

USER NOTES

JULY 1976

COMPLIMENTARY ISSUE

PAGE 1

MICROPROCESSOR SOFTWARE SEMINAR SCHEDULED FOR AUGUST 18-20

A three-day intensive course in software development, featuring the KIM-1, will be held at Turf Inn on Wolf Road in Colonie, New York. The course is being offered by three educators from Rensselaer Polytechnic Institute, Troy, New York.

"Hands-on" experience will be stressed as each student will receive his or her own KIM-1 and power supply. The cost of the seminar will be \$495.00 complete with KIM-1, power supply, course notes, I/O interface, etc. or \$275.00 if you already have a KIM-1.

Interested parties should contact J. C. Williams, R. K. MacCrone, or D. S. Yancy as soon as possible for registration or additional information about the course. They can be reached at (518) 270-6495.

They must have firm commitments by the end of July in order to insure that hardware will be ready by course time.

KIM-2, -3, AND -4 ARE ON THE WAY!!!

MOS TECHNOLOGY is now making more memory available for the KIM-1. Starting August 16, 1976, MOS will be shipping two new memory expansion boards--- the KIM-2 (4K static RAM) and the KIM-3 (8K static RAM). Both boards will be assembled, tested, guaranteed for 90 days, and full burned-in with high-speed static RAM. All buffering and control logic will be included as well as on board regulators. A single KIM-2 or KIM-3 can be wired directly to the KIM-1, but, if you need even more memory, you'll have to wait for the KIM-4 Motherboard, which they say will be available shortly. The price? \$179.00 + \$3.00 (shipping) for the KIM-2, and \$298.00 + \$3.00 (shipping) for the KIM-3.

If you didn't attend the recent Midwest Regional Computer Convention held in Cleveland, then you missed out on a great weekend. A lot of the hobby suppliers were there: Mostek, Motorola, Cromemco, HAL, Communications, National Multiplex, Digital Group, Texas Instruments, DEC, Compumart and others. Rick Simpson, MOS Technology's KIM-1 Product Manager, gave an interesting talk on the KIM-1 (which did not sound like a sales pitch). Dr. Suding compared the Z-80 CPU to other popular chips. Hal Chamberlain discussed an inexpensive floppy disc interface. There were seminars on every major chip manufacturer's products and even one on computer controlled radio networks.

In fact, it was at the convention that a group of KIM-1 users had the opportunity to meet with Rick Simpson of MOS. We all agreed that there was a need for KIM-1 owners everywhere to be able to exchange ideas much like other user groups. It was thought that a newsletter could fulfill that need - so I decided to take part in it as editor and Rick Simpson volunteered technical support from MOS as well as printing and distributing this first issue of KIM-1 User Notes to all KIM-1 purchasers.

Now, we need YOUR support!!!

The newsletter needs articles of all kinds - programming tricks, hardware hints, interface ideas, etc. etc. How did you out-board your keyboard and display? How did you gain access to the display to make it do what you wanted it to do? How did you package your KIM? Does anyone have assemblers or high level languages running on the 6502? How about a text editor?

Let us know what you are doing with your KIM-1.

All article contributions and software listings should be typewritten and include permission for publication. I will not knowingly publish proprietary information, and wish to be informed should this happen.

KIM-1 USER NOTES will be published every 5-8 weeks. Cost for subscriptions will be \$5.00 for the next six issues (including first class postage). This rate for U.S. subscribers only.

Make checks and money orders payable to:
KIM-1 USER NOTES c/o ERIC C. REHNKE
no cash please.

All article contributions should be sent to:

KIM-1 USER NOTES
c/o Eric C. Rehnke Apt. #207
7656 Broadview Rd.
Parma, Ohio 44134

USER NOTES PREVIEW - Calculator interfaces (pros - cons & ideas)
16 bit to BCD conversion program, formatted memory dump routines,
and some music programs.....

AUDIO CASSETTE PROBLEMS?

It has come to my attention that some of us have not received the KIM-1 APPLICATION NOTE #1. In essence, the note stated that since the unit could either be in the decimal or binary modes after power-up, the binary mode would have to be specified by loading \$00 (HEX 00) into location 00F1 prior to all operations with the unit. This sequence resets the decimal flag to assure that the machine is in the binary mode, which is a necessity for proper operation of ALL audio cassette routines as well as the sample programs in the KIM manual. OK?

ONE WAY TO LEARN MACHINE LANGUAGE PROGRAMMING

Learning how to communicate directly with a machine, in its own language, may seem like a formidable task at first. And, indeed, it is --- but, only at first. The more you work at it, the easier it gets --- and the more fun it becomes.

The first step on the road to mastering this esoteric lore is to read and re-read the MOS PROGRAMMING MANUAL. Complete understanding of the 6502 instruction set and chip architecture is a vital necessity. You must know exactly what the machine can and cannot do. Lucky for us that MOS has provided such an informative set of manuals.

The KIM-1 ROM listings will prove indispensable when developing your own programs because there are many valuable sub-routines which are user accessible. Here are some examples.

| <u>CALL</u> | <u>ADDRESS</u> | <u>ACTION</u> |
|-------------|----------------|--------------------------------------|
| JSR AK | 1EFE | Check for key depressed |
| JSR GETKEY | 1F6A | Get key from keyboard |
| JSR SCANS | 1F1F | Display F9,FA, and FB |
| JSR GETCH | 1E5A | Put character from TTY in A |
| JSR PRYBYT | 1E3B | Prints A as 2 Hex Char. |
| JSR PRYPT | 1E1E | Prints contents of FB and FA on TTY |
| JSR OUTCH | 1EAO | Print ASCII char in A on TTY |
| JSR OUTSP | 1E9E | Print a space |
| JSR CRLF | 1E2F | Carriage return and line feed on TTY |

Con't.

I have not found many books that teach machine language programming. One that I know of is called "MACHINE LANGUAGE PROGRAMMING FOR THE 8008 and similar microcomputers", by SCELBI COMPUTER CONSULTING INC. Although this book is slanted toward the 8008 chip architecture, the basic programming information presented is applicable to any machine. This book would seem to be the next logical choice for study material after a thorough digestion of the MOS PROGRAMMING MANUAL.

There are a lot of very interesting concepts presented in the SCELBI book which will prove important to novice programmers. Such things as look-up tables, control tables, counters and pointers are explained in no uncertain terms. One of the bigger sections in the book fully explain mathematical routines including floating-point operations in addition, subtraction, multiplication, and even division! There is sufficient information presented about the 8008 instruction set to enable one to convert all the routines over to the 6502 machine.

BYTE Magazine (March, 1976) had a good article on hand assembly of programs which is recommended reading. Carl Helmers went through the assembly of a subroutine for his 6800 machine which should give you some ideas on organizing your thoughts and getting your programs set up correctly.

I went ahead and made up a number of program development work sheets to assist me in correctly formatting my efforts and they proved to be quite beneficial.

Here is what they look like-----

| PAGE | LOC. | MACHINE CODE | LABELS | MNEMONICS | COMMENTS |
|------|------|--------------|--------|-----------|----------|
| | | | | | |
| | | | | | |
| | | | | | |

Set up a folder for your work and write down everything!!! Later on, when you try to remember how you accomplished a problem, you won't have to re-invent the wheel. And PLEASE keep the rest of us up to date on your efforts!!!!!!!!!!!!!!!!!!!!

There is something very intriguing and challenging about talking with the machine directly.

ERIC REHNKE

UTILITY PROGRAMS ARE NECESSARY FOR SOFTWARE DEVELOPMENT

Software development programs, or utility routines, as they are more frequently called, will aid you in designing software for your machine.

Here are some examples of useful utility routines:

MEMORY DUMP PROGRAM - this routine dumps memory out onto a printer or display device in one of several possible formats. One method would first print the starting address then maybe 8 to 16 spaced bytes of memory (depending on the users terminal column width) - a carriage return & line feed - followed by the next sequential line address and 8 or 16 more bytes of memory until the program has output all memory in the dump range as specified by a starting and ending address entered prior to the dump.

Here is an example:

```
Starting Address 0100
Ending Address   0114

0100 20 2F 1E 20 5A 1E 85
0108 00 20 9E 1E A9 08 85
0110 A9 30 06 00 90
```

Another type of memory dump could look like this:

```
Starting Address 0100
Ending Address   0114

0100 20
0101 2F
0102 1E
0103 20
0104 5A

0112 06
0113 00
0114 90
```

The first method is more space conservative as can be seen, but the second method has its advantages.

The KIM-1 has the necessary subroutines to develop a memory dump program in either of the two styles. The TIM ROM listings contain an example of the first style and could be converted to use the KIM subroutines.

Con't.

BREAKPOINT ROUTINES - This type of program would allow you to insert breakpoints (HEX 00) at various places in your program and by loading the appropriate breakpoint routine starting address into the IRQ vector (17FE,17FF) the program would jump to the breakpoint routine, which could then save all the internal registers and type them out on the users terminal. Then by typing the appropriate op code for that specific location, the original program could continue on from that point until it was finished or another breakpoint is encountered.

MEMORY TEST PROGRAM - It is very important to know the condition of every memory cell in your machine before you start filling it up with software. The memory test programs would first clear all specified memory to 0, write \$FF into one location, go through all the rest of specified memory locations making sure that they are all still cleared. Then, the program would clear that location, write FF into the next sequential location and repeat the whole process for each memory location in the specified test range. If it encountered a faulty memory cell, it would output that location to the terminal or write it into a special memory location.

A TIM documentation manual should be acquired to become familiar with some approaches which are used to design utility routines. The TIM manual is available from MOS for \$4.00.

There are other important utility programs, such as text editors, assemblers, etc. which will be discussed in future issues.

E.C.R.

KIM-1 EDGE CONNECTOR AVAILABILITY

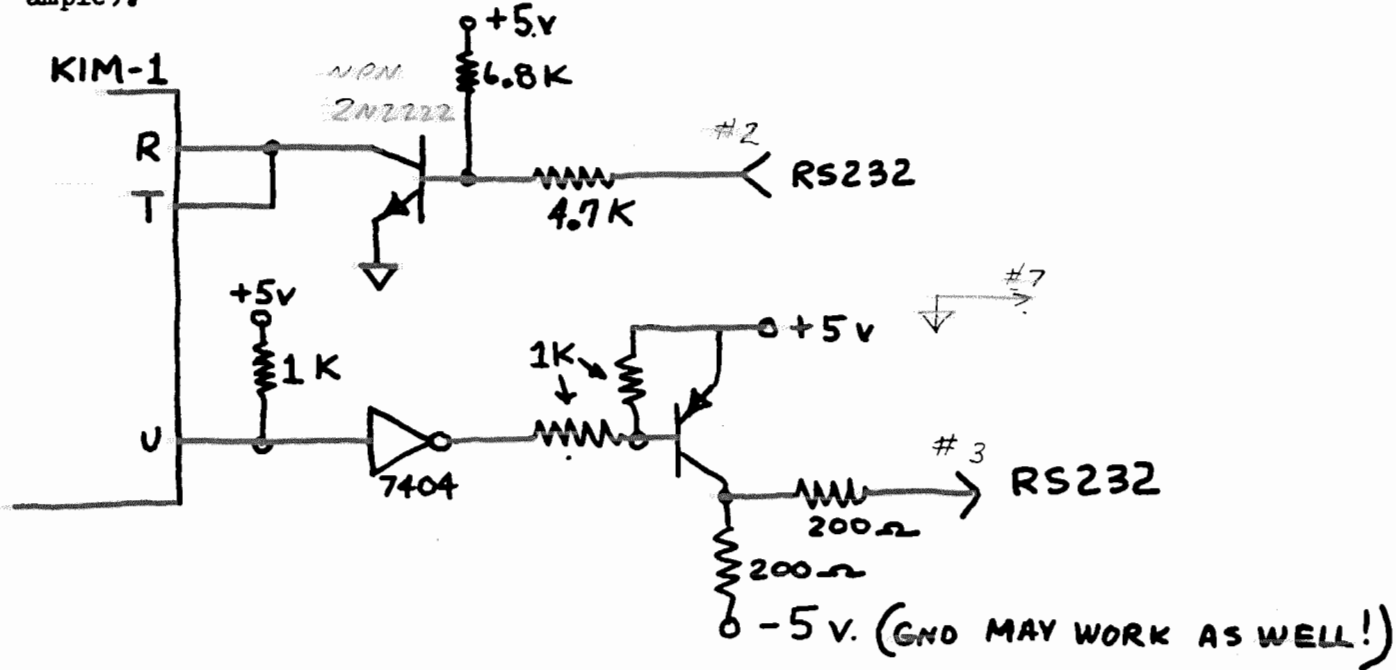
The edge connector that came with your KIM-1 is a widely used item in the industrial and hobbyist field. They are usually available at most ham flea markets as well as industrial and surplus parts stores. If you buy a 22-pin double readout edge connector, you'll have the correct part, or, specify the Vector #R644 which was supplied with the unit.

KIM-1 KEYPADS AVAILABLE FROM MOS

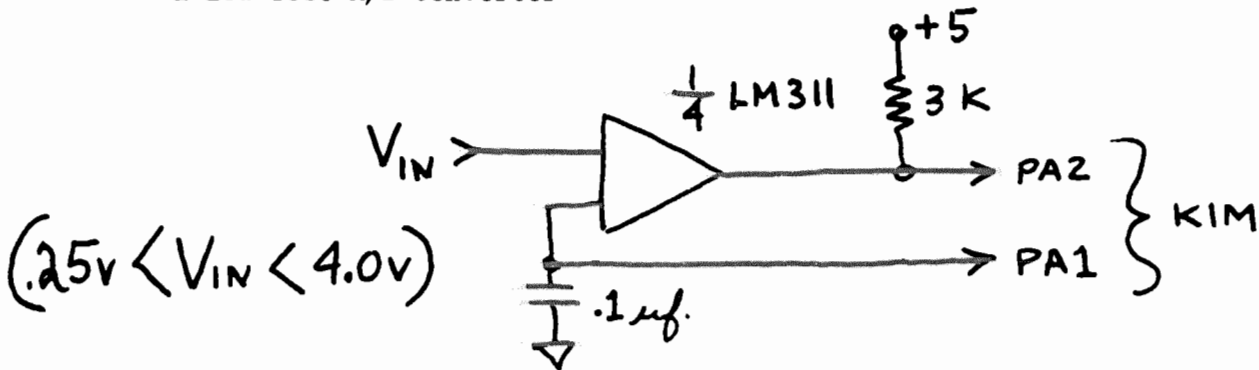
If you are planning an off-board control panel for the KIM-1 and don't wish to remove the existing on-board controls, then you may be interested in knowing that MOS is making the KIM-1 keypad available for \$15.00.

RS-232 INTERFACE & A/D CONVERTER IDEAS

Rick Simpson offers an alternative to his original RS-232 interface design (see BYTE, May 76, page 10). This one should be able to use any NPN transistor that you have in your junk box (2N2222 for example).



Also from Rick----- an idea for you experimenters to work on--- - a low-cost A/D converter-----



Here is an idea of the software necessary to drive it-----

1. Program PA1 as an output, PA2 as an input, and write a "0" to PA1
2. Load the interval timer with "FF"-- and start it counting
3. Write a "1" to PA1
4. Loop until PA2 goes high
5. Read the timer and subtract the reading from "FF"
6. Write a "0" to PA1 and let the capacitor discharge
7. Jump back to step 2
