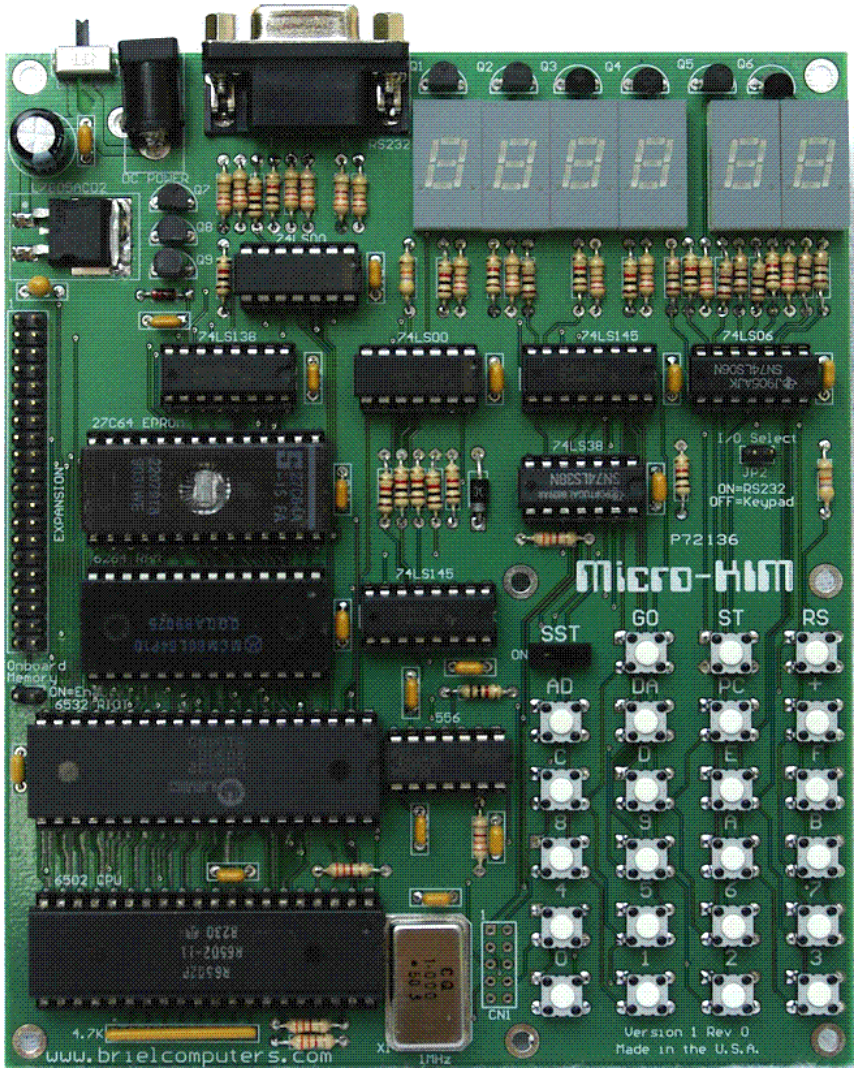
Micro-KIM

Setup and Users Manual

JULY 2007



1st Edition

Briel Computers
5392 Cornell Blvd
North Ridgeville, OH 44039

All materials, schematics, and hardware designs provided without any warranties. Although this material has been carefully examined, Briel Computers takes no responsibility for any errors in printing.

Version 1.1
Published by Briel Computers
5392 Cornell Blvd
North Ridgeville, OH 44039
USA

Forward

How the Micro-KIM came to be

For those of you familiar with Briel Computers, you know about the replica 1 computer. In short, it is a single board computer designed to replicate the Apple 1 computer. For the last 3 years I've been searching for a way to get this computer priced under $100 so more people can enjoy building them. With parts going up in price I've found this to be impossible. So, I went searching for a different platform to replicate.

Why redesign the KIM-1? Why offer it as a kit? Because there are still people out there who like to build and work with kits like those offered in the 1970's and 80's. The KIM-1 has been called the "first single-board computer" by many and was created by Chuck Peddle of MOS as a development board for the 6502 CPU. What better computer to replicate than the first computer designed for the 6502?

The major problem with replicating the KIM-1 was the 6530 RRIOT IC's which have custom ROM's embedded in them. These are no longer available and are difficult to locate. The KIM-1 used two and each one held a section of the monitor program. The solution was to use an external EPROM and a 6532 RIOT which is very close to the 6530 but with some minor differences. This required a tricky addressing scheme I developed to replicate the exact addressing as the original KIM-1 so programs would work the same.

The next decision was to drop the second 6530 (6532) to reduce chip count. This was an easy decision because it can be added later in an expansion. The problem was the 2nd 6530 on the KIM-1 held 64 bytes of RAM. As luck would have it, the 6532 has 128 bytes of onboard RAM vs. the 6530 which only has 64 bytes. I decided to use all 128 bytes available on the 6532 in place of the 128 bytes used in 2 6530's in the KIM-1.

There is no source for replacement keypads used on the KIM-1 so I decided to just go ahead and put the keyboard onboard and use tactile switches with silkscreen labels above them for identifying the buttons. This wasn't perfect but it really helped with the cost and keeping the Micro-KIM to a small footprint.

The last major decision was difficult to make but I felt it was necessary. The expansion slots on the KIM-1 were a disaster. They held unnecessary signals, orientated in a difficult setup and spaced far apart on the board. After debating this one for a long time, I decided to just dump the original interface and create a 40 pin header with all the major signals needed for expansion. Keeping as much of the original signals as possible, yet getting everything needed all in a 40 pin setup was difficult. It is very possible to recreate the original expansion connectors with this setup but there are a few minor signals missing.

Finally I'd like to thank my good friend Rich Dreher for helping me out more than I hoped for on this project. Not only did he redo the board layout for me and help clean up the schematics, he pushed me to add the TTY RS232 interface onboard to add value and much needed I/O onboard. Without Rich's help this project would not be what it is today.

I hope you enjoy your Micro-KIM and that it brings you as much happiness building and using it as I did designing it.

Vince Briel

Setup and Users Manual

Index

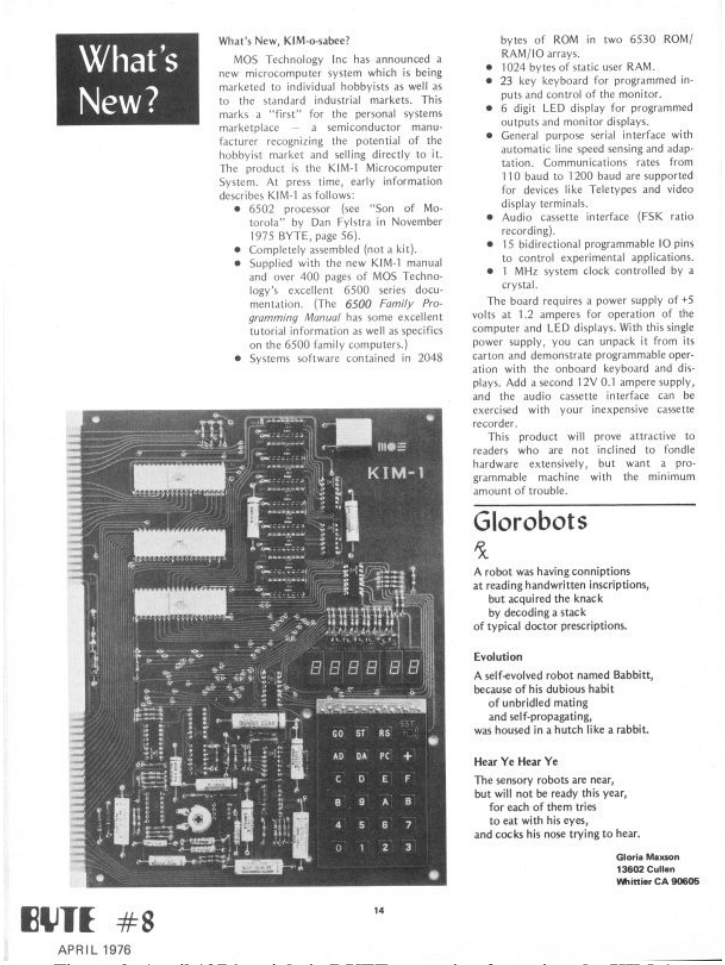
Chapter 1: A brief history of the KIM-1

The KIM-1 which stands for Keyboard Input Monitor was a small single-board computer designed by MOS technologies in 1975 as a development board for their new 6502 processor. The board was originally designed (by Chuck Peddle creator of the 6502) to be used by engineers as a development board. The KIM-1 was so well priced at $245 that it attracted more than just engineers; it attracted a whole new growing group of hobbyists eager to own their own computer.

Figure 1: MOS Technology KIM-1 Advertisement

With an onboard keyboard, 6 segment display, and the ability to connect to a TTY terminal and cassette player for program storage, it was a complete computer package all on a single board. The success of the KIM-1 was a catalyst for user groups, magazines based on the 6502 and the KIM-1, and even software companies.

One of the most well known success stories from the KIM-1 is by Peter Jennings who discovered the KIM-1 in an article in the April 1976 issue of BYTE magazine. Peter Jennings wrote the now infamous Microchess for the 1K KIM-1 and showed people what can be accomplished with a little memory and a lot of desire. You can read more about Peter's story on Microchess at: http://www.benlo.com/index.html



What's New?

What's New, KIM-o-sabee?

MOS Technology Inc has announced a new microcomputer system which is being marketed to individual hobbyists as well as to the standard industrial markets. This marks a "first" for the personal systems marketplace — a semiconductor manufacturer recognizing the potential of the hobbyist market and selling directly to it. The product is the KIM-1 Microcomputer System. At press time, early information describes KIM-1 as follows:

- 6502 processor (see "Son of Motorola" by Dan Fylstra in November 1975 BYTE, page 56).
- Completely assembled (not a kit).
- Supplied with the new KIM-1 manual and over 400 pages of MOS Technology's excellent 6500 series documentation. (The *6500 Family Programming Manual* has some excellent tutorial information as well as specifics on the 6500 family computers.)
- Systems software contained in 2048 bytes of ROM in two 6530 ROM/RAM/IO arrays.
- 1024 bytes of static user RAM.
- 23 key keyboard for programmed inputs and control of the monitor.
- 6 digit LED display for programmed outputs and monitor displays.
- General purpose serial interface with automatic line speed sensing and adaptation. Communications rates from 110 baud to 1200 baud are supported for devices like Teletypes and video display terminals.
- Audio cassette interface (FSK ratio recording).
- 15 bidirectional programmable IO pins to control experimental applications.
- 1 MHz system clock controlled by a crystal.

The board requires a power supply of +5 volts at 1.2 amperes for operation of the computer and LED displays. With this single power supply, you can unpack it from its carton and demonstrate programmable operation with the onboard keyboard and displays. Add a second 12V 0.1 ampere supply, and the audio cassette interface can be exercised with your inexpensive cassette recorder.

This product will prove attractive to readers who are not inclined to fondle hardware extensively, but want a programmable machine with the minimum amount of trouble.

KIM-1

Glorobots

℞

A robot was having conniptions
at reading handwritten inscriptions,
but acquired the knack
by decoding a stack
of typical doctor prescriptions.

Evolution

A self-evolved robot named Babbitt,
because of his dubious habit
of unbridled mating
and self-propagating,
was housed in a hutch like a rabbit.

Hear Ye Hear Ye

The sensory robots are near,
but will not be ready this year,
for each of them tries
to eat with his eyes,
and cocks his nose trying to hear.

Gloria Maxson
13602 Cullen
Whittier CA 90605

14

BYTE #8
APRIL 1976

Figure 2: April 1976 article in BYTE magazine featuring the KIM-1

With growing demand, MOS technology started designing add-on boards for the KIM-1. The KIM-2 was a 4K RAM board, the KIM-3 was an 8K RAM board and the KIM-4 was a motherboard with expansion slots so users could add multiple boards. The development and production of these boards was slow and people grew impatient, resulting in aftermarket boards being developed.

After Commodore had purchased MOS technology, they continued to produce the KIM-1. The onboard logo changed slightly to C=MOS but the layout was kept intact. There were a few revisions but I'm not sure what the revisions changed. Meanwhile the creator of the 6502 and the KIM-1, Chuck Peddle was working on an improved version with QWERTY keyboard and a video display. The result of that work turned out the PET computer in 1977.

The KIM-1 can be credited for being the first single board computer and the first computer for the 6502 CPU; what the KIM-1 did was to start the first generation of single board computer hobbyists.

Chapter 2: Introducing the Micro-KIM

The Micro-KIM is a single board computer that is closely based on the KIM-1 computer produced by MOS technology and Commodore in the mid 1970's. The Micro-KIM retains much of the original design of the KIM-1 with modifications made to allow extinct components to be replaced with more readily available parts. The addressing scheme was left intact so all the original programs from the KIM-1 should work on the Micro-KIM. Many of the programs have been tried but there are still many that haven't. The Micro-KIM consists of 5K RAM, 2K EPROM, a 6532 RIOT for I/O, a 23 key keyboard and a 6 digit display. The added features include a simple DC in feature for using a single source power supply such as a wall wart; a RS232 adapter for use with the monitor terminal program, and a 40 pin expansion header for future expansion.

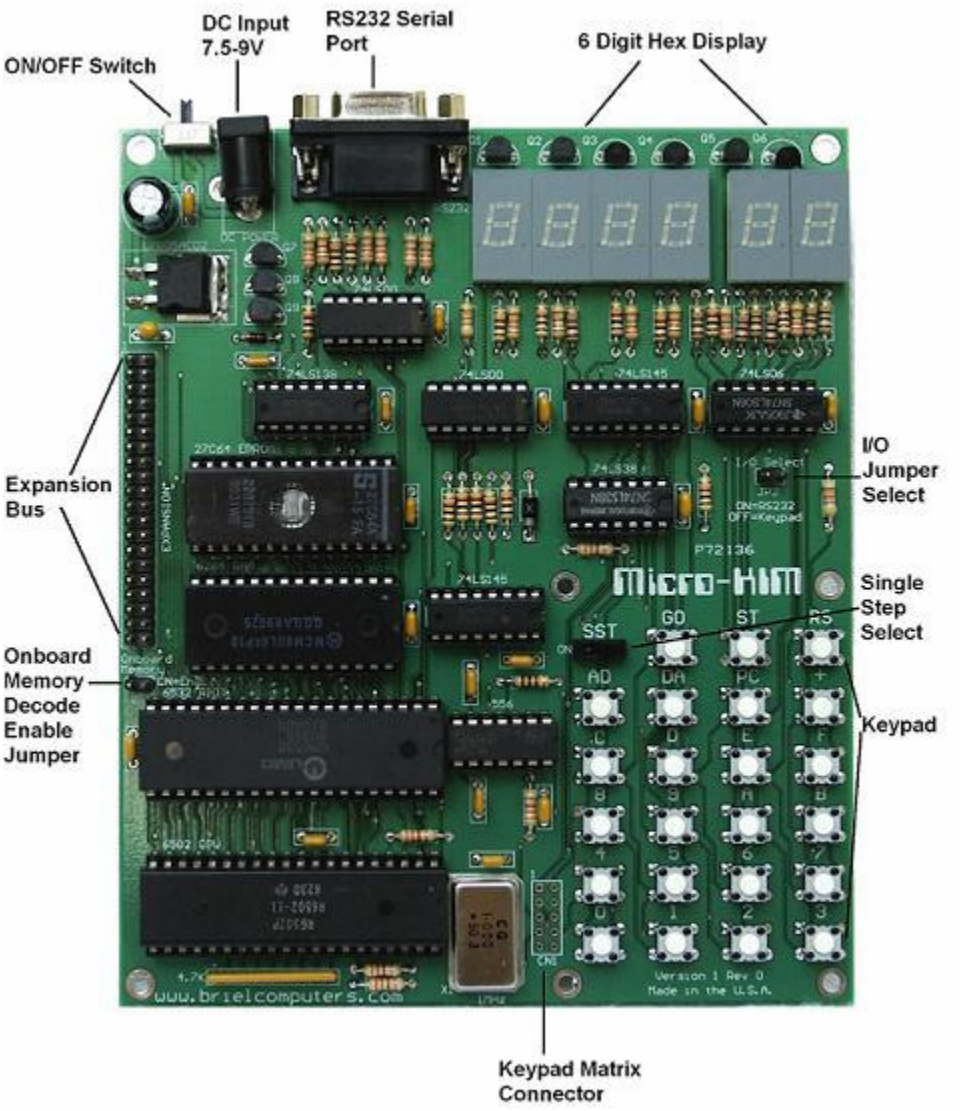


Figure 3: Micro-KIM computer kit

As a result of keeping much of the original design and the fact that there are several 7 segment displays, the Micro-KIM uses a lot of resistors. While resistors don't add much to the overall cost of the board, the board size is affected by this. Originally, the goal was to get the Micro-KIM as small as 4" X 4" but because it is a kit with thru-hole components it was just impossible to get the board down that small.

The onboard RS232 interface is the same circuit used to convert the TTY circuit back in 1976 into a more common RS232 level format. The original circuit was to use -5V which created a problem since the Micro-KIM runs off a single 5V source. The circuit allowed for the possibility to use ground instead and with several tests, the circuit worked with ground. This was huge because if it didn't work, the TTY/RS232 interface would have required extra circuitry to generate the -5V required.

The Micro-KIM has an 8K RAM chip onboard but only 5K is used. The reason is simple. The decoding scheme of the original KIM-1 decoded eight 1K blocks, of which only 1K was used onboard the KIM-1 for RAM and 3K for ROM and I/O. This leaves room for 4K of additional RAM without any further decoding. I simply used an 8K RAM IC to replace the five 1K blocks available for RAM. This means that 3K of RAM in the onboard RAM IC is wasted but it is necessary to keep compatibility with the original KIM-1.

The memory decoding scheme of the Micro-KIM is an exact match to the KIM-1, therefore, the system has a repeating stage of memory every 8K. In other words, the 6502 doesn't know the difference from $0000 and $2000. Anything you store in $0000 will show up in $2000 and visa versa. This can be stopped with expansion by re-decoding memory into 8K blocks and only activating the first 8K when it is directly addressed. For now, just know that if you see an address such as C12F on the display, that isn't the actual address being accessed.

The display is controlled by a 6532 RIOT I/O chip and the monitor keeps the current address and data displayed. When you first power up your Micro-KIM you probably won't see anything on the display. This is normal since the microprocessor needs a reset to get running. There is no auto-reset circuit on the KIM-1 therefore there isn't one on the Micro-KIM either. Pressing RS (RESET) on the keypad performs a system RESET on the Micro-KIM and starts the system up.

Chapter 3: Unpacking and setting up

Before you begin soldering your Micro-KIM kit together you need to inventory all of the parts in the kit. The following is a list of all the components in the kit and the quantity of each component. If your kit is short any parts, email me at vbriel@yahoo.com to arrange to get the missing part. Be sure to allow plenty of work space for assembly.

NOTE: If you have an assembled board, you only need to verify the jumper is on JP1 and off of JP2 to insure normal operation. You can skip this section and proceed to the next chapter. There is no CN1 connector.

The chips are in static protective sleeves and you may be able to inventory them just by looking through the plastic.

The sockets are also in a sleeve but mostly for ease of shipping. Use the figure 3 in the previous chapter to help identify the parts.

The 1 MHz oscillator is identified by its silver color and it looks like a 4 pin chip. Pin 1 is identified as the pointed corner.

The 7805 voltage regulator can be identified from figure 3 as well as the 6 seven segment displays.

There are 9 transistors in total and they can be identified by the number written on the flat spot on them.

There is only one 47K resistor pack that has 9 pins in a single-inline-package or (SIP).

To identify the different resistors, use the part description below to help you identify each resistor. Group all of the resistors into different piles according to their colors. There should be 5 different piles when you have finished.

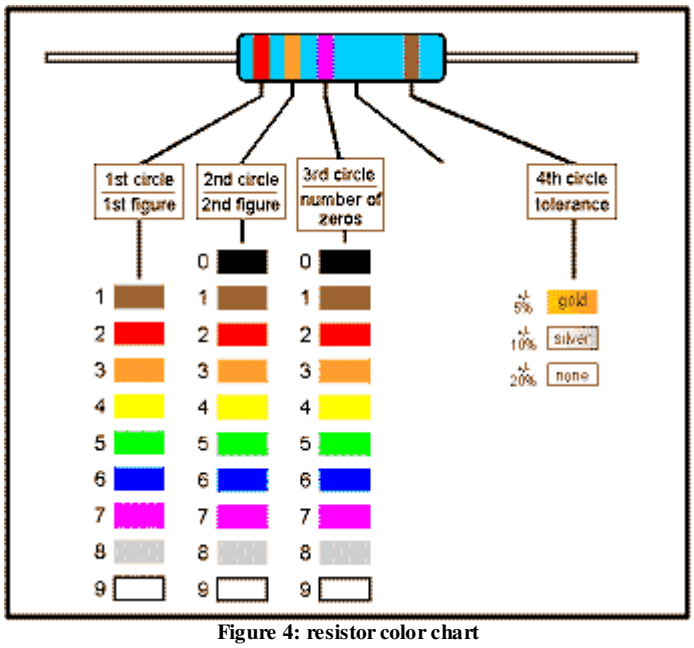


Figure 4: resistor color chart

Look at the resistor and try and find a gold or silver outer band. Hold the resistor so that band is on the right. Look at the first color on the left. Using the chart above, calculate the first value. For example, if the first color band on the left is orange, then the first value is 3. Look at the color band next to the first one, just off to the right and get that value. Let' assume the 2nd band is also orange. Now, the 3rd band is the multiplier. Let's say that color is red. Red is 2 which means we just put 2 zero's at the end of the value. So, it goes 3 (orange), 3 (orange), 00 or 3300. Simplify that to 3.3K by dividing by 1000 and now you have calculated the resistor value. This is a quick explanation on how to calculate the resistor value. There are many web sites out there to help you out but the parts chart gives you color values to help you out.

There are 3 different types of capacitors on the Micro-KIM. The easiest way to figure them out is just put them into 3 different piles.

The 1uF looks different than the .1uF caps and the .22uF caps.
The .22uF caps look close to the .1uF but the value on the capacitor is different.
The .22uF should have the number 224 on it but the number scheme on capacitors can vary.
On the .1uF capacitors, they should say 104. On the 1uF it should just have a 1 on it.

The 2 diodes are easy to identify. The 1N4001 is a larger black diode with a silver strip.
The 1N4148 is slightly smaller and looks like it's made of glass.

It is important to note that diodes are polarized. The white or black stripe tells which way they mount on the PCB. Pay attention when building your kit to ensure that the diodes are in the correct orientation.

The DB9 connector is your serial port connector and looks like a standard 9 pin connector.

The power supply connector is a male jack that fits many standard power supplies.

The 40 pin header has no incorrect orientation and can be mounted either way. However, one side is shorter and that is the side that goes into the PCB. Make sure the longer end sticks up when installing on PCB.

There should also be 2 jumper post and 2 shunts. The JP1 on the Micro-KIM is essential for operation but needs to be removed if an expansion board is used to decode memory differently than what is done on the Micro-KIM.

There are a total of 23 tactile push button switches that are easy to identify.

The power switch will mount next to the power jack on the PCB and has 5 mounting tabs on the bottom.

The SST (single-step) switch is black and sits with the keypad.

There is also a CD containing manuals and text files of some software. This completes your inventory check of the Micro-KIM single board computer.

QUANTITY	PART DESCRIPTION
1	6502 CPU
1	6532 RIOT
1	2764 EPROM
1	6264 SRAM
2	74LS00
1	74LS06
1	74LS38
1	74LS138
2	74LS145
1	NE556
1	1MHz OSCILLATOR
1	7805 VOLTAGE REGULATOR
6	7-SEGMENT DISPLAY
5	14 PIN DIP SOCKET
3	16 PIN DIP SOCKET
2	28 PIN DIP SOCKET
2	40 PIN DIP SOCKET
7	2N4403 PNP TRANSISTOR
2	2N4401 NPN TRANSISTOR
1	47K RESISTOR PAK
7	3.3K RESISTOR ORANGE ORANGE RED
2	47K YELLOW VIOLET ORANGE
10	220 OHM RED RED BROWN
16	1K BROWN BLACK RED
7	100 OHM BROWN BLACK BROWN
14	.1uF CAPACITORS
3	.22uF CAPACTORS
1	100uF CAP
1	1uF CAP
1	D1 1N4001 DIODE
1	D2 1N4148 DIODE
1	DB9 RS232 CONNECTOR
1	40 PIN HEADER
1	DC POWER CONNECTOR
1	SST SWITCH
23	TACTILE SWITCHES
1	POWER SWITCH
2	2 PIN JUMPER
2	JUMPER SHUNTS
1	PRINTED CIRCUIT BOARD
1	CD ROM

Table 1: Complete Micro-KIM parts Inventory

Chapter 4: Assembling the kit

Before you begin your adventure in building your Micro-KIM, make sure you have plenty of room to work with. You will need the following tools to assemble your kit:

- Soldering iron with solder
- Needle-nose pliers
- Wire cutters
- Multi meter or Volt meter
- 7.5V power supply with 2.1mm connector with positive center connection
- Magnifying glass is optional and used to better inspect solder points

The first components to start with are the resistors. By installing these first, this will insure that the resistors will be flat down on the board. They are not polarized so there is no incorrect way to install them. Here is the list of the resistors and their color codes. See page 9 figure 4 for a more complete explanation of the color code chart:

There are 7 100 ohm resistors labeled brown black brown
There are 10 220 ohm resistors labeled red red brown
There are 16 1K ohm resistors labeled brown black red
There are 7 3.3K ohm resistors labeled orange orange red
There are 2 47K ohm resistors labeled yellow violet orange

Place each resistor through the holes and bend the leads outward just a little to hold in place. After you have the resistors in the board, flip the board over and solder the leads. After you have soldered the leads, you can use the wire cutters to trim as much of the excess lead off as possible without removing any solder.

The next component type to install is the diodes. D1 is a 1N4001 and is a black diode with a white stripe. This is a polarized component. Match the stripe on the diode with the stripe on the Printed circuit board (PCB). If this is installed incorrectly, the serial port will not function.

D2 is a 1N4148 and is a little smaller than D1 and is clear with a small black stripe. Match the black stripe with the stripe on the PCB. Once again, if the diode is installed backwards, the serial port may not function.

The next component to install is the 23 keypad buttons. You can install these one at a time, or all at once with a simple trick. Find a board or a piece of metal that is very thin but strong. Place all of the buttons into the PCB. There is only one way they can be installed so you can not install them incorrectly. Next, place the piece of wood or metal on top of the PCB. Here you can either add a rubber band to hold the two together or just flip the two over holding them together. Set the PCB on your workbench and make sure the switches are pushed all the way into the PCB before soldering them in place.

Install the 4.7K resistor pack that is just below the 6502 CPU. Pin one on the PCB is next to the 4.7K label on the PCB. Pin 1 on the pack has a small dot above it. Do not install this component backward.

Next install the power on/off switch. It is a small slide switch with 5 posts on the bottom and is metal cased.

Install all the sockets, checking that the orientation with the notch in the narrow edge matches the PCB. With tin leads, make sure all pins come through the holes before soldering the socket into place.

Install SST switch, it is black and has 3 pins. It mounts next to the GO button on the keypad. It can mount either direction.

Next, install 7805 voltage regulator. It is a surface mount part but just put piece on board and line up 2 pins so they sit even on the contacts on the PCB then add a little solder between the pad and the post of the 7805. Do this for both posts and then also around the large flat spot on the 7805 top back area.

Next, install the 1MHz oscillator. Locate the 1 corner on the oscillator that is pointed and place the oscillator on the PCB so that it matches the pointed outline on the board. The pointed corner is next to the 6502 socket. Trim the extra lead from the pins after soldering them down.

Install the six 7-segment displays. Depending on the brand of display included in the kit, there may be 1 extra hole for each segment. This is normal. Because of the pin orientation, you can not install this component backwards. It is best to put all displays into the PCB, line them up flat, then solder them all in at the same time to ensure that they fit correctly. Trim the extra leads from the back of the displays.

Next, install the 40 pin expansion header. The shorter side goes into the PCB, and there is no incorrect orientation. The contacts are a little close here so take your time installing.

After you finish the expansion header, solder in the 2 jumper posts and install a jumper on JP1. You may need some tape to hold the jumper posts in place while you solder them.

Next install the DC power connector and fold the tabs over.

The next components to install are the 9 transistors.

- Q1-Q7 are 4403's
- Q8 & Q9 are 4401's.

The transistors are polarized so make sure you install them the same as the footprint on the PCB. There is a straight spot on the PCB and the transistor that you should match up prior to soldering. Trim the extra lead from the transistors when finished soldering.

Next, install the fourteen .1uF capacitors. They are labeled with a "104" on them. They are not polarized so you can put them in either direction. Be sure to trim the extra lead length when finished.

Next, install the three .22uF capacitors. They have the number 224 on them. They are also not polarized so they can be mounted either direction. Again, trim the extra lead length off when finished.

The next capacitor is a 1uF capacitor and just has a 1 on it. The longer lead goes next to the edge of board.

The last capacitor is a 100uF capacitor. It is polarized and there is an arrow on the capacitor pointing to the negative pin. The positive pin has a label on the PCB.

Caution: Make sure this capacitor is installed correctly. Failure to do so could cause the capacitor to explode and may cause injury.

The last piece to solder on is the DB9 connector for the RS232 interface. Install the connector slowly to make sure all the pins are lined up correctly. You should solder the outer posts so the connector holds better to the PCB.

Now is a good time to take a break and come back to check your work. The most common error in soldering boards is missing a contact on a socket. I like to take each socket one at a time and look at each pin with a magnifying lens. Look for missing, cold, or low solder points and correct when found.

Before you insert your chips, you can attach your power supply and power up the Micro-KIM. While it won't function, you can use your volt meter to test voltages. You should have 5V's on pin 14 or 16 of most of the smaller chips and pin 28 on the RAM and EPROM's.

The last stage of the assembly is installing the chips. The board is clearly labeled showing where all the chips go. Duel wipe sockets were used for people who have not built kits before. While machine sockets are much higher quality, they are difficult to insert chips into. For this reason, the duel wipe sockets were chosen.

Be sure to install the chips with pin 1 on the right. See the picture below for an example of where pin 1 is. Notice the notch at the top, the pin on left is pin 1.

Figure 5: IC showing pin 1

You may find that the 74LS145 and the 6532 are very close together, almost touching. This is normal. After all the chips are installed you are ready to test your Micro-KIM. See chapter 6 for operation and chapter 8 for any troubleshooting that may be required.

NOTE: There is no connector for CN1. It is optional to add if wanted for external keypad operation.

Chapter 5 Installing the USB drivers

If you have the optional USB to serial converter you can install the drivers. If you are using Windows XP you can plug in the converted and click NEXT when the new hardware found message appears. Select the directory where the drivers are stored on the CD and click next. When the message appears about the driver testing, click continue anyway. After the driver is installed you can check the system to verify that you have an additional com port on your system.

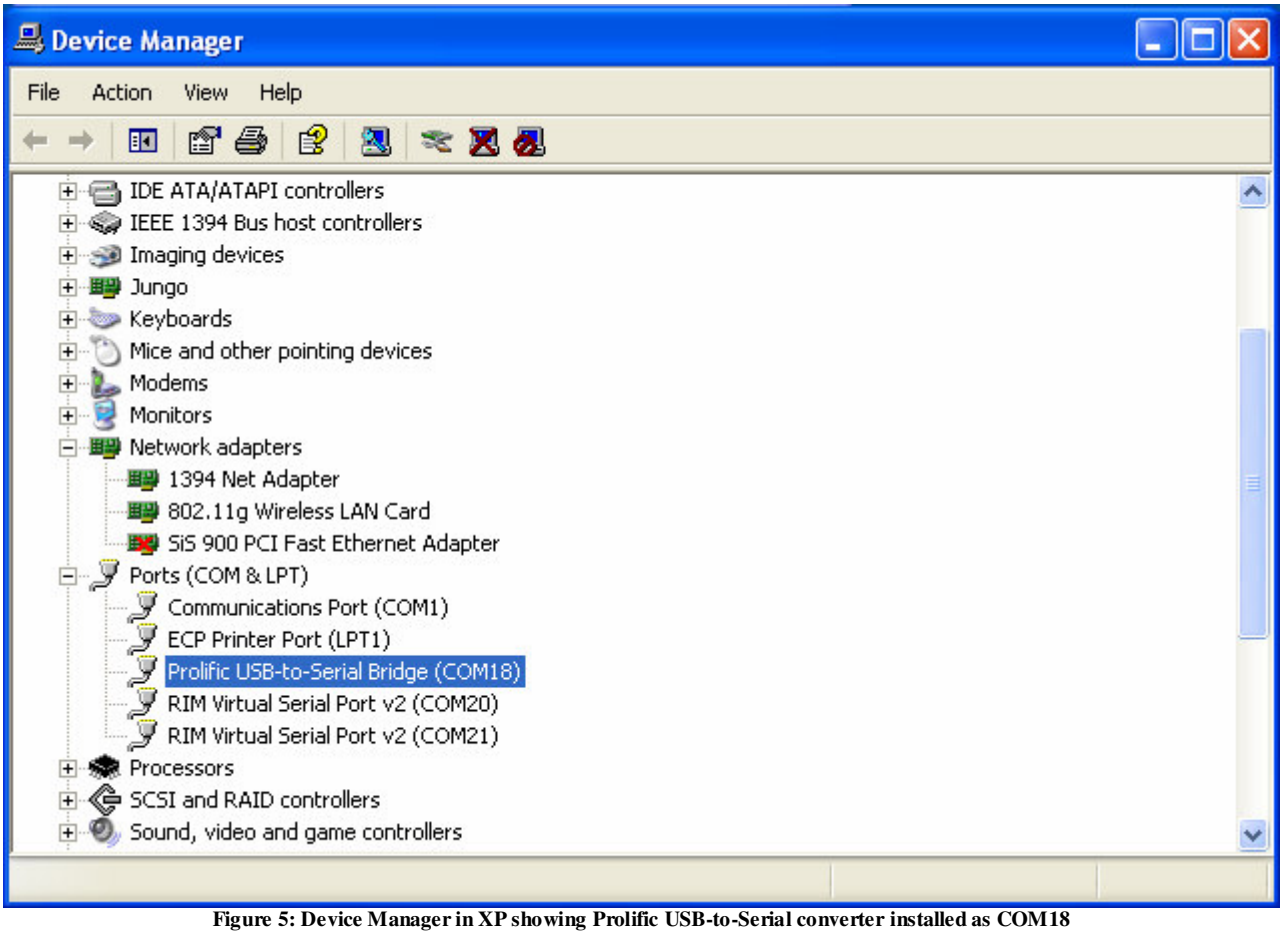


Figure 5: Device Manager in XP showing Prolific USB-to-Serial converter installed as COM18

If you are using a different operating system, follow the USB-to-Serial converter manual for instructions on how to install on your system. Be sure not to attach the serial connector to the Micro-KIM while power is on as damage may occur. With the drivers installed you may now attach the USB converter to an M/F DB9 serial cable and to your Micro-KIM. At this point you are ready to power up your Micro-KIM and follow the instructions for configuring the serial port in chapter 7.

Chapter 6: Using the Micro-KIM

This chapter will deal with using the Micro-KIM with the 6 digit display and the keypad onboard. Check chapter 7 for how to use the Micro-KIM with the serial port and Hyper terminal.

The first thing to do is make sure JP1 jumper is connected so that the onboard memory decoding is enabled. Make sure JP2 is off to use the onboard keypad and 6 digit display. Also be sure the Single-step switch SST is not in the ON position. Power up the Micro-KIM, and then press the **RS** button on the keypad. The display should then come up with an address on the 4 displays on the left and a value to that address in the 2 displays on the right.

At this point, the Micro-KIM is ready to take input. Here's how the keypad and display work:

There are two values on the display, the address and the data. There are two keys on the keypad to indicate which one you want to work with **AD** and **DA**. To get to a memory location, press the **AD** button. Let's say we want to look at $200 in memory. Now, that you have pressed the **AD** button to tell the Micro-KIM you want to look at an address, type in **0200** into the keypad. The value **0200** will show in four digits on the left and the 2 digits on the right show the value of location $200.

To change the value, all you have to do is tell the Micro-KIM that you want to go to data mode by pressing the **DA** button. Now enter in the new value for the location. That's it; you've just changed the location **0200** to your new value. Press the **+** key and the memory increments to the next location, so the display will now show **0201** and the value of $201 will be in the 2 right digits.

At this point, you are still in the data mode so you can enter new data if you wish or move on to the next location. To go back to the address mode, press **AD** button and enter in the value of the address you wish to jump to.

The first thing that you should always do when you first power up your Micro-KIM is set the vector locations in memory. From the keypad type:

AD then type in **17FA** press the **DA** button and type in **00.**
Press the **+** key and the address will change to **17FB**. You are still set to program the data so type in **1C**.
Press the **+** key again and the address will change to **17FC**, press + again until the display shows **17FE** then press **00**. If you went past the address you can press **AD** and type in **17FE** then press **DA** to change the data.
Press the + one last time and the display should show **17FF**. Press **DA** and enter the value **1C**.

This sets the reset and break addresses in the Micro-KIM for proper single step operation.

At this point you are ready to enter a program into memory. What better program as a "HELLO WORLD" program than a simple clock program? It turns the Micro-KIM into a functional clock. It is rather long to type by hand but it is a good example of how to enter a program.

0200 A2
0201 EA
0202 CA
0203 A5
0204 60
0205 85
0206 Fb
0207 A5
0208 61
0209 85
020A FA
020b A5
020C 62
020d 85
020E F9
020F 86
0210 63
0211 84
0212 64
0213 20
0214 1F
0215 1F
0216 A6
0217 63
0218 A4
0219 64
021A E0
021b 00
021C d0
021d E4
021E F8
021F 38
0220 A9
0221 00
0222 65
0223 62
0224 85
0225 62
0226 d8
0227 C9
0228 60
0229 d0
022A d5
022b F8
022C 38
022d A9
022E 00
022F 85
0230 62
0231 65
0232 61
0233 85
0234 61
0235 d8
0236 C9
0237 60
0238 d0
0239 C6

023A F8
023b 38
023C A9
023d 00
023E 85
023F 62
0240 85
0241 61
0242 65
0243 60
0244 85
0245 60
0246 d8
0247 C9
0248 13
0249 d0
024a b5
024b A9
024C 01
024d 85
024E 60
024F C9
0250 01
0251 F0
0252 AD
0253 20
0254 5C
0255 18

You can go back to 0200 and press the + key and double check your data. Once you have the program entered, you need to set the time. Address **0060** holds the hours, **0061**, holds the minutes and **0062** the seconds. Enter the correct time into those addresses then enter **0200** on the address display and press the **GO** key to begin the program. Your Micro-KIM should now be displaying the time.

You can find more programming samples in the First Book of KIM and the KIM-1 Users manual. Since the Micro-KIM is a replica based closely on the KIM-1 some of the known issues with the KIM-1 may also be in the Micro-KIM such as multi-key strokes lighting up the display, or pressing **ST** right after power up may stop the display on 1 character. These are examples of normal behavior of the Micro-KIM.

If you feel your Micro-KIM is not functioning properly, feel free to email Briel Computers at vbriel@yahoo.com and ask any concerns you may have.

Chapter 7: TTY Serial Interface

The Micro-KIM has the same TTY current loop interface as the KIM-1. The difference is that the RS232 conversion circuit was also included so you could attach a terminal or PC with Hyper Terminal to the Micro-KIM for a display interface. It is possible to attach the Micro-KIM to a Mac but since there are many terminal programs out there, we are just going to focus on a standard PC using Windows XP.

The first step is to run Hyper Terminal and set up a terminal port. We will use the slower 1200 baud rate to ensure that the system works correctly. Later you can change the baud rate to a higher level and find the highest value that works with your system.

First, on your PC, click on START, All Programs, Accessories, and Communications and select Hyper Terminal. A new connection window for Hyper Terminal will come up like this:

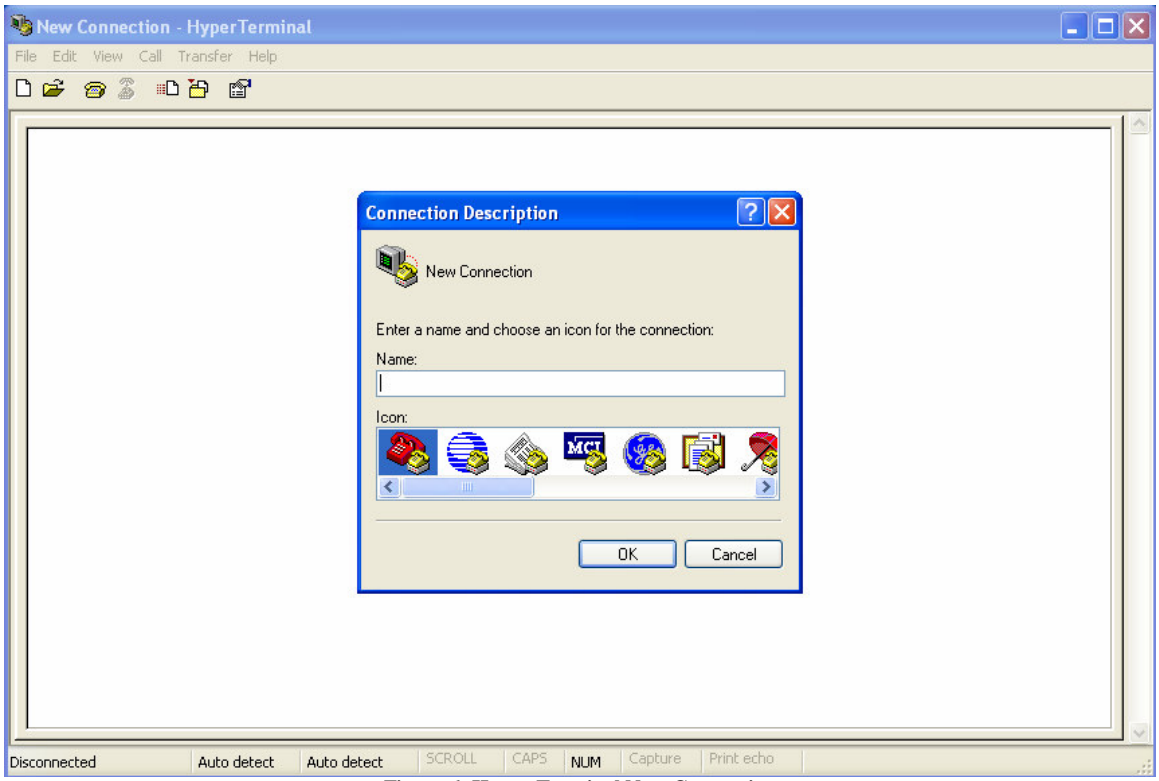


Figure 6: Hyper Terminal New Connection

Now you can type in a name for your connection such as, KIM, or Micro-KIM and click on OK. Next, it may ask for an area code if this is the first time you are using Hyper Terminal. Simply type in your Area code, and click on OK.

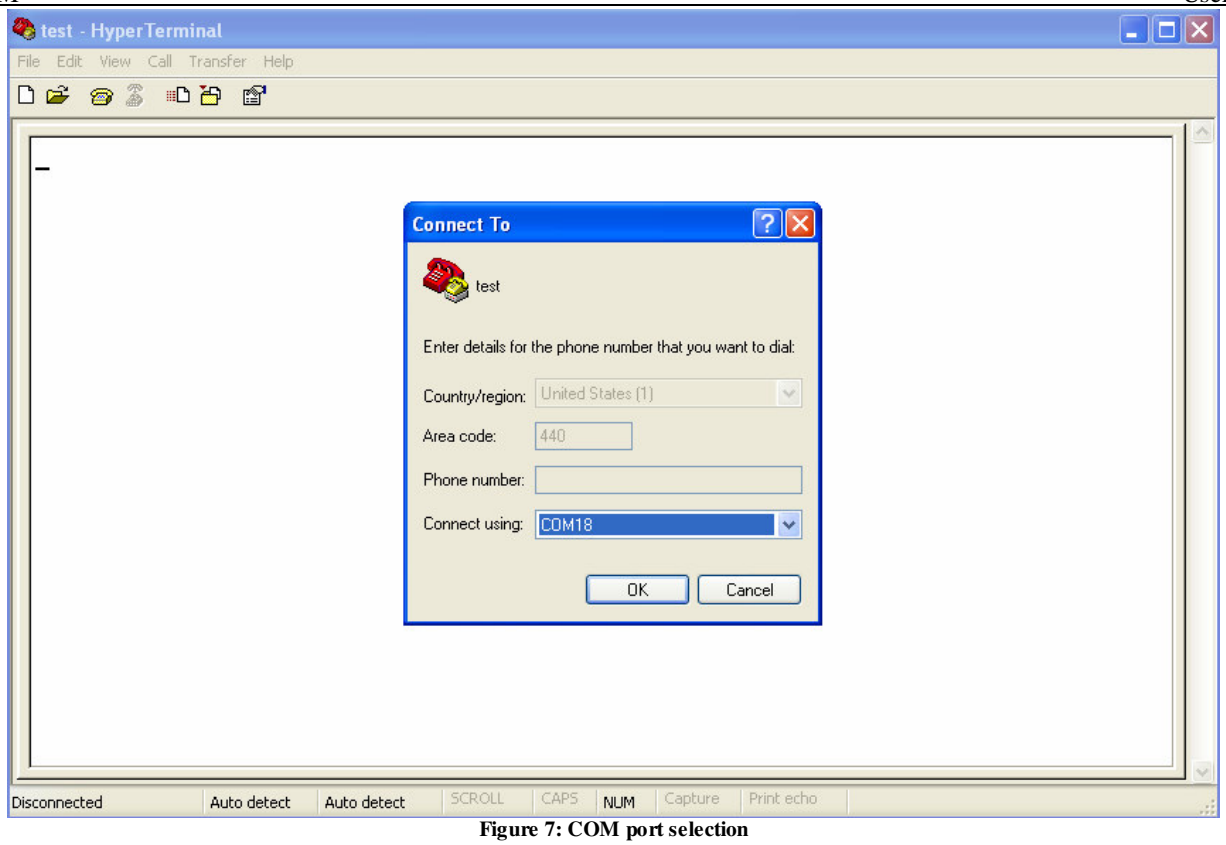


Figure 7: COM port selection

Next, select the COM port that the Micro-KIM is attached to. In this case, COM18 is my USB-to-serial port. Click OK to continue.

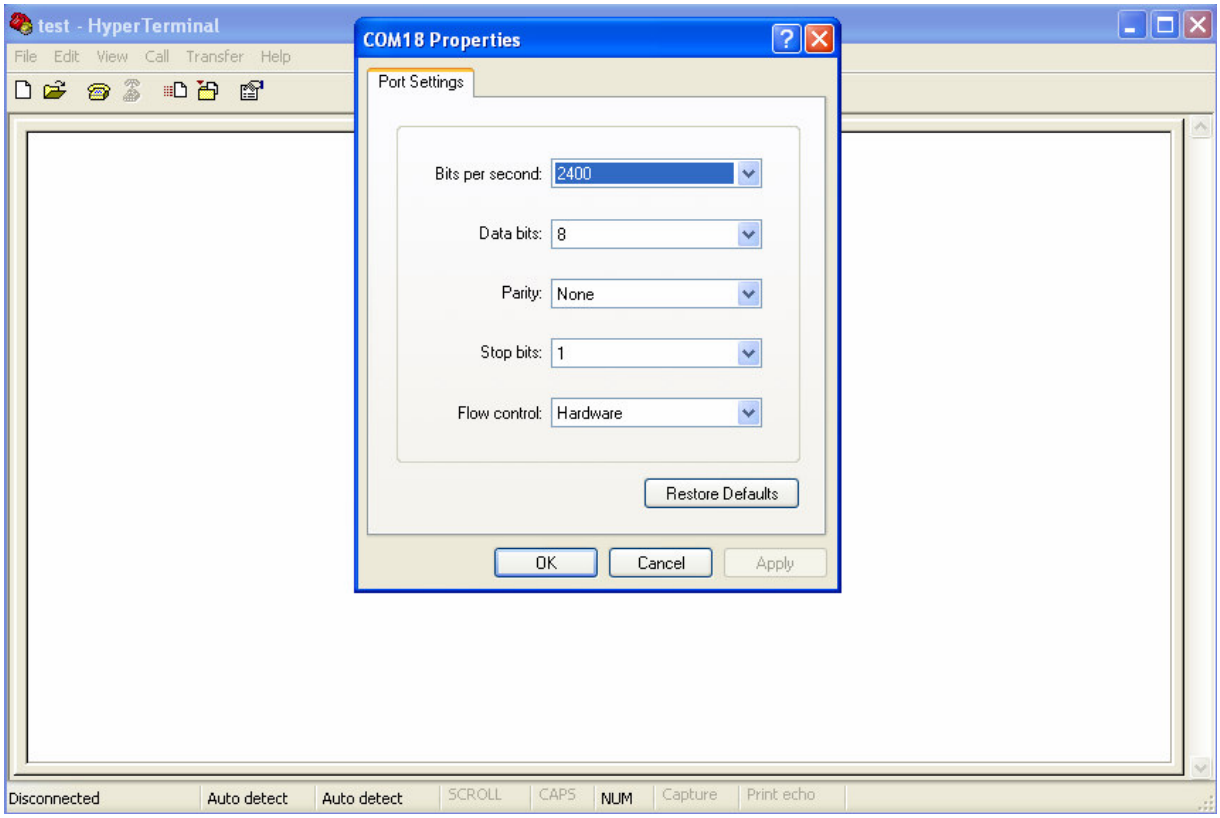


Figure 8: COM port properties

Next step is to set up the COM port with your settings. Figure 8 shows 2400 but it might be better to start with 1200 BAUD to ensure connection with out any problems. You can increase the BAUD rate at a later time. You must also change the Flow control to NONE. If you don't change this, the PC will not communicate with the Micro-KIM. With the changes made, select OK. The correct COM port settings are in figure 9 below.

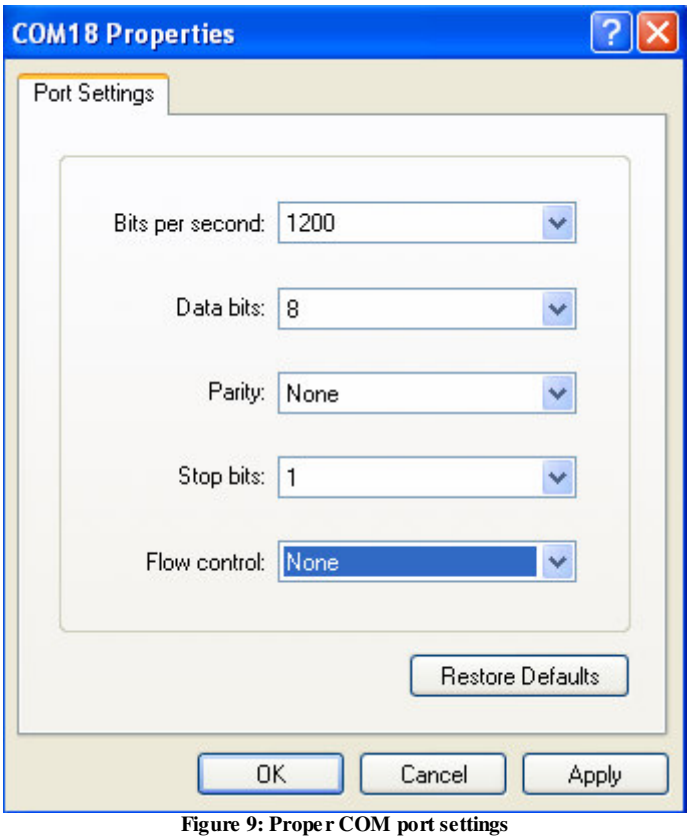


Figure 9: Proper COM port settings

Your connection is now complete. Make sure your cable is connected to your PC and to the Micro-KIM before turning on the Micro-KIM. The cable used should be a Male/Female DB9 RS232 extension cable. Do not use a Null Modem cable as this will not work.

As a last stage, you might want to change the character delay to 5ms and the line delay to 200ms under **FILE**, **PROPERTIES**, **SETTINGS** tab. Click on the **ASCII** setup and you can change the values there. This will help ensure that you don't lose data.

You can now prep your Micro-KIM for serial communication by placing a jumper shorting block on JP2 located above the Micro-KIM logo on the PCB.

Steps to starting the Micro-KIM in TTY mode:

- Place shorting block on JP2
- Connect cable to PC
- Power on Micro-KIM
- Press **RS** on the keypad
- Press Enter on PC keyboard

At this point, the word KIM will display on Hyper Terminal. If it doesn't, check your cable and your Hyper Terminal settings. See figure 10 for a sample of Micro-KIM at power up. The 6 digit display will be blank. This is normal.

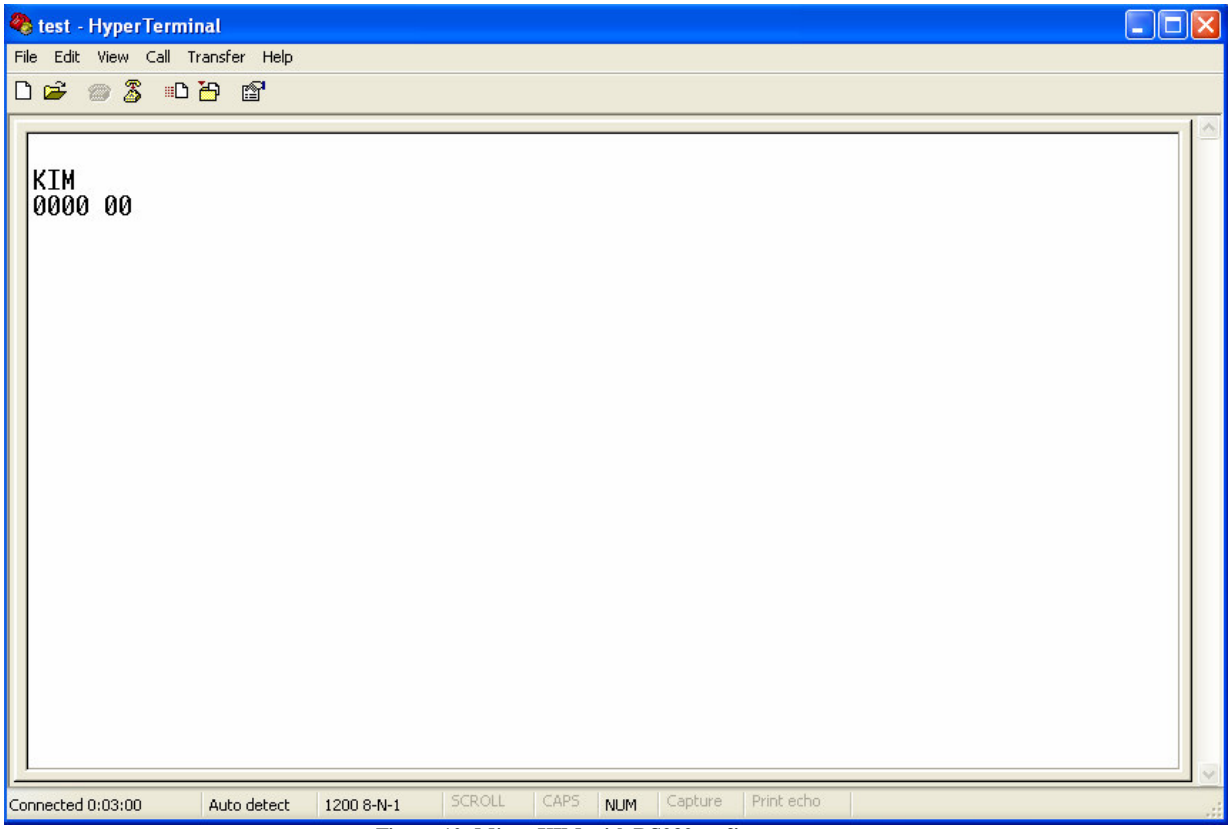


Figure 10: Micro-KIM with RS232 on first power up

Before you begin, make sure **Caps Lock** is on because the monitor does not recognize lower case letters.

There are only a few simple commands in the TTY monitor. To examine an address, type in the address and press the space bar. The address will be displayed along with the value of that address. The cursor will move to the right of the value.

To change the value of the current memory location type in the 2 digit HEX value and type in a period "." to tell the monitor to change the value. The monitor will respond by displaying the next address automatically.

To run a program type in the address and press space, next type **G** and the program will begin from the currently displayed address.

The TTY monitor also has paper tape functions built in to load and store programs from paper tape. Well, Hyper Terminal can act like a paper tape machine and this gives a great storage means for your Micro-KIM programs. This is discussed in detail in chapter 8.

Be sure to read the KIM-1 Users Manual for a more in depth explanation on using the TTY interface. The interface is different for every PC and may require adjusting of the character delay, line delay to get the right setting for your machine.

Chapter 8: Using the Paper Tape Feature

The TTY monitor includes a feature to load and store programs from paper tape. But wait, you don't have a paper tape machine? Well, yes you do, in a way. Hyper Terminal can act as a paper tape machine to load and store programs. Let's take you through the whole process. Read chapter 7 on using the TTY serial interface before proceeding.

Storing a program:

First step is to power up your Micro-KIM with jumper block on JP2 and press **RS** on the keypad and Enter on your PC to bring up the monitor. Next step is to enter your program and make sure it is working correctly. Next, you need to enter the vector variables for the paper-tape so that it knows where your program ends. **17F7** and **17F8** contains the ending address for your program. So let's say, your program starts at 0200 and ends at 0300. Program 17F7 and 17F8 with the following values:

17F7 00.
17F8 03.

17F7 contains the lower byte value of the address and 17F8 contains the upper byte to the address. Next, type in 200 and press the space bar on your PC to bring up the starting address.

In Hyper Terminal you can click on **Transfer** at the top and **Capture Text** in the drop down list as show in figure 11.

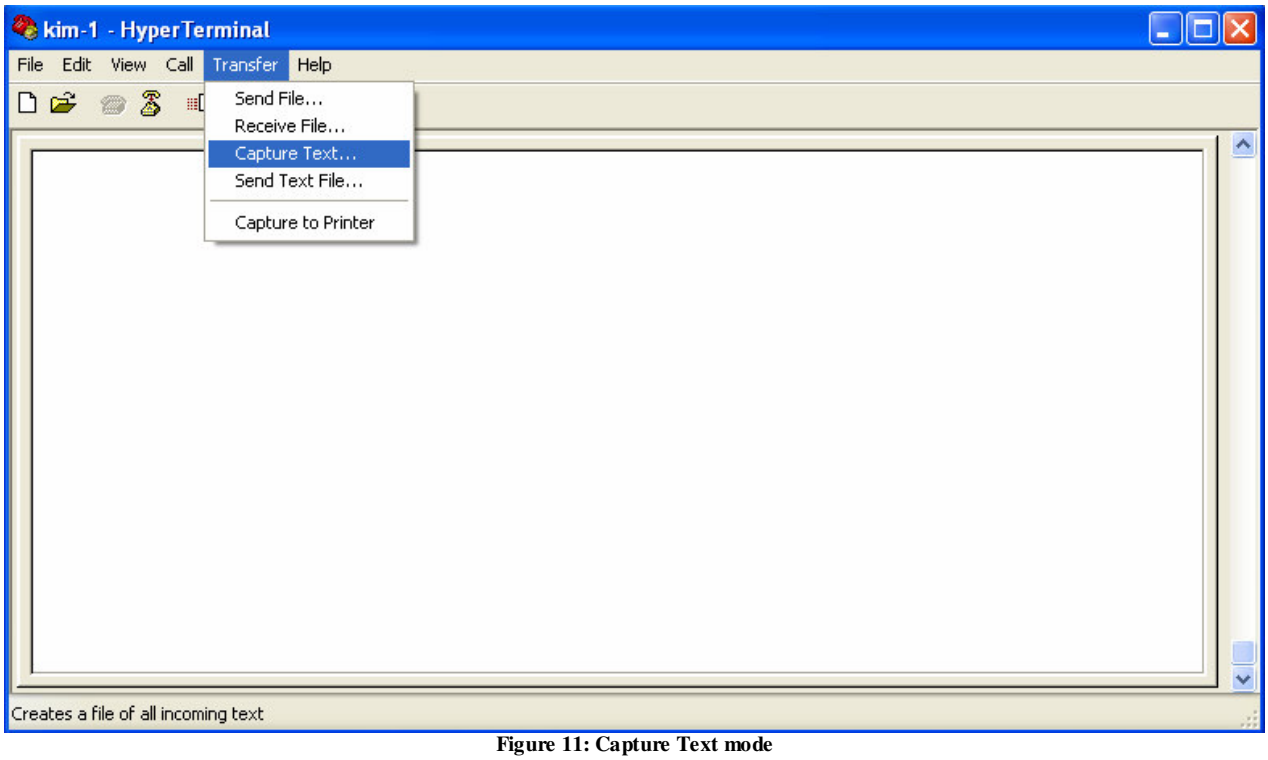


Figure 11: Capture Text mode

Now you can type in the file name to save it to and click on **Start button**. In Hyper Terminal press **Q** and your program will dump onto the display in a compressed format and Hyper Terminal will be capturing the text into your file.

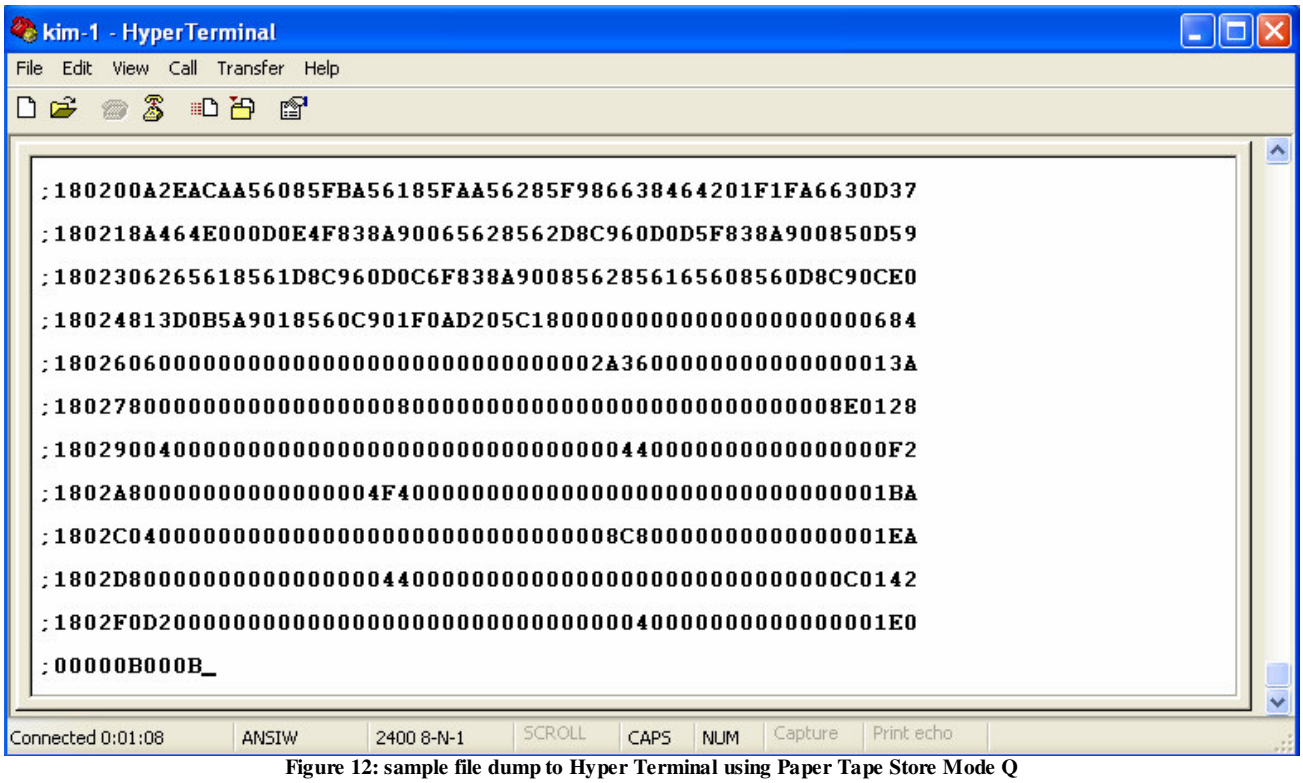


Figure 12: sample file dump to Hyper Terminal using Paper Tape Store Mode Q

After the program finishes dumping; click on **Transfer**, **Capture**, and **Stop** to end the capture process. You now have a text file of your program that you can load into the Micro-KIM at any time.

Loading a program:

Now that you know how to store a program, you will need to know how to load it into memory. First step is to power up your Micro-KIM in the TTY monitor. Press **RS** on your Micro-KIM, followed with Enter on your PC.

Simply press **L** in Hyper Terminal to load a program. Click on **Transfer** and **Send Text File** and select your program to dump into the Micro-KIM as shown in Figure 13.

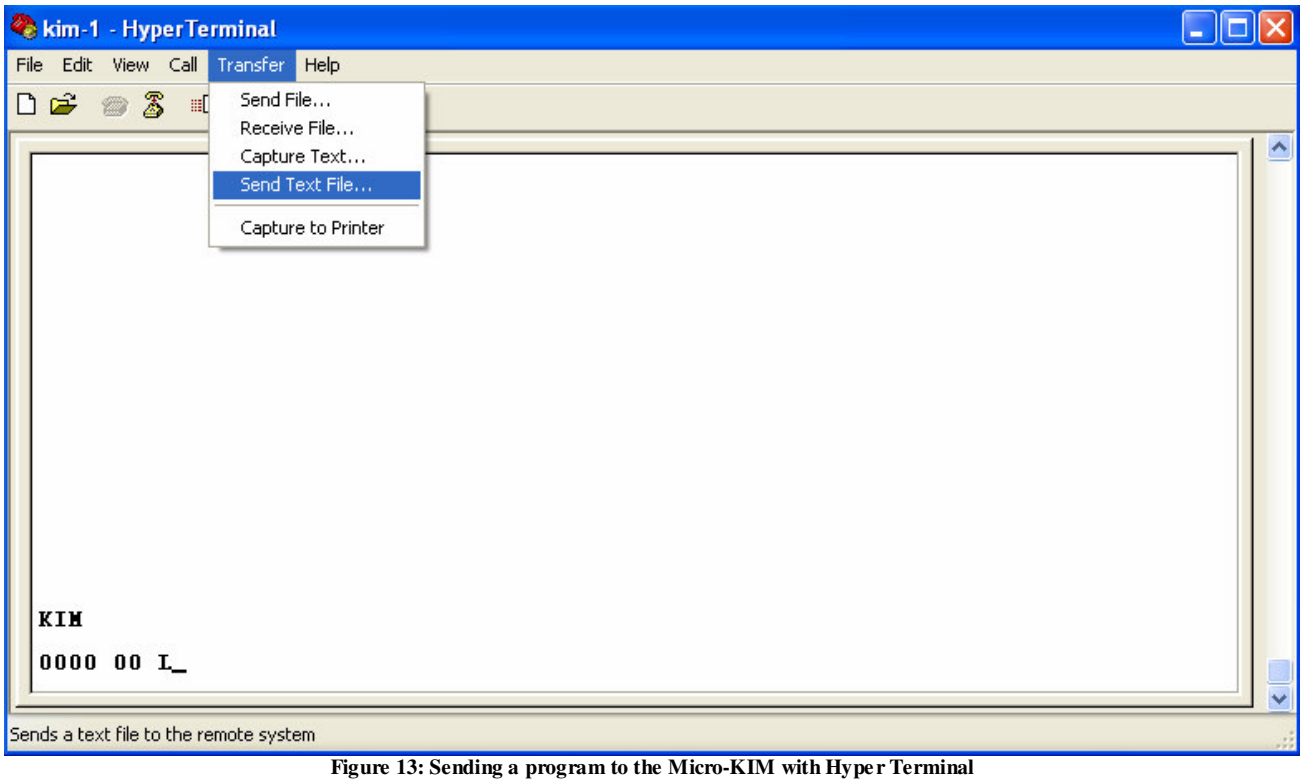


Figure 13: Sending a program to the Micro-KIM with Hyper Terminal

The program dumped looks just like when you stored it in Figure 12. Once your program is finished dumping; you can run it by typing in the address and pressing space bar to get to the starting address. If the program works off of the 6 digit display on the Micro-KIM, simply remove JP2 and you can run your program from the keypad.

Chapter 9: Troubleshooting your Micro-KIM

If you have just built your Micro-KIM or have been using it for some time and are now having problems, there are a few simple things you can check. Here is a small list of possible problems and fixes. Your problem may not be on the list. If you can not get your Micro-KIM working, contact Briel Computers for more help. Our goal is that every Micro-KIM shipped works.

Problem	Possible solution
Dead, nothing on display	Check JP2 and make sure it is **off** and JP1 is **on**. When Micro-KIM is first powered up check that **RS** is pressed to reset circuit. Make sure all IC's are fully seated in the sockets. Verify that power is coming into the Micro-KIM. Check that VCC is at least 4V by checking pin 28 of the EPROM or RAM or pin 14 or 16 on most of the TTL chips. Check 1MHz signal at pin 37 of 6502 to verify that system clock is working.
Keyboard doesn't work	Check JP2 and make sure it is removed. Check 6532 and 74LS145 just below 7-segment displays and make sure it is seated properly.
Only one digit is displayed and it is really bright	This is normal when display is not multiplexing. This can happen if you step through the monitor. Press **RS** to reset system. Display should return to normal. If not, verify all IC's are properly seated in their sockets.
Voltage regulator is very warm	The voltage regulator drops voltage down to TTL (5V) level from your input level. The higher the input level, the hotter the voltage regulator will get. Do not put more than 9V DC into the Micro-KIM or it could cause the Micro-KIM to overheat. Recommended input voltage is 7.5V
Nothing happens in HyperTerminal	Make sure you are using a standard DB9 Male/Female cable. Check your settings in HyperTerminal. Check that JP2 is on to enable RS232 and disable onboard display/keypad. Verify correct COM port is selected in HyperTerminal.
USB to Serial not working	Verify that the drivers are installed properly. Make sure correct COM port is selected in HyperTerminal

If this list of problems and solutions does not help you, feel free to email Briel Computers to help resolve any problems with your Micro-KIM.

Appendix A: Onboard Jumper Settings

There are only 2 jumpers on the Micro-KIM but they play a critical role in the operation of the Micro-KIM. If the jumpers are not selected properly, your Micro-KIM will not function properly, if at all.

JP1 is located directly below the 40 pin expansion header on the left edge of the board. This jumper enables the onboard memory decoding. This feature is installed to give KIM-1 compatibility to enabling memory decoding onboard. Future expansion boards will need to enable and disable the onboard memory as needed. Keep this jumper block on at all times unless an expansion board is installed requiring the removal of this block.

If this jumper is removed without some external decoding in place, the Micro-KIM cannot access the onboard memory and thus it will not function.

JP2 is located above the Micro-KIM logo on the right side of the board. This jumper is the selector between the onboard display/keypad and the RS232 interface when in the monitor mode. It does not mean that the keypad and display will not work when running a program. It only applies to input and output while in the system monitor.

If this jumper is off while the Micro-KIM is in use, all monitor I/O is handled on the keypad and 6 digit display.

If this jumper is on while the Micro-KIM is in use, all monitor I/O is deferred to the RS232 serial port.

Appendix B: Micro-KIM KIM-1 Memory Map

Micro-KIM		KIM-1	
$0000-$03FF	**1024 Bytes of RAM**	**$0000-$03FF**	**1024 Bytes of RAM**
$0400-$07FF	**1024 Bytes of RAM**	**$0400-$07FF**	**Optional Memory Area**
$0800-$0BFF	**1024 Bytes of RAM**	**$0800-$0BFF**	**Optional Memory Area**
$0C00-$0FFF	**1024 Bytes of RAM**	**$0C00-$0CFF**	**Optional Memory Area**
$1000-$13FF	**1024 Bytes of RAM**	**$1000-$13FF**	**Optional Memory Area**
$1400-$16FF	**Optional Memory Area**	**$1400-$16FF**	**Optional Memory Area**
$1700-$173F	**Optional 2nd 6532 I/O, Timer**	**$1700-$173F**	**6530-002 I/O, Timer**
$1740-$177F	**6532 I/O and Timer**	**$1740-$177F**	**6530-003 I/O, Timer**
$1780-$17BF	**64 Bytes RAM from 6532**	**$1780-$17BF**	**64 Bytes from 6530-003**
$17C0-$17FF	**64 Bytes RAM from 6532 ***	**$17C0-$17FF**	**64 Bytes from 6530-002**
$1800-$1BFF	**1024 Bytes of EPROM**	**$1800-$1BFF**	**1024 Bytes of ROM in 6530-003**
$1C00-$1FFF	**1024 Bytes of EPROM**	**$1C00-$1FFF**	**1024 Bytes of ROM in 6530-002**
$2000-$FFFF	**Unused memory**	**$2000-$FFFF**	**Unused memory**

* The 6532 has 128 bytes of RAM vs. only 64 bytes on the 6530. The Micro-KIM utilizes all 128 bytes from the single onboard 6532 so all original memory locations are available.

Appendix C: Expansion Port Description

When the Micro-KIM was under design a lot of thought went into every detail. One of the stumbling blocks was the expansion port. On the KIM-1 there are two 44 pin edge connectors that are spaced far apart from one another. While trying to keep with the original design it was determined that since the 2nd 6530 was not onboard, the application port was not required. Also, during the redesign, many of the signals were dropped simply because they were just inverted signals that already existed. So, it was decided to just put the necessary signals on the expansion connector so that it would still be possible to make an actual KIM-1 connector for compatibility.

Looking at the signals we see that all the CPU signals are there. Address bus, Data bus, R/W, RESET, etc.

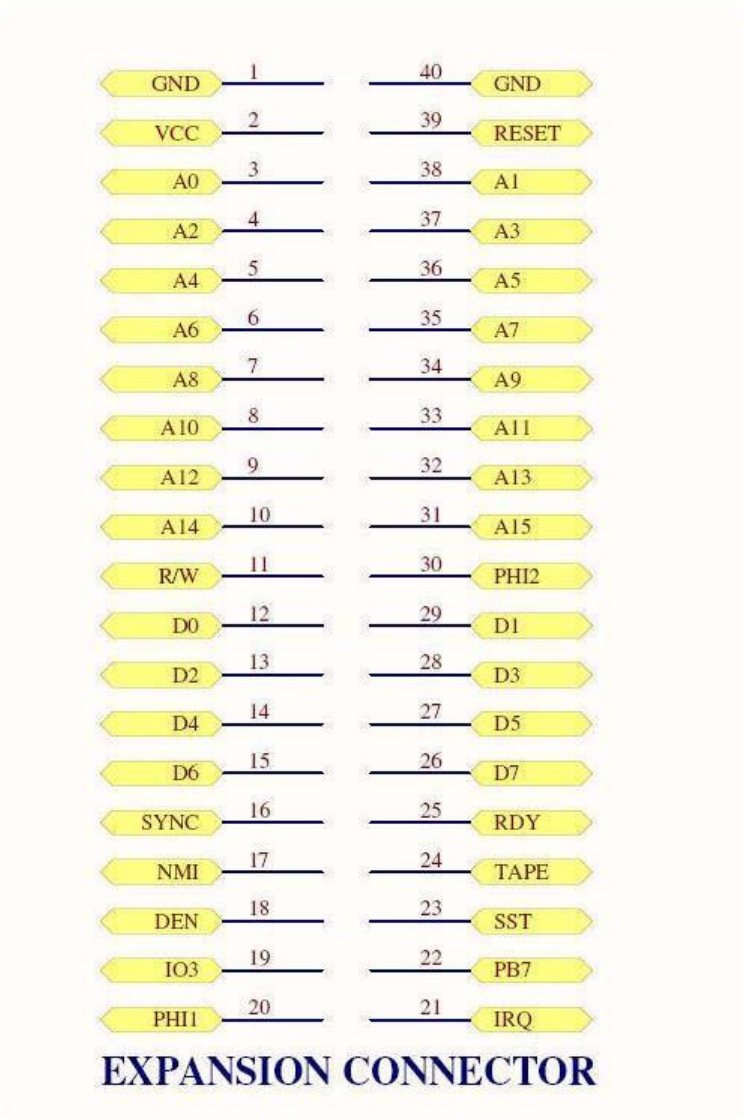


Figure 14: Expansion connector

Looking at figure 14, there are 40 pins. Pin 1 is labeled on the board and is near the edge at the upper-left corner of the connector. Here is a description of the 40 pins

1 and 40	there are two pins for ground to give proper grounding to optional expansion boards.
2	VCC, this is a 5V signal which powers the circuit
3-14, 31-38	These are the CPU address lines A0-A15 used to address memory or devices
11	R/W this is the read/write signal. Low when writing, high when reading memory
12-15, 26-29	These are the CPU data bus. Used to transfer data to/from RAM/EPROM or devices
16	Sync. This signal goes high during when an instruction is being fetched for the CPU
17	NMI. Non-Maskable Interrupt signal to the CPU. Active low to generate
18	DEN, Onboard memory decode Enable line. Control the enabling of the onboard memory
19	IO3 is the pre-decoded signal for the 2nd optional 6532. Attach to CS1 pin 38 on 6532
20	PHI1 Phase 1 clock signal. 180 degrees from phase 2
21	IRQ Interrupt request signal. Active low generates an IRQ.
22	PB7 is I/O port pin PB7 from 6532 required to complete cassette interface
23	SST Single step signal used to control CPU with single step
24	TAPE this signal is used to complete the cassette interface. See figure 15 for future information
25	RDY used to stop the CPU in single step circuit
30	PHI2 phase 2 main clock signal to the 6502
39	RESET 6502 RESET line, when pulled low will reset the 6502

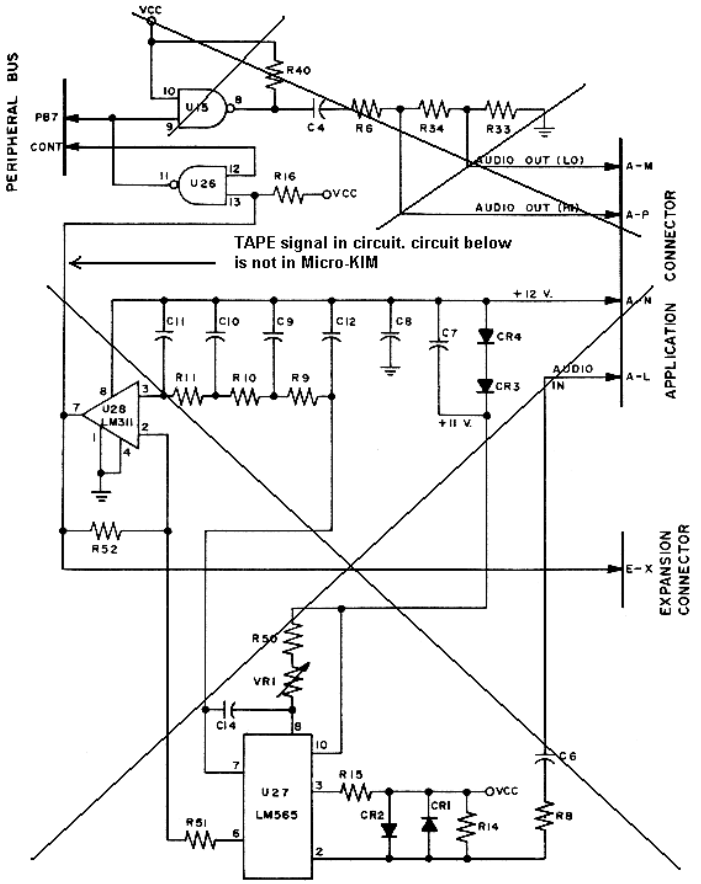
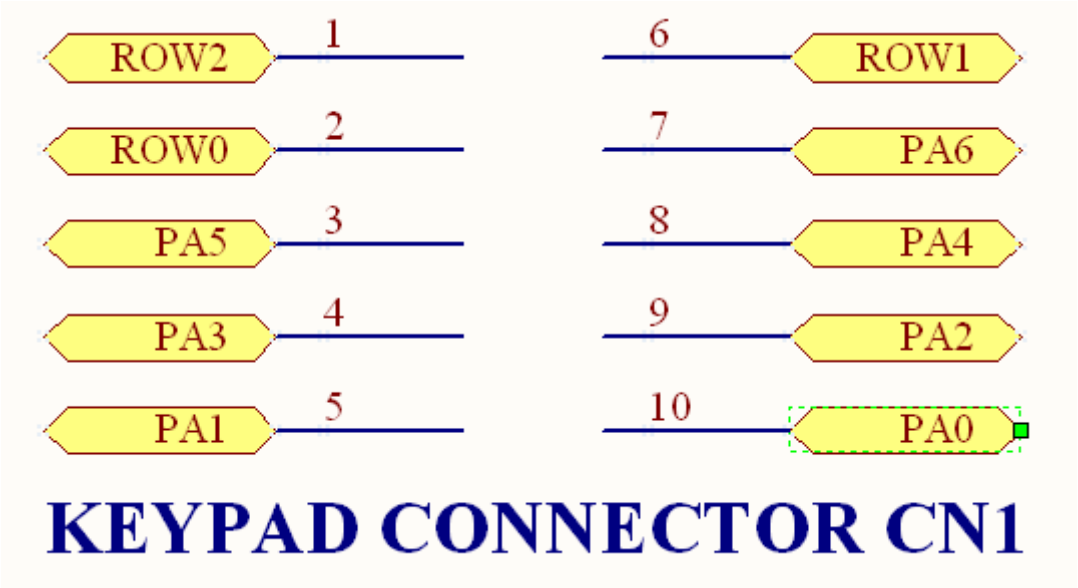


Figure 15: KIM-1 cassette circuit showing removed section and TAPE signal location



The keypad connector is between the 1MHz oscillator and the keypad. It is there for those who wish to add an external keypad to their Micro-KIM.

Appendix D: Monitor Listing

```
CARD # LOC     CODE          CARD
    3                  ;              666666  555555  333333  000000
    4                  ;              6       5           3   0    0
    5                  ;              6       5           3   0    0
    6                  ;              666666  555555  333333  0    0
    7                  ;              6    6      5       3   0    0
    8                  ;              6    6      5       3   0    0
    9                  ;              666666  666666  666666  000000
   10                  ;
   11                  ;
   12                  ;
   13                  ;                      000000  000000  333333
   14                  ;                      0    0  0    0       3
   15                  ;              ------  0    0  0    0       3
   16                  ;              ------  0    0  0    0  333333
   17                  ;              ------  0    0  0    0       3
   18                  ;                      0    0  0    0       3
   19                  ;                      000000  000000  333333
   20                  ;
   21                  ;
   22                  ;
   23                  ;
   24                  ;
   25                  ;      COPYRIGHT
   26                  ;      MOS TECHNOLOGY, INC
   27                  ;      DATE: OCT 18, 1975 REV-D
   28                  ;
   29                  ;
   30                  ;
   31                  ;      6530-003 I.C. IS AN AUDIO CASSETT TAPE
   32                  ;      RECORDER ENTENSION OF THE BASIC
   33                  ;      KIM MONITOR
   34                  ;
   35                  ;      IT FEATURES TWO BASIC ROUTINES
   36                  ;      LOADT-LOAD MEM FROM AUDIO TAPE
   37                  ;      DUMPT-STOR MEM ONTO AUDIO TAPE
   38                  ;
   39                  ;      LOADT
   40                  ;        ID=00     IGNORE ID
   41                  ;        ID=FF     IGN. ID USE SA FOR START ADDR
   42                  ;        ID=01-FE  IGN.ID USE ADDRESS ON TAPE
   43                  ;
   44                  ;      DUMPT
   45                  ;        ID=00     SHOULD NOT BE USED
   46                  ;        ID=FF     SHOULD NOT BE USED
   47                  ;        ID=01-FE  NORMAL ID RANGE
   48                  ;        SAL       LSB STARTING ADDRESS OF PROGRAM
   49                  ;        SAH       MSB
   50                  ;        EAL       ENDING ADDRESS OF PROGRAM
   51                  ;        EAH       MSB
   52                  ;
```

```
CARD # LOC     CODE          CARD
  54                  ;
  55                  ;       EQUATES
  56                  ;       SET UP FOR 6530-002 I/O
  57                  ;
  58                  SAD     =$1740          6530 A DATA
  59                  PADD    =$1741          6530 A DATA DIRECTION
  60                  SBD     =$1742          6530 B DATA
  61                  PBDD    =$1743          6530 B DATA DIRECTION
  62                  CLK1T   =$1744          DIV BY 1 TIME
  63                  CLK8T   =$1745          DIV BY 8 TIME
  64                  CLK64T  =$1746          DIV BY 64 TIME
  65                  CLKKT   =$1747          DIV BY 1024 TIME
  66                  CLKRDI  =$1747          READ TIME OUT BIT
  67                  CLKRDT  =$1746          READ TIME
  68                  ;
  69   0000                   *=$00EF
  70                  ;       MPU REG.  SAVX AREA IN PAGE 0
  71                  ;
  72   00EF           PCL     *=*+1 PROGRAM CNT LOW
  73   00F0           PCH     *=*+1 PROGRAM CNT HI
  74   00F1           PREG    *=*+1 CURRENT STATUS REG
  75   00F2           SPUSER  *=*+1 CURRENT STACK POINTER
  76   00F3           ACC     *=*+1 ACCUMULATOR
  77   00F4           YREG    *=*+1 Y INDEX
  78   00F5           XREG    *=*+1 X INDEX
  79                  ;
  80                  ;       KIM FIXED AREA IN PAGE 0
  81                  ;
  82   00F6           CHKHI   *=*+1
  83   00F7           CHKSUM  *=*+1
  84   00F8           INL     *=*+1 INPUT BUFFER
  85   00F9           INH     *=*+1 INPUT BUFFER
  86   00FA           POINTL  *=*+1 LSB OF OPEN CELL
  87   00FB           POINTH  *=*+1 MSB OF OPEN CELL
  88   00FC           TEMP    *=*+1
  89   00FD           TMPX    *=*+1
  90   00FE           CHAR    *=*+1
  91   00FF           MODE    *=*+1
  92                  ;
  93                  ;       KIM FIXED AREA IN PAGE 23
  94                  ;
  95   0100                   *=$17E7
  96   17E7           CHKL    *=*+1
  97   17E8           CHKH    *=*+1           CHKSUM
  98   17E9           SAVX    *=*+3
  99   17EC           VEB     *=*+6           VOLATILE EXECUTION BLOCK
 100   17F2           CNTL30  *=*+1           TTY DELAY
 101   17F3           CNTH30  *=*+1           TTY DELAY
 102   17F4           TIMH    *=*+1
 103   17F5           SAL     *=*+1           LOW STARTING ADDRESS
 104   17F6           SAH     *=*+1           HI STARTING ADDRESS
 105   17F7           EAL     *=*+1           LOW ENDING ADDRESS
 106   17F8           EAH     *=*+1           HI ENDING ADDRESS
 107   17F9           ID      *=*+1           TAPE PROGRAM ID NUMBER
 108                  ;
 109                  ;       INTERRUPT VECTORS
 110                  ;
 111   17FA           NMIV    *=*+2           STOP VECTOR (STOP=1C00)
```

```
CARD # LOC     CODE          CARD
 112   17FC           RSTV    *=*+2           RST VECTOR
 113   17FE           IRQV    *=*+2           IRQ VECTOR (BRK= 1C00)
 114                  ;
 116   1800                   *=$1800
 117                  ;
 118                  ;       INIT VOLATILE EXECUTION BLOCK
 119                  ;       DUMP MEM TO TAPE
 120                  ;
 121  1800  A9 AD     DUMPT   LDA   #$AD      LOAD ABSOLUTE INST
 122  1802  8D EC 17          STA   VEB
 123  1805  20 32 19          JSR   INTVEB
 124                  ;
 125  1808  A9 27             LDA   #$27      TURN OFF DATAIN PB5
 126  180A  8D 42 17          STA   SBD
 127  180D  A9 BF             LDA   #$BF      CONVERT PB7 TO OUTPUT
 128  180F  8D 43 17          STA   PBDD
 129                  ;
 130  1812  A2 64             LDX   #$64      100 CHARS
 131  1814  A9 16     DUMPT1  LDA   #$16      SYNC CHAR'S
 132  1816  20 7A 19          JSR   OUTCHT
 133  1819  CA               DEX
 134  181A  D0 F8             BNE   DUMPT1
 135                  ;
 136
 137  181C  A9 2A             LDA   #$2A      START CHAR
 138  181E  20 7A 19          JSR   OUTCHT
 139                  ;
 140  1821  AD F9 17          LDA   ID        OUTPUT ID
 141  1824  20 61 19          JSR   OUTBT
 142                  ;
 143  1827  AD F5 17          LDA   SAL       OUTPUT STARTING
 144  182A  20 5E 19          JSR   OUTBTC    ADDRESS
 145  182D  AD F6 17          LDA   SAH
 146  1830  20 5E 19          JSR   OUTBTC
 147                  ;
 148  1833  AD ED 17  DUMPT2  LDA   VEB+1     CHECK FOR LAST
 149  1836  CD F7 17          CMP   EAL       DATA BYTE
 150  1839  AD EE 17          LDA   VEB+2
 151  183C  ED F8 17          SBC   EAH
 152  183F  90 24             BCC   DUMPT4
 153                  ;
 154  1841  A9 2F             LDA   #'/       OUTPUT END OF DATA CHAR
 155  1843  20 7A 19          JSR   OUTCHT
 156  1846  AD E7 17          LDA   CHKL      LAST BYTE HAS BEEN
 157  1849  20 61 19          JSR   OUTBT     OUTPUT    NOW OUTPUT
 158  184C  AD E8 17          LDA   CHKH      CHKSUM
 159  184F  20 61 19          JSR   OUTBT
 160                  ;
 161                  ;
 162  1852  A2 02             LDX   #$02      2 CHAR'S
 163  1854  A9 04     DUMPT3  LDA   #$04      EOT CHAR
 164  1856  20 7A 19          JSR   OUTCHT
 165  1859  CA               DEX
 166  185A  D0 F8             BNE   DUMPT3
 167                  ;
 168  185C  A9 00             LDA   #$00      DISPLAY 0000
 169  185E  85 FA             STA   POINTL    FOR NORMAL EXIT
 170  1860  85 FB             STA   POINTH
```

```
CARD # LOC     CODE          CARD
 171  1862  4C 4F 1C          JMP   START
 172                    ;
 173  1865  20 EC 17  DUMPT4   JSR   VEB        DATA BYTE OUTPUT
 174  1868  20 5E 19          JSR   OUTBTC
 175                    ;
 176  186B  20 EA 19          JSR   INCVEB
 177  186E  4C 33 18          JMP   DUMPT2
 178                    ;
 179                    ;       LOAD MEMORY FROM TAPE
 180                    ;
 181                    ;
 182  1871  0F 19      TAB     .WORD LOAD12
 183  1873  A9 8D      LOADT   LDA   #$8D       INIT VOLATILE EXECUTION
 184  1875  8D EC 17          STA   VEB        BLOCK WITH STA ABS.
 185  1878  20 32 19          JSR   INTVEB
 186                    ;
 187  187B  A9 4C             LDA   #$4C       JUMP TYPE RTRN
 188  187D  8D EF 17          STA   VEB+3
 189  1880  AD 71 18          LDA   TAB
 190  1883  8D F0 17          STA   VEB+4
 191  1886  AD 72 18          LDA   TAB+1
 192  1889  8D F1 17          STA   VEB+5
 193                    ;
 194  188C  A9 07             LDA   #$07       RESET PB5=0 (DATA-IN)
 195  188E  8D 42 17          STA   SBD
 196                    ;
 197  1891  A9 FF      SYNC    LDA   #$FF       CLEAR SAVX FOR SYNC CHAR
 198  1893  8D E9 17          STA   SAVX
 199                    ;
 200  1896  20 41 1A  SYNC1   JSR   RDBIT      GET A BIT
 201  1899  4E E9 17          LSR   SAVX       SHIFT BIT INTO CHAR
 202  189C  0D E9 17          ORA   SAVX
 203  189F  8D E9 17          STA   SAVX
 204  18A2  AD E9 17          LDA   SAVX       GET NEW CHAR
 205  18A5  C9 16             CMP   #$16       SYNC CHAR
 206  18A7  D0 ED             BNE   SYNC1
 207                    ;
 208  18A9  A2 0A             LDX   #$0A       TEST FOR 10 SYNC CHARS
 209  18AB  20 24 1A  SYNC2   JSR   RDCHT
 210  18AE  C9 16             CMP   #$16
 211  18B0  D0 DF             BNE   SYNC       IF NOT 10 CHAR, RE-SYNC
 212  18B2  CA                DEX
 213  18B3  D0 F6             BNE   SYNC2
 214                    ;
 215                    ;
 216  18B5  20 24 1A  LOADT4  JSR   RDCHT      LOOK FOR START OF
 217  18B8  C9 2A             CMP   #$2A       DATA CHAR
 218  18BA  F0 06             BEQ   LOAD11
 219  18BC  C9 16             CMP   #$16       IF NOT , SHOULD BE SYN
 220  18BE  D0 D1             BNE   SYNC
 221  18C0  F0 F3             BEQ   LOADT4
 222                    ;
 223  18C2  20 F3 19  LOAD11  JSR   RDBYT     READ ID FROM TAPE
 224  18C5  CD F9 17          CMP   ID         COMPARE WITH REQUESTED ID
 225  18C8  F0 0D             BEQ   LOADT5
 226  18CA  AD F9 17          LDA   ID         DEFAULT 00, READ RECORD
 227  18CD  C9 00             CMP   #$00       ANYWAY
 228  18CF  F0 06             BEQ   LOADT5
```

```
CARD # LOC     CODE          CARD
 229  18D1  C9 FF            CMP   #$FF      DEFAULT FF, IGNORE SA ON
 230  18D3  F0 17            BEQ   LOADT6    TAPE
 231  18D5  D0 9C            BNE   LOADT
 232                    ;
 233  18D7  20 F3 19  LOADT5  JSR   RDBYT     GET SA FROM TAPE
 234  18DA  20 4C 19          JSR   CHKT
 235  18DD  8D ED 17          STA   VEB+1     SAVX IN VEB+1,2
 236  18E0  20 F3 19          JSR   RDBYT
 237  18E3  20 4C 19          JSR   CHKT
 238  18E6  8D EE 17          STA   VEB+2
 239  18E9  4C F8 18          JMP   LOADT7
 240                    ;
 241  18EC  20 F3 19  LOADT6  JSR   RDBYT     GET SA BUT IGNORE
 242  18EF  20 4C 19          JSR   CHKT
 243  18F2  20 F3 19          JSR   RDBYT
 244  18F5  20 4C 19          JSR   CHKT
 245                    ;
 246                    ;
 247  18F8  A2 02     LOADT7  LDX   #$02      GET 2 CHARS
 248  18FA  20 24 1A  LOAD13  JSR   RDCHT     GET CHAR (X)
 249  18FD  C9 2F             CMP   #$2F      LOOK FOR LAST CHAR
 250  18FF  F0 14             BEQ   LOADT8
 251  1901  20 00 1A          JSR   PACKT     CONVERT TO HEX
 252  1904  D0 23             BNE   LOADT9    Y=1 NON-HEX CHAR
 253  1906  CA                DEX
 254  1907  D0 F1             BNE   LOAD13
 255                    ;
 256  1909  20 4C 19          JSR   CHKT      COMPUTE CHECKSUM
 257  190C  4C EC 17          JMP   VEB       SAVX DATA IN MEMORY
 258  190F  20 EA 19  LOAD12  JSR   INCVEB    INCREMENT DATA POINTER
 259  1912  4C F8 18          JMP   LOADT7
 260                    ;
 261  1915  20 F3 19  LOADT8  JSR   RDBYT     END OF DATA, COMPARE CHKSUM
 262  1918  CD E7 17          CMP   CHKL
 263  191B  D0 0C             BNE   LOADT9
 264  191D  20 F3 19          JSR   RDBYT
 265  1920  CD E8 17          CMP   CHKH
 266  1923  D0 04             BNE   LOADT9
 267  1925  A9 00             LDA   #$00      NORMAL EXIT
 268  1927  F0 02             BEQ   LOAD10
 269                    ;
 270  1929  A9 FF     LOADT9  LDA   #$FF      ERROR EXIT
 271  192B  85 FA     LOAD10  STA   POINTL
 272  192D  85 FB             STA   POINTH
 273  192F  4C 4F 1C          JMP   START
 274                    ;
 276                    ;
 277                    ;       SUBROUTINES FOLLOW
 278                    ;
 279                    ;       SUB TO MOVE SA TO VEB+1,2
 280                    ;
 281  1932  AD F5 17  INTVEB  LDA   SAL       MOVE SA TO VEB+1,2
 282  1935  8D ED 17          STA   VEB+1
 283  1938  AD F6 17          LDA   SAH
 284  193B  8D EE 17          STA   VEB+2
 285  193E  A9 60             LDA   #$60      RTS INST
 286  1940  8D EF 17          STA   VEB+3
 287  1943  A9 00             LDA   #$00      CLEAR CHKSUM AREA
```

```
CARD # LOC     CODE          CARD
 288  1945  8D E7 17          STA   CHKL
 289  1948  8D E8 17          STA   CHKH
 290  194B  60                RTS
 291                  ;
 292                  ;       COMPUTE CHKSUM FOR TAPE LOAD
 293                  ;       RTN USES Y TO SAVEX A
 294                  ;
 295  194C  A8        CHKT    TAY
 296  194D  18                CLC
 297  194E  6D E7 17          ADC   CHKL
 298  1951  8D E7 17          STA   CHKL
 299  1954  AD E8 17          LDA   CHKH
 300  1957  69 00             ADC   #$00
 301  1959  8D E8 17          STA   CHKH
 302  195C  98                TYA
 303  195D  60                RTS
 304                  ;
 305                  ;       OUTPUT ONE BYTE USE Y
 306                  ;       TO SAVX BYTE
 307                  ;
 308  195E  20 4C 19  OUTBTC  JSR   CHKT      COMPARE CHKSUM
 309  1961  A8        OUTBT   TAY             SAVX DATA BYTE
 310  1962  4A                LSR   A         SHIFT OFF LSD
 311  1963  4A                LSR   A
 312  1964  4A                LSR   A
 313  1965  4A                LSR   A
 314  1966  20 6F 19          JSR   HEXOUT    OUTPUT MSD
 315  1969  98                TYA
 316  196A  20 6F 19          JSR   HEXOUT    OUTPUT LSD
 317  196D  98                TYA
 318  196E  60                RTS
 319                  ;
 320                  ;       CONVERT LSD OF A TO ASCII
 321                  ;       OUTPUT TO TAPE
 322                  ;
 323  196F  29 0F     HEXOUT  AND   #$0F
 324  1971  C9 0A             CMP   #$0A
 325  1973  18                CLC
 326  1974  30 02             BMI   HEX1
 327  1976  69 07             ADC   #$07
 328  1978  69 30     HEX1    ADC   #$30
 329                  ;
 330                  ;       OUTPUT TO TAPE ONE ASCII
 331                  ;       CHAR  USE SUB'S ONE + ZRO
 332                  ;
 333  197A  8E E9 17  OUTCHT  STX   SAVX
 334  197D  8C EA 17          STY   SAVX+1
 335  1980  A0 08             LDY   #$08      START BIT
 336  1982  20 9E 19  CHT1    JSR   ONE
 337  1985  4A                LSR   A         GET DATA BIT
 338  1986  B0 06             BCS   CHT2
 339  1988  20 9E 19          JSR   ONE       DATA BIT=1
 340  198B  4C 91 19          JMP   CHT3
 341  198E  20 C4 19  CHT2    JSR   ZRO       DATA BIT=0
 342  1991  20 C4 19  CHT3    JSR   ZRO
 343  1994  88                DEY
 344  1995  D0 EB             BNE   CHT1
 345  1997  AE E9 17          LDX   SAVX
```

```
CARD # LOC     CODE          CARD
 346  199A  AC EA 17          LDY   SAVX+1
 347  199D  60                RTS
 348                ;
 349                ;
 340                ;       OUTPUT 1 TO TAPE
 351                ;       9 PULSES, 138 MICROSEC EACH
 352                ;
 353  199E  A2 09     ONE     LDX   #$09
 354  19A0  48                PHA             SAVX A
 355  19A1  2C 47 17  ONE1    BIT   CLKRDI    WAIT FOR TIME OUT
 356  19A4  10 FB             BPL   ONE1
 357  19A6  A9 7E             LDA   #126
 358  19A8  8D 44 17          STA   CLK1T
 359  19AB  A9 A7             LDA   #$A7
 360  19AD  8D 42 17          STA   SBD       SET PB7=1
 361  19B0  2C 47 17  ONE2    BIT   CLKRDI
 362  19B3  10 FB             BPL   ONE2
 363  19B5  A9 7E             LDA   #126
 364  19B7  8D 44 17          STA   CLK1T
 365  19BA  A9 27             LDA   #$27
 366  19BC  8D 42 17          STA   SBD       RESET PB7=0
 367  19BF  CA                DEX
 368  19C0  D0 DF             BNE   ONE1
 369  19C2  68                PLA
 370  19C3  60                RTS
 371                ;
 372                ;
 373                ;       OUTPUT 0 TO TAPE
 374                ;       6 PULSES, 207 MICROSEC EACH
 375                ;
 376  19C4  A2 06     ZRO     LDX   #$06
 377  19C6  48                PHA             SAVX A
 378  19C7  2C 47 17  ZRO1    BIT   CLKRDI
 379  19CA  10 FB             BPL   ZRO1
 380  19CC  A9 C3             LDA   #$C3
 381  19CE  8D 44 17          STA   CLK1T
 382  19D1  A9 A7             LDA   #$A7
 383  19D3  8D 42 17          STA   SBD       SET PB7=1
 384  19D6  2C 47 17  ZRO2    BIT   CLKRDI
 385  19D9  10 FB             BPL   ZRO2
 386  19DB  A9 C3             LDA   #195
 387  19DD  8D 44 17          STA   CLK1T
 388  19E0  A9 27             LDA   #$27
 389  19E2  8D 42 17          STA   SBD       RESET PB7=0
 390  19E5  CA                DEX
 391  19E6  D0 DF             BNE   ZRO1
 392  19E8  68                PLA             RESTORE A
 393  19E9  60                RTS
 394                ;
 395                ;       SUB TO INC VEB+1,2
 396                ;
 397  19EA  EE ED 17  INCVEB  INC   VEB+1
 398  19ED  D0 03             BNE   INCVE1
 399  19EF  EE EE 17          INC   VEB+2
 400  19F2  60        INCVE1  RTS
 401                ;
 402                ;       SUB TO READ BYTE FROM TAPE
 403                ;
```

```
CARD # LOC     CODE          CARD
 404  19F3  20 24 1A  RDBYT   JSR   RDCHT
 405  19F6  20 00 1A          JSR   PACKT
 406  19F9  20 24 1A          JSR   RDCHT
 407  19FC  20 00 1A          JSR   PACKT
 408  19FF  60                RTS
 409                ;
 410                ;       PACK A=ASCII INTO SAVX
 411                ;       AS HEX DATA
 412                ;
 413  1A00  C9 30     PACKT   CMP   #$30
 414  1A02  30 1E             BMI   PACKT3
 415  1A04  C9 47             CMP   #$47
 416  1A06  10 1A             BPL   PACKT3
 417  1A08  C9 40             CMP   #$40
 418  1A0A  30 03             BMI   PACKT1
 419  1A0C  18                CLC
 420  1A0D  69 09             ADC   #$09
 421  1A0F  2A        PACKT1  ROL   A
 422  1A10  2A                ROL   A
 423  1A11  2A                ROL   A
 424  1A12  2A                ROL   A
 425  1A13  A0 04             LDY   #$04
 426  1A15  2A        PACKT2  ROL   A
 427  1A16  2E E9 17          ROL   SAVX
 428  1A19  88                DEY
 429  1A1A  D0 F9             BNE   PACKT2
 430  1A1C  AD E9 17          LDA   SAVX
 431  1A1F  A0 00             LDY   #$00      Y=0 VALID HEX CHAR
 432  1A21  60                RTS
 433  1A22  C8        PACKT3  INY             Y=1 NOT HEX
 434  1A23  60                RTS
 435                ;
 436                ;       GET 1 CHAR FROM TAPE AND RETURN
 437                ;       WITH CHAR IN A  USE SAVX+1 TO ASM CHAR
 438                ;
 439  1A24  8E EB 17  RDCHT   STX   SAVX+2
 440  1A27  A2 08             LDX   #$08      READ 8 BITS
 441  1A29  20 41 1A  RDCHT1  JSR   RDBIT     GET NEXT DATA BIT
 442  1A2C  4E EA 17          LSR   SAVX+1    RIGHT SHIFT CHAR
 443  1A2F  0D EA 17          ORA   SAVX+1    OR IN SIGN BIT
 444  1A32  8D EA 17          STA   SAVX+1    REPLACE CHAR
 445  1A35  CA                DEX
 446  1A36  D0 F1             BNE   RDCHT1
 447                ;
 448  1A38  AD EA 17          LDA   SAVX+1    MOVE CHAR INTO A
 449  1A3B  2A                ROL   A         SHIFT OFF PARITY
 450  1A3C  4A                LSR   A
 451  1A3D  AE EB 17          LDX   SAVX+2
 452  1A40  60                RTS
 453                ;
 454                ;       THIS SUB GETS ONE BIT FROM
 455                ;       TAPE AND RETURNS IT IN SIGN OF A
 456                ;
 457  1A41  2C 42 17  RDBIT   BIT   SBD       WAIT FOR END OF START BIT
 458  1A44  10 FB             BPL   RDBIT
 459  1A46  AD 46 17          LDA   CLKRDT    GET START BIT TIME
 460  1A49  A0 FF             LDY   #$FF      A=256-T1
 461  1A4B  8C 46 17          STY   CLK64T    SET UP TIMER
```

```
CARD # LOC     CODE          CARD
 462                  ;
 463  1A4E  A0 14             LDY   #$14
 464  1A50  88        RDBIT3  DEY             DELAY 100 MICROSEC
 465  1A51  D0 FD             BNE   RDBIT3
 466                  ;
 467  1A53  2C 42 17  RDBIT2  BIT   SBD
 468  1A56  30 FB             BMI   RDBIT2    WAIT FOR NEXT START BIT
 469
 470  1A58  38                SEC
 471  1A59  ED 46 17          SBC   CLKRDT    (256-T1)-(256-T2)=T2-T1
 472  1A5C  A0 FF             LDY   #$FF
 473  1A5E  8C 46 17          STY   CLK64T    SET UP TIMER FOR NEXT BIT
 474
 475  1A61  A0 07             LDY   #$07
 476  1A63  88        RDBIT4  DEY             DELAY 50 MICROSEC
 477  1A64  D0 FD             BNE   RDBIT4
 478                  ;
 479  1A66  49 FF             EOR   #$FF      COMPLEMENT SIGN OF A
 480  1A68  29 80             AND   #$80      MASK ALL EXCEPT SIGN
 481  1A6A  60                RTS
 483                  ;
 484                  ;       DIAGNOSTICS
 485                  ;          MEMORY
 486                  ;          PLLCAL
 487                  ;
 488                  ;
 489                  ;
 490                  ;       PLLCAL OUTPUT 166 MICROSEC
 491                  ;       PULSE STRING
 492                  ;
 493  1A6B  A9 27     PLLCAL  LDA   #$27
 494  1A6D  8D 42 17          STA   SBD       TURN OFF DATIN PB5=1
 495  1A70  A9 BF             LDA   #$BF      CONVERT PB7 TO OUTPUT
 496  1A72  8D 43 17          STA   PBDD
 497                  ;
 498  1A75  2C 47 17  PLL1    BIT   CLKRDI
 499  1A78  10 FB             BPL   PLL1
 500  1A7A  A9 9A             LDA   #154      WAIT 166 MICROSEC
 501  1A7C  8D 44 17          STA   CLK1T
 502  1A7F  A9 A7             LDA   #$A7      OUTPUT PB7=1
 503  1A81  8D 42 17          STA   SBD
 504                  ;
 505  1A84  2C 47 17  PLL2    BIT   CLKRDI
 506  1A87  10 FB             BPL   PLL2
 507  1A89  A9 9A             LDA   #154
 508  1A8B  8D 44 17          STA   CLK1T
 509  1A8E  A9 27             LDA   #$27      PB7=0
 510  1A90  8D 42 17          STA   SBD
 511  1A93  4C 75 1A          JMP   PLL1
 512                  ;
 513                  ;
 514                  ;       INTERRUPTS PAGE 27
 515                  ;
 516                          *=*+$164    RESERVED FOR TEST
 517  1BFA  6B 1A     NMIP27  .WORD PLLCAL
 518  1BFC  6B 1A     RSTP27  .WORD PLLCAL
 519  1BFE  6B 1A     IRQP27  .WORD PLLCAL
 520                  ;
```

```
CARD # LOC     CODE          CARD
 522                  ;
 523                  ;
 524                  ;
 525                  ;
 526                  ;              666666  555555  333333  000000
 527                  ;              6       5            3  0    0
 528                  ;              6       5            3  0    0
 529                  ;              666666  555555  333333  0    0
 530                  ;              6    6       5       3  0    0
 531                  ;              6    6       5       3  0    0
 532                  ;              666666  666666  666666  000000
 523                  ;
 534                  ;
 535                  ;
 536                  ;                      000000  000000  222222
 537                  ;                      0    0  0    0       2
 538                  ;              ------  0    0  0    0       2
 539                  ;              ------  0    0  0    0  222222
 540                  ;              ------  0    0  0    0  2
 541                  ;                      0    0  0    0  2
 542                  ;                      000000  000000  222222
 543                  ;
 545                  ;
 546                  ;
 547                  ;
 548                  ;         COPYRIGHT
 549                  ;         MOS TECHNOLOGY INC.
 550                  ;         DATE  OCT 13 1975    REV E
 551                  ;
 552                  ;    KIM   :TTY INTERFACE
 553                  ;          :KEYBOARD INTERFACE
 554                  ;          :7 SEG 6 DIGIT DISPLAY
 555                  ;
 556                  ;
 557                  ;    TTY CMDS:
 558                  ;          G   GOEXEC
 559                  ;          CR  OPEN NEXT CELL
 560                  ;          LF  OPEN PREV. CELL
 561                  ;          .   MODIFY OPEN CELL
 562                  ;          SP  OPEN NEW CELL
 563                  ;          L   LOAD (OBJECT FORMAT)
 564                  ;          Q   DUMP  FROM OPEN CELL ADDR TO HI LIMIT
 565                  ;          RO  RUB OUT  - RETURN TO START  (KIM)
 566                  ;              ((ALL ILLEGAL CHARS ARE IGNORED))
 567                  ;
 568                  ;    KEYBOARD COMMANDS:
 569                  ;        ADDR  SETS MODE TO MODIFY CELL ADDRESS
 570                  ;        DATA  SETS MODE TO MODIFY DATA IN OPEN CELL
 571                  ;        STEP  INCREMENTS TO NEXT CELL
 572                  ;        RST   SYSTEM RESET
 573                  ;        RUN   GOEXEC
 574                  ;        STOP  $1C00 CAN BE LOADED INTO NMIV TO
 575                  ;              USE STOP FEATURE
 576                  ;        PC    DISPLAY PC
 577                  ;
 578                  ;        CLOCK IS NOT DISABLED IN SIGMA 1
 579                  ;
 580                  ;
```

```
CARD # LOC     CODE          CARD
 581                  ;
 582                  ;
 584                          *=$1C00
 585                  ;
 586                  ;
 587  1C00  85 F3     SAVE    STA   ACC       KIM ENTRY VIA STOP (NMI)
 588  1C02  68                PLA             OR BRK (IRQ)
 589  1C03  85 F1             STA   PREG
 590  1C05  68                PLA             KIM ENTRY VIA JSR (A LOST)
 591  1C06  85 EF             STA   PCL
 592  1C08  85 FA             STA   POINTL
 593  1C0A  68                PLA
 594  1C0B  85 F0             STA   PCH
 595  1C0D  85 FB             STA   POINTH
 596  1C0F  84 F4             STY   YREG
 597  1C11  86 F5             STX   XREG
 598  1C13  BA                TSX
 599  1C14  86 F2             STX   SPUSER
 600  1C16  20 88 1E          JSR   INITS
 601  1C19  4C 4F 1C          JMP   START
 602                  ;
 603  1C1C  6C FA 17  NMIT    JMP   (NMIV)    NON-MASKABLE INTERRUPT TRAP
 604  1C1F  6C FE 17  IRQT    JMP   (IRQV)    INTERRUPT TRAP
 605                  ;
 606  1C22  A2 FF     RST     LDX   #$FF      KIM ENTRY VIA RST
 607  1C24  9A                TXS
 608  1C25  86 F2             STX   SPUSER
 609  1C27  20 88 1E          JSR   INITS
 610                  ;
 611                  ;
 612  1C2A  A9 FF             LDA   #$FF      COUNT START BIT
 613  1C2C  8D F3 17          STA   CNTH30    ZERO CNTH30
 614  1C2F  A9 01             LDA   #$01      MASK HI ORDER BITS
 615  1C31  2C 40 17  DET1    BIT   SAD       TEST
 616  1C34  D0 19             BNE   START     KEYBD SSW TEST
 617  1C36  30 F9             BMI   DET1      START BIT TEST
 618  1C38  A9 FC             LDA   #$FC
 619  1C3A  18        DET3    CLC             THIS LOOP COUNTS
 620  1C3B  69 01             ADC   #$01      THE START BIT TIME
 621  1C3D  90 03             BCC   DET2
 622  1C3F  EE F3 17          INC   CNTH30
 623  1C42  AC 40 17  DET2    LDY   SAD       CHECK FOR END OF START BIT
 624  1C45  10 F3             BPL   DET3
 625  1C47  8D F2 17          STA   CNTL30
 626  1C4A  A2 08             LDX   #$08
 627  1C4C  20 6A 1E          JSR   GET5      GET REST OF THE CHAR, TEST CHAR
 628                  ;
 629                  ;
 630                  ;
 631                  ;
 632                  ;
 633                  ;
 634                  ;       MAKE TTY/KB SELECTION
 635                  ;
 636  1C4F  20 8C 1E  START   JSR   INIT1
 637  1C52  A9 01             LDA   #$01
 638  1C54  2C 40 17          BIT   SAD
 639  1C57  D0 1E             BNE   TTYKB
```

```
CARD # LOC     CODE          CARD
 640  1C59  20 2F 1E          JSR   CRLF      PRT CR LF
 641  1C5C  A2 0A             LDX   #$0A      TYPE OUT KIM
 642  1C5E  20 31 1E          JSR   PRTST
 643  1C61  4C AF 1D          JMP   SHOW1
 644                    ;
 645  1C64  A9 00     CLEAR   LDA   #$00
 646  1C66  85 F8             STA   INL       CLEAR INPUT BUFFER
 647  1C68  85 F9             STA   INH
 648  1C6A  20 5A 1E  READ    JSR   GETCH     GET CHAR
 649  1C6D  C9 01             CMP   #$01
 650  1C6F  F0 06             BEQ   TTYKB
 651  1C71  20 AC 1F          JSR   PACK
 652  1C74  4C DB 1D          JMP   SCAN
 653                    ;
 654                    ;       MAIN ROUTINE FOR KEY BOARD
 655                    ;       AND DISPLAY
 656                    ;
 657  1C77  20 19 1F  TTYKB   JSR   SCAND     IF A=0 NO KEY
 658  1C7A  D0 D3             BNE   START
 659  1C7C  A9 01     TTYKB1  LDA   #$01
 660  1C7E  2C 40 17          BIT   SAD
 661  1C81  F0 CC             BEQ   START
 662  1C83  20 19 1F          JSR   SCAND
 663  1C86  F0 F4             BEQ   TTYKB1
 664  1C88  20 19 1F          JSR   SCAND
 665  1C8B  F0 EF             BEQ   TTYKB1
 666                    ;
 667  1C8D  20 6A 1F  GETK    JSR   GETKEY
 668  1C90  C9 15             CMP   #$15
 669  1C92  10 BB             BPL   START
 670  1C94  C9 14             CMP   #$14
 671  1C96  F0 44             BEQ   PCCMD     DISPLAY PC
 672  1C98  C9 10             CMP   #$10      ADDR MODE=1
 673  1C9A  F0 2C             BEQ   ADDRM
 674  1C9C  C9 11             CMP   #$11      DATA MODE=1
 675  1C9E  F0 2C             BEQ   DATAM
 676  1CA0  C9 12             CMP   #$12      STEP
 677  1CA2  F0 2F             BEQ   STEP
 678  1CA4  C9 13             CMP   #$13      RUN
 679  1CA6  F0 31             BEQ   GOV
 680  1CA8  0A        DATA    ASL   A         SHIFT CHAR INTO HIGH
 681  1CA9  0A                ASL   A         ORDER NIBBLE
 682  1CAA  0A                ASL   A
 683  1CAB  0A                ASL   A
 684  1CAC  85 FC             STA   TEMP      STORE IN TEMP
 685  1CAE  A2 04             LDX   #$04
 686  1CB0  A4 FF     DATA1   LDY   MODE      TEST MODE 1=ADDR
 687  1CB2  D0 0A             BNE   ADDR      MODE=0 DATA
 688  1CB4  B1 FA             LDA   (POINTL),Y GET DATA
 689  1CB6  06 FC             ASL   TEMP      SHIFT CHAR
 690  1CB8  2A                ROL   A         SHIFT DATA
 691  1CB9  91 FA             STA   (POINTL),Y STORE OUT DATA
 692  1CBB  4C C3 1C          JMP   DATA2
 693                    ;
 694  1CBE  0A        ADDR    ASL   A         SHIFT CHAR
 695  1CBF  26 FA             ROL   POINTL    SHIFT ADDR
 696  1CC1  26 FB             ROL   POINTH    SHIFT ADDR HI
 697  1CC3  CA        DATA2   DEX
```

```
CARD # LOC     CODE          CARD
 698  1CC4  D0 EA            BNE   DATA1     DO 4 TIMES
 699  1CC6  F0 08            BEQ   DATAM2    EXIT HERE
 700                  ;
 701  1CC8  A9 01     ADDRM   LDA   #$01
 702  1CCA  D0 02            BNE   DATAM1
 703                  ;
 704  1CCC  A9 00     DATAM   LDA   #$00
 705  1CCE  85 FF     DATAM1  STA   MODE
 706  1CD0  4C 4F 1C  DATAM2  JMP   START
 707                  ;
 708  1CD3  20 63 1F  STEP    JSR   INCPT
 709  1CD6  4C 4F 1C          JMP   START
 710                  ;
 711  1CD9  4C C8 1D  GOV     JMP   GOEXEC
 712                  ;
 713                  ;
 714                  ;       DISPLAY PC BY MOVING
 715                  ;       PC TO POINT
 716                  ;
 717  1CDC  A5 EF     PCCMD   LDA   PCL
 718  1CDE  85 FA             STA   POINTL
 719  1CE0  A5 F0             LDA   PCH
 720  1CE2  85 FB             STA   POINTH
 721  1CE4  4C 4F 1C          JMP   START
 722                  ;
 723                  ;       LOAD PAPER TAPE FROM TTY
 724                  ;
 725  1CE7  20 5A 1E  LOAD    JSR   GETCH     LOOK FOR FIRST CHAR
 726  1CEA  C9 3B             CMP   #$3B      SMICOLON
 727  1CEC  D0 F9             BNE   LOAD
 728  1CEE  A9 00             LDA   #$00
 729  1CF0  85 F7             STA   CHKSUM
 730  1CF2  85 F6             STA   CHKHI
 731                  ;
 732  1CF4  20 9D 1F          JSR   GETBYT    GET BYTE CNT
 733  1CF7  AA                TAX             SAVE IN X INDEX
 734  1CF8  20 91 1F          JSR   CHK       COMPUTE CHKSUM
 735                  ;
 736  1CFB  20 9D 1F          JSR   GETBYT    GET ADDRESS HI
 737  1CFE  85 FB             STA   POINTH
 738  1D00  20 91 1F          JSR   CHK
 739  1D03  20 9D 1F          JSR   GETBYT    GET ADDRESS LO
 740  1D06  85 FA             STA   POINTL
 741  1D08  20 91 1F          JSR   CHK
 742                  ;
 743  1D0B  8A                TXA             IF CNT=0 DONT
 744  1D0C  F0 0F             BEQ   LOAD3     GET ANY DATA
 745                  ;
 746  1D0E  20 9D 1F  LOAD2   JSR   GETBYT    GET DATA
 747  1D11  91 FA             STA   (POINTL),Y STORE DATA
 748  1D13  20 91 1F          JSR   CHK
 749  1D16  20 63 1F          JSR   INCPT     NEXT ADDRESS
 750  1D19  CA                DEX
 751  1D1A  D0 F2             BNE   LOAD2
 752  1D1C  E8                INX             X=1 DATA RECORD
 753                  ;                       X=0 LAST RECORD
 754  1D1D  20 9D 1F  LOAD3   JSR   GETBYT    COMPARE CHKSUM
 755  1D20  C5 F6             CMP   CHKHI
```

```
CARD # LOC     CODE          CARD
 756  1D22  D0 17            BNE   LOADE1
 757  1D24  20 9D 1F         JSR   GETBYT
 758  1D27  C5 F7            CMP   CHKSUM
 759  1D29  D0 13            BNE   LOADER
 760                ;
 761  1D2B  8A              TXA             X=0 LAST RECORD
 762  1D2C  D0 B9            BNE   LOAD
 763                ;
 764  1D2E  A2 0C    LOAD7   LDX   #$0C      X-OFF KIM
 765  1D30  A9 27    LOAD8   LDA   #$27
 766  1D32  8D 42 17         STA   SBD       DISABLE DATA IN
 767  1D35  20 31 1E         JSR   PRTST
 768  1D38  4C 4F 1C         JMP   START
 769                ;
 770  1D3B  20 9D 1F LOADE1  JSR   GETBYT    DUMMY
 771  1D3E  A2 11    LOADER  LDX   #$11      X-OFF ERR KIM
 772  1D40  D0 EE            BNE   LOAD8
 773                ;
 774                ;       DUMP TO TTY
 775                ;       FROM OPEN CELL ADDRESS
 776                ;       TO LIMHL,LIMHH
 777                ;
 778  1D42  A9 00    DUMP    LDA   #$00
 779  1D44  85 F8            STA   INL
 780  1D46  85 F9            STA   INH       CLEAR RECORD COUNT
 781  1D48  A9 00    DUMP0   LDA   #$00
 782  1D4A  85 F6            STA   CHKHI     CLEAR CHKSUM
 783  1D4C  85 F7            STA   CHKSUM
 784                ;
 785  1D4E  20 2F 1E         JSR   CRLF      PRINT CR LF
 786  1D51  A9 3B            LDA   #$3B      PRINT SEMICOLON
 787  1D53  20 A0 1E         JSR   OUTCH
 788  1D56  A5 FA            LDA   POINTL    TEST POINT GT OR ET
 789  1D58  CD F7 17         CMP   EAL       HI LIMIT GOTO EXIT
 790  1D5B  A5 FB            LDA   POINTH
 791  1D5D  ED F8 17         SBC   EAH
 792  1D60  90 18            BCC   DUMP4
 793                ;
 794  1D62  A9 00            LDA   #$00       PRINT LAST RECORD
 795  1D64  20 3B 1E         JSR   PRTBYT    0 BYTES
 796  1D67  20 CC 1F         JSR   OPEN
 797  1D6A  20 1E 1E         JSR   PRTPNT
 798                ;
 799  1D6D  A5 F6            LDA   CHKHI     PRINT CHKSUM
 800  1D6F  20 3B 1E         JSR   PRTBYT    FOR LAST RECORD
 801  1D72  A5 F7            LDA   CHKSUM
 802  1D74  20 3B 1E         JSR   PRTBYT
 803  1D77  4C 64 1C         JMP   CLEAR
 804                ;
 805  1D7A  A9 18    DUMP4   LDA   #$18      PRINT 24 BYTE COUNT
 806  1D7C  AA              TAX             SAVE AS INDEX
 807  1D7D  20 3B 1E         JSR   PRTBYT
 808  1D80  20 91 1F         JSR   CHK
 809  1D83  20 1E 1E         JSR   PRTPNT
 810                ;
 811  1D86  A0 00    DUMP2   LDY   #$00      PRINT 24 BYTES
 812  1D88  B1 FA            LDA   (POINTL),Y GET DATA
 813  1D8A  20 3B 1E         JSR   PRTBYT    PRINT DATA
```

```
CARD # LOC     CODE          CARD
 814  1D8D  20 91 1F          JSR   CHK       COMPUTE CHKSUM
 815  1D90  20 63 1F          JSR   INCPT     INCREMENT POINT
 816  1D93  CA                DEX
 817  1D94  D0 F0             BNE   DUMP2
 818                    ;
 819  1D96  A5 F6             LDA   CHKHI     PRINT CHKSUM
 820  1D98  20 3B 1E          JSR   PRTBYT
 821  1D9B  A5 F7             LDA   CHKSUM
 822  1D9D  20 3B 1E          JSR   PRTBYT
 823  1DA0  E6 F8             INC   INL       INCR RECORD COUNT
 824  1DA2  D0 02             BNE   DUMP3
 825  1DA4  E6 F9             INC   INH
 826  1DA6  4C 48 1D  DUMP3   JMP   DUMP0
 827                    ;
 828  1DA9  20 CC 1F  SPACE   JSR   OPEN      OPEN NEW CELL
 829  1DAC  20 2F 1E  SHOW    JSR   CRLF      PRINT CR LF
 830  1DAF  20 1E 1E  SHOW1   JSR   PRTPNT
 831  1DB2  20 9E 1E          JSR   OUTSP     PRINT SPACE
 832  1DB5  A0 00             LDY   #$00      PRINT DATA SPECIFIED
 833  1DB7  B1 FA             LDA   (POINTL),Y BY POINT AD=LDA EXT
 834  1DB9  20 3B 1E          JSR   PRTBYT
 835  1DBC  20 9E 1E          JSR   OUTSP     PRINT SPACE
 836  1DBF  4C 64 1C          JMP   CLEAR
 837                    ;
 838  1DC2  20 63 1F  RTRN    JSR   INCPT     OPEN NEXT CELL
 839  1DC5  4C AC 1D          JMP   SHOW
 840                    ;
 841  1DC8  A6 F2     GOEXEC  LDX   SPUSER
 842  1DCA  9A                TXS
 843  1DCB  A5 FB             LDA   POINTH    PROGRAM RUNS FROM
 844  1DCD  48                PHA             OPEN CELL ADDRESS
 845  1DCE  A5 FA             LDA   POINTL
 846  1DD0  48                PHA
 847  1DD1  A5 F1             LDA   PREG
 848  1DD3  48                PHA
 849  1DD4  A6 F5             LDX   XREG      RESTORE REGS
 850  1DD6  A4 F4             LDY   YREG
 851  1DD8  A5 F3             LDA   ACC
 852  1DDA  40                RTI
 853                    ;
 854  1DDB  C9 20     SCAN    CMP   #$20      OPEN CELL
 855  1DDD  F0 CA             BEQ   SPACE
 856  1DDF  C9 7F             CMP   #$7F      RUB OUT (KIM)
 857  1DE1  F0 1B             BEQ   STV
 858  1DE3  C9 0D             CMP   #$0D      NEXT CELL
 859  1DE5  F0 DB             BEQ   RTRN
 860  1DE7  C9 0A             CMP   #$0A      PREV CELL
 861  1DE9  F0 1C             BEQ   FEED
 862  1DEB  C9 2E             CMP   #'.       MODIFY CELL
 863  1DED  F0 26             BEQ   MODIFY
 864  1DEF  C9 47             CMP   #'G       GO EXEC
 865  1DF1  F0 D5             BEQ   GOEXEC
 866  1DF3  C9 51             CMP   #'Q       DUMP FROM OPEN CELL TO HI LIMIT
 867  1DF5  F0 0A             BEQ   DUMPV
 868  1DF7  C9 4C             CMP   #'L       LOAD TAPE
 869  1DF9  F0 09             BEQ   LOADV
 870  1DFB  4C 6A 1C          JMP   READ      IGNORE ILLEGAL CHAR
 871                    ;
```

```
CARD # LOC     CODE          CARD
 872  1DFE  4C 4F 1C  STV     JMP   START
 873  1E01  4C 42 1D  DUMPV   JMP   DUMP
 874  1E04  4C E7 1C  LOADV   JMP   LOAD
 875                  ;
 876  1E07  38        FEED    SEC
 877  1E08  A5 FA             LDA   POINTL    DEC DOUBLE BYTE
 878  1E0A  E9 01             SBC   #$01      AT POINTL AND POINTH
 879  1E0C  85 FA             STA   POINTL
 880  1E0E  B0 02             BCS   FEED1
 881  1E10  C6 FB             DEC   POINTH
 882  1E12  4C AC 1D  FEED1   JMP   SHOW
 883                  ;
 884  1E15  A0 00     MODIFY  LDY   #$00      GET CONTENTS OF INPUT BUFF
 885  1E17  A5 F8             LDA   INL       INL AND STORE IN LOC
 886  1E19  91 FA             STA   (POINTL),Y SPECIFIED BY POINT
 887  1E1B  4C C2 1D          JMP   RTRN
 888                  ;
 889                  ;       END OF MAIN LINE
 891                  ;       SUBROUTINES FOLLOW
 892                  ;
 893                  ;
 894                  ;
 895                  ;       SUB TO PRINT POINTL,POINTH
 896                  ;
 897  1E1E  A5 FB     PRTPNT  LDA   POINTH
 898  1E20  20 3B 1E          JSR   PRTBYT
 899  1E23  20 91 1F          JSR   CHK
 900  1E26  A5 FA             LDA   POINTL
 901  1E28  20 3B 1E          JSR   PRTBYT
 902  1E2B  20 91 1F          JSR   CHK
 903  1E2E  60                RTS
 904                  ;
 905                  ;       PRINT STRING OF ASCII CHARS FROM
 906                  ;       TOP+X TO TOP
 907                  ;
 908  1E2F  A2 07     CRLF    LDX   #$07
 909  1E31  BD D5 1F  PRTST   LDA   TOP,X
 910  1E34  20 A0 1E          JSR   OUTCH
 911  1E37  CA                DEX
 912  1E38  10 F7             BPL   PRTST     STOP ON INDEX ZERO
 913  1E3A  60                RTS
 914                  ;
 915                  ;       PRINT 1 HEX BYTE AS TWO ASCII CHAR'S
 916                  ;
 917  1E3B  85 FC     PRTBYT  STA   TEMP
 918  1E3D  4A                LSR   A         SHIFT CHAR RIGHT 4 BITS
 919  1E3E  4A                LSR   A
 920  1E3F  4A                LSR   A
 921  1E40  4A                LSR   A
 922  1E41  20 4C 1E          JSR   HEXTA     CONVERT TO HEX AND PRINT
 923  1E44  A5 FC             LDA   TEMP      GET OTHER HALF
 924  1E46  20 4C 1E          JSR   HEXTA     CONVERT TO HEX AND PRINT
 925  1E49  A5 FC             LDA   TEMP      RESTORE BYTE IN A AND RETURN
 926  1E4B  60                RTS
 927                  ;
 928  1E4C  29 0F     HEXTA   AND   #$0F      MASK HI 4 BITS
 929  1E4E  C9 0A             CMP   #$0A
 930  1E50  18                CLC
```

```
CARD # LOC     CODE          CARD
 931  1E51  30 02            BMI   HEXTA1
 932  1E53  69 07            ADC   #$07      ALPHA HEX
 933  1E55  69 30     HEXTA1 ADC   #$30      DEC HEX
 934  1E57  4C A0 1E         JMP   OUTCH     PRINT CHAR
 935                ;
 936                ;       GET 1 CHAR FROM TTY
 937                ;       CHAR IN A
 938                ;       X IS PRESERVED AND Y RETURNED = FF
 939                ;
 940  1E5A  86 FD     GETCH  STX   TMPX      SAVE X REG
 941  1E5C  A2 08            LDX   #$08      SET UP 8-BIT CNT
 942  1E5E  A9 01            LDA   #$01
 943  1E60  2C 40 17  GET1   BIT   SAD
 944  1E63  D0 22            BNE   GET6
 945  1E65  30 F9            BMI   GET1      WAIT FOR START BIT
 946  1E67  20 D4 1E         JSR   DELAY     DELAY 1 BIT
 947  1E6A  20 EB 1E  GET5   JSR   DEHALF    DELAY 1/2 BIT TIME
 948  1E6D  AD 40 17  GET2   LDA   SAD       GET 8 BITS
 949  1E70  29 80            AND   #$80      MASK OFF LOW ORDER BITS
 950  1E72  46 FE            LSR   CHAR      SHIFT RIGHT CHAR
 951  1E74  05 FE            ORA   CHAR
 952  1E76  85 FE            STA   CHAR
 953  1E78  20 D4 1E         JSR   DELAY     DELAY 1 BIT TIME
 954  1E7B  CA               DEX
 955  1E7C  D0 EF            BNE   GET2      GET NEXT CHAR
 956  1E7E  20 EB 1E         JSR   DEHALF    EXIT THIS RTN
 957                ;
 958  1E81  A6 FD            LDX   TMPX
 959  1E83  A5 FE            LDA   CHAR
 960  1E85  2A               ROL   A         SHIFT OFF PARITY
 961  1E86  4A               LSR   A
 962  1E87  60        GET6   RTS
 963                ;
 964                ;       INITIALIZATION FOR SIGMA
 965                ;
 966  1E88  A2 01     INITS  LDX   #$01      SET KB MODE TO ADDR
 967  1E8A  86 FF            STX   MODE
 968                ;
 969  1E8C  A2 00     INIT1  LDX   #$00
 970  1E8E  8E 41 17         STX   PADD      FOR SIGMA USE SADD
 971  1E91  A2 3F            LDX   #$3F
 972  1E93  8E 43 17         STX   PBDD      FOR SIGMA USE SBDD
 973  1E96  A2 07            LDX   #$07      ENABLE DATA IN
 974  1E98  8E 42 17         STX   SBD       OUTPUT
 975  1E9B  D8               CLD
 976  1E9C  78               SEI
 977  1E9D  60               RTS
 978                ;
 979                ;       PRINT 1 CHAR   CHAR IN A
 980                ;       X IS PRESERVED  Y RETURNED = FF
 981                ;       OUTSP   PRINTS 1 SPACE
 982                ;
 983  1E9E  A9 20     OUTSP  LDA   #$20
 984  1EA0  85 FE     OUTCH  STA   CHAR
 985  1EA2  86 FD            STX   TMPX
 986  1EA4  20 D4 1E         JSR   DELAY     10/11 BIT CODE SYNC
 987  1EA7  AD 42 17         LDA   SBD       START BIT
 988  1EAA  29 FE            AND   #$FE
```

```
CARD # LOC     CODE          CARD
 989  1EAC  8D 42 17          STA   SBD
 990  1EAF  20 D4 1E          JSR   DELAY
 991  1EB2  A2 08             LDX   #$08
 992  1EB4  AD 42 17  OUT1    LDA   SBD       DATA BIT
 993  1EB7  29 FE             AND   #$FE
 994  1EB9  46 FE             LSR   CHAR
 995  1EBB  69 00             ADC   #$00
 996  1EBD  8D 42 17          STA   SBD
 997  1EC0  20 D4 1E          JSR   DELAY
 998  1EC3  CA                DEX
 999  1EC4  D0 EE             BNE   OUT1
1000  1EC6  AD 42 17          LDA   SBD       STOP BIT
1001  1EC9  09 01             ORA   #$01
1002  1ECB  8D 42 17          STA   SBD
1003  1ECE  20 D4 1E          JSR   DELAY     STOP BIT
1004  1ED1  A6 FD             LDX   TMPX      RESTORE INDEX
1005  1ED3  60                RTS
1006                  ;
1007                  ;       DELAY 1 BIT TIME
1008                  ;       AS DETERMEND BY DETCPS
1009                  ;
1010  1ED4  AD F3 17  DELAY   LDA   CNTH30    THIS LOOP SIMULATES
1011  1ED7  8D F4 17          STA   TIMH      DETCPS SECTION AND WILL DELAY
1012  1EDA  AD F2 17          LDA   CNTL30    1 BIT TIME
1013  1EDD  38        DE2     SEC
1014  1EDE  E9 01     DE4     SBC   #$01
1015  1EE0  B0 03             BCS   DE3
1016  1EE2  CE F4 17          DEC   TIMH
1017  1EE5  AC F4 17  DE3     LDY   TIMH
1018  1EE8  10 F3             BPL   DE2
1019  1EEA  60                RTS
1020                  ;
1021                  ;                       DELAY 1/2 BIT TIME
1022  1EEB  AD F3 17  DEHALF  LDA   CNTH30    DOUBLE RIGHT SHIFT OF DELAY
1023  1EEE  8D F4 17          STA   TIMH      CONSTANT FOR A DIV BY 2
1024  1EF1  AD F2 17          LDA   CNTL30
1025  1EF4  4A                LSR   A
1026  1EF5  4E F4 17          LSR   TIMH
1027  1EF8  90 E3             BCC   DE2
1028  1EFA  09 80             ORA   #$80
1029  1EFC  B0 E0             BCS   DE4
1030                  ;
1031                  ;       SUB TO DETERMINE IF KEY IS
1032                  ;       DEPRESSED OR CONDITION OF SSW
1033                  ;             KEY NOT DEP OR TTY MODE    A=0
1034                  ;             KEY DEP OR KB MODE      A NOT ZERO
1035                  ;
1036                  ;
1037  1EFE  A0 03     AK      LDY   #$03       3 ROWS
1038  1F00  A2 01             LDX   #$01      DIGIT 0
1039                  ;
1040  1F02  A9 FF     ONEKEY  LDA   #$FF
1041  1F04  8E 42 17  AK1     STX   SBD        OUTPUT DIGIT
1042  1F07  E8                INX              GET NEXT DIGIT
1043  1F08  E8                INX
1044  1F09  2D 40 17          AND   SAD       INPUT SEGMENTS
1045  1F0C  88                DEY
1046  1F0D  D0 F5             BNE   AK1
```

```
CARD # LOC     CODE          CARD
1047                  ;
1048  1F0F  A0 07             LDY   #$07
1049  1F11  8C 42 17          STY   SBD
1050                  ;
1051  1F14  09 80             ORA   #$80
1052  1F16  49 FF             EOR   #$FF
1053  1F18  60                RTS
1054                  ;
1055                  ;       SUB  OUTPUT TO 7-SEGMENT DISPLAY
1056                  ;
1057  1F19  A0 00     SCAND    LDY   #$00      GET DATA SPECIFIED
1058  1F1B  B1 FA              LDA   (POINTL),Y BY POINT
1059  1F1D  85 F9              STA   INH       SET UP DISPLAY BUFFER
1060  1F1F  A9 7F     SCANDS   LDA   #$7F      CHANGE SEG
1061  1F21  8D 41 17           STA   PADD      TO OUTPUT
1062                  ;
1063  1F24  A2 09              LDX   #$09      INIT DIGIT NUMBER
1064  1F26  A0 03              LDY   #$03      OUTPUT 3 BYTES
1065                  ;
1066  1F28  B9 F8 00  SCAND1   LDA   INL,Y     GET BYTE
1067  1F2B  4A                 LSR   A         GET MSD
1068  1F2C  4A                 LSR   A
1069  1F2D  4A                 LSR   A
1070  1F2E  4A                 LSR   A
1071  1F2F  20 48 1F           JSR   CONVD     OUTPUT CHAR
1072  1F32  B9 F8 00           LDA   INL,Y     GET BYTE AGAIN
1073  1F35  29 0F              AND   #$0F      GET LSD
1074  1F37  20 48 1F           JSR   CONVD     OUTPUT CHAR
1075  1F3A  88                 DEY             SET UP FOR NXT BYTE
1076  1F3B  D0 EB              BNE   SCAND1
1077  1F3D  8E 42 17           STX   SBD       ALL DIGITS OFF
1078  1F40  A9 00              LDA   #$00      CHANGE SEGMENT
1079  1F42  8D 41 17           STA   PADD      TO INPUTS
1080  1F45  4C FE 1E           JMP   AK        GET ANY KEY
1081                  ;
1082                  ;       CONVERT AND DISPLAY HEX
1083                  ;       USED BY SCAND ONLY
1084                  ;
1085  1F48  84 FC     CONVD    STY   TEMP      SAVE Y
1086  1F4A  A8                 TAY             USE CHAR AS INDEX
1087  1F4B  B9 E7 1F           LDA   TABLE,Y   LOOKUP CONVERSION
1088  1F4E  A0 00              LDY   #$00      TURN OFF SEGMENTS
1089  1F50  8C 40 17           STY   SAD
1090  1F53  8E 42 17           STX   SBD       OUTPUT DIGIT ENABLE
1091  1F56  8D 40 17           STA   SAD       OUT PUT SEGMENTS
1092                  ;
1093  1F59  A0 7F              LDY   #$7F      DELAY 500 CYCLES APPROX.
1094  1F5B  88        CONVD1   DEY
1095  1F5C  D0 FD              BNE   CONVD1
1096                  ;
1097  1F5E  E8                 INX             GET NEXT DIGIT NUMBER
1098  1F5F  E8                 INX             ADD 2
1099  1F60  A4 FC              LDY   TEMP      RESTORE Y
1100  1F62  60                 RTS
1101                  ;
1102                  ;       SUB TO INCREMENT POINT
1103                  ;
1104  1F63  E6 FA     INCPT    INC   POINTL
```

```
CARD # LOC     CODE          CARD
1105  1F65  D0 02            BNE   INCPT2
1106  1F67  E6 FB            INC   POINTH
1107  1F69  60       INCPT2  RTS
1108                ;
1109                ;       GET KEY FROM KEY BOARD
1110                ;       RETURN WITH A=KEY VALUE
1111                ;       GT. 15 THEN ILLEGAL OR NO KEY
1112                ;
1113                ;
1114  1F6A  A2 21     GETKEY  LDX   #$21      START AT DIGIT 0
1115  1F6C  A0 01     GETKE5  LDY   #$01      GET 1 ROW
1116  1F6E  20 02 1F          JSR   ONEKEY
1117  1F71  D0 07             BNE   KEYIN     A=0 NO KEY
1118  1F73  E0 27             CPX   #$27      TEST FOR DIGIT 2
1119  1F75  D0 F5             BNE   GETKE5
1120  1F77  A9 15             LDA   #$15      15=NOKEY
1121  1F79  60                RTS
1122  1F7A  A0 FF     KEYIN   LDY   #$FF
1123  1F7C  0A        KEYIN1  ASL   A         SHIFT LEFT
1124  1F7D  B0 03             BCS   KEYIN2    UNTIL Y=KEY NUM
1125  1F7F  C8                INY
1126  1F80  10 FA             BPL   KEYIN1
1127  1F82  8A        KEYIN2  TXA
1128  1F83  29 0F             AND   #$0F      MASK MSD
1129  1F85  4A                LSR   A         DIVIDE BY 2
1130  1F86  AA                TAX
1131  1F87  98                TYA
1132  1F88  10 03             BPL   KEYIN4
1133  1F8A  18        KEYIN3  CLC
1134  1F8B  69 07             ADC   #$07      MULT (X-1) TIMES A
1135  1F8D  CA        KEYIN4  DEX
1136  1F8E  D0 FA             BNE   KEYIN3
1137  1F90  60                RTS
1138                ;
1139                ;       SUB TO COMPUTE CHECK SUM
1140                ;
1141  1F91  18        CHK     CLC
1142  1F92  65 F7             ADC   CHKSUM
1143  1F94  85 F7             STA   CHKSUM
1144  1F96  A5 F6             LDA   CHKHI
1145  1F98  69 00             ADC   #$00
1146  1F9A  85 F6             STA   CHKHI
1147  1F9C  60                RTS
1148                ;
1149                ;       GET 2 HEX CHAR'S AND PACK
1150                ;       INTO INL AND INH
1151                ;       X PRESERVED  Y RETURNED = 0
1152                ;       NON HEX CHAR WILL BE LOADED AS NEAREST HEX EQU
1153                ;
1154  1F9D  20 5A 1E  GETBYT  JSR   GETCH
1155  1FA0  20 AC 1F          JSR   PACK
1156  1FA3  20 5A 1E          JSR   GETCH
1157  1FA6  20 AC 1F          JSR   PACK
1158  1FA9  A5 F8             LDA   INL
1159  1FAB  60                RTS
1160                ;
1161                ;       SHIFT CHAR IN A INTO
1162                ;       INL AND INH
```

```
CARD # LOC     CODE          CARD
1163                  ;
1164  1FAC  C9 30     PACK    CMP   #$30      CHECK FOR HEX
1165  1FAE  30 1B             BMI   UPDAT2
1166  1FB0  C9 47             CMP   #$47      NOT HEX EXIT
1167  1FB2  10 17             BPL   UPDAT2
1168  1FB4  C9 40             CMP   #$40      CONVERT TO HEX
1169  1FB6  30 03             BMI   UPDATE
1170  1FB8  18                CLC
1171  1FB9  69 09             ADC   #$09
1172  1FBB  2A        UPDATE  ROL   A
1173  1FBC  2A                ROL   A
1174  1FBD  2A                ROL   A
1175  1FBE  2A                ROL   A
1176  1FBF  A0 04             LDY   #$04      SHIFT INTO I/O BUFFER
1177  1FC1  2A        UPDAT1  ROL   A
1178  1FC2  26 F8             ROL   INL
1179  1FC4  26 F9             ROL   INH
1180  1FC6  88                DEY
1181  1FC7  D0 F8             BNE   UPDAT1
1182  1FC9  A9 00             LDA   #$00      A=0 IF HEX NUM
1183  1FCB  60        UPDAT2  RTS
1184                  ;
1185  1FCC  A5 F8     OPEN    LDA   INL       MOVE I/O BUFFER TO POINT
1186  1FCE  85 FA             STA   POINTL
1187  1FD0  A5 F9             LDA   INH       TRANSFER INH- POINTH
1188  1FD2  85 FB             STA   POINTH
1189  1FD4  60                RTS
1190                  ;
1191                  ;
1192                  ;       END OF SUBROUTINES
1194                  ;
1195                  ;       TABLES
1196                  ;
1197  1FD5  00        TOP     .BYTE $00,$00,$00,$00,$00,$00,$0A,$0D,'MIK'
1197  1FD6  00
1197  1FD7  00
1197  1FD8  00
1197  1FD9  00
1197  1FDA  00
1197  1FDB  0A
1197  1FDC  0D
1197  1FDD  4D 49 4B
1198  1FE0  20                .BYTE ' ',$13,'RRE',' ',$13
1198  1FE1  13
1198  1FE2  52  52  45
1198  1FE5  20
1198  1FE6  13
```

```
CARD # LOC     CODE          CARD
1199                  ;
1200                  ;             TABLE HEX TO 7 SEGMENT
1201                  ;             0   1   2   3   4   5   6   7
1202  1FE7  BF        TABLE   .BYTE $BF,$86,$DB,$CF,$E6,$ED,$FD,$87
1202  1FE8  86
1202  1FE9  DB
1202  1FEA  CF
1202  1FEB  E6
1202  1FEC  ED
1202  1FED  FD
1202  1FEE  87
1203                  ;              8   9   A   B   C   D   E   F
1204  1FEF  FF                .BYTE $FF,$EF,$F7,$FC,$B9,$DE,$F9,$F1
1204  1FF0  EF
1204  1FF1  F7
1204  1FF2  FC
1204  1FF2  B9
1204  1FF4  DE
1204  1FF5  F9
1204  1FF6  F1
1206                  ;
1207                  ;
1208                  ;
1209                  ;
1210                  ;       INTERRUPT VECTORS
1211                  ;
1212  1FF7                    *=$1FFA
1213  1FFA  1C 1C     NMIENT  .WORD NMIT
1214  1FFC  22 1C     RSTENT  .WORD RST
1215  1FFE  1F 1C     IRQENT  .WORD IRQT
1269                          .END
```

Warranty

(Applies to factory assembled circuit boards only)

Briel Computers hereby warrants each of its products, and all components therein contained, to be free from defects in materials and/or workmanship for a period of thirty (30) days from date of purchase. In the event of the occurrence of malfunction or other indication of failure attributable directly to faulty workmanship and/or material, then, upon return of the product to Briel Computers, at 5392 Cornell Blvd, North Ridgeville, Ohio 44039 (postage prepaid), Briel Computers will, at its option, repair or replace said products or components thereof, to whatever extent Briel Computers shall deem necessary, to restore said product to proper operating condition. All such repairs or replacements shall be rendered by Briel Computers, without charge to the customer. The responsibility for the failure of any Briel Computers product, or component thereof, which, at the discretion of Briel Computers, shall have resulted either directly or indirectly from accident, abuse, or misapplication of the product, shall be assumed by the customer, and the Briel Computers shall assume no liability as a consequence of such events under the terms of this warranty. While every effort, on the part of Briel Computers, is made to provide clear and accurate technical instruction on the use, implementation, and application of its products, Briel Computers shall assume no liability in events which may arise from the application of such technical instruction, nor shall Briel Computers be held liable for the quality, interconnection, or application of peripheral products, which may have been recommended by Briel Computers, but which have not been supplied as part of the product. This warranty contains and embodies the limits of responsibility of Briel Computers, with regard to its products, and no other liability is expressed, implied, or should be assumed by the purchaser, and in no event shall Briel Computers be held liable for the loss of time, effort, or transportation costs, nor for loss of potential profits or other consequential losses which might arise from the purchase, assembly, use, application, or subsequent sale of the products of Briel Computers, nor from any instructions and/or technical information thereto related.