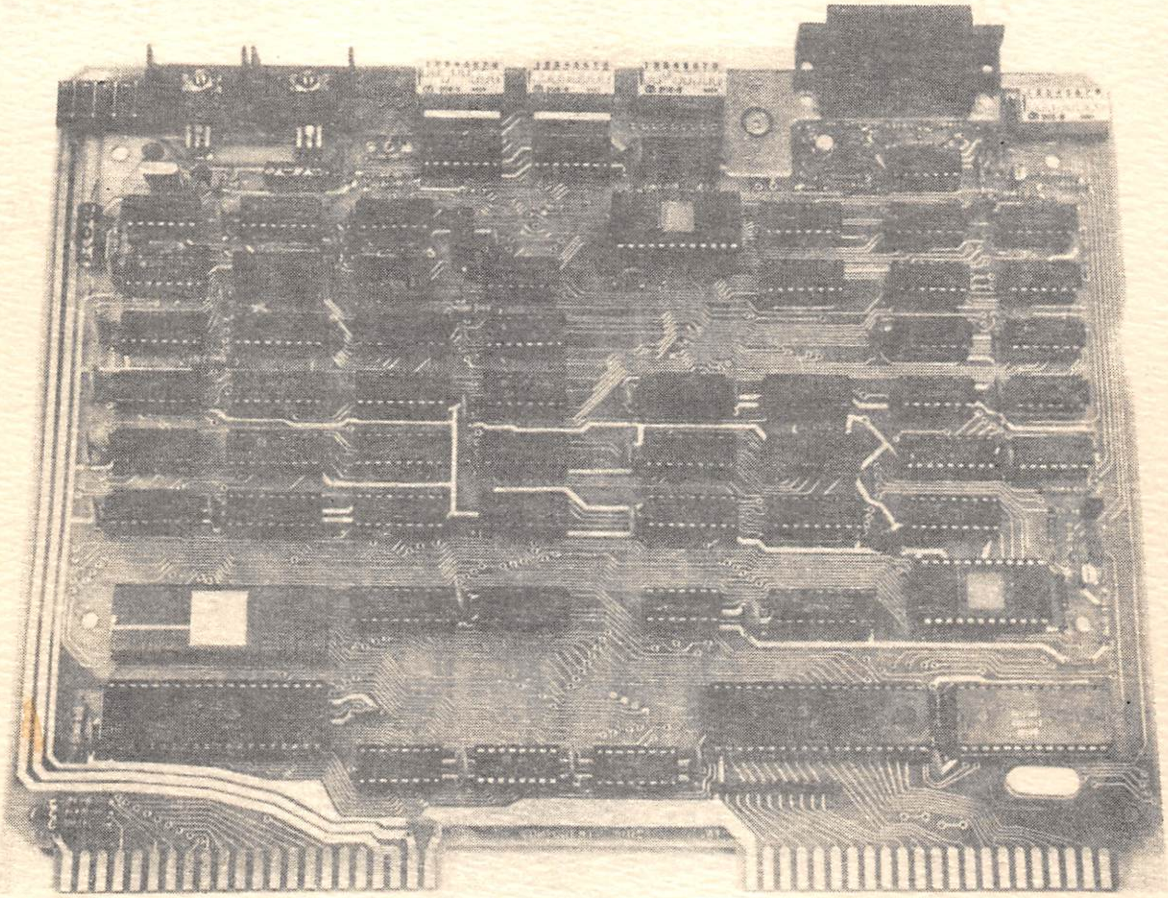
VIDEO PLUS II™



THE COMPUTERIST®

VIDEO PLUS II™

Versatile Video Expansion Board
for the

AIM ● SYM ● KIM

Programmable Screen Format
EPROM Character Generator
Programmable Character Generator
Up to 4K Display Memory
Includes 1K Program Memory
Supports ASCII Keyboard
Communications Option
Stand-Alone Option
AIM/SYM/KIM Software

The Computerist, Inc.
34 Chelmsford Street
Chelmsford, MA 01824

617/256-3649

rev. 1/81
809007

VIDEO PLUS II

Table of Contents

This is the first version of this manual. While every effort has been made to ensure its accuracy and completeness, we expect that the reader will find some mistakes and some areas for improvement. Please send us any suggestions, comments, or corrections. We will be continually updating and upgrading this information.

INTRODUCTION

VIDEO PLUS II™ has been designed specifically to work with the KIM-1, SYM-1 and AIM 65 microcomputers. It displays information stored in memory on the screen of a video monitor, or with the addition of an RF modulator, will display information on a TV screen. With the addition of a 6502 microprocessor, VIDEO PLUS II can be used as a stand-alone system or terminal without any other microcomputer board. VIDEO PLUS greatly expands the power of the basic AIM/SYM/KIM (collectively referred to as MICRO in this document) by providing the following facilities:

RAM Memory: Up to 4K bytes of Display RAM, up to 2K Programmable Character Generator RAM and 1K Program RAM for a total of 7K RAM [3K provided with basic board].

EPROM Memory: Includes a 2K EPROM with the ASK™ Video Plus "Instant" support software for the AIM/SYM/KIM. This allows these microcomputers to work directly with the Video Plus with no additional programming required. Supports BASIC, Editors, Assemblers, and most application packages.

Versatile Interface Adapter: Two 8-bit programmable I/O ports with additional handshaking lines. Two timers, a serial-to-parallel/parallel-to-serial shift register, may be used for handling a keyboard, light pen, communications, ...

CRT Controller: A specialized chip controls the display format, refreshes the display, services the light pen, provides the video timing signals, ...

Character Generator EPROM: This provides 128 permanently defined characters. It is supplied with a standard ASCII character set using a 7 x 9 dot matrix and lower case characters with true descenders. This may be reprogrammed by the user for any 128 character set with an up to 8 x 16 dot matrix. With a user supplied 2532 EPROM, up to 256 characters may be defined in an 8 x 16 dot matrix.

Special Video Features: These include programmable character width, reverse video, blank portions of display and flicker free display.

ASCII Keyboard Interface: Provision for easily attaching any standard ASCII keyboard. Other types of keyboards may also be supported - usually only requiring a change in the software.

Stand-Alone Capability: VIDEO PLUS is so complete that by adding a 6502, a 74LS161 and a reset switch, it can be used as a stand-alone system or terminal. This means that it is not dependent on nor limited to AIM/SYM/KIM systems and can be used with any computer system.

Communications: Directly supports RS232 or 20 mAmp communication similiar to that of the AIM/SYM/KIM systems. The ACIA option adds a more sophisticated Asynchronous Communications capability with programmable rates from 50 to 19.2K Baud, parity checks, modem disciplines, etc.

Other features of VIDEO PLUS II include:

Single +5 volt requirement: Since there are on-board +5 volt regulators, any voltage between +8 volts and +15 volts peak can be used, or +5 volts regulated may be supplied directly.

All IC's are Socketted: In the event of a chip failure, sockets make the problem easier to diagnose and to fix, often eliminating the need to send a unit back to service for minor problems.

MICRO Bus Compatible: The connections between the AIM/SYM/KIM and the VIDEO PLUS follow the same conventions used by the original KIM-4 mother board. The VIDEO PLUS may be interfaced via a cable or the MOTHER PLUS board.

Switchable Addressing: VIDEO PLUS may reside in any 8K segment of memory. This includes all of the RAM, ROM, VIA, ACIA, and CRT.

Works with American or European TV sets and/or Monitors.

Fully Assembled, Burned-In, and Tested.

Set Up and Check Out

Quick Set Up and Check Out

This assumes that you are using an Expansion Cable or MOTHER PLUS board to interface the VIDEO PLUS to your AIM, SYM, or KIM and have the ASK VIDEO PLUS Software in EPROM. Otherwise, please read the **Detailed Set Up and Check Out** section below.

1. If you are providing unregulated +8 volts, the power header must be positioned with no wire near the heat sinks. If you are providing regulated +5 volts, then the power header must be positioned with a wire near the heat sinks.
2. Plug your monitor into the RCA type connector at the upper right corner of the board.
3. Initialize the ASK Video Plus Software by running the VINIT program. See **ASK Start-Up** in the **ASK Video Plus Software** section. Normal output will now go to the video monitor.
4. Initialize the Keyboard service, if desired, by running the KINIT program. See **ASK Start-Up**. Keyboard input will now be accepted from your ASCII keyboard.
5. That's all there is to it. Now you can use your AIM, SYM or KIM easily with the video and keyboard service of the VIDEO PLUS. Read the remainder of the manual to find out how to use the other features of this board.

Detailed Set Up and Check Out

To insure proper operation and to avoid possible damage, please read EVERYTHING carefully. Start with the Introduction. Become familiar with all sections but do not memorize minute details. To get the VIDEO PLUS "up and running" with your MICRO (AIM/SYM/KIM), follow ALL steps, slowly and carefully!

1. All chips on VIDEO PLUS are prone to damage by static charge, especially the MOS (metal oxide semiconductor) circuitry - 6502, 6522, 6845, etc. It is advisable to unpack the VIDEO PLUS in an area that is not conducive to static buildup. It is always a good idea to discharge any static electricity before touching any computer boards. Usually touching a radiator or light switch will do the trick.
2. Unpack the VIDEO PLUS board from its individual box and remove the padding along with the anti-static bag. Examine the board for any visible damage which may have occurred during shipping. Push all ICs firmly into the sockets and straighten any capacitors which may have been bent.
3. If VIDEO PLUS is to be used with MOTHER PLUS or with the Expansion Cable, then proceed with the following steps. If not, refer first to the section on **Building a Connection Cable**.

   a) Set the RAM Address Selector Switch to the 4K position. The section on **RAM Memory** shows the switch positions.

   b) Plug your MICRO into the Computer slots on the MOTHER PLUS or attach the MICRO end of the Expansion Cable to the Expansion Connector of your MICRO.

   c) Plug VIDEO PLUS into any available pair of expansion slots or attach the other end of the Expansion Cable to the Expansion Connector of the VIDEO PLUS.
4. If your system has adequate +5 volts to run both your MICRO and VIDEO PLUS, then no additional power connections are required. The Power Header on the VIDEO PLUS board should be positioned so that there is a wire on the end toward the heat sinks and no wire on the end toward the edge of the board. If you are going to use unregulated +8 volts or a separate +5 volt regulated supply, then consult the section on **Power**.
5. Turn on the power. Using your MICRO's monitor, examine and modify a few RAM locations from each section which is implemented. With the RAM selector switch set at the 4K position, the following list indicates where the various system components will be located.

4000 - 47FF	Display RAM 2K
4800 - 4BFF	Reserved for additional Display RAM
4C00 - 4FFF	Program RAM 1K
5000 - 57FF	Programmable character generator - if implemented 1 or 2K
5800 & 5801	CRT controller Registers
5804	Option Control Switches
5808	Basic Control Switches
580C - 580F	ACIA if implemented
5810 - 581F	6522 VIA Registers
5820 - 5FFF	ASK Video Plus Software EPROM

   If unable to examine and modify RAM, check all steps through this point for errors. If no errors are found but the problem still exists, refer to the section on **VIDEO PLUS Testing and Field Repair** for guidance.
6. If the RAM test in step 5 is successful, then a more rigorous memory test can be run. Refer to the section **VIDEO PLUS Testing and Field Repair** for instructions on loading and running the memory test which was provided on the cassette tape.

7. The 6522 VIA (Versatile Interface Adapter) can be quickly tested as follows:

a) Reset your MICRO

b) If the following locations contain the contents listed for them, then the VIA is probably functioning correctly:

5810	FF
5811	FF
5812	00
5813	00
5814	?? (see note below)
5815	?? (see note below)

Locations 5814 and 5815 should cause the KIM display to flicker since a free running timer is being examined. These two addresses will show different values each time they are examined on an AIM or SYM, but will not flicker since the AIM and SYM only access each memory location once when examining memory.

If these results are obtained, then the VIA is most likely functioning properly. If there is any discrepancy, refer to **VIDEO PLUS Testing and Field Repair.**

8. The EPROM space can be tested by examining the locations of the ASK Video Plus Software. See the **ASK VIDEO PLUS Software** listing for information on what values should be in the various locations of the EPROM.

9. The CRT Controller may now be tested. Run a cable from the RCA jack on VIDEO PLUS to the video monitor being used. If a TV is being used, set the TV/Monitor mode dip switch in the TV position. The cable carrying the video should be coaxial with the shield connected to the RCA plug shield and the center wire connected to the RCA plug center pin. Now refer to the section **ASK VIDEO PLUS Software** for details on running the support software.

System Organization

Since the VIDEO PLUS board contains a number of separate elements which require addressing space, it is important that the user understand the organization of the board.

RAM Memory

There is provision on VIDEO PLUS for up to 7K bytes of RAM memory. This is 2114 type low power static RAM. Each 2114 chip contains 4096 bits of memory, and is organized as 1024 locations with 4 bits of information per location. A pair of 2114 chips in parallel produce 1024 (1K) 8 bit bytes. 2114 chips must always be added in pairs to provide the required 8 bit bytes.

4K of the total 7K bytes are used as Display RAM, the memory where display information is stored. This memory is located in the lower half of the 8K addressing space used by the VIDEO PLUS board. VIDEO PLUS is normally sold with 2K of the Display RAM in place. The user may add Display RAM as his application requires. The additional RAM may be added in 1K segments.

An additional 2K RAM may be used as Programmable Character Generator RAM (PGC). With this RAM, the user can develop his own special characters working with a dot matrix up to 8 x 16. This memory is also of the 2114 type. All of the decoding and addressing circuitry is present on the standard VIDEO PLUS board to implement the PCG features. All that is required is that the RAM be added to the PGC RAM sockets.

1K bytes of Program RAM is supplied. This may be addressed within the Video Plus 8K addressing space in place of the fourth 1K segment of Display RAM. Or, if the application requires the use of all 4K Display RAM, the extra 1K Program RAM can be addressed at any 8K [2K hexadecimal] boundary that is available within the total system. This extra RAM is provided to ensure that the Video Plus software can operate without using any of the normal microcomputer system RAM.

The address of all of the RAM is determined by the board Address Selection Switch. This 8 position DIP switch [SW 3] may be set to any one of the 8K starting addresses in the memory map of the MICRO: 0000, 2000, 4000, 6000, 8000, A000, C000, or E000. Note that the E000 address has implications for the interrupt vectors and should, in general, be avoided. Likewise, the 0000 address is normally reserved for the system and cannot generally be selected. In this manual and on the cassette tape provided, it is assumed that the VIDEO PLUS board has been selected at 4000. To set the switch for this, or any other, address, use the following table as a guide:

1 = 0000, 2 = 2000, 3 = 4000, 4 = 6000, 5 = 8000, 6 = A000, 7 = C000, 8 = E000

Display RAM

There is provision for up to 4K of Display RAM. The ASK Video Plus Software only assumes the 2K that the board normally comes with. The user can add 1 or 2K by simply inserting pairs of 2114 type RAM chips in the sockets already installed. The Display RAM (DSPRAM) occupies the lowest 4K addressing space of the board. If the VIDEO PLUS has been selected to start at 4000 (the default address for all discussions in this manual), then DSPRAM will be from 4000 to 47FF (the original 2K provided), from 4800 to 4BFF (the first 1K added by the user), and from 4C00 to 4FFF (the second 1K added by the user). The 4C00 to 4FFF address space is normally used by the 1K Program RAM and requires some minor jumper block changes if it is to be used for DSPRAM instead. The VIDEO PLUS board automatically refreshes the display based on the contents of the DSPRAM. The DSPRAM is simply additional memory to the microcomputer. Any location may be directly examined and/or modified by the user program. The CRT Controller chip determines what portion of the DSPRAM is to be displayed, based on parameters provided by the microcomputer. Within the portion of DSPRAM selected for presentation to the display, each 8 bit byte represents a character. The user may write his own programs to place information into the DSPRAM, or may use the ASK Video Software provided with the VIDEO PLUS. In any case, all that is required to produce a character on the screen is to place the code for that character into the proper location in the DSPRAM. No code conversion or dot pattern generation is required on the part of the user.

EPROM Character Generator

When a character code in the range 00 through 7F is selected by the CRT controller for displaying, it is used to "look up" a set of dot patterns contained in the Character Generator EPROM. The EPROM supplied with VIDEO PLUS is a 2716 or 2516. These EPROMs are organized as 8 bits by 2048 locations. Each character is defined by 16 contiguous addresses so that the character matrix is 8 dots wide by 16 dots high. The EPROM has space for 128 individual characters. The user can define any characters he requires, as long as they can fit into the 8 by 16 matrix. The character set normally supplied with the VIDEO PLUS board contains a set of characters which follow the standard ASCII conventions for the displayable characters. These include UPPER case and lower case alphabetics with true descenders, numerics, and standard punctuation. In addition, the standard VIDEO PLUS Character Generator EPROM contains a set of line graphics and other special characters. Refer to the section on **Standard EPROM Character Generator** for details.

More than 128 characters can be defined in the EPROM Character Generator by substituting a 2532 EPROM in place of the 2716/2516. The 2532 EPROM has 8 bits by 4096 locations. This permits all possible 256 characters to be defined. Any character from 00 through FF will now be defined by the EPROM. A jumper change is required on the board, and is described in the **Jumpers** section.

Program EPROM

Since software is required to operate VIDEO PLUS, and since the board should have minimal impact on other system memory, provision has been made for a 2K EPROM on-board. The EPROM is the INTEL 2716 or TI 2516 type. The only slightly unusual thing about its use on the VIDEO PLUS board is that 20 hex locations have been "borrowed" from the EPROM addressing space to make room for the VIA, CRT Controller, ACIA, and switches. The EPROM only has, therefore, 2016 usable locations instead of the normal 2048. The EPROM addresses start at 5820 normally and run up to 5FFF. The EPROM was given the highest addresses on the board so that the "stand-alone" option would have the interrupt vectors in the EPROM up in the highest memory where they belong. If your system contains a MEMORY PLUS or DRAM PLUS board, then you have everything you need to program your own EPROMs. If not, it would be quite easy to build an EPROM Programmer based on the VIDEO PLUS VIA chip. The **ASK Video Plus Software** is supplied on EPROM which can run in any memory space on an AIM, SYM, or KIM. See the **ASK Video Plus Software** section for details.

CRT Controller - 6845

The CRT Controller performs a number of functions which are crucial in getting data from RAM memory onto the video monitor or TV. It requires two addresses. The first address is a "register select" and the second address accesses the "register" selected by the first address. This may be a bit confusing at first, but information and details can be found in the 6845 Data Sheet provided as an appendix to this manual. The locations normally assigned are: **5800** for the CRT Register and **5801** for the CRT Data.

Versatile Interface Adapter - 6522

This is a general purpose I/O and timer device which is used for a number of purposes on the VIDEO PLUS. Its main purpose is to control the dot width for the display characters. It also provides an interface for an ASCII Keyboard, is used by the Light Pen interface, and can be used as an RS 232 or 20 mA Current Loop communication channel. The timers may be used for a variety of functions. The VIA occupies the addressing space 5810 to 581F. See the 6522 Data Sheet provided in your AIM or SYM Users manual for details. Individual data sheets may be available from your 6502 sales representative.

Programmable Character Generator RAM:

There is provision in the Programmable Character Generator (PCG) section of the board for up to 2K RAM. This may be used for the Programmable Character Generator or, with minimal display interference, as general system RAM. The addition of 1K can provide the user with 64 Programmable Characters, the second 1K can provide an additional 64 characters for a total of 128 characters. The address of the PCG RAM is, assuming the board is addressed at 4000:

5000 to 53FF (the first 1K added) and
5400 to 57FF (the second 1K added).

The PCG is composed of two basic parts. The first is the decoding and latching logic which determines if a character is being generated by the PCG or the EPROM. A code in the range 00 to 7F is always handled by the EPROM Character Generator and does not involve the PCG. A code from 80 to FF may be handled by the EPROM Character Generator but is normally handled by the PCG . Bit 80 is used to " switch" between the two display character generation modes. The displayable matrix supported by the CRT Controller is 8 dots wide by 16 rows deep, just as in the EPROM. The display pattern for the PCG is contained in the PCG RAM memory. The organization of the PCG RAM is set up so that the first sixteen (16) bytes of the PCG RAM contain the 16 rows for the display pattern for character number 80, the next 16 bytes for character 81, and so on.

The first byte of the 16 represents the dot pattern for the top row of the character, the second byte represents the second row, and so on down to the last row. The highest bit (MSB) in each byte is the left-most dot, and so on down to the lowest bit (LSB) in each byte which is the right-most dot. For example, character 80 would have the following memory contents to display the dot pattern shown: (assume that the PCG RAM starts at 5000)

Memory Address	Contents Hex	Contents Binary	Display Pattern
5000	C3	11000011	`**    **`
5001	C3	11000011	`**    **`
5002	66	01100110	` **  ** `
5003	3C	00111100	`  ****  `
5004	24	00100100	`  *  *  `
5005	24	00100100	`  *  *  `
5006	3C	00111100	`  ****  `
5007	66	01100110	` **  ** `
5008	C3	11000011	`**    **`
5009	C3	11000011	`**    **`
500A	7E	01111110	` ****** `
500B	7E	01111110	` ****** `
500C	42	01000010	` *    * `
500D	42	01000010	` *    * `
500E	7E	01111110	` ****** `
500F	FF	11111111	`********`

A few ideas to ponder. The VIDEO PLUS defines the number of dots per character, both width and heigth. The PCG and EPROM can have characters that are contiguous. This means that it is possible to define a good graphics set with contiguous lines, to use several individual characters together to form a "super" character, and so forth. While all 8 dots across and 16 dots down are defined for each character, the number of rows and columns being displayed at any time is determined by the VIDEO PLUS board. The number of rows and columns being displayed may be easily changed under program control. An 8 by 8 array might be useful for graphics, providing smooth diagonals; a 5 x 7 character set with a 6 x 8 display space would provide a high density display; the normal 7 x 12 (including descenders) character set on an 8 x 13 display space provides reasonable character separation for general usage; the same 7 x 12 characters on a 10 x 14 display space would provide enhanced character separation for word processing; and so forth. With the character definition in RAM instead of EPROM, characters can be changed under program control. For example, it is possible to have a program invert some or all of the PCG characters. Or underline them. Or even turn them "upside-down"! The possibilities are almost unlimited. More information, ideas and programs will be available as application notes (free) and application packages (for sale).

System Address Summary

On-Board Address	'Normal' 4000 Address	Function
0000 — 07FF	4000 — 47FF	Display RAM — First 2K Provided with Video Plus
0800 — 0BFF	4800 — 4BFF	Display RAM — 1K Optional
0C00 — 0FFF	4C00 — 4FFF	Program RAM — 1K Provided with Video Plus, or Display RAM 1K Optional
1000 — 17FF	5000 — 57FF	Programmable Character Generator RAM — 2K Optional
1800 — 1801	5800 — 5801	CRT Controller Registers
1804	5804	Option Switches — SW 2
1808	5808	Basic Video Plus Switches — SW 1
180C — 180F	580C — 580F	ACIA 6551 Option Registers
1810 — 181F	5810 — 581F	VIA 6522 Registers
1820 — 18FF	5820 — 58FF	EPROM Memory — ASK Video Plus Software

Note: The few missing addresses in the above table contain duplicate information due to partial address decoding, and can be ignored.

Power

VIDEO PLUS requires only +5 volts at about 1.2 amps. There are two ways in which this power requirement may be met.

Regulated +5 volt supply. If the VIDEO PLUS is to be powered by regulated +5 volts, then the supply should be connected to pins E-21 and E-Y on the VIDEO PLUS Expansion connector. The supply should be capable of supplying at least 1.2 amps in addition to any other board it is driving, such as a KIM-1, SYM-1 or an AIM 65. The Power Header at the top of the board should be positioned so that the end nearest the heat sink has a wire connected.

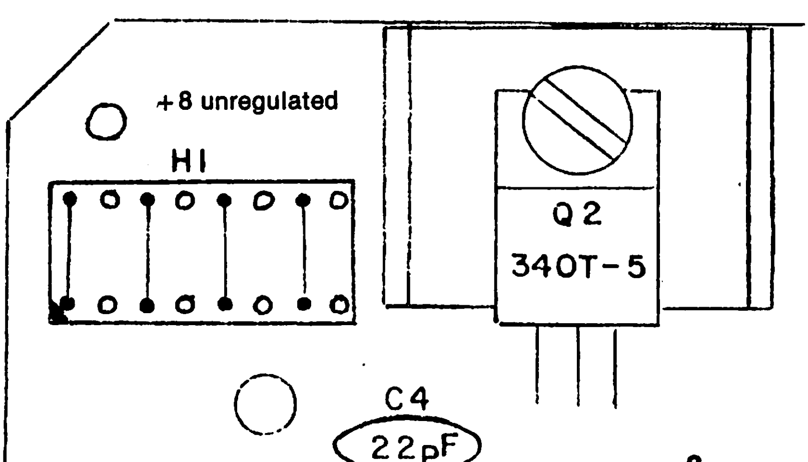
Regulated or unregulated supply between +8 and +15 volts. If a supply is to be regulated by the VIDEO PLUS regulators, the positive lead should be connected to pins E-19 and/or E-20, and the negative or ground lead should be connected to pin E-22 or any available ground. These pins are found on the VIDEO PLUS Expansion connector. The supply must be capable of supplying at least 1.5 amps.

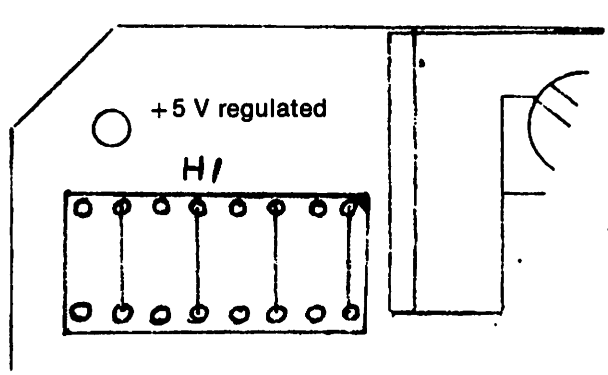
The two on-board regulators will regulate any voltage between +8 and +15 volts down to the required +5 volts. The Power Header at the top of the board should be positioned so that the end nearest heat sinks does not have a wire connected.

If a supply voltage greater than +10 volts is to be regulated, (depending on the physical configuration of the system) the heat dissipated by the regulators should be monitored. If they generate an excessive amount of heat, then the system should be exposed to more air or a lower supply voltage should be used. The two regulators are designed to provide the power required for a fully loaded VIDEO PLUS board — 7K RAM, 2K EPROM, and the optional 6502, but may not have enough "extra" power for use with other devices. In particular, the on-board regulators should not be used to power a keyboard, since many ASCII keyboards draw up to 1.0 amps (or more). Separate power should be provided for these.

Both sets of power connections may be applied to the board at the same time without any harm. The Power Header determines which of the two methods, +5 volt regulated or +8 volt unregulated, is being used at any time. There is no need to remove either set of connections from the VIDEO PLUS Expansion connector when using the other set.

The ACIA option may require additional voltages. Please refer to the **ACIA Option** section for details.





Switches and Jumpers

VIDEO PLUS has a variety of features and functions. Many of these are selectable through the use of the DIP switches, jumpers located on the board, or a combination of the two. This section summarizes the switch and jumper functions. Refer to specific sections for details on the operation of the features.

Switches

Switch 1 [SW 1]

Switch 1 contains all of the basic option switches. Each of the eight independent switches (except switch 4) is pulled high in the **OFF** position and either connected to ground or to some signal in the **ON** position. The table below shows the function of each switch position. Some of these switches may be read by the microcomputer at the address 1808 above the base address of the VIDEO PLUS board. For the standard 4000 base address referenced in this manual, this would be location 5808.

Switch Number	1808 Value	OFF Position Function	ON Position Function
1	01	TV Low Bandwidth Video Output	Monitor High Bandwidth Video Output
2		Negative Keyboard Strobe	Positive Keyboard Strobe
3	04	Negative Keyboard Data	Positive Keyboard Data
4		Normal Video Output on Character Bit 80	Invert Video Output on Character Bit 80
5		Always get Character from EPROM	Get Character from PCG RAM if Bit 80
6	20	Undefined	Undefined
7		Disable Off-board Program RAM	Enable Off-board Program RAM
8	80	Always Permit Refresh	Only Refresh During Vertical Retrace

Switch 2 [SW 2]

This switch does not have any specific system connections. Its eight positions each control one data bit in a location which may be read by the microcomputer. The location is at 1804 relative to the base address of the Video Plus board. For the standard 4000 address assumed in this manual, this would be address 5804. In the **OFF** position, each switch appears as a **1** in the data byte. In the **ON** position, each switch appears as a **0**. Position **1** is mapped to data bit **0** or hex value **01**; position **2** is mapped to data bit **1** or hex value **02**; and so on to position **8** which is mapped to data bit **7** or hex value **80**. Some of these switches will be used with the ACIA option and other application software which will be developed.

Switch 3 [SW 3]

This switch is used to select the base address for the Video Plus board. Only one position may be **ON** at any time. Each switch corresponds to an 8K boundary (2K hex). Position **1** enables the board for **0000** to **1FFF**; position **2** enables the board for **2000** to **3FFF**; up to position **8** which enables the board for **E000** to **FFFF**. For discussion in this manual, it is assumed that the board is enabled at **4000** which would be switch position **3** in the **ON** position with all other switches **OFF**.

Switch 4 [SW 4]

This switch is used with the communication options. Please refer to the **ACIA Option** section for details. This switch is not included or required on the basic Video Plus board.

Switch 5 [SW 5]

This is a momentary SPDT switch used as a RESET switch. Refer to the **Stand-Alone Option** section for details. This switch is not included or required on the basic Video Plus board.

Jumpers

The Video Plus Switches provide the user an easy way to select various commonly used options. There are a number of other options and system configuration choices which are not brought to switches. Some of these options are brought to jumper pins which may be interconnected with a small plastic jumper block, a jumper wire, a wire-wrap wire, a soldered wire, etc. Other options require an etched trace to be cut and/or a wire to be soldered to the board. All of the etch jumpers have pads associated with them so an etch that is cut may be reconnected with a soldered wire. The table below lists the function of these jumper options. Refer to the **Component Layout** and the **Schematic** for position and detailed information.

Jumper	Function
W1	Select 8K address of Program RAM. If the extra 1K Program RAM is to be addressed outside of the normal Video Plus board range, then connnect a jumper wire from the pin labelled W1 to the pin directly below Switch 3 which corresponds to the 8K address segment desired. W10 must be removed and Switch 1 Position 7 set to the ON position to enable this memory.
W2	Override the Control Code Supression logic so that the VIA CB2 line may be used for some other purpose. This is normally not connected. Cut the W3 etch and solder a wire to the W2 pads.
W3	Enable the Control Code Supression logic. This is the normal state and is etched on the board. It must be cut if **W2** is to be connected.
W4	This etched jumper connects VSYNC (vertical sync) to the composite video output circuitry. If VSYNC is not used in the composite video for a particular monitor, then this etch may be cut.
W5	This etched jumper connects HSYNC (horizontal sync) to the composite video output circuitry. If HSYNC is not used in the composite video for a particular monitor, then this etch may be cut.
W6	This etched jumper connects the combined HSYNC and VSYNC to the composite video output circuitry. If these are not used in the composite video for a particular monitor, then this etch may be cut.
W7	This etched jumper connects the Receive Data pin of the ACIA to the Application connector pin M.
W8	This etched jumper connects the Transmit Data pin of the ACIA to the Application connector pin K.
W9	This etched jumper connects the RS 232 connector pin 1 to ground. This may be removed for systems which require some other signal or level on pin 1.
W10,W11	This jumper block determines whether the the 1C00 relative address will be used for the 1K Program RAM or the highest 1K of the Display RAM. In the W10 position, the Program RAM is addressed. This is standard for the ASK Video Plus Software. In the W11 position, the Display RAM is addressed.
W12-W15	These etched jumpers connect the Row Selects from the CRT controller to the EPROM Character generator to provide row selection for each character. See the application note on **Bit Mapped Video** for details on how to use these jumpers.
W16,W17	A jumper block is provided which enables either the 2716 EPROM Character Generator or a 2532 EPROM. The jumper block should be installed in position W16 for the 2716 and in position W17 for the 2532, where it will cause the high half of the EPROM to be accessed whenever Bit 80 is on in a character.
W18,W19	These were reserved for a feature which would have allowed characters **FE** and **FF** to be used to control blanking instead of **1E** and **1F** which are normally used. This feature was not implemented. W18 is etched and should not be touched.
W20	This etched jumper enables the blanking circuitry. If the blanking controlled by **1E** and **1F** is not desired, then this etch may be cut and a resistor **R47** installed to override the blanking.
W21,W22	This jumper block determines whether the CRT controller can access 2K or 4K of Display RAM. In the typical system, this will be set in the W21 position, enabling all 4K. If for any reason only 2K of Display RAM is to be addressed, then the jumper block should be moved to the W22 position. Hardware scrolling is an example of a situation in which a smaller amount of Display RAM would need to be addressed.

Using Video Plus Features

In addition to providing the basic video display functions, VIDEO PLUS II offers many other capabilities. This section will show how to utilize the main features.

Blanking Portions of the Display

A portion of the display may be blanked simply by writing a Start Blank ['1E' hex or ASCII 'RS' character] immediately preceding the first position to be blanked. This will cause blanking of all characters until the end of the line is reached. To end the blanking before the end of the line, simply write an End Blank ['1F' hex or ASCII 'US' character] immediately following the last position to be blanked. This will cause the remaining characters to be displayed until another Start Blank character is encountered. Any number of Start Blanks and End Blanks may be used. They will appear as spaces. If this feature is not desired, it may be eliminated by cutting the jumper etch **W20** and installing resistor **R47**. See **Jumpers** for details.

Flicker Supression

Flicker, normally a brief white flash, may occur when the microcomputer and the CRT Controller both try to access the Display RAM at the same time. Occasional flicker is acceptable for many applications, and may not even be noticeable. Flicker may not be acceptable in some applications, and provisions have been made to supress it. The ASK Video Plus Software tests a bit in memory controlled by Position 8 of Switch Number 1. With the switch in the **OFF** position, updating will always occur and flicker will sometimes occur. With the switch in the **ON** position, updating will occur only during vertical retrace and no flicker will occur. Since the flicker supression only permits updating during the vertical retrace, it may slow down the system. Certain operations such as scrolling will be noticeably slower when flicker supression is on. The switch permits the operator to determine whether or not to enable the flicker supression.

Programmable Character Generator Select/Disable

Characters may be generated by the Programmable Character Generator [PCG] or the EPROM Character Generator. Characters in the range 00 through 7F always use the EPROM. Characters in the range 80 through FF may use the EPROM or the PCG as a function of Position 5 of Switch 1. In the **OFF** position only the EPROM will be used, even for characters with Bit 80 on. In the **ON** position characters with Bit 80 on will be generated by the PCG.

Inverse Video

Characters are normally displayed as light dots on a dark background. Characters with Bit 80 on, that is those in the range 80 through FF, may be displayed with Inverted Video. If Position 4 of Switch 1 is **ON**, then all characters with Bit 80 on will appear inverted - the dots will be dark and the background will be light. If this switch is used in conjunction with the PCG switch described above, then the normal EPROM characters may be inverted as well. An ASCII 'A' has a hex value of '41'. If Bit 80 is turned on for this character in the Display RAM, it becomes a 'C1'. With switch 4 **ON** and 5 **OFF**, a '41' in the Display RAM would appear on the video as a capital 'A' - light on dark, and a 'C1' would appear as a capital 'A' - dark on light. This permits the standard EPROM Character Generator to be used to generate two entirely separate sets of display characters, one in normal and one in inverted video.

Keyboard Configuration Control

ASCII keyboards come with varying interface parameters. Some generate a positive strobe and some a negative one. Some have a data bit positive and some a data bit negative. To permit the varying keyboards to simply interface to Video Plus, provision has been made to allow switch selection of these two important parameters. Position 2 of Switch 1 will enable a Positive keyboard strobe in the **ON** position and a Negative keyboard strobe in the **OFF** position. Position 3 of Switch 1 is tested by the ASK Video Plus Software and will treat the keyboard data as Positive in the **ON** position and Negative in the **OFF** position. The keyboard strobe selection is handled purely in hardware. The keyboard data requires that the software invert the data when the switch is in the **ON** position.

Selecting Monitor or TV Mode

A switch is provided to permit selection between the high bandwidth Monitor mode and the low bandwidth TV mode. Position 1 of Switch 1 will enable the Monitor mode when **ON** and the TV mode when **OFF**. No other modifications are necessary in the hardware. The software must make changes to the CRT initialization parameters. It may test Bit 01 of the Basic Switch Address to determine which parameters to use. The ASK Video Plus Software does this testing automatically.

Video Interfacing

There are several ways in which VIDEO PLUS may be interfaced to a Video Monitor (VM) or a Television (TV), and both the American and European protocols are supported.

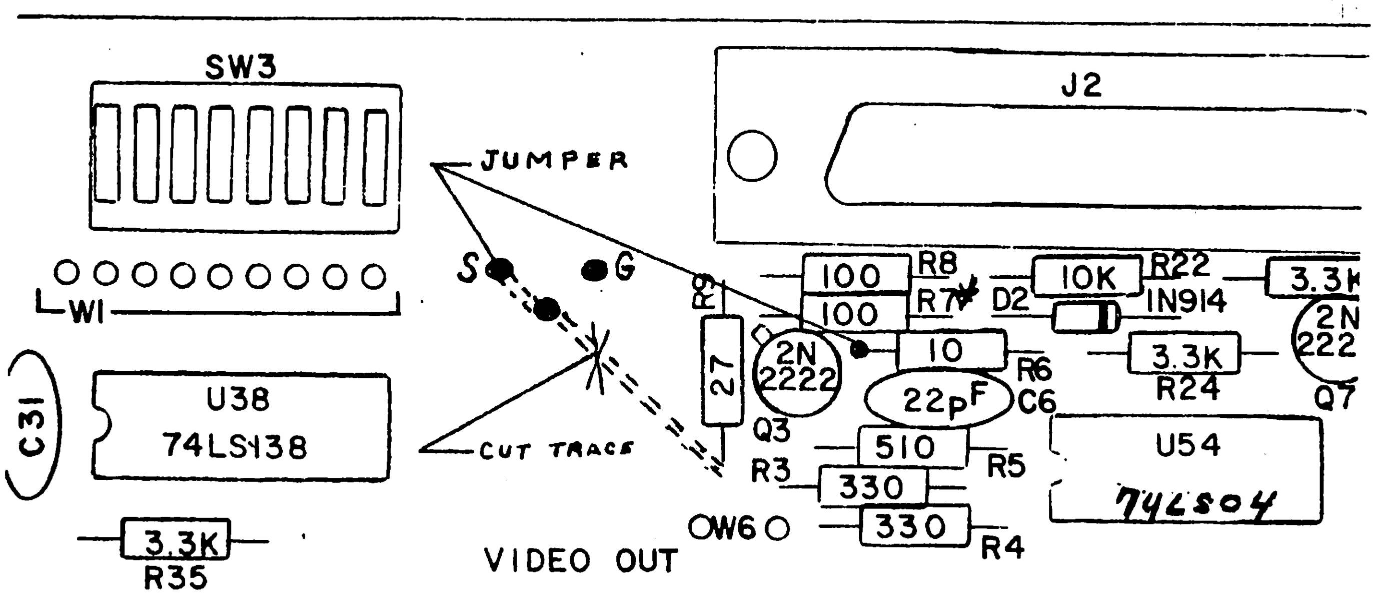
1. American Video Monitor. Connect a coaxial cable with an RCA type of jack to the connector provided on the board, or connect a cable directly to the two pads located near the phono jack, or to the Application connector: A-B for the signal and A-C for the ground. The Video output will be a standard EIA type signal: +1.0 volt for white, +0.25 volts for black, 0.0 volts for sync.

2. American Television. A commercially available RF modulator may be connected to the video output jack and the output of the RF connector fed into a standard TV set. Since the standard TV does not have the high band-width of a quality monitor, provisions have been made to change the output band-width of VIDEO PLUS to match the requirements of the TV. To do this, set the TV/Monitor switch to the TV position. See **Switches** for details. This will change the basic clock from 16 MHz to 8 MHz, effectively halving the band-width. A TV will only be able to display about 40 characters. The ASK VIDEO PLUS Software tests the position of the TV/Monitor switch and adjusts the initialization parameters automatically. The address 5808 (assuming 4000 as the base address) will have bit 01 low in the Monitor mode and bit 01 high in the TV mode. Any program can test this address and set the appropriate parameters for the CRT initialization, just as the ASK program does. This means that an EPROM program can be self-adapting to the Monitor/TV modes.

3. European Video Monitor. A separate output is provided for the European protocol. A wire must be run from the connecting point of Q3, R6 and C6 to the 'S' pad near the video jack. The existing connection from R9 to the video output should be cut. The signal is now available at the video output jack and at the Application connector at A-B with the ground at A-C. Refer to the diagram below to locate wiring changes.

4. European Television. The instructions and comments listed for American TV apply to converting from European monitors to European TV.

General Information. The CRT Controller chip performs most of the "VIDEO" functions of the VIDEO PLUS board. Its operation is not necessarily easy to understand, but the complete data sheet has been provided. Some experimentation may be required to determine the best set of parameters for any particular equipment and any set of requirements. An alternate initialization entry point in the ASK VIDEO PLUS Software supports this experimentation.



Application, Keyboard and RS 232 Connections

Application Connector Pin No.	Keyboard Connector J1 Pin No.	Function of Signal		
1	1	Ground		
2	2	Keyboard Strobe	VIA Pin 40 — CA1	Gated by U58
3	3	Light Pen Strobe	VIA Pin 39 — CA2	CRT Pin 3
4	4	Keyboard Bit 0	VIA Pin 2 — PA0	
5	5	Keyboard Bit 1	VIA Pin 3 — PA1	
6	6	Keyboard Bit 2	VIA Pin 4 — PA2	
7	7	Keyboard Bit 3	VIA Pin 5 — PA3	
8	8	Keyboard Bit 4	VIA Pin 6 — PA4	
9	9	Keyboard Bit 5	VIA Pin 7 — PA5	
10	10	Keyboard Bit 6	VIA Pin 8 — PA6	
11	11	Keyboard Bit 7	VIA Pin 9 — PA7	
12		Character Width	VIA Pin 10 — PB0	
13		Character Width	VIA Pin 11 — PB1	
14		Character Width	VIA Pin 12 — PB2	
15		Character Width	VIA Pin 13 — PB3	
16		Unassigned	VIA Pin 14 — PB4	
17		Unassigned	VIA Pin 15 — PB5	
18		Unassigned	VIA Pin 16 — PB6	
19		Unassigned	VIA Pin 17 — PB7	
20		Unassigned	VIA Pin 18 — CB1	
21		Supress Control Codes	VIA Pin 19 — CB2	
22		No Connection		
	12	+5 volt Keyboard Power		

Application Connector Pin No.	RS 232 Connector J2 Pin No.	Function of Signal		
A		+5 volts if provided by host computer		
B		Video Signal Output		
C		Video Signal Ground		
D		No Connection		
E		No Connection		
F		RxC	Receive Clock	ACIA Pin 5
H	4	$\overline{\text{RTS}}$	Ready to Send	ACIA Pin 8
J	5	$\overline{\text{CTS}}$	Clear to Send	ACIA Pin 9
K		TxD	Transmit Data	ACIA Pin 10
L	20	$\overline{\text{DTR}}$	Data Terminal Ready	ACIA Pin 11
M		RxD	Receive Data	ACIA Pin 12
N	6	$\overline{\text{DSR}}$	Data Set Ready	ACIA Pin 17
P	8	$\overline{\text{DCD}}$	Data Carrier Detect	ACIA Pin 16
R	24	TTY Keyboard Return	20 mA Current Loop	
S	25	TTY Printer Return	20 mA Current Loop	
T	12	TTY Keyboard	20 mA Current Loop	
U	13	TTY Printer	20 mA Current Loop	
V	2	RS 232 IN	Data Input	
W	3	RS 232 OUT	Data Output	
X		Optional -12 volts for standard RS 232		
Y		Unassigned		
Z		Unassigned		
	1	Ground		
	7	Ground		

Building a Cable

If you are not connecting the VIDEO PLUS to your AIM/SYM/KIM via the MOTHER PLUS or an Expansion Cable, then you must build your own cable. While this is a fairly straightforward task, it must be done carefully. The cable is comprised of two dual 22 pin connectors interconnected by wires which should be between 5 and 8 inches in length. The following table gives the wiring list. Note that it is almost one-to-one except for a shift in the address lines.

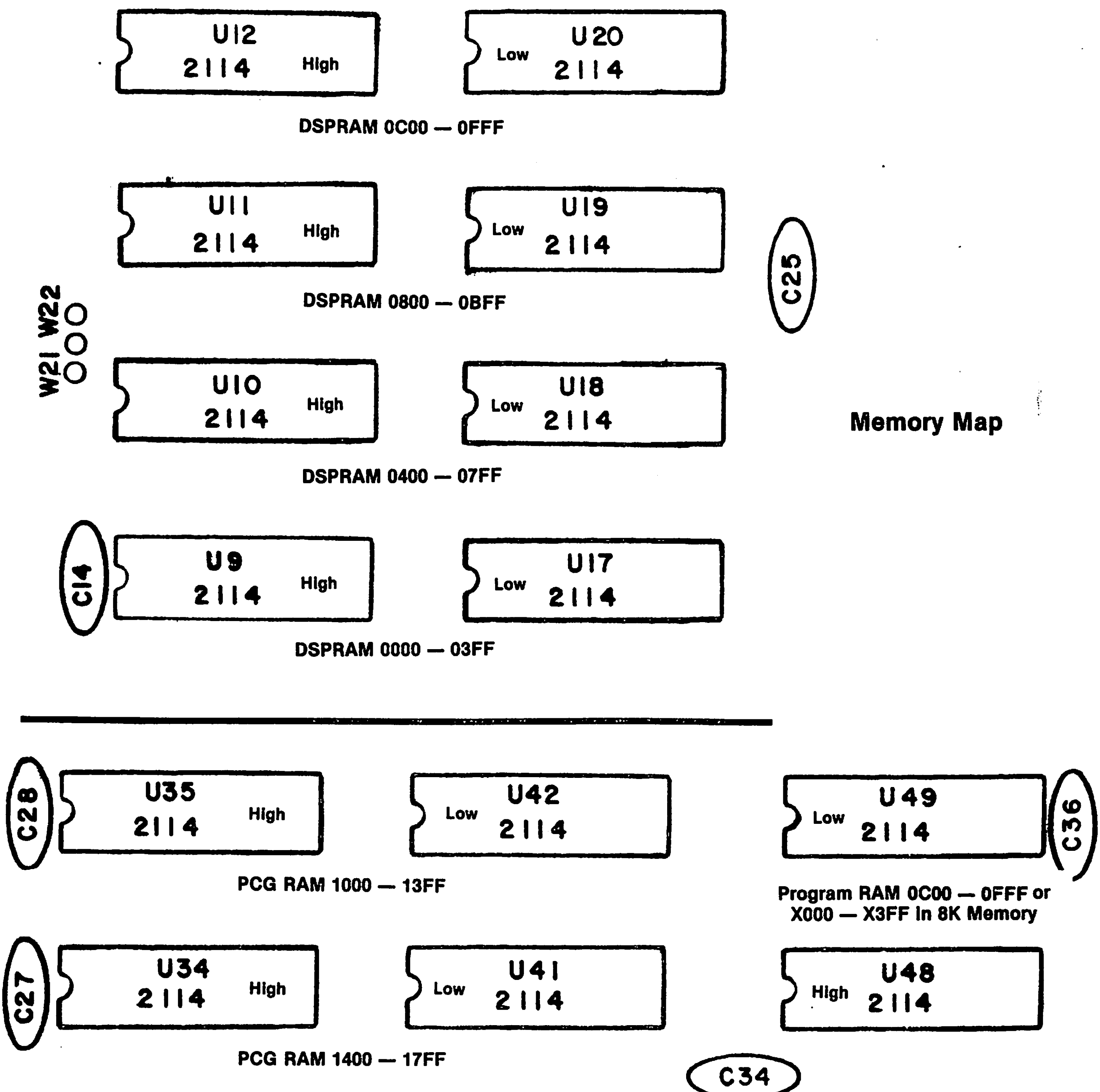
Expansion Connector Cable

AIM/SYM/KIM PIN No.	VIDEO PLUS PIN No.	Function of Signal
1	NC	[SYNC is not used]
[22]	1	Ground
2	NC	[RDY is not used]
3	NC	[Phase 1 is not used]
4	4	IRQ Interrupt Request. Interrupt LOW.
5	NC	[Not used]
6	NC	[NMI Non-Maskable Interrupt. Not used]
7	7	RES Reset. True when LOW.
8	8	Data Bit 7
9	9	Data Bit 6
10	10	Data Bit 5
11	11	Data Bit 4
12	12	Data Bit 3
13	13	Data Bit 2
14	14	Data Bit 1
15	15	Data Bit 0
NC	16	Decode Signal to KIM only. Application Connector A-K
17	NC	[Varies by microcomputer]
18	NC	[Varies by microcomputer]
NC	19	+8 volts optional power
NC	20	+8 volts optional power
21	21	+5 volts power. **Do NOT** connect to microcomputer pin 21 if separate + 5 V supplies are to be used for the microcomputer and VIDEO PLUS!
22	22	Ground. Also connects to 1, A, and Z on VIDEO PLUS
[22]	A	Ground
A	B	Address Bit 0
B	C	Address Bit 1
C	D	Address Bit 2
D	E	Address Bit 3
E	F	Address Bit 4
F	H	Address Bit 5
H	J	Address Bit 6
J	K	Address Bit 7
K	L	Address Bit 8
L	M	Address Bit 9
M	N	Address Bit 10
N	P	Address Bit 11
P	R	Address Bit 12
R	S	Address Bit 13
S	T	Address Bit 14
T	U	Address Bit 15
U	V	Phase 2
V	W	Read/Write bar
W	NC	[Read bar/Write. Not used]
X	NC	[Cassette Test. Not used]
Y	NC	[Phase 2 bar. Not used]
[+5]	Y	+5V external or from pin 21
Z	NC	[Ram R/W. Not used]
[22]	Z	Ground

Notes: There is one ground connection on the microcomputer expansion connector. This is normally tied to all four ground connections of the VIDEO PLUS expansion connector. An additional ground may be run between pin 1 on the AIM/SYM/KIM application connector and pin 1 on the VIDEO PLUS application connector. This is not required, but, in general, the more grounds the better. All five separate ground connections on the VIDEO PLUS are tied together.

If the +5 volts is common between the microcomputer and the VIDEO PLUS, then the connections should be made between pin 21 on the micro and pins 21 and Y on the VIDEO PLUS. If the +5 volts is provided by separate supplies, then there should be no connection between these pins. Pins 21 and Y of the VIDEO PLUS should go directly to the power supply.

The somewhat clumsy interconnection scheme presented above was defined by MOS Technology, inventors of the 6502 and KIM, for their original memory expansion boards and mother board. Even though these boards are no longer made, the definitions of the interconnections have been adopted by most manufacturers of expansion boards for the KIM. Since the AIM and SYM have the same microcomputer expansion pinouts, they will also work with this expansion bus scheme. This bus is sometimes called the 'KIM-4 Bus', referring to the MOS Technology Mother Board which defined the bus in hardware.



Memory Map

Video Plus Testing and Field Repair

You should test the VIDEO PLUS RAM, both Display and Programmable Character Generator [PCG], when you initially set up your system. You may also want to test it occasionally to make sure it is all working correctly. Finally, you will definitely want to test it whenever you have any reason to suspect that it is not working properly. The VIDEO PLUS Memory Test was adapted from "Memory Test" by Jim Butterfield, **The First Book of KIM**, edited by Butterfield, Ockers and Rehnke, pages 122-123.

> "Testing RAM isn't just a question of storing a value and then checking it. It's important to test for interference between locations. Such tests often involve writing to one location and then checking all other locations to see they haven't been disturbed; this can be time consuming.
>
> This program checks memory thoroughly and runs exceptionally fast. It is adapted from an algorithm by Knaizuk and Hartmann published in "**IEEE Transactions on Computers**", April 1977.
>
> The program first puts value FF in every location under test. Then it puts 00 in every third location, after which it tests all locations for correctness. The test is repeated twice more with the positions of the 00's changed each time. Finally, the whole thing is repeated with the FF and 00 values interchanged.
>
> To Run: Set the addresses of the first and last memory pages you wish to test into location 0000 and 0001 respectively. Start the program at address 0002; it will halt with a memory address on the display. If no faults were found, the address will be one location past the last address tested. If a fault is found, its address will be displayed."

The VIDEO PLUS version of the Memory Test will run on an AIM, SYM or KIM. See **MEMORY TEST** for a complete source listing. Load the program from the cassette tape provided or by hand from the listing. Set the starting page in 0000 [a 40 if the VIDEO PLUS is selected to start at location 4000], the ending page in 0001 [a 47 if the VIDEO PLUS is selected at 4000 and has 2K bytes of RAM], and then start the program at 0002. The program should only take a few seconds, depending on the amount of memory to be tested. It should exit by printing a memory address on the display of your AIM, SYM, or KIM. If the address is the last address tested plus 1 [4800 if 47 was the last address specified in location 0001], then the test did not detect any errors and your memory is probably functioning correctly. If any other address is displayed, it indicates that an error was found at the address displayed. The contents of the displayed address provide a clue as to the problem. The contents of a correct address after the test will be an FF or a 00.

If the error address contains a value that has missing bits [FB for example] or added bits [20 for example], then a particular 2114 memory chip may be defective. Using the **Memory Map**, determine which chip is suspect and try exchanging it with another chip. Several things may happen:

If the chip is truly defective, the problem will move with the chip. For example the test which failed before at some location with 'FB' may now fail and show 'BF'. If this occurs, the 2114 probably needs to be replaced.

If the problem was due to improper seating of the 2114 in its socket, or due to oxidation or dirt in the socket, then the test may now pass since the problem may have been corrected by physically moving the chip.

If the problem is due to some cause other than a defective 2114, then the problem may continue to occur at the same address no matter what 2114 chip is put there. In this case, you can try to solve the problem by a little more sophisticated chip swapping. The board has been designed so that there is generally more than one IC chip of each type. This makes it easy to swap identical chips to help pin-point an IC problem. For example, you can swap the two 74LS245's at U16 and U40 and see if this solves, or at least changes, the problem. Other common ICs include the 74LS157s, 74LS74s, 74LS367s, and so forth. If a chip can be determined to be defective by swapping it, then its replacement is straight-forward. If none of these 'cures' work, then you may have to send the board back for service.

Your VIDEO PLUS board was burned in and fully tested before shipment. If after following the steps outlined in **Set Up and Check Out** the board does not appear to work correctly, or if it ever seems to stop working properly, then the following steps should be taken:

1. Check that the board is receiving adequate power.

   Place the ground lead of a voltmeter or 'scope on any convenient ground on the board. The edge connector pins E-1, E-22, E-A, E-Z and A-1 are all ground.

   Measure the voltage at the three wire jumpers on the header located next to the heat sinks. They should be +5V. If they are all correct, skip to step 2. Otherwise, check that the header is in the correct position with a jumper in the position nearest the heat sink for +5 regulated or without a jumper in that position for +8V unregulated.

If any voltage does not meet the specification, then there is a problem. If the input being provided is not adequate, either +8V unregulated or +5V regulated, then you must provide the correct values. If the input is +8V and is correct but one or more of the output jumpers are not +5V, then a voltage regulator may have gone bad. A simple test is to remove the header and put a single wire jumper between the two pins at the end of the header socket nearest the edge of the board to provide the input voltage to the regulators. Now measure the voltage output at each of the regulators. The output voltage is on the pin furthest from the header. If one of the regulators does not show the proper +5V, then it must be replaced. If the regulators now show the proper voltage, then the problem is probably *not* in the regulator, but is due to something on the board causing a short or overload condition.

It might be a faulty regulator which fails under load. If possible, try running the VIDEO PLUS board with regulated +5V and see if it works. If it does, then it is most likely a regulator which must be replaced. If it does not work with the +5V, then there is a problem elsewhere on the board.

2. Make sure that all IC chips are firmly in their sockets. It is possible for the chips to come loose during shipping and handling. Push each chip firmly into its socket. Our service department has received boards for repair which needed nothing more than this simple step to "repair". Also check for any loose wire, broken capacitor, scratches, etc.

3. Check all of the connections between your microcomputer, the VIDEO PLUS board, the power supply, and any other boards in your system. If you have other boards in your system, such as DRAM PLUS, MOTHER PLUS, etc., make sure that the addressing is properly set up so that there are no addressing conflicts. Make sure that the VIDEO PLUS address switch is **ON** for the address space you require.

4. If the problem seems to be in the memory portion of the board, then run the Memory Test described above and follow the instructions on making repairs.

5. If the problem is in the video output, there are several things you can do. First, remember that there will be no video output, no horizontal sync, no vertical sync, no cursor, etc. until the video initialization program has been run.

   If you are connected to a video monitor capable of handling 80 characters, then skip to the next paragraph. If you are connected to a television set via an RF modulator and can only handle 40 characters per line, then make sure the TV/Monitor switch is in the TV position.

   If your video does not seem to stabilize when the initialization program is run, check the connections between the monitor and the VIDEO PLUS. Also try adjusting the controls on your monitor or TV: Sync, Contrast, etc.

6. If you get an image on your display, but it jitters, tears, is off-center, or any similar problem, you may have to adjust the parameters used by the video initialization program. A complete description is provided in the CRT Controller data sheet included in the Manual. The calculation of the various timings is non-trivial. You will probably do better by trial-and-error, using the parameters provided in the VIDEO PLUS software as a starting point and making gradual modifications to improve your display.

   If your problems seem to be in the horizontal dimension, then concentrate on the first four entries in the initialization table. This table may be found in the **ASK VIDEO PLUS Software** Manual. **H TOTAL** controls the total time of a single horizontal scan across the display. **H DISPLAYED** controls the number of characters which will be displayed across a line. **H SYNC POSITION** controls the position along the line at which the horizontal sync occurs. **H SYNC WIDTH** is a "fudge factor" to make the horizontal timing come out right. Feel free to experiment with these four values. You cannot harm anything. Make small adjustments, reinitialize the video program, and see what happens. The values provided initially have been found to work on a number of different types of monitors.

   If your problem appears to be in the vertical dimension, as indicated by jitter, missing lines, and roll, then try working with the vertical controls. **V TOTAL** determines the maximum number of character display lines and is set to the maximum - 1, e.g. 17 hex [23 decimal] for a 24 line display. **V ADJUST** is a vertical "fudge factor". **V DISPLAYED** is the number of character lines to actually be displayed and may not exceed V TOTAL + 1, e.g. 18 hex [24 decimal]. **V SYNC POSITION** is normally set to equal V TOTAL + 1, e.g. 18 hex.

   You will normally not have to modify the other initialization values. For more information, consult the CRT Controller data sheet.

7. If possible, try running the VIDEO PLUS with another microcomputer. With the addition of this board you may be excercising portions of you AIM, SYM or KIM which have not been used before. Since all that VIDEO PLUS requires from the host microcomputer is proper address, data, and control signals, your host computer is probably not the problem, but it could be and trying another system is a very simple test.

8. If none of the above steps work, give us a call at 617/256-3649 and ask for **Service**, and perhaps we can help make a diagnosis. The schematic in the manual should provide enough information for a skilled electronics engineer or service person to make further tests. If the above steps do not solve the problem, you may have to send the board back for service. We have had very few boards returned to date, and we make every effort to get service boards out immediately.

Parts List

Item	Part Number	Qty.	Description	
			Integrated Circuits	
1.	U21, U22	2	74LS00	Quad 2-Input Positive NAND Gates
2.	U8, U33, U50, U52	4	74LS04	Hex Inverters
3.	U59	1	74LS08	Quad 2-Input Positive AND Gates
4.	U51, U57	2	74LS30	8-Input Positive NAND Gates
5.	U30, U53, U56	3	74LS32	Quad 2-Input Positive OR Gates
6.	U13, U55	2	74LS74	Dual D-type Positive Edge Trig. Flip Flop
7.	U58	1	74LS86	Quad 2-Input Exclusive OR Gates
8.	U38	1	74LS138	3 to 8 Line Decoder
9.	U6	1	74LS139	Dual 2 to 4 Line Decoder
10.	U45	1	74LS156	3 to 8 Line Decoder Open Collector
11.	U3 — U5, U25 — U27, U36, U43	8	74LS157	Quad 2 to 1 Line Data Selector/Multiplexers
12.	U14	1	74LS161	Synchronous 4-Bit Binary Counter
13.	U44	1	74LS166	8-Bit Shift Register
14.	U28, U29	2	74LS174	Hex Quad D-type Flip Flops
15.	U16, U40	2	74LS245	Octal Bus Tranceivers with 3 State Outputs
16.	U15, U23, U24, U31, U32	5	74LS367	Hex Bus Drivers with 3 State Outputs
17.	U9, U10, U17, U18, U48, U49	6	2114L	Static RAM 450 NS Low Power
18.	U37, U47	2	2716	2K by 8-bit EPROM
19.	U39	1	6522	Versatile Interface Adapter
20.	U2	1	6845/6405/6545	CRT Controller
			Resistors and Capacitors	
21.	R6	1	10 Ohm	1/4 Watt Resistor
22.	R9	1	27 Ohm	1/4 Watt Resistor
23.	R7*, R8	1—2	100 Ohm	1/4 Watt Resistor
24.	R13, R14	2	150 Ohm	1/4 Watt Resistor
25.	R11, R12	2	200 Ohm	1/4 Watt Resistor
26.	R10, R25	2	300 Ohm	1/4 Watt Resistor
27.	R1, R2, R3, R4	4	330 Ohm	1/4 Watt Resistor
28.	R5	1	510 Ohm	1/4 Watt Resistor
29.	R26, R48, R49	3	680 Ohm	1/4 Watt Resistor
30.	R19, R20, R21	3	1.0K Ohm	1/4 Watt Resistor
31.	R40	1	1.6K Ohm	1/4 Watt Resistor
32.	R15	1	2.2K Ohm	1/4 Watt Resistor
33.	R41	1	2.4K Ohm	1/4 Watt Resistor
34.		22	3.3K Ohm	1/4 Watt Resistors
35.	R17	1	5.6K Ohm	1/4 Watt Resistor
36.	R18	1	6.8K Ohm	1/4 Watt Resistor
37.	R22	1	10K Ohm	1/4 Watt Resistor
38.	RP1, RP2	2	3.3K Ohm	SIP 8 Resistor Pack
39.	C4, C5*, C6, C39*, C40*	2—5	22 pF	35V Capacitor
40.	C8 — C38	31	.01 Mfd	50V Capacitor
41.	C7	1	.1 Mfd	50V Capacitor
42.	C1, C2, C3	3	22 Mfd	25V Capacitor
			Diodes and Transistors	
43.	D1, D2	2	1N914	Diode
44.	Q3, Q5, Q7	3	2N2222	Transistor
45.	Q4, Q6	2	2N2907	Transistor
46.	Q1, Q2	2	7805	5V T-220 Voltage Regulator
			Sockets	
47.		16	14 pin	Low Profile Socket
48.		25	16 pin	Low Profile Socket
49.		14	18 pin	Low Profile Socket
50.		3	20 pin	Low Profile Socket
51.		2	24 pin	Low Profile Socket
52.		1	28 pin	Low Profile Socket
53.		3	40 pin	Low Profile Socket

			Miscellaneous
54.	Y1	1	16 Mhz Crystal
55.	H1	1	16 pin Header
56.	SW1, SW2, SW3	3	8 position DIP switch
57.		2	T-220 Heat Sink
58.		1	Video Output Jack & Bracket
59.		30	Amp Jumper Post
60.		3	Jumper Block
61.		1	Jumper Wire
62.		4	4-40 Nuts and Bolts

Notes:
1. ICs may vary from the list. Some may be provided as non-LS parts; some may have a different manufacturers part number — particularly the RAM memory and the CRT Controller.
2. Resistor values may vary slightly from the list where the value is not critical.
3. Items marked by an * are determined at testing time and may or may not be included.

Option Parts Lists

Item	Part Number	Qty.	Description
			Additional Memory — Option 1
1.	U11, U12, U19, U20 [Display] U34, U35, U41, U42 [Prog. Char]	8	2114L Static RAM 450 NS Low Power
			ACIA Communications — Option 2
1.	U46	1	6551 Asynch. Comm. Interface Adapter
2.	Y2	1	1.8432 Mhz Baud Rate Generator Crystal
3.	J2	1	RS 232 Standard D-type Connector
4.	SW4	1	10 position SPDT DIP Switch
5.	U54	1	74LS04 Hex Inverter
			Stand-Alone — Option 3
1.	U1	1	6502 Microprocessor
2.	U7	1	74LS161 Sychronous 4-bit Counter
3.	SW5	1	SPDT Momentary Contact Switch

Notes:
1. These options may be purchased at any time and easily installed by the user in the field. All of the support components such as resistors, capacitors, diodes, etc. are already in place on the board, and sockets are provided for the ICs.
2. If these options are ordered with the VIDEO PLUS board, then they will be installed and tested.
3. The code to support these options is already included in the ASK Video Plus Software EPROM. Additional software may be provided in the future to users requesting it.

```
                MEMORY TEST  9 FEBRUARY 1979

                MEMORY ORG   $0000

                ACCESS *     $8B86  SYM-1 ACCESS ENTRY
                OUTBYT *     $82FA  SYM-1 OUTPUT BYTE
                SCANDS *     $8906  SYM-1 SCAN DISPLAY

                GOKIM  *     $1C4F  KIM-1 ENTRY POINT
                LPOINT *     $00FA  KIM DISPLAY POINTERS
                HPOINT *     $00FB


                ASCOUT *     $EF7B  AIM 65 OUTPUT ASCII

0000 20         BEGIN  =     $20    STARTING TEST PAGE
0001 3F         END    =     $3F    ENDING TEST PAGE

0002 A9 00      START  LDAIM $00    ZERO POINTERS
0004 A8                TAY          FOR LOW ORDER ADDRESSES
0005 48                PHA          SET ALL STATUS BITS TO ZERO
0006 28                PLP
0007 85 D1             STAZ  POINTL

0009 85 D3      BIGLP  STAZ  FLAG   = 00 FIRST PASS, = FF SECOND PASS
000B A2 02             LDXIM $02
000D 86 D5             STXZ  PASS   SET 3 TESTS EACH PASS

000F A5 00      NPASS  LDAZ  BEGIN  SET POINTER TO
0011 85 D2             STAZ  POINTH START OF TEST AREA
0013 A6 01             LDXZ  END
0015 A5 D3             LDAZ  FLAG
0017 49 FF             EORIM $FF    REVERSE FLAG
0019 85 D4             STAZ  FLIP   = FF FIRST PASS, = 00 SECOND PASS

001B 91 D1      CLEAR  STAIY POINTL WRITE FLIP VALUE
001D C8                INY          INTO ALL LOCATIONS
001E D0 FB             BNE   CLEAR
0020 E6 D2             INCZ  POINTH
0022 E4 D2             CPXZ  POINTH
0024 B0 F5             BCS   CLEAR

                FLIP VALUE IN ALL LOCATIONS.  NOW CHANGE 1 IN 3

0026 A6 D5             LDXZ  PASS
0028 A5 00             LDAZ  BEGIN  SET POINTER
002A 85 D2             STAZ  POINTH BACK TO START

002C A5 D3      FILL   LDAZ  FLAG   CHANGE VALUE
002E CA         TOP    DEX
002F 10 04             BPL   SKIP   SKIP 2 OUT OF 3
0031 A2 02             LDXIM $02    RESTORE 3 COUNTER
0033 91 D1             STAIY POINTL CHANGE 1 OUT OF 3
0035 C8         SKIP   INY
0036 D0 F6             BNE   TOP
0038 E6 D2             INCZ  POINTH NEW PAGE
```

```
003A A5 01           LDAZ   END    HAVE WE PASSED
003C C5 D2           CMPZ   POINTH END OF TEST AREA?
003E B0 EC           BCS    FILL   NO. KEEP GOING

               MEMORY SET UP.  NOW TEST IT

0040 A5 00           LDAZ   BEGIN  SET POINTER
0042 85 D2           STAZ   POINTH BACK TO START
0044 A6 D5           LDXZ   PASS   SET UP 3 COUNTER

0046 A5 D4     POP   LDAZ   FLIP   TEST FOR FLIP VALUE
0048 CA              DEX           2 OUT OF 3 TIMES
0049 10 04           BPL    SLIP   OR
004B A2 02           LDXIM  $02    1 OUT OF 3 TIMES
004D A5 D3           LDAZ   FLAG   TEST FOR FLAG VALUE

004F C1 D1     SLIP  CMPIY  POINTL HER IS THE TEST
0051 D0 15           BNE    OUT    BRANCH IF FAILED
0053 C8              INY           BUMP POINTER
0054 D0 F0           BNE    POP    IF NOT DONE, KEEP BOING
0056 E6 D2           INCZ   POINTH
0058 A5 01           LDAZ   END    TEST END
005A C5 D2           CMPZ   POINTH
005C B0 E8           BCS    POP

               ABOVE TEST OKAY.  CHANGE AND REPEAT

005E C6 D5           DECZ   PASS   CHANGE 1 IN 3 POSITION
0060 10 AD           BPL    NPASS  AND DO NEXT PASS
0062 A5 D3           LDAZ   FLAG   INVERT FLAG
0064 49 FF           EORIM  $FF    FOR PASS TWO
0066 30 A1           BMI    BIGLP  AND REPEAT BIG LOOP

0068 84 D1     OUT   STYZ   POINTL PUT LOW ORDER ADDRESS FOR DISPLAY
006A AD FD FF        LDA    $FFFC  TEST HIGH BYTE OF INTERRUPT VECTOR
006D C9 8B           CMPIM  $8B    = SYM-1
006F F0 46           BEQ    SYM
0071 C9 E0           CMPIM  $E0    = AIM 65
0073 F0 0B           BEQ    AIM

0075 A5 D1     KIM   LDAZ   POINTL MOVE POINTERS FOR KIM
0077 85 FA           STAZ   LPOINT
0079 A5 D2           LDAZ   POINTH
007B 85 FB           STAZ   HPOINT
007D 4C 4F 1C        JMP    GOKIM  RETURN TO KIM MONITOR

0080 A5 D2     AIM   LDAZ   POINTH MOVE DATA FOR AIM DISPLAY
0082 85 D6           STAZ   AHIGH
0084 A5 D1           LDAZ   POINTL
0086 85 D7           STAZ   ALOW
0088 A2 00           LDXIM  $00    GET DATA AT ADDRESS
008A A1 D1           LDAIX  POINTL
008C 85 D8           STAZ   ADATA
008E A2 13           LDXIM  $13    START AT POSITION 19.

0090 8A        ALOOP TXA           SAVE X VALUE
```

```
0091 48                    PHA              ON STACK
0092 A0 04                 LDYIM $04        SHIFT 4 POSITIONS PER CHARACTER
0094 A5 D8                 LDAZ  ADATA      GET DATA
0096 29 0F                 ANDIM $0F        MASK TO NIBBLE
0098 C9 0A                 CMPIM $0A        TEST DECIMAL
009A 30 03                 BMI   AOKAY      DECIMAL
009C 18                    CLC              A - F.  MUST CONVERT TO
009D 69 07                 ADCIM $07        ASCII
009F 18           AOKAY    CLC              FINISH CONVERSION
00A0 69 B0                 ADCIM $B0        ASCII + AIM FLAG
00A2 20 7B EF              JSR   ASCOUT     OUTPUT TO DISPLAY
00A5 46 D6        AMOVE    LSRZ  AHIGH      MOVE TO NEXT NIBBLE
00A7 66 D7                 RORZ  ALOW
00A9 66 D8                 RORZ  ADATA
00AB 88                    DEY
00AC D0 F7                 BNE   AMOVE

00AE 68                    PLA              RESTORE X
00AF AA                    TAX
00B0 CA                    DEX
00B1 E0 0E                 CPXIM $0E        DONE?
00B3 B0 DB                 BCS   ALOOP      NO
00B5 90 C9                 BCC   AIM        YES. REPEAT

00B7 20 86 8B     SYM      JSR   ACCESS     ENABLE SYM MOEMORY
00BA A5 D2                 LDAZ  POINTH
00BC 20 FA 82              JSR   OUTBYT     OUTPUT
00BF A5 D1                 LDAZ  POINTL
00C1 20 FA 82              JSR   OUTBYT
00C4 A0 00                 LDYIM $00
00C6 B1 D1                 LDAIY POINTL     GET DATA
00C8 20 FA 82              JSR   OUTBYT

00CB 20 06 89     DISPLY   JSR   SCANDS     SCAN DISPLAY
00CE 4C CB 00              JMP   DISPLY     CONTINUE

00D1 00           POINTL   =     $00
00D2 00           POINTH   =     $00
00D3 00           FLAG     =     $00
00D4 00           FLIP     =     $00
00D5 00           PASS     =     $00
00D6 00           AHIGH    =     $00
00D7 00           ALOW     =     $00
00D8 00           ADATA    =     $00
```

ASK VIDEO PLUS II SOFTWARE IN EPROM

Introduction

All the software required to run the Video Plus II board is supplied on an EPROM installed in every board. This software provides the following capabilities:

- Instant start-up for most configurations
- Complete compatibility with AIM/SYM/KIM Monitors, Editors and BASICs
- Full ASCII keyboard support including lower case
- Expanded functions from the AIM internal keyboard
- Completely relocatable, including start-up code

Instant Start-up

To start the video board, first make sure the dip switches SW1 and SW3 are set as follows:

SW1	on	1 3
	off	2 45678
SW3	on	4
	off	123 5678

This setting of SW1 will configure the keyboard for positive data and negative strobe, and disable the flicker suppression. This setting of SW3 locates the board at addresses 6000 to 7FFF. (SW2 settings are ignored.) Also, make sure jumper W10 is installed. This will configure the board for most applications. Refer to the section on Switches and Jumpers on Page 9 for other configurations.

Now run the start-up routine as follows:

AIM 65:

<*> = 7EB8

G/ (return) will fully initalize the video board. All monitor output will go to the video monitor. Input will continue to come from the internal keyboard.

SYM-1:

GO 7EB8 (return) will fully initialize the video board. Monitor output will go to the video display and monitor input will come from the external keyboard connected to the Video Plus board.

KIM-1:

Load address	17FE with	00
	17FF	02
	00E8	04
	00E9	02
	00EA	01
	00EB	02

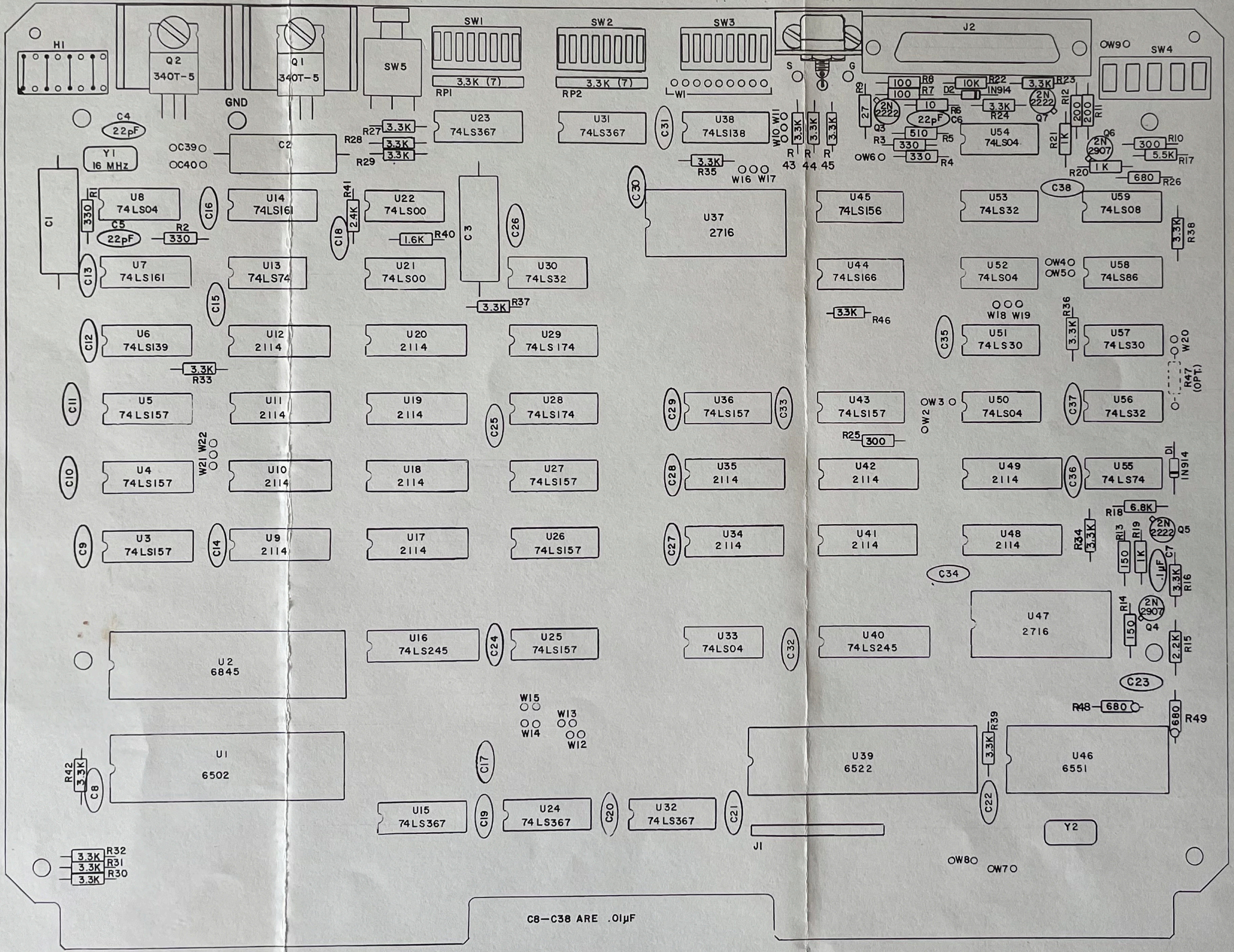
and load the following short program:

```
                    ORG   $0200
0200 20 00 00       JSR   KBWAIT   ADDRESS WILL BE FILLED IN BY KINIT
0203 20 00 00       JSR   OUTTV    ADDRESS WILL BE FILLED IN BY VINIT
0206 4C 00 02       JMP   $0200
```

Now load address 7EB8 and GO. The video board will be initialized and any characters typed on the keyboard will appear on the display.


*R7 VALUE ADJUSTED TO PROVIDE 1 VOLT FINAL VIDEO OUTPUT.
*C5 VALUE ADJUSTED TO PROVIDE CLEAN 16MHz CLK. SIGNAL.
OPTION ①= 4K RAM
OPTION ②= COMMUNICATIONS
OPTION ③= STAND ALONE
(OPTIONAL TUNING)
© COPYRIGHT 21 MARCH 1980
THE COMPUTERIST,® INC.
SCHEMATIC
VIDEO PLUS II™
D
TBC-112
D

REVISIONS			
LTR	DESCRIPTION	DATE	APPR



UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN INCHES, TOLERANCES: FRACTIONS ± 1/64, ANGLES ± 1/2°, 2 PL DEC ± .02, 3 PL DEC ± .005, PROJECTION	
NEXT ASSY	USED ON
APPLICATION	
DRAWN	DATE
DESIGNED	
CHECKED	
ENG APPR	
PROVEN	

THE COMPUTERIST® INC.

TITLE: ASSEMBLY VIDEO PLUS II™

SIZE	JOB NO.	DRAWING NO.	REV
D		TBC-112	D

SCALE: | SHEET | OF

Control Character Functions

Function		Control Character
Cursor Functions:	Up	U
	Down	D
	Left	L
	Right	R, FS
	Home	H
	Carriage return	M
	Line feed	J
Screen Functions:	Clear screen	X
	Clear from cursor to end	E
	Scroll to cursor position	S
Delete:	Always	K
	Except SYM BASIC	Delete (7F hex)
	SYM BASIC	__ (underline) (5F hex)
Break/Escape:	AIM, KIM	Escape (1B hex)
	SYM	B
Mode Toggles:	Lower case enable	A
	Keyboard echo	F
	Line feed after CR	Q
	PCG character enable	P
	Control character blanking	N
	Flicker test enable	T
	Special SYM mode	V
Other:	Start blanking	RS (1E hex)
	Stop blanking	US (1F hex)
	User-defined	W
Unassigned:		C, G, I, O, Z, GS (1D hex)

Control Character	Function
A	Toggle lower case to upper case conversion in keyboard routine
B	(SYM-only) Break key
C	(Unassigned)
D	Cursor down
E	Erase to end of screen
F	Toggle echo from keyboard routine (full/half duplex mode)
G	(Unassigned)
H	Home cursor
I	(Unassigned)
J	Line feed
K	Delete
L	Cursor left (backspace)
M	Carriage return (new line)
N	Toggle control character blanking
O	(Unassigned)
P	Toggle bit 80 mask on output (enable/disable PCG characters)
Q	Toggle suppression of line feed after CR
R	Cursor right (forward space)
S	Scroll to cursor
T	Toggle flicker test enable
U	Cursor up
V	Toggle RAE mode (SYM only)
W	User-defined function (initialized to null)
X	Clear screen
Y	Display control character
Z	(Unassigned on output) on input: return character from current cursor position
ESC	Escape - break on KIM or AIM (control F1 on AIM)
FS	Forward space - control \
GS	(Unassigned) - control] ∧control F2 on AIM
RS	Start blanking - control - control F3 on AIM internal keyboard
US	End blanking - control __ (not available on AIM internal keyboard)

Operating Instructions

AIM Monitor and Editor:

The Monitor and Editor will fail to read and write tapes properly if flicker suppression is enabled. This is easily remedied by turning off the flicker suppression, either by the control T toggle or by means of the switch on the Video Plus board (SW1, #8). The problem arises because the Monitor outputs the block number every time it reads or writes a block on tape. The flicker suppression slows this output so much that the cassette routine loses synchronization with the data on tape.

AIM BASIC:

The start-up routine at 7EB8 will initialize the video and an external keyboard. To use the added functions of the internal keyboard, re-initialize only the video by a JSR 6C00 (if the board is addressed at 6000), or change location 6C8E from 00 to 10, thus enabling the internal keyboard functions.

The ASK software adds two functions to the AIM internal keyboard, lower case characters and the control Z function. These functions are normally available when using an external keyboard. To use these additional features, initialize the video, but *not* the external keyboard. Then, from BASIC, POKE 42002,85. This will enable the additional features. You will use the same keyboard, but it will now provide lower case characters and the control Z function.

POKE 42002,13 will restore the original internal keyboard function.

These additional features are not supported for the AIM EDITOR.

SYM RAE:

The SYM Resident Assembler-Editor program requires a special mode. After initialization, while in the monitor, type control V and RETURN. Ignore the error message. Now start RAE with a .G B000. RAE will function normally, with all I/O going through the Video Plus board.

To restart RAE from the monitor, if the software has not been re-initialized, do not type control V again as this will turn off the RAE mode.

Lower case characters may be entered while using RAE, but the control A toggle will not work while in RAE. Type the control A before starting RAE to enable lower case characters.

SYM BASIC:

Type control B from RAE or .G C000 from the monitor. To enable or disable lower case character input, type control A.

Please note that BASIC will not accept lower case commands. Lower case characters may be used only within quotes in BASIC.

KIM-1:

KIM-1 users must first set up the break vector at 17FE-17FF, and the KIM initialization vectors at 00E8 to 00EB. The break vector should either return control to the monitor and contain address 1C00, or transfer control to a user program. The start-up routine ends with a break instruction and this vector must point to something. The KIM initialization vectors are used by the ASK Video software as pointers to the user's I/O vectors. The address of the output vector must be in 00E8-00E9 and the address of the input vector must be in 00EA-00EB.

The video and keyboard initialization routines assume that these pointers contain the addresses of these vectors. If the user has set up his own vectors, then these pointers must be set to point to ROM or other unused memory.

After the above vectors have been set, the KIM user may load address 7EB8 and GO to initialize the video display and the external keyboard.

Special Start-up Routines

This software package provides the AIM or SYM user with true "instant video" capability. The user need only load one address and go for all I/O to be handled by the Video Plus II board. No complicated setup procedures are required for most applications.

The address of the start-up routine is 1EB8 (hex) above the base address of the board. If the board is addressed at 6000, then the start-up address is 7EB8. This start-up routine will work regardless of where the board is addressed.

Base Address	Start-up Routine
2000	3EB8
4000	5EB8
6000	7EB8
8000	9EB8
A000	BEB8
C000	DEB8

The standard start-up assumes:

1. 24 lines x 80 character display format,
2. 2K display RAM,
3. ASK software uses first page of 1K on-board program RAM located at 0C00 above the board base address. Jumper W10 must be installed,
4. Both video and external keyboard are to be initialized.

The standard start-up may be used with the PCG RAM installed.

The user must supply a short start-up routine in any of the following cases:

1. User supplies new initialization table, to change display format,
2. Display RAM expanded to 4K,
3. ASK Software must use alternate RAM swapping page, specified by the user,
4. User wishes to initialize video but not keyboard, for AIM BASIC with expanded internal keyboard functions.

Any user initialization sequence must call one or more of the following routines: SETUP, VINIT, KINIT, VKINIT.

SETUP sets up the RAM swapping page (see "Details of Operation" for explanation),
VINIT initializes only the video,
KINIT initializes only the keyboard,
VKINIT initializes both video and keyboard.

The SETUP routine is always located at address 1900 (hex) plus the base address of the board:

Board Address	SETUP Address	USER SETUP Address
2000	3900	3902
4000	5900	5902
6000	7900	7902
8000	9900	9902
A000	B900	B902
C000	D900	D902

If the on-board program RAM is available, then the user should call SETUP at the first address (offset 1900, not 1902). If the user is specifying a different RAM swapping page then the address of that page must be in A register and the setup routine must be called at the USER SETUP address (offset 1902, not 1900).

When the user specifies a RAM swapping page, the following options are available:

1. PCG RAM, if no PCG characters are used,
2. On-board 1K program RAM relocated to an address outside the 8K space used by the video board, by means of Jumpers W11 and W1 and DIP switch SW1, position 7,
3. Off-board RAM.

Do not use:

1. Display RAM,
2. PCG RAM if PCG characters are being used,
3. Page zero or page one.

The ASK Video software has six entry points in addition to the two for SETUP discussed above. All calls to the Video software, except for the SETUP, must be made through the RAM swapping page. The entry points are as follows:

Routine	Offset on RAM Page	Default Case (Board at 6000)
VINIT	00	6C00
KINIT	05	6C05
VKINIT	0A	6C0A
OUTTV	0F	6C0F
KBTEST*	14	6C14
KBWAIT	19	6C19

The high byte of the address of these entry points is the page address of the RAM swapping page, 6C in the default case when the board is addressed at 6000, or whatever page is specified in the call to USER SETUP, as outlined above.

*NOTE: The keyboard test routine (KBTEST) in the ASK Video software tests a bit in the interrupt flags register of the VIA on the Video Plus and sets the carry flag if this bit is set. This bit in the IFR is also cleared by this routine. Therefore, a call to keyboard wait (KBWAIT) immediately after a call to keyboard test will *not* pick up the character which set the carry flag. This keyboard test routine was included to support the SYM monitor which uses it to wait until the BREAK key is released.

Video Initialization Table

This table, located in the EPROM at address 7E58 and in the RAM swapping page at address 6C58 (in the default configuration) is used to load the sixteen CRT controller registers during video initialization. These registers are described in detail in the CRT Controller Data Sheet in Appendix C.

Briefly, this is what the table contains:

R0	Horizontal Total Sweep Time as a multiple of character width
R1	Number of Characters Displayed per line
R2	Horizontal Sync Pulse Position
R3	Horizontal Sync Pulse Width
R4	Vertical Total Sweep Time
R5	Vertical Adjust
R6	Number of Lines Displayed
R7	Vertical Sync Pulse Position
R8	Interlace Mode and Dots per Character
R9	Number of Scan Lines per Character
R10	Cursor Scan Line Start and Cursor Mode
R11	Cursor Scan Line End
R12	High byte of screen address
R13	Low byte of screen address
R14	High byte of cursor address
R15	Low byte of cursor address

Examples of Alternate Video Initialization Tables

For a word-processing application, the user may want more space between characters and between lines. The following table provides a 72 character by 20 line display with three dots between characters and two scan lines between character lines. This allows a total of 1480 characters on the screen.

Table Entry	Hex Value		Address In Default Case
R0	60		6C58
R1	48	(72 characters/line)	6C59
R2	4C		6C5A
R3	0A		6C5B
R4	14	(20 lines displayed)	6C5C
R5	14		6C5D
R6	14		6C5E
R7	14		6C5F
R8	18	(10 dots/character)	6C60
R9	0D	(14 scan lines/character)	6C61
R10	6D		6C62
R11	0D		6C63
R12-R15	00		6C64-6C67

For an application requiring more than 80 characters per line and more than 24 lines, the following table provides a character width of only seven dots, thus allowing 112 characters per line. The standard character generator assumes an eight dot per character display and therefore is not suitable in this case. A new EPROM should be used to provide characters which are only five or six dots wide.

Table Entry	Hex Value		Address In Default Case
R0	83		6C58
R1	70	(112 characters/line)	6C59
R2	70		6C5A
R3	0A		6C5B
R4	1A	(26 lines displayed)	6C5C
R5	12		6C5D
R6	1A		6C5E
R7	1A		6C5F
R8	24	(7 dots/character)	6C60
R9	0B	(11 scan lines/character)	6C61
R10	6B		6C62
R11	0B		6C63
R12-15	00		6C64-6C67

This table allows a total of 2912 characters on the screen. Therefore, the third 1K of display RAM must be installed at U11 and U19. The default RAM swapping page may still be used in this case.

In general, the user may calculate values for a specialized Video Initialization Table by following these rules:

1. The Number of Dots per Character subtracted from 16 (decimal) and multiplied by four gives the value for R8 (Mode). For example, 8 dots: 16 – 8 = 8, 8*4 = 32 or 20 (hex); and 10 dots: 16 – 10 = 6, 6*4 = 24 or 18 (hex).

2. The Number of Dots per Character multiplied by the Horizontal Total (R0) should equal about 976. This is the length of the horizontal trace in dot times. Divide the Number of Dots per Character into 976 to get the Horizontal Total. For example, 8 dots: 976/8 = 122 or 7A (hex); and 9 dots: 976/9 = 108 or 6C (hex).

3. The Horizontal Characters Displayed (R1) must be less than the Horizontal Total and also less than the Horizontal Sync Position. It may be much less than the Horizontal Total. The Horizontal Sync Position must also be less than the Horizontal Total. Decreasing the value of the Horizontal Sync Position moves the image across the screen to the right. (H Total ≥ H Sync ≥ H Displayed)

4. The Number of Scan Lines per Character (R9) multiplied by the Vertical Total (R4) should equal about 288. This is the screen height in scan lines. Divide the Number of Scan Lines per Character into 288 to find the maximum Number of Lines Displayed.

5. The Number of Lines Displayed (R6) and the Vertical Sync Position (R7) should be equal to or less than the Vertical Total (R4). The Vertical Adjust (R5) should be between 16 and 32 decimal (10 and 20 hex) for all applications. Experimentation is the best way to find the optimal Vertical Adjust value, but in general it should decrease as the Vertical Total increases.

6. In general, the last four values (R12-R15) should be left as zero.

The user may provide a new Initialization Table or modify the existing one.

To supply a new table, load the low byte of absolute address of the table into the X register and high byte of the address into the Y register. Load the page address of the Display RAM start, usually the high byte of the base address of the board, into the A register and call VINIT or VKINIT. If the A register is zero when VINIT is called, then the default video initialization table, at address 58 on the RAM swapping page, will be used, and the program will assume that the Display RAM begins at the 8K boundary below the RAM swapping page.

To modify the existing Video Initialization Table, change the table in the RAM swapping page at addresses 58-67 (6C58-6C67 in the default case). Then call VINIT with the zero in the A register. A call to SETUP will restore the original video initialization table, so any changes to the table must be made after the call to SETUP and before the call to VINIT or VKINIT.

Do not call VINIT or VKINIT with zero in the A register unless the RAM swapping page is located within the 8K address space of the Video Plus board. This means that the RAM swapping page must either be in the 1K on-board program RAM with Jumper W10 installed in the PCG RAM.

If an alternate RAM swapping page is used, and the supplied video initialization table is also being used, then the user must load the X register with 58 (hex), the Y register with the address of the RAM swapping page and the A register with the display RAM start, as above, before the call to VINIT or VKINIT.

Examples of Alternate Start-up Routines

To use 0300 to 03FF as the RAM swapping page, with the video board addressed at 6000, and with the standard video initialization table:

```
0200 A9 03          LDAIM $03     RAM SWAPPING PAGE
0202 20 02 79       JSR   USER    SETUP
0205 A2 58          LDXIM $58     VIDEO INIT TABLE LOW
0207 A0 03          LDYIM $03     RAM SWAPPING PAGE
0209 A9 60          LDAIM $60     DISPLAY RAM START
020B 20 0A 03       JSR   VKINIT  OR VINIT ($0300)
020E 00             BRK
```

To use an alternate video initialization table located at 0300 to 030F and the default RAM swapping page at 6C00:

```
0200 20 00 79       JSR   SETUP
0203 A2 00          LDXIM $00     VIDEO INIT TABLE LOW
0205 A0 03          LDYIM $03     VIDEO INIT TABLE HIGH
0207 A9 60          LDAIM $60     DISPLAY RAM START
0209 20 00 6C       JSR   VINIT   OR VKINIT ($6C0A)
020C 00             BRK
```

ASCII Character Set In EPROM

A standard component of the Video Plus II board is a character generator in EPROM. This chip produces a full set of ASCII characters for the display. In addition, this EPROM includes a set of line graphics characters as well as bar graph elements. The complete contents of the EPROM are listed in Appendix D.

Each box of the chart contains a binary representation of the sixteen bytes corresponding to one character, with one byte per row. The hex code for this character is the same as the first two hex digits of its address. These two digits may be found above the character in the corresponding row of EPROM addresses. The third digit of the hex address specifies a row within the character. The chart is arranged in rows of eight characters with four rows per group of 32 characters. The four groups are printed two per page for a total of 128 characters in the 2K EPROM.

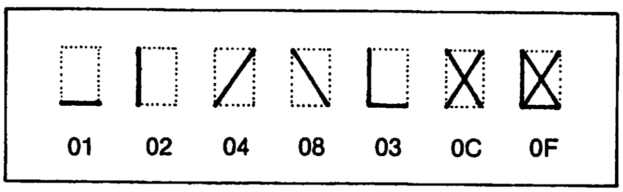
Characters 20 (hex) and 7F (hex) are standard ASCII printing characters on a 7 x 12 matrix with true descenders for lower case g, j, p, q, y. The eighth column and the thirteenth and fourteenth rows are blank to provide space between adjacent characters. The last two rows contain the binary code for the character with and without bit 80 for implementation of bit-mapped graphics (see relevant application note).

Line Graphics

Character codes 00 to 0F (hex) in the character generator EPROM are a set of line graphics, organized as follows:

if bit 01 is set a line is displayed at the bottom of the 8 x 12 character,
if bit 02 is set a line is displayed at the left edge of the 8 x 12 character,
if bit 04 is set a diagonal line is displayed from the upper right to lower left,
if bit 08 is set a diagonal line is displayed from upper left to lower right.

For example: character 00 is blank — 03 contains the left edge and bottom line,
0C contains both diagonals, and
0F contains all four lines.



These graphics characters cannot usually be output via the OUTTV routine because they are control characters and will be processed as such. They may be sent directly to the display RAM via the BASIC POKE function or the via system monitor. If control codes 00 to 0F are mapped to 19 (hex) then these codes will be displayed but the usual cursor down, cursor left, control code display toggle, etc. will not be available. Other control codes, above 0F, may be mapped to the required control functions to provide the ability to send graphics to the screen via OUTTV. Refer to the section on Control Character Mapping for details.

Bar Graph Elements

Characters 10 (hex) to 1D (hex) provide elements for generating horizontal and vertical bar graphs with various kinds of shading (refer to Appendix D for diagrams of these characters):

10 is an empty box
11 is a full box
12 is horizontal bars within a box
13 is vertical bars within a box
14 left diagonal lines wtihin a box
15 right diagonal lines within a box
16 left diagonals without the surrounding box
17 right diagonals without the surrounding box

18 ¼ vertical bar
19 ½ vertical bar
1A ¾ vertical bar
1B ¼ horizontal bar
1C ½ horizontal bar
1D ¾ horizontal bar
1E and 1F are intentionally left blank as they are used by the Video Plus board to provide character blank/unblank functions.

All graphics characters are on an 8 x 12 matrix and will be displayed without surrounding space. This allows adjacent characters to form continuous bars or shading.

Details of Operation

Program RAM and Swap Code

The ASK Video Plus II software is designed to be compatible with AIM, SYM and KIM Monitors, Editors, BASICs and user programs. A potential source of conflict is from use of pages zero and one. To avoid any problems from this direction, all use of page zero and one is completely transparent to the rest of the system. This is done by swapping about one third of page zero with RAM elsewhere in memory before and after execution of the software routines. This RAM swapping area is located in the RAM swapping page referred to above. The code which performs the swap is also located on this RAM page. Because the software uses this technique, two precautions must be observed: one, any interference with this program RAM page, or any change of its contents, except those changes described in this manual, will probably cause the video software to crash and require a new setup. Two, all calls to the video software must be made through this RAM page. This second point is important and easily explained: the code in EPROM could be called directly, but would not function properly since it refers to page zero locations which must first be loaded by the swap routine.

User-Defined Control Character Function

Provision has been included to call a user subroutine in response to a particular control character. This character is now the control W but any other control code may trigger this user function if the appropriate change is made to the map discussed below.

The user-defined control code function does a JSR to a page zero vector and updates the cursor position on return. The page zero vector is initialized to jump to an RTS instruction but the two-byte absolute address of a user subroutine may be substituted at addresses C0 and C1 on the RAM swapping page (6CC0 and 6CC1 in the default case). These locations are initialized to contain A0 and 7B, respectively, in the default configuration.

The user routine called by this method must observe the following rules. It must return by means of an RTS. It must not call any of the major Video Plus I/O routines, although it may call subroutines such as HOMECU and FLCKR. Locations 0000 (hex) to 0058 (hex) will contain the Video Plus Firmware variables and the usual contents of these locations will have been swapped into the RAM swapping page and may be found on that page, at address 88 (hex) and above (6C88 and above in the default case).

Do not use this function to start BASIC or any other program. The video software will crash immediately if the user function does not return.

Control Character Mapping

On output, every control character is translated by means of a map before it is processed. This map is found on the RAM swapping page at address 68 (6C68 to 6C87 in the default case). This map is initialized by the SETUP routine to provide a one-to-one translation; that is, each control code is mapped to itself unless the map is changed by the user.

The user may wish to disable certain control codes or substitute one control code for another. This may be done by means of the control code map. To disable a particular code, the location in the map corresponding to the code to be disabled must be changed to an unassigned control code like 03 or 1D. On output, the map will then substitute the "unassigned" or null function for the usual function of the selected control character. To find the address of the location in the map to change for a particular character, add 68 (hex) to the value of the character to be changed. For instance, control C is at location 68 + 03 = 6B on the RAM swapping page. To use control C as a break key, put a 1B (the code for ESCAPE) in location 6B (6C6B in the default RAM swapping page).

If the user program expects a control H to function as a backspace, then put a 0C (which is a control L, or "cursor left" command) in the map location corresponding to the control H, that is 68 + 08 = 70 on the RAM swapping page. The control H will now duplicate the usual control L function instead of its usual cursor home function.

If the cursor home function is also required, put a 08 in the map location corresponding to another, preferably unassigned, control character. For instance, to use control C, put the code corresponding to the desired function, in this case 08 for Home Cursor, in the map location corresponding to the control character which is to trigger this function, in this case control C, at location 68 + 03 = 6B on the RAM swapping page. In this way, any control character can be used for any available function. No control character can have more than one function since only one map location corresponds to each control character. But more than one control character can generate the same function because the corresponding control code can appear in more than one place in the map.

ASK VIDEO PLUS II FIRMWARE
BY ROBERT M. TRIPP

08 DECEMBER 1980
REVISION BY PAUL GEFFEN

PAGE ZERO EQUATES

```
CURSOR   *   $0000   CURSOR LINE ADDRESS
CRTREG   *   $0002   CRT REGISTER ADDRESS
SCRLOW   *   $0004   SCREEN LENGTH
ASK      *   $0006   AIM/SYM/KIM FLAGS
XTEMP    *   $0007   X SAVE
YTEMP    *   $0008   Y SAVE
LCHAR    *   $0009   LAST CHARACTER OUTPUT
CURPRM   *   $000A   CURSOR POSITION FOR AIM EDITOR
CURPO    *   $000B   CURSOR COLUMN
RAMPAG   *   $000C   DISPLAY RAM START PAGE
RAMEND   *   $000D   DISPLAY RAM END PAGE
COLMAX   *   $000E   LINE LENGTH
TEMP     *   $000F   TEMPORARY STORAGE
```

VECTORS

```
INDVEC   *   $0010   INDIRECT VECTOR
RSTVEC   *   $0013   RESTORE VECTOR
OVEC     *   $0016   OUTPUT TO VIDEO DISPLAY
KTVEC    *   $0019   KEYBOARD TEST FOR SYM
KWVEC    *   $001C   KEYBOARD INPUT
VVEC     *   $001F   VIDEO INITIALIZATION
KVEC     *   $0022   KEYBOARD INITIALIZATION
VKVEC    *   $0025   VIDEO AND KEYBOARD INIT

ZUSR     *   $0028   USER-DEFINED CONTROL CHAR. FUNCTION
ZHOMEC   *   $002B   HOME CURSOR
ZSTORE   *   $002E   STORE CHAR. SUBROUTINE
ZSTCHA   *   $0031   ALTERNATE ENTRY POINTS
ZSTCHB   *   $0034
ZSTCHY   *   $0037   STORE WITHOUT TESTS
ZDECRE   *   $003A   DECREMENT CURSOR

YTEMPK   *   $003D   TEMPORARY STORAGE
XTEMPK   *   $003E   FOR KEYBOARD ROUTINE
INVERT   *   $003F   INVERT KEYBOARD DATA

ZSTCHX   *   $0040   YET ANOTHER ENTRY POINT
ZSDEL    *   $0043   CHARACTER DELETE
ZCCVEC   *   $0046   CONTROL CHAR. VECTOR TABLE POINTER
ZCCXFR   *   $0049   CONTROL CHAR. TRANSFER VECTOR
ZCCMAP   *   $004C   CONTROL CHAR. MAP IN RAM
ZFLCKR   *   $004F   FLICKER TEST
OLD      *   $0052   SCROLL POINTER
NEW      *   $0054   SCROLL POINTER
```

ZERO PAGE VECTORS END AT 0055

ASK FLAGS

```
0X = AIM
1X = AIM INTERNAL KEYBOARD SERVICE
4X = KIM
8X = SYM
9X = SYM RAE SERVICE
X1 = UPPER CASE (0)/LOWER CASE (1)
X2 = STRIP BIT 80 (0)/PERMIT BIT 80 (1)
X4 = FULL DUPLEX (0)/HALF DUPLEX (1)
X8 = NOT AUTO CRLF (0)/AUTO CRLF (1)
```

AIM EQUATES

```
UIN      *   $0108   USER INPUT VECTOR
DILINK   *   $A406   DISPLAY VECTOR
CURPOZ   *   $A415   AIM MONITOR CURSOR (CURPO2)
DIBUFF   *   $A438   DISPLAY BUFFER
GETKEY   *   $EC40   AIM INTERNAL KEYBOARD SUBROUTINE
```

SYM EQUATES

```
ACCESS   *   $8B86   UNPROTECT SYSTEM RAM
NACCES   *   $8B9C   PROTECT SYSTEM RAM
TECHO    *   $A653   ECHO FLAG
INVEC    *   $A660   INPUT VECTOR
OUTVEC   *   $A663   OUTPUT VECTOR
INSVEC   *   $A666   INPUT TEST VECTOR
```

KIM EQUATES

```
NOTE: USER MUST SET UP THE FOLLOWING VECTORS
      FOR KIM-BASED SYSTEM.

KOUT     *   $00E8   CONTAINS ADDRESS OF KIM OUTPUT VECTOR
KIN      *   $00EA   CONTAINS ADDRESS OF KIM INPUT VECTOR
```

SETUP ROUTINE

```
SETUPS   ORG   $7900

FROM     *   $0000
TO       *   $0002
OFFSET   *   $0004

SIZE     *   $0067   NUMBER OF BYTES MOVED BY SETUP -1

SWPA     *   $002A   BYTES IN SWAP ROUTINE
```

```
                  SWPB    *    $0032   NEEDING RELOCATION INFO
                  TSTR    *    $0068   START OF TABLES IN ROM
                  TEND    *    $0094   END OF ROM VECTOR,TABLE
                  CCSTR   *    $0068   START OF CONTROL CHAR. MAP IN RAM
                  VSTR    *    $0098   START OF VECTORS IN RAM PAGE

                  SET UP RAM PAGE

7900 A9 00        SETUP   LDAIM $00    SET UP DEFAULT RAM PAGE
7902 48           USERS   PHA          OR USER SUPPLIES RAM PAGE NO. IN A
7903 D8                   CLD          CLEAR DECIMAL MODE
7904 A2 04                LDXIM $04    MOVE 5 PAGE ZERO LOCS
7906 B5 00        SLOOP   LDAZX $00    MOVE TO STACK
7908 48                   PHA
7909 CA                   DEX
790A 10 FA                BPL   SLOOP

                  FIND ACTUAL BOARD ADDRESS

790C A9 60                LDAIM $60    PUT RTS IN 0000
790E 85 00                STA   $0000  FOR LOCATION TEST
7910 20 00 00             JSR   $0000  FIGURE OUT WHERE WE ARE
7913 BA                   TSX          BACKUP STACK POINTER
7914 CA                   DEX
7915 9A                   TXS
7916 68                   PLA          PULL CURRENT PAGE NUMBER FROM STACK
7917 85 01                STA   FROM   +01 SET TO TRANSFER TABLES FROM
7919 85 04                STA   OFFSET
791B BD 07 01             LDAX  $0107  TEST ENTRY VALUE
791E D0 06                BNE   USR    IF NOT ZERO, THEN USER PAGE
7920 A5 01                LDA   FROM   +01 IF ZERO THEN CALC. FROM ROM PAGE
7922 29 E0                ANDIM $E0    FIND BOARD START ADDRESS
7924 09 0C                ORAIM $0C    DEFAULT RAM PAGE IS HERE
7926 85 03        USR     STA   TO     +01 RAM PAGE
7928 A5 04                LDA   OFFSET CALCULATE START OF
792A 18                   CLC          SWAP ROUTINE IN ROM
792B 69 05                ADCIM $05
792D 85 01                STA   FROM   +01
792F A9 00                LDAIM $00    CLEAR LOW BYTE OF ADDRESSES
7931 85 02                STA   TO
7933 85 00                STA   FROM

                  MOVE THE CODE TO ITS RUN SPACE AND THEN FIX UP
                  THE TABLES FOR RELOCATION

7935 A0 67                LDYIM SIZE   MOVE SWAP CODE
7937 B1 00        SSWAP   LDAIY FROM
7939 91 02                STAIY TO
793B 88                   DEY
793C 10 F9                BPL   SSWAP

793E A5 03        FSWAP   LDA   TO     +01 FIX UP THE RAM SWAP POINTERS
7940 A0 2A                LDYIM SWPA   WHICH PERMIT NON DESTRUCTIVE PAGE
7942 91 02                STAIY TO     ZERO SWAPPING
7944 A0 32                LDYIM SWPB   TWO POINTERS TO CHANGE
7946 91 02                STAIY TO

7948 A9 68        CTABLE  LDAIM CCSTR  INITIALIZE CONTROL CODE TRANSLATE
```

```
794A 85 02                STA   TO     TABLE TO ONE-TO-ONE MAP
794C A0 1F                LDYIM $1F
794E 98           CLOOP   TYA
794F 91 02                STAIY TO
7951 88                   DEY
7952 10 FA                BPL   CLOOP

7954 A9 98        JTABLE  LDAIM VSTR   CREATE JUMP VECTOR TABLE
7956 85 02                STA   TO     POINTER TO VECTOR AREA
7958 A9 00                LDAIM $00    RESET FROM POINTER
795A AA                   TAX          X = 00 FOR INDEXING
795B 85 00                STA   FROM   TO START OF ROM PAGE
795D A0 68                LDYIM TSTR   POINTER TO TABLE AREA IN ROM

795F A9 4C        JLOOP   LDAIM $4C    CREATE A JMP
7961 81 02                STAIX TO
7963 E6 02                INCZ  TO
7965 B1 00                LDAIY FROM   GET VECTOR FROM TABLE
7967 81 02                STAIX TO
7969 E6 02                INCZ  TO
796B C8                   INY
796C B1 00                LDAIY FROM   GET HIGH ADDRESS
796E 18                   CLC          CLEAR FOR FOLLOWING ADDITION
796F 65 04                ADC   OFFSET RELOCATE HIGH BYTE
7971 81 02                STAIX TO
7973 E6 02                INC   TO
7975 C8                   INY          NEXT TABLE ENTRY
7976 C0 94                CPYIM TEND
7978 D0 E5                BNE   JLOOP  NOT AT END

                  NOW PATCH UP VECTOR TABLE

797A A9 98                LDAIM VSTR
797C 85 02                STA   TO
797E A9 00                LDAIM $00
7980 A0 02                LDYIM $02    INDVEC +2
7982 91 02                STAIY TO     POINTS TO PAGE ZERO
7984 A5 03                LDA   TO     +01
7986 A0 05                LDYIM $05    RSTVEC +2
7988 91 02                STAIY TO     POINTS TO PGM RAM PAGE
798A A0 3E                LDYIM $3E    ZCCMAP +2
798C 91 02                STAIY TO     POINTS TO PGM RAM PAGE

                  SET ASK FLAG

798E A9 40                LDAIM $40    FOR KIM
7990 AE FD FF             LDX   $FFFD
7993 E0 8B                CPXIM $8B    SYM?
7995 D0 02                BNE   SETAK  NO
7997 A9 88                LDAIM $88    FOR SYM, AUTO LF OFF
7999 E0 E0        SETAK   CPXIM $E0    AIM?
799B D0 02                BNE   SETASK NO, MUST BE KIM
799D A9 00                LDAIM $00    FOR AIM
799F A2 8E        SETASK  LDXIM $8E    POINT TO ASK FLAG IN RAM PAGE
79A1 86 02                STX   TO
79A3 A2 00                LDXIM $00
79A5 81 02                STAIX TO
79A7 B8                   CLV
```

```
            RESTORE LOCATIONS 0000 - 0004

79A8 68     RLOOP   PLA             FROM STACK
79A9 95 00          STAZX $00
79AB E8             INX
79AC E0 05          CPXIM $05       TEST DONE
79AE D0 F8          BNE   RLOOP
79B0 68             PLA             RESTORE INITIAL VALUE IN A REG
79B1 60             RTS             RETURN TO USER


            VIDEO INITIALIZATION ROUTINE

79C0        TABLE   *     $0058     START OF VIDEO INIT TABLE IN RAM

79C0 C9 00  VINIT   CMPIM $00       TEST A = 00
79C2 D0 07          BNE   USER      USER SUPPLIED VIDEO INIT TABLE ADDR IN X,Y
79C4 A2 58          LDXIM TABLE     TABLE LOW ADDRESS
79C6 A5 15          LDA   RSTVEC    +02 RAM PAGE
79C8 A8             TAY
79C9 29 E0          ANDIM $E0       CALC. RAM DISPLAY START

79CB 85 0C  USER    STA   RAMPAG    SAVE RAM DISPLAY START
79CD 86 00          STX   CURSOR    SET TABLE POINTER LOW
79CF 84 01          STY   CURSOR    +01 AND HIGH IN POINTER
79D1 29 E0          ANDIM $E0
79D3 09 18          ORAIM $18       CALC. CRTC ADDRESS
79D5 85 03          STA   CRTREG    +01
79D7 A2 00          LDXIM $00       FIX LOW ADDRESSES
79D9 86 02          STX   CRTREG

79DB A0 12          LDYIM $12       SET BITS/CHAR FROM TABLE
79DD B1 02          LDAIY CRTREG    THROUGH VIA
79DF 09 0F          ORAIM $0F
79E1 91 02          STAIY CRTREG
79E3 A0 08          LDYIM $08       PICK UP MODES BYTE FROM VIDEO INIT TABLE
79E5 B1 00          LDAIY CURSOR
79E7 4A             LSRA
79E8 4A             LSRA            SHIFT OUT MODES BITS
79E9 85 0F          STA   TEMP
79EB A0 10          LDYIM $10       VIA OFFSET
79ED B1 02          LDAIY CRTREG    GET OLD DATA
79EF 29 F0          ANDIM $F0       CLEAR OLD VALUE
79F1 05 0F          ORA   TEMP      REPLACE WITH NEW
79F3 91 02          STAIY CRTREG

79F5 A0 1C          LDYIM $1C       ENABLE DISPLAY OF
79F7 B1 02          LDAIY CRTREG    CONTROL CODES, I.E. LINE
79F9 29 1F          ANDIM $1F       GRAPHICS, BY TURNING ON
79FB 09 C0          ORAIM $C0       CB2 ON VIA
79FD 91 02          STAIY CRTREG

            TEST TV OR MONITOR MODE

79FF A0 08          LDYIM $08
7A01 B1 02          LDAIY CRTREG    READ ONBOARD SWITCH
7A03 4A             LSRA            BIT 01 IS SET FOR TV MODE
7A04 90 04          BCC   INIT      AND CLEAR FOR MONITOR
7A06 A9 7F          LDAIM $7F       TV, SO SET OVERFLOW FOR
7A08 69 02          ADCIM $02       TESTING BELOW

7A0A A0 00  INIT    LDYIM $00       SET INDEXES
7A0C 84 05          STY   SCRLOW    +01
7A0E 98     INITA   TYA             NEXT REGISTER IN CRTC
7A0F 81 02          STAIX CRTREG
7A11 E6 02          INC   CRTREG    POINT TO REAL REGISTER
7A13 B1 00          LDAIY CURSOR    TABLE VALUE
7A15 50 01          BVC   INITB     TEST TV/MONITOR
7A17 4A             LSRA            TV, SO DIVIDE HORIZONTAL VALUES
7A18 C0 08  INITB   CPYIM $08       MODES?
7A1A D0 02          BNE   INITBB    NO
7A1C 29 03          ANDIM $03       YES, MASK OFF BITS/CHAR
7A1E 81 02  INITBB  STAIX CRTREG    STORE VALUE
7A20 C6 02          DEC   CRTREG    POINT TO DUMMY REGISTER
7A22 C0 01          CPYIM $01       CHARACTERS/LINE?
7A24 D0 04          BNE   INITC
7A26 85 0E          STA   COLMAX    YES, SAVE FOR LATER
7A28 85 04          STA   SCRLOW
7A2A C8     INITC   INY             BUMP INDEX
7A2B C0 04          CPYIM $04       TEST DONE WITH HORZ.
7A2D 30 DF          BMI   INITA     NO, MAINTAIN TV TEST
7A2F B8             CLV             YES, CLEAR TV TEST
7A30 C0 07          CPYIM $07       ROW MAX?
7A32 D0 02          BNE   INITD
7A34 85 07          STA   XTEMP     ROW MAX FOR COLROW
7A36 C0 10  INITD   CPYIM $10
7A38 D0 D4          BNE   INITA     SEND 16 BYTES TO CRTC

7A3A C6 07  COLROW  DEC   XTEMP     CALCULATE SCRLOW FROM
7A3C 18             CLC
7A3D F0 0C          BEQ   STVX      COLMAX * ROWMAX
7A3F A5 0E          LDA   COLMAX
7A41 65 04          ADC   SCRLOW
7A43 85 04          STA   SCRLOW
7A45 90 F3          BCC   COLROW
7A47 E6 05          INC   SCRLOW    +01
7A49 10 EF          BPL   COLROW    ALWAYS

7A4B A5 0C  STVX    LDA   RAMPAG
7A4D 65 05          ADC   SCRLOW    +01 ASSUME ENOUGH MEMORY
7A4F 85 0D          STA   RAMEND    FOR DISPLAY SPECIFIED IN TABLE

7A51 A2 0F          LDXIM OUTTV     LOW BYTE OF ENTRY POINT
7A53 A5 15          LDA   RSTVEC    +02 HIGH BYTE
7A55 24 06          BIT   ASK       AIM/SYM/KIM ?
7A57 70 10          BVS   INITK     KIM
7A59 10 19          BPL   INITE     AIM
7A5B 20 86 8B       JSR   ACCESS
7A5E 8E 64 A6       STX   OUTVEC    +01
7A61 8D 65 A6       STA   OUTVEC    +02
7A64 20 9C 8B       JSR   NACCES
```

```
7A67 30 1F               BMI   FINIS   ALWAYS

7A69 A0 01      INITK    LDYIM $01     STORE FOR KIM INIT
7A6B 91 E8               STAIY KOUT
7A6D 88                  DEY
7A6E 8A                  TXA
7A6F 91 E8               STAIY KOUT
7A71 B8                  CLV           CLEAR OVERFLOW SET BY BIT ASK
7A72 50 14               BVC   FINIS   ALWAYS

7A74 8E 06 A4   INITE    STX   DILINK  AIM TRANSFER VECTORS
7A77 8D 07 A4            STA   DILINK  +01

7A7A A2 19               LDXIM KBWAIT  LOW BYTE OF ENTRY POINT TO
7A7C 8E 08 01            STX   UIN     SET UP AIM INTERNAL KEYBOARD
7A7F 8D 09 01            STA   UIN     +01 SERVICE
7A82 A5 06               LDA   ASK
7A84 09 10               ORAIM $10     SET SPECIAL SERVICE BIT IN ASK FLAG
7A86 85 06               STA   ASK

7A88 A5 0D      FINIS    LDA   RAMEND  CLEAR LINE BELOW SCREEN SO FIRST
7A8A 85 01               STA   CURSOR  +01 SCROLL WILL NOT BRING UP JUNK
7A8C A5 04               LDA   SCRLOW
7A8E 85 00               STA   CURSOR  USE CURSOR BYTES AS TEMP POINTER
7A90 A4 0E               LDY   COLMAX
7A92 A9 00               LDAIM $00     CLEAR WITH NULLS
7A94 88         CLR      DEY
7A95 91 00               STAIY CURSOR
7A97 D0 FB               BNE   CLR

7A99 20 2B 00            JSR   ZHOMEC  NOW HOME THE CURSOR
7A9C B8                  CLV           AND RETURN VIA SWAP ROUTINE
7A9D 60                  RTS


                VIDEO OUTPUT ROUTINES

                AIM/SYM/KIM OUTPUT VECTORED HERE
                THROUGH SWAP ROUTINE ON RAM PAGE

                OUTPUT A CHARACTER TO TV

7AA0 48         VIDOUT   PHA           SAVE A
7AA1 B8                  CLV           CLEAR OVERFLOW
7AA2 86 07               STX   XTEMP
7AA4 84 08               STY   YTEMP
7AA6 A8                  TAY           SAVE CHARACTER
7AA7 A5 06      VIDX     LDA   ASK     TEST PCG FLAG, BIT 2
7AA9 4A                  LSRA
7AAA 4A                  LSRA
7AAB 98                  TYA           RESTORE CHARACTER
7AAC B0 03               BCS   CCTEST  IF BIT SET, DO NOT STRIP
7AAE 29 7F               ANDIM $7F     IF NOT SET, CONVERT CHARACTER
7AB0 A8                  TAY           UPDATE SAVED CHARACTER

                TEST FOR CONTROL CHARACTER

7AB1 C9 20      CCTEST   CMPIM $20
7AB3 B0 4D               BCS   CCHIGH  CHARACTER ABOVE 1F
7AB5 B1 4D               LDAIY ZCCMAP  +01 SUBSTITUTE CHAR. FROM MAP
7AB7 A8                  TAY           UPDATE SAVED CHARACTER
7AB8 B1 47               LDAIY ZCCVEC  +01 GET LOW BYTE OF DESTINATION
7ABA 85 4A               STAZ  ZCCXFR  +01 SET UP JUMP
7ABC A9 01               LDAIM $01     IN CASE OF BIT TOGGLE
7ABE 4C 49 00            JMP   ZCCXFR  AND GO TO CC PROCESSOR

                CONTROL CHARACTER PROCESSING

                SCROLL TO CURSOR

7AC1 A4 0C      SCROLL   LDY   RAMPAG  START SCROLL AT TOP LINE
7AC3 88                  DEY           MOVE CHARACTERS FROM OLD
7AC4 84 53               STY   OLD     +01 POSITION TO ONE LINE ABOVE
7AC6 A0 FF               LDYIM $FF
7AC8 84 52               STY   OLD

7ACA A5 53      NEXT     LDA   OLD     +01 POINTER TO OLD LINE
7ACC 85 55               STA   NEW     +01 BECOMES POINTER TO NEXT LINE DESTINATION
7ACE A5 52               LDA   OLD
7AD0 85 54               STA   NEW
7AD2 18                  CLC
7AD3 65 0E               ADC   COLMAX  AND NOW NEXT LINE IS OLD LINE
7AD5 85 52               STA   OLD
7AD7 A5 53               LDA   OLD     +01
7AD9 69 00               ADCIM $00
7ADB 85 53               STA   OLD     +01
7ADD C5 01      TEST     CMP   CURSOR  +01 HAS OLD PASSED CURSOR?
7ADF 90 08               BCC   MOVE    LESS, MOVE A LINE
7AE1 D0 14               BNE   BLANK   GREATER, CLEAR LAST LINE MOVED
7AE3 A5 52               LDA   OLD
7AE5 C5 00               CMP   CURSOR
7AE7 B0 0E               BCS   BLANK   CLEAR LAST LINE MOVED

7AE9 20 4F 00   MOVE     JSR   ZFLCKR  WAIT FOR VERTICAL RETRACE, THEN
7AEC A4 0E               LDY   COLMAX  MOVE DATA UP ONE LINE
7AEE B1 52      LOOP     LDAIY OLD     MOVE FROM OLD LINE
7AF0 91 54               STAIY NEW     TO NEW POSITION ONE LINE UP
7AF2 88                  DEY           WORK BACKWARDS FROM END
7AF3 D0 F9               BNE   LOOP
7AF5 F0 D3               BEQ   NEXT    ALWAYS

7AF7 A4 0E      BLANK    LDY   COLMAX  CLEAR LAST LINE MOVED
7AF9 A9 00               LDAIM $00     USE NULLS, NOT SPACES
7AFB 88         LOOPB    DEY
7AFC 91 00               STAIY CURSOR
7AFE D0 FB               BNE   LOOPB
7B00 F0 45               BEQ   CURUP   DONE, MOVE CURSOR UP

                ADDITIONAL CONTROL CHARACTERS

7B02 C9 5F      CCHIGH   CMPIM $5F
7B04 F0 0C               BEQ   SYMDEL  IS UNDERLINE
7B06 C9 7F               CMPIM $7F
7B08 D0 68               BNE   ENTCHR  IS NOT CONTROL CODE
```

```
                 HAVE A DELETE CHARACTER

7B0A A5 06                LDA   ASK
7B0C 29 40                ANDIM $40
7B0E F0 6C                BEQ   HOMA     AIM OR SYM
7B10 D0 08                BNE   KIMDEL   ALWAYS

7B12 A5 06       SYMDEL   LDA   ASK
7B14 10 5B                BPL   ENTCY    NOT A SYM
7B16 29 10                ANDIM $10      TEST RAE BIT
7B18 D0 57                BNE   ENTCY    RAE IGNORES THIS DELETE

7B1A A0 00       KIMDEL   LDYIM $00      USE NULL FOR DELETE
7B1C 20 43 00             JSR   ZSDEL
7B1F D0 5B                BNE   HOMA     ALWAYS

7B21 50 9E       SCRLX    BVC   SCROLL   VECTOR

                 MOVE CURSOR TO THE RIGHT

7B23 A9 C0       CURIGH   LDAIM $C0
7B25 25 06                AND   ASK      AIM?
7B27 D0 05                BNE   CURIGX   NO
7B29 A9 13                LDAIM $13      YES
7B2B 8D 15 A4             STA   CURPOZ
7B2E 20 40 00    CURIGX   JSR   ZSTCHX
7B31 50 42                BVC   ENTCHA

                 MOVE CURSOR TO THE LEFT

7B33 A9 C0       CURLEF   LDAIM $C0
7B35 25 06                AND   ASK
7B37 D0 05                BNE   CURLEX
7B39 A9 13                LDAIM $13
7B3B 8D 15 A4             STA   CURPOZ
7B3E C6 0B       CURLEX   DEC   CURPO
7B40 10 3A                BPL   HOMA     IF NO UNDERFLOW
7B42 A4 0E                LDY   COLMAX
7B44 88                   DEY
7B45 84 0B                STY   CURPO    ELSE SET CURPO TO END OF LINE AND

                 MOVE CURSOR UP

7B47 20 3A 00    CURUP    JSR   ZDECRE
7B4A 50 2B                BVC   ENTCHB   ALWAYS

                 CLEAR SCREEN

7B4C 20 2B 00    CLEAR    JSR   ZHOMEC   HOME CURSOR AND

                 CLEAR FROM CURSOR TO END OF SCREEN

7B4F A5 01       ERASER   LDA   CURSOR   +01
7B51 48                   PHA            SAVE CURSOR
7B52 A5 00                LDA   CURSOR
7B54 48                   PHA
7B55 A6 0B                LDX   CURPO
7B57 A9 00       SPACES   LDAIM $00      CLEAR WITH NULLS

7B59 20 37 00             JSR   ZSTCHY   STORE WITHOUT TESTS
7B5C D0 F9                BNE   SPACES   UNTIL END OF SCREEN
7B5E 86 0B                STX   CURPO
7B60 68                   PLA
7B61 85 00                STA   CURSOR   RESTORE CURSOR
7B63 68                   PLA
7B64 85 01                STA   CURSOR   +01
7B66 50 14                BVC   HOMA     ALWAYS

7B68 A5 06       SYMBRK   LDA   ASK      SYM?
7B6A 30 39                BMI   SETV     YES, SET BREAK FLAG
7B6C 10 0E                BPL   HOMA     ALWAYS

7B6E 68          DISPLA   PLA            USE CHARACTER SAVED ON
7B6F 48                   PHA            STACK FOR DISPLAY
7B70 A8                   TAY

7B71 98          ENTCY    TYA            RESTORE CHARACTER

                 ENTER A CHARACTER AND SCROLL IF NEEDED

7B72 20 2E 00    ENTCHR   JSR   ZSTORE   STORE CHARACTER
7B75 F0 AA       ENTCHA   BEQ   SCRLX    SCREEN OVERFLOW SO SCROLL
7B77 D0 03       ENTCHB   BNE   HOMA

                 HOME CURSOR

7B79 20 2B 00    HOME     JSR   ZHOMEC

                 TRANSFER CURSOR POINTERS TO CRT CONTROLLER

7B7C A0 00       HOMA     LDYIM $00
7B7E A9 0F                LDAIM $0F
7B80 91 02                STAIY CRTREG
7B82 C8                   INY
7B83 18                   CLC
7B84 A5 00                LDA   CURSOR
7B86 65 0B                ADC   CURPO
7B88 91 02                STAIY CRTREG
7B8A 88                   DEY
7B8B A9 0E                LDAIM $0E
7B8D 91 02                STAIY CRTREG
7B8F C8                   INY
7B90 A5 01                LDA   CURSOR   +01
7B92 69 00                ADCIM $00
7B94 29 1F                ANDIM $1F      BECAUSE CRTC HAS 16K ADDRESS SPACE
7B96 91 02                STAIY CRTREG
7B98 B8                   CLV            JUST IN CASE

7B99 A6 07       NULL     LDX   XTEMP    AND RESTORE REGISTERS
7B9B A4 08                LDY   YTEMP
7B9D 68                   PLA            RESTORE OUTPUT CHAR.
7B9E 85 09                STA   LCHAR    AND SAVE FOR CRLF
7BA0 60                   RTS            DONE WITH OUTPUT PROCESS
                 RETURN TO SWAP VIA ABOVE RTS

7BA1 A5 06       ESCAPE   LDA   ASK
7BA3 30 D7                BMI   HOMA     SYM IGNORES ESCAPE
```

```
7BA5 A9 7F      SETV   LDAIM $7F     ONE WAY TO SET V FLAG
7BA7 69 02             ADCIM $02
7BA9 70 EE             BVS   NULL    ALWAYS

               LINE FEED

7BAB A9 C0      LFEED  LDAIM $C0
7BAD 25 06             AND   ASK
7BAF F0 CB             BEQ   HOMA    IF AIM LF
7BB1 A9 08             LDAIM $08     TEST Q FLAG
7BB3 25 06             AND   ASK
7BB5 F0 06             BEQ   CURDN   NOT SET
7BB7 A5 09             LDA   LCHAR   TEST PRIOR CR
7BB9 C9 0D             CMPIM $0D     TO IGNORE EXTRA LF
7BBB F0 BF             BEQ   HOMA    IF SO, EXIT

               MOVE CURSOR DOWN

7BBD 20 34 00   CURDN  JSR   ZSTCHB
7BC0 50 B3             BVC   ENTCHA

               CARRIAGE RETURN

7BC2 A9 00      CRLFTV LDAIM $00
7BC4 A6 0B             LDX   CURPO
7BC6 85 0B             STA   CURPO
7BC8 A9 C0             LDAIM $C0
7BCA 25 06             AND   ASK
7BCC D0 EF             BNE   CURDN   NOT AN AIM
7BCE 85 0A             STA   CURPRM
7BD0 AC 15 A4          LDY   CURPOZ
7BD3 8D 15 A4          STA   CURPOZ
7BD6 68                PLA
7BD7 48                PHA           TEST FOR CRCK
7BD8 10 E3             BPL   CURDN
7BDA 98                TYA
7BDB F0 9F             BEQ   HOMA
7BDD 8A                TXA
7BDE D0 DD             BNE   CURDN
7BE0 F0 9A             BEQ   HOMA

               TOGGLE BITS IN ASK FLAG

7BE2 A6 06      RAE    LDX   ASK     SYM?
7BE4 10 96             BPL   HOMA    NO - DO NOT TOGGLE
7BE6 0A                ASLA          TOGGLE RAE MODE

7BE7 0A         NCRLF  ASLA          LF AFTER CR TOGGLE
7BE8 0A         FULL   ASLA          FULL/HALF DUPLEX TOGGLE
7BE9 0A         PCG    ASLA          PCG MODE TOGGLE
7BEA 45 06      ASCII  EOR   ASK     UPPER CASE ONLY TOGGLE
7BEC 85 06             STA   ASK
7BEE 50 8C      HOMAX  BVC   HOMA    ALWAYS

               CALL USER SUBROUTINE FOR CONTROL CHARACTER FUNCTION

7BF0 20 28 00   USR    JSR   ZUSR

7BF3 B8                CLV
7BF4 50 F8             BVC   HOMAX   ALWAYS

               TOGGLE FLICKER TEST

7BF6 A5 50      TFTST  LDA   ZFLCKR  +01 GET LOW BYTE OF TEST ADDR.
7BF8 49 08             EORIM $08     CHANGE ONE BIT
7BFA 85 50             STA   ZFLCKR  +01 NOW POINTS TO OTHER END OF ROUTINE
7BFC 50 F0             BVC   HOMAX

               TOGGLE CONTROL CHARACTER BLANKING

7BFE A0 1C      CCBT   LDYIM $1C
7C00 B1 02             LDAIY CRTREG
7C02 49 20             EORIM $20
7C04 91 02             STAIY CRTREG
7C06 50 E6             BVC   HOMAX


               VIDEO SUBROUTINES

               STORE CHARACTER IN A REGISTER VIA CURSOR AND
                INCREMENT CURSOR, TESTING FOR OVERFLOW

7C0B 24 06      STORE  BIT   ASK     AIM/SYM/KIM
7C0D 70 32             BVS   STCHAA  KIM AND
7C0F 30 31             BMI   STCHA   SYM SKIP AIM DELETE TEST

               FIRST CHECK IF DELETE WAS KEYED.
               IF SO, PSLS ROUTINE IN AIM MONITOR CALLED OUTPUT
               AND RETURN ADDRESS $E7F2 WILL BE ON STACK.

7C11 BA                TSX           GET STACK POINTER
7C12 BD 07 01          LDAX  $0107
7C15 C9 E7             CMPIM $E7
7C17 D0 07             BNE   STCHR   IT IS NOT A DELETE
7C19 BD 06 01          LDAX  $0106
7C1C C9 F2             CMPIM $F2
7C1E F0 52             BEQ   DELETE  YES, DELETE ONE CHAR

               STORE CHARACTER IN RAM AND MAINTAIN CURSOR

7C20 98         STCHR  TYA           CHAR WAS SAVED IN Y

               KEEP CURPOZ LESS THAN 20 TO SEE DELETES

7C21 AC 15 A4          LDY   CURPOZ
7C24 C0 13             CPYIM $13
7C26 B0 04             BCS   XXX
7C28 84 0B             STY   CURPO   OTHER CURSORS FOLLOW CURPOZ
7C2A 84 0A             STY   CURPRM
7C2C C8         XXX    INY
7C2D C0 14             CPYIM $14     IS CURPOZ LESS THAN 20?
7C2F 90 02             BCC   XXA     YES, INCR CURPOZ
7C31 A0 13             LDYIM $13     NO, RESET TO 19
7C33 8C 15 A4   XXA    STY   CURPOZ
```

```
                      MAINTAIN DISPLAY BUFFER FOR EDITOR

7C36 A4 0A                   LDY    CURPRM   PUT CHAR. IN DIBUFF ONLY IF
7C38 C0 3C                   CPYIM  $3C      CURPRM IS LESS THAN 60
7C3A B0 05                   BCS    STCHAA
7C3C 99 38 A4                STAY   DIBUFF   EDITOR & M-COMMAND USE THIS
7C3F E6 0A                   INC    CURPRM   BUFFER

7C41 B8           STCHAA     CLV             CLEAR FROM BIT ASK TEST
7C42 20 4F 00     STCHA      JSR    ZFLCKR   WAIT FOR VERTICAL RETRACE
7C45 A4 0B        STCHAY     LDY    CURPO
7C47 91 00                   STAIY  CURSOR   STORE INTO DISPLAY RAM
7C49 E6 0B        STCHAX     INC    CURPO
7C4B A5 0B                   LDA    CURPO
7C4D C5 0E                   CMP    COLMAX
7C4F 90 0F                   BCC    STCHD
7C51 A9 00                   LDAIM  $00
7C53 85 0B                   STA    CURPO
7C55 18           STCHB      CLC             MOVE CURSOR TO NEXT LINE
7C56 A5 00                   LDA    CURSOR
7C58 65 0E                   ADC    COLMAX
7C5A 85 00                   STA    CURSOR
7C5C 90 02                   BCC    STCHD
7C5E E6 01                   INC    CURSOR   +01
7C60 A5 01        STCHD      LDA    CURSOR   +01
7C62 C5 0D                   CMP    RAMEND   TEST FOR CURSOR BELOW SCREEN
7C64 90 0A                   BCC    STCHE
7C66 D0 06                   BNE    STCHZ
7C68 A5 00                   LDA    CURSOR
7C6A C5 04                   CMP    SCRLOW
7C6C 90 02                   BCC    STCHE
7C6E A9 00        STCHZ      LDAIM  $00      Z FLAG SET WILL CAUSE SCROLL
7C70 B8           STCHE      CLV
7C71 60                      RTS

7C72 A9 13        DELETE     LDAIM  $13      SET CURPOZ TO 19 TO SEE DELETES
7C74 8D 15 A4                STA    CURPOZ
7C77 C6 0A                   DEC    CURPRM   DECR OTHER POINTER
7C79 C6 0B        SDEL       DEC    CURPO    WRAP AROUND ZERO
7C7B 10 0E                   BPL    DELA
7C7D A5 0E                   LDA    COLMAX   RESET
7C7F 85 0B                   STA    CURPO
7C81 C6 0B                   DEC    CURPO    SET TO COLMAX - 1
7C83 20 3A 00                JSR    ZDECRE
7C86 D0 03                   BNE    DELA
7C88 20 2B 00                JSR    ZHOMEC
7C8B A5 06        DELA       LDA    ASK
7C8D 29 C0                   ANDIM  $C0      SYM OR KIM?
7C8F D0 11                   BNE    SDELA    ONE OF THE ABOVE
7C91 A9 E8                   LDAIM  $E8      STORE NEW RTN ADDRESS FOR OUTDP1 - AIM
7C93 9D 07 01                STAX   $0107
7C96 A9 04                   LDAIM  $04      RTN TO PSL00+3 WITH NEW POINTER
7C98 9D 06 01                STAX   $0106
7C9B A5 0A                   LDA    CURPRM   NEW POINTER TO SAVED ACC
7C9D 9D 03 01                STAX   $0103    REPLACE SAVED CHARACTER
7CA0 A0 00                   LDYIM  $00      USE NULL FOR DELETE
7CA2 98           SDELA      TYA
7CA3 20 4F 00                JSR    ZFLCKR   WAIT FOR RETRACE

7CA6 A4 0B                   LDY    CURPO
7CA8 91 00                   STAIY  CURSOR
7CAA C8                      INY             SET Z FLAG TO 1
7CAB 60                      RTS

7CAC 38           DECREM     SEC             MOVE CURSOR UP ONE LINE
7CAD A5 00                   LDA    CURSOR
7CAF E5 0E                   SBC    COLMAX
7CB1 85 00                   STA    CURSOR
7CB3 B0 02                   BCS    DECRB
7CB5 C6 01                   DEC    CURSOR   +01
7CB7 A5 0C        DECRB      LDA    RAMPAG   CURSOR ABOVE SCREEN?
7CB9 38                      SEC
7CBA E9 01                   SBCIM  $01      IF SO, CURSOR PAGE WILL BE RAMPAG - 1
7CBC C5 01                   CMP    CURSOR   +01
7CBE B8                      CLV
7CBF 60                      RTS             Z FLAG SET WILL HOME CURSOR

                  FLICKER SUPRESSION SUBROUTINE
7CC0 48           FLCKR      PHA             SAVE A
7CC1 A0 08                   LDYIM  $08      WAIT FOR VERTICAL
7CC3 B1 02        FLWAIT     LDAIY  CRTREG   RETRACE
7CC5 10 FC                   BPL    FLWAIT
7CC7 68                      PLA             RESTORE A
7CC8 60                      RTS

7CC9 FF                      =      $FF      PAD

                  HOME CURSOR

7CCA A9 00        HOMECU     LDAIM  $00      PUT CURSOR AT START
7CCC 85 00                   STA    CURSOR
7CCE 85 0B                   STA    CURPO
7CD0 A5 0C                   LDA    RAMPAG   OF DISPLAY
7CD2 85 01                   STA    CURSOR   +01
7CD4 60                      RTS


                  KEYBOARD ROUTINES

                  INITIALIZE BOTH VIDEO AND KEYBOARD

7CE0 20 1F 00     VKINIT     JSR    VVEC     INIT VIDEO FIRST

                  INITIALIZE KEYBOARD SERVICE

7CE3 A2 14        KBINIT     LDXIM  KBTST    LOW BYTE OF ENTRY POINT FOR KBTEST
7CE5 A5 15                   LDA    RSTVEC   +02 HIGH BYTE OF ALL ENTRY POINTS
7CE7 A4 06                   LDY    ASK      AIM/SYM/KIM?
7CE9 10 09                   BPL    KBDA     AIM OR KIM
7CEB 20 86 8B                JSR    ACCESS   SYM
7CEE 8E 67 A6                STX    INSVEC   +01
7CF1 8D 68 A6                STA    INSVEC   +02
```

```
                SET UP KEYBOARD VECTOR

7CF4 A2 19      KBDA   LDXIM KBWAT   LOW BYTE OF KBWAIT ENTRY
7CF6 24 06             BIT   ASK     AIM/SYM/KIM ?
7CF8 70 0D             BVS   KIMB    KIM
7CFA 10 16             BPL   KBDB    AIM
7CFC 8E 61 A6          STX   INVEC   +01 STORE LOW BYTE OF VECTOR
7CFF 8D 62 A6          STA   INVEC   +02 AND HIGH BYTE
7D02 20 9C 8B          JSR   NACCES
7D05 30 17             BMI   KBDC    ALWAYS

7D07 B8         KIMB   CLV
7D08 A0 01             LDYIM $01
7D0A 91 EA             STAIY KIN     STORE LOW BYTE OF VECTOR
7D0C 88                DEY
7D0D 8A                TXA
7D0E 91 EA             STAIY KIN     STORE HIGH BYTE
7D10 50 0C             BVC   KBDC    ALWAYS

7D12 8E 08 01   KBDB   STX   UIN     LOW VECTOR FOR AIM
7D15 8D 09 01          STA   UIN     +01 HIGH VECTOR
7D18 A5 06             LDA   ASK
7D1A 29 EF             ANDIM $EF     CLEAR BIT 10 TO TURN OFF
7D1C 85 06             STA   ASK     AIM INTERNAL KEYBOARD SERVICE

7D1E A0 1E      KBDC   LDYIM $1E     VIA IER OFFSET FROM CRTREG
7D20 A9 7F             LDAIM $7F     DISABLE ALL INTERRUPTS
7D22 91 02             STAIY CRTREG
7D24 88                DEY           VIA IFR OFFSET IS 1D
7D25 A9 FF             LDAIM $FF     CLEAR PENDING INTERRUPTS
7D27 91 02             STAIY CRTREG
7D29 88                DEY           VIA PCR OFFSET IS 1C
7D2A B1 02             LDAIY CRTREG
7D2C 29 FE             ANDIM $FE     LEAVE CB2, CB1, CA2 ALONE
7D2E 91 02             STAIY CRTREG  CA1 SET ON POSITIVE TRANSITION
7D30 88                DEY           VIA ACR OFFSET IS 1B
7D31 B1 02             LDAIY CRTREG
7D33 09 01             ORAIM $01     TURN BIT 01 ON AND
7D35 91 02             STAIY CRTREG  LEAVE THE REST ALONE
7D37 A0 08             LDYIM $08     TEST SW1
7D39 B1 02             LDAIY CRTREG
7D3B 29 04             ANDIM $04     KEYBOARD DATA INVERT SWITCH
7D3D F0 02             BEQ   KFINIS  SWITCH IS ON
7D3F A9 FF             LDAIM $FF     SWITCH OFF, SO INVERT DATA
7D41 85 3F      KFINIS STA   INVERT
7D43 B8                CLV
7D44 60                RTS           RETURN VIA SWAP ROUTINE

                TEST USED BY SYM MONITOR TO WAIT FOR BREAK KEY RELEASED

7D45 84 3D      KBTEST STY   YTEMPK  SAVE Y
7D47 18                CLC           C = 0 FOR NO DATA
7D48 A0 1D             LDYIM $1D     READ IFR
7D4A B1 02             LDAIY CRTREG
7D4C 29 02             ANDIM $02     MASK TO CA1 FLAG
7D4E 91 02             STAIY CRTREG  CLEAR FLAG IF SET
7D50 F0 01             BEQ   NODATA  IF ZERO, THEN NO DATA
7D52 38                SEC           ELSE, SET C = 1 FOR DATA
7D53 A4 3D      NODATA LDY   YTEMPK
7D55 60         KMDONE RTS

                MAIN INPUT ROUTINE
                WAIT FOR CHARACTER FROM KEYBOARD,
                ECHO, CONVERT LOWER TO UPPER CASE, ETC.

7D56 84 3D      KBWAIT STY   YTEMPK  SAVE Y
7D58 86 3E             STX   XTEMPK  SAVE X
7D5A A9 10      GETMOR LDAIM $10     AIM INTERNAL KBD SVC?
7D5C 24 06             BIT   ASK     AIM/SYM/KIM ?
7D5E 70 19             BVS   KBGET   KIM, SO KBGET
7D60 30 17             BMI   KBGET   SYM
7D62 90 F1             BCC   KMDONE  RETURN ON AIM INIT CALL
7D64 F0 13             BEQ   KBGET   AIM INTERNAL KBD SVC OFF

                AIM INTERNAL KEYBOARD HANDLER

7D66 20 40 EC   AIKH   JSR   GETKEY  CALL AIM MONITOR FOR KEY INPUT
7D69 E0 20             CPXIM $20     WAS SHIFT KEY DOWN?
7D6B B0 1B             BCS   KBGOT   YES, DO NOT SHIFT TO LOWER CASE
7D6D C9 41             CMPIM $41     WAS UPPER CASE ALPHABETIC?
7D6F 90 17             BCC   KBGOT   NO, DO NOT SHIFT
7D71 C9 5B             CMPIM $5B
7D73 B0 13             BCS   KBGOT
7D75 69 20             ADCIM $20     ADD TO SHIFT UPPER CASE TO LOWER
7D77 D0 0F             BNE   KBGOT   ALWAYS

                GET CHARACTER FROM EXTERNAL KEYBOARD VIA VIA

7D79 B8         KBGET  CLV           FROM BIT ASK
7D7A A0 1D             LDYIM $1D     TEST FOR DATA PRESENT
7D7C B1 02      KBWX   LDAIY CRTREG
7D7E 29 02             ANDIM $02
7D80 F0 FA             BEQ   KBWX    WAIT FOR DATA
7D82 A0 11             LDYIM $11     READ DATA
7D84 B1 02             LDAIY CRTREG
7D86 45 3F             EOR   INVERT  WILL INVERT IF BYTE WAS $FF

7D88 A8         KBGOT  TAY           SAVE CHARACTER
7D89 C9 1A             CMPIM $1A     CTRL Z ?
7D8B D0 0A             BNE   KNORM   NO
7D8D 20 4F 00   CTRLZ  JSR   ZFLCKR
7D90 A4 0B             LDY   CURPO   GET CHARACTER FROM CURRENT CURSOR
7D92 B1 00             LDAIY CURSOR  POSITION
7D94 A8                TAY           SAVE CHARACTER
7D95 50 10             BVC   NOTUP   SKIP CASE CONVERSION

7D97 A5 06      KNORM  LDA   ASK     TEST UPPER/LOWER ASCII FLAG
7D99 4A                LSRA
7D9A 98                TYA           RESTORE CHARACTER
7D9B B0 0A             BCS   NOTUP   IF SET, NOT UPPER ONLY
7D9D C9 61             CMPIM $61     UPPER CASE ONLY
7D9F 90 06             BCC   NOTUP   NOT LOWER CASE ALPHA
7DA1 C9 7B             CMPIM $7B
7DA3 B0 02             BCS   NOTUP
```

```
7DA5 29 DF                ANDIM $DF      CONVERT LOWER TO UPPER CASE

7DA7 A4 06       NOTUP    LDY   ASK      AIM/SYM/KIM ?
7DA9 10 06                BPL   AUOUT    AIM OR KIM
7DAB 2C 53 A6             BIT   TECHO    ECHO FLAG IN SYM
7DAE B8                   CLV            JUST IN CASE
7DAF 30 0C                BMI   ECHO     YES, ECHO BIT SET
7DB1 A8          AUOUT    TAY            SAVE CHARACTER
7DB2 C9 06                CMPIM $06      CONTROL F?
7DB4 F0 07                BEQ   ECHO     ALWAYS OUTPUT ECHO TOGGLE
7DB6 A9 04                LDAIM $04      TEST ECHO FLAG
7DB8 25 06                AND   ASK      TOGGLED BY CTRL F
7DBA F0 0D                BEQ   ALLTST   NO ECHO IF ZERO
7DBC 98                   TYA            RESTORE CHARACTER

7DBD 20 16 00    ECHO     JSR   OVEC
7DC0 50 01                BVC   EOK      TEST FOR BREAK RETURN
7DC2 60                   RTS            WAS A BREAK, DROP EVERYTHING
7DC3 C9 06       EOK      CMPIM $06      CONTROL F?
7DC5 38                   SEC            SO AIM WILL NOT THINK THIS IS INIT CALL
7DC6 F0 92                BEQ   GETMOR   IF CTRL F, DO NOT RETURN TO USER
7DC8 A8                   TAY            SAVE CHARACTER
7DC9 A6 06       ALLTST   LDX   ASK
7DCB 10 11                BPL   ALLOUT   AIM/KIM

7DCD A9 10       SYMOUT   LDAIM $10      TEST FOR SYM RAE SERVICE
7DCF 25 06                AND   ASK
7DD1 D0 0A                BNE   KRAE     SKIP IF RAE
7DD3 BA                   TSX            MODIFY RETURN TO AVOID
7DD4 BD 03 01             LDAX  $0103    AUTOMATIC CONVERSION TO UPPER CASE
7DD7 18                   CLC
7DD8 69 0C                ADCIM $0C
7DDA 9D 03 01             STAX  $0103
7DDD 38          KRAE     SEC            SET CARRY FLAG FOR CHARACTER

7DDE 98          ALLOUT   TYA            RESTORE CHARACTER
7DDF A6 3E                LDX   XTEMPK
7DE1 A4 3D                LDY   YTEMPK
7DE3 60                   RTS            RETURN TO USER VIA SWAP ROUTINE

                 RESTORE AIM OUTPUT TO LED DISPLAY

7DE4 A9 05       LEDS     LDAIM $05
7DE6 8D 06 A4             STA   $A406    DILINK
7DE9 A9 EF                LDAIM $EF
7DEB 8D 07 A4             STA   $A407
7DEE 00                   BRK            RETURN TO THE MONITOR
```

```
                 SWAP ROUTINE AND TABLES

                 THE FOLLOWING CODE WILL BE COPIED BY THE SETUP ROUTINE
                 INTO RAM FOR EXECUTION.  DO NOT EXECUTE THIS COPY.

7E00 48          VIDEO    PHA            VIDEO INIT ENTRY
7E01 A9 1F                LDAIM VVEC
7E03 D0 1C                BNE   SAVE
7E05 48          KBRD     PHA            KEYBOARD INIT ENTRY
7E06 A9 22                LDAIM KVEC
7E08 D0 17                BNE   SAVE
7E0A 48          IBOTH    PHA            VIDEO AND KEYBOARD INIT
7E0B A9 25                LDAIM VKVEC
7E0D D0 12                BNE   SAVE
7E0F 48          OUTTV    PHA            NORMAL CHARACTER OUTPUT
7E10 A9 16                LDAIM OVEC
7E12 D0 0D                BNE   SAVE
7E14 48          KBTEST   PHA            KEYBOARD TEST FOR DATA
7E15 A9 19                LDAIM KTVEC
7E17 D0 08                BNE   SAVE
7E19 48          KBWAIT   PHA            KEYBOARD WAIT FOR DATA
7E1A A9 1C                LDAIM KWVEC
7E1C D0 03                BNE   SAVE

7E1E 48          RSTORE   PHA            ENTRY TO RESTORE PAGE ZERO
7E1F A9 00                LDAIM $00      AND RETURN TO USER

                 THIS ROUTINE SWAPS PAGE ZERO AND THE
                 SPECIAL PCG RAM SAVE AREA

7E21 48          SAVE     PHA            SAVE VECTOR
7E22 8A                   TXA            SAVE X
7E23 48                   PHA
7E24 98                   TYA            SAVE Y
7E25 48                   PHA

7E26 A0 58       ASWAP    LDYIM SAVSIZ   PAGE ZERO LOCS TO SAVE
7E28 BE 88 FF    LOOP     LDXY  SAVSTR   THIS GETS MODIFIED
7E2B B9 00 00             LDAY  $0000    THIS IS REALLY 0000
7E2E 96 00                STXZY $0000
7E30 99 88 FF             STAY  SAVSTR   THIS GETS MODIFIED
7E33 88                   DEY
7E34 10 F2                BPL   LOOP     UNTIL Y = 0

7E36 68                   PLA            RESTORE REGISTERS
7E37 A8                   TAY
7E38 68                   PLA
7E39 AA                   TAX
7E3A 68                   PLA            MODE
7E3B D0 06                BNE   VECTOR
7E3D 68                   PLA            RESTORE A FOR RETURN
7E3E 50 02                BVC   NORML
7E40 B8                   CLV
7E41 00                   BRK
7E42 60          NORML    RTS

                 VECTORS HAVE BEEN SWAPPED INTO ZERO
```

```
7E43 85 11      VECTOR  STA   INDVEC  +01 SET POINTER TO VECTOR
7E45 68                 PLA           GET A REGISTER
7E46 85 0F              STA   TEMP    SETUP RETURN VIA RESTORE
7E48 A5 15              LDA   RSTVEC  +02
7E4A 48                 PHA
7E4B A5 14              LDA   RSTVEC  +01
7E4D 48                 PHA
7E4E A5 0F              LDA   TEMP
7E50 4C 10 00           JMP   INDVEC  GO TO DESIRED FUNCTION

7E53 00                 =     $00     PAD
7E54 00                 =     $00
7E55 00                 =     $00
7E56 00                 =     $00
7E57 00                 =     $00

                VIDEO INITIALIZATION TABLE
                WILL BE IN RAM AT PAGE ADDR +58
                FOR EASY USER ACCESS

7E58 7A         VIDTBL  =     $7A     H TOTAL
7E59 50                 =     $50     H DISPLAYED (80 CHARS.)
7E5A 60                 =     $60     H SYNC POSITION
7E5B 0A                 =     $0A     H SYNC WIDTH
7E5C 18                 =     $18     V TOTAL (24 LINES)
7E5D 18                 =     $18     V TOTAL ADJUST
7E5E 18                 =     $18     V DISPLAYED
7E5F 18                 =     $18     V SYNC POSITION
7E60 20                 =     $20     NON-INTERLACE & 8 BITS/CHARACTER
7E61 0B                 =     $0B     SCAN LINES/CHARACTER
7E62 6B                 =     $6B     CURSOR RASTER START & SLOW FLASH
7E63 0B                 =     $0B     CURSOR RASTER END
7E64 00                 =     $00     START ADDRESS HIGH BYTE
7E65 00                 =     $00     START ADDRESS LOW BYTE
7E66 00                 =     $00     CURSOR HIGH
7E67 00                 =     $00     CURSOR LOW

                VECTOR GENERATING TABLE
                USED BY SETUP ROUTINE TO GENERATE PAGE ZERO
                VECTORS INTO ASK I/O ROUTINES

7E68 00         VEGTBL  =     $00     INDVEC
7E69 00                 =     $00     SET BY SWAP ROUTINE
7E6A 1D                 =     $1D     RSTVEC
7E6B 00                 =     $00     ADDR OF RSTORE-1

7E6C A0                 =     $A0     OVEC
7E6D 01                 =     $01     POINTS TO VIDOUT
7E6E 45                 =     $45     KTVEC
7E6F 04                 =     $04     POINTS TO KBTEST
7E70 56                 =     $56     KWVEC
7E71 04                 =     $04     POINTS TO KBWAIT
7E72 C0                 =     $C0     VVEC
7E73 00                 =     $00     POINTS TO VINIT
7E74 E3                 =     $E3     KVEC
7E75 03                 =     $03     POINTS TO KBINIT
7E76 E0                 =     $E0     VKVEC
7E77 03                 =     $03     POINTS TO VKINIT
7E78 A0                 =     $A0     ZUSR
7E79 02                 =     $02     POINTS TO AN RTS
7E7A CA                 =     $CA     ZHOMEC
7E7B 03                 =     $03     POINTS TO HOMECU
7E7C 0B                 =     $0B     ZSTORE
7E7D 03                 =     $03     POINTS TO STORE
7E7E 42                 =     $42     ZSTCHA
7E7F 03                 =     $03     POINTS TO STCHA
7E80 55                 =     $55     ZSTCHB
7E81 03                 =     $03     POINTS TO STCHB
7E82 45                 =     $45     ZSTCHY
7E83 03                 =     $03     POINTS TO STCHAY
7E84 AC                 =     $AC     ZDECRE
7E85 03                 =     $03     POINTS TO DECREM
7E86 00                 =     $00     TEMP
7E87 00                 =     $00     TEMP
7E88 49                 =     $49     ZSTCHX
7E89 03                 =     $03     POINTS TO STCHAX
7E8A 79                 =     $79     ZSDEL
7E8B 03                 =     $03     POINTS TO SDEL
7E8C 98                 =     $98     ZCCVEC
7E8D 05                 =     $05     POINTS TO CCVECT TABLE
7E8E 00                 =     $00     ZCCXFR
7E8F 02                 =     $02     CONTROL CHARACTER TRANSFER VECTOR
7E90 68                 =     $68     ZCCMAP
7E91 00                 =     $00     POINTS TO CCMAP IN RAM
7E92 C0                 =     $C0     ZFLCKR
7E93 03                 =     $03     POINTS TO FLICKR

                END OF VECTOR GENERATING TABLE

7E94 00                 =     $00     PAD
7E95 00                 =     $00
7E96 00                 =     $00
7E97 00                 =     $00

                CONTROL CHARACTER VECTOR TABLE
                CONTAINS LOW BYTE OF ROUTINE ADDRESS,
                HIGH BYTE IS THIRD ROM PAGE ADDR.

7E98 99         CCVECT  =     $99     NULL
7E99 EA                 =     $EA     ASCII
7E9A 68                 =     $68     SYMBRK
7E9B 7C                 =     $7C     HOMA (C)
7E9C BD                 =     $BD     CURDN
7E9D 4F                 =     $4F     ERASER
7E9E E8                 =     $E8     FULL
7E9F 7C                 =     $7C     HOMA (G)
7EA0 79                 =     $79     HOME
7EA1 7C                 =     $7C     HOMA (!)
7EA2 AB                 =     $AB     LFEED
7EA3 1A                 =     $1A     KIMDEL
7EA4 33                 =     $33     CURLEF
7EA5 C2                 =     $C2     CRLFTV
7EA6 FE                 =     $FE     CCBT
7EA7 7C                 =     $7C     HOMA (C)
```

```
7EA8 E9          =    $E9     PCG
7EA9 E7          =    $E7     NCRLF
7EAA 23          =    $23     CURIGH
7EAB 21          =    $21     SCRLX
7EAC F6          =    $F6     TFTST
7EAD 47          =    $47     CURUP
7EAE E2          =    $E2     RAE
7EAF F0          =    $F0     USR
7EB0 4C          =    $4C     CLEAR
7EB1 6E          =    $6E     DISPLA
7EB2 7C          =    $7C     HOMA (Z)
7EB3 A1          =    $A1     ESCAPE
7EB4 23          =    $23     CURIGH
7EB5 7C          =    $7C     HOMA
7EB6 71          =    $71     ENTCY
7EB7 71          =    $71     ENTCY

            NOTE: WHEN ANY CONTROL CODE IS MAPPED TO 1E OR 1F,
                  THAT CODE WILL START/STOP BLANKING AS THOUGH 1E OR 1F
                  WAS SENT.
                  WHEN ANY CONTROL CODE IS MAPPED TO 19 (CTRL Y), THE
                  ORIGINAL CHARACTER WILL BE STORED IN THE DISPLAY.

            END OF CCVECTOR TABLE
```

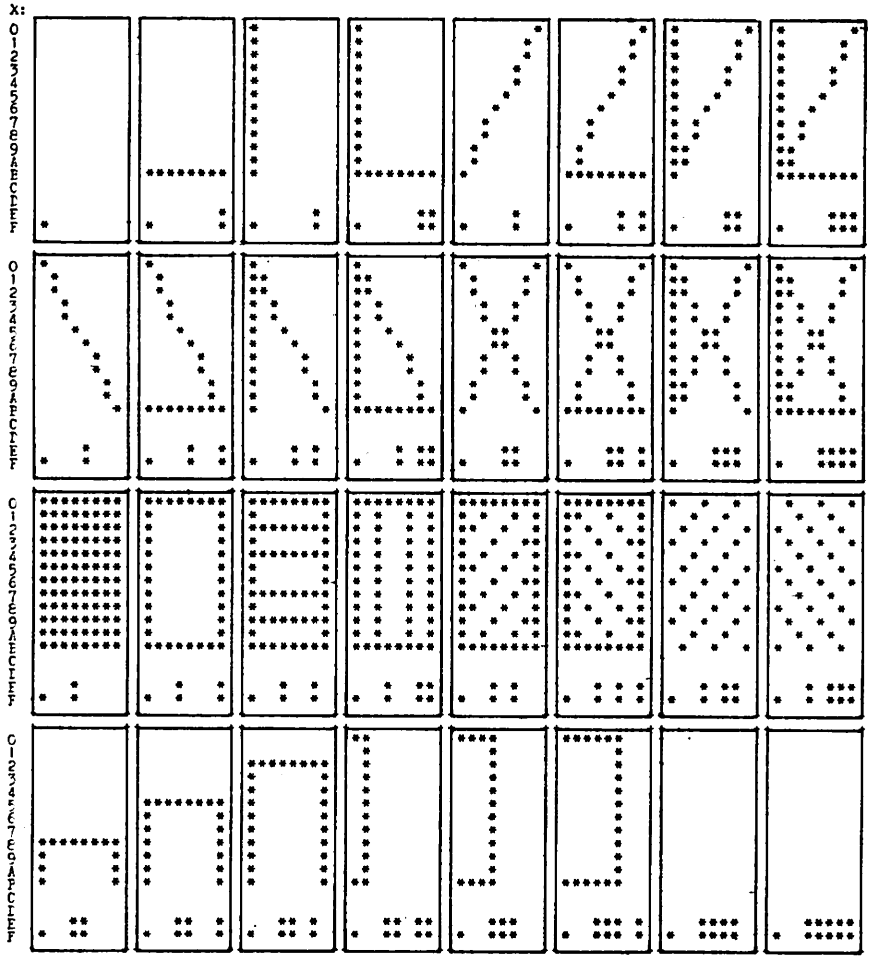
B-20

```
            STARTUPS

            START HERE
            TO INIT BOTH VIDEO AND KEYBOARD
            KIM USERS MAY JUMP HERE ONLY IF BREAK VECTOR
            HAS BEEN SET AND KIM INIT VECTORS AT
            00E8 - 00EB HAVE BEEN SET

7EB8 A9 00  GO     LDAIM $00
7EBA 48            PHA
7EBB 28            PLP          CLEAR PROCESSOR STATUS
7EBC A2 02         LDXIM $02
7EBE B5 00  SAVE   LDAX  $0000  SAVE LOCATIONS 0000 - 0002
7EC0 48            PHA          ON STACK
7EC1 CA            DEX
7EC2 10 FA         BPL   SAVE

7EC4 A9 60         LDAIM $60    PUT AN RTS
7EC6 85 00         STA   $0000  IN LOCATION 0000
7EC8 20 00 00      JSR   $0000  AND GO THERE TO
7ECB BA            TSX          PUT ADDRESS OF THIS CODE ON STACK
7ECC CA            DEX
7ECD 9A            TXS
7ECE 68            PLA          PULL PAGE ADDRESS OF THIS CODE
7ECF 29 E0         ANDIM $E0    GET BASE ADDRESS OF BOARD
7ED1 09 19         ORAIM $19    GET ADDRESS OF SETUP ROUTINE
7ED3 85 02         STA   $0002  CREATE A JUMP TO SETUP
7ED5 A9 4C         LDAIM $4C
7ED7 85 00         STA   $0000
7ED9 A9 00         LDAIM $00
7EDB 85 01         STA   $0001
7EDD 20 00 00      JSR   $0000  DO THE SETUP
7EE0 A5 02         LDA   $0002  NOW CREATE JUMP
7EE2 29 E0         ANDIM $E0    TO THE PGM RAM PAGE
7EE4 09 0C         ORAIM $0C
7EE6 85 02         STA   $0002

            CHANGE THE FOLLOWING TO LDAIM $00 TO INIT ONLY VIDEO

7EE8 A9 0A         LDAIM $0A    OFFSET TO INIT BOTH
7EEA 85 01         STA   $0001  VIDEO AND KEYBOARD
7EEC A9 00         LDAIM $00    FOR DEFAULT DISPLAY RAM
7EEE 20 00 00      JSR   $0000  DO THE INITIALIZATION
7EF1 A2 00         LDXIM $00
7EF3 68     RESTOR PLA          RESTORE LOCATIONS 0000 - 0002
7EF4 95 00         STAX  $0000
7EF6 E8            INX
7EF7 E0 03         CPXIM $03
7EF9 D0 F8         BNE   RESTOR
7EFB 00            BRK          DONE
```

Character Generator EPROM

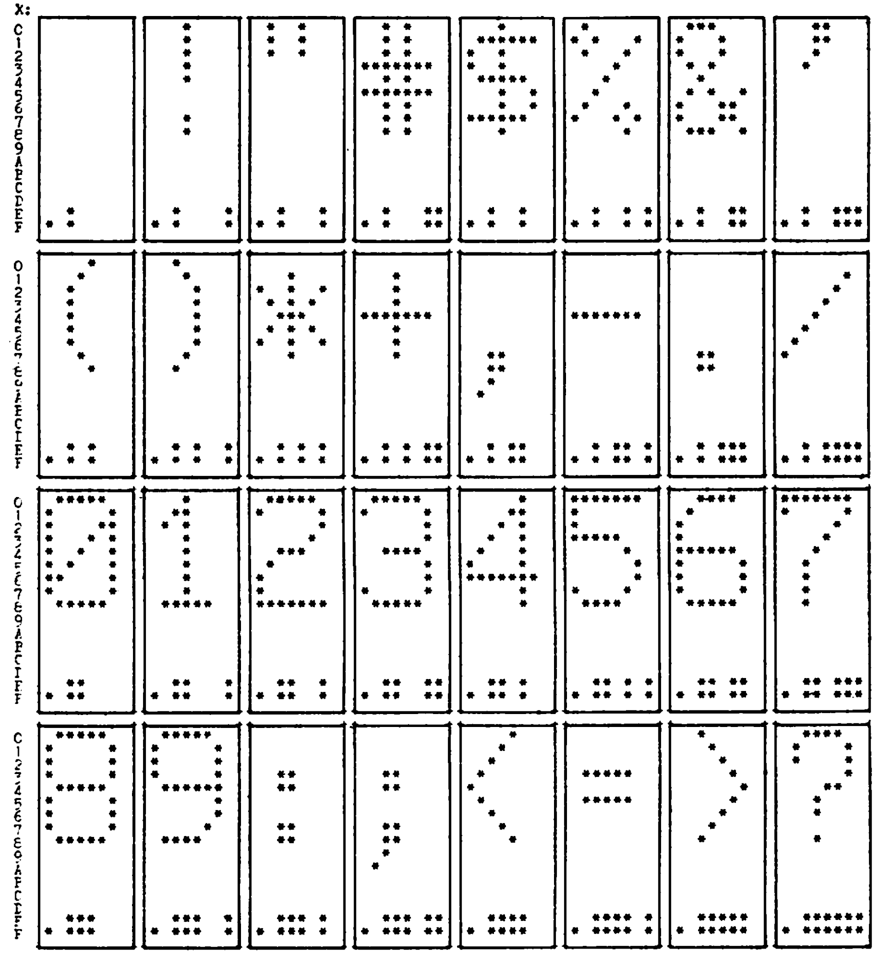
EPROM ADDRESS:

00X	01X	02X	03X	04X	05X	06X	07X
08X	09X	0AX	0BX	0CX	0DX	0EX	0FX
10X	11X	12X	13X	14X	15X	16X	17X
18X	19X	1AX	1BX	1CX	1DX	1EX	1FX

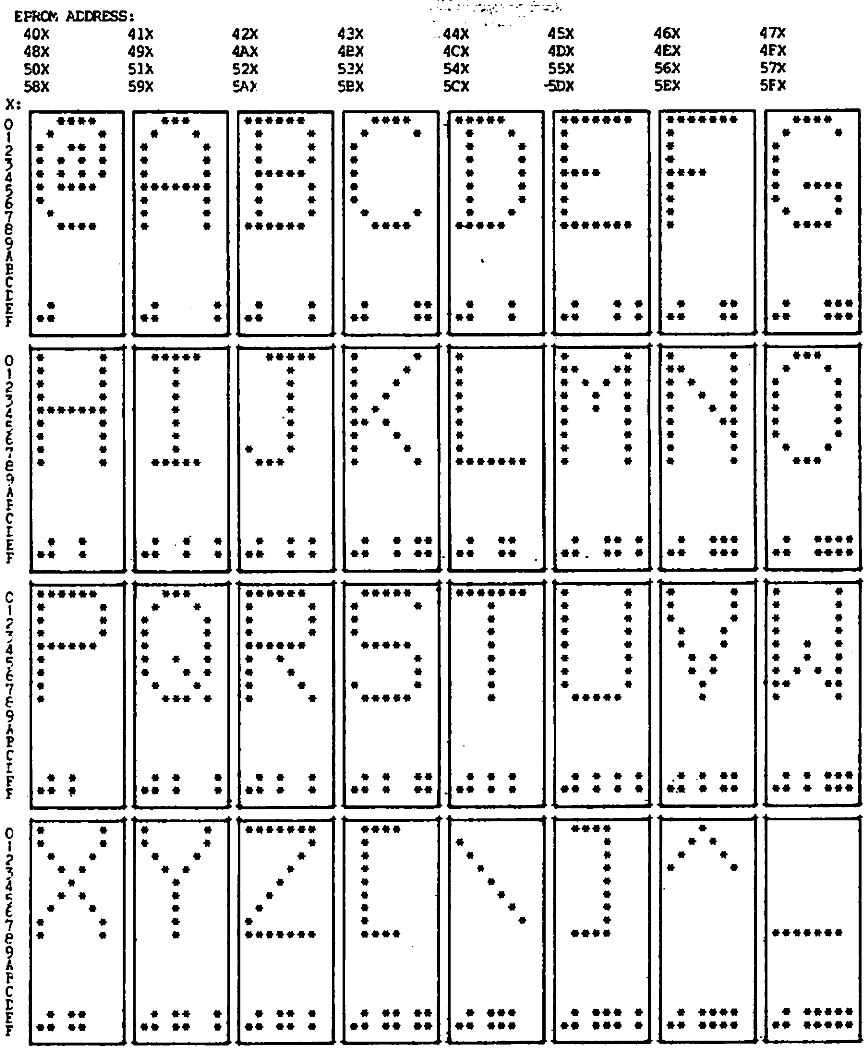


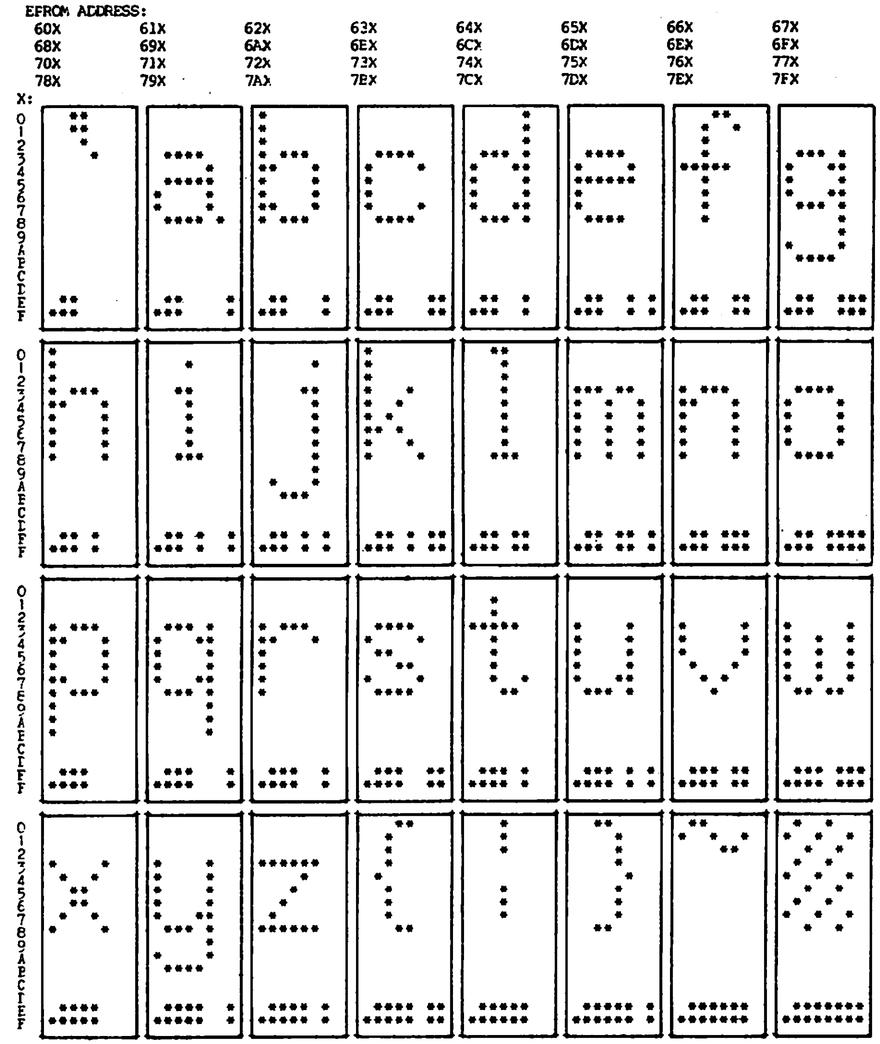
EPROM ADDRESS:

20X	21X	22X	23X	24X	25X	26X	27X
28X	29X	2AX	2BX	2CX	2DX	2EX	2FX
30X	31X	32X	33X	34X	35X	36X	37X
38X	39X	3AX	3BX	3CX	3DX	3EX	3FX



Character Generator EPROM





Video Plus II Stand-Alone Option

Introduction

The Video Plus II Stand-Alone option converts the Video Plus II into a complete single-board microcomputer with the following features:

6502 Microprocessor, the very popular CPU chip found in the APPLE, PET, AIM, SYM, KIM, OSI, etc., Up to 7K bytes of RAM of which up to 4K may be used for display memory and 2K used for program memory or Programmable Character Generator RAM, 2K ROM containing system software; includes monitor, screen editor, etc.

The Stand-Alone option may be installed at the factory or purchased separately as a kit of parts to be installed in the field. Complete installation instructions follow this introduction.

To complete a minimum stand-alone configuration, the following additional components are required:

a decoded ASCII keyboard with parallel data output and strobe,
a video monitor or TV set and RF modulator and
a power supply capable of providing at least +5 volts at 2 Amps.

Interfacing to peripherals is simplified by VIA ports on the Video Plus II. The VIA provides parallel-serial and serial-parallel conversion, timers and interrupt-driven I/O capability. In addition, the ACIA Communications Option provides RS 232 or 20 mA port. See Appendix F for details.

The Stand-Alone option package includes the following components:

one SPDT switch,
one 74LS161 divide-by-16 counter,
one 6502 microprocessor chip.

The switch is used to reset the 6502. It is connected to an RS flip-flop which acts to debounce the switch contacts. The reset switch is necessary to allow the user to stop the processor and bring it to a known state. The divide-by-16 counter converts the 16 MHz clock which drives the CRT controller on the Video Plus II to a 1 MHz clock for the 6502 CPU.

The 6502 microprocessor drives the address, data and control lines and executes the programming in the ROM. Detailed documentation for the 6502 is available from many sources including Rockwell International, Synertek and MOS Technology.

Installation Procedure

(If the stand-alone option was installed at the factory, then these installation instructions may be skipped.)

To install the stand-alone option, plug in the 74LS161 at U6, near the upper left corner of the Video Plus II. Plug in the 6502 at U1, in the lower left corner of the board. Be sure pin 1 of each IC is in the lower left corner. Install the reset switch at SW5, between the voltage regulator heat sinks and the DIP switches. The area of the board marked SW5 includes five holes. The pair nearest the top edge of the board are both grounded and are provided to allow some switches to be fastened securely to the board. Before soldering the switch, an etched lead must be cut on the back of the board between the center pad and the outer pad of the switch nearest the DIP switches (see figure 1). This lead was provided to connect one side of the debounce flip-flop to ground and leave the $\overline{RESET}$ line at logic 'one.' When this lead is cut and the switch is installed, the debounce flip-flop will pull the $\overline{RESET}$ line to logic 'zero' when the reset switch is on and to logic 'one' when the switch is off.

The SPDT switch must be conected with the wiper to the center pad on the Video board and the two poles connected to the outer pads. If a momentary-contact switch is used (and this is recommended) the normally-closed pole should be connected to the outer pad nearest the DIP switches and the normally-open pole should be connected to the outer pad near the voltage regulator heat sinks.

Stand-Alone Option Configuration

The Video Plus II is configured as follows to run stand-alone. The main 8K address space of the board, containing the display RAM, PCG RAM, CRTC, VIA and program EPROM, is located at $E000 to $FFFF. This is done to put the system reset vectors in on-board ROM at the top of memory where the CPU requires them. This board address is selected by turning SW3 position 8 ON and the rest of SW3 OFF.

The processor requires RAM at addresses $0000 to $01FF for page zero and page one (stack). The 1K program RAM provided with the Video Plus II must therefore be addressed at $0000. This is done as follows:

1. Turn SW1 position 7 ON to enable the 1K program RAM to be addressed outside the 8K space of the Video board.

2. Move the jumper from W11 to W10 to switch the fourth 1K of on-board address space from program RAM to display RAM.

3. Install a jumper from W1 to the adjacent pin, under position 1 of SW3, to select $0000 as the address of the off-board 1K RAM (see figure 2).

If you are using the ACIA communications option, refer to Appendix F at this time for installation and configuration instructions.

Stand-Alone System Memory Map

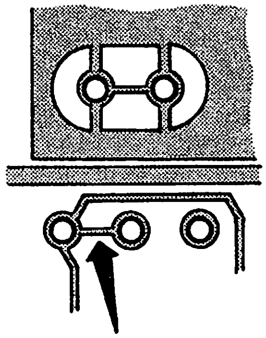
When the Video Plus II is running stand-alone with no other boards connected to it, the system memory map looks like this:

Address	Contents	
0000-00FF	Page Zero:	0000-0057 - Used by ASK software 0058-00DE - Free RAM 00DF-00FF - Used by stand-alone software
0100-01FF	Stack page	
0200-02FF	Free RAM	
0300-0387	ASK software program RAM:	0300-031D - Entry points to ASK Video Software 031E-0352 - Swapping routine 0358-0367 - Video Initialization table 0368-0387 - Control code map
0388-03FF	Free RAM	
0400-0FFF	Three images of addresses 0000-03FF, not usable	
1000-DFFF	Unused address space available for RAM/ROM expansion	
E000-EFFF	Display RAM (2, 3 or 4K)	
F000-F7FF	PCG RAM (1 or 2K)	
F800,F801	CRT Controller registers	
F804	SW2 Option switches	
F808	SW1 Basic control switches	
F80C-F80F	ACIA registers, if implemented	
F810-F81F	VIA Registers	

F820-FFFF Program EPROM:

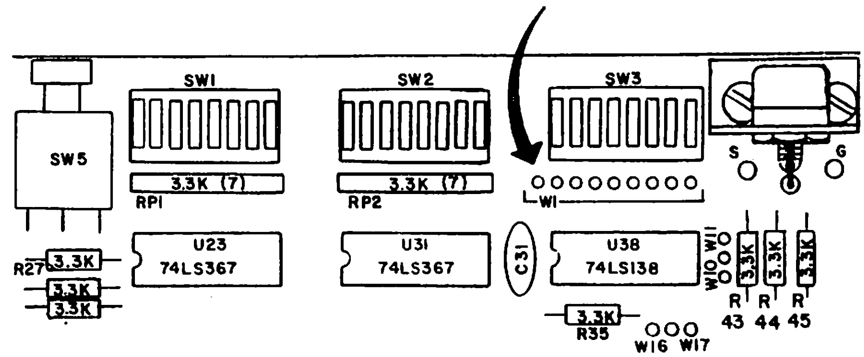
F820-F8F2	System Monitor program
F8F3	NMIX (Vector through NMIVEC at 00FA)
F8F6	IRQX (Vector through IRQVEC at 00FC)
F900-FEFF	Ask Video Software
FF00-FF60	System Monitor program (RESET)
FF09	COLD start
FF31	WARM start
FF3C	IRQH (System IRQ handler)
FF47	Monitor start from editor
FF61-FF8E	ACIA Subroutines:
FF61	AINIT (ACIA initialization)
FF75	XMIT (Transmit subroutine)
FF83	RCVR (Receive subroutine)
FF8F-FFBC	Screen Editor (EDITOR to start ACIA)
FF92	LOOP (Restart screen editor)
FFBD-FFF9	UPDATE output subroutine for System Monitor
FFFA	NMIV (Vector to NMIX)
FFFC	RSTV (Vector to RESET)
FFFE	IRQV (Vector to IRQX)

Figure 1



Cut etch here.

Figure 2



Stand-Alone Operation

With the stand-alone option (and the ACIA option, if it is being used) installed and configured, the power supply, display and keyboard should now be connected. +5 volts at 2 Amps should be connected at Application connector pin A-A, with ground at pin A-1. The display should be plugged into the RCA connector at the top of the board, or connected to the Application connector at pins A-B and A-C where A-B is the signal and A-C is ground. SW1 position 1 should be turned ON to select 80-character (monitor) mode, OFF for 40-character (TV) mode. The keyboard may be connected by means of the 12-pin MOLEX connector, J1, or via the Application connector (see page 13 for connector pin-outs). Set SW1 positions 2 and 3 according to whether your keyboard has positive or negative data and strobe signals (see also page 11 under Keyboard Configuration Control).

SW1 #2 ON for positive strobe
OFF for negative strobe

SW1 #3 ON for positive data
OFF for negative data

When the power is first switched on, no characters will appear on the display. The system must first be reset by turning SW5 ON and then OFF. The system will not run with the reset switch ON. Also, SW2 position 8 must be OFF to select a cold start. When the reset switch has been turned ON and then OFF, the display should come up with random characters and the cursor should be at the upper left corner of the display. If this display does not appear, then check over your installation and configuration switches and jumpers.

Cold vs. Warm Start

The first reset operation after power-up must always be a cold start in order to set up the RAM portion of the ASK Video software at locations $0300 to $0387. In addition to this set-up, the cold start initializes the second level (system) IRQ vector at $00FC-00FD to point to IRQH (see program listing or memory map for address) for break instruction processing, resets the stack pointer to $FF and then falls through to the warm start. A warm start on reset may be selected by turning SW2, position 8 ON.The warm start initializes the video and keyboard service and transfers control to the screen editor at LOOP (see program listing or memory map for address). A warm start avoids changing the video initialization parameters and system IRQ vector at $00FC-00FD, but it does not save the registers, status or stack pointer.

Screen Editor

After reset, the Video Plus II is in Screen Editor mode. At this time, all of the functions described in the ASK software documentation (Appendix B) are available as they apply to the KIM (the stand-alone board looks like a KIM to the ASK software).

The display may be cleared with a control X, upper and lower case characters are available if your keyboard generates them, the cursor may be moved up, down, left, right and home. When the cursor reaches the bottom of the screen, the display scrolls automatically. The screen editor may be used to produce and edit a screen full of text. The screen editor is a short (43 byte) machine-code loop which waits for characters from the keyboard and sends them to the ASK software output routine (OUTTV). All functions available in the screen editor are performed by the ASK software. The Editor also recognizes one additional control character, control C, which starts the system monitor.

The screen editor loop is also used by the ACIA communications option. The editor loop checks the ACIA transmitter and receiver if the ACIA has been set up. The editor loop does not call the ASK software keyboard wait routine (KBWAIT), but checks the keyboard flag directly, so the Full/Half duplex mode switch and the upper-case-only toggle do not function (these are keyboard routine functions).

System Monitor

When a control C is typed while in the screen editor, control transfers to the System Monitor. This utility program allows the user to examine and modify locations in memory and to transfer control to a user program. For instance, the monitor is used to set up and start the communications software.

The monitor is also called in response to a Break instruction. When the monitor is called, it borrows eight character positions from the display, starting at the current cursor position. The characters are stored on page zero and restored to the display when the monitor transfers control to another program. The eight saved characters are replaced by the following display: a four-character hexadecimal address, known as the "open address," followed by a less-than sign and a two-character hex data field and a space. For example: 034F < 4C.

The two-character data field displays the contents of the open address. Any hexadecimal characters typed at this point will be shifted into the address field and the data field will display the contents of the new open address. This allows any memory location to be examined.

The user may modify the contents of the open address (assuming the address is in RAM, not ROM) by typing RETURN to switch to the data input mode. The less-than sign will be replaced by a greater-than sign to indicate data input instead of address input mode. (For example: 034F > 4C).

The RETURN key may be used at any time to switch between these two modes. In the data input mode, all hex characters typed will be shifted into the data field of the display as well as into the contents of the open address. The data field of the display is continually updated from memory and always shows the actual contents of the memory location (as on the KIM). Try looking at the timers in the VIA at F814-F817. Do *not* try to look at the keyboard data register in the VIA (F811) because this will cause the system to hang up. (All characters typed will appear on the display in the data field but will not be recognized by the monitor program. The only remedy is to RESET the system.) Any number of hex digits may be shifted through the open location. To advance to the next location, type SPACE while in either address or data mode. To open the previous location, type "." (period).

Manual vs. Increment Mode

The system monitor provides two methods for entering data into memory. In Manual mode, when the data field is open for modification as indicated by the greater-than sign, all hex characters are shifted into the data field and the space bar may be used to increment the address field. In Increment mode, when entering data, the address field increments automatically after every second hex character typed. This mode is useful for entering data into consecutive addresses. The period and space are available in either mode confirming data or skipping a location. The Manual mode is selected by typing M at any time, and Increment mode is selected by typing I.

Go Command

The Go command is used to transfer control to a user program after restoring the display and registers. This command transfers to the open address displayed on the screen. To start a user program, type the starting address of the program into the address field and type G.

The registers saved by the Break interrupt handler are restored by the G command. If the G command is used to start rather than restart a user program, then these saved registers should first be initialized. Location $00F2 is loaded into the A register, $00F3 into the X register and $00F4 into the Y register, location $00F5 is loaded into the Processor Status register, and stack pointer is loaded from location $00F6.

PC Command

The system monitor provides a way to transfer the saved PC into the address field of the display. The PC is saved when a Break instruction is processed and also when the system monitor is called from the screen editor. The P command transfers the saved PC to the address field. If the monitor was called from the editor, the restart address of the editor is saved. To restart the editor from the monitor, type P to display the restart address, and G to transfer control to the open address. To restart a program after a Break instruction, type P to display the saved PC. This address is two greater than the address of the Break instruction. At this point you may type "." (period) twice to open the address of the Break instruction, restore the original instruction, and type G to resume program execution.

System Monitor Summary

Command	Function
(hex characters)	Address or data input
SPACE	Increment address field
. (period)	Decrement address field
RETURN	Switch between address and data input modes
I	Increment mode (increment address after data input)
M	Manual mode (no automatic address increment)
P	Transfer contents of saved PC to open address
G	Transfer control to open address after restoring saved registers and display

Page Zero Address	Contents		
00DF-00E7	Stand-alone monitor variables		
00E8-00EF	Bytes saved from display		
00F0-00F6	Registers saved by Break:	00F0	PC low byte
		00F1	PC high byte
		00F2	A register
		00F3	X register
		00F4	Y register
		00F5	Status register
		00F6	Stack pointer (unless ACIA in use)
00F6-00F9	ACIA initialization parameters:	00F6	Pointer to ACIA page (zero when ACIA in use)
		00F7	Page address of ACIA
		00F8	ACIA control byte
		00F9	ACIA command byte
00FA-00FF	Interrupt vectors:	00FA	NMI vector
		00FC	System IRQ vector
		00FE	User IRQ vector

Stand-Alone Interrupt Processing

The 6502 microprocessor supports two levels of interrupt processing. The lower-priority interrupt is called IRQ and is triggered by a low level on the IRQ input line to the 6502. This causes the processor to transfer control via the first level IRQ vector at $FFFE. In the stand-alone configuration, this vector points to IRQX (see program listing or memory map for address) which contains a Jump Indirect via $00FC-00FD to IRQH which is the system IRQ interrupt handler. The code at IRQH first checks for a Break instruction, which uses the same vector. If a Break caused the transfer, then the system monitor is called. If an IRQ caused the interrupt, the control is transferred via the User IRQ vector at $00FE-00FF. The user must set up one of these two page zero vectors to point to his interrupt processing routine. If the System IRQ vector at $00FC is used, then the user routine must also handle the Break instruction processing. Locations $00FC-00FD are initialized by the stand-alone software to point to IRQH on a cold start but not on a warm start. If the user program is not going to process Break instructions, then the pointer to the IRQ handler should be stored at $00FE-00FF. The user must set up one of these vectors in order to use IRQ interrupts. The IRQ interrupt must also be enabled by means of a bit in the Processor Status register. Refer to the 6502 programming manual for more information.

NMI Interrupts

The 6502 also provides a high-level or priority interrupt. This is the NMI, or Non-Maskable Interrupt. When an IRQ interrupt occurs, the IRQ line is masked and no further IRQs can take place until the IRQ disable bit in the status register is cleared. The NMI has no mask and therefore it can interrupt the code which is processing an IRQ. The NMI is also triggered by a low level on an input line to the 6502, in this case the NMI line.

NMI causes the processor to transfer control via the level one NMI vector at $FFFA-FFFB, to NMIX. NMIX is a Jump Indirect instruction which transfers via $00FA-00FB to the user's NMI handler. $00FA-00FB is not set up by the stand-alone software and must be initialized by the user for NMI processing. A single-step mode is available (see application note) which uses the NMI, as is done on the KIM.

Break Instruction

The Break instruction also causes the processor to transfer control via the IRQ vector at $FFFE, effectively simulating an IRQ condition except that the Break cannot be masked or disabled. The section above on IRQ processing describes the beginning of Break processing.

When the system monitor determines that a Break instruction caused the interrupt, the A, X and Y registers are saved in page zero, along with the stack pointer and the processor status register. The PC counter which points back to the Break instruction is also saved. These saved values are restored by a monitor G command, except the saved PC, which must be restored by the P command.

Example of Display Parameter Modification

It is possible to change the format of the display by following this procedure while operating stand-alone. The data specified is taken from Appendix B, page B-6. In outline, this procedure modifies the video initialization table in RAM and uses a warm start on reset to cause these parameters to be displayed.

Detailed procedure:

1. Configure the Video Board as described above for cold start, with SW2 position 8 OFF.
2. Turn the reset switch, SW5, ON and then OFF.
3. Type control X to clear the screen.
4. Type control C to start the system monitor.
   The display will show a random four-digit hex address and the contents of that address: XXXX < XX.
5. Enter address '0358' which is the address of the Video Init table in RAM. Display: 0358 < 7A.
6. Type RETURN to start data input, 0358 > 7A.
7. Enter data '60', 0358 > 60.
8. Type SPACE to increment the address, 0359 > 50.
9. Enter data '48', 0359 > 48.
10. Type SPACE to increment the address, 035A > 60.
11. Enter data '4C', 035A > 4C.
12. Type two SPACEs to increment the address twice, 035C > 18.
13. At this point you may want to type 'I' to switch to increment mode and avoid typing a space after every other digit.
14. Enter data '14141414' in Increment mode or '14 14 14 14' in Manual mode, 0360 > 20.
15. Enter data '180D6D0D' in Increment mode or '18 0D 6D 02' in Manual mode, 0364 > 00.
16. The new Video Init table has now been entered. Turn SW2 position 8 ON for warm start.
17. Turn the reset switch, SW5, ON and then OFF.
18. The display will now be organized as 20 lines of 72 characters instead of 25 lines of 80 characters and there is now more space between characters.

```
VIDEO PLUS II STAND-ALONE
SYSTEM MONITOR AND SCREEN EDITOR
INCLUDING ACIA SUBROUTINES
BY ROBERT M. TRIPP

06 JANUARY 1981
REVISION BY PAUL GEFFEN

COPYRIGHT (C) 1980
THE COMPUTERIST, INC.
P.O. BOX 3
S. CHELMSFORD, MA 01824
ALL RIGHTS RESERVED

PAGE ZERO LOCATIONS

COUNT    *    $00DF   CHARACTER COUNTER
WORK     *    $00E0   ADDR OF WORK AREA ON SCREEN
WORKH    *    $00E1
INMODE   *    $00E2   INPUT MODE
MODE     *    $00E3   ADDRESS/DATA MODE
ADRL     *    $00E4   LOW BYTE OF OPEN ADDRESS
ADRH     *    $00E5   HIGH BYTE OF OPEN ADDRESS
DATA     *    $00E6   CONTENTS OF OPEN ADDRESS
DATAX    *    $00E7   DATA SHIFTS INTO HERE

DSPSAV   *    $00E8   SAVE DISPLAY AREA

PCL      *    $00F0   LOW BYTE OF SAVED PC
PCH      *    $00F1   HIGH BYTE
AREG     *    $00F2   SAVED A REGISTER
XREG     *    $00F3
YREG     *    $00F4
FLAGS    *    $00F5   SAVED STATUS REGISTER
STACK    *    $00F6   SAVED STACK POINTER

ACIA     *    $00F6   POINTER TO ACIA PAGE, CLEARED BY AINIT
ACIAPG   *    $00F7   HIGH BYTE OF POINTER, SET BY USER
USERN    *    $00F8   ACIA CONTROL BYTE, SET BY USER
USERM    *    $00F9   ACIA COMMAND BYTE, SET BY USER

INTERRUPT VECTORS

NMIVEC   *    $00FA   NON-MASKABLE INTERRUPT
IRQVEC   *    $00FC   IRQ AND BREAK INTERRUPTS
USRVEC   *    $00FE   USER IRQ VECTOR

ASK VIDEO SOFTWARE VECTORS

VKINIT   *    $030A
OUTTV    *    $030F
KBWAIT   *    $0319
RSTORE   *    $031E
ASK      *    $0006   SINCE NO SWAPPING

DSPAGE   *    $E000   DISPLAY MEMORY PAGE

                CRTC     *      $F800    CRT CONTROLLER REGISTER #
                CRTD     *      $F801    CRT CONTROLLER REGISTER DATA
                KBFLAG   *      $F81D    VIA IFR
                KBDATA   *      $F811    VIA PORT
                FTEST    *      $F808    SW1 BASIC CONTROL SWITCHES

                SWITCH   *      $F804    SW2 OPTION SWITCHES
                ACON     *      $F80F    ACIA CONTROL REG
                ACOM     *      $F80E    ACIA COMMAND REG
                ACST     *      $F80D    ACIA STATUS REG
                ACDA     *      $F80C    ACIA DATA REGISTER

                USRSET   *      $F902    USER INIT ENTRY

                         ORG    $FF00    NOT RELOCATABLE

FF00 A9 00      RESET    LDAIM  $00      CLEAR ALL FLAGS
FF02 48                  PHA
FF03 28                  PLP
FF04 AD 04 F8            LDA    SWITCH   TEST WARM/COLD START
FF07 10 28               BPL    WARM     SWITCH 2, POSITION 8 ON FOR WARM START

FF09 A2 3C      COLD     LDXIM  IRQH     RESET IRQ INTERRUPT VECTOR
FF0B 86 FC               STX    IRQVEC   ON COLD START
FF0D A2 FF               LDXIM  IRQH     /
FF0F 86 FD               STX    IRQVEC   +01
FF11 9A                  TXS             X = FF, RESET STACK
FF12 CA                  DEX             MAKE X = FE FOR DUMMIES
FF13 86 E9               STX    $00E9    MAKE ASK KIM VECTORS
FF15 86 EB               STX    $00EB    POINT TO ROM FOR SAFETY
FF17 A9 03               LDAIM  $03      USE PAGE 3 FOR SWAPPING
FF19 20 02 F9            JSR    USRSET   AND INIT ASK SOFTWARE

FF1C 20 1E 03            JSR    RSTORE   SET UP NON-SWAPPING MODE
FF1F A0 18               LDYIM  $18      FOR CLC
FF21 8C 21 03            STY    $0321
FF24 88                  DEY
FF25 8C 23 03            STY    $0323
FF28 A0 90               LDYIM  $90      FOR BCC
FF2A 8C 22 03            STY    $0322

FF2D A9 00               LDAIM  $00
FF2F 85 F8               STA    USERN    START IN LOCAL MODE

FF31 A0 03      WARM     LDYIM  $03      RE-INIT VIDEO ON WARM START
FF33 A9 E0               LDAIM  DSPAGE   / OR RESET
FF35 A2 58               LDXIM  $58      POINTER TO TABLE
FF37 20 0A 03            JSR    VKINIT
FF3A 50 56               BVC    LOOP

FF3C 85 F2      IRQH     STA    AREG     SAVE REGISTERS ON INTERRUPT
FF3E 68                  PLA             GET STATUS FROM STACK
FF3F 48                  PHA             AND RESTORE ON STACK
FF40 29 10               ANDIM  $10      TEST BREAK OR IRQ
FF42 D0 0A               BNE    BREAK
FF44 6C FE 00            JMI    USRVEC   IRQ
```

```
FF47 A9 FF      MONTOR  LDAIM LOOP    / SET POINTER TO LOOP
FF49 48                 PHA           FOR EASY RETURN
FF4A A9 92              LDAIM LOOP
FF4C 48                 PHA
FF4D 08                 PHP           SAVE STATUS

FF4E 86 F3      BREAK   STX   XREG
FF50 84 F4              STY   YREG    SAVE REGISTERS AND
FF52 68                 PLA           STATUS FROM STACK
FF53 85 F5              STA   FLAGS
FF55 68                 PLA
FF56 85 F0              STA   PCL
FF58 68                 PLA
FF59 85 F1              STA   PCH
FF5B BA                 TSX
FF5C 86 F6              STX   STACK
FF5E 4C 20 F8           JMP   SERVCE  GO TO SERVICE ROUTINES AT $F820

                ACIA INITIALIZATION SUBROUTINE
                $00F7-00F9 MUST BE SET UP BEFORE CALL TO AINIT
                $00F6 SET TO ZERO BY AINIT

FF61 A9 00      AINIT   LDAIM $00     CLEAR LOW BYTE OF POINTER
FF63 85 F6              STA   ACIA
FF65 A0 0F              LDYIM $0F
FF67 A5 F8              LDA   USERN
FF69 91 F6              STAIY ACIA    SEND CONTROL BYTE TO ACIA
FF6B 88                 DEY
FF6C A5 F9              LDA   USERM
FF6E 91 F6              STAIY ACIA    SEND COMMAND BYTE TO ACIA
FF70 88                 DEY
FF71 88                 DEY
FF72 B1 F6              LDAIY ACIA    READ DATA TO START RECEIVER
FF74 60                 RTS

                ACIA TRANSMIT SUBROUTINE
                WAITS FOR TRANSMITTER TO CLEAR

FF75 AA         XMIT    TAX           SAVE DATA
FF76 A0 0D              LDYIM $0D
FF78 B1 F6      WAIT    LDAIY ACIA    READ FLAG
FF7A 29 10              ANDIM $10
FF7C F0 FA              BEQ   WAIT    WAIT FOR FLAG
FF7E 8A                 TXA
FF7F 88                 DEY
FF80 91 F6              STAIY ACIA
FF82 60                 RTS

                ACIA RECEIVE SUBROUTINE
                RETURNS WITH Z FLAG SET IF RECEIVER BUSY,
                Z FLAG CLEAR IF DATA WAS RECEIVED.

FF83 A0 0D      RCVR    LDYIM $0D
FF85 B1 F6              LDAIY ACIA    Basis, Y   im ZB T00D = Statusreg.
FF87 29 08              ANDIM $08
FF89 F0 03              BEQ   RRTN
FF8B 88                 DEY
FF8C B1 F6              LDAIY ACIA    GET DATA
FF8E 60         RRTN    RTS
```

```
                MAIN COMMUNICATIONS LOOP
                INCLUDES LOCAL SCREEN EDITOR

FF8F 20 61 FF   EDITOR  JSR   AINIT   START ACIA

                RESTART HERE AND START HERE IF NOT USING ACIA

FF92 AD 1D F8   LOOP    LDA   KBFLAG  LOCAL KEYBOARD DATA?
FF95 29 02              ANDIM $02
FF97 F0 14              BEQ   BLOOP   NO, CHECK ACIA RECEIVER
FF99 AD 11 F8           LDA   KBDATA
FF9C C9 03              CMPIM $03     CONTROL C?
FF9E F0 A7              BEQ   MONTOR  YES, GO TO MONITOR

FFA0 A6 F8      ALOOP   LDX   USERN   ACIA SET UP?
FFA2 F0 14              BEQ   DLOOP   NO, LOCAL MODE
FFA4 20 75 FF           JSR   XMIT    SEND BYTE TO ACIA
FFA7 A5 06              LDA   ASK
FFA9 29 04              ANDIM $04     FULL DUPLEX?
FFAB D0 0A              BNE   CLOOP   NO, ECHO

FFAD A5 F8      BLOOP   LDA   USERN   ACIA SET UP?
FFAF F0 E1              BEQ   LOOP    NO, STAY LOCAL
FFB1 20 83 FF           JSR   RCVR    TEST ACIA RECEIVER
FFB4 F0 DC              BEQ   LOOP
FFB6 AA                 TAX

FFB7 8A         CLOOP   TXA
FFB8 20 0F 03   DLOOP   JSR   OUTTV   SEND TO SCREEN
FFBB 50 D5              BVC   LOOP    ALWAYS

                DISPLAY SUBROUTINE FOR SYSTEM MONITOR

FFBD A0 00      UPDATE  LDYIM $00     RESET POINTER
FFBF A5 E5              LDA   ADRH
FFC1 20 DE FF           JSR   HBYTE   OUTPUT HIGH BYTE
FFC4 A5 E4              LDA   ADRL
FFC6 20 DE FF           JSR   HBYTE   OUTPUT LOW BYTE
FFC9 A5 E3              LDA   MODE
FFCB 38                 SEC
FFCC E9 A8              SBCIM $A8
FFCE 20 F1 FF           JSR   OUTPUT
FFD1 A2 00              LDXIM $00     SET INDEX
FFD3 A1 E4              LDAIX ADRL    FETCH OPEN DATA
FFD5 85 E6              STA   DATA    AND SAVE
FFD7 20 DE FF           JSR   HBYTE
FFDA A9 20              LDAIM $20
FFDC D0 13              BNE   OUTPUT  SPACE AFTER DATA

FFDE 48         HBYTE   PHA           OUTPUT HEX BYTE
FFDF 4A                 LSRA
FFE0 4A                 LSRA
FFE1 4A                 LSRA
FFE2 4A                 LSRA
FFE3 20 E7 FF           JSR   HEXOUT  FIRST HALF
FFE6 68                 PLA           NOW DO SECOND HALF
```

```
FFE7 29 0F     HEXOUT  ANDIM $0F
FFE9 C9 0A             CMPIM $0A     CONVERT TO ASCII
FFEB 90 02             BCC   HEXX
FFED 69 06             ADCIM $06
FFEF 69 30     HEXX    ADCIM $30

FFF1 AE 08 F8  OUTPUT  LDX   FTEST
FFF4 10 FB             BPL   OUTPUT  WAIT FOR VERTICAL RETRACE
FFF6 91 E0             STAIY WORK
FFF8 C8                INY
FFF9 60                RTS

               FIRST LEVEL INTERRUPT VECTORS

FFFA F3        NMIV    =     $F3     NMIX
FFFB F8                =     $F8
FFFC 00        RSTV    =     $00     RESET
FFFD FF                =     $FF
FFFE F6        IRQV    =     $F6     IRQX
FFFF F8                =     $F8


               SYSTEM MONITOR SUBROUTINES

                       ORG   $F820   NOT RELOCATABLE

F820 A9 0F     SERVCE  LDAIM $0F     GET CURSOR ADDRESS
F822 8D 00 F8          STA   CRTC    LOW BYTE OF ADDRESS
F825 AD 01 F8          LDA   CRTD    FROM CRT CONTROLLER
F828 85 E0             STA   WORK
F82A A9 0E             LDAIM $0E
F82C 8D 00 F8          STA   CRTC    CURSOR ADDRESS
F82F AD 01 F8          LDA   CRTD    HIGH BYTE
F832 09 E0             ORAIM DSPAGE  / COMBINE WITH DISPLAY PAGE
F834 85 E1             STA   WORK    +01

F836 A0 07             LDYIM $07     SAVE WORK AREA
F838 B1 E0     SLOOP   LDAIY WORK
F83A 99 E8 00          STAY  DSPSAV  IN PAGE ZERO
F83D 88                DEY
F83E 10 F8             BPL   SLOOP

F840 C8                INY           Y=0
F841 98                TYA           CLEAR STATUS
F842 48                PHA
F843 28                PLP
F844 A9 E4             LDAIM ADRL    SET MODE TO ADDRESS INPUT
F846 85 E3             STA   MODE
F848 84 E2     SETMOD  STY   INMODE  SET MANUAL OR INCREMENT MODE

F84A A9 00     INLOOP  LDAIM $00     ASSUME ZERO COUNT
F84C A6 E3             LDX   MODE    TEST ADDRESS OR DATA MODE
F84E E0 E4             CPXIM ADRL
F850 F0 02             BEQ   SETCNT  ADDRESS FIELD
F852 A5 E2             LDA   INMODE  DATA FIELD, RESTORE INMODE
F854 85 DF     SETCNT  STA   COUNT   SET COUNT
```

```
F856 20 BD FF  LOOP    JSR   UPDATE  UPDATE DISPLAY
F859 AD 1D F8  KBTST   LDA   KBFLAG  TEST KEYBOARD
F85C 29 02             ANDIM $02
F85E F0 F6             BEQ   LOOP    NO DATA
F860 AD 11 F8          LDA   KBDATA  INPUT CHARACTER FROM VIA

F863 C9 30             CMPIM $30
F865 90 45             BCC   STST    MAYBE A SPACE
F867 C9 3A             CMPIM $3A
F869 90 5F             BCC   DIGIT   MUST BE A DIGIT
F86B 29 DF             ANDIM $DF     CONVERT TO UPPER CASE
F86D C9 41             CMPIM $41
F86F 90 E5             BCC   LOOP
F871 C9 47             CMPIM $47
F873 90 53             BCC   ATOF

F875 C9 49     ITST    CMPIM 'I      INSERT MODE?
F877 D0 04             BNE   MTST    NO
F879 A0 02             LDYIM $02     YES
F87B 10 CB             BPL   SETMOD

F87D C9 4D     MTST    CMPIM 'M      MANUAL MODE?
F87F D0 04             BNE   PTST    NO
F881 A0 00             LDYIM $00     YES, CLEAR AUTO INCREMENT MODE
F883 10 C3             BPL   SETMOD

F885 C9 50     PTST    CMPIM 'P      PUT SAVED PC INTO WORK ADDRESS?
F887 D0 08             BNE   GTST    NO
F889 A6 F0             LDX   PCL     YES
F88B 86 E4             STX   ADRL
F88D A6 F1             LDX   PCH
F88F 86 E5             STX   ADRH

F891 C9 47     GTST    CMPIM 'G      GO FROM OPEN ADDRESS?
F893 D0 17             BNE   STST    NO
F895 A0 07             LDYIM $07     YES, BUT FIRST

F897 B9 E8 00  RLOOP   LDAY  DSPSAV  RESTORE DISPLAY
F89A 91 E0             STAIY WORK
F89C 88                DEY
F89D 10 F8             BPL   RLOOP

F89F A5 F5     RESTOR  LDA   FLAGS   THEN RESTORE REGISTERS
F8A1 48                PHA
F8A2 A6 F3             LDX   XREG
F8A4 A4 F4             LDY   YREG
F8A6 A5 F2             LDA   AREG
F8A8 28                PLP           RESTORE SAVED STATUS
F8A9 6C E4 00          JMI   ADRL

F8AC C9 20     STST    CMPIM $20     SPACE
F8AE F0 3A             BEQ   INCADR

F8B0 C9 2E     PRTST   CMPIM '.      FOR PREVIOUS LOCATION
F8B2 D0 08             BNE   RTST    NO
F8B4 A6 E4     DECADR  LDX   ADRL    ALREADY ZERO?
F8B6 D0 02             BNE   DECA    NO
F8B8 C6 E5             DEC   ADRH    YES, SO FIX ADRH
F8BA C6 E4     DECA    DEC   ADRL    DECREMENT LOW
```

```
F8BC C9 0D     RTST    CMPIM $0D      RETURN?
F8BE D0 96     YLOOP   BNE   LOOP     NO, IGNORE
F8C0 A5 E3             LDA   MODE     SWITCH MODES
F8C2 49 02             EORIM $02      ADDR = $E4
F8C4 85 E3             STA   MODE     DATA = $E6
F8C6 30 28             BMI   XLOOP    ALWAYS

F8C8 69 09     ATOF    ADCIM $09      CHANGE LETTER TO HEX VALUE
F8CA 29 0F     DIGIT   ANDIM $0F      GET REAL VALUE
F8CC A0 03     SHIFT   LDYIM $03      SHIFT LEFT BY FOUR BITS
F8CE A6 E3             LDX   MODE
F8D0 16 00     SHIFTA  ASLX  $0000    SHIFT ZEROS IN
F8D2 36 01             ROLX  $0001
F8D4 88                DEY
F8D5 10 F9             BPL   SHIFTA
F8D7 15 00             ORAX  $0000    HEX VALUE INPUT ALREADY IN A
F8D9 95 00             STAX  $0000
F8DB E0 E4             CPXIM ADRL
F8DD F0 11             BEQ   XLOOP    ADDRESS MODE
F8DF C8                INY            TO ZERO
F8E0 91 E4             STAIY ADRL     STORE NEW DATA BYTE
F8E2 A4 DF             LDY   COUNT
F8E4 F0 0A             BEQ   XLOOP    WAS ZERO
F8E6 C6 DF             DEC   COUNT
F8E8 D0 D4             BNE   YLOOP    WAS TWO

F8EA E6 E4     INCADR  INC   ADRL     WAS ONE
F8EC D0 02             BNE   XLOOP
F8EE E6 E5             INC   ADRH
F8F0 4C 4A F8  XLOOP   JMP   INLOOP

               SECOND-LEVEL INTERRUPT VECTORS

F8F3 6C FA 00  NMIX    JMI   NMIVEC
F8F6 6C FC 00  IRQX    JMI   IRQVEC
```

Video Plus II ACIA Option

Introduction

Most computer systems, from single-board micros to the largest business machines, are able to transmit and receive data in one of two standard formats, known as RS 232 and 20 mA. The RS 232 standard specifies voltage and current levels as well as a serial encoding scheme for data bytes. Most data communication over telephone lines or radio is done according to the RS 232 standard and with a modem. "Modem" is an abbreviation for "modulator-demodulator" and a modem is necessary for all telephone and radio data communication.

The 20 mA standard is also known as 20 milliamp or TTY and became popular through the widespread use of Teletype® equipment.

In addition to its function as a display and keyboard driver for AIM, SYM and KIM systems, the Video Plus II can easily be expanded to include a 20 mA or RS 232 port. This port allows high speed communication with almost any other computer or peripheral device. It enables the video board to drive a printer, a remote terminal or another Video Plus II.

The 20 mA port allows the stand-alone video board to be connected to an AIM, SYM or KIM (or any other computer) by means of a standard TTY port, thus freeing the micro's CPU from spending time in display routines.

In addition to providing 20 mA and RS 232 compatible signals, the ACIA option also supports modem handshaking logic. The Video Plus II may be plugged directly into a modem and the modem may in turn be connected over telephone lines to almost any computer system in the world. The Video Plus II now provides access to virtually all timesharing networks and on-line data bases, thereby putting the user in touch with a tremendous range of information resources.

The Video Plus II ACIA option gives the Video Plus II user a sophisticated programmable communication capability, considerably more powerful than that found in most microcomputer TTY ports. At the heart of this option is a Synertek 6551 ACIA chip which allows the user to program the clock source, baud rate, number of start and stop bits, word length, type of parity check performed, turns the receiver and transmitter on and off enables or disables the interrupts and contains error and interrupt flags. The 6551 can simultaneously receive and transmit bytes of data.

In addition to the ACIA chip, the Video Plus II contains discrete circuitry to generate the proper voltage and current levels for either 20 mA or RS 232 communication and a crystal-controlled clock circuit to drive the ACIA.

Complete specifications and operating instructions for the 6551 ACIA can be found in the data sheets included with this option. Briefly, the ACIA provides programmable baud rates from 50 to 19.2K baud, an external clock option, word length selectable from 5 to 8 bits, odd or even parity, parity enable/disable, loop-back mode for self-test and modem handshaking lines.

The ACIA option may be installed at the factory or purchased separately as a kit of parts to be installed in the field. The ACIA option may be used with the stand-alone option or with AIM, SYM or KIM driving the Video Plus II as a peripheral board.

Set-up

The ACIA option package includes the following components:

- one 10-position DIP switch
- one 74LS04 Hex inverter
- one 6551 ACIA chip
- one 1.8432 MHz crystal
- one optional 25-pin RS 232 connector

All other circuitry required for communication, either RS 232 or 20 mA, is already installed on the Video Plus II.

Installation

Plug the DIP switch into socket SW4, in the top right corner of the board. Plug in the 74LS04 at U54, next to SW4 and the 6551 at U46, in the bottom right corner of the board. Be sure that pin 1 of each IC is in the lower left corner. Solder the crystal at Y2, below the 6551 at U46. Install the 25-pin connector (if it is used) at J2, above U54. In addition, two etch cuts must be made and four jumper wires installed to invert the phase of the TTY or 20 mA input and output signals. Cut the trace between pin 10 of SW4 and pin 12 of SW4, and the trace between pin 8 of SW4 and pin 14 of SW4. Now install four jumpers as follows:

from SW4 pin 10 to U54 pin 1
from SW4 pin 12 to U54 pin 2
from SW4 pin 14 to U54 pin 3
from SW4 pin 8 to U54 pin 4

Configuration

Configuring the ACIA option is done entirely by means of SW4. This 10-position DIP switch is organized as five pairs of switches and only one from each pair should be turned on at one time.

SW4 positions 7, 8, 9 and 10 determine whether 20 mA current loop or RS 232 signals are generated by the board. For 20 mA (TTY) operation, turn positions 7 and 9 ON and 8 and 10 OFF. RS 232 communication requires positions 8 and 10 ON and 7 and 9 OFF. SW4 positions 1 through 6 determine the low reference level of the input and output signals. For 20 mA, this level is usually ground and positions 2, 4 and 6 of SW4 should be ON, and positions 1, 3 and 5 should be OFF. When positions 1, 3 and 5 are ON and 2, 4 and 6 are OFF then the low reference level of the signal is obtained from the Application connector at pin A-X. This may be ground, -5 volts, -12 volts for various applications. SW4 may be seen as a DPDT switch (positions 7-10) and three SPDT switches (positions 1&2, 3&4, 5&6).

Standard 20 mA setting:	2	4	6	7	9	ON
	1	3	5	8	10	OFF
RS 232, ground reference:	2	4	6	8	10	ON
	1	3	5	7	9	OFF
RS 232, -12 v reference:	1	3	5	8	10	ON
	2	4	6	7	9	OFF, -12 volts supplied at A-X.

The externally supplied low reference level need not be switched to both input and output sides of the RS 232 or 20 mA line.

The ACIA option may be connected to a modem by means of J2 and a standard cable. The ACIA chip provides all necessary handshaking signals. A 20 mA terminal may also be connected to J2 at pins 12, 13, 24 and 25, as follows. Pin 12 is the keyboard input signal and pin 13 is the keyboard input return which may be switched to ground or a supplied negative reference level. Pin 25 of J2 is the printer output and pin 24 is the printer return which is pulled up to +5 volts through 150 ohms. These signals are also brought out to the Application connector as follows:

J2 pin	Application Connector	Function
12	A-T	TTY Keyboard Input
13	A-U	TTY Keyboard Return
24	A-R	TTY Printer Return
25	A-S	TTY Printer Output

The RS 232 Printer output is at J2, pin 3 and Application connector pin A-W. RS 232 Keyboard input is on J2 pin 2 and Application connector A-V. RS 232 return is through ground on pin 1 and 7 of J2. In addition, modem handshaking lines are brought out to J2. Two of these lines must be grounded if no modem is connected, Data Carrier Detect and Data Set Ready, on J2 pins 6 and 8 and Application connector pins A-N and A-P.

J2 pin	Application Connector	Function
1	A-1	Ground
2	A-V	RS 232 Keyboard Input
3	A-W	RS 232 Printer Output
4	A-H	Modem $\overline{RTS}$ (Ready To Send)
5	A-J	Modem $\overline{CTS}$ (Clear To Send)
6	A-N	Modem $\overline{DSR}$ (Data Set Ready)
7	A-1	Ground
8	A-P	Modem $\overline{DCD}$ (Data Carrier Detect)
20	A-L	Modem $\overline{DTR}$ (Data Terminal Ready)

Communications Operation

The standard ASK Video Plus II software EPROM contains support programming for the communications option as part of the stand-alone programming. This software consists of the screen editor described in the stand-alone option notes and three ACIA support subroutines. The screen editor will support ACIA communications on a character-by character basis, but only when running stand-alone. The ACIA support routines may also be called by a user program and provide the core of a communications software package. We will consider the following cases:

1. Stand-Alone and ACIA implemented, smart terminal application, no additional programming involved.

   The stand-alone Video Plus II equipped with the ACIA option is a complete terminal package by itself. The software provided with the Video Plus II allows character I/O in either full or half duplex mode. To use this mode, first start the stand-alone system monitor, as outlined in the stand-alone option description. From the monitor, initialize the following locations with the ACIA initialization parameters:

   $00F7 to $F8 to point to the ACIA page,

   $00F8 to the ACIA control byte ($13 for 100 baud, $16 for 300 baud, see data sheet for other baud rates and word lengths),

   $00F9 to the ACIA command byte ($0B for no parity, $6B for odd parity, see data sheet for other options),

   $0006 to $40 for full duplex, $44 for half duplex operation,

   Load the address of EDITOR (see program listing or memory map for address) and type G to start the editor. The ACIA will be initialized, all characters typed on the local keyboard will be sent via the ACIA, all characters received will be displayed. A control C from the local keyboard will return control to the system monitor and suspend I/O operations.

2. ACIA implemented on peripheral Video Plus II with AIM, SYM or KIM and minimal additional programming required.

   The ACIA support software provides three subroutines, for ACIA initialization, transmit and receive operation. These subroutines assume that location $00F7 contains the page address of the ACIA registers. Location $00F6 is cleared by the initialization subroutine AINIT. Locations $00F7 through $00F9 must be loaded with this page address and with the ACIA control and command bytes before the call to AINIT. The initialization subroutine loads the ACIA command and control registers from $00F8 and $00F9 and reads the ACIA data register to start the receiver. The user may call this subroutine or write a similar one. The ACIA transmit subroutine is labelled XMIT (see program listing or memory map for address). This routine waits for the transmit flag to clear, then transfers data from the A register to the ACIA data register and returns. The ACIA receive subroutine is labelled RCVR (see program listing or memory map for address). This routine returns with the Z flag set if the ACIA receiver was busy. If the receiver was not busy, then a byte of data is read from the ACIA into the A register and the Z flag is clear on return. The screen editor in the stand-alone software, starting at EDITOR, is an example of how these routines may be called.

   Please note that the screen editor supplied can not be used except when running stand-alone because it calls the ACIA subroutines in the last page of memory. If the Video Plus II is being used as a peripheral board for an AIM/SYM/KIM then this code will not be at this location, but will instead be in the last page of the 8K space of the video board ($7F00-$7FFF in the default case with the board addressed at $6000). Therefore, the user must supply a program similar to the screen editor which calls the ACIA support subroutines and the ASK video subroutines at the proper addresses.

ACIA Option Memory Map

Address	Contents		
0000-00FF	Page Zero:	0000-0057	May be used by ASK software
		0058-00DE	Free RAM
		00DF-00F6	May be used by Stand-Alone System Monitor
		00F7	Page address of ACIA for ACIA subroutines
		00F8	ACIA control byte for AINIT subroutine
		00F9	ACIA command byte for AINIT subroutine
		00FA-00FF	May be used by Stand-Alone System
0100-01FF	Stack page		
0200-02FF	Free RAM		
0300-038F	May be used as ASK software program RAM		
0388-03DF	May be used by ASK software for page zero swapping		

Address Relative to Video board base address

0000-0FFF	Display RAM		
1000-17FF	PCG RAM (may be used as program RAM)		
1800,1801	CRT Controller registers		
1804	SW2 Option Switches		
1808	SW1 Basic control switches		
180C-180F	ACIA Registers:	180C	Data register
		180D	Flags and status register
		180E	Command register
		180F	Control Register
1810-181F	VIA Registers		
1820-1FFF	ASK Software EPROM:	1820-18FF	Stand-Alone Software
		1900-1EFF	ASK Video Software
		1F00-1F8E	ACIA Subroutines:
		1F61	AINIT Initialization Subroutine
		1F75	XMIT Transmit Subroutine
		1F83	RCVR Receive Subroutine
		1F8F-1FFF	Stand-Alone Software