone Sound to Indicate Hours

Line	Code	Label	Instruction	Comment
0320	A 582	BEEP	LDA MIN	On The Hour?
0322	D029		RNE END	If Not Return
0324	A 581		LDA SEC	Execute Until SEC = HR
0326	38		SEC	
0327	E583		SBC HR	
0329	1024		BPL END	
032B	A 580	AGAIN	LDA QSEC	First $\frac{1}{2}$ Second?
032D	D006		RNE ONE	
032F	A91E		LDA #31E	Set High Note
0331	8570		STA NOTE	
0333	D00A		RNE GO	Sound Note For $\frac{1}{2}$ Second
0335	A901	ONE	LDA #301	Second $\frac{1}{2}$ Second?
0337	C580		CMF QSEC	
0339	D014		BNE END	
033B	A928		LDA #328	Set Low Note
033D	8570		STA NOTE	
033F	A901	GO	LDA #301	Set I/O Ports
0341	8D0317		STA PRDD	
0344	EE0217		INC PRD	Toggle Speaker
0347	A570		LDA NOTE	
0349	AA		TAX	Set Delay
034A	CA		DEX	
034B	10FD		RPL	
034D	30DC		RMI AGAIN	Keep Sounding
034F	60	END	RTN	

Additional Zero Page Locations

0070	NOTE	Sets Frequency of Note
------	------	------------------------

This is a subroutine which when added to the clock display routine will use the real time clock data to produce one sound per hour on the hour. The output is a speaker circuit as shown on Pg. 57 of the KIM-1 Manual. It is hooked to PR0 rather than PA0. The specific notes can be changed by altering 0330 and 033C.

Consecutive Minute Timer

Line	Code	Label	Instruction	Comment
0200	A 580	MTIME	LDA QSEC	Test QSEC
0202	F041		REQ RESET	If Zero Reset State
0204	C901		CMF #301	
0206	F048		REQ SOUND	If One Sound Signal
0208	C902		CMF #302	
020A	F00F		REQ TIME	If Two Look For Delays
020C	C903		CMF #303	
020E	D00A		BNE OUT1	If Three Initialize
0210	A 573	IN	LDA STATE	
0212	D006		BNE OUT1	If State is Zero
0214	E673		INC STATE	Put State=1
0216	A 581		LDA SEC	Put SEC in RSEC
0218	8572		STA RSEC	For Reference
021A	60	OUT1	RTN	

0213	A573	TIME	LDA STATE	Look For Delays
021D	C901		CMP #301	If State=1 And
021F	D0F9		BNE OUT1	
0221	A581		LDA SEC	Second= RSEC
0223	C572		CMP RSEC	
0225	D0F3		BNE OUT1	
0227	A900		LDA #300	Clear X
0229	AA		TAX	
022A	E673		INC STATE	Put State=2
022C	B574	AGAIN	LDA T1,X	Look For Nonzero'S
022E	F00D		PEQ NEXT	In 0074 Through 007B
0230	A905		LDA #305	Put Number of Sounds=5
0232	8571		STA NSOUND	
0234	D674		DEC T1,X	Subtract One From Delay
0236	D004		BNE OUT2	When Delay Goes to Zero
0238	E673		INC STATE	Put State=3
023A	867E		STX EVENT	Put Tx in Event Counter
023C	60	OUT2	RTN	
023D	E8	NEXT	INX	Look at Next Tx
023E	E008		CPX #308	Do Eight Times
0240	DOEA		BNE AGAIN	
0242	A900		LDA #300	Clear State
0244	8573		STA STATE	
0246	60	OUT3	RTN	
0247	A573	RESET	LDA STATE	Put State=1 if it is 2
0249	C902		CMP #302	
024B	D002		BNE OUT4	
024D	C673		DEC STATE	
024F	60	OUT4	RTN	
0250	A573	SOUND	LDA STATE	Sound if State=3
0252	C903		CMP #303	
0254	D0F9		BNE OUT4	
0256	A901		LDA #301	Set I/O Ports
0258	8D0317		STA PHDD	
025B	A580	KEEPS	LDA QSEC	QSEC Still=1?
025D	C901		CMP #301	
025F	D00B		BNE DEC	If Not Subtract One Sound
0261	EE0217		INC PRD	Toggle Speaker
0264	A918		LDA #318	Set Note
0266	A8		TAY	
0267	88	NOTE	DEY	Decrement Delay
0268	10FD		BPL NOTE	
026A	30EF		BMI KEEPS	Keep Sounding For One
026C	C671	DEC	DEC NSOUND	Quarter Second
026E	D004		BNE OUT5	
0270	A901		LDA #301	Put 1 in State to Look
0272	8573		STA STATE	Again When Finished Sounding
0274	60	OUT5	RTN	

Additional Zero Page Locations

0071	NSOUND	Sets Number of Notes
0072	RSEC	Store Starting Second
0073	STATE	State Counter
0074	T1	First Time Delay
0075	T2	Second Time Delay
0076	T3	Third Time Delay
0077	T4	Fourth Time Delay
0078	T5	Fifth Time Delay
0079	T6	Sixth Time Delay
007A	T7	Seventh Time Delay
007B	T8	Eighth Time Delay

This is a subroutine which when added to the clock display routine will use the real time clock data to sound a signal five times after consecutive minute delays which are entered in locations 0074 through 007B. The minute delays are in HEX which will allow a maximum of a little over four hours. Locations 0073 through 007B should be cleared when starting up. Location 0073 should be cleared each time the delays are entered. The program clears the delays when they are executed. At each sounding the number of the delay is entered in location 007E for future reference. (0074=00,0075=01,... 007B=07).

Various Tidbits About How the KIM-1 Keyboard and Display Operate

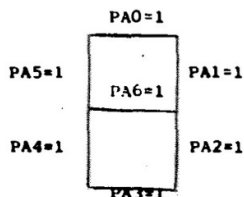
Most of the game programs written for KIM-1 use the keyboard and the display in real time interactive mode under program control rather than under control of the operating system located on the ROM. To be able to write such programs one has to understand the operation of the display and the keyboard. Referring to Fig. 3.5 on page 28 of the User's Manual one can see that four leads of the peripheral I/O bus B; PB1 - PB4 and 7 leads of the peripheral bus A; PA0 - PA6 are connected either directly or through the decoding IC 74145 to the keyboard and the 6 display digits. The peripheral buses A and B are controlled by memory locations 1740 (data on Port A), 1741 (data direction on A), 1742 (data on B) and 1743 (data direction on B). This is similar to memory locations 1700 - 1703 which are controlling the non-committed application buses A and B which are the standard I/O ports to the KIM-1.

Display

The KIM-1 display consists of 6 common-anode LED digits with the corresponding cathode segments connected in parallel between all six digits. The segments are controlled by PA0 - PA6 and the digits by PB1 - PB4 decoded by the 74145 IC. For a particular segment to light up both the corresponding segment cathode and the digit anode have to be activated by the appropriate outputs on the peripheral buses A and B. The following table shows the state of PB1 - PB4 required to activate each of the 6 digits:

Digit on (left to right):	1	2	3	4	5	6
PB4	0	0	0	0	1	1
PB3	1	1	1	1	0	0
PB2	0	0	1	1	0	0
PB1	0	1	0	1	0	1
Word to be stored in loc. 1742 e.g.	08	0A	0C	0E	10	12

The following figure shows the bit pattern on the port A to activate the seven LED segments:



E.g. "E" = x1111001 (79) on port A
(in location 1740)
x stands for don't care

The segments and digits have to be activated in close succession. The bit patterns on Port B are such that the bit pattern for the next digit to the right can be obtained by adding 2 to the bit pattern for the previous digit. There can only be one digit activated at any one time due to the logic of the IC 74145 decoder. To display successive digits one would increment port B (loc. 1742) by 2 to scan from left to right or decrement by 2 to scan from right to left. The digit and the corresponding segment commands should be within a few consecutive program statements. The scan should "rest" for about 1 ms at each digit; if the scan rate is too fast then the whole display will glow including unwanted segments.

Keyboard

The P01 - P04 ports are also used for sending pulses to ports PA0 - PA6 via the keyboard and thus sense the key status. To set P01 - P04 to output and PA0 - PA6 to input you have to write 00011110 = 1E to location 1743 and 10000000 = 80 to location 1741 (addresses for data direction on ports A and B). To activate keys 0 through 6 the bit pattern on P04 - P01 has to be 0000, for keys 7 through D - 0001, for keys E,F,DA,AD, +,GO and PC the bit pattern has to be 0010. The remaining keys RS and ST are hardwired to the microprocessor. With no keys depressed input on port A (loc. 1740) will consist of all 1's or FF (bit 7 is automatically set to 1). Depressing a key will insert a 0 in the bit pattern. Writing xxx0010x, for example 04, on port B and depressing the key GO will result in word FD being received on port A. Of course the simplest way to check for key depression is to call ROM routines AK or GETKEY. The following table shows what these 2 routines put into the accumulator. Note that both of them destroy the contents of X and Y registers.

Key	GETKEY (decimal flag set)	GETKEY (dec. flag cleared)	AK
0	0	0	40
1	1	1	20
2	2	2	10
3	3	3	08
4	4	4	04
5	5	5	02
6	6	6	01
7	7	7	40
8	8	8	20
9	9	9	10
A	10	A	08
B	11	B	04
C	12	C	02
D	13	D	01
E	14	E	40
F	15	F	20
AD	16	10	10
DA	17	11	08
+	18	12	04
GO	19	13	02
PC	20	14	01
No Key	15	15	00

HERE'S SOME INTERESTING ITEMS FROM MIKE FIRTH:

Please mention the fact that my large type 6502 Instruction Summary Summary is missing the command B6 from the last column (LDX, z page,y), as you pointed out. If anyone else wants one, I had so many requests I had them printed on green paper to make them easier to find on the desk. Send a Self-Addressed, Stamped Envelope (#10 is best), plus a 9¢ stamp loose for one copy, or a 13¢ stamp for two copies to: Mike Firth/6500, 104 N.St.Mary, Dallas, TX 75214.

I would like to mention MIXIM, as I have labeled the system I am working on. Because I expect to be expanding my system for some time, and because I expect to develop a number of different jobs for my system, including control of things around the house, games, and data management and editing, I want to define a system which will let me put routines in memory at will (i.e. they must be relocatable). Perhaps others will find my thinking useful.

Because I expect to use some large arrays, many of my routines will use indexing and because of not wanting to move the data unnecessarily, I am going to pass the address of the data to the subroutine. However, if I ever want to put my programs in ROM, I am going to have to put this address someplace besides inside the program to index on it. Because of the variety of choices, using the Zero Page (zpage, from now on) is the logical place.

p.8

So, I need to partition up zpage a bit. Because I expect to keep KIM for some time, I will leave its reserved space at the top and reserve E0 on up for some of my system items (more on those below). Somewhat arbitrarily, I am going to define 00-0F as absolute scratch pad. Any program can use the area, no program can assume anything saved in the area. 10 on up to DF will be used for two purposes and the limits of the two areas will be marked by pointers kept in MIKIM system area. 10 up to the lower pointer will be reservable scratch area in which a routine may keep data for its own use or to pass to another routine. DF down to the upper pointer will be for storing data addresses for various kinds of indirect addressing.

Besides these two areas, three others are available: The stack can be considered a Reservable area, if push/pull's are matched; page 1 below the pointer can be considered absolute scratch pad if the pointer location is checked, and the RAM in the 6530's can be used. I expect to limit use of the latter only to routines that also use the 6530 I/O and timers.

No matter what I call a routine, I will get to it with a JSR and use an RTS to return. The difference will only depend on how I get the variables to the routine. So, here is my thinking so far:

FUNCTION: Variables are passed in the registers, usually one variable and that in the accumulator. Includes things like random, sine, time, hashing.
 ZFUNC: In the accumulator is a zpage address of the start of the data, with the number of items in X when needed. If a specific order is required, the calling program must provide it. Might include averaging, maximum, minimum, multiply. Answer is returned as a zpage address in A, with the number of items in X.
 ZSUB: A contains the zpage address of the start of the addresses of the data, X contains the number of addresses. All addresses, even Z page, stored as two bytes. Changed data is stored at addresses assigned to variables. (IND,X)
 SUBROUTINE: A contains the address of the Subroutine Stack Pointer, a two byte address in zpage. The SSP indicates where in memory to find the beginning of the addresses containing the data. For example (and this can get messy, but it is the most flexible):
 A might contain E0
 and E0 and E1 might contain 25 23
 then location 2325 would have an address for the first variable used by the Subroutine. This might be the start of an array.
 So, if 2325 is moved to zpage, I can index on it to access the whole array without including it in my program, [(IND),Y], without knowing where it is in memory, without knowing where the pointer is in memory.

A quiet examination will reveal that many Subroutines could be Zsubs with housekeeping to get the addresses into zpage and that Zsubs could be Zfuncs with housekeeping to get data into zpage, but in either case, with large blocks of data, like I expect to use, it would be possible to outgrow the smaller routines rather quickly.

I may have more later, but now, having thought out some of the possibilities, I am going to write some of the simpler display and game programs and see if what I hope can happen will happen.

Good Idea!

My standard connections for peripherals (which I would love to see put in the corner of every card of any system) are shown at the right. It is a 16 pin DIP viewed from the plugin side.

	Ground	Pins	
	Data Bit0	1 16	+5 (may be other if labeled)
	B2	2 15	B1
	B4	3 14	B3
	B6	4 13	B5
	B8	5 12	B7

	Data status	6 11	R/W when needed
	Clock or 16x rate	7 10	2nd clock, if needed
	Int or optional	8 9	2nd volts or optional

In many applications, the last pair or pairs of pins are not needed. My keyboard has this standard plug and goes very nicely to the sockets wired to A port. (B7 is odd parity) If I am going to do interput drive, either on a socket or (more pleasing) on the keyboard, I am going to have to connect the available data stable strobe (now on pin 6) to bit7 (for port use) or pin 8 and create an interrupt buss.

		Pins
My address standard socket, simply alternates lines back and forth across the socket, LSB first. This permits use of an 8,10 or 12 pin plug to take those lines if that is all that is needed.	Bit0	1 16 B1
	Bit2	2 15 B3
		...and so forth.

ITEM

If you bought your KIM early and have the early manual, you may not have the note on page H-7 of the later ones. There it says that if you want to use the interrupt mode of the timer, you have to run a wire from pin 15 of the application connector to either pin 4 (IRQ) or pin 6 (NMI) of the expansion connector and PB7 should be programmed for input (normal after RESET).

I would like to ask for help on one problem. Sometimes when I am plugging in to my setup, or when I touch the aluminum plate that is at ground, the display suddenly switches to one very bright digit. It may then display a location after a few moments, usually in the middle of the tape write routine. My first thought was a power loss, but memory remains intact (I have a clock program that sometimes gets interrupted but resarts and runs fine.

If you live in the Chicago area and want to help me, I would really appreciate it. I am running a panel on the use of Small and Timesharing Computers in Theatre at the American Theatre Association Convention, August 16. I would like to show off some really low cost working systems, but I have no budget. I will be hauling as much of my system as I can. If you have a KIM rigged to a teletype, especially with an editing routine that might show how the system could read info from a cassette and type labels AND you could help me get it into the Palmer House in the middle of a weekday and then out again, please let me know. Any other intermediate help would be useful. Mike Firth, 104 N. St. Mary, Dallas, TX 75214.

USE OF KIM-1 KEYBOARD WITH USER PROGRAMS

Halnh W. Burhans
161 Grosvenor St.
Athens, OH 45701

We have found a simple way of using the KIM-1 keyboard monitor software to load hex numbers into memory while operating some user loop program continuously. The KIM GETKEY subroutine located at 1F6A recognizes hex numbers from \$0 to \$15. A \$15 is loaded into the accumulator if no key is pressed, and a key operated loads the hex numbers according to the table:

KEY OPERATED	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	AD	DA	GO	PC	
HEX NUMBER	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14

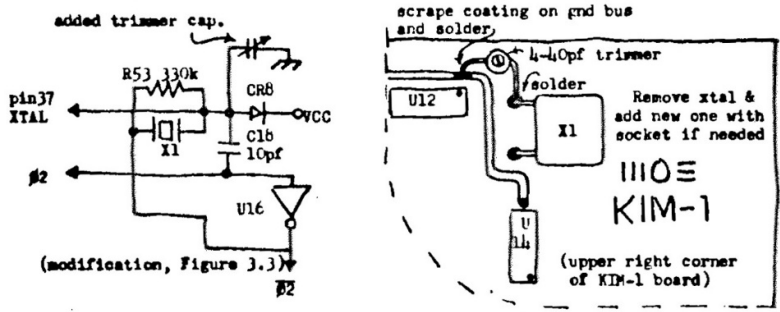
A simple example subroutine for page zero use might involve:

ADDRESS	DATA	LABEL	COMMENT
001F	xx		Location of data desired
START 0020	20	JSR	to KIM GETKEY at 1F6A
0021	6A		
0022	1F		
0023	C9	CMF	compare immediate with 15
0024	15		
0025	F0	BEQ	branch if 1, otherwise store key
0026	02		
0027	85	STA	in memory location 1F
0028	1F		
0029	4C	JMP	back to START
002A	20		
002B	00		

One use we have made is to locate a program like this as a return loop after servicing some interrupt request involving perhaps several pages. When not servicing the interrupt, the program keeps scanning through this loop looking for a new number to enter into some memory location which is used in the interrupt routine or main user program. When an interrupt comes along the program jumps to the desired routine, services the interrupt which may include some new command entered just previously from the keyboard, but then jumps back to this loop after servicing the interrupt with an RTI. A somewhat more complex routine could be used to enter a larger batch number of several hex digits, and even to keep track of several different memory locations using the 5th bit with the AD DA GO or PC keys. The above example has one peculiarity in that when the routine is first initialized at START, an automatic hex 13 is entered because the GO key is this number. However the user can immediately enter some other number as required in the interrupt service program originated by the user. For more general use with any memory location an absolute STA of 8D should be used which adds one more cycle before jumping back to START.

p.9

The 1 MHz crystal on my KIM-1 board was 200 Hz too high. This results in a 4 Hz error when using the frequency counter routine at 20 KHz and similar small errors for precision time interval measurements. We needed to set the clock to within ± 10 Hz for some Loran-C timing experiments. The original crystal was removed, a new one from JAMES ELECTRONICS was obtained and soldered in place with a right-angle crystal socket. A miniature trimmer capacitor with a negative temperature coefficient of about -1600 was obtained (MEPCO CT5A6R540A - $6\text{-}4\text{0pf}$) and soldered in parallel with the input side of the crystal to the ground bus running adjacent to the U-12, U-14 IC's on top of the board (see sketch and circuit modification below). The mod allows trimming the KIM clock oscillator to within ± 1 Hz with respect to an external standard and it is somewhat temperature compensated for room temperature variations, holding the frequency within ± 5 Hz for a $\pm 5^\circ\text{C}$ change.



R. W. Burhans, E. E. Dept., Ohio University, Athens, Ohio 45701

THE TRENTON COMPUTER FESTIVAL

RONALD KUSHNER
3108 ADDISON COURT
CORNWELLS HEIGHTS, PA. 19038

There are only a few words to describe the Trenton Computer Festival. #0s! FANTASTIC! I think you can learn more from one day of "Computer Festing", than from several months of reading and experimenting. If there is a similar festival or show in a 100-mile radius of where you live, by all means go! I'm sure it will be worth your while.

The real star of the Trenton Festival was none other than good old Kim. From a demonstration of Peter Jennings Micro-shess to Rod Loofbourrow's Microcomputer controlled robot (see April '77 Interface), it was Kim's day all the way. I feel that the most fantastic demonstration was Hal Chamberlain's computerized music. With just a basic Kim and a handful of parts, Chamberlain got the computer to play Exodus - in four part harmony! Now, the audience was used to computer music sounding like little more than a group of disjointed beeps put together to sound something like a melody of a song, so you can well imagine the reaction when Hal pressed the "Go" button and out from the speaker came the most beautiful, melodious, rich pear-shaped organ music I have heard in a long time. When the composition was finished, there was a moment of absolute silence as the audience tried to comprehend the full impact and significance of what they had just heard - then came a round of thunderous applause. Once again, the power of Kim shattered my mind. It was just unbelievable. At that point, I think I would have done just about anything to get my hands on that program! Hal assured us that it was being published in the September issue of BYTE. To paraphrase a song "It's gonna be a long long time from May to September..."

[By the way, if you're giving out the back issues to new members, it might be prudent to mention that in issue #1, Robert Lloyd's light blinking connection to Kim is a definite no-no. You're asking Kim to sink about 20 ma. per LED. This is much more than the ports can take (1.6 ma.) I suggest the driver circuit using the 75492 as per issue #5 be employed.]

One thing that became quite obvious at the Trenton Computer Festival, with all the Kim's floating around, was that no one has yet found a decent way to package their computer - with the exception perhaps of Tod Loofbourrow, who built a robot around his.

Since the user's notes cannot publish photographs, I would like to suggest that those members who do feel that they have found a reasonable approach to packaging Kim, send me a spare photo. I'll collate them and send them off to BYTE or Interface for possible publication. I think a pictorial article of novel packaging ideas would be quite useful.

RONALD RUSHNER

Now that we have a frequency counter for KIM, it's only fitting that we get a square wave generator program also. Bob also has a bit of info for those of us who have a Burroughs Airline Terminal...

from: Bob Slagle, K4GR
3515 25th St. North
Arlington, Va. 22207

SQUARE WAVE GENERATOR. Output on PA0.

```

GO      0000 D8          CLD.  Clear Decimal.
        01 18          CLC.  Clear Carry.
        02 A9 FF          LDA.  Load Accumulator with #FF.
START   04 8D 01 17     STA.  Set PADD to output.
        07 A9 01          LDA.  Load Accumulator with #01.
        09 8D 00 17     STA.  Set PAD to PA0, "ON"
        0C 20 40 00     JSR.  Delay
        0F A9 00          LDA.  Load Accumulator with #00.
        11 8D 00 17     STA.  Set PAD to PA0, "OFF"
        14 20 40 00     JSR.  Delay again.
        17 20 07 00     JSR.  Do it again. Go back to START
EXIT    001A 20 5C 18   JSR.  If "0000 00" shows, you goofed.

DELAY   0040 A0 FF*     LDY.  Load Y Index with #FF.
        42 A2 FF*     LDX.  Load X Index with #FF.
        44 CA          DEX.  Decrement X.
        45 D0 FD          BNE.  If result not 0, go back to 44.
        47 88          DEY.  Decrement Y.
        48 D0 F8          BNE.  If result not 0, go back to 42.
        4A 60          RTS.  Go back to where you were in the
                           main program.

```

*Change to make higher frequency. 'FF' in each gives slightly faster than 1 Hz, '01' in 'Y', and '1B' in 'X' gives 3.069 Khz.

PS: I bought the Burroughs Airline Terminal being advertised in KILOBAUD - If anyone else does they should know that pressing the CLEAR and the Processing Keys will bring up the 'P' symbol on the scope - pressing the CLEAR key alone will not do it. Not knowing this probably cost me twoweeks in trouble shooting before I got it playing. The book says pressing the CLEAR key only will bring up the symbol.

An industrial application for KIM from: Charles P. Pizura, Director of Marketing,
Hundley Controls Inc., 183 Columbia Rd., Hanover, Mass. 02339 Phone (617) 826-5019

I thought you might be interested in our application for the KIM boards - so here is a brief rundown on what we are doing: We are putting-together a KIM-1 and a KIM-3, packaging it within a brief case (~~attached picture~~) and offering it to the fuel oil industry as a degree day dispatching computer. The device includes a main and an auxiliary power supply (4 NI-CAD batteries), a cassette recorder, a TI 5050M, thermal, 10-digit calculator and a temperature probe. The system is programmed to take an hourly temperature sample and at a predetermined time each day, it spits-out a list of customers that the fuel oil dealer should deliver that day. The list represents a degree day calculation, based on the daily mean temperature, showing the gallons required by a particular customer. A tiny 3-byte master record is maintained for each customer, showing tank size, usage factor, etc. The file is scanned each day to determine which customers are below a tank threshold level that is defined by the user. The tank threshold level is variable, allowing the fuel oil dealer to select different delivery schemes, based on his particular requirements for the day. In other words, if he wants to deliver more customers, he raises the tank reserve factor; if he wants to deliver fewer customers, he lowers it. The printed listing routes the customers by zone and truck run, showing the fuel oil dealer a recommended run scheme for the day.

We call the system " the degree day dispatching computer, (3DC) ". Future enhancements are planned, including general accounting functions, wind chill and solar monitoring, a high speed printing capability and a floppy disk hook-up. In brief, it is a revolutionary device at an unheard of price. We are excited about it.

Your readers may be interested in the printing calculator hook-up and we want to make it available to them. We will provide the calculator, plus all hardware and software which is necessary to interface it. It will go for approximately \$250.00. Please have interested parties contact me directly.

KNOWN KIM-1 DISTRIBUTORS - for your information.

Johnson Computer, P.O. Box 523, Medina, Ohio 44256 Phone (216) 725-4560
Contemporary Marketing Inc., 790 Maple Lane, Bensenville, Ill. 60106 Phone (312) 595-0461
Cybersystems, Inc., 4306 Governors Dr., Huntsville, Ala. 35805 Phone (205) 837-2080
(they have a nifty KIM enclosure and may or may not sell the basic KIM)
Newman Computer Exchange, 1250 N. Main St., Ann Arbor, Mich. 48104
Computer Warehouse Store, 584 Commonwealth Ave., Boston, Mass. 02215 Phone (617) 261-2701

PERSONAL COMPUTING 77

Personal Computing 77 will be two full days of seminars, major exhibits and demonstrations in home and personal computers to be held 27 and 28 August in Atlantic City, NJ. Last year over 4,000 computer hobbyists and radio amateurs enjoyed Personal Computing 76. This year, Personal Computing 77 hopes to be able to sponsor a part of the microprocessor module to be included in the Phase III satellite that the Radio Amateur Satellite Corp. (AMSAT) is building for launch in 1979. By attending Personal Computing 77, you will help this organization to extend its support to AMSAT and you will see many fine radio and computer exhibits. For a free TRIP-KIT, write PC 77, Rt. 1, Box 242, Mays Landing, New Jersey 08390.

p11