Hi! Due to a foulup on my part, the last issue was marked #10 & #11. Well, that should have read #9 & #10----this is issue #11.
-----no kidding-----

****

THE FIRST TAPE OF KIM has been discontinued due to production problems. The first batch of 30 tapes were good because they were made one at a time but continuing in this fashion would have been cost prohibitive. We found out that trying to duplicate a 90 minute tape isnt that easy.

****

THE TRENTON COMPUTERFEST

This years TRENTON COMPUTERFEST was great fun! We had the pleasure of sharing a booth with Jim and Joanne Pollock of Pyramid Data Systems, who were showing their 65XX powered morse code keyboard p.c. board (industrial quality and plated-through holes), their extended I/O monitor "XIM", and a new product called "TTY HINTS" whicb explains the teletype routines from the KIM monitor software and gives some representative examples of their usage.

Hal ChamberIain, Micro Technology Unlimited, was very prominent with his KIM product line. Perhaps the most interesting of his products is the "VISABLE MEMORY" board. This board features 8K of dynamic RAM with totally transparent refresh and a high resolution (320x200) graphics interface that gets displayed on a normal raster scan video monitor. Actually the automatic dynamic RAM refresh is a free by-product of the video interface since the video portion must read all the addresses to refresh the screen and this, then, automatically refreshes the RAM. More on this and other products in a press release later in this issue.

GGRS Microtech (Box 368, Southhampton, Pa 18966) was there with a 6502 based S100 system which included such goodies as a Persci disc controller board, a TIM serial I/O board, and software to drive it. Bob Selzer, of GGRS, is a very enthusiastic proponent of FOURTH (a new high level language) and had some interesting demos to back up his enthusiasm. Bob says that he has FORTH running on an 8080 also and mentions that the 6502 version runs at a noticeably faster speed. (!)

Hudson Digital Electronics was present with their full size floppy disc interface, 8K static RAM cards, and prototypes of their RS-232 I/O board and wire wrap card. All their products are plug compatible with the "Standard" KIM-4 motherboard pinout and are constructed on the "industry STANDARD" 4.5" by 6.0" card size.

This brings up a very important point. A number of people have clamoring to get a "set of standards" for 6502 hardware and software, but still go off in their own directions when it comes down to hardware or software design even though a set of perfectly suitable 6502 standards have existed for quite some time. Thise standards consist of the MOS Technology assembler mnemonics and the KIM-4 bus design.

more

****

KIM-1 USER NOTES IS PUBLISHED BI-MONTHLY (whenever possible) by Eric C. Rehnke, 109 Centre Ave., West Norriton, Pa 19403. Subscription rates are $5.00 for six issues (U.S. & Canada) and $10.00 elsewhere.

It has been said that the MOS Technology assembler syntax is horrible, but the fact of the matter is that these mnemonics are "logically" correct, are not at all difficult to learn, and really make good sense.

A perfect example of this is the indirect modes of addressing, which seem to present the biggest problems in understanding to programming newcomers. The Micro-ade assembler (by Peter Jennings) uses the mnemonic LDAIX to portray the Load Accumulator Indexed Indirect instruciton while the MOS Tech. assembler uses LDA (label, X) to portray the same instruction. The second mnemonic graphically explains that the zero page indirect pointer to the address which contains the data to be loaded into the Accumulator is computed by adding the "X" register to the zero page address referenced by the "label". The first mnemonic imparts no such information.

Of course, neither of these two mnemonics would be very clear to the neophytes in the hobby but wouldn't it be better for newcomers to learn things the right way instead of some non-standard method? The biggest argument in favor of assemblers using non-standard mnemonics is that they are easier to write. Let's not let lazy programmers stand in the way of an already proven software standard. By the way, these two assemblers will be compared in greater detail late on in this issue.

As far as hardware goes, you'd have to go a long way to find a bus configuration that offers more versatility, modularity, and utility that a 4.5" by 6.0" card residing on the 44-pin bus.

Admittedly, the KIM-4 does not use the 4.5" by 6.0" size card, but it does use a 44-pin bus that should be adopted no matter what card size you choose to utilize. Actually, if new hardware manufacturers adopt this 4.5x6x44 style card configuration, their products would be directly plug compatible with around 1000 KIM-4s already in the field as well as any new system configurations which are generated by forward thinking hardware design firms. At this time Hudson Digital Electronics is the only known source of this 4.5x6x44 style card but this, I feel, will change shortly as soon as more people see the ultimate utility this type of system has to offer.

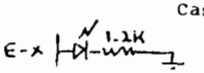
The only problem with this style configuration is that cards can inadvertently be installed backwards destroying IC's and causing many headaches in the process. This problem is easily solved, though, by installing a keyway between pin 18 and pin 19 on the edge connector and cutting a slot between the corresponding positions on the circuit boards. This procedure will shortly be adopted by MOS Tech. and is hereby recommended for general usage.

The 4.5x6x44 is ideal for installing in a Vector 19" wide rack mounted card cage which makes it quite suitable for industrial installation and compact, high performance hobby systems can be designed easily using this card "standard".

AN LED PROVIDES VISUAL INDICATION OF TAPE INPUT

To see that your tape recorder is feeding proper signals to KIM install permanently an LED in series with a 1.2 kohm resistor between R16 and ground. This point also appears on the expansion connector as E-X. Proper output of the tape recorder will generate a bright steady light. Voice or other signals coming from the tape recorder will make the LED flash or go dark.

Cass R. Lewart, 12 Georjean Dr., Holmdel, N.J. 07733



HARDWARE COMPARISON

KIMSI vs. KIM-4

Now that MOS Technology has reintroduced the KIM-4 Motherboard, I feel that you could benefit more from a comparison of these two KIM expansion alternatives than just a review of the KIMSI system alone.

The biggest difference right off the bat is that the KIMSI is set up to mate to the S-100 style bus while the KIM-4 has its own unique 44 pin bus. This immediately lets KIMSI owners expand to the plentiful and popular "S-100" boards. In that marketplace, competition among the many companies making boards to fit this bus configuration has forced the prices down while making many boards available. Of ocurse, you must realize that the S-100 was designed for the 8080 CPU with a front panel and the signals generated on the bus are far from 6502 compatible. The KIMSI handles the conversion from the simple 6502 timing to the rather complex 8080 timing, but it must be realized that since some manufacturers have chosen to deviate from the "not too well" defined S-100 bus the KIMSI can't possibly mate the KIM to all boards of this style. It does, however, allow KIMSI people to use most memory and video boards, which seem to be the most necessary anyway.

One of the disadvantages of the KIMSI is the method it uses to decode I/O ports in the system. Normally, the S-100 decodes I/O boards in a different way than it decodes memory. Because the 6502 has no special I/O instructions, all I/O devices must be mapped in the normal memory map. KIMSI designers placed this special section of memory up at the top 4K of KIM memory (F000-FFFA) which precludes the use of some good software in the KIMSI system. Namely KIMATH, the MOS assembler/editor from ARESCO and the disc system software from HDE. This could add up to a pretty serious disadvantage depending on you system usage. Also, the 4K section of memory map right below the KIM monitor is unusable in the KIMSI system. MOS Tech's KIM-4, on the other hand makes all of the memory (except) what's already used in KIM) available for use.

We might as well cover price comparisons while we're at it. To be fair, we have to consider comparable units. Since the KIM-4 comes assembled and includes 6 connectors, let's use that configuration for our example.

KIM-4, assembled and tested with 6 connectors costs $120.00

KIMSI, assembled and tested with 6 connectors costs $202.50

We must keep in mind that the KIMSI is also available as a kit for $125.00 and includes 1 connector. I purchased the kit version and had it up and running in several hours. It functioned perfectly the first time up, much to my surprise-after having built several kits in the past from other sources (including HEATHKIT) which required some debugging before things functioned correctly. The documentation that is included with the KIMSI seems to be adequate.

Much of the space is devoted(understandably) to the various S-100 boards which are compatible with KIMSI and some of the problems with those that aren't compatible. Several application notes are enclosed which outline methods of interfacing to two of the more popular video boards, other computer boards besides KIM, and even the KIM-2 or 3.

I have personally used Kent-Moore's 4K, 8K and video boards as well as Polymorphic's VTI-64 video board and Problem Solver's Systems 8K RAM board with the KIMSI motherboard. They all worked OK.

The KIM-4, on the contrary, doesn't enjoy such a great profusion of available accessory boards. This is showing signs of changing, though, and the future looks quite good. 8K RAM boards for the KIM-4 selling for around $190 and a floppy disc interface as well as a PROM board are now available. A look at the bus structure of the KIM-4 will indicate a fairly straightforward design which is much more easily understood than its S-100 cousin. This is an important consideration if you have any plans of using custom boards in you system. Also, it's possible to adapt one or more S-100 style boards to the KIM-4 bus by constructing a mating adaptor and making the proper electrical connections. S-100 cards and KIM-4 cards are exactly the same width.

My KIM-4 system is populated with the 8K RAM cards from Hudson Digital Electronics. This board comes in my favorite card size (4.5" x 6.0") and has recently been reduced in price to $195.00. Since these boards are narrower than the normally 10" wide KIM-4 size boards, a set of special card guides are necessary to fully mate the HDE boards to KIM-4. These guides are also available from HDE. Hopefully, more cards will be made available in this size for the KIM system, in the near future.

My 65XX "dream machine" will definitely use this size card.

To sum it up then, KIMSI users are able to utilize a good number of the very popular "S-100" style cards which are widely available at the price of losing some memory map usage at a critical part of KIM's memory map, namely the top 4K and having a much more complicated bus structure to have to design around. KIM-4 users have the disadvantage of not having an extremely wide assortment of boards to choose from (at the present time, anyway) BUT with a bus design so straighforward that building custom boards with parts from the 65XX or 68XX families are relativeley simple.

****

PRODUCT ANNOUNCEMENTS — FROM VARIOUS SOURCES

Several interesting flyers arrived from MICRO TECHNOLOGY UNLIMITED, Box 4596, Manchester, NH 03108. They are offering the digital-to-analog converter/music output board that was featured in Hal Chamberlin's magazine article (BYTE, Sept. 1977), a combination 8K memory and graphic output board with some unique sounding features, and a power supply for the KIM.

The 8K memory/graphic board (K-1008) uses 4K dynamic RAMS in such a way, according to the flyer, that is entirely transparent to the processor but visible to the user in the form of a 320x200 matrix of dots. (Maybe they solved the biggest hassle in using those low-cost "dynamics"?)

Total power for this board is specified at around 500 ma. and the price is $289.00 assembled and tested. Bare boards are $40.00.

The DAC/music board (K-1002) sells for $35.00 assembled and includes a listing of a 4-part harmony music program. Bare boards are $6.00.

The power supply has enough reserve to power a KIM and two of their memory/graphic boards.

Get more info from MTU at the above address.

SOFTWARE COMPARISON

The MOS Technology Assembler/Editor from ARESCO
vs.
The Micro-Ade Assembler/Disassembler/Editor
from Peter Jennings, Toronto

Micro-Ade is a two-pass assembler, editor, disassembler, and cassette operating system in one nicely integrated package. The program itself needs 4K of memory, (resides from $2000-$3FFF) is romable and sells for $50.00 with the complete source listing (which I recommend getting) or $25.00 with just the operating manual. Either way, you get it on a KIM cassette.

The biggest failing of Micro-Ade is the fact that it does not use the standard MOS Technology assembler mnemonics. This means that you can't assemble program instructions like you learned them in the 6502 Programming Manual.

Apart from that, Micro-Ade does boast a very adequate editor which commands such as: ADD, CLEAR, DELETE, END, FIX, INSERT, LIST, MOVE, NUMBER and WHERE. The assembler allows you to assemble from a source cassette to an object cassette for large programs or directly in memory for small programs. The cassettes can be relay controlled for automatic start/stop control or manually operated by making a few patches to the program. The cassettes can run up to 6 times normal KIM speed.

The MOS Technology Assembler/Editor distributed by ARESCO is a one-pass assembler, resides in 6K of memory (starting either at $2000 or $E000) and does not include a disassembler. The package sells for $70.00 on Kim cassette or paper tape and includes the complete source listing.

My biggest gripe with this assembler is that it is a one-pass style, which means that the assembler listing will not indicate the values for forward references. Futhermore, the assembler reserves two bytes for all forward references even though they may be one-byte instructions.

```
0110   022B   C9  61          CMP   #$61       ;LOWER CASE?
0115   022D   10  **  **       BPL   PRINT      ;YEP
0120   0230   4C  1D  02       JMP   NEXT       ;LOOP BACK
0125   0233   A5  02     PRINT LDA   $02        ;1ST BYTE
```

Apart from this one disadvantage, the MOS assembler boasts some very powerful features which become apparent only after having used both of these assemblers for a time. First of all, using Micro-Ade, all numbers must be entered in hexadecimal while the MOS assembler allows number entry in decimal, octal, binary, or hexadecimal. Both assemblers allow the use of Ascii literals. The MOS assembler also comes out on top when it comes to setting up byte tables. While Micro-Ade requires one line for each byte, the MOS assembler allows you to put as many bytes on a line as you desire as long as you don't exceed the 72 character line limit. This definitely saves alot of time if you use tables to any great extent.

Micro-Ade strikes back by allowing one to assemble programs anywhere in memory while its MOS counterpart allows you to assemble programs only where you have spare RAM. In other words, you can't assemble a program over the assembler with the MOS Assembler while you can with Micro-Ade because Micro-Ade installs all object code in a special file which is determined in advance by the programmer.

Another thing I don't like about Micro-Ade is the fact that it's field oriented, which means that you have to remember which field you are in when you enter source code. For example, if you are entering a label, an opcode, and a comment, you've got no problem, but, if you are entering only an opcode you have to space over to the opcode field and ditto if you are entering just a comment. I would imagine this would become second nature after ashile but I still goofed up on occasion even after using Micro-Ade for around four months. The MOS Assembler doesn't care anything about fields as long as you have a space between fields and if the line is just a comment, you have to precede it with a semi-colon.

So that's about how they stack up. Now you make the decision. They both have alot to offer and either one of them will make programming the 6502 one helluva lot easier.

****

REMEMBER 'SKEET SHOOT' (BY JIM BUTTERFIELD) FROM THE LAST ISSUE? WELL, LEW EDWARDS TIED IT TOGETHER WITH THE RON KUSHNIER NOISE GENERATOR (ALSO FROM THE LAST ISSUE) TO MAKE A NEAT DIVERSION...

....WAY TO GO, LEW....

Had a lot of fun fooling around with Ron Kushnier's sound effect routine. I took you up on the challenge to use it to add sound to Jim Butterfield's SKEET SHOOT which I have had for some time prior to publication in KUN. I modified the sound effect generator to suit, and used the time to display the "explosion". It worked out nicely because sometimes the "explosion" in the original form was so brief that you couldn't tell if you had a hit. I also changed location 0219 to 1F to increase the minimum speed of the target slightly. The following patch will add add sound to SKEET SHOOT if an amplifier is connected to PA0 (A-14). With sound, it's a hell of a lot more interesting.

Change 0272 to 12, and 0276 to 0E, and substitute the following:

```
0283 90 31          BCC PLOP     branch to sound patch
0285 38      SHINE  SEC          no hit flag
0286 B0 2E          BCS PLOP
0288 EA             NOP

          SOUND PATCH

02B6 8D 40 17 PLOP  STA SAD
02B9 8C 42 17       STY SBD
02BC B0 CB          BCS ZAP      no hit, no sound
02BE A9 60          LDA #60      starting pitch
02C0 85 DA          STA BURST
02C2 A9 01          LDA #01      open channel
02C4 8D 01 17       STA PADD2
02C7 EE 00 17 PULSE INC PAD2     toggle port 0
02CA A6 DA          LDX BURST    pulse time
02CC CA       TONE  DEX
02CD D0 FD          BNE TONE
02CF C6 DA          DEC BURST    raise pitch by decreasing
02D1 10 F4          BPL PULSE    time of each pulse that follows
02D3 30 C1          BMI ZAP-13   sound done, another target?
```

LEW EDWARDS end

Looking for some real world application for your toy...how about a DIGITAL CARDIOTACHOMETER.....from Marvin De Jong, Dept of Math, The School of the Ozarks, Point Lookout, MO 65726.....

I. The program:

The period between every two successive pulses is measured by counting the number of 10ms intervals which occur. The 10 ms intervals are produced by the interval timer on the KIM-1. Each pulse produces an interrupt (IPQ) which causes the KIM to convert the count to the traditional heartbeats per minute, and to display this number while it is measuring the next pulse period.

ADDRESS	INSTRUCTION			MNEMONIC		COMMENTS
0300	78			START	SEI	Disable interrupt.
0301	A2	01			LDX 01	
0303	8E	00	17		STX PAD	PAØ will be 1 when PADD = 1.
0306	8E	01	17		STX PADD	PAØ now is output pin., and
0309	EA				NOP	7474 is preset.
030A	CE	00	17	AGN	DEC PAD	7474 now can be clocked.
030D	A2	FF			LDX FF	Initialize counter to 255.
030F	86	00			STX COUNTER	
0311	58				CLI	Enable interrupt.
0312	A9	9C		LOOP	LDA 9C **	Start timer for 10 millisec.
0314	8D	06	17		STA TIMER	
0317	E6	00			INC COUNTER	Counter is incremented.
0319	20	1F	1F		JSR SCANDS	Display pulse rate.
031C	20	1F	1F		JSR SCANDS	Do it again.
031F	AD	07	17	CHEK	LDA TIMEOUT	Check timer, if not finished
0322	10	FB			BPL CHEK	branch to check again.
0324	4C	12	03		JMP LOOP	Start timer again.
0327	EA				NOP	
0328	EA				NOP	
0329	EE	00	17	IPQ	INC PAD	PAØ=1, 7474 preset.
032C	A5	00			LDA COUNTER	
032E	D0	03			BNE 03	If counter=0, go to AGN,
0330	4C	0A	03		JMP AGN	otherwise, continue.
0333	85	01			STA CNTLO	Set up double precision
0335	A9	00			LDA 00	add and subtract locations.
0337	85	02			STA CNTHI	
0339	85	F9			STA INH	Clear display registers.
033B	85	FA			STA POINTL	
033D	85	FB			STA POINTH	
033F	38			SUBT	SEC	Clear borrow flag.
0340	A9	66			LDA 66	Subtract from 1766_{16}=6000.
0342	E5	01			SBC CNTLO	
0344	A9	17			LDA 17	
0346	E5	02			SBC CNTHI	
0348	90	03			BCC BACK	If borrow, go to AGN,
034A	4C	51	03		JMP FWRD	Otherwise continue.
034D	58			BACK	CLI	
034E	4C	0A	03		JMP AGN	
0351	18			FWRD	CLC	Clear carry for double
0352	A5	01			LDA CNTLO	precision addition.
0354	65	00			ADC COUNTER	
0356	85	01			STA CNTLO	
0358	A5	02			LDA CNTHI	
035A	69	00			ADC 00	
035C	85	02			STA CNTH I	
035E	18				CLC	Clear carry flag for
035F	F8				SED	next addition, done in
0360	A5	F9			LDA INH	decimal. Set up display
0362	69	01			ADC 01	registers with pulse
0364	85	F9			STA INH	rate.
0366	A5	FA			LDA POINTL	
0368	69	00			ADC 00	
036A	85	FA			STA POINTH	
036C	D8				CLD	
036D	4C	3F	03		JMP SUBT	Try another subtraction.

++++++++++++++++ INTERRUPT VECTOR ++++++++++++++++++++++++++++++++++

17FE	29					
17FF	03					

** This number should be checked and adjusted to give precise 10 millisecond intervals. Only a rough check was made with an oscilloscope, so 9C may be slightly incorrect.

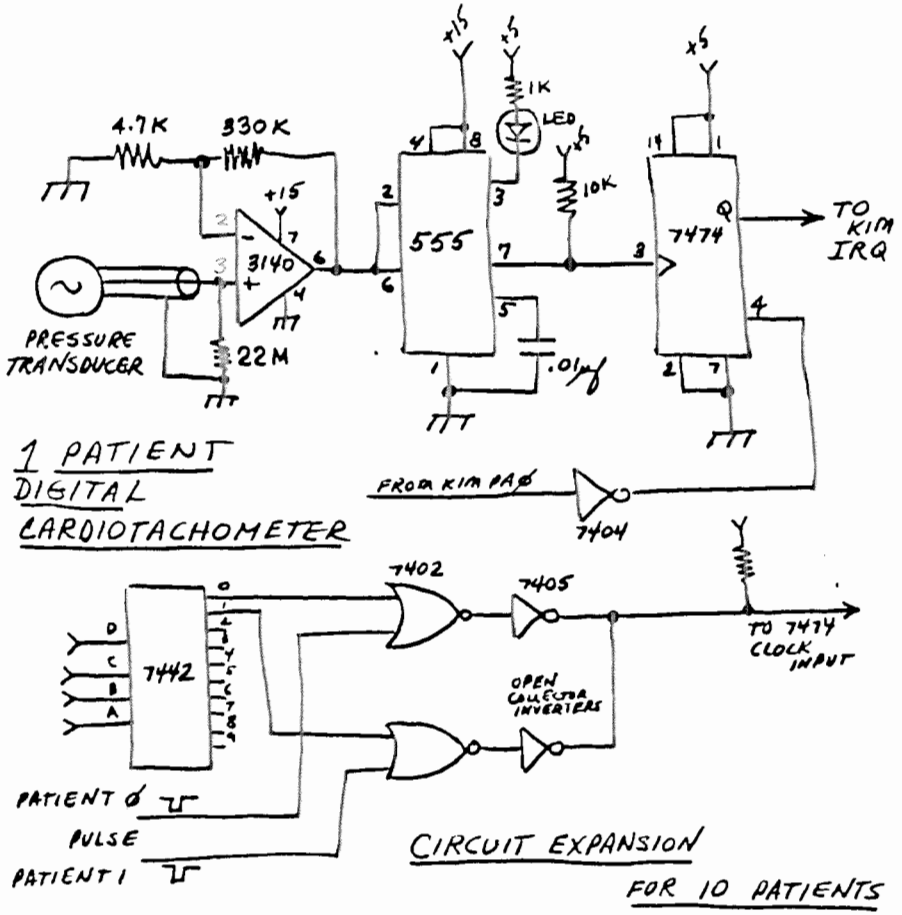
4

more

II. The interface circuit:

The transducer, an idea of Dr. Robert A. Pretlow, III, is a crystal earphone with the speculum removed and subsequently filled with silicone glue. The silicone should come in contact with the skin, and the earphone held snugly in place with tape. (An LED on one side of the fingertip and a photoresistor on the other will also produce a pulse signal which can be amplified and fed to a 555.) In the cicuit shown, an RCA 3140 (available from James Electronics) is used as an amplifier. The pulse signal is quite noisy so a 555 timer is used as a Schmitt trigger. TTL level signals are produced by a 10K pull-up resistor from pin 7 of the 555. The Q output of the 7474 produces an interrupt when connected to pin 4 of the KIM expansion connector. The interrupt is cleared by presetting the 7474 with a logical 1 on pin PA7. In the reset state of the KIM the interrupt will be cleared so the program can start. Without the 7404 inverter this would not be the case and the interrupt flag must be set by loading 04 in the status register.

The whole system can be expanded to say a 10 patient system with a 7442 decoder which, with the appropriate signal from Port PB7, would enable any one of 10 pulse signals to produce an interrupt.



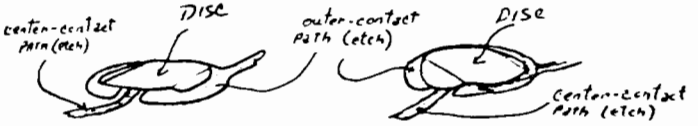
1 PATIENT DIGITAL CARDIOTACHOMETER

CIRCUIT EXPANSION FOR 10 PATIENTS

end

More on BOUNCY KEYS of the "old"style keyboard from Tim Bennett.

Thanks to Robert Dahlstrom for his article (see K.U.N. #10/11-9) on bouncy keys. In addition to this I had one other easily repairable problem which should be checked for prior to dis-assembly of your keyboard. Lightly wiggle each of your keys while observing the display. Ensure that no entry is made until a definite snap-action occures. If an entry is made prior to the snap-action, the internal disc for the offending key/keys should be rotated slightly so that the discs bent edges (which normally bridge the disc over the center-contact path) do not make contact with the "center-contact" path. If you find this fix necessary it should preceed the Dahlstrom fix as it will require lifting a portion of the clear tape to gain access to the disc.



Correct Disc Placement Poor Disc Placement

BOOK REVIEW

BY THE EDITOR

PROGRAMMING A MICROCOMPUTER: 6502

Author : Caxton C. Foster
Publisher: Addison-Wesley Publishing Co.

A few short months ago, if you wanted to learn about computer programming, you had to go to a book specifically about the 8080, or perhaps the 6800, and then translate to 6502 lingo all the way through the book. Admittedly, this is a great way to learn about microcomputers but, let's face it, some of us just don't have the patience for those kinds of mental gymnastics.

Finally, here's a how-to book written just for the 6502, and it uses the KIM no less!

PROGRAMMING A MICROCOMPUTER assumes you know nothing about micros and takes you through to writing an interpreter which makes the 6502 look like a 16 bit machine. He does this with a series of experiments designed to make clear all the esoteric computer jargon like "addressing modes", "table accessing with indexes", "semaphores", "interrupts", "parameter passing", "linked lists", etc. (I really wish that this book was available when I started into this field).

(EDUCATORS take note) This book is set up to be an excellent text book for classroom work using the KIM-1.

Some of the experiments consist of making music, programming a combination lock, running a two engine railroad on a single track, controlling an elevator, a computer ciphers, etc. Setting up and running these exercises (experiments) involves hooking up some garden variety transistors, resistors, LED's, etc. (nothing out of the ordinary).

Foster has a unique style of prose which enables him to impart some heavy information in alight and easy fashion.

All in all this is an excellent book. Very highly recommended.

It should be available at your local computer store.

ERIC

****

A LOW COST EPROM PROGRAMMER FOR KIM was mentioned in the last issue of the "Notes". After evaluating the unit we have come to the conclusion that for the money, you can't beat it. We programmed 2708's but it also can burn 2716's, according to the literature that accompanied the EP-2A-K EPROM PROGRAMMER from Optimal Technology. The documentation includes instructions to connect the unit to KIM as well as complete KIM software.

The price is $59.95 for the assembled unit or $49.95 for the kit (add $10.00 for a zero force programming socket).

The programmer is built on a 4.3" x2.2" pc board and includes the edge connector.

Now you can take advantage of the low price of 2708's at a reasonable price.

Get mor infor from: OPTIMAL TECHNOLOGY INC.
Blue Wood 127
Earlysville, Va 22936
After 1pm 804-973-5482

****

Here's our first FOCAL program-from Vince Coppola, 12 Charles St., Plantsville, Ct. 06479. Telephone 203-621-5954

I would like to announce that I have Focal-65 (available from the 6502 Program Exchange, 2920 Moana Ln., Reno, Nev 89509) on my KIM system, in 5K of memory. My memory is contiguous, from $0000 to $13FF. Normally, FCL-65 resides in $0000-0696 and $2000-$3082 approx. The Program Exchange group made me a version that resides in my system. It occupies $0020-$00D4 and $0200-$128A.

FCL-65 occupies about 4.7K, so it leaves only some 300 bytes of program space in a 5K system. I later plan to add another 4K of memory starting at $2000-2FFF, and use that for program space. But for now I am using only the 300 bytes--and it is really surprising the programs you can write in that small area, because of the power of FCL-65. To prove this, I am sending along this program that I whipped up, and in no way do I claim to be a programmer. One note I would like to make: To do an exponential function in FCL-65, you need the symbol "↑" = 5E, which is not available on my keyboard. I had to change it to a key I did have, so I looked into the cross-listing in-order to change its value. It is located at $11C6 in this low version of FCL. It is located in $2FC6 in the version that starts at $2000.

(editors addendum: Vince has the early version of FOCAL in his system. In version 3D, the exponential symbol is located in $34ED).

Example on how the enclosed program works:
You take out a loan from a bank at the amount of $24000.00. It is borrowed for a term of 30 years (360 months), at an interest rate of 9.25% per annum. What is your monthly payment?

```
C FOCAL-65 (V3D) 26-AUG-77

1.01 A "TOTAL LOAN=$",A
1.02 A "%/YEAR=",P
1.03 A "# OF MONTHLY PAYMENTS=",N
1.10 S W=(1+(P/1200))↑N
1.20 S X=1-(1/W)
1.30 S Y=X/(P/1200)
1.40 S R=A/Y
1.50 T "YOU PAY $",R," A MONTH"!
1.60 T "TOTAL PAID AFTER ",N/12," YEARS IS $",N*R
1.70 Q

#G
TOTAL LOAN=$24000
%/YEAR=9.25
# OF MONTHLY PAYMENTS=360
YOU PAY $  197.44211 A MONTH
TOTAL PAID AFTER     30.00000 YEARS IS $71079.15910
```

exponential function

...MORE ON FOCAL from the editor....the biggest appeal of FOCAL is that, besides being a fairly powerful math oriented language, a complete source listing is provided. This has two immediate advantages--first, its now possible to see just how a high level language is constructed (a very valuable experience) and --second, digging in to modify it, debug it, or extend it is now trivial (once you understand it, of course) The biggest disadvantage of FOCAL is that ,in my version anyway, saving programs and data on cassette (or disc, for that matter) is a function not included in the language. That seems to be left up to the user.

Has anyone figured out how to do this?? If so, please let the rest of us in on this procedure. If their is enough interest, maybe we could have a section of the 'NOTES dedicated to information on this language. Lets hear from YOU!!!

How 'bout a JOYSTICK INTERFACE? Heres one from Roy Flacco (remember the graphics interface?) By the way, Roy brought his Kim and graphics interface over to a local KIM user group meeting for a demonstration of 6502 power. His Lunar Lander and pattern generator were the life of the party and quite impressive. Thanks alot Roy.....

Here's the analog input circuit I promised you a while back. Essentially it converts an analog voltage in the range 0 to +2.55 volts into an 8-bit digital number which is presented to KIM via the applications connector. In deciding to do many functions in hardware I chose speed and simplicity of software over simplicity of hardware...most of the logic in the circuit could be done by KIM but would tie up the processor doing dumb (?) things. The cost is about $12 to $15 per channel depending on your suppliers. I happened to have 8212 latches available, but using a 74100 cuts the cost by $3 per channel, though you must add Tri-state buffers.

I constructed two of these ADC's on a 4x6 vectorcard with plenty of space for my usual point-to-point wiring and they have run without a hitch since the first power-up.

Circuit Description

The circuit is a straightforward single-slope ramp generator with a 311 comparator and latching on the digital outputs. The 2N4258 is the same DAC/ADC chip used in my point-plot graphics board (KUN 10/11) and is still available for $8 from Ferranti Electric Inc., East Bethpage Rd., Plainview, NY 11803. They tell me it will be an off-the-shelf stock item for a long time, and I can easily see why. I'm using them for all sorts of things including analog X digital multiplication, complex waveform generation, etc.

The comparator compares the analog voltage output of the 425 to the applied voltage V_{in}, and as long as V_{in} is greater it allows the gate/divider FF4 to pass clock pulses to the 8-bit counter in the 425. This incrementally increases V_{out}. At the point where V_{out} (from the 425) exceeds Vin, the 311 changes state and initiates the sequence diagramed in the schematic.

At time t_1 the pulse which will cause the 311 to change is being generated by FF4. This is ③. When it falls, the 425 internal counter increments, and V_{out} exceeds Vin by less than 10 millivolts.

The 311's output goes high at t_2 and forces FF4 inactive; hence no more counts are recorded.

At t_3 the clock pulses from FF1 (which is driven from Ø2) cause the output of FF2 to go high for exactly one pulse, which is used to strobe the data into the 8-bit latch. This is ④.

At t_4 the strobe pulse causes FF3 to go active, and the $\overline{Q}$ output is used to reset the 425's counter. This is ⑤.

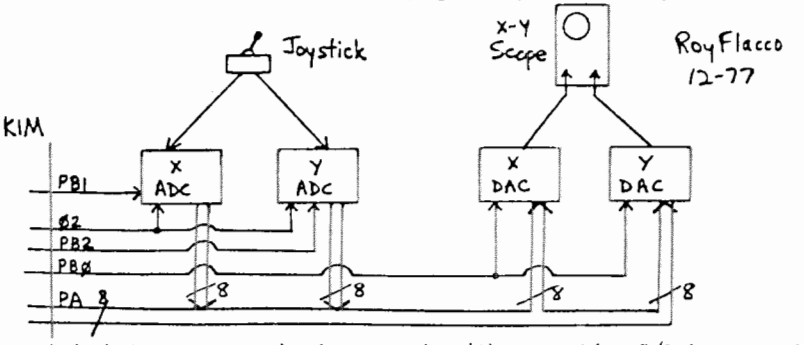
Because the internal counter is now zero, the 425's analog voltage output V_{out} is also zero, and the comparator changes state back to the original condition. This frees FF4 to once again generate clocking pulses for the 425. The pulse in ③ at t_5 is the first such pulse. The counter counts up to the digital value again and the data in the latches is updated automatically at the end of the cycle again.

The 311 is wired to produce the lowest offset voltage for inputs near ground (always a problem when running from only +5 volts); the 24 pf capacitor speeds up the change of state and the diode protects the inputs. The npn transistors can be almost anything (as can the pnp buffer at the latch). I used 74107's for the flip-flops because they were handy and cheap; if another type of flop is used the timing and logic connections might have to be altered since not all flops work the same.

Since I was building two identical circuits on the same board I chose to have one FF1 in common and run one channel from each of the complementary outputs Q and $\overline{Q}$. I assumed this would reduce the size of the current spikes in Vcc as the flip-flops changed since one channel was exactly out of phase with the other. While I did not try it the other way I would reccommend doing the same if you intend to have multiple channels on a board. Noise spikes are a loser around analog as well as digital.

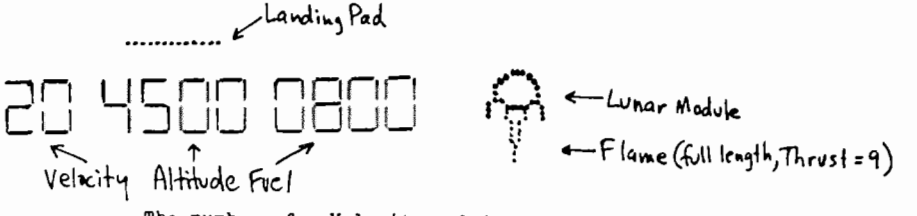
Note that if you use 74100's for latches and intend to have more than one channel you have to multiplex the outputs since the 74100 is not Tri-state (the 8212's are).

In my own setup I have two channels of ADC with separate Tri-state latches, and two channels of DAC (the graphics board), all data bussed together on the PA peripheral bus (PAØ-PA7). This allows all input and output to pass through PA. The strobes on the graphics board are controlled by PBØ, PB1 enables the X-latch (channel one of the joystick ADC), and PB2 enables the Y-latch (channel two). Thus without dedicating PA to any particular board, and using only three bits of PB, I have a complete X/Y graphics I/O interface.



And what, you may ask, does one do with a graphics I/O interface? Well, the first thing is calibrate the joysticks for fullscale=FF. I've included a short routine which displays the instantaneous values of the X and Y ADCs in the LED displays for ease in adjusting the trimpots. Also included is a routine which I call the Joystick Auto-Erasing Sketcher. This is a good demonstration of the value of having high-speed ADCs. It samples both X and Y every 10 milliseconds and updates a list of the most recent 256 values of X and Y, then displays the entire list (which is what takes 10 milliseconds). The effect is that of a long streamer trailing out from the dot which corresponds to the joystick's present position. Because the list is constantly being updated, the oldest data (actually about 2½ seconds old) is replaced by the newest, and the streamer erases itself automatically. Nifty toy, indeed; it has obvious applications, though in terms of menu selection, prototype drawing, even a storing Etch-a-Sketch display. That would admittedly take more memory, though, since every point is stored as two bytes.

My real pride and joy, though, is an adaptation of Jim Butterfield's incredible Lunar Lander Program (KUK and First book of KIM). This was altered to allow graphic presentation of all vital data simultaneously (Altitude, Velocity, and Fuel) in digital form, while at the same time displaying a Lunar Lander Module and landing pad. As the really nice touch, the joystick is used as a throttle to instantaneously control the Thrust, which is displayed as a variable-length flame under the Lunar Module. On the scope CRT this appears:



The numbers for Velocity, Altitude, and Fuel are the same as JB concocted for the original Lunar Lander, and the arithmetic routines are entirely his.

The altitude in decimal is converted into hex and used as an offset for the lander's height, so that as the altitude decreases, the module sinks slowly toward the landing pad. As you move the throttle the flame grows or shrinks, and of course the numbers change in the same way as the original lander program. All in all a very dynamic display and a good example of the value of high speed I/O.

...e routines for processing data for graphic/numeric display are similar in use to the KIM monitor routines, and in fact can be started easily to display 6 digits of seven-segments each in a 4/2 grouping, exactly like the KIM LEDs.

A Suggestion for the Graphics OUTPUT Board from PIM 10/11. If you find the outputs settle too slowly and blur the display try buffering them with 3140 BiFET op amps running on just +5. The 425 chips are not meant to drive long lengths of coax or high capacitance.

JOYSTICK FULL-SCALE CALIBRATOR — Roy Flacco

```
A2 FF       CAL  LDX #$FF      set PB= all outputs
8E 02 17         STX PBDD
8E 02 17         STX PBD       disable all latches
AA               TAX           X=Ø
8E B4            STX DIR       DDR=ØØ
8E Ø1 17         STX PADD      set PA to all inputs
A2 FB       LOOP LDX #$FB      disable latch 2, enable latch 1
8E Ø2 17         STX PBD
AD ØØ 17         LDA PAD       get A/D 1 data
8E FA            STA DISPLY
A2 FD            LDX #$FD      disable latch 1, enable latch 2
8E Ø2 17         STX PBD
AD ØØ 17         LDA PAD       get A/D 2 data
8D F9            STA DISPLY+1
A2 FF            LDX #$FF      disable both latches
8E Ø2 17         STX PBD
8D 1F 17         STA $171F     display latch contents
18 90            BCC LOOP
D0 F9            BNE LOOP
```

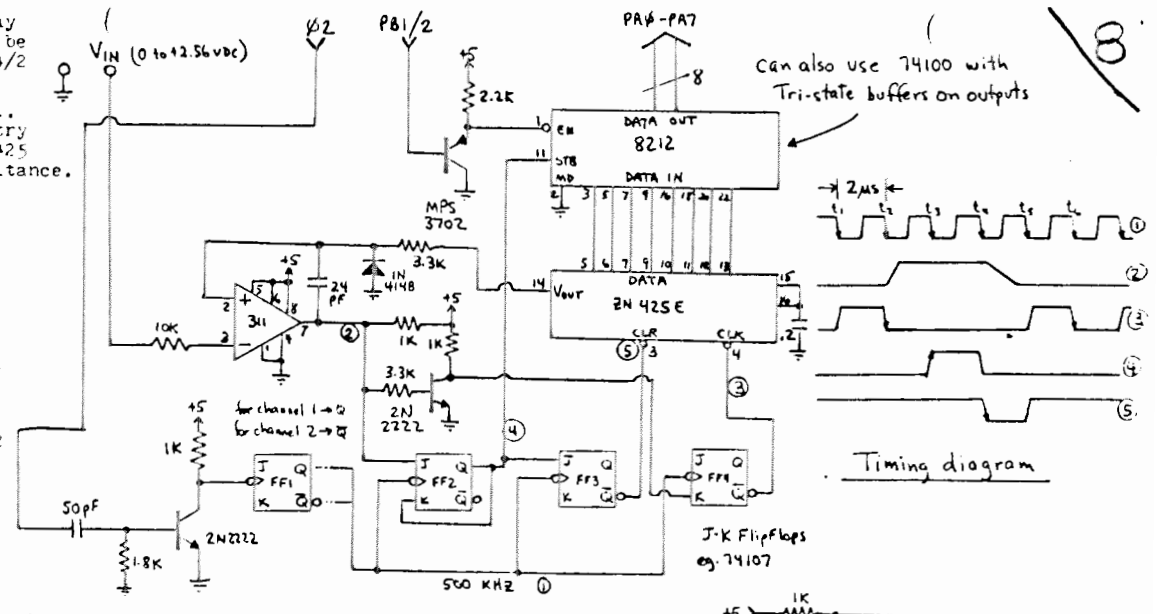
Because this program is fully relocatable, where you put is entirely up to you. I usually put it up at 1780.

JOYSTICK ALL-ERASE SKETCHER — Roy Flacco

```
Ø1ØØ  A2 3F     SKETCH  LDX #$3F
Ø1Ø2  8E Ø2 17          STX PBDD    PB=all outputs
Ø1Ø5  8E Ø2 17          STX PBD     all disabled
Ø1Ø8  E8                INX         X=Ø
Ø1Ø9  8E Ø1 17  UPDATE  STX PADD    PA=all inputs
Ø1ØC  A2 FB             LDX #$FB    enable the Y latch
Ø1ØE  8E Ø2 17          STX PBD
Ø111  AD ØØ 17          LDA PAD     read Y (channel of ADC)
Ø114  99 ØØ Ø2          STA $Ø2ØØ,Y store in page 2 indexed by Y
Ø117  A2 FD             LDX #$FD    enable X latch
Ø119  8E Ø2 17          STX PBD
Ø11C  AD ØØ 17          LDA PAD     read X (channel of ADC)
Ø11F  99 ØØ Ø3          STA $Ø3ØØ,Y store in page 3 indexed by Y
Ø122  C8                INY
Ø123  A2 FF     OUTPUT  LDX #$FF    disable latches
Ø125  8E Ø2 17          STX PBD
Ø128  8E Ø1 17          STX PADD    PA=all outputs
Ø12B  E8                INX         X=Ø
Ø12C  BD ØØ Ø2    LOOP  LDA $Ø2ØØ,X read a Y-coordinate
Ø12F  8D ØØ 17          STA PAD     load into the Y DAC latch
Ø132  CE Ø2 17          DEC PBD     strobe
Ø135  BD ØØ Ø3          LDA $Ø3ØØ,X read an X-coordinate
Ø138  8D ØØ 17          STA PAD     load into the X DAC latch
Ø13B  EE Ø2 17          INC PBD     strobe
Ø13E  E8                INX
Ø13F  D0 EB             BNE LOOP    done?
Ø141  F0 C4             BEQ UPDATE  get a new point. X=Ø
```

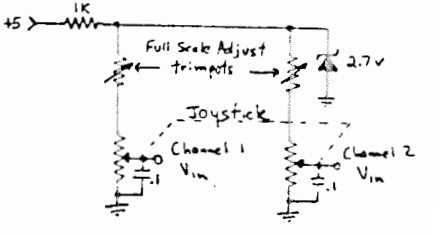
Note that if PB7 is tied to the IRQ line, bit 7 of PBDD must be left as an input, otherwise it causes strange interrupts.

The program is fully relocatable, but of course if you move it into pages 2 or 3 you must find somewhere else to store the data. Either page 1 or the 17ØØ space is suggested for this routine.



FREE-RUNNING VOLTAGE A/D CONVERTER

(one of 2 channels) R. Flacco 12-77



SPACE DOES NOT PERMIT PRINTING ALL OF ROY'S ARTICLE IN THIS ISSUE. PART TWO OF THE ARTICLE WILL BE THE COMPLETE LISTING OF THE SCOPE LUNAR LANDER PROGRAM.

....MORE FROM HDE

Hudson Digital Electronics has announced that purchasers of the File Oriented Disc System can now request a version set up especially for the KIMSI (S-100) system.

HDE says they will supply a relocated version of the FODS software as well as instructions on how to adapt the disc interface board to the S-100 bus.

BASIC programmers will be happy to hear that HDE is including a BASIC linker program in their documentation to interface MICRO-SOFT BASIC to the FODS software.

I've used this BASIC linker program and appreciate having the ability to save and load BASIC programs by name. The version of BASIC used is from Johnson Computer, P.O. Box 523, Medina, Ohio 44256.

This version of the linker will not allow you to save BASIC data files but it is intended that later versions will have this capability.

YOU'LL HAVE TROUBLE KEEPING KIM OUT OF THE HAMSHACK AFTER TRYING THIS MORSE CODE READER PROGRAM. THIS ROUTINE RAN FINE EVEN ON MY RELATIVELY SLOW (300 baud) TERMINAL. SHOULD BE GREAT WITH A FAST VIDEO TERMINAL OR MEMORY MAPPED DISPLAY. I HAVENT TRIED THE INTERFACE CIRCUIT YET, BUT IT LOOKS LIKE IT SHOULD WORK ALRIGHT.....Eric

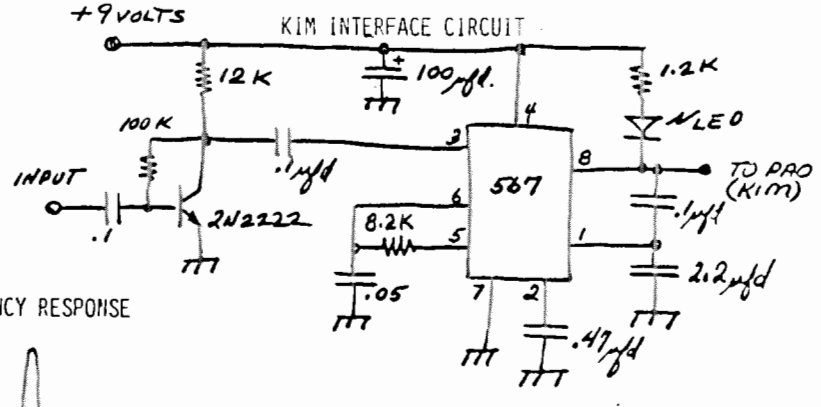
BY THE WAY, THIS PROGRAM COMES FROM BOB KURTZ, MICRO-Z CO., Box 2426, Rolling Hills, California 90274

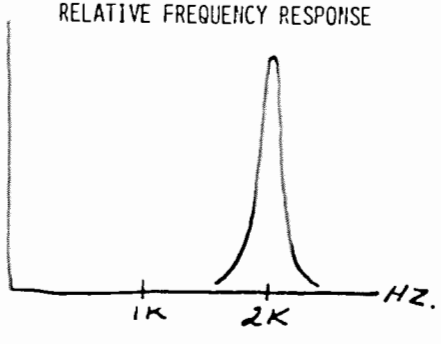
```
0200-  AD 00 17    LDA  1700   } WAIT FOR KEY DOWN
0203-  29 01       AND  #01    }
0205-  D0 F9       BNE  0200
0207-  A9 00       LDA  #00    } YES - THEN DASH & DOT REGISTERS TO ZERO
0209-  85 04       STA  04     }
0208-  85 05       STA  05     }
020D-  A9 00       LDA  #00    } 'TIME' TO ZERO
020F-  85 06       STA  06     }
0211-  20 0F 02    JSR  020F   } START TIMING
0214-  E6 06       INC  06     }
0215-  AD 00 17    LDA  1700   } KEY UP?
0219-  29 01       AND  #01    }
021B-  F0 0E       BEQ  022B NO - JUMP FORWARD
021D-  06 06       ASL  06     } YES - ① UP-DATE TIME
021F-  20 9A 02    JSR  029A   }   ② STORE DOT
0222-  06 05       ASL  05     }   ③ GO TO 'KEY-UP'
0224-  06 04       ASL  04     }
0226-  E6 05       INC  05     }
0228-  4C 4C 02    JMP  024C   }
022B-  A5 03       LDA  03     2X DASH TIME
022D-  0A          ASL  A
022E-  65 03       ADC  03     + DASH TIME
0230-  4A          LSR  A      ÷ 4 = 3/4 DASH TIME
0231-  4A          LSR  A
0232-  CD 06 00    CMP  0006 'TIME' LESS THAN THIS?
0235-  B0 DA       BCS  0211 YES - GO BACK
0237-  06 05       ASL  05   NO -
0239-  06 04       ASL  04     STORE A DASH
023B-  E6 04       INC  04
023D-  20 0F 02    JSR  020F } ADD MORE 'TIME'
0240-  E6 06       INC  06   }
0242-  AD 00 17    LDA  1700 } KEY UP YET?
0245-  29 01       AND  #01  }
0247-  F0 F4       BEQ  023D NO - MORE TIME
0249-  20 9A 02    JSR  029A YES - UP-DATE DASH TIME
024C-  A9 00       LDA  #00  } 'TIME' TO ZERO
024E-  85 06       STA  06   }
0250-  20 0F 02    JSR  020F } START TIMING
0253-  E6 06       INC  06   }
0255-  AD 00 17    LDA  1700 } KEY DOWN?
0258-  29 01       AND  #01  }
025A-  F0 B1       BEQ  020D YES - BACK TO START - CHARACTER NOT COMPLETE
025C-  A5 03       LDA  03   NO -
025E-  0A          ASL  A      2X DASH TIME
025F-  65 03       ADC  03     + DASH TIME
0261-  4A          LSR  A    } ÷ 4 = 3/4 DASH TIME
0262-  4A          LSR  A    }
0263-  C5 06       CMP  06   'TIME' LESS THAN THIS?
0265-  B0 E9       BCS  0250 YES - GO BACK
0267-  A5 04       LDA  04   NO -
0269-  2A          ASL  A      } DEVELOP LOOK-UP NUMBER
026A-  65 05       ADC  05     }
026C-  AA          TAX
026D-  BD AA 02    LDA  02AA,X   LOOK-UP CHARACTER
0270-  20 A0 1E    JSR  1EA0     AND PRINT IT OUT
0273-  20 0F 02    JSR  020F  } ADD 'TIME'
0276-  E6 06       INC  06    }
0278-  AD 00 17    LDA  1700  } KEY DOWN YET?
027B-  29 21       AND  #21   }
027D-  D0 03       BNE  0282
027F-  4C 07 02    JMP  0207 YES - BACK TO START - CLEAR REGISTERS
```

```
0282-  A5 03       LDA  03   NO -
0284-  0A          ASL  A      2X DASH TIME
0285-  C5 06       CMP  06   'TIME' LESS THAN THIS?
0287-  B0 EA       BCS  0273 YES - MORE TIME
0289-  20 9E 1E    JSR  1E9E NO - PRINT SPACE (END OF WORD)
028C-  4C 00 02    JMP  0200 GO BACK AND WAIT FOR 'KEY DOWN'
028F-  A0 05       LDY  #05  } TIMER ROUTINE (TIME WASTER)
0291-  A2 FF       LDX  #FF  }
0293-  CA          DEX       }
0294-  D0 FD       BNE  0293 }
0296-  88          DEY       }
0297-  D0 F8       BNE  0291 }
0299-  60          RTS       }
029A-  A5 03       LDA  03   ——— UPDATE DASH TIME ROUTINE
029C-  0A          ASL  A      2X DASH TIME (OLD)
029D-  65 03       ADC  03     + DASH TIME (OLD)
029F-  65 06       ADC  06     + NEW TIME
02A1-  4A          LSR  A      ÷ 2
02A2-  4A          LSR  A      ÷ 2   = (3 X OLD) + 1 NEW / 4 = 'WEIGHTED' UP-DATE
02A3-  85 03       STA  03
02A5-  60          RTS
```

NOTE: ① 'PROGRAM' RESIDES FROM $0200 (H) TO $02A5 (H).
② 'LOOK-UP' TABLE RESIDES FROM $02AA (H) TO $02FF (H)
③ KEY DOWN TO PA0

```
     0  1  2  3  4  5  6  7  8  9  A  B  C  D
0200 AD 00 17 29 01 D0 F9 A9 00 85 04 85 05 A9 00 85
0210 06 20 0F 02 E6 06 AD 00 17 29 01 F0 0E 06 05 20
0220 9A 02 06 05 06 04 E6 05 4C 4C 02 A5 03 0A 65 03
0230 4A 4A CD 06 00 B0 DA 06 05 06 04 E6 04 20 0F 02
0240 E6 06 AD 00 17 29 01 F0 F4 20 9A 02 A9 00 85 05
0250 20 0F 02 E6 06 AD 00 17 29 01 F0 B1 A5 03 0A 65
0260 03 4A 4A C5 06 B0 E9 A5 04 0A 65 05 AA BD AA 02
0270 20 A0 1E 20 0F 02 E6 06 AD 00 17 29 01 D0 03 4C
0280 07 22 A5 03 0A C5 06 B0 EA 20 9E 1E 4C 00 02 A0
0290 05 A2 FF CA D0 FD 88 D0 F8 60 A5 03 0A 65 03 65
02A0 06 4A 4A 85 03 60 [redacted] 03 45 54 49 41 4E
02B0 4D 53 55 52 57 44 4B 47 4F 48 56 46 28 4C 28 50
02C0 4A 42 58 43 59 5A 51 20 20 35 34 20 33 20 20 20
02D0 32 20 20 20 20 20 20 20 31 36 2D 2F 20 20 20 20
02E0 20 37 20 20 20 38 20 39 30 20 20 20 20 20 20 20
02F0 20 20 20 20 20 3F 20 20 20 20 20 20 20 20 2E 20
```





EVERY (OFTEN, USER NOTES WILL PURCHASE EQUIPTMENT FOR EVALUATION OR JUST (SE AND THEN FIND ITS NOT GETTING THE USE IT SHOULD. NOWS YOUR CHANCE TO PICK UP SOME QUALITY STUFF AT REASONABLE PRICES. HELP ME TO CLEAR A PATH INTO MY COMPUTER ROOM. DOCUMENTATION AND UPS SHIPPING IS INCLUDED ON ALL ITEMS UNLESS OTHERWISE SPECIFIED.

MEMORY-PLUS (FROM THE COMPUTERIST) 8K OF 2102 STYLE RAM. SOCKETS FOR 4 2716'S. PROGRAMMING CAPABILITY FOR 2716'S, AND A 6522 VIA I/O PORT. USED VERY LITTLE, WORKS GREAT. REVIEWED IN LAST ISSUE. $210.00

KIMSI MOTHERBOARD COMPLETE WITH 6 S-100 EDGE CONNECTORS AND KIM CONNECTOR. A GOOD WAY TO TAKE ADVANTAGE OF ALL THAT S-100 STUFF. READY TO RUN. $145.00

8K RAM FOR S-100 BY KENT-MOORE. COSTS $197.00 NEW. SELLING FOR $160.00 WORKS GREAT ON KIMSI.

8K RAM FOR S-100 BY PROBLEM-SOLVERS.WORKS GREAT ON KIMSI SELLING FOR $150.00

32x16 VIDEO DISPLAY FOR S-100 BY KENT-MOORE. MEMORY MAPPED WITH 8080 DRIVER. WILL INCLUDE A 6502 DRIVER. $75.00

64x16 VIDEO DISPLAY FOR S-100 BY POLYMORPHICS. ASSEMBLED FROM KIT BY SUPER CAREFUL ENGINEER. WORKS GREAT IN KIMSI. WILL INCLUDE A 6502 DRIVER PROGRAM $295.00

TVT-6 FOR KIM. NEVER ASSEMBLED. STILL IN SEALED CONTAINER. $26.00

9" VIDEO MONITOR BY BALL BROTHERS. A-1 COND. USED VERY LITTLE. TAKES SEPERATE VIDEO, VERTICAL, AND HORIZONTAL INPUT SIGNALS (LIKE PET) AND HAS 12 MHZ. BANDWIDTH FOR HIGH DENSITY DISPLAYS. $60.00

KIM ENCLOSURE BY THE ENCLOSURE GROUP. FOR OLD-STYLE KEYBOARD. NEW $19.00

VECTOR GRAPHIC CRT DISPLAY TERMINAL BY SANDERS. FEATURED IN BYTE AND '73 MAGAZINE FOR GRAPHICS CONVERSION. THIS TERMINAL HAS BEEN TESTED AND FOUND TO BE IN OPERATIONAL CONDITION. THE GRAPHICS INTERFACE PORTION IS INCLUDED IN THE DEAL AND INCLUDES EVERYTHING NEEDED TO TURN THIS THING INTO A VECTOR GRAPHICS TERMINAL. (a vector terminal is one which draws lines to connect points on a screen instead of using dots to connect the points like some conventional oscilloscope interfaces. the resolution available on a true vector display is fantastic) ALL THAT'S NEEDED TO BRING THIS DISPLAY UP IN ITS FULL GLORY IS A LITTLE WORK IN SETTING UP THE INTERFACE BOARDS D/A CONVERTERS. I WOULD PREFER THAT YOU PICK UP THE UNIT BECAUSE OF ITS WEIGHT (70 LBS) AND BULKINESS. THE PRICE OF $100.00 INCLUDES FULL DOCUMENTATION AND A HAND GETTING IT OUT TO YOUR CAR.

SEND A SELF ADDRESSED STAMPED ENVELOPE WITH YOUR CERTIFIED CHECK OR MONEY ORDER AND YOUR PAYMENT WILL BE RETURNED IN THE EVENT THAT SOME EARLY BIRD BEATS YOU TO A GOOD DEAL.

FOR MORE INFORMATION ON ANY OF THIS STUFF, CALL OR WRITE...

ERIC REHNKE 109 CENTRE AVE, NORRISTOWN PA 19403 (NOTE NEW ZIP)
HOME PHONE- 215-631-9335 BETWEEN 7 AND 9 PM.

RANDOM ACCESS CORNER

BACK ISSUES of the 'NOTES are still available from Mark Kantrowitz, 15 Midway Ct., Rockaway, NJ 07866. Issues 1-6 are available for $6.50 (third class mail), $7.00 (first class mail), and $12.00 (overseas airmail).

Would like hardware and software for interfacing KIM to a Texas Instruments 5050M calculator. John Connely, 16W260 W. 83 rd St., Hansdale, Ill., 60521

Before using GETKEY (1F6A), initialize PADD (1741) with $00 for input or strange things will happen. Gary Grzebienik, 22600 W. Outer dr., Dearborn, MI 48124

LOCAL KIM USER CLUB getting started in the San Fernando Valley area. Anyone interested should contact--Jim Zuber, 20224 Cohasset #16, Canoga Park, CA 91306 (213) 341-1610.

FORTRAN CROSS ASSEMBLER for the 6502. This 2-pass assembler runs on any FORTRAN GP computer with 18K or more core and some temporary file storage (floppy disc) Outputs hex code for target machine. Manuals listings and examples available for $20 handling charge from Fred Osborne, 6315 Mill Pond Rd., Byron, NY 14422

FOR SALE-KIM-3 8K RAM board..new condition with all documentation and original packaging--$200. J.C. Williams, 35 greenbrook DR. Cranbury, NJ 08512

LOCAL KIM USER CLUB getting started in the ITHACA NY area. Contact Roy Flacco, 200 Highland Ave., Ithaca NY 14850.

COSMAC 1802 simulator program runs on KIM and lets you develop 1802 software. All internal 1802 regs may be examined in either trace or single step modes. Documentation includes KIM cassette, user manual, and source code for $11.50 (includes postage & handling) Dann McCreary, 4758 Mansfield St. #2H, San Diego, CA 92116

TVT-6 ENTHUSIASTS TAKE NOTE---I'D LIKE TO DEVOTE EITHER OF THE NEXT TWO ISSUES OF THE 'NOTES TO ARTICLES, COMMENTS, SOFTWARE, AND THE LIKE ABOUT THE FAMOUS TVT-6. I WON'T BE ABLE TO VERIFY CORRECT OPERATION OF HARDWARE OR SOFTWARE FOR THE TVT-6 SO PLEASE DOUBLE CHECK YOUR LISTINGS AND SCHEMATICS.

AUTHORS NOTES: ALL ARTICLES SHOULD BE TYPED SINGLE-SPACED USING A NEW RIBBON AND 8" WIDE COLUMNS. DRAWINGS AND SCHEMATICS SHOULD BE DONE WITH BLACK INK (A FELT TIP PEN WORKS GOOD)

A couple of thoughts from Andy Chakires, 5738 Waring Ave, Los Angeles CA 90038

Good ol' SST switch, sitting there black sunk into black, and further made difficult to see because KIM's display likes the shadows. If you're new to KIM (like me) you foul up because you forget to turn it off. Do this. Paint the switch's top and the ridges of the letters ON with, say, white correction fluid such as Liquid Paper used by typists.

Add Sears 57-3417 Cassette Recorder to the list that KIM likes. Works perfectly with Memorex MRX2 and Butterfield's Hypertape. This audio recorder sold in the $70-$90 range in 1973-74 and can now be occasionally found at Sears Catalogue Surplus Stores, stripped with mfg. model 564.34202200 or similar.
Output voltage is -7.5. The owner's manual includes a complete schematic.

INTERFACING THE SWTPC PR-40 PRINTER TO THE KIM-1

by Jim Zuber
20224 Cohasset
Canoga Park, CA
91306

The PR-40 printer is a 40 column, 75 line per minute matrix printer. It is the lowest cost printer ($259.) on the market today and is very easy to interface to the KIM-1. Wire the KIM application port to the printer buss in the following manner:

KIM		PR-40
PA0	to	ASCII Bit 0
PA1	to	ASCII Bit 1
PA2	to	ASCII Bit 2
PA3	to	ASCII Bit 3
PA4	to	ASCII Bit 4
PA5	to	ASCII Bit 5
PA6	to	ASCII Bit 6
PB0	to	DATA READY
PB1	to	DATA ACCEPTED
GROUND	to	GROUND

I found that the easyest way to set up the software interface was to set up a 40 character buffer in page 0 of the KIM memory (loc 0050-0077). The following subroutines manipulate and print this buffer area:

1. Clear buffer subroutine (1780-1789)-loads the ASCII character "20" (space) into locations 0050 to 0077.

2. Initalize printer subroutine (178A-17AE)-sets the data direction registers for ports "A" and "B", intiates a carrage return on the printer, and calls the clear buffer subroutine.

3. Load buffer subroutine (0100-010F)-picks up ASCII data from any location in memory, and loads the ASCII data into any location in the buffer. The following items must be defined in memory before calling this subroutine:

   007B starting location in memory for
   007C ASCII data to be picked up
   007D number of characters (in hex) to be picked up and loaded
   0079 starting location in buffer to load ASCII data (must be between 50 and 77 hex)

4. Print buffer subroutine (17AF-17E0)-outputs and prints data stored in the buffer and calls clear buffer sub after printing is completed.

5. Hex to ASCII subroutine (0117-0143)-converts the hex number loaded in 0009 into two ASCII characters, which are stored in 000E and 000F.

The subroutines referenced above are included in the following hex dump program for the KIM. To use the program load the first address you want to list (low order first) into 000A and 000B, then load the ending address into 000C and 000D. Start the program at 0144 and the printer will give you a hex dump. Although the formating used in the hex dump is unconventional, it works and it beats the hell out of doing it by hand. The following hex dump was done using this program.

```
0100  A2 00 B5 7A A0 00 B1 78 91 79 C8
0108  C6 7D D0 F7 60 A0 00 B1 0A 85 09
0110  60 A5 05 85 08 29 0F 85 09 66 08
0121  66 08 66 08 66 08 A5 08 29 0F C9
0120  0A 18 30 02 69 07 69 30 85 0E A5
0137  09 C9 0A 18 30 02 69 07 69 30 85
0142  0F 60 20 6A 17 EA EA EA A9 0E 85
0140  7B A9 00 85 7C A9 50 85 79 A9 82
0150  85 7D A5 06 25 09 20 17 01 20 00
0160  01 A9 5C 85 79 A9 82 85 7D A5 08
016E  85 09 20 17 01 20 00 01 A9 0E 85
0179  07 A8 56 85 79 A9 82 85 7D 20 10
0184  01 20 17 01 20 00 01 E6 79 E6 79
018F  E6 7A A5 00 10 E9 01 85 00 A5 FB
019A  6A 01 85 FB C5 FC D0 EC A5 FC C5
01A5  FD D0 E6 20 FF 17 4C 4F 1C C6 07
01B0  D3 CC 20 FF 17 4C 52 01

1760  A9 20 A5 20 95 4F CA 00 F9 60 A9
176B  FF 85 02 17 A9 01 80 03 17 A9 80
1796  85 03 17 A9 01 80 02 17 CE 02 17
17A1  EE 02 17 60 02 17 29 02 F0 F9 20
17AC  88 17 60 A2 00 B5 50 8D 00 17 A9
17B7  01 80 02 17 CE 02 17 EF A2 17 E8
17C2  E0 27 D0 EB A9 0D 8D 00 17 A9 01
17CD  8D 02 17 CE 02 17 EF 02 17 A0 02
17D8  17 29 02 F0 F9 20 88 17 60
```

REVISION TO BATTLESHIP GAME

by Jody Nelis K3JZD, 132 Autumn Drive, Trafford, Pa. 15085

I had trouble getting Ron Kushnier's Battleship program to run reliably in my KIM (from U.N. #6, page 8). Half of the time it ran fine but the rest of the time, after firing 20 shots without a hit, the program would seemingly stop without displaying the co-ordinates of the target ship as it should.

I found the problem to be with the ship positioning random number generator. If a number exceeding $99 was generated, the ship was placed outside of the playing field at a location impossible to hit and impossible for the end of game search routine to locate and display.

Included is a hex listing of my revised battleship program which corrects this problem with a random number limiting test. I also revised the method of positioning the ship to distribute it more equally amoungst the four possible orientations. Also, I made a change to let the program score the number of shots that were used when a kill is made - it displays 'dEAd xx' with the xx being the shots used. All else remains the same as Ron's original program.

Anyone desiring a complete assembly listing of the program can have a copy by sending me a business size SASE with 13¢ postage afixed. Put 24¢ postage on it and I'll include a sheet I made up giving the game instructions and a playing grid to score the shots on - I found this very handy when sitting a new player down in front of the KIM.

REVISED BATTLESHIP PROGRAM - HEX DUMP

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
0200	A9	02	85	00	A9	00	85	E8	A2	99	A9	02	95	00	CA	D0
0210	F9	A9	11	85	E7	85	E3	A2	07	18	A0	07	A9	00	91	E7
0220	88	10	FB	F8	A5	E7	69	10	85	E7	CA	10	EC	F8	38	A5
0230	EA	65	ED	65	EE	85	E9	A2	04	B5	E9	95	EA	CA	10	F9
0240	38	E9	99	B0	E8	A5	EB	29	06	C9	00	F0	41	C9	02	F0
0250	36	C9	04	F0	19	18	A0	02	A6	E9	B5	00	C9	02	F0	B1
0260	A9	01	95	00	8A	69	09	AA	88	10	EF	4C	95	02	A0	02
0270	A6	E9	B5	00	C9	02	F0	99	A9	01	95	00	8A	38	E5	E3
0280	AA	88	10	EE	4C	95	02	A9	10	85	E3	4C	6E	02	A9	01
0290	85	E3	4C	6E	02	A9	20	85	FA	A9	00	85	F9	85	E4	85
02A0	FB	85	B6	D8	20	1F	1F	20	6A	1F	C9	0F	F0	37	C9	09
02B0	10	F1	C9	00	F0	ED	85	E5	A5	B6	C9	01	F0	16	B6	B6
02C0	06	E5	06	E5	06	E5	06	E5	A5	E5	85	FB	20	FE	1E	D0
02D0	FB	4C	A3	02	18	A5	E5	65	FB	85	FB	C6	B6	20	FB	1E
02E0	D0	FB	4C	A3	02	A5	FB	C5	E4	F0	07	AA	B5	00	C9	01
02F0	F0	17	F8	A5	FA	38	E9	01	F0	36	85	FA	D8	A5	FB	85
0300	E4	20	FE	1E	D0	FB	4C	A3	02	B6	F9	A5	F9	C9	03	F0
0310	08	20	FE	1E	D0	FB	4C	F2	02	F8	A9	21	38	E5	FA	85
0320	F9	D8	A9	DE	85	FB	A9	AD	85	FA	20	1F	1F	4C	2A	03
0330	A0	02	A2	99	B5	00	C9	01	F0	06	CA	D0	F7	4C	48	03
0340	8A	99	F9	00	88	4C	3A	03	20	1F	1F	4C	48	03		