

APPLE II
SERIAL INTERFACE CARD (A2L0008)
INSTALLATION AND OPERATING MANUAL

Apple Computer Inc.
P-A2L0008/030-0012 Printed in U.S.A. 10/78-2500

**PLEASE READ THIS MANUAL BEFORE ATTEMPTING TO INSTALL
THE SERIAL INTERFACE CARD INTO THE APPLE II.**

**SERIAL INTERFACE MANUAL
INSTALLATION AND OPERATING MANUAL**

TABLE OF CONTENTS

SECTION	TITLE	PAGE
	INTRODUCTION	
I	INSTALLATION	3
	Installing the Serial Interface	
	Compatibility with External Devices	
	RS232 Connector Usage	
	Current-Loop Operation with a Teletype	
II	OPERATION	9
	Using the Serial Interface	
	Preliminary Discussion of IN# and PR#	
	Sending Output and Receiving Input	
III	DEFAULT PARAMETERS AND THE DIP SWITCH	15
	Initialization	
	Operating Parameters	
	Setting the DIP Switch Defaults	
	Permanent Defaults	
IV	ACCESS TO OPERATING PARAMETERS	19
	Description of Serial Interface Operation	
	Transmission Data Sequence	
	Lower-Case Characters	
	Changing Interface Parameters through Software Commands	
V	DIRECT USE OF THE INTERFACE	27
	Transmitting a Character without Using PR#1	
	Batch Moves	
VI	APPENDIX: SERIAL INTERFACE TIMING	31
	Table of Baud Rate Quantum Numbers	
	Circuit Diagram	
	Assembly Listing	

APPLE II SERIAL INTERFACE CARD

INTRODUCTION

These are the fundamental abilities of the APPLE Serial Interface, using the nearly universal RS232 standard:

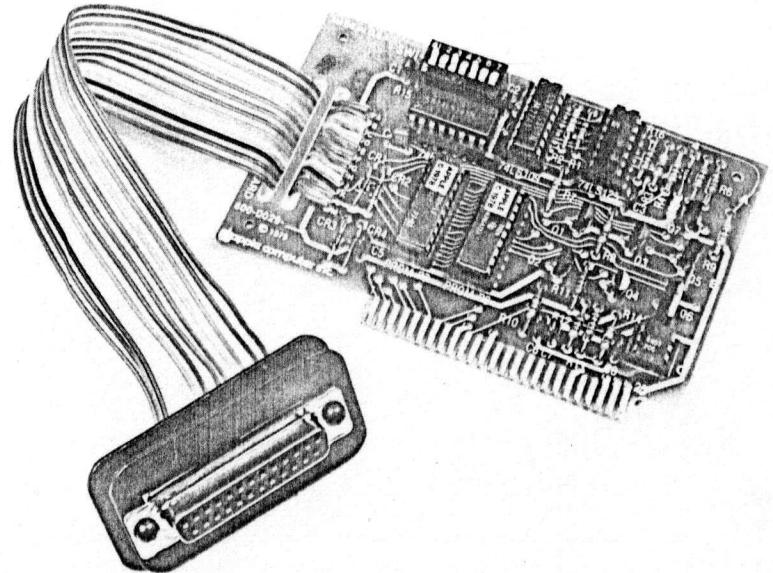
1. Output from the APPLE II can be sent to a serial printer or other external serial device, to the APPLE's TV screen, or to both. The Serial Interface can supply the necessary line-feeds with carriage-returns, etc.
2. Input for the APPLE II can be taken either from an external device or from the APPLE's keyboard, or from both simultaneously.
3. The APPLE II can handle half-duplex communications at rates from 75 to 19,200 baud, in both directions, with a printer, another APPLE, a terminal, modem or other RS232 external device.
4. The Serial Interface can also be connected for current-loop operation with a Teletype.

While this document is intended primarily for APPLE users who are familiar with the RS232 interface, many of the terms and concepts will be explained.

I INSTALLATION

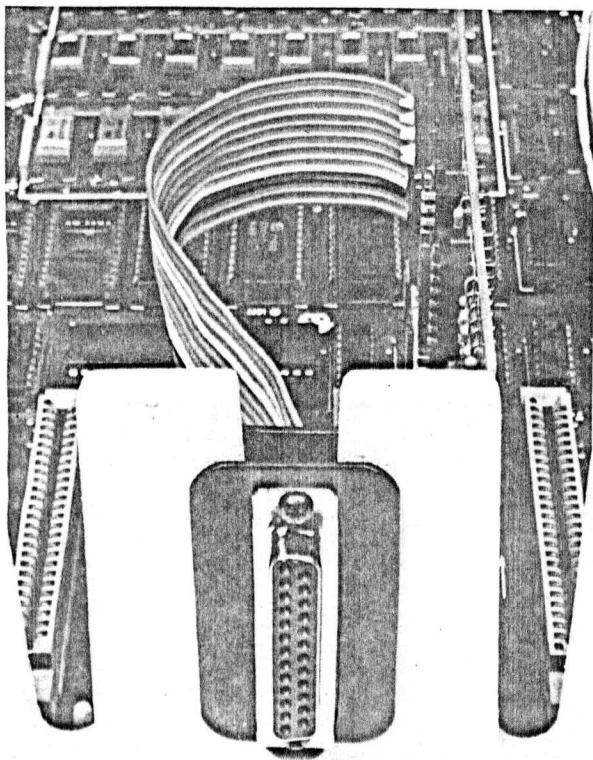
HOW TO INSTALL THE SERIAL INTERFACE

The Serial Interface consists of three parts: the Interface printed-circuit card itself, a female DB-25 connector, and a flat ribbon cable between them.



To install the Serial Interface, you will simply plug the Interface card into a socket inside the APPLE II, and then tighten a clamp to hold the DB-25 connector in place, as follows:

1. **Turn off the power switch** on the back of the APPLE II. This is important to prevent damage to the computer.
2. Remove the cover from the APPLE II. This is done by pulling up on the cover at the rear edge (the edge farthest from the keyboard) until the two corner fasteners pop apart. Do not continue to lift the rear edge, but slide the cover backward until it comes free.
3. Inside the APPLE II, across the rear of the circuit board, there is a row of eight long, narrow sockets called "**slots.**" The leftmost one (looking at the computer from the keyboard end) is slot #0, and the rightmost one is slot #7. Insert the "fingers" portion of the Serial Interface card into slot #1, the *second* socket from the left. The "fingers" portion will enter the socket with some friction and will then seat firmly. The Interface card may be placed in any slot except slot #0, the leftmost. However, APPLE's stan-



standard location for printer interfaces is slot #1 (the second from the left). This manual and most APPLE software for the Serial Interface are written assuming you have installed the Serial Interface card in slot #1.

4. Slip the DB-25 connector and its two metal plates as far down as possible into one of the three long vertical openings in the back of the APPLE II case. One plate goes on the inside of the case; the other plate goes on the outside of the case with the connector's flange on the outside of this plate. Any of the three large vertical openings may be used, but it is customary to use the middle one. Notice that the connector is not symmetrical. When seen from the back of the APPLE II, the longer side of the connector should be on the left (although it will work in either position).
5. Tighten the screws on the DB-25 connector just until the connector assembly can no longer be moved in the opening. Excessive tightening will cause the metal plates to bend.
6. Replace the cover of the APPLE II, remembering to start by sliding the front edge of the cover into position. Press down on the two rear corners until they pop into place.
7. The Serial Interface is installed, and the APPLE II may now be turned on.

COMPATIBILITY WITH EXTERNAL DEVICES

For communications between computers and computer-related equipment, the most widespread and universal standard is the RS232 standard. The RS232 standard specifies the electrical parameters, the form of the signals, and even the type of connector to be used in an interface. The APPLE Serial Interface complies with this standard.

The RS232 standard allows for a number of different communication speeds. These speeds are measured in terms of a unit called the "baud." Each multiple of 10 baud is equal to about 1 character sent or received per second; 300 baud is roughly equal to 30 characters per second. The Serial Interface can operate at any of 256 different speeds, from 75 baud to 19,200 baud.

Computers and their related devices do not actually send the keyboard characters themselves, of course. Each character is encoded in the form of electrical signals, and it is these electrical signals which are sent and received.

The APPLE Serial Interface can communicate with any device that specifies RS232 operation between 75 and 19,200 baud. Many devices can operate at a number of speeds. Very often a set of switches or a rotary dial selects the baud rate. These external devices should be set to a particular baud rate before being connected to the APPLE. The highest baud rate available is usually preferred. The Serial Interface should be set for the same baud rate, using the first 3 levers of the Serial Interface's DIP switch (this is explained in the section, SERIAL INTERFACE OPERATING PARAMETERS). All common baud rates are listed in the section, SERIAL INTERFACE TIMING.

While such operation does not conform to the RS232 standard, the APPLE Serial Interface can also be operated in the current-loop mode necessary to communicate with a serial teleprinter such as the Teletype Model 33ASR.

RS232 CONNECTOR USAGE

The standard DB-25 connector, which is supplied with the Serial Interface, has 25 pins. Six of these are connected internally to the APPLE Serial Interface, but for most applications only three of them need be used. If you don't have a ready-made cable that can go from the Interface's DB-25 connector to the external device, then you will have to wire an interconnecting cable. A cable is just a number of electrically distinct wires that physically run alongside each other. When you wire the cable, you will have to refer to the DB-25 connector's pin numbers. These numbers are molded into the connector, although sometimes they are almost vanishingly small.

The following list describes the functions of the active pins on the Serial Interface connector. The other pins may be left unconnected.

PINS 4 & 5 These pins have been wired together (jumped) at the Serial Interface card. No connection need be made to these pins.

PINS 6, 8 & 20 These pins have been wired together (jumped) at the Serial Interface card. No connection need be made to these pins.

PIN 7 This is called "signal ground." It should be wired to pin 7 at the other end of the cable. If there is no connector at the other end of the cable, then the Serial Interface's pin 7 should be wired to a signal ground connection on the external device. (If this is insufficient information, any additional data would have to be supplied by the manufacturer or designer of the external device.)

PIN 2 The characters from the external device arrive at the computer via this pin.

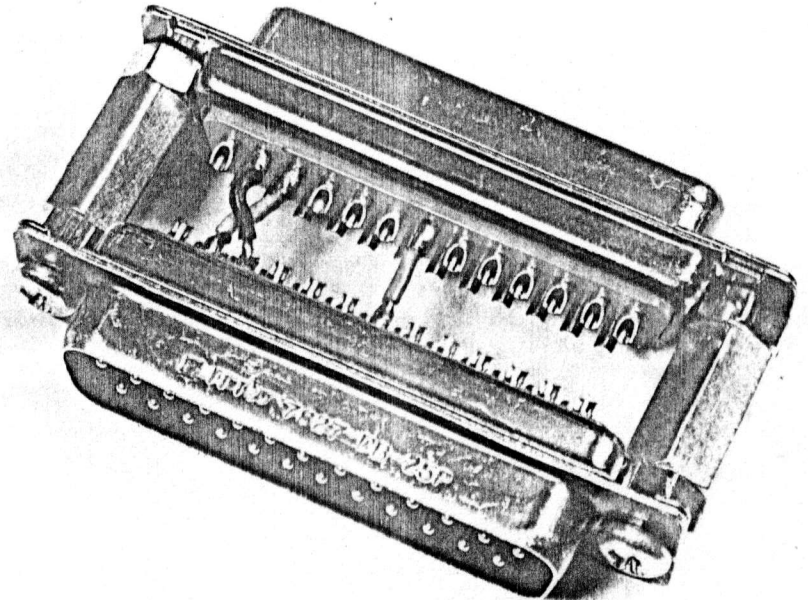
PIN 3 The characters leaving the computer, on their way to the external device, exit via this pin.

Pins 2 and 3 have been left for last since, if the external device end of the cable is another 25 pin connector, there are two ways that they might be wired. No damage is caused by wiring these pins the wrong way, but characters will not be sent out or received.

A. If the external device is a terminal or printer with an RS232 interface itself, then pin 2 on the APPLE's end of the cable should be wired to pin 2 at the external device's end of the cable. Similarly, pin 3 on the APPLE's end of the cable should be wired to pin 3 at the external device's end of the cable. Most of these devices, like the APPLE Serial Interface, also have a **female** DB-25 connector. Therefore your cable will (most likely) need to have a **male** DB-25 connector at *each* end.

B. If the external device is a modem, or another computer with a standard serial interface, then *its* interface will send characters out via pin 3 and receive characters via pin 2 just as the APPLE Serial Interface does. Therefore, you must wire pin 2 at the APPLE's end of the cable to pin 3 at the modem end of the cable; and wire pin 3 at the APPLE's end of the cable to

pin 2 at the modem end of the cable. Modems usually have a **male** DB-25 connector. Therefore, you will probably need a cable with a **female** DB-25 connector at the modem end, and a **male** DB-25 connector at the APPLE end.



Note that only the wires to pins 2 and 3 are involved. No other wires need be changed, no matter what external RS232 device is connected to the APPLE Serial Interface.

Most commercially prepared cables are simply extension cables: they connect identically numbered pins at the two ends of the cable. For use with a modem, you may have to re-wire the cable, interchanging the wires to pins 2 and 3 at one end. If you would rather not tamper with a ready-made cable, you can make an adapter with a male DB-25 connector at one end and a female DB-25 connector at the other end. It should be wired with pins 7 connected, and pins 2 and 3 cross-connected. This adapter, when connected between the APPLE's Serial Interface and a standard RS232 cable, allows the Interface to "talk" to most modems.

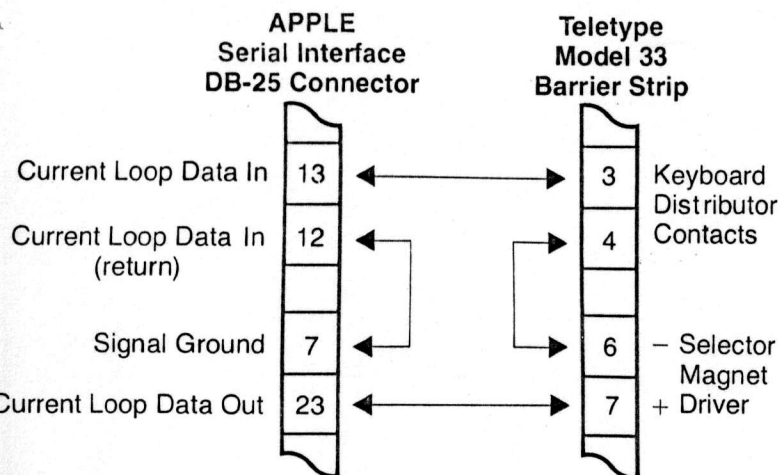
CURRENT-LOOP OPERATION WITH A TELETYPE

If you wish to use the APPLE Serial Interface to communicate with a current-loop teleprinter such as the Teletype Model 33ASR, you will be interested in three other Interface pins.

PIN 13 In current-loop operation, the characters from the Teletype arrive at the APPLE via this pin. This pin should be wired to terminal 3 on the Teletype Model 33's barrier strip.

PIN 12 The return path of the input current loop (the loop for characters arriving from the Teletype) should be connected to this pin. (In fact, the Serial Interface does not care which way current flows through this input loop. We have arbitrarily chosen pin 13 as the input and pin 12 as the return, but the roles of these two pins can be interchanged.) We suggest using signal ground (at pin 7) for the return path, in which case you should connect pin 12 to pin 7.

PIN 23 In current-loop operation, the characters leaving the APPLE, on their way to the Teletype, exit via this pin. Wire this pin to terminal 7 on the Teletype Model 33's barrier strip. The return path for this output current loop is also the signal ground at pin 7. For half-duplex operation, connect terminals 4 and 6 on the Teletype Model 33's barrier strip.



Caution! Pins 1 and 2 on the Model 33 barrier strip are connected to 120 Volts AC.

II OPERATION





USING THE APPLE SERIAL INTERFACE

The Serial Interface allows the APPLE II to communicate with other electronic devices which are *external* to the computer. These devices may be—to give a few examples—terminals, printers, or other computers. The Serial Interface can be controlled through BASIC programs or through assembly-language programs. It can also be controlled directly, by typing a few characters on the APPLE's keyboard.

In the following discussion, it will be assumed that you are familiar with the APPLE II BASIC Programming Manual, and that your APPLE II is operating in BASIC, with the Serial Interface installed in slot #1.

Here is a list of the most common tasks the Serial Interface is called upon to do, and the commands that accomplish them.

1. Send subsequent output to the Serial Interface.
PR#1
2. Cancel the effect of PR#1, sending output only to APPLE's TV screen.
PR#0
3. Accept subsequent input from the Serial Interface, as well as from the APPLE's keyboard.
IN#1
4. Force the APPLE to convert all lower-case characters to upper-case, as they arrive from the external device.

  (type the  key, then type )

5. Allow the APPLE to accept lower-case characters that arrive from the external device. If displayed on APPLE's TV screen, the lower-case characters will appear as upper-case characters in inverse video.

These tasks are more fully explained, and some fine points considered, in the next few pages.

To understand exactly how the Serial Interface operates, it is useful to think of the APPLE II as divided into three parts:

1. The APPLE's keyboard, which **generates** characters (when you type on it).
2. The APPLE's TV screen, which can **absorb** characters (and make them visible).
3. The APPLE's processor, or "brain," which can **control** the flow of characters, and act upon them.

You can also think of any external device as being able to **generate** characters or **absorb** characters, or both. The external device may or may not have a "brain," but this is not important in understanding the operation of the Serial Interface.

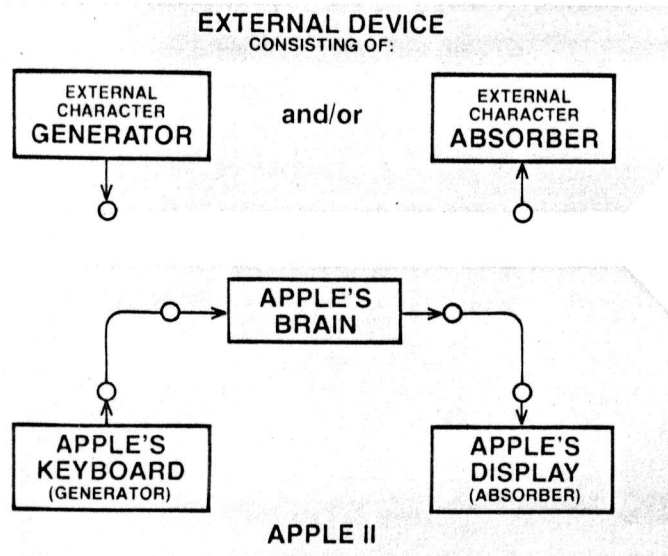


FIGURE 1

Figure 1 shows the three parts of the APPLE II, and the only parts of any external device that affect the Serial Interface. It also shows the normal interconnection between these parts. When the APPLE II is first turned on, it will ignore the external world, listening only to its own keyboard and displaying characters only on its own TV screen.

PRELIMINARY DISCUSSION OF THE IN# AND PR# COMMANDS

There are eight sockets, called "slots," on the back of the main circuit board inside the APPLE II. The leftmost one (as viewed from the keyboard end of the computer) is slot #0, and the rightmost one is slot #7. (See the section, HOW TO INSTALL THE SERIAL INTERFACE.) APPLE BASIC has two commands for selecting among these slots for input and output. In effect, when you first invoke BASIC, the commands

IN#0
and
PR#0
are automatically executed. The first of these commands, **IN#0**, tells the APPLE to

Take **IN**put from the APPLE keyboard.
And the second command, **PR#0**, instructs the APPLE to

Send **PR**inting to the APPLE's TV screen. This is the "normal," or APPLE-alone condition shown in Figure 1. Now, however, if the command (or program statement)

IN#1
is executed, the APPLE will henceforth take its input from whatever is plugged into slot #1. Similarly, if the command **PR#1** is executed, all output will be sent to whatever is plugged into slot #1. If there is nothing plugged into the specified slot, then the system may hang, or your program may be erased, or other strange behavior may result. Notice that slot #0 is special, and refers to the APPLE itself.

SENDING YOUR OUTPUT TO AN EXTERNAL DEVICE AND RECEIVING INPUT FROM AN EXTERNAL DEVICE

In the following examples, the commands **PR#1** and **IN#1** will be typed on the APPLE's keyboard. If you have put your Serial Interface into slot #1 (the second one from the left, as described in the section, HOW TO INSTALL THE SERIAL INTERFACE) the commands will work exactly as shown. If you use some other slot, you will have to substitute the number of that slot. **Slot #0 may not be used for the Serial Interface.**

Attach an appropriate cable (see the section, RS232 CONNECTOR USAGE) from the DB-25 connector of the Serial Interface to the external device with which you wish to communicate. Reset your APPLE II by pressing the

CTRL **CTRL**
RESET key, and invoke BASIC typing a **B** (**B** means depressing the **B** key while simultaneously holding down the key marked **CTRL**). If you are not familiar with this procedure, see your APPLE II BASIC Programming Manual. When the prompt character appears, type the command

PR#1
(and press the **RETURN** key, of course) . From now on, any characters you type will be sent out through the Serial Interface to the external device. The characters will appear on the APPLE's TV screen only if levers 5 and 6 of the Serial Interface's **DIP** switch (located directly on the Interface card itself, near its upper edge) are **ON** when **PR#1** is typed. (This is more fully explained in the section, SERIAL INTERFACE OPERATING PARAMETERS.) Characters coming in from the external device will be ignored. The operation of the system after you type **PR#1** is shown in Figure 2.

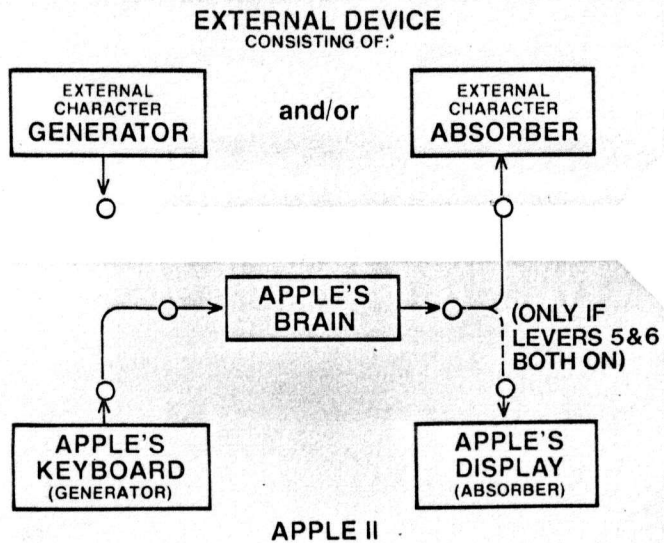


FIGURE 2

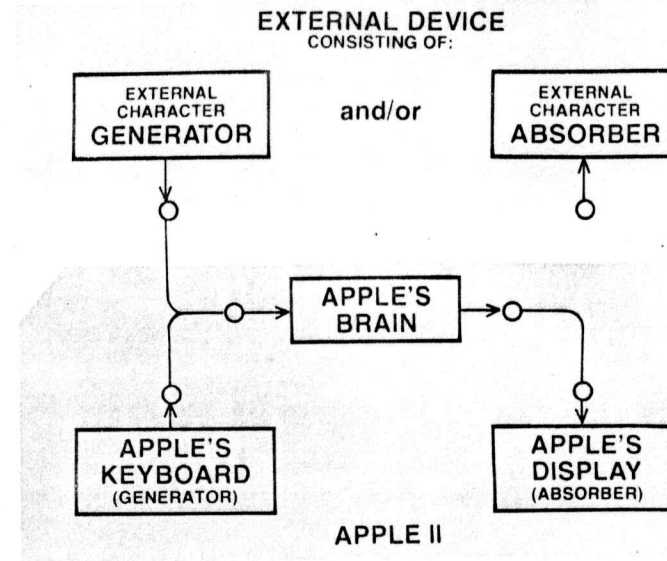


FIGURE 3

Operation of the system after executing PR#1

The APPLE's "brain" is still connected, and the command **PR#0** will restore normal (Figure 1) operation, in which the APPLE's output characters are not sent through the Serial Interface. Normal operation can also be

restored by pressing the **RESET** key and then typing a **CTRL C**, but this option is not available if the Interface is being controlled by a program.

To let the external source of characters control the APPLE II, use the command

IN#1
After this command, the APPLE will accept input from the external device connected to the Serial Interface, as well as from its own keyboard. Figure 3 shows this condition. If there is no external device connected to the Serial

Interface, the system will "hang" after this command. Use **RESET CTRL C** to recover.

Operation of the system after executing IN#1

Normal operation is restored if the command **IN#0** is typed on the APPLE's keyboard. Normal operation is also restored if the external device sends the command **IN#0**

Pressing the **RESET CTRL C** on the APPLE's keyboard will also restore normal operation; but this cannot be done from a program, and will not be mentioned again.

Typing **both** commands, **PR#1** and **IN#1** will give the external device full control of the APPLE II. In this "remote mode" (shown in Figure 4), a friend could use your APPLE from across the country—or across the room.

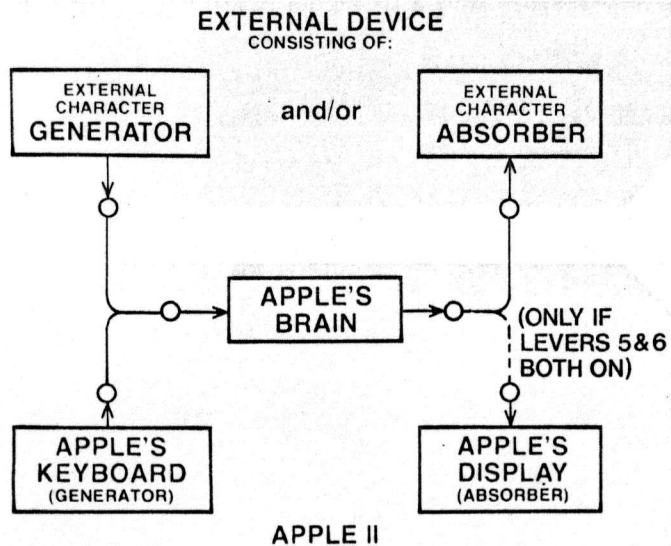




FIGURE 4

Remote Mode: operation of the system after executing PR#1 and IN#1.










III DEFAULT PARAMETERS AND THE DIP SWITCH

INITIALIZING THE SERIAL INTERFACE

Before the Serial Interface can be used, it must be **initialized**. Initializing the Interface sets all of the Interface operating parameters to their **default** values. Assuming slot #1, the Interface is initialized each time either of the following BASIC commands is typed:

PR#1  or IN#1 

and each time any of the following Monitor commands are typed:

    or     or C100G 

When used within a *program*, a command (such as PR#1) that transfers APPLE's output to the Serial Interface does *not* initialize the Interface until the first character is actually sent out (with a PRINT statement, for instance). Similarly, if during a *program* a command (such as IN#1) tells the APPLE to get its input from the Serial Interface, the Interface is *not* initialized until the APPLE actually looks for its first input character (in an INPUT statement, for instance).

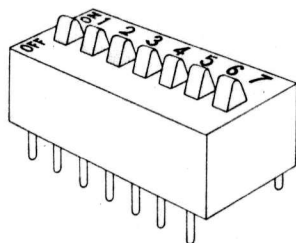
SERIAL INTERFACE OPERATING PARAMETERS

The Serial Interface has ten user-definable operating parameters. Each time the Serial Interface is initialized, the ten operating parameters are given their **default** values. Five of the default values are determined by the 7 levers of the Serial Interface's **DIP** switch (located on the Interface's printed-circuit card, near the upper edge). The **DIP** switch levers set the default values for these five operating parameters: **Baud Rate** (levers 1, 2 and 3), **Carriage Return Delay** (lever 4), **Line Width** plus **APPLE Video** (levers 5 and 6), and **Line Feed** (lever 7). Changing the settings of the **DIP** switch levers *after* initialization has no effect until the *next* initialization.

SETTING THE DIP SWITCH DEFAULTS

1) Baud Rate

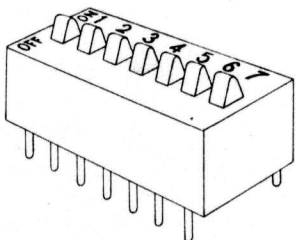
DIP Switch Levers			Default Baud Rate
1	2	3	
On	On	On	= 110 baud
Off	On	On	= 134.5 baud
On	Off	On	= 300 baud
Off	Off	On	= 1200 baud
On	On	Off	= 2400 baud
Off	On	Off	= 4800 baud
On	Off	Off	= 9600 baud
Off	Off	Off	= 19200 baud



On initialization, the settings of **DIP** switch levers 1, 2 and 3 determine the rate at which bits may be transmitted to the external device. 300 baud is 300 bits per second. Under default conditions, each character is transmitted using 11 bits (1 start bit, 8 data bits, and 2 stop bits).

2) Carriage Return Delay

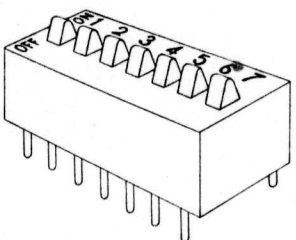
DIP Switch Lever	Default Car. Return Delay
4	
On	= Disabled
Off	= Enabled



If **DIP** switch lever 4 is Off (Delay Enabled), the Serial Interface will wait briefly (approximately 1/4 second) after transmitting a carriage return, to allow the printer to complete this movement. If you are transmitting to an external TV screen, this delay is probably unnecessary, and lever 4 may be turned On (Delay Disabled).

3) Line Width plus APPLE Video

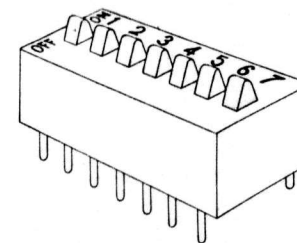
DIP Switch Levers		Default Line Width	Default APPLE Video
5	6		
On	On	= 40 Char/Line	Enabled
Off	On	= 72 Char/Line	Disabled
On	Off	= 80 Char/Line	Disabled
Off	Off	= 132 Char/Line	Disabled



After a carriage return, the settings of **DIP** switch levers 5 and 6 determine the maximum number of characters transmitted before the Serial Interface will force another carriage return to be sent out. Characters will be displayed on the APPLE's TV screen only if the default line width is set to 40 characters per line (levers 5 and 6 On). After initialization, the line width can be changed from 40 characters per line, but the display on the APPLE's TV screen will not correspond to the display on the external device, as transmitted carriage returns are not accompanied on the APPLE's TV screen by line feeds.

4) Line Feed

DIP Switch Lever	Default Line Feed
7	
On	= Disabled
Off	= Enabled



If **DIP** switch lever seven is Off (Line Feed Enabled), the Serial Interface will transmit a line feed after each carriage return it transmits. If the external device automatically supplies its own line feed after each carriage return received, you can set lever seven to On (Line Feed Disabled) to avoid double-spacing.

PERMANENT DEFAULTS

During each initialization, the five remaining operating parameters are set to their **permanent** default values:

1. **Parity** defaults to its disabled condition (no parity bit).
2. **Checksum** defaults to its disabled condition (no checksum character).
3. **Lower-Case** defaults to its disabled condition (converts all incoming lower-case characters to upper-case).
4. **Number of Data Bits** defaults to 9 (8 data bits plus one start bit).
5. **Number of Stop Bits** defaults to 2.

These parameters will be explained in the following section; they can only be changed by software commands *after* initialization.

IV ACCESS TO OPERATION PARAMETERS

DESCRIPTION OF SERIAL INTERFACE OPERATION

For most applications, the default operating parameters (both those that are fixed and those that can be set with the **DIP** switch) will be just what you need: your parameters will be set each time the Interface is initialized. In that case, the following section will be interesting but not necessary. However, the Serial Interface is designed to be very flexible, so that its operating parameters can be easily modified for use in a wide variety of special applications. This section gives a rather detailed description of the Serial Interface's operation. The following section shows you how to make any of the many possible modifications to that operation, should they be necessary.

Each character that is sent out through the Serial Interface is transmitted as a series of bits, in the following sequence:

1. One "Start Bit," a "low" voltage which tells the external device that a character is going to be transmitted.
2. From two to eight "Data Bits" (default is eight bits; can be changed by the user), a sequence of "high" and "low" voltages that represent the actual character code being transmitted. The default is eight bits because the APPLE normally handles data in eight-bit groups. If your external device sends and receives data in groups of fewer than eight bits, you must set the Serial Interface to send and receive these smaller groups.
3. *If enabled*, one "Parity Bit" (default is no parity bit; even or odd parity can be enabled by the user). This is a transmission-accuracy checking bit which the external device looks at for errors and then discards.

The parity bit is found as follows: all the 1-bits in the actual character code are added together, and the result's evenness or oddness is compared with the type of parity-checking selected. For instance, the binary ASCII code for the letter S is 1010011; the sum of the 1-bits is 4, an *even* number. If *Even* parity has been enabled, the comparison is *true*, and a 0-bit is sent at the end of the character. If *Odd* parity has been enabled, the comparison is *false*, and a 1-bit is sent. If parity has not been enabled, no extra bit is sent. Check your device's operation manual to see if it sends and receives parity bits.

4. From one to 127 "Stop Bits" (default is two bits; can be changed by the user), a "high" voltage which tells the external device that a character has been completely transmitted. Each external device requires a particular number of stop bits after every character; see the device's operation manual.

This same sequence will be used by the external device when it transmits to the APPLE. Timing is very important to correct transmission and reception. The Serial Interface sends out and receives bits at fixed intervals of time set by the "Baud Rate" (default is set by **DIP** switch levers 1, 2 and 3; can be

changed by the user). The Serial Interface and the external device *must* be set to the same baud rate and parity option, in order to interpret the sequence of "high" and "low" voltages correctly. The same is true when the external device is transmitting characters to the APPLE Serial Interface.

When the Serial Interface has sent out the number of characters set by the "Line Width" (default is set by DIP levers 5 and 6; can be changed by the user), it transmits a "Carriage Return" to the external device. After sending a carriage return, it may wait during a fixed ¼-second "Carriage Return Delay" (if enabled: default set by DIP lever 4; can be changed by the user) before sending the next character, to allow the printer to complete this movement. Then the Interface may send a "Line Feed" (if enabled: default set by DIP lever 7; can be changed by the user), so that subsequent characters will appear on the following line.

Finally, each time it completes sending 256 characters in a "Batch Move," the Serial Interface may send a "Checksum" character (default is no checksum character; can be changed by the user). This is a transmission-accuracy checking character which the external device looks at for errors and then discards. The checksum character is found by XORing the previous 256 characters, as follows: the second character is XORed with the first, the third with that result, the fourth with *that* result, and so on. Check your device's operation manual to see if it sends and receives checksum characters. During Batch Moves, the Serial Interface and the external device must be set to the same checksum option.

LOWER-CASE CHARACTERS

While the APPLE generates and displays characters only in upper case (capital letters), your external device may generate both upper-case and lower-case characters. When lower-case characters are received by the APPLE through the Serial Interface, they can be treated in two different ways:

1. Convert all incoming lower-case characters to upper-case characters (incoming upper-case characters are not affected). Since this is the usual APPLE mode, any incoming characters displayed on the APPLE's TV screen will look fine. This is the permanent default condition; but it can be changed by the user, after initialization.
2. Accept all incoming lower-case characters as lower-case (again, incoming upper-case characters are not affected). APPLE's TV screen *display* of characters is designed for upper-case only, so any display of incoming lower-case characters will look strange. If the characters are being displayed as they arrive from the external device, APPLE will show the lower-case characters as upper-case characters in *inverse video* (black letters on a white background). Once stored in APPLE's memory, lower-

case characters will be displayed (when LISTed, for instance) as a strange assortment of upper-case characters in normal (white on black) video. There is one exception to this: if the stored lower-case characters are being displayed as they are sent out through the Serial Interface, they will again appear as upper-case characters in inverse video. Note that these peculiar *displays* do not reflect the lower-case characters themselves: in this mode they are *stored* correctly in APPLE's memory, and may be printed correctly on any appropriate external device. Note also that this mode does not add any capability to *generate* lower-case characters from the APPLE keyboard.

CHANGING SERIAL INTERFACE PARAMETERS THROUGH SOFTWARE COMMANDS

Ten of the Serial Interface parameters can be changed from their initialized (default) values, through the use of commands in machine language or BASIC. Once an Interface parameter is set by a software command, that parameter remains unchanged until the Serial Interface is reinitialized or the parameter is reset by another software command. For more discussion of the various parameters functions, see the previous section.

In the following descriptions, the letter "s" refers to the number of the printed-circuit board slot inside the APPLE, in which the Serial Interface card is installed (see the section, HOW TO INSTALL THE SERIAL INTERFACE).

1. BAUD RATE (assembly-listing variable: BRATE)

Memory location 1144+s (\$478+s, in Hexadecimal) contains APPLE's baud "quantum" number, which specifies how many "quanta" the APPLE is to wait between sending out bits through the Serial Interface. One quantum equals 53 APPLE II cycles (51.94 microseconds) per transmitted bit. The default value is set with levers 1, 2 and 3 of the Interface card's DIP switch (see the section, SETTING THE DIP SWITCH DEFAULTS). From BASIC, to change the baud rate from the default value to B use the command

POKE 1144+s, r

where r is the integer, from 0 through 255, that is closest to $1/((.00005194 * B)$

For further information, see the section SERIAL CARD TIMING.

2. STOP BITS (assembly-listing variable: STBITS)

Memory location 1272+s (\$4F8+s, in hexadecimal) contains the Number of Stop Bits (Note: the one parity bit is *included* in this number, if

parity is enabled). The default value is 2 stop bits (and no parity bit). To change the number of Stop Bits from BASIC, use the command **POKE 1272+s, r** where r is an integer, from 1 through 127. To determine the correct number of stop bits for your external device, see the external device's operation manual.

Note: you must add the one parity bit to the Number of Stop Bits, if parity is enabled.

3. PARITY/CHECKSUM OPTIONS (assembly-listing variable: STATUS)

Memory location 1400+s (\$578+s, in hexadecimal) contains a number, the lower three bits of which determine two parity options (enable/disable and even/odd) and one checksum option (enable/disable). If the remote device with which your Serial Interface is communicating requires a parity bit to be sent or received with each character, you can tell your Serial Interface to do this task. You can also specify which type of parity check (even parity or odd parity) is to be sent and received. If your remote device requires that a checksum be sent after every 256 characters in a Batch move, you can tell the Serial Interface to send one. To decide whether your external device requires either a parity bit or a checksum character (or both), consult the device's operation manual. The three Parity/Checksum options are changeable from BASIC by using the command

POKE 1400+s, r

where r is an integer from 0 through 7. The actual value that r should be assigned is determined as follows:

- | | | |
|--------|---------------------|---------------------------------|
| Bit 0: | 1 = odd parity | (This is the least significant, |
| | 0 = even parity | or rightmost, bit.) |
| Bit 1: | 1 = no parity | (initial default value) |
| | 0 = parity enabled | |
| Bit 2: | 1 = no checksum | (initial default value) |
| | 0 = checksum enable | |

First determine whether or not a parity bit need be sent (Bit 1). If yes, then decide whether the parity should be odd or even (Bit 0). Also, determine whether or not a checksum character need be sent during Batch moves (Bit 2). For example, let's assume that an even parity bit must be sent, with no checksum. Bit 0 gets a value of 0, Bit 1 gets a value of 0, and Bit 2 gets a value 1. This binary number 100 is converted to its decimal equivalent of 4 and POKE'd (assuming slot #1):

POKE 1401, 4

4. INPUT/OUTPUT BUFFER (assembly-listing variable: BYTE)

Memory location 1656+s (\$678+s, in hexadecimal) is the input buffer for the individual character that has just been received through the Serial Interface from the external device. Assuming the Interface is in slot #1, the BASIC command

PRINT PEEK (1657)

will print on the APPLE's TV screen the ASCII value of the character just received.

5. LINE WIDTH (assembly-listing variable: PWDTH)

Memory location 1784+s (\$6F8+s, in hexadecimal) contains the "Printer Width," or number of characters per line. After transmitting this number of characters, the Serial Interface will then transmit its Carriage-Return sequence. To change the number of characters per line, from BASIC, use the command

POKE 1784+s, r

where r is an integer, from 0 through 255, specifying the number of characters per output line. To determine the maximum line width for your external device, consult the device's operation manual.

Note: if r is set to zero, the Serial Interface will not force any carriage returns to be transmitted. The output characters will be transmitted in a continuous stream.

6. DATA BITS (assembly-listing variable: NBITS)

Memory location 1912+s (\$778+s, in hexadecimal) contains the number of Data Bits, plus one for the start bit. In the APPLE, data is handled in groups containing eight bits. If you are communicating with an external device which also handles data in eight-bit groups, the default Number of Data Bits is perfect (8 data bits plus 1 start bit). However, if your external device handles data in groups of fewer than eight bits, you must set the Serial Interface to send and receive these smaller data groups.

When receiving data groups of fewer than eight bits, the Serial Interface will supply 1's to fill the remaining high-order bits of each eight-bit group in APPLE's memory. Similarly, when the Serial Interface is transmitting data groups of fewer than eight bits, the unused high-order bits in each of the APPLE's eight-bit data groups must be set to 1's.

To change the Number of Data Bits from BASIC, use **POKE 1912+s, r**

where r is an integer, from 3 (2 data bits plus one start bit) through 9 (8 data bits plus one start bit).

Note: to calculate *r*, you must add one start bit to the number of data bits. If *r* is set to less than the default value of 9 (8 data bits plus one start bit), you must set the unused high-order data bits to *ones* before transmitting the data. Received data will also have unused high-order data bits set to *ones*.

Example: Binary Coded Decimal is a code for sending numbers in four-bit data groups. The BCD code for the number 7 is 0111. If the Number of Data Bits, *r*, is set to 5 (4 data bits plus 1 start bit), BCD for the number 7 must be stored in the APPLE's eight-bit byte as 11110111 before the data group 0111 can be transmitted. Similarly, if the data group 0111 is received by the Serial Interface, it will be stored in the APPLE's eight-bit byte as 11110111.

7. OPERATION MODES (assembly-listing variable: FLAGS)

Memory location $2040+s$ ($\$7F8+s$, in hexadecimal) contains a number, four of whose bits determine four separate modes of operation. To alter the operation modes from BASIC, use the command

POKE 2040+s, r

where the value of *r* is determined by use of the following table:

r's Binary Bit	(Decimal Equiv., If Bit=1)	Operation Set By Bit Value	Default
Bit#0	(1)	1 = Line feed after carriage return 0 = No line feed	(Set by Lever 7)
Bit#5	(32)	1 = Lower-case input enable 0 = Convert lower-case to Upper-case	(Permanent Default)
Bit#6	(64)	1 = No delay after carriage return 0 = Carriage return delay enable	(Set by Lever 7)
Bit#7	(128)	1 = No display on APPLE's TV 0 = APPLE's display enabled	(Set by Levers 5&6)*





*APPLE's TV display is only enabled during initialization if DIP switch levers 5 and 6 are both On.

For example, let us assume that you wish to have line feeds, upper-case only, carriage return delay, and no APPLE display. This would require that bits 0 and 7 have a value of 1, and bits 5 and 6 have a value of 0. Add up the decimal equivalents of all of the bits that were assigned the value 1 (the decimal equivalents are the numbers in parentheses, next to

the Bit #'s). The decimal equivalents for bits 0 and 7 are 1 and 128 respectively; therefore the total of the decimal equivalents is 129. This value is assigned to *r*, and POKE'd (assuming slot #1):

POKE 2041, 129



If you wish to change only Bit #5 (lower-case input enable/convert), you can do so with the following commands:

  (Press and release the  key, and then type )

This changes *r*'s Bit #5 to a zero, the default value. After this command, all lower-case characters arriving through the Serial Interface from an external device will be converted to upper-case characters. Incoming upper-case characters are not affected. This is the APPLE's usual mode, so any APPLE display will look fine.

This changes *r*'s Bit#5 to a one. After this command, lower-case characters arriving through the Serial Interface from an external device will be stored as lower-case characters in APPLE's memory. Upper-case characters are not affected. Since the APPLE was designed for upper-case characters only (BASIC will accept lower-case characters only in quoted strings), any APPLE *display* of these lower-case characters will look strange on the TV screen. See the previous section for details. However, the characters are *stored* correctly, and may be printed correctly on any appropriate external device.

Note: the commands  and  are Serial Interface *input* commands. They will have no effect unless the Interface has been initialized for input (by IN#1, for instance).

8. TAB

The TAB and comma functions in Integer BASIC (HTAB in APPLESOFT) will sometimes work in conjunction with the Serial Interface, but have several restrictions (fewest for comma-tabbing). A TAB of less than 18, if it would end directly on a character already printed, may be simply tabbed from that character's position. No TAB can cause printing to occur to the *left* of the last printed character on the current line. An attempt to do so usually causes printing to occur in the first available position to the *right* of the last printed character. Both Integer BASIC's TAB and APPLESOFT's HTAB send out a carriage return for every 40 positions in the tab instruction, and then tab the remaining positions. For tabbing to any position (including those beyond position 40), you can use the BASIC command

POKE 36, r

where *r* is an integer, from 0 through 255, equal to the number of print

positions to be tabbed. This command suffers most of TAB's restrictions, except for the 40-position limit. In APPLESOFT, the TAB function (used inside a PRINT statement) can also cause tabbing of more than 40 positions.

V DIRECT USE OF THE INTERFACE

TRANSMITTING A CHARACTER WITHOUT USING PR#1

Occasionally, it is useful to send a character out through the Interface *without* using a PR# command to change the "output vector" (the system pointer that tells your APPLE where to send its output, normally to its TV screen). To use the Serial Interface *directly*, follow these two steps:

1. Into APPLE's accumulator, put the ASCII code of the character to be sent.
2. CALL -16384 + (256 * s)

where s is the Interface's slot number (the equivalent hexadecimal location to CALL is \$Cs00). There are various ways to get a number into the APPLE's accumulator, but one way is to write a very small machine-language subroutine to do it, and then CALL that subroutine from your BASIC program. To begin, press the **RESET** key to enter the Monitor (prompt character: *), and then (assuming slot #1) type

```
300: A9 11 4C 00 C1 RETURN
```

Check your work by typing

```
300L
```

Ignoring most of the resulting display, the first two lines should look like this:

```
0300- A9 11 LDA #$11
0302- 4C 00 C1 JMP $C100
```

The first instruction (at hexadecimal location \$300) tells the APPLE to **LoaD** the **Accumulator** with the number in the next location (hexadecimal location \$301). For now, that number is \$11. The second instruction is equivalent to CALL -16128 in BASIC: it tells the APPLE to **JuMP** to hexadecimal location \$C100, which starts the Serial Interface character output routine. To use this subroutine in your BASIC program, you must first put into hexadecimal location \$301 (that's location 769, in decimal) the ASCII code for the character you want the Serial Interface to send out. Then you will CALL the subroutine at hexadecimal location \$300 (768, in decimal). Here is a short program that uses the above machine-language routine to send out one character at a time:

```
10 INPUT "LETTER?", L$
20 POKE 769, ASC(L$)
30 CALL 768
40 GOTO 10
```

BATCH MOVES

At times it is useful to send or receive large amounts of information very quickly. This can be accomplished through use of a *Batch Move*. The Batch subroutines are "utility" routines. They are intended to be used for special

applications such as Data Collection, Mass Storage and Retrieval Systems, and sending program sequences to control external devices. To understand how to use the Batch routines on the simplest level, refer to the examples below. For an example of using the Batch Moves from BASIC, see the next section, BATCH MOVES FROM BASIC.

Note: before using the Batch routines, you must have initialized the Serial Interface (by typing PR#1, for instance) and you must have set the desired parity and checksum parameters. The Batch Move commands deal directly with the Serial Interface (not through the input and output "vectors" set by IN# and PR#); therefore these commands are the same, regardless of which slot contains the Interface card.

1. Batch Output

When in the Monitor (prompt character: *), type

3F8: 4C 41 C9 **CTRL**

This prepares the APPLE to jump to hexadecimal location \$C941 in the

CTRL

Serial Interface's Read-Only Memory when a **Y** is typed on the keyboard. This jump causes the Batch Output routine to execute. When you are ready to actually send the data, type

CTRL

addr1 . addr2 **Y** **RETURN**

where "addr1" is the hexadecimal starting address of the data, and "addr2" is the hexadecimal ending address of the data. For example, if we wanted to send the information that is stored in memory from address \$2000 through address \$3FFF, we would type

CTRL

2000.3FFF **Y** **RETURN**

As soon as the return is typed, the data from address 2000 to address 3FFF will be sent through the Serial Interface from the APPLE to the external receiving device.

2. Batch Input

When in the Monitor, type

3F8: 4C 3D C9 **CTRL**

This prepares the APPLE to jump to hexadecimal location \$C93D in the Serial Interface's Read-Only Memory when a control Y is typed on the keyboard. This jump causes the Batch Input routine to execute. When you are ready to actually receive the data, type

CTRL

addr1 . addr2 **Y** **RETURN**

where "addr1" is the hexadecimal starting address in which the incoming data will be stored, and "addr2" is the hexadecimal ending address in which the incoming data will be stored. For example, if we wanted to receive data from an external device, and store it in our APPLE's memory from address \$4000 through address \$5FFF, we would type

CTRL

4000.5FFF **Y** **RETURN**

As soon as return is typed, the serial data can be sent by the external transmitting device to the Serial Interface. As it is received, the incoming data will be stored in your APPLE's memory from address 4000 through address 5FFF.

Note: when the Serial Interface is instructed to receive a batch move, the cursor on the receiving APPLE's TV screen disappears, and the Interface waits patiently until *all* the specified locations have been filled with received data. Then the cursor returns.

BATCH MOVES FROM BASIC

While it is easiest to use the Batch routines from the Monitor, it is also possible to do Batch Moves from BASIC. In the following discussion these definitions will hold:

- BAL = Beginning Address Low
(the two rightmost digits of the 4-digit hexadecimal starting address for the move, converted to decimal)
- BAH = Beginning Address High
(the two leftmost digits of the 4-digit hexadecimal starting address for the move, converted to decimal)
- EAL = Ending Address Low
(the two rightmost digits of the 4-digit hexadecimal ending address for the move, converted to decimal)
- EAH = Ending Address High
(the two leftmost digits of the 4-digit hexadecimal ending address for the move, converted to decimal)

Suppose you wish to send someone a picture from your APPLE's high-resolution screen (page 1). The memory for this screen lies between 8K and 16K, from hexadecimal address \$2000 to hexadecimal address \$4000. For the Beginning Address \$2000:

BAL = \$00 (hex) = 0 (decimal)
 BAH = \$20 (hex) = 32 (decimal)

For the Ending Address \$4000:

EAL = \$00 (hex) = 0 (decimal)
 EAH = \$40 (hex) = 64 (decimal)

1. Batch Output from BASIC

The following BASIC program does the same task that the Monitor Batch Output command did. See the discussion of the Monitor Batch Output for more details.

```

10 PR#1 : PRINT " "           (initializes Interface)
20 POKE 60, BAL : POKE 61, BAH (sets starting address)
30 POKE 62, EAL : POKE 63, EAH (sets ending address)
40 CALL -14015                (jumps to Output Routine at $C941)
50 PR#0                       (returns to normal TV output)
60 END
    
```

2. Batch Input from BASIC

In the Batch Output program, above, change line 40 to

```

40 CALL -14019                (jumps to Input Routine at $C93D)
    
```

The resulting BASIC program does the same task that the Monitor Batch Input command did. See the Monitor Batch Input discussion for more details. Note that the IN#1 is *not* necessary for accepting input through the Serial Interface, because CALL -14019 deals directly with the Interface (not through the input and output "vectors" set by PR# and IN#). In fact the PR#1 in line 10 was necessary *only* to initialize the Interface;

10 CALL -16128
 would have done as well.

VI APPENDIX: SERIAL INTERFACE TIMING

TABLE OF BAUD RATE QUANTUM NUMBERS

The following is a table that gives seventeen of the most commonly used **baud** rates, along with their quantum value (for POKEing) and percent error. Although only seventeen different **baud** rates are shown here, any integer from 0 through 255 may be POKE'd into the proper address (1144+s), and each will give a different **baud** rate.

Average APPLE II Frequency 1.0204842 MHz
 Period .979926 microseconds
 Jitter +139.7 nanoseconds every 65th cycle

Baud Rate Loop Quantum** 53 APPLE II Cycles (51.94 microseconds)

Baud Rate	Quantum No. (Hex)	(Dec)	Period (microsec.)	Actual Baud Rate	% Error
75	\$ 00	*** 0	13296	75.20	+ .28
90	\$ D6	214	11115	89.96	- .037
110*	\$ B0	176	9141	109.4	- .548
134.5*	\$ 90	144	7479	133.7	- .586
150	\$ 80	128	6648	150.4	+ .28
240	\$ 50	80	4155	240.67	+ .28
300*	\$ 40	64	3324	300.85	+ .28
450	\$ 2B	43	2234	447.74	- .51
600	\$ 20	32	1662	601.7	+ .28
900	\$ 15	21	1091	916.8	+1.86
1200*	\$ 10	16	831	1203.4	+ .28
1800	\$ 0B	11	571.3	1750.2	-2.78
2400*	\$ 8	8	415.5	2406.8	+ .28
3600	\$ 5	5	259.7	3850.59	+6.96
4800*	\$ 4	4	207.7	4813.6	+ .28
9600*	\$ 2	2	103.9	9627.2	+ .28
19200*	\$ 1	1	51.9	19254.4	+ .28

*DIP switch selectable

$$**\text{Quantum} = \frac{1}{(.00005194) * \text{Baud Rate}}$$

***The quantum number zero is treated as 256 (\$100).


```

C160 A8 24 2240 FLA :RECOVER CHAR AT CURSOR
C161 A4 24 2250 LDY CH :GET CURSOR HORIZONTAL INDEX
C162 2260 FMP :SAVE CARRY
C163 2270 JMP INFINSH : (NO ROOM LEFT IN THIS R.O.M.)
C167 2290 ;
C168 2300 MOVUI LDY #0
C169 PD 30 05 2310 LDA COL,X :CHECK FOR COLUMN
C16C FD 30 04 2320 SBC PWDTH,X :WIDTH WITHIN B CHARACTERS
C171 90 01 2330 CMP #0FB :OF PRINTER WIDTH
C172 90 01 2340 BCC SETCH :BRANCH IF NOT
C173 49 27 2350 ROR #27 :FORM 32-39 FOR BASIC LIST
C175 A8 2360 TAY :FORMATTING...
C176 2370 SETCH SEC
C177 2380 JMP EXIT2
C17A 2390 ;
C17B 2400 ;
C17C 2410 ;
C17D BD 00 04 2420 SEROUT1 LDA STATUS,X
C17E 2430 PFL OUT1 :CHECK FOR "AFTER KEYIN" FLAG
C181 90 00 04 2440 AND #17F :RESET "AFTER KEYIN" FLAG
C184 90 30 07 2450 LDA FLAGS,X
C188 A8 2470 LSR A :CARRY SET ON LINE FEED MODE
C189 49 00 2480 FOR #0BD :RECOVER CHARACTER
C18B 00 09 2490 BNE DONE :CHECK FOR CARRAGE RETURN
C190 90 04 05 2500 STA COL,X :FIX THE COLUMN COUNT
C191 49 0A 05 2510 BCC DONE :BRANCH IF NOT AUTO LF MODE
C192 90 30 07 2520 LDA #0BA :LOAD AND OUTPUT A LINE FEED
C193 49 0A 05 2530 BCS CTRL :BRANCH ALWAYS
C194 90 30 07 2540 LDA FLAGS,X :CHECK VIDEO BIT
C195 49 0A 05 2550 BCC MOVUI :NO VIDEO EXIT
C196 49 0A 05 2560 JMP VIDEO :OUTPUT VIDEO
C197 2570 ;
C198 2580 ;
C199 PD 30 07 2600 OUT1 LDA FLAGS,X
C1A1 14 2610 RPL CURSOK :DON'T MESS WITH THE CURSOR
C1A2 90 30 05 2620 LDA CH :IF THE VIDEO IS ON
C1A3 49 0A 05 2630 BCC COL,Y :BRANCH IF CH=COL
C1A4 90 00 00 2640 ROR CURSOK
C1A5 PD 09 2650 CMP #111 :GENERATE SPACES
C1A6 90 09 2660 BCC CURSOK
C1A7 49 0A 05 2670 ORA CURSOK :CHAR TO SHIFT-REG
C1A8 30 30 05 2680 AND COL,X :GO SHIFT IT OUT
C1A9 05 24 2690 ADC #0 :RECOVER CHARACTER
C1AB 90 30 05 2700 STA CH :CARRAGE RETURN CHECK
C1AC 05 24 2710 AND COL,X :BRANCH IF NOT CR
C1AD 05 24 2720 STA CH :RESET COLUMN COUNTERS
C1AE 05 24 2730 AND COL,X :MUST CHECK FOR DELAY OPTION
C1AF 05 24 2740 STA CH
C1B0 68 2750 CMP CH :TAB CHECK
C1B1 49 0A 05 2760 FLD CTRLST :RECOVER CHARACTER
C1B2 49 0A 05 2770 FLD CTRLST :BRANCH IF NO TAB
C1B3 49 0A 05 2780 FLD CTRLST :GENERATE SPACES
C1B4 49 0A 05 2790 BIT IORTS :DON'T COUNT CTRL CHAR
C1B5 49 0A 05 2800 INC COL,X :IN THE COL COUNT
C1B6 49 0A 05 2810 CRFL FMP :SAVE P-STATUS
C1B7 49 0A 05 2820 STA BYTE,X :CHAR TO SHIFT-REG
C1B8 49 0A 05 2830 JSR SHOUT :GO SHIFT IT OUT
C1B9 49 0A 05 2840 FLD #0BD :RECOVER CHARACTER
C1BA 49 0A 05 2850 EOR #0BD :CARRAGE RETURN CHECK
C1BB 49 0A 05 2860 BNE STOP :BRANCH IF NOT CR
C1BC 49 0A 05 2870 STA COL,X :RESET COLUMN COUNTERS
C1BD 49 0A 05 2880 STA CH :MUST CHECK FOR DELAY OPTION
C1BE 49 0A 05 2890 LDA FLAGS,X :MUST CHECK FOR DELAY OPTION
C1BF 29 40 FC 2900 AND #440 :MUST CHECK FOR DELAY OPTION
C1C0 00 03 2910 BNE LFCHK :MUST CHECK FOR DELAY OPTION
C1C1 20 40 FC 2920 JSR WAIT :MUST CHECK FOR DELAY OPTION
C1C2 49 0A 05 2930 FLA :GET FLAGS AGAIN
C1C3 49 0A 05 2940 LSR A :LOAD A LINE FEED
C1C4 90 30 04 2950 LDA #0BA :OUTPUT IT IF LINE-FEED MODE
C1C5 49 0A 05 2960 BCS CRFL :CHECK FOR COL-MAX
C1C6 49 0A 05 2970 LDA PWDTH,X :IF PWDTH=0 THEN NEVER MUX IN A CR.
C1C7 49 0A 05 2980 BEQ NITCHR :BRANCH ON COL < PWDTH
C1C8 49 0A 05 2990 SBC COL,X :MUST IN A CARRAGE RETURN
C1C9 49 0A 05 3000 BCC CRFL :BRANCH ALWAYS
C1CA 49 0A 05 3010 LDA #0BD :BRANCH ON TAB
C1CB 49 0A 05 3020 BCC CRFL :FINISH OUTPUT
C1CC 49 0A 05 3030 PFL DONE
C1CD 49 0A 05 3040 JMP PAGE
C1CE 49 0A 05 3050 ;
C1CF 49 0A 05 3060 ;
C1D0 49 0A 05 3070 ;
C1D1 49 0A 05 3080 ;
C1D2 49 0A 05 3090 ;
C1D3 49 0A 05 3095 ;
C1D4 49 0A 05 3100 ;
C1D5 49 0A 05 3105 ;
C1D6 49 0A 05 3110 ;
C1D7 49 0A 05 3115 ;
C1D8 49 0A 05 3120 ;
C1D9 49 0A 05 3125 ;
C1DA 49 0A 05 3130 ;
C1DB 49 0A 05 3135 ;
C1DC 49 0A 05 3140 ;
C1DD 49 0A 05 3145 ;
C1DE 49 0A 05 3150 ;
C1DF 49 0A 05 3155 ;
C1E0 49 0A 05 3160 ;
C1E1 49 0A 05 3165 ;
C1E2 49 0A 05 3170 ;
C1E3 49 0A 05 3175 ;
C1E4 49 0A 05 3180 ;
C1E5 49 0A 05 3185 ;
C1E6 49 0A 05 3190 ;
C1E7 49 0A 05 3195 ;
C1E8 49 0A 05 3200 ;
C1E9 49 0A 05 3205 ;
C1EA 49 0A 05 3210 ;
C1EB 49 0A 05 3215 ;
C1EC 49 0A 05 3220 ;
C1ED 49 0A 05 3225 ;
C1EE 49 0A 05 3230 ;
C1EF 49 0A 05 3235 ;
C1F0 49 0A 05 3240 ;
C1F1 49 0A 05 3245 ;
C1F2 49 0A 05 3250 ;
C1F3 49 0A 05 3255 ;
C1F4 49 0A 05 3260 ;
C1F5 49 0A 05 3265 ;
C1F6 49 0A 05 3270 ;
C1F7 49 0A 05 3275 ;
C1F8 49 0A 05 3280 ;
C1F9 49 0A 05 3285 ;
C1FA 49 0A 05 3290 ;
C1FB 49 0A 05 3295 ;
C1FC 49 0A 05 3300 ;
C1FD 49 0A 05 3305 ;
C1FE 49 0A 05 3310 ;
C1FF 49 0A 05 3315 ;
C200 49 0A 05 3320 ;
C201 49 0A 05 3325 ;
C202 49 0A 05 3330 ;
C203 49 0A 05 3335 ;
C204 49 0A 05 3340 ;
C205 49 0A 05 3345 ;
C206 49 0A 05 3350 ;
C207 49 0A 05 3355 ;
C208 49 0A 05 3360 ;
C209 49 0A 05 3365 ;
C20A 49 0A 05 3370 ;
C20B 49 0A 05 3375 ;
C20C 49 0A 05 3380 ;
C20D 49 0A 05 3385 ;
C20E 49 0A 05 3390 ;
C20F 49 0A 05 3395 ;
C210 49 0A 05 3400 ;
C211 49 0A 05 3405 ;
C212 49 0A 05 3410 ;
C213 49 0A 05 3415 ;
C214 49 0A 05 3420 ;
C215 49 0A 05 3425 ;
C216 49 0A 05 3430 ;
C217 49 0A 05 3435 ;
C218 49 0A 05 3440 ;

```

```

C21B 68 3450 CBR8 68
C21C 2A 3460 CBR10 2A
C21D 2A 3470 CBR10 2A
C21E 29 03 3480 CBR12 29
C21F 03 3490 CBR20 03
C220 B9 39 C9 3500 CBR21 B9
C221 9D 38 06 3510 CBR24 9D
C222 A9 02 3520 CBR27 A9
C223 9D 38 04 3530 CBR29 9D
C224 A9 09 06 3540 CBR2C A9
C225 9D 09 06 3550 CBR2E 9D
C226 A9 07 04 3560 CBR31 A9
C227 9D 08 04 3570 CBR33 9D
C228 E4 37 3580 CBR36 E4
C229 0D 09 3590 CBR3D 0D
C22A 35 36 3600 CBR3A 35
C22B 05 36 3610 CBR3C 05
C22C 35 36 3620 CBR3E 35
C22D 1B 36 3630 CBR40 1B
C22E 09 39 3640 CBR41 09
C22F E4 39 3650 CBR43 E4
C230 0D 09 3660 CBR45 0D
C231 A9 05 3670 CBR47 A9
C232 85 38 3680 CBR49 85
C233 38 3690 CBR4B 38
C234 68 3700 CBR4C 68
C235 68 3710 CBR4D 68
C236 35 3720 CBR4E 35
C237 07 3730 CBR4F 07
C238 07 3740 CBR50 07
C239 8D 78 07 3750 CBR54 8D
C23A 8D 78 07 3760 CBR55 8D
C23B 8D 78 06 3770 CBR58 8D
C23C AC FB 06 3780 CBR5D AC
C23D 38 06 3790 CBR60 38
C23E 08 06 3800 CBR61 08
C23F AC FB 06 3810 CBR63 AC
C240 C6 35 3820 CBR66 C6
C241 FD 40 3830 CBR69 FD
C242 8D 78 07 3840 CBR70 8D
C243 90 1F 3850 CBR76 90
C244 0D 00 3860 CBR77 0D
C245 10 F4 3870 CBR79 10
C246 9D 88 05 3880 CBR7E 9D
C247 2C 10 00 3890 CBR7A 2C
C248 68 3900 CBR7D 68
C249 60 3910 CBR7E 60
C24A 80 88 03 3920 CBR7F 80
C24B 00 09 3930 CBR82 00
C24C 35 07 3940 CBR84 35
C24D 0F 01 3950 CBR85 0F
C24E 0F 08 3960 CBR87 0F
C24F 08 05 3970 CBR88 08
C250 0D F5 3980 CBR8D 0D
C251 A8 3990 CBR8F A8
C252 2A 78 05 4000 CBR90 2A
C253 7E 88 05 4010 CBR91 7E
C254 4D 78 07 4020 CBR95 4D
C255 8D 78 07 4030 CBR98 8D
C256 8D 88 03 4040 CBR9E 8D
C257 88 01 4050 CBR9F 88
C258 0F 00 4060 CBA1 0F
C259 0D 09 4070 CBA3 0D
C25A 58 01 4080 CBA5 58
C25B 0F 05 4090 CBA6 0F
C25C 0F 05 4100 CBA8 0F
C25D 68 08 4110 CBAA 68
C25E 08 08 4120 CBA8 08
C25F 8D 78 07 4130 CBA0 8D
C260 4D 78 07 4140 CBA3 4D
C261 8D 88 03 4150 CBA5 8D
C262 8D 88 03 4160 CBA7 8D
C263 90 96 4170 CBB3 90
C264 8D 88 03 4180 CBB5 8D
C265 10 F8 4190 CBB8 10
C266 A9 0A 4200 CBA9 A9
C267 0D 88 06 4210 CBB0 0D
C268 88 08 4220 CBBF 88
C269 88 08 4230 CBB0 88
C26A 88 08 4240 CBBF 88
C26B 7E 88 05 4250 CBB7 7E
C26C 4D 78 07 4260 CBB9 4D
C26D 8D 88 03 4270 CBBE 8D
C26E 88 01 4280 CBBF 88
C26F 0F 00 4290 CBA3 0F
C270 58 01 4300 CBA5 58
C271 0F 05 4310 CBA6 0F
C272 0F 05 4320 CBA8 0F
C273 68 08 4330 CBAA 68
C274 08 08 4340 CBA8 08
C275 8D 78 07 4350 CBA0 8D
C276 4D 78 07 4360 CBA3 4D
C277 8D 88 03 4370 CBA5 8D
C278 10 F8 4380 CBB8 10
C279 A9 0A 4390 CBA9 A9
C27A 0D 88 06 4400 CBB0 0D
C27B 88 08 4410 CBBF 88
C27C 88 08 4420 CBB0 88
C27D 7E 88 05 4430 CBB7 7E
C27E 4D 78 07 4440 CBB9 4D
C27F 8D 88 03 4450 CBBE 8D
C280 88 01 4460 CBBF 88
C281 0F 00 4470 CBA3 0F
C282 58 01 4480 CBA5 58
C283 0F 05 4490 CBA6 0F
C284 0F 05 4500 CBA8 0F
C285 68 08 4510 CBAA 68
C286 08 08 4520 CBA8 08
C287 8D 78 07 4530 CBA0 8D
C288 4D 78 07 4540 CBA3 4D
C289 8D 88 03 4550 CBA5 8D
C28A 88 01 4560 CBBF 88
C28B 0F 00 4570 CBA3 0F
C28C 58 01 4580 CBA5 58
C28D 0F 05 4590 CBA6 0F
C28E 0F 05 4600 CBA8 0F
C28F 68 08 4610 CBAA 68
C290 08 08 4620 CBA8 08
C291 8D 78 07 4630 CBA0 8D
C292 4D 78 07 4640 CBA3 4D
C293 8D 88 03 4650 CBA5 8D
C294 10 F8 4660 CBB8 10
C295 A9 0A 4670 CBA9 A9
C296 0D 88 06 4680 CBB0 0D
C297 88 08 4690 CBBF 88
C298 88 08 4700 CBB0 88
C299 7E 88 05 4710 CBB7 7E
C29A 4D 78 07 4720 CBB9 4D
C29B 8D 88 03 4730 CBBE 8D
C29C 88 01 4740 CBBF 88
C29D 0F 00 4750 CBA3 0F
C29E 58 01 4760 CBA5 58
C29F 0F 05 4770 CBA6 0F
C29A 0F 05 4780 CBA8 0F
C29B 68 08 4790 CBAA 68
C29C 08 08 4800 CBA8 08
C29D 8D 78 07 4810 CBA0 8D
C29E 4D 78 07 4820 CBA3 4D
C29F 8D 88 03 4830 CBA5 8D
C29A 88 01 4840 CBBF 88
C29B 0F 00 4850 CBA3 0F
C29C 58 01 4860 CBA5 58
C29D 0F 05 4870 CBA6 0F
C29E 0F 05 4880 CBA8 0F
C29F 68 08 4890 CBAA 68
C29A 08 08 4900 CBA8 08
C29B 8D 78 07 4910 CBA0 8D
C29C 4D 78 07 4920 CBA3 4D
C29D 8D 88 03 4930 CBA5 8D
C29E 10 F8 4940 CBB8 10
C29F A9 0A 4950 CBA9 A9
C29A 0D 88 06 4960 CBB0 0D
C29B 88 08 4970 CBBF 88
C29C 88 08 4980 CBB0 88
C29D 7E 88 05 4990 CBB7 7E
C29E 4D 78 07 5000 CBB9 4D
C29F 8D 88 03 5010 CBBE 8D
C29A 88 01 5020 CBBF 88
C29B 0F 00 5030 CBA3 0F
C29C 58 01 5040 CBA5 58
C29D 0F 05 5050 CBA6 0F
C29E 0F 05 5060 CBA8 0F
C29F 68 08 5070 CBAA 68
C29A 08 08 5080 CBA8 08
C29B 8D 78 07 5090 CBA0 8D
C29C 4D 78 07 5100 CBA3 4D
C29D 8D 88 03 5110 CBA5 8D
C29E 10 F8 5120 CBB8 10
C29F A9 0A 5130 CBA9 A9
C29A 0D 88 06 5140 CBB0 0D
C29B 88 08 5150 CBBF 88
C29C 88 08 5160 CBB0 88
C29D 7E 88 05 5170 CBB7 7E
C29E 4D 78 07 5180 CBB9 4D
C29F 8D 88 03 5190 CBBE 8D
C29A 88 01 5200 CBBF 88
C29B 0F 00 5210 CBA3 0F
C29C 58 01 5220 CBA5 58
C29D 0F 05 5230 CBA6 0F
C29E 0F 05 5240 CBA8 0F
C29F 68 08 5250 CBAA 68
C29A 08 08 5260 CBA8 08
C29B 8D 78 07 5270 CBA0 8D
C29C 4D 78 07 5280 CBA3 4D
C29D 8D 88 03 5290 CBA5 8D
C29E 10 F8 5300 CBB8 10
C29F A9 0A 5310 CBA9 A9
C29A 0D 88 06 5320 CBB0 0D
C29B 88 08 5330 CBBF 88
C29C 88 08 5340 CBB0 88
C29D 7E 88 05 5350 CBB7 7E
C29E 4D 78 07 5360 CBB9 4D
C29F 8D 88 03 5370 CBBE 8D
C29A 88 01 5380 CBBF 88
C29B 0F 00 5390 CBA3 0F
C29C 58 01 5400 CBA5 58
C29D 0F 05 5410 CBA6 0F
C29E 0F 05 5420 CBA8 0F
C29F 68 08 5430 CBAA 68
C29A 08 08 5440 CBA8 08
C29B 8D 78 07 5450 CBA0 8D
C29C 4D 78 07 5460 CBA3 4D
C29D 8D 88 03 5470 CBA5 8D
C29E 10 F8 5480 CBB8 10
C29F A9 0A 5490 CBA9 A9
C29A 0D 88 06 5500 CBB0 0D
C29B 88 08 5510 CBBF 88
C29C 88 08 5520 CBB0 88
C29D 7E 88 05 5530 CBB7 7E
C29E 4D 78 07 5540 CBB9 4D
C29F 8D 88 03 5550 CBBE 8D
C29A 88 01 5560 CBBF 88
C29B 0F 00 5570 CBA3 0F
C29C 58 01 5580 CBA5 58
C29D 0F 05 5590 CBA6 0F
C29E 0F 05 5600 CBA8 0F
C29F 68 08 5610 CBAA 68
C29A 08 08 5620 CBA8 08
C29B 8D 78 07 5630 CBA0 8D
C29C 4D 78 07 5640 CBA3 4D
C29D 8D 88 03 5650 CBA5 8D
C29E 10 F8 5660 CBB8 10
C29F A9 0A 5670 CBA9 A9
C29A 0D 88 06 5680 CBB0 0D
C29B 88 08 5690 CBBF 88
C29C 88 08 5700 CBB0 88
C29D 7E 88 05 5710 CBB7 7E
C29E 4D 78 07 5720 CBB9 4D
C29F 8D 88 03 5730 CBBE 8D
C29A 88 01 5740 CBBF 88
C29B 0F 00 5750 CBA3 0F
C29C 58 01 5760 CBA5 58
C29D 0F 05 5770 CBA6 0F
C29E 0F 05 5780 CBA8 0F
C29F 68 08 5790 CBAA 68
C29A 08 08 5800 CBA8 08
C29B 8D 78 07 5810 CBA0 8D
C29C 4D 78 07 5820 CBA3 4D
C29D 8D 88 03 5830 CBA5 8D
C29E 10 F8 5840 CBB8 10
C29F A9 0A 5850 CBA9 A9
C29A 0D 88 06 5860 CBB0 0D
C29B 88 08 5870 CBBF 88
C29C 88 08 5880 CBB0 88
C29D 7E 88 05 5890 CBB7 7E
C29E 4D 78 07 5900 CBB9 4D
C29F 8D 88 03 5910 CBBE 8D
C29A 88 01 5920 CBBF 88
C29B 0F 00 5930 CBA3 0F
C29C 58 01 5940 CBA5 58
C29D 0F 05 5950 CBA6 0F
C29E 0F 05 5960 CBA8 0F
C29F 68 08 5970 CBAA 68
C29A 08 08 5980 CBA8 08
C29B 8D 78 07 5990 CBA0 8D
C29C 4D 78 07 6000 CBA3 4D
C29D 8D 88 03 6010 CBA5 8D
C29E 10 F8 6020 CBB8 10
C29F A9 0A 6030 CBA9 A9
C29A 0D 88 06 6040 CBB0 0D
C29B 88 08 6050 CBBF 88
C29C 88 08 6060 CBB0 88
C29D 7E 88 05 6070 CBB7 7E
C29E 4D 78 07 6080 CBB9 4D
C29F 8D 88 03 6090 CBBE 8D
C29A 88 01 6100 CBBF 88
C29B 0F 00 6110 CBA3 0F
C29C 58 01 6120 CBA5 58
C29D 0F 05 6130 CBA6 0F
C29E 0F 05 6140 CBA8 0F
C29F 68 08 6150 CBAA 68
C29A 08 08 6160 CBA8 08
C29B 8D 78 07 6170 CBA0 8D
C29C 4D 78 07 6180 CBA3 4D
C29D 8D 88 03 6190 CBA5 8D
C29E 10 F8 6200 CBB8 10
C29F A9 0A 6210 CBA9 A9
C29A 0D 88 06 6220 CBB0 0D
C29B 88 08 6230 CBBF 88
C29C 88 08 6240 CBB0 88
C29D 7E 88 05 6250 CBB7 7E
C29E 4D 78 07 6260 CBB9 4D
C29F 8D 88 03 6270 CBBE 8D
C29A 88 01 6280 CBBF 88
C29B 0F 00 6290 CBA3 0F
C29C 58 01 6300 CBA5 58
C29D 0F 05 6310 CBA6 0F
C29E 0F 05 6320 CBA8 0F
C29F 68 08 6330 CBAA 68
C29A 08 08 6340 CBA8 08
C29B 8D 78 07 6350 CBA0 8D
C29C 4D 78 07 6360 CBA3 4D
C29D 8D 88 03 6370 CBA5 8D
C29E 10 F8 6380 CBB8 10
C29F A9 0A 6390 CBA9 A9
C29A 0D 88 06 6400 CBB0 0D
C29B 88 08 6410 CBBF 88
C29C 88 08 6420 CBB0 88
C29D 7E 88 05 6430 CBB7 7E
C29E 4D 78 07 6440 CBB9 4D
C29F 8D 88 03 6450 CBBE 8D
C29A 88 01 6460 CBBF 88
C29B 0F 00 6470 CBA3 0F
C29C 58 01 6480 CBA5 58
C29D 0F 05 6490 CBA6 0F
C29E 0F 05 6500 CBA8 0F
C29F 68 08 6510 CBAA 68
C29A 08 08 6520 CBA8 08
C29B 8D 78 07 6530 CBA0 8D
C29C 4D 78 07 6540 CBA3 4D
C29D 8D 88 03 6550 CBA5 8D
C29E 10 F8 6560 CBB8 10
C29F A9 0A 6570 CBA9 A9
C29A 0D 88 06 6580 CBB0 0D
C29B 88 08 6590 CBBF 88
C29C 88 08 6600 CBB0 88
C29D 7E 88 05 6610 CBB7 7E
C29E 4D 78 07 6620 CBB9 4D
C29F 8D 88 03 6630 CBBE 8D
C29A 88 01 6640 CBBF 88
C29B 0F 00 6650 CBA3 0F
C29C 58 01 6660 CBA5 58
C29D 0F 05 6670 CBA6 0F
C29E 0F 05 6680 CBA8 0F
C29F 68 08 6690 CBAA 68
C29A 08 08 6700 CBA8 08
C29B 8D 78 07 6710 CBA0 8D
C29C 4D 78 07 6720 CBA3 4D
C29D 8D 88 03 6730 CBA5 8D
C29E 10 F8 6740 CBB8 10
C29F A9 0A 6750 CBA9 A9
C29A 0D 88 06 6760 CBB0 0D
C29B 88 08 6770 CBBF 88
C29C 88 08 6780 CBB0 88
C29D 7E 88 05 6790 CBB7 7E
C29E 4D 78 07 6800 CBB9 4D
C29F 8D 88 03 6810 CBBE 8D
C29A 88 01 6820 CBBF 88
C29B 0F 00 6830 CBA3 0F
C29C 58 01 6840 CBA5 58
C29D 0F 05 6850 CBA6 0F
C29E 0F 05 6860 CBA8 0F
C29F 68 08 6870 CBAA 68
C29A 08 08 6880 CBA8 08
C29B 8D 78 07 6890 CBA0 8D
C29C 4D 78 07 6900 CBA3 4D
C29D 8D 88 03 6910 CBA5 8D
C29E 10 F8 6920 CBB8 10
C29F A9 0A 6930 CBA9 A9
C29A 0D 88 06 6940 CBB0 0D
C29B 88 08 6950 CBBF 88
C29C 88 08 6960 CBB0 88
C29D 7E 88 05 6970 CBB7 7E
C29E 4D 78 07 6980 CBB9 4D
C29F 8D 88 03 6990 CBBE 8D
C29A 88 01 7000 CBBF 88
C29B 0F 00 7010 CBA3 0F
C29C 58 01 7020 CBA5 58
C29D 0F 05 7030 CBA6 0F
C29E 0F 05 7040 CBA8 0F
C29F 68 08 7050 CBAA 68
C29A 08 08 7060 CBA8 08
C29B 8D 78 07 7070 CBA0 8D
C29C 4D 78 07 7080 CBA3 4D
C29D 8D 88 03 7090 CBA5 8D
C29E 10 F8 7100 CBB8 10
C29F A9 0A 7110 CBA9 A9
C29A 0D 88 06 7120 CBB0 0D
C29B 88 08 7130 CBBF 88
C29C 88 08 7140 CBB0 88
C29D 7E 88 05 7150 CBB7 7E
C29E 4D 78 07 7160 CBB9 4D
C29F 8D 88 03 7170 CBBE 8D
C29A 88 01 7180 CBBF 88
C29B 0F 00 7190 CBA3 0F
C29C 58 01 7200 CBA5 58
C29D 0F 05 7210 CBA6 0F
C29E 0F 05 7220 CBA8 0F
C29F 68 08 7230 CBAA 68
C29A 08 08 7240 CBA8 08
C29B 8D 78 07 7250 CBA0 8D
C29C 4D 78 07 7260 CBA3 4D
C29D 8D 88 03 7270 CBA5 8D
C29E 10 F8 7280 CBB8 10
C29F A9 0A 7290 CBA9 A9
C29A 0D 88 06 7300 CBB0 0D
C29B 88 08 7310 CBBF 88
C29C 88 08 7320 CBB0 88
C29D
```

```

C8CE 09 80      4680      ORA      #480
C8D0 49 0E      4690      EOR      #40
C8D4 90 02      4700      CMP      #1F
C8D8 48 0E      4710      BCC      MXTOUT
C8DB 28 0E      4720      EOR      #40
C8D9 4C FD FD   MXTOUT    FLP      #40
C8DC          4730      JMP      COUT1
C8DE 91 28      4740      ;
C8E1 09 05      4750      INFINSH STA      (BASL).Y ;REPLACE FLASHING CURSOR
C8E3 C9 95      4760      LDA      BYTE.X ;GET MOST RECENT INPUT
C8E5 D0 02      4770      ORA      #80
C8E9 C9 E0      4780      CMP      #1FICK ;SCREEN PICK (CONTROL-U)?
C8EB 90 11      4790      LDA      (BASL).Y ;GET SCREEN CHARACTER
C8ED C0 02      4800      BCC      NOTPICK ;DO NOTHING TO UPPER AN LC-NALFA
C8EF C9 E0      4810      AND      CASE ;MAKE LOWER TO UPPER IF LC NOT ENABLED
C8F1 90 0E      4820      CMP      #40 ;RETURN LOWER CASE IF ENABLED
C8F3 A4 37      4830      AND      CASE
C8F5 C0 FD      4840      AND      CASE
C8F7 FD 03      4850      AND      CASE
C8F9 29 21      4860      LDY      CSWH ;IF VIDEO OUT ONLY, THEN CHAR
C8FB AC 58 FF   IVERSE   ORG      #40 ; SHOULD BE DISPLAYED INVERSE.
C8FD 29 1F      4870      AND      #47F ; OTHERWISE JUST STRIP BIT 7
C8FF 80 78 06   IVERSE   LDA      IORTS ; (DUMMY LDY ABSOLUTE)
C901 00 00      4880      AND      #1F
C903 00 00      4890      STA      OLDDBYTE ;MAKE IT INVERSE VIDEO
C905 00 00      4900      PLS      #2 ;IF CARRY CLR, DO "ESC" FUNCTION
C907 00 00      4910      BCS      NOTESC ;IS IT AN ASCII "L"?
C909 00 00      4920      BMI      UPPER ;CAPSLOCK COMMAND IF >L
C90B 00 00      4930      BNE      NOTESC ;ENABLE LOWER CASE
C90D 00 00      4940      LDA      #20 ;BRANCH ALWAYS TAKEN
C90F 00 00      4950      ORA      #0 ;DISABLE LOWER CASE
C911 00 00      4960      AND      #0
C913 00 00      4970      AND      #0
C915 00 00      4980      AND      #0
C917 00 00      4990      AND      #0
C919 00 00      5000      AND      #0
C91B 00 00      5010      AND      #0
C91D 00 00      5020      AND      #0
C91F 00 00      5030      AND      #0
C921 00 00      5040      AND      #0
C923 00 00      5050      AND      #0
C925 00 00      5060      AND      #0
C927 00 00      5070      AND      #0
C929 00 00      5080      AND      #0
C92B 00 00      5090      AND      #0
C92D 00 00      5100      AND      #0
C92F 00 00      5110      AND      #0
C931 00 00      5120      AND      #0
C933 00 00      5130      AND      #0
C935 00 00      5140      AND      #0
C937 00 00      5150      AND      #0
C939 00 00      5160      AND      #0
C93B 00 00      5170      AND      #0
C93D 00 00      5180      AND      #0
C93F 00 00      5190      AND      #0
C941 00 00      5200      AND      #0
C943 00 00      5210      AND      #0
C945 00 00      5220      AND      #0
C947 00 00      5230      AND      #0
C949 00 00      5240      AND      #0
C94B 00 00      5250      AND      #0
C94D 00 00      5260      AND      #0
C94F 00 00      5270      AND      #0
C951 00 00      5280      AND      #0
C953 00 00      5290      AND      #0
C955 00 00      5300      AND      #0
C957 00 00      5310      AND      #0
C959 00 00      5320      AND      #0
C95B 00 00      5330      AND      #0
C95D 00 00      5340      AND      #0
C95F 00 00      5350      AND      #0
C961 00 00      5360      AND      #0
C963 00 00      5370      AND      #0
C965 00 00      5380      AND      #0
C967 00 00      5390      AND      #0
C969 00 00      5400      AND      #0
C96B 00 00      5410      AND      #0
C96D 00 00      5420      AND      #0
C96F 00 00      5430      AND      #0
C971 00 00      5440      AND      #0
C973 00 00      5450      AND      #0
C975 00 00      5460      AND      #0
C977 00 00      5470      AND      #0
C979 00 00      5480      AND      #0
C97B 00 00      5490      AND      #0
C97D 00 00      5500      AND      #0
C97F 00 00      5510      AND      #0
C981 00 00      5520      AND      #0
C983 00 00      5530      AND      #0
C985 00 00      5540      AND      #0
C987 00 00      5550      AND      #0
C989 00 00      5560      AND      #0
C98B 00 00      5570      AND      #0
C98D 00 00      5580      AND      #0
C98F 00 00      5590      AND      #0
C991 00 00      5600      AND      #0
C993 00 00      5610      AND      #0
C995 00 00      5620      AND      #0
C997 00 00      5630      AND      #0
C999 00 00      5640      AND      #0
C99B 00 00      5650      AND      #0
C99D 00 00      5660      AND      #0
C99F 00 00      5670      AND      #0
C9A1 00 00      5680      AND      #0
C9A3 00 00      5690      AND      #0
C9A5 00 00      5700      AND      #0
C9A7 00 00      5710      AND      #0
C9A9 00 00      5720      AND      #0
C9AB 00 00      5730      AND      #0
C9AD 00 00      5740      AND      #0
C9AF 00 00      5750      AND      #0
C9B1 00 00      5760      AND      #0
C9B3 00 00      5770      AND      #0
C9B5 00 00      5780      AND      #0
C9B7 00 00      5790      AND      #0
C9B9 00 00      5800      AND      #0
C9BB 00 00      5810      AND      #0
C9BD 00 00      5820      AND      #0
C9BF 00 00      5830      AND      #0
C9C1 00 00      5840      AND      #0
C9C3 00 00      5850      AND      #0
C9C5 00 00      5860      AND      #0
C9C7 00 00      5870      AND      #0
C9C9 00 00      5880      AND      #0
C9CB 00 00      5890      AND      #0
C9CD 00 00      5900      AND      #0
C9CF 00 00      5910      AND      #0
C9D1 00 00      5920      AND      #0
C9D3 00 00      5930      AND      #0
C9D5 00 00      5940      AND      #0
C9D7 00 00      5950      AND      #0
C9D9 00 00      5960      AND      #0
C9DB 00 00      5970      AND      #0
C9DD 00 00      5980      AND      #0
C9DF 00 00      5990      AND      #0
C9E1 00 00      6000      AND      #0
C9E3 00 00      6010      AND      #0
C9E5 00 00      6020      AND      #0
C9E7 00 00      6030      AND      #0
C9E9 00 00      6040      AND      #0
C9EB 00 00      6050      AND      #0
C9ED 00 00      6060      AND      #0
C9EF 00 00      6070      AND      #0
C9F1 00 00      6080      AND      #0
C9F3 00 00      6090      AND      #0
C9F5 00 00      6100      AND      #0
C9F7 00 00      6110      AND      #0
C9F9 00 00      6120      AND      #0
C9FB 00 00      6130      AND      #0
C9FD 00 00      6140      AND      #0
C9FF 00 00      6150      AND      #0
CA00          6160      AND      #0

```

```

C9B1          6170      AND      #0
C9B3          6180      AND      #0
C9B5          6190      AND      #0
C9B7          6200      AND      #0
C9B9          6210      AND      #0
C9BB          6220      AND      #0
C9BD          6230      AND      #0
C9BF          6240      AND      #0
C9C1          6250      AND      #0
C9C3          6260      AND      #0
C9C5          6270      AND      #0
C9C7          6280      AND      #0
C9C9          6290      AND      #0
C9CB          6300      AND      #0
C9CD          6310      AND      #0
C9CF          6320      AND      #0
C9D1          6330      AND      #0
C9D3          6340      AND      #0
C9D5          6350      AND      #0
C9D7          6360      AND      #0
C9D9          6370      AND      #0
C9DB          6380      AND      #0
C9DD          6390      AND      #0
C9DF          6400      AND      #0
C9E1          6410      AND      #0
C9E3          6420      AND      #0
C9E5          6430      AND      #0
C9E7          6440      AND      #0
C9E9          6450      AND      #0
C9EB          6460      AND      #0
C9ED          6470      AND      #0
C9EF          6480      AND      #0
C9F1          6490      AND      #0
C9F3          6500      AND      #0
C9F5          6510      AND      #0
C9F7          6520      AND      #0
C9F9          6530      AND      #0
C9FB          6540      AND      #0
C9FD          6550      AND      #0
C9FF          6560      AND      #0
CA00          6570      AND      #0

```

```

LABELS
003C AIL
C9E1 BDLY1
C9E2 BDLY2
C9E3 BDLY3
C9E4 BDLY4
C9E5 BDLY5
C9E6 BDLY6
C9E7 BDLY7
C9E8 BDLY8
C9E9 BDLY9
C9EA BDLY10
C9EB BDLY11
C9EC BDLY12
C9ED BDLY13
C9EE BDLY14
C9EF BDLY15
C9F0 BDLY16
C9F1 BDLY17
C9F2 BDLY18
C9F3 BDLY19
C9F4 BDLY20
C9F5 BDLY21
C9F6 BDLY22
C9F7 BDLY23
C9F8 BDLY24
C9F9 BDLY25
C9FA BDLY26
C9FB BDLY27
C9FC BDLY28
C9FD BDLY29
C9FE BDLY30
C9FF BDLY31
CA00 BDLY32

```