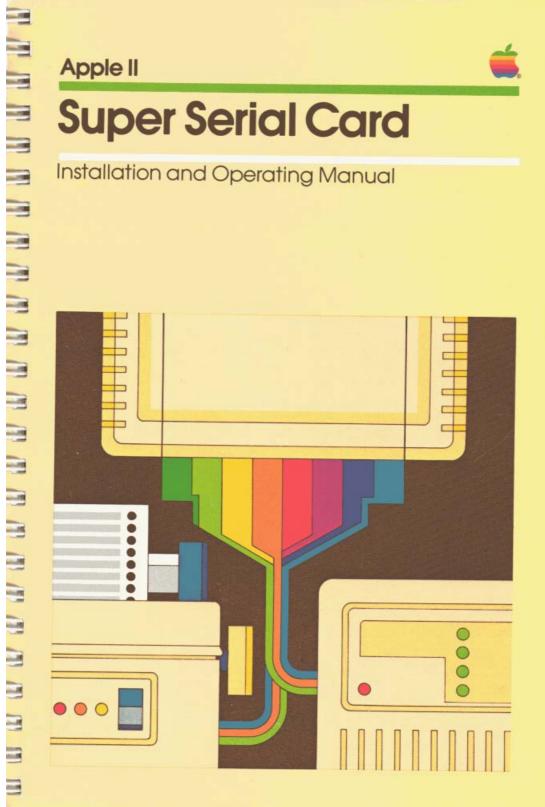
Apple II



# **Super Serial Card**

Installation and Operating Manual



#### **Notice**

Apple Computer Inc. reserves the right to make improvements in the product described in this manual at any time and without notice.

#### **Disclaimer of All Warranties And Liability**

Apple Computer Inc. makes no warranties, either express or implied, with respect to this manual or with respect to the software described in this manual, its quality, performance, merchantability, or fitness for any particular purpose. Apple Computer Inc. software is sold or licensed "as is." The entire risk as to its quality and performance is with the buyer. Should the programs prove defective following their purchase, the buyer (and not Apple Computer Inc., its distributor, or its retailer) assumes the entire cost of all necessary servicing, repair, or correction and any incidental or consequential damages. In no event will Apple Computer Inc. be liable for direct, indirect, incidental, or consequential damages resulting from any defect in the software, even if Apple Computer Inc. has been advised of the possibility of such damages. Some states do not allow the exclusion or limitation of implied warranties or liability for incidental or consequential damages, so the above limitation or exclusion may not apply to you.

44

44

44

44

dil.

44

4

44

44

-

48

d.

a.

C

(

This manual is copyrighted. All rights are reserved. This document may not, in whole or part, be copied, photocopied, reproduced, translated or reduced to any electronic medium or machine readable form without prior consent, in writing, from Apple Computer Inc.

© 1981 by Apple Computer Inc. 10260 Bandley Drive Cupertino, California 95014 (408) 996-1010

The word Apple and the Apple logo are registered trademarks of Apple Computer Inc.

Reorder Apple Product #A2L0044

WARNING: This equipment has been certified to comply with the limits for a Class B computing device, pursuant to Subpart J of Part 15 of FCC Rules. Only peripherals (computer input/output devices, terminals, printers, etc.) certified to comply with the Class B limits may be attached to this computer. Operation with non-certified peripherals is likely to result in interference to radio and TV reception.

# Apple II

# **Super Serial Card**

1

nade.

1

4

-12

100

- id

-

1

200

- 195

-19

-195

-

Installation and Operating Manual

Please read this manual before attempting to install the Super Serial Card in the Apple Computer. Incorrect installation could cause permanent damage to both the Super Serial Card and the Apple.

## RADIO AND TELEVISION INTERFERENCE

The equipment described in this manual generates and uses radio frequency energy. If it is not installed and used properly, that is in strict accordance with our instructions, it may cause interference to radio and television reception.

41

41

44

SIL.

al.

41.

Ril.

M.A.

This equipment has been tested and complies with the limits for a Class B computing device in accordance with the specifications in Subpart J of Part 15 of FCC rules. These rules are designed to provide reasonable protection against such interference in a residential installation. However, there is no guarantee that the interference will not occur in a particular installation.

You can determine whether your computer is causing interference by turning it off. If the interference stops, it was probably caused by the computer. If your computer does cause interference to radio or television reception, you can try to correct the interference by using one or more of the following measures:

- Turn the TV or radio antenna until the interference stops.
- Move the computer to one side or the other of the TV or radio.
- Move the computer farther away from the TV or radio.
- Plug the computer into an outlet that is on a different circuit from the TV or radio. (That is, make certain the computer and the TV or radio are on circuits controlled by different circuit breakers or fuses.)

If necessary, you should consult your dealer or an experienced radio/television technician for additional suggestions. You may find the following booklet prepared by the Federal Communications Commission helpful:

"How to Identify and Resolve Radio-TV Interference Problems"

This booklet is available from the U.S. Government Printing Office, Washington, DC 20402, Stock number 004-000-00345-4.

# TABLE OF CONTENTS

-0.86

110

12.00

I IN	EFACE	
Ol-		
	apter 1	
GE	TTING STARTED	
	H. Constitution	
1 2	Unpacking A Close Look	
	Preparing Cable and Clamp Assembly	
3	Attaching Internal Cable to SSC	
Ch	apter 2	
	INTER MODE	
FR	INTER MODE	
5	Preparing the SSC for Printer Mode	
6	Setting the Switches	
6	Commonly Used Settings	
6	Baud Rate	
7	Data Format and Parity	
7	Carriage Return Delay	
8	Line Width and Video On/Off	
8	Generate <lf> Out</lf>	
8	Special Switches	
9	Installation Procedure	
10	External Cable and Connector	
11	Using the SSC in Printer Mode With a Printer	
11	With a Terminal	
11	Printer Mode Commands	
12	Command Formats	
12	The Command Character	
13	Printer Mode Command Summary	
15	Commands That Change Switch Settings	
15	Baud Rate <n>B</n>	
15	Data Format <n>D</n>	
15	Parity <n>P</n>	
16	Set Time Delay <n>C, <n>L, <n>F</n></n></n>	
17	Generate <cr> On Column OverflowC</cr>	
17	Generate (LF) OutL (E/D)	
17 17	Mask (Suppress) <lf> InM_<e d=""> Reset the SSCR</e></lf>	
11	weder rue pag. V	

17	Other Commands	and the second
18	Translate Lowercase Characters(n)T	
18	Zap (Suppress) Control CharactersZ	-
18 18	Find KeyboardF_ <e d=""> XOFF RecognitionX <e d=""></e></e>	
19	Tab in BASICT <e d=""></e>	
17	Ido III bhold I_dijo/	all.
		44
Ch	apter 3	
C	OMMUNICATIONS MODE	21
0.00	9 C 10 5070 N N N 10 10 100 10	
21	Preparing the SSC for Communications Mode	44
22	Setting the Switches	_
22	Commonly Used Settings Baud Rate	46.
23	Data Format and Parity	
24	Generate <lf> Out</lf>	
24	Special Switches	44.
24	Installation Procedure	_
26	External Cable and Connector	
26	Using the SSC in Communications Mode	-
27	Communications Mode Commands	
27	Command Formats	44.
28	The Command Character	
28	Communications Mode Command Summary	66.
30	Commands That Change Switch Settings	
30	Baud Rate <n>B</n>	_
30	Data Format <n>D</n>	<b>EEL.</b>
30	Parity <n>P</n>	
31	Generate <lf> OutL <e d=""></e></lf>	wit.
31	Mask (Suppress) <lf> InM <e d=""></e></lf>	
31	Reset the SSCR	_
31	Other Commands	68.
32	Set Time Delays <n>C, <n>L, <n>F</n></n></n>	
32	Translate Lowercase Characters <n>T</n>	Sid.
33	Zap (Suppress) Control CharactersZ	_
33	Find KeyboardF_ <e d=""></e>	-
33	XOFF RecognitionX <e d=""></e>	E16.
34	Specify Screen Slot <n>S</n>	
34	Echo Characters on the ScreenE <e d=""></e>	<b>E</b>
34	Terminal Mode	_
35	Terminal Mode Commands	100
35	Enter Terminal ModeT	
36	Transmit a Break SignalB	
36	Special CharactersS <e d=""></e>	
36	Quit (Exit from) Terminal ModeQ	_
36	A Terminal Mode Example	
		well.
		_
		_
iv	SUPER SERIAL CARD	
5.50		

Sa.

- 18		
-08	Chapter 4	
	HOW THE SSC WORKS	37
114	37 Serial Data Communication 38 Parallel Data in the Apple II	
	39 Serial Data for Long Distances	
100	40 RS-232-C Data Formats	
_	40 RS-232-C Signals 41 Modems	
	42 Modem Eliminators 43 SSC Modes and Configurations	
100	43 SSC Modes and Configurations 45 Theory of Operation	
	46 Addressing and Control Logic	
-158	46 ROM/RAM Space 47 Registers in Peripheral I/O Space	
	47 The ACIA	
-18	48 Data Input and Output 48 Data Bus	
	48 Jumper Block	
-108		
- 18	Appendix A	
	FIRMWARE	
189	FIRMWARE	49
- int	49 Pascal 1.1 Firmware Protocol 50 I/O Routine Entry Points	
	51 Device Identification	
	52 SSC Firmware Memory Usage	
	52 Zero Page Locations 53 Scratchpad RAM Locations	
- inte	54 Peripheral Card I/O Space	
-	55 SSC Entry Points 55 Nonitor ROM Entry Points	
100	55 Monitor ROM Entry Points 55 BASIC Entry Points	
- 100	56 Pascal 1.0 Entry Points	
-	56 Pascal I.1 Entry Points 57 Other Special Firmware Locations	
1000	58 SSC Firmware Listings	
-		
1000	Appendix B	
-100	APPLE INTERFACE CARD EMULATION	91
- 25	91 Old Serial Interface Card Emulation	
	92 P8 Emulation POKEs	
	94 P8A Emulation POKES 95 Other Emulation Mode Differences	
100		

1-28

96 Old Communications Card Commands	
96 Switch to Terminal Mode <ctrl-t> 96 Bypass Terminal Mode<ctrl-r></ctrl-r></ctrl-t>	
96 XOFF <ctrl-s> 96 Parallel Card Commands</ctrl-s>	
96 Line Width n and Video Off <ctrl-i><n>N 96 Line Width 40 and Video On<ctrl-i>I 96 Disable Automatic Linefeed<ctrl-i>K</ctrl-i></ctrl-i></n></ctrl-i>	
Appendix C	
Appendix C SPECIFICATIONS AND SCHEMATICS	97
97 SSC Specifications 98 Connector Pin Assignments 99 Jumper Block Wiring 100 Schematic Diagram	
Appendix D	
ASCII CODE TABLE	101
Appendix E	
TROUBLESHOOTING HINTS	105
Appendix F	
ERROR CODES	109
GLOSSARY	111
FIGURES AND TABLES	119
INDEX	123
vi SUPER SERIAL CARD	

# **PREFACE**

-cHE

- 140

uelli.

46

The Super Serial Card (SSC) provides a two-way serial interface to a wide variety of devices, including printers, terminals, plotters, and other computers. All these devices can be connected to the SSC either directly or via modem.

The SSC replaces both the P8 and P8A variety of Apple II Serial Interface Card, although it does not manipulate all specific Apple II memory locations in the same way. The SSC also replaces the Apple II Communications Card, and supports Terminal Mode. Finally, the SSC supports Apple II parallel interface card software commands.

The Super Serial Card conforms to the Electronic Industries Association (EIA) interface definitions A through E. (To obtain a copy of the EIA RS-232-C Standard, write to the EIA Engineering Department, Electronics Industries Association, 2001 Eye Street, N.W., Washington, D.C. 20006.)

The SSC can be configured to the attached external device in three ways: (1) by setting switches on the card itself, (2) by typing in commands at the keyboard under the Monitor, Integer BASIC, Applesoft or DOS, or (3) by issuing commands from assembly language, BASIC or Pascal programs. The SSC can be configured and operated by programs in Integer BASIC, APPLESOFT, Pascal, and assembly language.

How you prepare, install and use the Super Serial Card depends on what you connect to it:

- Read Chapter 1 for unpacking and cable clamp preparation instructions.
- If you are going to connect a printer, terminal or some other device directly to the SSC, then read the first four sections of Chapter 2. (Many commonly used switch settings are listed in Table 2-1 for your convenience.) You only need to read the section Printer Mode Commands of Chapter 2 if you need special commands to change the SSC's characteristics.
- If you are going to connect a device to the SSC via a modem or similar communications equipment, then read the first four sections of Chapter 3. (Switch settings for many Communications Mode applications are listed in Table 3-1.) You only need to read the section Communications Mode Commands of Chapter 3 if you need special commands to change the SSC's characteristics.
- If you want to use the Apple II as an unintelligent terminal connected via a modem, read the section Terminal Mode of Chapter 3.
- Troubleshooting Hints are discussed in Appendix E.

The SSC also emulates ("imitates") the Apple II Serial Interface Card (both the P8 and P8A varieties), and supports many of the software commands used by the Apple II parallel printer interface card and the Apple II Communications Card. These are all discussed in Appendix B.

Chapter 4 explains how the SSC works, both in everyday terms (Serial Data Communication Simply Explained) and from an engineering viewpoint (Theory of Operation). The Theory of Operation section is keyed to the schematic diagram in Appendix C. Chapter 4 also contains a section on SSC modes and configurations.

al.

ES.

-

44

44.

all.

Appendix A discusses SSC firmware and its entry points in the SSC ROM, as well as the Apple II memory locations the firmware uses.

Appendix C contains SSC specifications and connector pin assignments, and its schematic diagram.

Appendix D lists the ASCII codes and their equivalents. Appendix E has troubleshooting hints. Appendix F explains the SSC error codes.

A glossary explains the meaning of most important terms as they apply to the  $\ensuremath{\mathsf{SSC}}_{\:\raisebox{1pt}{\text{\circle*{1.5}}}}$ 

The Reference Card summarizes the switch settings and commands for the SSC Printer Mode and Communications Mode.

There are three symbols that set off information of special importance:



This symbol points to a paragraph that contains especially useful information.



Watch out! This symbol precedes a paragraph that warns you to be careful.



This symbol precedes a warning that you are about to harm hardware or destroy data.

# CHAPTER 1 GETTING STARTED

This chapter takes you through the first steps of getting acquainted with your Super Serial Card (SSC). After unpacking the SSC and examining it, you will assemble the short internal cable (if it is not already assembled) that connects the  $1\emptyset$ -pin cable socket on the SSC to the 25-pin socket at the back of the Apple II case.

# UNPACKING

-0.00

or did

-

As you unpack your Super Serial Card (Figure 1-1), check the contents against the items described on the packing list.

Fill out the pre-addressed warranty card and mail it in. If any items are missing, contact the dealer you purchased the SSC from.

You will need a shielded external cable (not provided as part of the SSC package) to connect the external device—the printer, modem, terminal, or other computer—to your Apple II. Suitable cables are available through your Apple dealer.

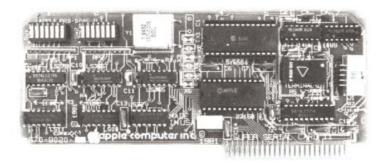


Figure 1-1. Photo of the Super Serial Card

# A CLOSE LOOK

Let's examine the Super Serial Card for a moment. Pick up the SSC carefully and put it on a flat surface oriented as shown in Figure 1-1. Now use Figure 1-2 to help identify the chief parts of the SSC. Those that you will have to deal with as you prepare it for installation are:

 The jumper block. This ordinarily points toward the word TERMINAL; if you attach a modem to the SSC, you will turn this around so the arrow points toward the word MODEM (Chapter 3). Ed.

E.E.

Rifferin

- The switches. The left group is numbered from SW1-1 through SW1-7; the right group is numbered from SW2-1 through SW2-7. You can see the characters "SW1" and "SW2" printed on the SSC.
- The edge connector. It is important not to touch the gold fingers on this connector: they must make a clean electrical contact in the Apple II connector slot when you install the SSC (Chapter 2 or Chapter 3).
- The <u>cable socket</u>. The next section of this chapter explains how to install the short internal cable between the SSC and the Apple II case.

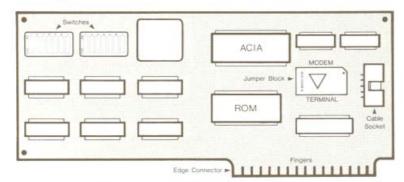


Figure 1-2. Line Drawing of the SSC

# PREPARING CABLE AND CLAMP ASSEMBLY

Before preparing and installing the SSC, you may need to prepare the clamp assembly for the internal cable that will go from the SSC to the back of the Apple II's case. The components of this clamp assembly are shown in Figure 1-3. If these components are already assembled, skip to the next section, Attaching the Internal Cable to the SSC.

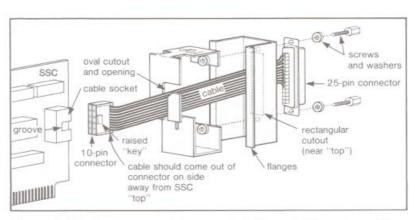


Figure 1-3. Components of Internal Cable and Clamp Assembly

Lay the short cable down as shown in Figure 1-3. Pick up the clamp piece that has the word TOP stamped on one end. Hold this clamp piece with the word TOP facing away from you, and the oval cutout toward the smaller connector on the cable. Bend the cable slightly, and insert it into the oval cutout through the opening; then straighten the cable in the cutout so that it moves easily.

The other clamp piece has flanges (Figure 1-3) and a rectangular opening that is closer to one end (its top end) than to the other. Hold this clamp piece with its top end away from you and its flanges facing the 25-pin connector end of the cable. Then tilt the connector and feed it completely through the rectangular cutout.

Now slide the two clamp pieces all the way down the cable until they are right up against the 25-pin connector, and their screw holes line up with the connector's screw holes. Slide the washers onto the screws and then thread the screws a couple of turns into the lined-up holes. Don't screw them in very far.

# ATTACHING INTERNAL CABLE TO SCC

This step in the preparation of your Super Serial Card is simple to do, but you must do it carefully.



It is very important to connect the cable to the SSC correctly. Improper connection of the cable to the SSC may result in damage to the Apple and the SSC; such damage is NOT covered by your warranty.

Lay the SSC down on a flat surface, component-side up and gold fingers at the lower right. Examine the  $1\emptyset$ -pin end of the cable: the wires come out of the SIDE of the connector-the same side as the raised "key" in the plastic (Figure 1-3). Hold the connector so

the wires are on the side away from the SSC, and insert the connector firmly into the cable socket along the right edge of the SSC. The raised "key" should slide into the groove in the cable socket (Figure 1-4).



If the cable is now jammed between the  $1\emptyset$ -pin cable socket and the SSC board, the connector is plugged in backwards. Unplug the connector and reconnect it so that the cable is on the side AWAY from the SSC (Figure 1-5).

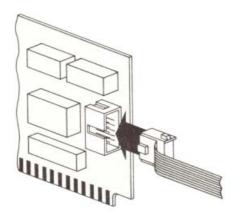


Figure 1-4. Sliding the "Key" into the Groove

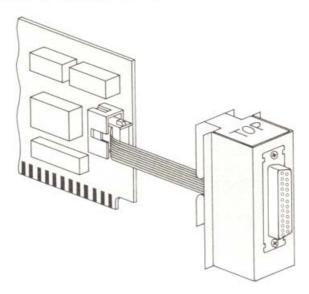


Figure 1-5. Internal Cable Attached Correctly to SSC

# CHAPTER 2 PRINTER MODE

This chapter explains how to prepare, install and use the SSC in Printer Mode, and change the SSC's activities via commands.

# PREPARING THE SSC FOR PRINTER MODE

The SSC is ready to operate in Printer Mode when the jumper block and switches SW1-5 and SW1-6 are correctly positioned (Figure 2-1).

If the triangle on the jumper block is pointing down toward the word MODEM, remove the block (using an IC Extractor, if necessary) and carefully reinsert it so the triangle is pointing toward TERMINAL.

Using a pointed object, set switch SW1-5 OFF and switch SW1-6 ON as shown in Figure 2-1.

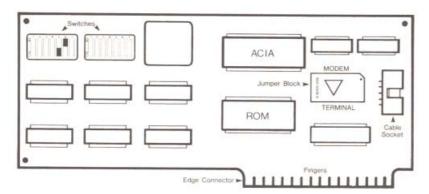


Figure 2-1. SSC Set for Printer Mode



nie de

CONTRACTOR OF THE PERSON NAMED IN

-

When the jumper block is pointing toward TERMINAL, it is acting as a Modem Eliminator. Therefore, DO NOT connect a separate Modem Eliminator, or it will cancel the effect of the jumper block, and the attached device will not work.

## SETTING THE SWITCHES

Use a pointed object, such as the tip of a ballpoint pen, to flip the appropriate tiny switches on the SSC. A switch is ON when the top of the switch rocker is pushed in, and OFF when the bottom is in. The following subsections explain what settings to use.

MI.

M.

OK.

06...

OR.

865...

00

#### **COMMONLY USED SETTINGS**

Table 2-1 lists the switch settings you can use for direct connection, via the SSC, of some commonly used printers. Most printers can use any one of several setups.

Printer	Switch Settings, Cable Connections, Other Information
	$\frac{\text{SW1}: \text{ OFF OFF OFF ON OFF ON ON}}{Printer Mode, HW Hndshk, 9600 baud, 1 stop bit, ** width IDS SW1: ON ON OFF OFF SW2: OFF SSC/IDS pins: 3/3, 7/7, 20/20; all IDS jumpers removed$
NEC 551Ø Spinwriter	$\frac{\mathrm{SWl}}{\mathrm{P8A}}$ OFF ON ON ON OFF OFF OFF $\frac{\mathrm{SW2}}{\mathrm{1}}$ : ON ON * * OFF OFF ON P8A Mode, ETX/ACK, 1200 baud, $\frac{1}{\mathrm{1}}$ stop bit, ** line width NEC switches: OFF ON OFF OFF OFF ON ON SSC/NEC pins: 2/2, 3/3, 7/7, 20/6&8; 4&5 tied on NEC end May need keystroke to force first ETX after power-up.
NEC 551Ø Spinwriter	SW1: OFF ON ON ON OFF ON OFF SW2: ON ON * * OFF OFF ON Printer Mode, hardware handshake, rest same as above NEC switches: OFF ON OFF OFF OFF OFF ON ON SSC/NEC pins: 3/3, 6/6&8, 7/7, 20/20; 4&5 NOT tied
Qume Sprint 5	SW1: OFF ON ON ON OFF ON ON SW2: ON OFF * * OFF OFF OFF Printer Mode, HW Hndshk, $1200$ baud, 1 stop bit, ** width Qume switches: $1200$ baud, no modem; pins: 3, 4, 7, $20$ Qume asserts RTS and DTR only when ready to receive data
Qume Sprint 9/35	$\frac{\text{SW1}}{\text{Printer Mode, HW Hndshk, 9600}}$ sw2: ON OFF * * OFF OFF OFF Qume ETX-ACK/XON-XOFF switch set to ETX-ACK for HW Hndshk

Table 2-1. Commonly Used Switch Settings for Printer Mode

#### **BAUD RATE**

No matter what type of printer or terminal you connect to the SSC, the SSC is going to pass information between the Apple II and the device at a certain prearranged speed, called the baud rate. Since the Apple II can usually send and receive information faster than what is connected to it, the simplest way to determine the baud rate is to consult the user manual for the device you will connect. Find out what rate is the fastest the device can handle (up to 19,200 baud). Once you know this, you are ready to set the baud rate switches on the SSC.

Baud	SW1-1	SW1-2	SW1-3	SW1-4	Baud	SW1-1	SW1-2	SW1-3	SW1-4
5.4	ON	ON	ON	OFF	1200	OFF	ON	ON	ON
5Ø	ON	ON	OFF	ON	1800	OFF	ON	ON	OFF
75	ON	ON	OFF	OFF	2400	OFF	ON	OFF	ON
110*	ON	OFF	ON	ON	3600	OFF	ON	OFF	OFF
	ON	OFF	ON	OFF	4800	OFF	OFF	ON	ON
150	ON	OFF	OFF	ON	7200	OFF	OFF	ON	OFF
300	ON	OFF	OFF	OFF	9600	OFF	OFF	OFF	ON
600	S 1911	(**		100000000000000000000000000000000000000	19200	OFF	OFF	OFF	OFF

Table 2-2. Baud Rate Switch Settings



3

1

Make sure the printer or terminal you connect is set (with its own switches, dials or thumb wheels) to the SAME baud rate! If you don't, the SSC will send and receive unrecognizable garbage.

# DATA FORMAT AND PARITY

The SSC sends each character (such as a "3" or an "F" or a Carriage Return) as a string of zeroes and ones ( $\underline{\text{bits}}$ ). The way it can send a character in Printer Mode, using switch settings, is this:

- first a single start bit to signal to the printer or terminal that a character is coming;
- then a string of 8 data bits representing the character;
- · no error-checking parity bit;
- one or two stop bits to signal the end of a character.

For Printer Mode, the only aspect of the data format you can change with switch settings is whether to send one stop bit or two. If you set the baud rate switches to  $5\emptyset$ , 75 or  $11\emptyset$  baud, set switch SW2-1 OFF (two stop bits). For all other baud rates, set switch SW2-1 ON (one stop bit) unless the documentation for the device you are connecting specifies otherwise.

The SSC does not send or check parity bits in Printer Mode unless you select some parity using the  $\langle n \rangle P$  command, explained later in this chapter.

### CARRIAGE RETURN DELAY

If you connect a slow printer to the SSC, and it has no handshaking capability, you may need to set switch SW2-2 ON to cause the Apple II to wait 1/4 second after a Carriage Return ( $\langle CR \rangle$ ). This gives

the print head assembly time to reposition to the beginning of the next line. Otherwise, set switch SW2-2 OFF (no delay).

Additional delay values (32 ms and 2 s) are available via the <n>C command described later in this chapter.

-

-

NI.

.

RS.

84.

ES.

AIL.

MIL.

MIL.

RIE.

RIE.

Mil.

#### LINE WIDTH AND VIDEO ON/OFF

Switches SW2-3 and SW2-4 determine the printer or terminal line width and also turn the Apple II video screen on or off.

If you are connecting a printer to the SSC, select the appropriate switch settings for the number of characters the printer can fit on a line. If you set the line width to  $4\emptyset$ , the Apple II video screen is turned on, since it too can display  $4\emptyset$  characters per line, and so can display an exact replica of what is being printed.

If you plan to connect a terminal to the SSC, set the switches for the number of characters the terminal screen can display on a line--usually 72 or 80. For these line widths, the Apple II video screen is off.

Line	Width	Video	Screen	SW2-3	SW2-4
40	char/line	01	n	ON	ON
72	char/line	of	ff	ON	OFF
80	char/line	of	Ef	OFF	ON
132	char/line	of	EE	OFF	OFF

Table 2-3. Line Width and Video Switch Settings

The switch settings that turn off the Apple II video screen take effect only after PR# under BASIC or DOS. <CTRL-I> commands are still recognized, and cause the message APPLE SSC: to appear on the Apple II video screen.

#### GENERATE (LF) OUT

If you are connecting a printer to the SSC, check the printer's user manual to see if it automatically generates a linefeed ( $\langle LF \rangle$ ) after a carriage return ( $\langle CR \rangle$ ). If it does not, set switch SW2-5 ON.

If your printer does automatically generate a linefeed after a carriage return, or if you are connecting some other device that does not need automatic linefeed generation, set switch SW2-5 OFF.

#### SPECIAL SWITCHES

Switch SW2-6 controls forwarding of interrupts to the Apple II. Since the Apple II and II+ do not handle interrupts, set SW2-6 OFF. Normally, switch SW1-7 is ON and switch SW2-7 is OFF. In the rare cases where the device uses pin 19, Secondary Clear To Send, in place of pin 4 or 20, Clear To Send, set SW1-7 OFF and SW2-7 ON.

Your Super Serial Card is now ready to install and use in Printer

## INSTALLATION PROCEDURE

This section explains how to install the SSC and its internal cable in the Apple II. If the cable clamp is not already assembled, do so now, following the instructions given in Chapter 1.



nine d

-

NAME OF

-

Before connecting or disconnecting anything on the Apple, turn off the power with the switch at the back left corner of the Apple case. THIS IS ABSOLUTELY NECESSARY. If you try to connect or disconnect anything from the inside of your Apple when the power is on, you are likely to damage the circuits.

Do not unplug the Apple, just turn it off. If you unplug the Apple, you will isolate it from earth ground and leave it vulnerable to static discharges.

Remove the Apple cover by pulling up on the two back corners of the cover until the two corner fasteners pop apart. Slide the cover back until it is free of the case and then lift the cover off.

Look inside the Apple and locate the power supply case--the rectangular metal box along the left inside the Apple II. To avoid damaging the SSC, touch the power supply case with one hand; this discharges any static charge that may be on your clothes or body.

Along the back inside edge of the Apple you will see eight long narrow slots called connector slots. The connector slots are numbered from  $\emptyset$  at the left to 7 at the right. The numbers are printed along the back edge behind the connector slots. For use with Pascal, install the SSC in slot #1 for a printer, or slot #3 for a terminal. For use with BASIC, install the SSC in any slot from #1 through #7.



Handle the Super Serial Card as you would handle an expensive phonograph record. Grasp it only by the corners or edges, and do not touch the components or pins, especially the gold fingers on the edge connector.

There are three deep notches along the back of the Apple II case. Temporarily set the SSC down near the desired slot. Then take the clamp assembly and slide it down into the notch closest to the slot that the SSC will be in. Tighten the screws until the connector assembly can no longer be moved in the opening.

Grasp the upper corners of the SSC and insert the gold fingers of the edge connector into the slot in the back of the Apple, rear edge first. Gently push the front edge of the card down until it is level and firmly seated.

Note that the outer ends of the screws in the clamp assembly can act as nuts. They are threaded and can receive screws from the printer or terminal connector, to ensure a good connection with the Apple.

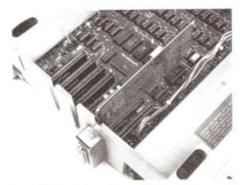


Figure 2-2. SSC in Slot #1 and Clamp Assembly in Notch

EL.

44

4

Slide the Apple case top plate in place and press down on the rear corners until the corner fasteners pop into place. The Super Serial Card is now installed.

# EXTERNAL CABLE AND CONNECTOR

The SSC cable connector you installed in the notch is a standard DB-25 connector with 25 pins. Ten pins of the connector are connected internally to the SSC. Connector pin assignments are listed in Appendix C.

You will need a cable to connect your external device to the SSC connector on the Apple II. Shielded cables with 25-pin connectors on one end are available from your Apple dealer.

The cable must have internal shielding, with the shielding properly terminated at both ends, to prevent electromagnetic interference to nearby radios, television sets, and communication equipment. This shielding is necessary for the system to comply with Class B Federal Communications Commission limits as defined by Subpart J of Part 15 of the FCC rules. Unshielded cables are not recommended.



Make sure that all devices are connected to the same grounded AC power circuit (three-wire wall outlet) as the Apple II. Connecting ungrounded equipment to your Apple II can cause severe electrical damage.

# USING THE SSC IN PRINTER MODE

Printer Mode allows you to use the SSC with a local (that is, directly connected) printer or terminal, as well as other local serial devices. After installing the SSC, you can control its operation from a BASIC, Pascal or assembly-language program, or even directly from the keyboard. The two parts of this section explain the easiest way to get the SSC up and running from the keyboard with a printer or terminal.

#### WITH A PRINTER

750

-78

- PR

- 19

PS

100

To use the SSC with a printer, do the following:

- · Make sure the jumper block points toward TERMINAL.
- . Under BASIC or DOS, boot the Apple II and then type in PR#s to send output to the printer (with the SSC in slot s).
- Under Pascal, boot the Apple II and then use the F(iler T(ransfer command to send output data to #6: or PRINTER: (with the SSC in slot #1).
- If the printer doesn't work, refer to Appendix E for troubleshooting hints, or consult your Apple dealer.

#### WITH A TERMINAL

To use the SSC with a terminal, do the following:

- · Make sure the jumper block points toward TERMINAL.
- . Under BASIC or DOS, boot the Apple II and then type in PR#s and IN#s to route both input and output through the terminal (with the SSC in slot #s).
- Under Pascal, boot the Apple II and then use the terminal as the input/output console (with the SSC in slot #3).
- If the terminal doesn't work, refer to Appendix E for troubleshooting hints, or consult your Apple dealer.

# PRINTER MODE COMMANDS

You can issue any of the commands described in this section by embedding them in a computer program. Under BASIC, DOS or the Apple Monitor, you can also enter them directly at the Apple (or terminal) keyboard.

In a BASIC program, put the control character and command in a PRINT statement. In a Pascal program, issue the command in a WRITE or WRITELN statement.

N.

N.

-

RIE.

88.

N.

MR.

NE.

M.

44

NR.

88...

ma.

44

100

S.

EE.

O IL

10E

When you enter the command character (usually <CTRL-I); see below), the prompting message APPLE SSC: appears on the display screen. Subsequent characters, up to <RETURN>, will be interpreted as an SSC command. Pressing the left arrow key before pressing <RETURN> cancels the command and causes the APPLE SSC: prompt to reappear.

Many of these commands override the physical switch settings on the SSC. This makes it unnecessary to open the Apple II case and manually flip the SSC switches. To change the values back to the physical switch settings, reboot or reset the Apple II, or type in the Reset command described below.

#### COMMAND FORMATS

All commands are preceded by the Printer Mode command character (usually <CTRL-I>, see below) and followed by <RETURN>. The notation <CTRL-I> means "hold down the CTRL key while pressing I." There are three types of command formats:

- a number <n> followed by an uppercase letter (for example, 4D to set Data Format 4)
- · simply an uppercase letter (for example, R to Reset the SSC)
- an uppercase letter followed by a space and then either E to Enable or D to Disable a feature (for example, L D to Disable automatic generation of linefeed characters)

The allowable range of  $\langle n \rangle$  is given in each command description (next section). The choice of Enable or Disable is indicated as  $\langle E/D \rangle$ .



The underscore character (\_) before the <E/D> in Enable/Disable commands is merely a reminder that a space is required there.

The SSC checks only numbers and the first letters of commands and options. All such letters must be uppercase. Further letters, which you can add to assist your memory, have no effect on the SSC For example, X(OFF E(nable is the same as X E. The SSC ignores invalid commands.

#### THE COMMAND CHARACTER

The normal command character in Printer Mode is <CTRL-I> (decimal 9; Appendix D). You can send the command character itself through the SSC by typing it twice in a row: <CTRL-I><CTRL-I>; no <RETURN> is required after this command. This special command allows you to transmit the command character without affecting the operation of the SSC, and without having to change to another command character and then back again later.

If you want to change the command character from  $\langle CTRL-I \rangle$  to  $\langle CTRL-something else \rangle$ , type  $\langle CTRL-I \rangle \langle CTRL-something else \rangle$ . For example, to change the command character to  $\langle CTRL-W \rangle$ , type  $\langle CTRL-I \rangle \langle CTRL-W \rangle$ . To change back, type  $\langle CTRL-W \rangle \langle CTRL-I \rangle$ . No  $\langle RETURN \rangle$  is required after either of these commands.

The command character  $\langle \text{CTRL-I} \rangle$  is ASCII code 9. Here is how to generate this character in BASIC and Pascal:

Integer BASIC: PRINT "\*command" \*embedded <CTRL-I>

Applesoft BASIC: PRINT CHR\$(9): "command"
Pascal: WRITELN (CHR(9), 'command');

# PRINTER MODE COMMAND SUMMARY

Table 2-4 is a summary of the commands available in Printer Mode. Some details, explained fully in the remainder of this chapter, have been omitted from the table for the sake of brevity. Commands marked with an asterisk are not supported by Pascal.

Format	Command Name	Values	Interpretation
<n>B</n>	Baud Rate	Ø - 15	see Table 2-5
<n>C</n>	<cr> Delay</cr>	Ø 1 2 3	no delay 32 milliseconds 25Ø milliseconds (1/4 s) 2 seconds
<n>D</n>	Data Format	Ø 1 2 3 4 5 6 7	8 data bits, 1 stop bit 7 data bits, 1 stop bit 6 data bits, 1 stop bit 5 data bits, 1 stop bit 8 data bits, 2 stop bits 7 data bits, 2 stop bits 6 data bits, 2 stop bits 5 data bits, 2 stop bits 5 data bits, 2 stop bits
<n>F</n>	<ff> Delay</ff>	Ø 1 2 3	no delay (default) 32 milliseconds 250 milliseconds (1/4 s) 2 seconds
<n>L</n>	<lf> Delay</lf>	Ø 1 2 3	no delay (default) 32 milliseconds 250 milliseconds (1/4 s) 2 seconds
<n>P</n>	Parity	Ø,2,4,6 1 3 5 7	no parity (default = ØP) odd parity even parity MARK (parity bit always 1) SPACE (parity bit always Ø)
* <n>T</n>	Translate Lowercase (LC)	Ø 1 2 3	change LC to UC (default) leave LC (possible garbage) LC to UC inverse; leave UC LC to UC; UC to inverse
* C * R Z	Column Overflow Reset the SSC Zap <ctrl></ctrl>		auto- <cr> at column's end reset SSC + PR#Ø and IN#Ø ignore all <ctrl> commands</ctrl></cr>
F_ <e *="" d)="" l_<e="" m_<e="" not="" su<="" t_<e="" td="" x_<e=""><td>Generate <lf> Out Mask <lf> In Tab in BASIC</lf></lf></td><td>E or D E or D E or D E or D E or D</td><td>accept keyboard entries send <lf> out after <cr> drop <lf> in after <cr> recognize BASIC tabs detect XOFF; await XON</cr></lf></cr></lf></td></e>	Generate <lf> Out Mask <lf> In Tab in BASIC</lf></lf>	E or D E or D E or D E or D E or D	accept keyboard entries send <lf> out after <cr> drop <lf> in after <cr> recognize BASIC tabs detect XOFF; await XON</cr></lf></cr></lf>

III.

118.

Mi.

BE.

Table 2-4. Printer Mode Commands

#### COMMANDS THAT CHANGE SWITCH SETTINGS

The group of commands discussed in this section either directly override the SSC switch settings, or affect related behavior of the SSC. The Reset command restores the switch selections.

#### Baud Rate-(n)B

700

1

W)

ASS

100

750

This command overrides the physical settings of switches SW1-1 through SW1-4 on the SSC. For example, to change the baud rate to 135 baud, type in <CTRL-I>4B<RETURN>.

<n>=</n>	SSC Baud Rate	<n>=</n>	SSC Baud Rate
Ø	use SW1-1 to SW1-4	8	1200
1	50	9	1800
2	75	10	2400
3	109.92 (110)	11	3600
4	134.58 (135)	12	4800
5	150	13	7200
6	300	14	9600
7	600	15	19200

Table 2-5. Baud Rate Selections

### Data Format-(n)D

With this command you can override the settings of switch SW2-1. The table below shows how many data and stop bits correspond to each value of  $\langle n \rangle$ . For example,  $\langle \text{CTRL-I} \rangle \text{ETURN} \rangle$  causes the SSC to transmit each character in the form: one start bit (always transmitted), six data bits, and one stop bit.

<n>=</n>	Data Bits	Stop Bits
Ø	8	1
1	7	1
2	6	1
3	5	1
4	8	2 (1 with Parity options 4 through 7)
5	7	2
6	6	2
7	5	2 (1-1/2 with Parity options ∅ through 3)

Table 2-6. Data Format Selections

#### Parity-(n)P

You can use this command to determine the kind of parity the SSC is to generate when sending data and check for when receiving data. In general, parity checking is not needed in Printer Mode. However, there are five parity options available (Table 2-4).

<n>=</n>	Parity to Use
Ø, 2, 4 or 6 1 3 5	none (default value) odd parity (odd total number of ones) even parity (even total number of ones)
5	MARK parity (parity bit always 1)
7	SPACE parity (parity bit always Ø)

OR.

OR.

Sec.

OF.

OB.

Kha

OK.

W.

of L

oli.

MS.

100

Table 2-7. Parity Selections

For example, type <CTRL-I>IP<RETURN> to cause the SSC to transmit and check for odd parity. Odd parity means that the high bit of every character is Ø if there is already an odd number of 1 bits in that character, or 1 if there is otherwise an even number of 1 bits in the character, making the total always odd. This is an easy (but not foolproof) way to check data for transmission errors. Parity errors are recorded in a status byte (Appendix F).

# Set Time Delay- $\langle n \rangle C$ , $\langle n \rangle L$ , $\langle n \rangle F$

Some printers are slow and do not provide a "printer busy" or handshake signal to the Apple II. The <n>C command causes the Apple II to delay a specified amount of time, after sending a carriage return character, before sending another group (usually another line) to it. This gives the print head enough time to return to the left side of the page so it is ready to continue printing.

The  $\langle n \rangle C$  command overrides the setting of switch SW2-2 on the SSC. That switch provides only two choices: no delay or a 250 millisecond delay.

The  $\langle n \rangle$ L command allows time after a linefeed character for a printer platen to turn so the paper is vertically positioned to receive the next line.

The  $\langle n \rangle$ F command allows time after a form feed character for the printer platen to move the paper form to the top of the next page (typically a longer time than a linefeed).

<n>=</n>	Time Delay
Ø	none
1	32 milliseconds
2	250 milliseconds (1/4 second)
3	2 seconds

Table 2-8. Time Delay Selections

Consult the user manual for the printer to find out how much time it takes to move its print head and platen, and so to determine an appropriate set of values for these three delays. The idea is to have at least enough time for the printer parts to move the required distance, but not so much time that overall printing speed is slowed down drastically. A typical set for a VERY slow printer would be <CTRL-I>2C<RETURN>, <CTRL-I>2L<RETURN>, <CTRL-I>3F<RETURN>; that is, the SSC waits 250 milliseconds after transmitting carriage returns, 250 milliseconds after transmitting linefeeds, and 2 seconds after transmitting form feed characters.

### Generate (CR) On Column Overflow-C

Typing <CTRL-I>C<RETURN> causes the SSC to generate a carriage return character automatically any time the column count exceeds the printer line width.



-

CONT.

Once this is on, only clearing the high-order bit at location \$578+s (where s is the slot the SSC is in) can turn this option back off. This option is normally off.

#### Generate (LF) Out-L\_(E/D)

You can use this command to have the SSC automatically generate and transmit a linefeed character after each carriage return character. This overides the setting of switch SW2-5. For example, you can type <CTRL-I>L E<RETURN> to cause your printer to print listings or double-spaced manuscripts for editing.

#### Mask (Suppress) $\langle LF \rangle \ln M_{\langle E/D \rangle}$

If you type <CTRL-I>M E<RETURN>, the SSC will suppress any incoming linefeed character that immediately follows a carriage return character.

#### Reset the SSC-R

Typing  $\langle \text{CTRL-I} \rangle \text{R} \langle \text{RETURN} \rangle$  has the same effect as sending a PR#Ø and an IN#Ø to a BASIC program and then resetting the SSC. This keyboard command cancels all previous commands to the SSC and puts the physical switch settings back into force.

#### OTHER COMMANDS

The commands described here affect the handling of characters and tabs. The Translate command determines how characters will appear on the video screen. The Z and F commands prevent the SSC from responding to control characters or ALL characters coming from the keyboard, respectively. The X command causes the SSC to respond to the XON/XOFF software protocol. Finally, the T command implements the tabbing feature of BASIC.

#### Translate Lowercase Characters-(n)T

The Apple II Monitor "translates" all incoming lowercase characters into uppercase ones before sending them to the video screen or to a BASIC program. The SSC offers four translation options:

#### <n>= What to Do with Lowercase Characters

- Ø Change all Lowercase characters to uppercase ones before passing them to a BASIC program or to the video screen. This is the way the Apple II monitor handles lowercase.
- Pass along all lowercase characters unchanged. The appearance of the lowercase characters on the Apple II screen is undefined (garbage).
- Display lowercase characters as uppercase inverse characters (that is, as black characters on a white background).

100

Pass lowercase characters to programs unchanged, but display lowercase as uppercase, and uppercase as inverse uppercase (that is, as black characters on a white background).

Table 2-9. Lowercase Character Displays

#### Zap (Suppress) Control Characters-Z

Typing <CTRL-I>Z<RETURN> prevents the SSC from recognizing any further control characters (and hence commands) whether coming from the keyboard or contained in a stream of characters moving through the SSC.

If you issue the Z command described here, all further commands are ignored; this is useful if the data you are transmitting contains bit patterns that the SSC can mistake for control characters.



The only way to reinstate command recognition after the Z command is to reinitialize the SSC, or clear the high-order bit at location \$5F8+s (where s is the slot in which the SSC is installed).

#### Find Keyboard $-F_{\langle E/D \rangle}$

You can protect incoming data from disruption by keystrokes with this command. For example, you can include an F D command in a program, followed by a routine that retrieves data coming in through the SSC, followed by F E later in the program. Default is F E.

# XOFF Recognition $-X_{E/D}$

Typing <CTRL-I>X E<RETURN> causes the SSC to look for any XOFF (decimal 19; Appendix D) character coming from a device attached to the SSC, and to respond to it by halting transmission of characters

until the SSC receives an XON (decimal 17; Appendix D) from the device, signalling the SSC to continue transmission. In Printer Mode, the default value of this command is X D.



in EX

11100

In Printer Mode, full duplex communication may not work with XOFF recognition turned on, so be careful.

#### Tab in BASIC-T\_ $\langle E/D \rangle$

If you type in <CTRL-I>T E<RETURN>, the BASIC horizontal position counter is left equal to the column count. All TABs work, including back-tabs. TABs beyond column 40 require a POKE to location 36, as usual. Commas only work as far as column 40, and BASIC programs will be listed in 40-column format.

H.

E ...

1

N.

16

100

18

100

100

.

100

100

RG.

100

NO.

100

NIK.

ESK.

Mil.

Bio

Part I

MIL.

# **CHAPTER 3 COMMUNICATIONS MODE**

-

100

the the the the

10

100

100

in

11 11

- 10

100

100

-

This chapter explains how to prepare, install and use the SSC in Communications Mode, and change the SSC's activities via commands.

# PREPARING THE SSC FOR **COMMUNICATIONS MODE**

The SSC is ready to operate in Communications Mode when the jumper block and switches SW1-5 and SW1-6 are correctly positioned.

If the triangle on the jumper block is pointing up toward the word MODEM, remove the block (using an IC Extractor, if necessary) and reinsert it with the triangle pointing toward MODEM (Figure 3-1).

Using a pointed object, set switches SW1-5 and SW1-6 both ON as shown in Figure 3-1. This puts the SSC in Communications Mode.

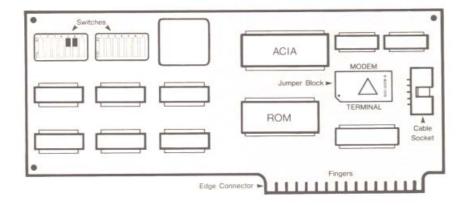


Figure 3-1. SSC Set for Communications Mode

# SETTING THE SWITCHES

Use the tip of a ballpoint pen or some other sharp object to flip the appropriate tiny switches on the SSC. A switch is ON when the top of the switch rocker is pushed in. The following subsections explain what settings to use. 100

#### COMMONLY USED SETTINGS

Table 3-1 lists the switch settings you can use for connection to various devices and services via the SSC and a modem.

1 - li-tier Cuitel Cettings Cable Connections Other Information

Application	Switch Settings, Cable Connections, Other Information
Apple II via modem	$\frac{\mathrm{SW1}:}{\mathrm{Comm}}$ ON OFF OFF ON ON ON ON $\frac{\mathrm{SW2}:}{\mathrm{I}}$ ON ON * * OFF OFF OFF OFF OFF OFF Using SSC in each Apple, set both the same; for local connection, second jumper block points toward TERMINAL.
Apple III via modem	SW1: ON OFF OFF ON ON ON ON SW2: ON ON * * OFF OFF OFF OFF Comm Mode, 300 baud, 8 data, 1 stop, * * parity Set Apple III RS-232-C Device Control Block to same values (See Apple III Standard Device Drivers manual).
Printer via modem	$\frac{\mathrm{SW1:}}{\mathrm{Comm}}$ ON OFF OFF ON ON ON ON $\frac{\mathrm{SW2:}}{\mathrm{ON}}$ OFF * * OFF OFF OFF OFF OFF Mode, 300 baud, 7 data, $\frac{\mathrm{SW2:}}{\mathrm{ON}}$ stop, * * parity Baud rate is limited by modem and transmission lines; some modems can also use 1200 baud; $\mathrm{SW1-7}$ is always ON, and $\mathrm{SW2-7}$ is always OFF; SCTS hookup is at remote modem.
Dow Jones News and Quotes Reporter	SW1: ON OFF OFF ON ON ON ON <u>SW2</u> : ON OFF - ON OFF OFF OFF Comm Mode, 300 baud, 7 data, 1 stop, no parity Sample program at end of this chapter sets same traits. Use T command for Terminal Mode operation.

Table 3-1. Commonly Used Switch Settings for Communications Mode

Make sure that the settings on the SSC, modem and remote device are all compatible. Successful operation using a modem depends on this.

After setting the switches on the SSC, you can go on to the next major section of this chapter, Installation Procedure.

#### **BAUD RATE**

No matter what kind of modem and remote device you connect to the SSC, the SSC is going to pass information between the Apple II and the device at a certain prearranged speed, called the <u>baud rate</u>. Since the Apple II can usually send and receive information faster than what is connected to it, the simplest way to determine the maximum baud rate you can use is to consult the user manual for the modem and remote device you will connect. Find out what rate is the fastest they both can handle. Once you know this, you are ready to

set the baud rate switches on the SSC. The following table shows the correct switch positions.

Baud	SW1-1	SW1-2	SW1-3	SW1-4	Baud	SW1-1	SW1-2	SW1-3	SW1-4
5Ø	ON	ON	ON	OFF	1200	OFF	ON	ON	ON
75	ON	ON	OFF	ON	1800	OFF	ON	ON	OFF
110*	ON	ON	OFF	OFF	2400	OFF	ON	OFF	ON
135**	ON	OFF	ON	ON	3600	OFF	ON	OFF	OFF
150	ON	OFF	ON	OFF	4800	OFF	OFF	ON	ON
300	ON	OFF	OFF	ON	7200	OFF	OFF	ON	OFF
600	ON	OFF	OFF	OFF	9600	OFF	OFF	OFF	ON
(* 109	.92)	(**	134.5	8)	19200	OFF	OFF	OFF	OFF

Table 3-2. Baud Rate Switch Settings



-

100

| The | The

- 100

COLUMN TO SERVICE STREET

1000

If you are connecting a printer or terminal at the other end of the modem, make sure that it is set (with its own switches, dials or thumb wheels) to the SAME baud rate! If you don't, the SSC will send and receive unrecognizable garbage.

#### **DATA FORMAT AND PARITY**

The SSC sends each character (such as a "7" or an "H" or a "?") as a string of zeroes and ones (bits). The way it can send a character in Communications Mode, using switch settings, is this:

- first a single start bit to signal to the printer or terminal that a character is coming;
- then a string of 7 or 8 data bits representing the character;
- · possibly a parity bit for error checking;
- lastly one or two stop bits that signal the end of a character.

For Communications Mode, you can use switch settings to change three aspects of the data format: the number of data bits, the number of stop bits, and the kind (if any) of parity bit to send. Switches SW2-1 through SW2-4 determine the data format as shown in this table.

Stop		Data		Parity		
Bits	SW2-1	Bits	SW2-2	Bits	SW2-3	SW2-4
1	ON	8	ON	none		ON
2	OFF	7	OFF	odd	ON	OFF
				even	OFF	OFF

Table 3-3. Data Format Selections

If SW2-1 is OFF, the number of stop bits will be 1 instead of 2 if both 8 data bits (SW2-2 ON) and a parity bit (SW2-4 OFF) have been selected.

To determine the correct combination of switch settings, consult the literature describing the device or timesharing service you plan to connect to the SSC in this mode.

Keri

R.

18.

ik.

1

M.

HE.

in.

Mil.

Maria.

The most commonly used format for ASCII data is: 7 data bits, 1 stop bit, and no parity bit (SW2-1 and SW2-4 ON; SW2-2 OFF).

If you set the data rate switches to 50, 75 or 110 baud, choose a switch combination that specifies 2 stop bits; for all data rates 135 baud or higher, use 1 stop bit (switch SW2-1 ON), unless device or timesharing service literature specifies otherwise.



To set the SSC for a data format different from those shown in this table, or to change the data format temporarily, use the SSC commands described later in this chapter.

#### GENERATE (LF) OUT

If the remote device (for example, a faraway printer) does not automatically generate linefeeds after carriage returns, and it desperately needs them, then set switch SW2-5 ON. Otherwise set SW2-5 OFF.

In Communications Mode, the SSC automatically discards incoming linefeeds that immediately follow carriage returns, unless you use the M D command as described later in this chapter.

#### SPECIAL SWITCHES

Switch SW2-6 controls forwarding of interrupts to the Apple II. Since the Apple II and II+ do not handle interrupts, set SW2-6 OFF.

For Communications Mode, set SW1-7 ON and SW2-7 OFF.

Your Super Serial Card is now ready to install and use in Communications Mode.

## INSTALLATION PROCEDURE

This section explains how to install the SSC and its internal cable in the Apple II. If the cable clamp is not already assembled, do so now, following the instructions given in Chapter 1.



-

-

100

-

Before connecting or disconnecting anything on the Apple, turn off the power with the switch at the back left corner of the Apple case. THIS IS ABSOLUTELY NECESSARY. If you try to connect or disconnect anything from the inside of your Apple when the power is on, you are likely to damage the circuits.

Do not unplug the Apple, just turn it off. If you unplug the Apple, you will isolate it from earth ground and leave it vulnerable to static discharges.

Remove the Apple cover by pulling up on the two back corners of the cover until the two corner fasteners pop apart. Slide the cover back until it is free of the case and then lift the cover off.

Look inside the Apple and locate the power supply case--the rectangular metal box along the left inside the Apple II. To avoid damaging the SSC, touch the power supply case with one hand; this discharges any static charge that may be on your clothes or body.

Along the back inside edge of the Apple you will see eight long narrow slots called connector slots. The connector slots are numbered from Ø at the left to 7 at the right. The numbers are printed along the back edge behind the connector slots. For use with Pascal and a modem, install the SSC in slot #2. For use with BASIC, install the SSC in any slot from #1 through #7.



Handle the Super Serial Card as you would handle an expensive phonograph record. Grasp it only by the corners or edges, and do not touch the components or pins, especially the gold fingers on the edge connector.

There are three deep notches along the back of the Apple II case. Temporarily set the SSC down near the desired slot. Then take the clamp assembly and slide it down into the notch closest to the slot that the SSC will be in. Tighten the screws until the connector assembly can no longer be moved in the opening.

Grasp the upper corners of the SSC and insert the gold fingers of the edge connector into the slot in the back of the Apple, rear edge first. Gently push the front edge of the card down until it is level and firmly seated. Figure 3-2 shows how the SSC looks when installed in slot #2.

Note that the outer ends of the screws in the clamp assembly can act as nuts. They are threaded and can receive screws from the printer or terminal connector, to ensure a good connection with the Apple.

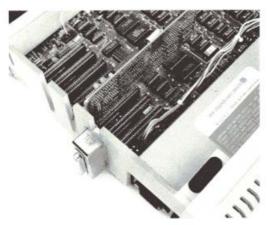


Figure 3-2. SSC in Slot #2 and Clamp Assembly in Notch

Slide the Apple case top plate in place and press down on the rear corners until the corner fasteners pop into place. The Super Serial Card is now installed.

# EXTERNAL CABLE AND CONNECTOR

The SSC cable connector you installed in the notch is a standard DB-25 connector with 25 pins. Ten pins of the connector are connected internally to the SSC.

You will need a cable to connect the modem or other device to the SSC connector on the Apple II. Cables with 25-pin connectors on one end are available from your Apple dealer.

The cable must have internal shielding, with the shielding properly terminated at both ends, to prevent electromagnetic interference to nearby radios, television sets, and communication equipment. This shielding is necessary for the system to comply with Class B Federal Communications Commission limits as defined by Subpart J of Part 15 of the FCC rules. Unshielded cables are not recommended.



Make sure that all devices are connected to the same grounded AC power circuit (three-wire wall outlet) as the Apple II. Connecting ungrounded equipment to your Apple II can cause severe electrical damage.

# USING SSC IN COMMUNICATIONS MODE

Communications Mode allows you to use the SSC with a modem, connected to a remote device (such as a remote printer, terminal, or other computer). After installing the SSC, you can control its operation

from a BASIC, Pascal or assembly-language program, or even directly from the keyboard. To use the SSC in Communications Mode, do the following:

· Make sure the jumper block points toward MODEM.

- 10

- 11

-

- 100

1000

1000

- . Under BASIC or DOS, boot the Apple II, and then type in PR#s and IN#s to route input and output, respectively, to and from the remote device. (The SSC is in slot s.)
- Under Pascal, boot the Apple II and then use #7: or REMIN: for input, and #8: or REMOUT: for output. (The SSC is in slot #2.)
- If the modem and remote device don't work, refer to Appendix E for troubleshooting hints, or consult your Apple dealer.

# COMMUNICATIONS MODE COMMANDS

You can issue any of the commands described in this section by embedding them in a computer program. Under BASIC or DOS, you can also enter them directly at the Apple (or remote terminal) keyboard.

In a BASIC program, put the control character and command in a PRINT statement. In a Pascal program, embed the command in a WRITE or WRITELN statement.

Before keyboard entry of these commands has any effect on the SSC, you must first issue an IN#s command (with the SSC in slot s). When you then enter the command character (usually <CTRL-A>, see below), the prompt APPLE SSC: appears on the display screen. Subsequent characters up to <RETURN> will be interpreted as an SSC command. Pressing the left arrow key before pressing <RETURN> cancels the command and causes the APPLE SSC: prompt to reappear.

Many of these commands override the physical switch settings on the SSC. This makes it unnecessary to open the Apple II case and manually change the SSC switch settings. To change the values back to the physical switch settings, reboot or reset the Apple II, or type in the Reset command described below.

#### COMMAND FORMATS

All commands are preceded by the Communications Mode command character (usually <CTRL-A>, see below) and followed by <RETURN>.
The notation <CTRL-A> means "hold down the CTRL key while pressing A." There are three types of command formats:

- a number <n> followed by an uppercase letter (for example, 4D to set Data Format 4)
- simply an uppercase letter (for example, R to Reset the SSC)
- an uppercase letter followed by a space and then either E to Enable or D to Disable a feature (for example, L D to Disable automatic generation of linefeed characters)

.

100

The allowable range of  $\langle n \rangle$  is given in each command description below. The choice of Enable or Disable is written as  $\langle E/D \rangle$ .



The underscore character (\_) before the  $\langle \text{E/D} \rangle$  in Enable/Disable commands is merely a reminder that a space is required there.

The SSC checks only numbers and the first letters of commands and options. All such letters must be uppercase. Further letters, which you can add to assist your memory, have no effect on the SSC. For example, E(cho E(nable is the same as E E. The SSC ignores invalid commands.

#### THE COMMAND CHARACTER

The normal command character in Communications Mode is <CTRL-A>.
You can send the command character itself through the SSC by typing it twice in a row: <CTRL-A><CTRL-A> (no <RETURN> necessary). This special command allows you to transmit the command character without affecting the operation of the SSC, and without having to change to another command character and then back again later.

If you want to change the command character from <CTRL-A> to <CTRL-something else>--for example, <CTRL-W>--type <CTRL-A><CTRL-W>. To change back, type <CTRL-W><CTRL-A>. No <RETURN> is required after either of these commands.



Do not change the control character to <CTRL-S>, <CTRL-T> or <CTRL-R>, since in Communications Mode the SSC interprets these as special control commands from a remote device.

The command character <CTRL-A> is ASCII code 1. Here is how to generate this character in BASIC and Pascal:

Integer BASIC: Applesoft BASIC: Pascal: PRINT "\*command" \*embedded <CTRL-A> PRINT CHR\$(2): "command" WRITELN (CHR(2), 'command');

#### COMMUNICATIONS MODE COMMAND SUMMARY

Table 3-4 is a summary of the commands available in Communications Mode. Some details, explained fully in the remainder of this chapter, have been omitted from the table for the sake of brevity. Commands marked with an asterisk are not supported by Pascal.

	Format	Command Name	Values	Interpretation
	<n>B</n>	Baud Rate	Ø - 15	see Table 3-5
	<n>C</n>	<cr> Delay</cr>	Ø 1 2 3	no delay 32 milliseconds 25Ø milliseconds (1/4 s) 2 seconds
	<n>D</n>	Data Format	Ø 1 2 3 4 5 6	8 data bits, 1 stop bit 7 data bits, 1 stop bit 6 data bits, 1 stop bit 5 data bits, 1 stop bit 8 data bits, 2 stop bits 7 data bits, 2 stop bits 6 data bits, 2 stop bits 5 data bits, 2 stop bits 5 data bits, 2 stop bits
	<n>F</n>	<ff> Delay</ff>	Ø 1 2 3	no delay (default) 32 milliseconds 250 milliseconds (1/4 s) 2 seconds
	<n>L</n>	<pre><lf> Delay</lf></pre>	Ø 1 2 3	no delay (default) 32 milliseconds 25Ø milliseconds (1/4 s) 2 seconds
	<n>P</n>	Parity	Ø,2,4,6 1 3 5 7	no parity (default = ØP) odd parity even parity MARK (parity bit always 1) SPACE (parity bit always Ø)
*	<n>S</n>	Screen Slot	Ø-7	chain SSC output to slot n
*	<n>T</n>	Translate Lowercase (LC)	Ø 1 2 3	change all LC to UC leave LC (possible garbage) LC to UC inverse; leave UC LC to UC; UC to inverse
*	B R T Z	Break Reset the SSC Terminal Mode Zap <ctrl></ctrl>		transmit 233 ms BREAK SW reset + PR#Ø and IN#Ø (see Terminal Mode section) ignore all <ctrl> commands</ctrl>
*	E_ <e d=""> F_<e d=""> L_<e d=""> M_<e d=""> X_<e d=""> Not supp</e></e></e></e></e>	Echo Find Keyboard Generate <lf> Out Mask <lf> In XOFF Recognition orted by Pascal.</lf></lf>	E or D E or D E or D E or D E or D	echo input on the screen accept keyboard entries send <lf> out after <cr> drop <lf> in after <cr> detect XOFF; await XON</cr></lf></cr></lf>

- 15

100

100

Table 3-4. Summary of Communications Mode Commands

#### **COMMANDS THAT CHANGE SWITCH SETTINGS**

The commands discussed in this section either override the SSC switch settings, or affect related behavior of the SSC. The Reset command restores the switch selections.

#### Baud Rate-(n)B

This command overrides the physical settings of switches SWI-1 to SWI-4 on the SSC. For example, to change the rate to 9600 baud, type <CTRL-A>14B<RETURN>.

<n>=</n>	SSC Baud Rate	<n>=</n>	SSC Baud Rate
Ø	use SW1-1 to SW1-4	8	1200
1	5Ø	9	1800
2	75	10	2400
3	109.92 (110)	11	3600
4	134.58 (135)	12	4800
5	150	13	7200
6	300	14	9600
7	600	15	19200

Table 3-5. Baud Rate Selections

## Data Format-(n)D

With this command you can override the settings of switches SW2-1 and SW2-2. The table below shows how many data and stop bits correspond to each value of  $\langle n \rangle$ . For example, typing  $\langle \text{CTRL-A} \rangle \text{3D} \langle \text{RETURN} \rangle$  causes the SSC to transmit each character in the form: one start bit (always transmitted), five data bits, and one stop bit.

<n>=</n>	Data Bits	Stop	Bits
Ø	8	1	
1	7	1	
2	6	1	
3	5	1	
4	8	2	(1 with <n>P options 4 through 7)</n>
5	7	2	
6	6	2	
7	5	2	(1-1/2 with <n>P options Ø through 3)</n>

Table 3-6. Data Format Selections

# Parity-(n)P

You can use this command to determine the kind of parity the SSC is to generate when sending data and check for when receiving data. There are five parity options available:

<n>=</n>	Parity to Use
Ø, 2, 4 or 6 1 3 5 7	none odd parity (odd number of l's) even parity (even number of l's) MARK parity (parity bit always l) SPACE parity (parity bit always Ø)

Table 3-7. Parity Selections

For example, type <CTRL-A>1P<RETURN> to cause the SSC to transmit and check for odd parity. Odd parity means that the high bit of every character is Ø if there is already an odd number of 1 bits in that character, or 1 if there is otherwise an even number of 1 bits, making the total always odd. This is an easy (but not foolproof) way to check data for transmission errors. (See Appendix F.)

# Generate (LF) Out-L\_(E/D)

You can use this command to have the SSC automatically generate and transmit a linefeed ( $\langle LF \rangle$ ) character after each carriage return ( $\langle CR \rangle$ ) character. This overides the setting of switch SW2-5. For example, you can type  $\langle CTRL-A \rangle L$  E $\langle RETURN \rangle$  to cause your printer to produce double-spaced listings or manuscripts for editing.

## Mask (Suppress) $\langle LF \rangle$ In-M\_ $\langle E/D \rangle$

If you type <CTRL-A>M D<RETURN>, the SSC will not remove incoming linefeed (<LF>) characters that immediately follow carriage return (<CR>) characters.

#### Reset the SSC-R

Typing <CTRL-A>R<RETURN> has the same effect as sending a PR $\#\emptyset$  and an IN $\#\emptyset$  to a BASIC program and then resetting the SSC. This keyboard command cancels all previous commands to the SSC and puts the physical switch settings back into force.

#### OTHER COMMANDS

The commands described in this subsection control the handling of characters and of the video screen. Three commands control timed delays following transmission of  $\langle \text{CR} \rangle$ ,  $\langle \text{LF} \rangle$  and  $\langle \text{FF} \rangle$  characters. The Translate command controls the display of lowercase and uppercase characters. The Z and F commands suppress control characters and characters entered at the keyboard, respectively. The X command causes the SSC to check the character stream for XOFF, as part of the XON/XOFF protocol. Finally, the  $\langle n \rangle$ S command routes video output to a selected slot, and the E command suppresses display (echo) of characters on the screen.

# Set Time Delays- $\langle n \rangle C$ , $\langle n \rangle L$ , $\langle n \rangle F$

Some printers are slow and do not provide a "printer busy" or handshake signal to the Apple II. If such a printer is connected to the SSC via a modem, you may want to use these three delay commands.

The  $\langle n \rangle$ C command causes the Apple II to delay a specified amount of time, after sending a carriage return character, before sending another group (usually another line) to it. This gives the print head enough time to return to the left side of the page so it is ready to continue printing.

The  $\langle n \rangle$ L command allows time after a linefeed character for a printer platen to turn so the paper is vertically positioned to receive the next line.

100

The  $\langle n \rangle$ F command allows time after a form feed character for the printer platen to move the paper form to the top of the next page (typically a longer time than a Linefeed).

<n>=</n>	Time Delay
Ø	none
1	32 milliseconds
2	250 milliseconds (1/4 second)
3	2 seconds

Table 3-8. Time Delay Selections

Consult the user manual for the printer to find out how much time it takes to move its print head and platen, and so to determine an appropriate set of values for these three delays if a printer is used as the remote device. The idea is to have at least enough time for the printer parts to move the required distance, but not so much time that overall printing speed is slowed down drastically.

#### Translate Lowercase Characters-(n)T

The Apple II monitor "translates" all incoming lowercase characters into uppercase ones before sending them to the video screen or to a BASIC program. With the  $\langle n \rangle T$  command, four options are available:

- Change all lowercase characters to uppercase before passing them to a BASIC program or to the video screen. This is what the Apple II monitor does to lowercase.
- Pass along all lowercase characters unchanged. The appearance of the lowercase characters on the Apple II screen is undefined (garbage).
- Display lowercase characters as uppercase inverse characters (that is, as black characters on a white background).
- Pass lowercase characters to programs unchanged, but display lowercase as uppercase, and uppercase as inverse uppercase (that is, as black characters on a white background).

Table 3-9. Lowercase Character Displays

# Zap (Suppress) Control Characters-Z

Typing <CTRL-A>Z<RETURN> prevents the SSC from recognizing any further control characters (and hence commands) in the stream of characters moving through the SSC.

If you issue the Z command, all further commands are ignored; this is useful if the data you are transmitting contains bit patterns that the SSC can mistake for control characters.



-

-

- 15

-

The only way to reinstate command recognition after invoking the Z command is to reset the SSC, or clear the high-order bit at location \$5F8+s\$ (with the SSC in slot s).

#### Find Keyboard-F\_(E/D)

You can protect incoming data from disruption by keystrokes with this command. For example, you can include <CTRL-A>F D in a program, followed by a routine that retrieves data coming in through the SSC, followed by <CTRL-A>F E later in the program.

#### XOFF Recognition-X\_(E/D)

In Communications Mode, the SSC automatically recognizes any XOFF (decimal 19; Appendix D) character coming from a device attached to it, and responds to it by halting transmission of characters. The SSC resumes transmission as soon as it receives an XON character (decimal 17; Appendix D) from the device. To disable XOFF recognition, use <CTRL-A>X D<RETURN>.

Specify Screen Slot-(n)S

With this command you can specify the slot number of the device where you want text or listings displayed. (Normally this is slot #Ø, the Apple II video screen.) This allows "chaining" of the SSC to another card slot, such as an 8Ø-column-display peripheral card. For the firmware in the SSC to pass on information to the firmware in the other card, the other card must have an output entry point within its CsØØ space; this is the case for all currently available 8Ø-column-display cards for the Apple II.

For example, let's say you have the SSC in slot #2 with a remote terminal connected to it, and an 80-column-display card in slot #3. Type  $\langle \text{CTRL-A} \rangle 3S \langle \text{RETURN} \rangle$  to cause the data from the remote terminal to be chained through the card in slot #3, so that it is displayed on the Apple II in 80-column format. (Not available in Pascal.)

# Echo Characters on the Screen- $E_{\langle E/D \rangle}$

For the Apple II, as for most computers, displaying (echoing) a character on the video screen is a separate step from receiving it from the keyboard, though we tend to think if these as one step, as on a typewriter. For example, if you type in <CTRL-A>E D<RETURN>, the SSC does not forward incoming characters to the Apple II screen. This can be used to hide someone's password entered at a terminal, or to avoid double-display of characters.

N.

1

SK.

# TERMINAL MODE

Under Communication Node, the SSC can enter Terminal Mode and make the Apple II act like an unintelligent terminal. This is useful for connecting the Apple II to a computer timesharing service, or for conversing with another Apple II.

Terminal Mode makes it possible to generate lowercase characters, plus the ten ASCII characters not provided on the Apple II keyboard (plus ESC, since <ESC) is used for this feature).

To generate lowercase characters, press <ESC> (the "ESCAPE" key near the upper left corner of the Apple II keyboard) once, and then type alphabetic characters as you would normally do. After that, to capitalize a single letter, press <ESC> again before typing the letter. To lock the keyboard in uppercase, press <ESC> twice in succession. To get back to lowercase, press <ESC> once, as before.

To generate one of the special ASCII characters listed in Table 3-10, first press  $\langle ESC \rangle$  once (if necessary) to place the keyboard in lowercase mode. Then press  $\langle ESC \rangle$  a second time, followed by one of the top-row keys as shown in Table 3-10. For example, to send a tilde, make sure the keyboard is in lowercase mode, then type  $\langle ESC \rangle$  followed by 9.

<esc> followed by:</esc>	1	2	3	4	5	6	7	8	9	Ø	:
generates:	FS	US	[	1	_	{	1	}	~	ESC	RUB
or in hexadecimal:	9C	9F	DB	DC	DF	FB	FC	FD	FE	9 B	FF

Table 3-1∅. Special ASCII Character Generation

#### TERMINAL MODE COMMANDS

The commands that specifically affect Terminal Mode are listed in Table 3-11. The Translate, Echo and XOFF commands are described earlier in this chapter.

Format	Command Name	Interpretation
T	Enter Terminal Mode	Go into Terminal Mode.
В	Transmit a Break Signal	Send a 233-millisecond BREAK (signoff) signal.
* E_ <e d=""></e>	Echo Enable/Disable	Default E D (full-duplex); use E I for half-duplex.
S_ <e d=""></e>	Special Characters Enable/Disable	Default S E; allows/defeats generation of lowercase and special characters (Table 3-10).
* <n>T</n>	Translate Lowercase Characters	Determine treatment of incoming lowercase characters.
* X_ <e d=""></e>	XOFF Recognition Enable/Disable	Default X E; in Terminal Mode, X makes SSC detect <ctrl-r> and <ctrl-t> (remote-control OFF &amp; ON respectively), but not <ctrl-s>.</ctrl-s></ctrl-t></ctrl-r>
Q	Quit (Exit from) Terminal Mode	Return to normal Communications Mode operation.

Table 3-11. Terminal Mode Commands

# Enter Terminal Mode-T

This causes the Apple II to function as a full-duplex unintelligent terminal. You can use this command in conjunction with the ECHO command to simulate the half-duplex terminal mode of the old Apple II Communications Card. Type <CTRL-A>T<RETURN> to enter this mode.



If you enter Terminal Mode and don't see what you type echoed on the Apple video screen, probably the modem link has not yet been established, or you need to use the E(cho E(nable command.

## Transmit a Break Signal-B

Typing <CTRL-A>B<RETURN> causes the SSC to transmit a 233-millisecond break signal, recognized by most time-sharing systems as a signoff.

## Special Characters-S\_(E/D)

Typing <CTRL-A>S E<RETURN> causes the SSC to interpret <ESC><n> pairs as special characters, allowing a keyboard in this way to generate all possible ASCII characters. If you type <CTRL-A>S D<RETURN>, the SSC will treat the <ESC> key like any other key.

# Quit (Exit from) Terminal Mode-Q

Type (CTRL-A)Q(RETURN) to exit from terminal mode.

#### A TERMINAL MODE EXAMPLE

You can use the sample program below to change the SSC temporarily from the characteristics you ordinarily use, to the characteristics needed to make the Apple II into a dumb terminal connected to the Dow Jones News & Quotes Reporter. This program assumes that the SSC is set for Communications Mode and that the jumper block is pointing toward MODEM. Neither of these conditions can be changed by software. This program also assumes that the SSC is in slot #1 and that you want to chain I/O to an 80-column card in slot #3; these conditions you can change via software. To change this Integer BASIC program to an Applesoft program, substitute CHR\$(5) for D\$ and CHR\$(2) for A\$, and leave out program lines 40 and 42.

Wind.

S.

```
10 REM **********************
20 REM * THIS PROGRAM SETS UP THE SSC FOR DOW JONES *
30 REM ******************
40 DS="": REM TYPE <CTRL-D> ESCAPE CHARACTER BETWEEN QUOTES
42 AS="": REM TYPE <CTRL-A> COMMAND CHARACTER BETWEEN QUOTES
5Ø PRINT D$;"PR#1":
                         REM SSC IS IN SLOT #1;
52 PRINT AS;"6 BAUD": REM SET BAUD RATE TO 300;
54 PRINT AS;"1 DATA": REM DATA FORMAT OF 7 DATA, 1 STOP
56 PRINT AS;"0 PARITY": REM AND NO PARITY;
58 PRINT AS; "LF DISABLE": REM NO < LF> GENERATION AFTER < CR>.
60 PRINT AS;"3 SLOTCHN": REM CHAIN TO CARD IN SLOT #3
62 PRINT AS;"TERM MODE": REM AND ENTER TERMINAL MODE.
 7Ø REM ****************************
 72 REM * NOW YOU SHOULD BE IN TERMINAL MODE, GETTING THE *
 74 REM * INFO YOU NEED FROM THE DOW JONES SERVICE. WHEN
 76 REM * FINISHED, EXIT WITH THE <CTRL-A>Q(UIT COMMAND.
 78 REM ****************************
100 REN QUIT COMMAND SENDS CONTROL BACK TO THIS PROGRAM:
110 PRINT AS; "RESET": REM RESET SWITCH-SELECTED OPTIONS
120 END
```

# CHAPTER 4 HOW THE SCC WORKS

- 18

-

-18

in the

mode

28

- 18

18

-16

18

-

-

1

26

L L L L L

- 18

78

-

This chapter is divided into three major sections. The first explains what the SSC does, using everyday terms wherever possible. Those of you already familiar with serial data communication can skip this section.

The second section is for anyone who wants an overview of the SSC's operating modes and configuration possibilities.

The third section is a dyed-in-the-wool hardware theory of operation for both the expert and the adventuresome layperson.

# SERIAL DATA COMMUNICATION

The SSC is a device that performs <u>serial</u> data communication. Let's consider <u>communication</u> first, then <u>data</u>, and then <u>serial</u> data and data transfer.

Communication is easy enough: getting information from here to there or from there to here. In this discussion, the Apple II is "here." "There" can be nearby (local) or far enough away (remote) that some intermediate device, like a telephone, is needed. Information moving from here to there (out of the Apple) is called output; information moving from there to here (into the Apple) is called input.

<u>Data</u> denotes information in its many forms. For successful data communication, it is essential that both the sender and receiver agree on their interpretation of the data transferred.

Inside the Apple II, data can be numbers and letters and symbols, or program instructions for the computer to carry out, or pointers to storage locations, or error message numbers, or codes for generating pictures or sounds (or lots of other things).

In the Apple II, as in all other computers, data is represented in codes made up of ones and zeros, the only two digits allowed in the binary (two-element) system. Each one or zero is called a BInary digiT or bit. In the binary system, as in our ordinary decimal

system, you can count to as high a number as you want -- it just takes more digits to get there than in the decimal system -- and use each number as a code to represent that number of different items. Table 4-1 gives some examples of how many items you can represent with various quantities of digits.

System	Digits	Using	You can represent
decimal	Ø - 9	1	ten items (Ø through 9)
	S (2)	2	one hundred (Ø through 99)
		3	one thousand (Ø through 999)
binary	Ø and 1	1	two items (Ø or 1)
Drintry		2	four (0, 1, 10 or 11)
		3	eight (Ø through 111)
		4	sixteen (Ø through 1111)
		5	thirty-two (∅ through 11111)
		6	sixty-four (Ø through 111111)
		7	one hundred twenty-eight
		8	two hundred fifty-six, etc.

Table 4-1. Binary and Decimal Digits and Quantities

For printers, plotters, terminals, and many other devices, 128 codes are enough to distinguish all the necessary characters: 52 for the upper and lowercase alphabet, 10 for the decimal digits, and dozens of others for punctuation marks and special symbols. As a result, the 128-character American Standard Code for Information Interchange (ASCII) is widely used. (This 7-bit code is listed in Appendix D.)

Throughout the world, post, telegraph, telex and wire services use 5-bit and 6-bit code sets, even though so few bits cannot represent a very large selection of items. Meanwhile, computers have a penchant for sending each other streams of 8-bit codes with obscure meanings. As long as sender and receiver agree on interpretation, any set of codes will do. The SSC can send all of

# PARALLEL DATA IN THE APPLE II

The Apple II is called an eight-bit processor because the basic unit of data it uses and moves around internally is an eight-bit byte. The Apple II has sets of eight lines interconnecting its various internal parts, so it can move around all eight bits at the same time. Since the bits travel together like eight cars side by side on an eight-lane highway, data in the Apple II is called parallel data, and data movements within the Apple II are called parallel data transfers (Figure 4-1).

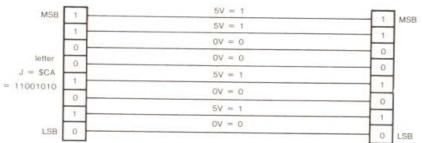


Figure 4-1. Parallel Data Transfer

# SERIAL DATA FOR LONG DISTANCES

Just as it would be extremely costly to build highways with eight lanes in each direction over great distances, so it is costly to connect two widely separated pieces of equipment using eight lines in each direction. So, many manufacturers produce computers, printers, plotters, terminals and so forth that send and receive information along one line in each direction, one bit after another. Such a setup, with bits moving from one place to another like a string of cars in a single lane, is called a serial data transfer (Figure 4-2).

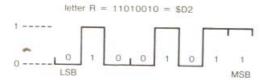


Figure 4-2. Serial Data Transfer

#### DATA CONVERSION

- 190

inches ministra

Changing parallel data to serial data or vice versa is called data conversion (Figure 4-3). By convention (see the later subsection describing RS-232-C), whenever parallel data is converted to serial data, the right-hand bit is sent first. It is as though there were a traffic law that when a multi-lane highway narrows to a single lane, the car in the right lane goes first, then the car from the next lane to the left, etc.

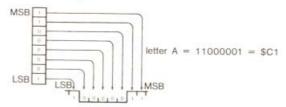


Figure 4-3. Parallel-to-Serial Data Conversion

#### **RS-232-C DATA FORMATS**

Serial data communication became popular so quickly that a group of manufacturers and the telephone company formed the Electronic Industries Association (EIA) to agree upon standard ways of sending and receiving data. What has become the most widely used standard in the world is called Revision C of standard RS-232, or RS-232-C. The SSC sends and receives data in accordance with this standard. The serial data has the form shown in Figure 4-3, plus a start bit at the beginning, an optional parity bit after the five to eight data bits, and finally one or two stop bits at the end (Figure 4-4). This is the data format that most RS-232-C devices use.

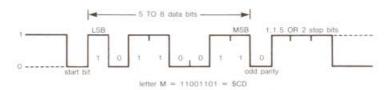


Figure 4-4. RS-232-C Serial Data Format

No.

Min.

What is this mysterious parity bit all about? It is an optional extra bit set to  $\emptyset$  or 1 to make the total number of data and stop bits set to 1 an odd number (odd parity) or an even number (even parity); or this extra bit can always be set to  $\emptyset$  (called SPACE parity) or to 1 (MARK parity).

The combined total of data and parity bits set to 1 in Figure 4-4 is 5, an odd number (and the parity bit is 1), so it qualifies as a correct character if odd parity (or MARK parity) has been agreed upon by sender and receiver. However, if that same character were received under even parity (or SPACE parity), the receiving device would signal that a transmission error had occurred. If one bit in a character changes during transmission, parity checking will detect the error. If two bits change, the error will go undetected.

#### RS-232-C SIGNALS

Since the RS-232-C standard stems from the early days of telephone and telegraph, the names given to its signals may sound quaint to our "modern" ears. However, the signals correspond to familiar conditions that we take for granted when using a telephone. Table 4-2 lists the basic signals required by the RS-232-C standard, and what conditions they correspond to in a telephone call that you originate. Think of yourself as the Data Terminal (a terminus or end point of the conversation), and the phone as the Data Set (the communication device). Note: not is indicated by a bar above a signal name.

RS-232-C Signal	Abbrev.	Similar to
Data Terminal Ready	DTR	you pick up the phone
Data Set Ready	DSR	the phone is working
Request To Send	RTS	you want to talk
Clear To Send	CTS	the phone has established a connection and the person at the other end is ready to listen
Transmit Data	TxD	you speak into the phone
not Request To Send	RTS	you've finished talking and are ready to listen or to hang up
not Clear To Send	CTS	the phone has sent your words and is ready for your next request to send a message
not Data Terminal Rdy	DTR	you hang up

Table 4-2. RS-232-C Signals As Interpreted by the Sender

Here are the RS-232-C signals and how you would interpret them if you were to answer a telephone call (Table 4-3).

RS-232-C Signal	Abbrev.	Similar to
Ring Indicator	RI	the phone rings (optional)
Data Set Ready	DSR	you pick up the phone; it works
Data Carrier Detect	DCD	you hear background noise
Receive Data	RxD	you hear what is said
not Data Set Ready	DSR	the other party has hung up

Table 4-3. RS-232-C Signals As Interpreted by the Receiver

#### Modems

All of the above signals refer to the interaction between what RS-232-C calls Data Terminal Equipment (DTE--end points of data transfers, such as the Apple II or a printer) and what it calls Data Communication Equipment (DCE--transmitting or receiving devices, such as modems).

What is a modem? The name is short for MOdulator/DEModulator. As a modulator it takes electrical signals from a computer or printer (or other device) that it is connected to, and turns them into musical tones over a telephone line. As a demodulator it takes the musical tones it detects on a telephone line and turns them back into electrical signals for use by the printer or computer (or other device) that it is connected to. It also handles the RS-232-C control signals to and from that device (Figure 4-5).

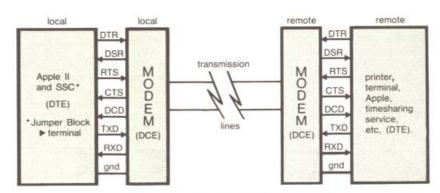


Figure 4-5. An RS-232-C Setup with Modems

By convention, the calling (originate) modem produces a fairly high tone (let's say LA) as the background or carrier signal that it sends; it then modulates (changes) that tone to SO to mean Ø and TI to mean 1. Meanwhile, the called (answer) modem plays a lower tone, MI, as a carrier signal, and modulates that tone to RE to indicate Ø or FA to indicate 1. In this way, both modems can send and receive information along the same wires without interpreting what they send as received messages and vice versa. (All their voices sound alike.)

#### **Modem Eliminators**

RS-232 signals are designed for the interactions of two DTE's, two DCE's, and telephone lines, as shown in Figure 4-5. What if you just want to connect two DTE's together in the same room, directly (for example, an Apple II and a printer)? You can use what is called a null modem or modem eliminator. The jumper block on the SSC does just that when it is connected with its triangle pointing toward the word TERMINAL.

By using different tones to send and receive information, modems can make sure that what comes from the "mouthpiece" (<a href="transmitregister">transmitregister</a>) of one DTE gets routed to the "earpiece" (<a href="receive-register">receive-register</a>) of the other. A null modem simply crosses those two wires (Figure 4-6).

To simulate the other signal exchanges that modems would perform, the null modem interconnects the signal wires as shown in Figure 4-6. Thus RTS gets turned back to the sender as CTS as though the phone had instantly established a connection; RTS is also connected to DCD on the other side to pretend that a carrier signal has been detected. Finally, connecting DTR (willing to transfer data) from one side to both RI and DSR (a call arriving) on the other side completes the simulated telephone connection. (RI is optional.) The jumper block does it all!

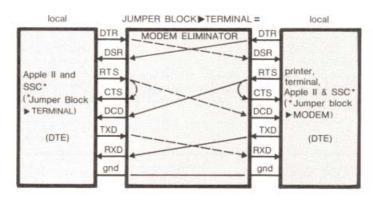


Figure 4-6. An RS-232-C Setup with a Modem Eliminator

# SSC MODES AND CONFIGURATIONS

178

1750

-158

Figure 4-7 outlines the possible operating modes of the Super Serial Card and their relationships to each other.

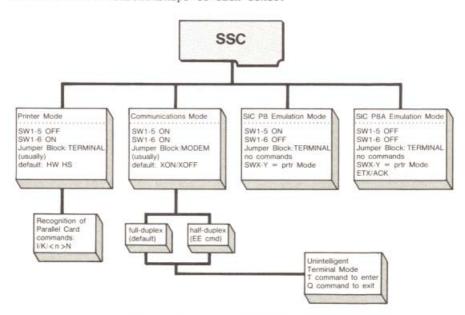


Figure 4-7. SSC Operating Modes

Figure 4-8 illustrates the chief configurations possible with the Super Serial Card and how to set them up.

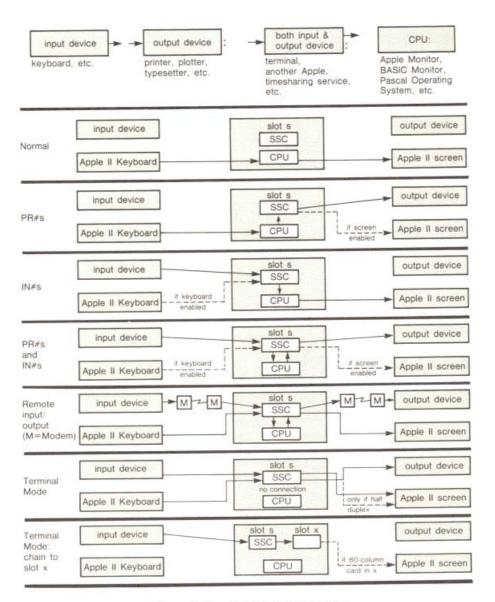


Figure 4-8. SSC Configurations

# THEORY OF OPERATION

This section explains the SSC's overall theory of operation, but not the internal workings of each IC chip. If you would like such information, it is best to obtain specifications from the IC manufacturers. The most complex component is the ACIA, which is a Synertek 6551 or equivalent.

While reading through this section, you may find it useful to refer to Figure 4-9, a block diagram of the SSC, or to the schematic diagram in Appendix C. All references in the form 1A, 3C, etc., pertain to coordinates on the printed circuit board itself. Here is an inventory of the main components of the SSC:

- 50-pin connection to the Apple II peripheral connector slot
- a 12-line address bus

15

- 19

- 25

100

-

150

- addressing and control logic (1B, 1C, 2C, 3C)
- a 2K-by-8-bit ROM (4B-5C)
- jumpers and bow ties for optional substitution of RAM (3-4A)
- two blocks of 7 switches each (IA, 2A)
- two registers for reading the switch settings (2B, 3B)
- a 1.8432 MHz oscillator (3A) for the ACIA
- a transmit interface (6A) and a receive interface (7A)
- an 8-line data bus
- a buffer for the data bus (6C)
- a jumper block (6B) that can function as a modem eliminator
- a 10-pin header (7B) to connect the SSC to a DB-25 jack via a short internal cable (discussed in Appendix C)

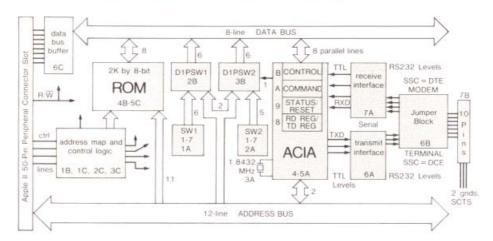


Figure 4-9. Overall Block Diagram of the SSC

#### ADDRESSING AND CONTROL LOGIC

The twelve address lines (A $\emptyset$  - All) from the Apple II provide all the necessary \$C $\emptyset$ 00 addressing on the SSC. Control logic at 1B, 1C, 2C and 3C, plus the signals RESET, DEVICE SELECT, I/O SELECT, and I/O STROBE, ensure the routing of signals to the appropriate addresses.

The SSC follows the Apple II protocol in its use of the \$C800 address space. An LS279 (1B) serves as a NAND gate, a pair of inverters, and a set-reset latch. The latch is set by an access to the \$Csxx space, and is reset by access to the \$CFxx space or by a reset. When this set-reset latch is set, the Apple II can access the \$C800 space on the SSC. A small RC filter prevents the latch from being reset by spurious noise.

R.

## ROM/RAM Space

The 2K ROM (4B-5C) containing the SSC driver firmware resides in the \$C800 - \$CFFF address space. However, an LS00 (2C) and an LS32 (3C) remap the addresses from the range \$C\$00 - \$C\$FF to the range \$CF00 - \$CFFF, since the \$CFXX addresses are unusable. (Access to them disables use of the \$C\$00 address space.) As a result of this remapping, only one ROM is required, and none of the ROM space is wasted.

The SSC can use a 2K-by-8-bit RAM in place of the ROM. Between columns 3 and 4 and rows A and B on the SSC, there are three jumper pads and three bow ties. If you solder the jumper pads and cut the bow ties, pins 18,  $2\emptyset$  and 21 will be, respectively, chip enable, output enable and read-write control (instead of ROM enables).

The ROM (or RAM) addresses are mapped as follows (Table 4-4). The first 256-byte block is the Peripheral Card ROM Space, selected when I/O SELECT from the Apple II drops to  $\emptyset$  volts. The remaining seven blocks are in the I/O Expansion ROM Space, selected when I/O STROBE from the Apple II drops to  $\emptyset$  volts.

SSC ROM/RAM Addresses	Become Apple II Addresses
\$Ø7ØØ - \$Ø7FF	\$CsØØ - \$CsFF
\$ØØØØ - \$ØØFF	\$C8ØØ - \$C8FF
\$Ø1ØØ - \$Ø1FF	\$C9ØØ - \$C9FF
\$Ø2ØØ - \$Ø2FF	\$CAØØ - \$CAFF
\$Ø3ØØ - \$Ø3FF	\$CBØØ - \$CBFF
\$Ø4ØØ - \$Ø4FF	\$CCØØ - \$CCFF
\$Ø5ØØ - \$Ø5FF	\$CDØØ - \$CDFF
\$Ø6ØØ - \$Ø6FF	\$CEØØ - \$CEFF

Table 4-4. SSC Address Remapping

# Registers in Peripheral I/O Space

100

-

....

PER

-

Whenever DEVICE SELECT drops to  $\emptyset$  volts, the Apple II is addressing the SSC's Peripheral I/O Space (the sixteen bytes starting at  $\$C\emptyset 8\emptyset + \$\emptyset$ ). This signal is combined logically with address lines A $\emptyset$  through A3 to select one of the six registers that reside in that space (Table 4-5).

Chip selected	Address(+s∅)	Purpose of register
LS365 (2B)	\$CØ81	store state of SW1 (1A) (read)
LS365 (3B)	\$CØ82	store state of SW2 (2A) and state of CTS (read)
ACIA (4-5A)	\$CØ88	receive (read), transmit (write)
ACIA (4-5A)	\$CØ89	status (read), reset (write)
ACIA (4-5A)	\$CØ8A	command (read and write)
ACIA (4-5A)	SCØ8B	control (read and write)

Table 4-5. Registers in SSC Peripheral I/O Space

The two LS365 chips act as buffers so that the state of eleven of the fourteen available switches, plus the state of RS-232-C signal Clear To Send (CTS), can be read. There are 3.3K ohm pullup resistors at the switch inputs of the LS365 chips. A closed switch pulls down an input, and it is read as zero.

Three switches are not connected to the LS365s. Switch SW2-6, when ON, passes interrupt requests from the ACIA to the Apple II. (The Apple II, however, currently does not support interrupts.) Setting switches SW1-7 ON and SW2-7 OFF connects DB-25 pin 8 (DCD) to the DCD input of the ACIA. Setting SW1-7 OFF and SW2-7 ON splices pin 19, Secondary Clear To Send (SCTS), onto the DCD input of the ACIA when the jumper block is in the TERMINAL position.

The ACIA has two pins used to select one of its four registers. While address lines A2 and A3 select the chip, AØ and A1 select the actual register. The SSC firmware reads and writes ACIA register contents; these registers are discussed in detail in Appendix A.

#### THE ACIA

The Asynchronous Communications Interface Adapter (ACIA) is the central and most complex element of the SSC. It and the crystal at 3A form a 1.8432 MHz oscillator. The ACIA divides this frequency down to one of the fifteen baud rates it supports. The ACIA also handles all incoming and outgoing primary RS-232-C signals. The ACIA registers (discussed fully in Appendix A) control hardware handshaking and select the baud rate, data format and parity. Finally, the ACIA performs parallel/serial and serial/parallel data conversion, and single-buffers data transfers.

## DATA INPUT AND OUTPUT

The MC1489 at 7A converts the incoming serial data from RS-232-C to TTL voltage levels. The MC1488 at 6A converts the outgoing serial data from TTL to RS-232-C voltage levels, and in conjunction with three capacitors limits the output slew rate. Three of the received handshake lines (Clear To Send, Data Carrier Detect, and Data Set Ready) have 15K ohm pullup resistors so the SSC will work with devices that do not assert those signals.

#### **DATA BUS**

The 8-bit data bus on the SSC is, of course, a parallel bus. The ACIA takes output from it and gives input to it in parallel form. Also connected to the bus are the two switch detection registers (2B and 3B) and the ROM or RAM chip.

An LS245 (6C) buffers the output to the data bus, and minimizes input loading. The data bus has a 3.3K ohm pullup resistor on each line so the data inputs on the LS245 are not floating when it turns on in output mode.

#### JUMPER BLOCK

The jumper block has two positions: when its arrow points toward MODEM, the SSC looks like Data Terminal Equipment (DTE); that is, the SSC is prepared to talk to Data Communication Equipment (DCE), such as a modem. When installed with its arrow pointing toward TERMINAL, the jumper block acts as a modem eliminator (null modem); that is, the SSC looks like the DCE on the other device's side of a serial communication connection. In this position, the SSC can talk directly to a printer or any other DTE. Figure 4-6 shows the signal swapping that the jumper block in the TERMINAL position performs.

# APPENDIX A FIRMWARE

This appendix contains the following information:

- an explanation of the Pascal 1.1 firmware card protocol
- · a firmware memory map
- a description of the SSC's use of its peripheral slot scratchpad RAM addresses
- a description of the ACIA registers and switch detection registers in the SSC's peripheral I/O space
- a list of firmware entry points and 6502 register values
- the actual SSC firmware listings

## PASCAL 1.1 FIRMWARE PROTOCOL

The old Apple II Serial Interface Card (SIC) ran under Pascal  $1.\emptyset$  with three direct firmware entry points, one for each of the three I/O functions it supported:

Address	Contains
\$C8ØØ \$C84D	initialization routine entry point read routine entry point
\$C9AA	write routine entry point

New peripheral cards can be "accepted" into the Pascal 1.0 system by appearing to be a SIC; that is, with these same three entry points and with \$38 at \$Cs05 and \$18 at \$Cs05 (see Device ID section below).

Pascal 1.1, on the other hand, has a more flexible setup, and also supports more I/O functions. It can make indirect calls to the firmware in a (new) peripheral card through addresses in a branch table in the card's firmware. It also has facilities for uniquely identifying new peripheral I/O devices.

I/O ROUTINE ENTRY POINTS

The I/O routine entry point branch table is located near the beginning of the Cs00 address space (s being the slot number where the peripheral card is installed). This space was chosen instead of the \$CS00 space, since under BASIC protocol the \$Cs00 space is required, while the \$C800 space is optional.

The branch table locations that Pascal 1.1 uses are:

Address	Contains
\$CsØD	initialization routine offset (required)
\$CsØE	read routine offset (required)
#CsØF	write routine offset (required)
\$Cs10	status routine offset (required)
\$Cs11	\$00 if optional offsets follow; non-zero if not
\$Cs12	control routine offset (optional)
SCs13	interrupt handling routine offset (optional)

Notice that \$Csll contains \$DD only if the control and interrupt handling routines are supported by the firmware. (For example, the SSC does not support these two routines, and so location \$Csll contains a (non-zero) firmware instruction.) Apple II Pascal 1.0 and 1.1 do not support control and interrupt requests, but such requests may be implemented in future versions of the Pascal BIOS and other future Apple II operating systems.

Here are the entry point addresses, and the contents of the 65%2 registers on entry to and on exit from Pascal 1.1 I/O routines:

Addr.	Offset for	X Register	Y Register	A Register
\$CsØD	Initialization On entry On exit	\$Cs error code	\$sØ (unchanged)	(unchanged)
\$CsØE	Read On entry On exit	\$Cs error code	\$sØ (unchanged)	character read
\$CsØF	Write On entry On exit	\$Cs error code	\$s∅ (unchanged)	char. to write (unchanged)
\$Cs10	Status On entry On exit		\$s∅ (changed)	request (Ø or 1) (unchanged)
Notes:	Request code 1	means, "Do y	ou have input tatus request	is in the carry

Table A-1. I/O Routine Offsets and Registers under Pascal 1.1

#### DEVICE IDENTIFICATION

Pascal 1.1 uses four firmware bytes to identify the peripheral card. Both the identifying bytes and the branch table are near the beginning of the CSOO ROM space. The identifiers are listed in Table A-2.

Address	Value						
\$CsØ5	\$38 (like the old Serial Interface Card)						
\$CsØ7	\$18 (like the old Serial Interface Card)						
\$CsØB	\$Ø1 (the Generic Signature of new FW cards)						
SCsØC	\$ci (the Device Signature; see below)						

Table A-2. Bytes Used for Device Identification

The first digit, c, of the Device Signature byte identifies the device class as listed in Table A-3.

Digit	Class
\$Ø	reserved
\$1	printer
\$2	joystick or other X-Y input device
\$2 \$3	serial or parallel I/O card
\$4	modem
\$5	sound or speech device
\$6	clock
\$7	mass storage device
\$8	8Ø-column card
\$9	network or bus interface
\$A	special purpose (none of the above)
\$B-F	reserved for future expansion

Table A-3. Device Class Digit

The second digit, i, of the Device Signature byte is a unique identifier for the card, assigned by Apple Technical Support. For example, the SSC has a Device Signature of \$31: the 3 signifies that it is a serial or parallel I/O card, and the 1 is the low-order digit supplied by Apple Technical Support.

Although version 1.1 of Pascal ignores the Device Signature, applications programs can use them to identify specific devices.

# SSC FIRMWARE MEMORY USAGE

Table A-4 is an overall map of the locations that the SSC uses, both in the Apple II and in the SSC's own firmware address space.

E.

Ni.

Addresses	Name of area	Contents
\$ØØØØ-\$ØØFF	Page Zero	Monitor pointers, I/O hooks, and temporary storage (Table A-5)
\$Ø4xx-\$Ø7xx (selected locations)	Peripheral Slot Scratchpad RAM	Locations (8 per slot) in Apple's pages \$04 through \$07. SSC uses all eight of them (Table A-6)
\$CØ(8+s)Ø - \$CØ(8+s)F	Peripheral Card I/O Space	Locations (16 per slot) for general I/O; SSC uses 6 bytes (Table A-7)
\$CsØØ-\$CsFF	Peripheral Card ROM Space	One 256-byte page reserved for card in slot s; first page of SSC FW
\$C8ØØ-\$CFFF	Expansion ROM	Eight 256-byte pages reserved for a 2K ROM or PROM; SSC maps its FW onto \$C800-\$CEFF (Table 4-4)

Table A-4. Memory Usage Map

#### ZERO PAGE LOCATIONS

The SSC makes use of these zero-page locations (Table A-5):

I	Address	Name	Description
*	\$24	CH	Monitor pointer to current position of cursor on screen
	\$26	SLOT16	Usually (slot# x 16); that is, \$sØ
	\$27	CHARACTER	Input or output character
*	\$28	BASL	Monitor pointer to current screen line
	\$2A	ZPTMP1	Temporary storage (various uses)
	\$2B	ZPTMP2	Temporary storage (various uses)
	\$35	ZPTEMP	Temporary storage (various uses)
*	\$36	CSWL	BASIC output hook (not for Pascal)
*	s37	CSWH	(high byte of CSW)
*	\$38	KSWL	BASIC input hook (not for Pascal)
*	\$39	KSWH	(high byte of KSW)
*	\$4E	RNDL	random number location, updated when looking for a keypress (not used when initialized by Pascal)

<sup>\*</sup> Not used when Pascal initializes SSC.

Table A-5. Zero-Page Locations Used by SSC

# SCRATCHPAD RAM LOCATIONS

थि। थि। थि। थि। थि। थि। थि। थि। थि। थि।

100

The SSC uses the Scratchpad RAM locations as listed in Table A-6.

Address	Field name	Bi	t	(s)	Interpretation
\$Ø478+s	DELAYFLG	Ø	_	1	<ff> delay selection</ff>
		2			<pre><lf> delay selection</lf></pre>
				5	<cr> delay selection</cr>
		6			Translate option
					Translate Option
\$Ø4F8+s	HANDSHKE	Ø	_	7	Buffer count for handshake (P8A Mode)
	PARAMETER	Ø	-	7	Accumulator for FW's command processor
\$Ø578+s	STATEFLG	Ø	-	2	Command mode when not Ø (Printer and
					Communications Modes only)
		1		4	Enquire character (P8A Mode); dflt ETX
		3		5	Slot to chain to (Communications Mode)
			6		Set to 1 after lowercase input character
			7		Terminal Mode when 1 (Comm Mode)
			7		Enable <cr> gen. when 1 (other 3 modes)</cr>
\$Ø5F8+s	CMDBYTE	Ø	_	6	Printer Mode default is <ctrl-i>;</ctrl-i>
					Comm Mode default is (CTRL-A)
			7		Set to 1 to Zap control commands
\$Ø678+s	STSBYTE				Status and IORESULT byte (Appendix F)
\$Ø6F8+s	CHNBYTE	Ø	_	2	Current Apple screen slot (Comm Mode);
					when slot = Ø, chaining is enabled
		3	-	7	\$Cs00 space entry point (Comm Mode)
	PWDBYTE	Ø	_	7	Current printer width (other modes);
					for listing compensation, auto- <cr></cr>
\$Ø778+s	BUFBYTE	Ø	-	6	One-byte input buffer (Comm Mode); used
					in conjunction with XOFF recognition
			7		Set to 1 when buffer full (Comm Mode)
	COLBYTE	Ø	-	7	Current-column counter for tabbing,
					etc. (other 3 modes)
\$Ø7F8+s	MISCFLG		Ø		Generate <lf> after <cr> when 1</cr></lf>
			1		Printer Mode when Ø; Comm Mode when 1
			2		Keyboard input enabled when 1
			3		<ctrl-s> (XOFF), <ctrl-r> and <ctrl-t></ctrl-t></ctrl-r></ctrl-s>
					input checking when 1
			4		Pascal Op Sys when 1; BASIC when Ø
			5		Discard (LF) input when 1
			6		Enable lowercase and special character
					generation when 1 (Comm Mode)
			6		Tabbing option on when 1 (Printer Mode)
			7		Echo output to Apple screen when 1

Table A-6. Scratchpad RAM Locations Used by SSC

# PERIPHERAL CARD I/O SPACE

There are 16 bytes of I/O space allocated to each slot in the Apple II. Each set begins at address  $C\emptyset B\emptyset + (\text{slot x 16})$ ; for example, if the SSC is in slot 3, its group of bytes extends from  $C\emptyset B\emptyset$  to  $C\emptyset BF$ . Table A-7 interprets the 6 bytes the SSC uses.

Address	Register	Bit(s)	Interpretation
\$CØ81+sØ	DIPSW1 (SW1-x)	Ø 1 4 - 7	SW1-6 is OFF when 1, ON when $\emptyset$ SW1-5 is OFF when 1, ON when $\emptyset$ same as above for SW1-4 through SW1-1
\$CØ82+sØ	DIPSW2 (SW2-x)	Ø 1 - 3 5 & 7	Clear To Send (CTS) is true (-) when $\emptyset$ same as above for SW2-5 through SW2-3 same as above for SW2-2 & SW2-1
\$CØ88+sØ	TDREG RDREG	Ø - 7 Ø - 7	ACIA Transmit Register (write) ACIA Receive Register (read)
\$CØ89+sØ	STATUS	Ø 1 2 3 4 5 6 7	ACIA Status/Reset Register Parity error detected when 1 Framing error detected when 1 Overrun detected when 1 ACIA Receive Register full when 1 ACIA Transmit Register empty when 1 Data Carrier Detect (DCD) true when Ø Data Set Ready (DSR) true when Ø Interrupt (IRQ) has occurred when 1
\$CØ8A+sØ	COMMAND	Ø 2 - 3 4 5 - 7	ACIA Command Register (read/write) Data Terminal Ready (DTR): enable (1) or disable (0) receiver and all interrupts When 1, allow STATUS bit 3 to cause IRQ Control transmit interrupt, Request To Send (RTS) level, and transmitter When 0, normal mode for receiver; when 1 echo mode (but bits 2 and 3 must be 0) Control parity (values: Table 2-7)
\$CØ8B+sØ	CONTROL	Ø - 3 4 5 - 6	ACIA Control Register (read/write) Baud rate: $\$\emptyset = 16$ times external clock; $\$1 - \$F = \text{decimal}$ in Table 2-5 When 1, use baud rate generator; when $\emptyset$ , use external clock (not supported) Number of data bits: 8 (bit 5 and 6 = $\emptyset$ ) 7 (5 = 1, 6 = $\emptyset$ ), 6 (5 = $\emptyset$ , 6 = 1) or 5 (bit 5 and 6 both = 1) Number of stop bits: 1 (bit 7 = $\emptyset$ ); if bit 7 = 1, then 1-1/2 (with 5 data bits, no parity), 1 (8 data plus parity) or 2

Table A-7. SSC Registers in Peripheral Card I/O Space

# SSC ENTRY POINTS

This section contains the SSC firmware entry points for the Apple II Monitor, BASIC, Pascal 1.0 and Pascal 1.1. The Pascal 1.1 entry point offsets conform to the Firmware card protocol outlined in the first section of this appendix.

#### MONITOR ROM ENTRY POINTS

The SSC uses these entry points in the Monitor ROM, unless Pascal initializes the SSC.

Address	Name	Description
\$FDED	COUT	sends a character to output hook (chaining)
\$FE89	SETKBD	sets KSW to point to keyboard (reset)
\$FE93	SETSCR	sets CSW to point to Apple screen (reset)
\$FF58	IORTS	known position of an RTS instruction
\$FDF6	VIDOUT	sends a character to the Apple screen

Table A-8. Monitor ROM Entry Points Used by SSC

#### **BASIC ENTRY POINTS**

-20

Here are the entry point addresses, and the contents of the  $65\emptyset2$  registers on entry to and on exit from BASIC I/O routines:

Addr.	Routine	X Register	Y Register	A Register
\$CsØØ	Initialization On entry On exit	anything (unchanged)	anything (unchanged)	
Notes:		oints to \$CsØØ.	The character	in the A
\$CsØ5	Input On entry On exit	anything (unchanged)	4 60	anything character in
Notes:	Character in is			
\$CsØ7	Output On entry On exit	anything (unchanged)	anything	character out
Notes:	Character out is			(changed)

Table A-9. BASIC Entry Points Used by SSC

## **PASCAL 1.O ENTRY POINTS**

There are three Pascal 1.0 entry points: one for initialization, one for read operations, and one for write operations. These entry points are direct addresses.

Niles Miles

Ni.

Ni.

Addr.	Routine	X Register	Y Register	A Register
\$C8ØØ	Initialization	í		
	On entry	\$Cs	\$sØ	anything
	On exit	\$Cs	\$sØ	(unchanged)
Notes:	\$C800 space is	enabled. Firmwa	are initializes	SSC to default
	values plus SW	1 and SW2 select	ions.	
\$C84D	Read			
Q004D	On entry	\$Cs	\$sØ	anything
	On exit	\$Cs		character in
Notes:		enabled. Pasca		
		ter and location		
\$C9AA	Write			
	On entry	\$Cs	\$sØ	character out
	On exit	error code	\$Cs	(changed)
Notes:	\$C800 space is	enabled. Outpu	t character is	transmitted
		IA. Pascal post		

Table A-1∅. Pascal 1.∅ Entry Points Used by SSC

# **PASCAL 1.1 ENTRY POINTS**

The Pascal 1.1 entry point protocol is outlined in the first section of this appendix. The values given here are the addresses of the routines. Unlike Pascal 1.0, Pascal 1.1 enters these routines using indirect addressing.

Addr.	Offset for	Value	X Register	Y Register	A Register
40000	Initialization On entry On exit		\$Cs \$ØØ	\$50	anything (changed)
Notes:	\$C800 space is values plus SW1	enabled. and SW2	Firmware selections	initializes :	SSC to default
\$CsØE	Read On entry On exit	\$(Cs)94	\$Cs	\$sØ \$Cs	
Notes:	\$C800 space is is returned in			in from ACI	A or keyboard
\$CsØF	Write On entry On exit	\$(Cn)97	\$Cs	ŞsØ ŞCs	
Notes:	\$C800 space is enabled. out through the ACIA.		The byte	in the A Reg	ister is sent
\$Cs1Ø	On entry On exit	\$(Cs)9A	\$Cs error code	\$sØ \$sØ	
Notes:	\$C800 space is enabled. Request = 0 asks ACIA whether it is ready to transmit another byte; request = 1 asks ACIA whether it has an input character available. On exit, carry bit = 0 for Yes or 1 for No.				

Table A-11. Pascal 1.1 Offsets Used by SSC

# OTHER SPECIAL FIRMWARE LOCATIONS

-

The SSC firmware uses several other addresses for predefined purposes. Table A-12 lists these locations.

Address	Value	Purpose		
\$CsØ5 \$38		Pascal serial/firmware card identifier (as well as BASIC input entry point)		
\$CsØ7	\$18	Pascal serial/firmware card identifier (as well as BASIC output entry point)		
\$CsØB	şø1	Pascal 1.1 generic signature byte (\$\psi 1 = firmware card)		
\$CsØC	\$31	Pascal 1.1 Device Signature byte (\$31 = serial or parallel I/O card #1)		
\$Csll	\$85	Pascal 1.1 optional routines flag (nonzero value = not supported)		
ŞCsFF	\$Ø8	Firmware revision level		

Table A-12. SSC Special Firmware Locations

# SSC FIRMWARE LISTINGS

```
00000:
   0000:
                    4 * APPLE II SSC FIRMWARE
   0000:
   00000
   0000:
                    6 * BY LARRY KENYON
                          -JANUARY 1981-
   0000:
                    7 *
   0000:
                    8 *
   0000:
                     9 * (C) COPYRIGHT 1981 BY APPLE COMPUTER, INC. *
   0000:
                   10 *
                    11 *************************
   0000:
0000:
   0000:
12 *
   ;DISABLES CO-RES. $C800 ROMS
             44 DIPSW1 EQU $C081 ;(+$N0) DIPSWITCH BLOCK 1
45 DIPSW2 EQU $C082 ;(+$N0) DIDGWITCH BLOCK 1
   0000:
   0000
C081:
                 46 TDREG EQU $C088
47 RDREG EQU $C088
                                           ;(+$NO) TRANSMIT DATA REG (WRITE)
;(+$NO) READ DATA REG (READ)
   CORR:
   C088:
                  48 STREG EQU $C089 ;(+$NO) STATUS REGISTER (READ)
49 RESET EQU $C089 ;(+$NO) SOFTWARE RESET (WRITE)
50 CMDREG EQU $C08A ;(+$NO) COMMAND REGISTER (R/W)
51 CTLREG EQU $C08B ;(+$NO) CONTROL REGISTER (R/W)
   C089:
   C089:
   COBA:
   COSB:
```

```
0000:
                               53 ********************************
                               54 * BIT-> B7 B6 B5 B4 B3 B2 B1 B0
             00000:
                               55 *
             00000:
                               56 * DIPSW1 S1 S2 S3 S4 Z Z S5 S6 (LEFT DIPSWITCH)
             0000:
             0000:
             0000:
                               58 * (S1-S4 USED FOR BAUD RATE, S5-S6 FOR FIRMWARE MODE)
             0000:
                               59 *
             0000:
                               60 * DIPSW2 S1 Z S2 Z S3 S4 S5 CTS (RIGHT DIPSWITCH)
             00000:
                               61 *
                               62 * STREG INT DSR DCD TDR RDR OVR FE PE
             0000:
             00000:
                               63 *
                               64 * CTLREG STB << WL >> CK << BAUD RATE >>
             00001
             0000:
                               65 *
             0000:
                               66 * CMDREG <<PARITY >> ECH <<XMIT>> RE DTR
             0000:
                               67 :
                               68 *********************************
             0000:
                               69 ***************
             00000:
 10
                               70 * SCREEN VARIABLES: PPC AND SIC MODES *
             00000:
             0000:
                               72 CMDBYTE EQU $5F8-$CO ;HOLDS COMMAND CHARACTER (PPC & CIC)
73 HANDSHKE EQU $4F8-$CO ;SIC P8A CHAR COUNTER FOR ETX/ACK
             0538:
 -- 198
             0438:
                               74 PARAMETER EQU $4F8-$CO ; ACCUMULATOR FOR CMD PARAMETER
                               75 STATEFLG EQU $578-$CO ;
76 * B7=CR GEN ENB FLAG B6=AFTER LC INPUT FLG
             0488:
                               77 * B2-B0=COMMAND INTERPRETER STATES
             0000:
                               78 * 0 0 0 IDLE
             0000:
 -10
                               79 * 0 0 1 CMD CHAR RECEIVED
80 * 0 1 0 COLLECT (N> UNTIL CHAR THEN DO COMMAND
             00000:
             10000
                               81 *
                                      0 1 1
                                              SKIP UNTIL SPACE, THEN GOTO STATE 4
                               82 *
             00000:
                                      1 0 0
                                              E/D COMMANDS
             0000:
                               83 *
                                      1 0 1
                                              UNUSED
                               84 * 1 1 0 WAIT UNTIL CR THEN SET STATE TO ZERO
85 * 1 1 1 WAIT UNTIL CR THEN DO PROC INDICATED BY PARM
 -18
             0000:
                               86 *
             00000:
                               87 * (B4-B0 DETERMINE ENQUIRE CHAR FOR P8A MODE)
             0000:
             0000:
             0388:
                               89 DELAYFLG EQU $478-$CO
 -15
                               90 * B7-B6=SCREEN TRANSLATION OPTIONS
             0000:
                               91 * 0 0 LC->UC

92 * 0 1 NO TRANSLATION

93 * 1 0 LC->UC INVERSE

94 * 1 1 LC->UC, UC->UC INVERSE
             0000:
             0000:
-
             0000:
             0000:
                               95 * (1-3 WILL ALLOW LC CHARS TO PASS THRU MONITOR)
             0000:
             0000:
                               96 *
                               97 * B5-B4=CR DELAY
                                                       O O = NO DELAY
             0000:
                               98 * B3-B2=LF DELAY 0 1 = 32 MILLISEC
99 * B1-B0=FF DELAY 1 0 = 1/4 SEC
             0000:
             0000:
             0000:
                              100 *
                                                       1 1 = 2 SEC
             0000:
                              101 *
                              102 STSBYTE EQU $678-$CO ;STATUS/IORESULT/INPUT BYTE 103 PWDBYTE EQU $678-$CO ;PRINTER (FORMAT) WIDTH
             05B8:
             0638:
             06B8:
                               104 COLBYTE EQU $778-$CO ; COLUMN POSITION COUNTER
             0738:
                               105 MISCPLG EQU $7F8-$C0 ;
                                                          B6=TABBING OPTION ENABLE
                               106 * B7=ECHO BIT
             00000:
                              107 * B5=LINEFEED EAT
                                                          B4=PASCAL/BASIC FLAG
             00000:
                              108 * B3=XOFF ENB FLAG B2=KEYBOARD ENB
109 * B1=PPC/CIC MODE B0=LF GENERATE ENB
             00000:
             00000:
             0000:
```

```
112 ***********************
              113 * TEMP SCREEN VARS (SLOT INDEPENDENT) *
              114 ***********************
0000:
                                         ;BUFFER FOR HI SLOT ADDR ($CN)
               115 MSLOT EQU $7F8
07F8:
               116 **********************
               117 * SCREEN VARIABLES: CIC MODE
               118 *********************
0000:
               119 *
0000:
               120 * STATEFLG: B7=TERMINAL MODE FLAG
0000:
               121 *
00000:
               122 *
               123 CHNBYTE EQU $6F8-$CO ; CURRENT OUTPUT SCREEN ($CN00 ENTRY)
0638:
0000:
0000:
               125 * BO-B7=CNOO ENTRY
0000:
               126 *
               127 BUFBYTE EQU $778-$CO ; BUFFER FOR ONE
06B8:
                                           INPUT BYTE: HIGH BIT IS SET
0000:
               128 *
               129 *
                                          WHEN BUFFER IS FULL
00000:
0000:
               131 * MISCFLG:
                                         B6=TERM MODE SHIFT ENB
0000:
               133 * OTHER SLOT VARIABLES AS DEFINED FOR PPC AND SIC MODES
0000:
0000:
               135 ****************
0000:
0000:
               136 * MONITOR SUBROUTINES *
0000:
               137 ****************
                                       ;CHARACTER OUT (THRU CSW)
;SETS KSW TO APPLE KEYBOARD
               138 COUT
                          EQU $FDED
FDED:
               139 SETKBD EQU
                                $FE89
FE89:
                                          ; KNOWN "RTS" LOCATION
                                SFF58
FF58:
               140 IORTS EQU
                                         ; INCREMENT A1H, L AND CMP TO A2H, L
                                SECBA
FCBA:
               141 NXTA1
                           EOU
               142 SETSCR EQU $FE93
143 VIDOUT EQU $FDF6
                                         ;SETS CSW TO APPLE SCREEN
FE93:
                                         ;OUTPUT A CHAR TO APPLE SCREEN
PDF6:
                           CHN
                                SSC.CN00
0000:
               144
00000:
:0000
                 3 * APPLE II SSC FIRMWARE
00000:
0000:
                 5 * BY LARRY KENYON
                 6 *
 0000:
                       -JANUARY 1981-
 0000:
 0000:
                 9 * (C) COPYRIGHT 1981 BY APPLE COMPUTER, INC. *
 0000:
                10 *
 0000:
 0000:
 00000:
                13 * CNOO SPACE CODE
 00000:
                14 *
 0000:
 0000:
 ---- NEXT OBJECT FILE NAME IS SSC.DCLS.OBJO
                           ORG $C700
                 17 *
 C700:
                18 BINIT BIT IORTS
19 BVS BENTRY
                                          ;SET THE V-FLAG
 C700:2C 58 FF
                                          ; <ALWAYS>
 C703:70 OC
                                           ; BASIC INPUT ENTRY
                 20 IENTRY SEC
 C705:38
                                          ;OPCODE FOR BCC
                           DFB $90
                 21
 C706:90
                                          ;BASIC OUTPUT ENTRY
                 22 OENTRY CLC
 C707:18
                           CLV
 C708: B8
                 23
                           BVC BENTRY ; < ALWAYS> SKIP AROUND PASCAL 1.1 ENTRY
```

24

C709:50 06

```
C70B:01
                            DFB $01
                                            ;GENERIC SIGNATURE BYTE
C70C:31
                            DFB $31
                                            ;DEVICE SIGNATURE BYTE
C70D:8E
                 27
                                 >PINIT
                                 >PREAD
C70E:94
                 28
                            DFB
C70F:97
                 29
                            DFB
                                 >PWRITE
C710:9A
                            DFB
                                 >PSTATUS
C711:85 27
                 31 BENTRY
                            STA
                                 CHARACTER
                                           ; INPUT BUFFER INDEX
C713:86 35
                 32
                            STX
                                 ZPTEMP
                                             ; SAVE X AND Y REGS ON STACK
C715:8A
                 33
                            TXA
C716:48
                 34
                            PHA
C717:98
                 35
                            TYA
C718:48
                 36
                            PHA
                                             ; SAVE ENTRY FLAGS
C719:08
                 37
                            PHP
C71A:78
                 38
                            SEI
                                             , NO RUPTS DURING SLOT DETERMINATION
C71B:8D FF CF
                 39
                            STA
                                 ROMSOFF
                                            SWITCH OUT OTHER $C800 ROMS
C71E: 20 58 FF
                 40
                            JSR
                                 IORTS
C721:BA
                 41
                            TSX
C722: BD 00 01
                 42
                            LDA
                                 STACK, X
                                           ; RECOVER SCN
C725:8D F8 07
                 43
                            STA
                                 MSLOT
                                             : X-REG WILL GENERALLY BE SCN
C728:AA
                 44
                            TAX
C729:0A
                 45
                            AST.
C72A:0A
                            ASL
                                            ;DETERMINE $NO
                 45
C72B: 0A
                 47
                            ASL
                                 A
C72C: 0A
                 48
                            ASL
C72D:85 26
                            STA
                                  SLOT16
                                             ; Y-REG WILL GENERALLY BE $NO
C72F: A8
                                             RESTORE RUPTS
C730:28
                 51
                            PI.P
C731:50 29
                 52
                            BVC
                                 NORMIO
                 53 *
C733:
C733:
                 54 * BASIC INITIALIZATION
C733:
                 55 *
C733:1E 38 05
                            ASL CMDBYTE, X : ALWAYS ENABLE COMMANDS
                 56
C736:5E 38 05
                 57
                            T.SR
                                 CMDBYTE, X
C739:B9 8A CO
                 58
                            LDA
                                 CMDREG, Y ; JUST HAD A POWER-ON OR PROGRAM RESET?
C73C:29 1F
                 59
                            AND
                                  #$1F
C73E: DO 05
                 60
                            BNE
                                 BINITI
                                            : IF SO. GO JOIN INIT IN PROGRESS
C740: A9 EF
                 61
                            LDA
                                 #SEF
                            JSR
C742:20 05 C8
                 62
                                 INITI
C745:
                 63 *
C745: E4 37
                 64 BINIT1 CPX
                                 CSWH
C747:D0 0B
                                  FROMIN
                            BNE
                 65
C749:A9 07
                            LDA
                                  #>OENTRY
C74B:C5 36
                 67
                            CMP
                                 CSWL
                                            ; IF CSW IS ALREADY POINTING TO CENTRY,
                                            ; THEN WE MUST HAVE COME FROM KSW
C74D:F0 05
                 68
                            BEO
                                 FROMIN
                                            ;OTHERWISE, SET CSW TO DENTRY
C74F:85 36
C751:18
                 69
                             STA
                                 CSWL
                                            ; INDICATE WE ARE CALLED FOR OUTPUT
                 70 FROMOUT CLC
C752:90 08
                 71
                                            : (ALWAYS)
                             BCC
                                  NORMIO
                                            ; MAKE SURE KSW POINTS HERE
                            CPX
C754: E4 39
                 72 FROMIN
                                  KSWH
                                  FROMOUT
C756:D0 F9
                 73
                            BNE
C758: A9 05
                 74
                            LDA
                                  #>IENTRY
C75A:85 38
                 75
                            STA KSWL
                                            ; SET UP KSW (NOTE CARRY SET FROM CPX)
                 76 *
C75C:
C75C:
                  77 * BRANCH TO APPROPRIATE BASIC I/O ROUTINE
 C75C:
 C75C:BD 38 07
                  79 NORMIO
                             LDA MISCFLG, X ; SEPARATE CIC MODE FROM OTHERS
                                            ; NOT ZERO FOR CIC MODE
C75F:29 02
                 80
                             AND #S02
                                            SAVE CIC MODE INDICATION
C761:08
                 81
                             PHP
C762:90 03
                 82
                             BCC
                                  BOUTPUT
```

```
83 JMP BINPUT
84 *
C764:4C BF C8 83
C767: BD B8 04 85 BOUTPUT LDA STATEFLG, X ; CHECK FOR AFTER LOWERCASE INPUT
C76A:48
                     86 PHA
87 ASL A
C76B: 0A
C76C:10 0E 88 BPL BOUTPUT1 ;SKIP IF NOT

C76E:A6 35 89 LDX ZPTEMP

C770:A5 27 90 LDA CHARACTER

C772:09 20 91 ORA #$20

C774:9D 00 02 92 STA INBUFF,X ;RESTORE LOWERCASE IN BUFFER

C777:85 27 93 STA CHARACTER ;AND FOR OUTPUT ECHO

C779:AE F8 07 94 LDX MSLOT
C76C:10 0E
                   88
                                 BPL BOUTPUT1 ; SKIP IF NOT
C779:AE F8 07
C77C:68
C77D:29 BF
                     95 BOUTPUT1 PLA
C77D:29 BF 96 AND #$BF ;ZERO THE FLAG
C77F:9D B8 04 97 STA STATEFLG,X
C782:28 98 PLP ;RETRIEVE CIC MODE INDICATION
C783:F0 06 99 BEQ BOUTPUT2 ;BRANCH FOR PPC, SIC MODES
C785:20 63 CB 100 JSR OUTPUT ;CIC MODE OUTPUT
C788:4C B5 C8 101 JMP CICEXIT ;FINISH BY CHECKING FOR TERM M
                                 JSR OUTPUT ;CIC MODE OUTPUT
JMP CICEXIT ;FINISH BY CHECKING FOR TERM MODE
C788:4C B5 C8 101
C78B:
                    102 *
 C78B:4C FC C8 103 BOUTPUT2 JMP SEROUT
C78E:
                    104 ********************
C78E:
                    105 *
C78E:
                    106 * NEW PASCAL INTERFACE ENTRIES *
C78E:
                    107 *
                    108 ********************
C78E:
C78E:20 00 C8 109 PINIT JSR PASCALINIT;
C791:A2 00 110 LDX #0 ;NO ERROR POSSIBLE
C791:A2 00 110 LDX #0 ;NC
C793:60 111 RTS
C794:4C 9B C8 112 PREAD JMP PASCALREAD ;
 C797:4C AA C9 113 PWRITE JMP PASCALWRITE;
 C79A:
                    114 *
 C79A:
                    115 * NEW PASCAL STATUS REQUEST
 C79A:
                    116 *
                    117 * A-REG=0 -> READY FOR OUTPUT?
 C79A:
                    118 * A-REG=1 -> HAS INPUT BEEN RECEIVED?
 C79A:
                    119 *
 C79A:
                                                      ; SAVE REQUEST TYPE IN CARRY
 C79A:4A
                    120 PSTATUS LSR A
C79A:4A 120 PSTATUS LSR A ;SAVE REQUEST TYPE C79B:20 9B C9 121 JSR PENTRY ;(PRESERVES CARRY) C79E:B0 08 122 BCS PSTATIN C7A0:20 F5 CA 123 JSR SROUT ;READY FOR OUTPUT? C7A3:F0 06 124 BEQ PSTATUS2 CCA5:18 125 CLC
 C7A6:90 03
                                 BCC PSTATUS2 ; CARRY CLEAR FOR NOT READY
                    126
                    127 *
 C7A8:
 C7A8:20 D2 CA 128 PSTATIN JSR SRIN ;SETS CARRY CORRECTLY
 C7AB:BD B8 05 129 PSTATUS2 LDA STSBYTE, X ;GET ERROR FLAGS
                    130 TAX
131 RTS
 C7AE: AA
 C7AF:60
                     132 *******************************
 C7B0:
                    133 * ROUTINE TO SEND A CHARACTER TO ANOTHER CARD *
 C7B0:
                    134 ********************************
 C7B0:
                    135 SENDCD LDX #3
 C7B0:A2 03
 C7B2:B5 36
                     136 SAVEHOOK LDA CSWL, X
 C7B4:48
                          PHA
DEX
 C7B5:CA
                    138
                            BPL SAVEHOOK
 C7B6:10 FA
                    139
```

C7B8:

140 \*

```
C7B8:
                           141 * NOW PUT CARD ADDRESS IN HOOK
            C7B8:
                           142 *
            C788:AE F8 07
                           143
                                       LDX MSLOT
            C7BB: BD 38 06 144
                                       LDA CHNBYTE, X
            C7BE:85 36
                           145
                                       STA CSWL
            C7C0:BD B8 04
                           146
                                       LDA STATEFLG, X ; GET SLOT #
            C7C3:29 38
                           147
                                       AND #$38
            C7C5:4A
                           148
                                       LSR A
            C7C6:4A
                           149
                                       LSR A
            C7C7:4A
                           150
                                       LSR A
            C7C8:09 C0
                           151
                                       ORA #$CO
                                                      ; FORM $CN
            C7CA:85 37
                           152
                                       STA CSWH
            C7CC:
                           153 *
            C7CC:
                           154 * OUTPUT TO THE PERIPHERAL
 HTS
                           155 *
            C7CC:8A
                           156
                                       TXA
                                                       ;SAVE $CN
            C7CD: 48
                           157
                                       PHA
 MIR
            C7CE: A5 27
                           158
                                       LDA CHARACTER
            C7D0:48
                           159
                                       PHA
            C7D1:09 80
                           160
                                       ORA #$80
                                                      :80 COL BOARDS WANT HI-BIT ON
 HES
            C7D3:20 ED FD
                           161
                                       JSR COUT
            C7D6:
                           162 *
            C7D6:
                           163 * NOW RESTORE EVERYTHING THE OTHER CARD MAY HAVE CLOBBERED
            C7D6:
                           164 *
            C7D6:68
                           165
            C7D7:85 27
                                       STA CHARACTER
                           166
 100
            C7D9:68
                           167
                                       PLA
            C7DA: 8D F8 07
                           168
                                       STA MSLOT
            C7DD: AA
                           169
                                       TAX
            C7DE: OA
                           170
                                       ASL A
            C7DF: OA
                                       ASL A
            C7E0:0A
                           172
                                       ASL A
 101
            C7E1:0A
                           173
                                       ASL A
            C7E2:85 26
                           174
                                       STA SLOT16
            C7E4:8D FF CF
                           175
                                       STA ROMSOFF
                           176 *
            C7E7:
            C7E7:
                           177 * PUT BACK CSWL INTO CHNBYTE
            C7E7:
                           178 *
100
            C7E7:A5 36
                           179
                                       LDA CSWL
            C7E9:9D 38 06
                           180
                                       STA CHNBYTE, X
            C7EC:
                           181 *
-
            C7EC: A2 00
                           182
                                      LDX #0
            C7EE:68
                           183 RESTORHOOK PLA
            C7EF:95 36
                           184
                                       STA CSWL, X
            C7F1:E8
                           185
                                       INX
            C7F2:E0 04
                           186
                                       CPX
            C7F4:90 F8
                           187
                                           RESTORHOOK
                                       BCC
            C7F6:
                           188 *
            C7F6: AE F8 07
                           189
                                       LDX MSLOT
            C7F9:60
                           190
                                       RTS
            C7FA:
                           191 *
            C7FA:C1 D0 D0
                           192
                                       ASC "APPLE"
            C7FD:CC C5
            C7FF:08
                           193
                                       DFB $8
            C800:
                           194 *
```

```
CHN SSC.C800
C800:
               196
                 1 ******************
C800:
C800:
C800:
                  3 * APPLE II SSC FIRMWARE
C800:
                 5 * BY LARRY KENYON
C800:
C800:
                 6 *
C800:
                       -JANUARY 1981-
C800:
C800:
                 9 * (C) COPYRIGHT 1981 BY APPLE COMPUTER, INC. *
C800:
                 10 *
                 11 ***************
C800:
C800:
                 12 *
C800:
                 13 * C800 SPACE: HIGH LEVEL STUFF *
C800:
                 14 +
C800:
                 15 ******************
C800:
                 16 * PASCAL 1.0 INIT ENTRY *
C800:
                 17 **************
---- NEXT OBJECT FILE NAME IS SSC.DCLS.OBJ1
                 18 ORG $C800
19 PASCALINIT JSR PENTRY ; PASCAL 1.0 INITIALIZATION ENTRY
C800:
C800:20 9B C9
C803:A9 16 20 LDA #$16 ;NO XOFF, ECHO, LF EAT, OR LF GEN
C805:48 21 INIT1 PHA ;GOES TO MISCFLG AFTER MODIFICATION
C806:A9 00 22 LDA #0
C808:9D 88 04 23 STA STATEFLG, X
C80B:9D B8 03 24 STA DELAYFLG, X
C80E:9D 38 04 25 STA HANDSHKE, X
C811:9D B8 05 26 STA STSBYTE, X
C814:9D 38 06 27 STA PWDBYTE, X
C817:9D B8 06 28
                           STA COLBYTE, X
                           LDA DIPSW2,Y ;SET LF GEN OPTION FROM D2-S5
STA ZPTMP2 ;SAVE FOR LATER
C81A: B9 82 C0
                29
C81D:85 2B
                 30
                           LSR A
LSR A
                                            ;S5-> CARRY
C81F:4A
                 31
                 32
                                           ; IF S5=ON=O THEN LEAVE MISCFLG ALONE
C820:4A
C821:90 04
                 33
                            BCC INITIA
                                            ;OTHERWISE, MAKE SURE LF GEN
; ENABLE IS RESET
C823:68
                            PLA
                 34
C824:29 FE
                            AND #SFE
C826:48
                            PHA
                 37 INIT1A CLV
                                            ; V WILL BE CLEAR FOR CIC MODE
CB27:B8
                            LDA DIPSW1, Y
C828:B9 81 C0
                38
                                           SIC MODES SET CARRY
                            LSR A
C82B:4A
                 39
C82C:B0 07
                            BCS
                                 INIT2
                                            BRANCH FOR SIC MODES
                 40
C82E:4A
                 41
                            LSR A
C82F:B0 OE
                            BCS INIT2B
                                            ; PPC MODE BRANCH
C831:A9 01
                43
                            LDA #$01
                                           ;CTL-A
                                         ; < ALWAYS> CIC MODE BRANCH
C833:D0 3D
                44
                            BNE INITS
                45 *
C835:
C835:4A
                 46 INIT2 LSR A
                                           ;SET CARRY FOR P8A
C836: A9 03
                            LDA
                                #$03
                                            ;SET ETX AS DEFAULT INQUIRY CHAR
C838:B0 02
                 48
                            BCS
                                 INIT2A
                                            ; BRANCH FOR P8A
C83A:A9 80
                 49
                            LDA #$80
                                            FOR P8 SET AUTO CR GEN
C83C:9D B8 04
                 50 INIT2A STA
                                 STATEFLG, X
                                         ;SET V-FLAG FOR PPC, SIC MODES
C83F:2C 58 FF
                 51 INIT28 BIT IORTS
C842: A5 2B
                 52
                            LDA
                                 ZPTMP2
                            AND #$20
C844:29 20
                 53
                                            ;SET CR DELAY
                                            ;SO 1=ENB, O=DISABLE
C846:49 20
                 54
                           EOR #$20
C848:9D B8 03
                           STA DELAYFLG, X ; FROM D2-S2
                 55
```

```
57 BVS INIT3 ;<ALWAYS> BRANCH AROUND PASCAL 58 **********************
            C84B:70 OA
            C84D:
            C84D:
                            59 * PASCAL 1.0 READ ENTRY *
            C84D:
                            60 * (MUST BE AT $C84D)
            C84D:
                            61 **************
            C84D: 20 9B C8
                            62 PREADO JSR PASCALREAD ;DO PASCAL 1.1 READ
            C850: AE F8 07
                            63
                                      LDX MSLOT ; MODIFY FOR 1.0
            C853:9D B8 05
                                      STA STSBYTE, X ; CHARACTER READ
                            64
- 46
            C856:60
                            65
                                      RTS
            C857:
                            66 *************
            C857:
                            67 * NOW WHERE WERE WE??? *
            C857:
                            68 ************
            C857:
                            69 *
            C857: A5 2B
                                                    ; PPC, SIC MODES USE SWITCHES ; TO SET PWIDTH, CR DELAY
                            70 INIT3 LDA ZPTMP2
100
            C859:4A
                                      LSR
           C85A:4A
                            72
                                      LSR A
           C85B: 29 03
                            73
                                      AND
                                           #$03
200
           C85D: A8
                            74
                                      TAY
            C85E:F0 04
                            75
                                      BEQ INIT4
            C860:
                            76 *
            C860:68
                                      PLA
                                                    ; RESET VIDEO ENABLE FOR PWIDTH#40
           C861:29 7F
                                      AND #$7F
            C863:48
                            79
                                      PHA
            C864:
                            80 *
            C864: B9 A6 C9
                            81 INIT4
                                      LDA PWDTBL, Y
           C867:9D 38 06
                            82
                                      STA
                                           PWDBYTE, X
           C86A:A4 26
                            83
                                      LDY
                                           SLOT16
           C86C:
           C86C:68
                                                     ;CLEAR CIC BIT IN FUTURE MISCFLG
                            85
           C86D: 29 95
                                      AND
                                           #$95
                                                    ; (AND TABBING, XOFF AND LF EAT BITS)
           C86F:48
                            87
                                      PHA
           C870:A9 09
                            88
                                      LDA
                                           #$09
                                                     ;CTL-I
                            89 *
           C872:
           C872:9D 38 05
                           90 INITS
                                      STA CMDBYTE, X ; CMD ESC CHAR (IGNORED FOR SIC MODES)
           C875:68
                           91
                                      PLA
           C876:9D 38 07
                           92
                                      STA MISCFLG, X ; SET MISCFLG FLAGS
           C879:
                            93 *
           C879:
                            94 * NOW FOR THE ACIA INITIALIZATION ROUTINE
           C879:
                           95 *
           C879:A5 2B
                           96 INITACIA LDA ZPTMP2
                                                     ;DIPSW2
           C87B:48
                           97
                                      PHA
           C87C:29 AO
                           98
                                      AND #SAO
                                                     ;DATA BIT OPTIONS FOR CIC MODE
           C87E:50 02
                           99
                                           INITACIA1 ; BRANCH FOR CIC MODE
                                      BVC
           C880: 29 80
                         100 AND #$80 ;8 DATA, 1 OR 2 S
101 INITACIA1 JSR DATACMD1 ;SET CONTROL REG
                                                    ;8 DATA, 1 OR 2 STOP FOR SIC, PPC
           C882:20 A1 CD
           C885:20 81 CD
                          102
                                     JSR BAUDCMD1 ;SET DIPSWITCH BAUD RATE
           C888:68
                           103
                                      PLA
           C889:29 OC
                           104
                                      AND #$OC
                                                     ; PARITY OPTIONS FOR CIC MODE
           C88B:50 02
                          105
                                      BVC INITACIA2 ; BRANCH FOR CIC MODE
           C88D: A9 00
                          106
                                      LDA #$0
                                                     ; DISABLE PARITY FOR SIC, PPC MODES
           C88F: 0A
                          107 INITACIA2 ASL A
           C890:0A
                          108
                                      ASL A
           C891:0A
                          109
                                      ASL
           C892:09 0B
                          110
                                      ORA #SOB
           C894:99 8A CO
                          111
                                      STA CMDREG, Y
-18
           C897: 89 88 CO
                          112
                                      LDA RDREG, Y ; THROW OUT THE STRANGE STUFF
           C89A:60
                          113
                                     RTS
           C89B:
                          114 ****************
```

```
C89B:
                      115 * PASCAL READ ROUTINE *
                      116 ***************
C89B:
C89B:20 9B C9 117 PASCALREAD JSR PENTRY ; SHARED BY BOTH PASCAL VERSIONS C89E:20 AA C8 118 PASCALREAD1 JSR GETCHAR ; GET ACIA/KBD DATA
                      119 AND #$7F
120 PASEXIT LDY MSLOT
C8A1:29 7F
C8A3:AC F8 07
                                                           ;CLEAR HIGH BIT FOR PASCAL
C8A6:BE B8 05 121 LDX STSBYTE,Y ;ERROR STATUS-> X-REG C8A9:60 122 RTS
CSAA:
                       123 *******************
                     123 ** GETCHAR ROUTINE WAITS FOR **
125 * THE NEXT CHAR FROM EITHER **
126 * THE ACIA OR KEYBOARD (IF **
127 * ENABLED). USED BY PASCAL **
128 * READ ROUTINE, XON WAIT, **
129 * AND ACK WAIT. DATA IS RE- **
130 * TURNED IN THE A-REGISTER **
CBAA:
CBAA:
CBAA:
CBAA:
CBAA:
CBAA:
CBAA:
                       131 ********************
CSAA:
C8AA: 20 FF CA 132 GETCHAR JSR INPUT ,ACIA DATA?
C8AD: 80 05 133 BCS GETCHAR1
C8AF: 20 2C CC 134 JSR CKKBD ;KEYBOARD INCRES: 20 CC 135 BCC GETCHAR
                                                              ;KEYBOARD INPUT?
C8B2:90 F6
C8B4:60
                      136 GETCHAR1 RTS
                                                             ; EXIT WHEN WE HAVE SOMETHING
C8B5:
                       137 *
C8B5:
                      138
                                        CHN SSC.HILEV
```

```
C8B5:
                2 ****************
C8B5:
C8B5:
                4 * APPLE II SSC FIRMWARE
C8B5:
C885:
                6 * BY LARRY KENYON
C8B5:
C8B5:
                8 *
                       -FEBRUARY 1981-
CRB5:
                9 *
C8B5:
                10 * (C) COPYRIGHT 1981 BY APPLE COMPUTER, INC. *
C8B5:
                12 *****************
C8B5:
                13 *
C8B5:
C8B5:
                14 * CIC, SIC, PPC MODE HIGH-LEVEL *
C8B5:
                15 *
C8B5:
                17 * CIC EXIT ROUTINE . . .
C8B5:
                18 *******************
C8B5:
C8B5: 20 1E CA
               19 CICEXIT JSR CHECKTERM ; SEE IF WE'VE ENTERED TERMINAL MODE
                20 *************
C8B8:
C8B8:
                21 * BASIC EXIT ROUTINE * 22 ***************
C8B8:
C8B8:68
                23 BASICEXIT PLA
                        TAY
C8B9: A8
                24
C8BA: 68
                25
                          PLA
C8BB: AA
                26
                          TAX
C8BC: A5 27
                27
                          LDA CHARACTER
C8BE: 60
                          RTS
                28
                29 *************
C8BF:
C8BF:
                30 * BASIC INPUT ROUTINE *
CSBF:
                31 **************
C8BF:F0 29
                32 BINPUT BEQ BINACIA ; BRANCH IF NOT CIC MODE
C8C1:BD B8 06
                           LDA BUFBYTE, X ; INPUT BUFFER PULL?
                33
C8C4:10 05
                           BPL BINKBD
                34
C8C6:5E B8 06
                35
                          LSR BUFBYTE, X ; RESET BUFFER FULL
C8C9:D0 24
                36
                          BNE BINACIA1 ; <ALWAYS>
                37 *
C8CB:
C8CB: 20 3E CC
                38 BINKBD JSR GETKBD
                                         ; KEYBOARD DATA?
C8CE:90 1A
                39
40 *
                          BCC BINACIA
C8D0:
C8D0:BD B8 03
                41 BINEND LDA
                               DELAYFLG, X
                               #SCO ;TRANSLATE LOWERCASE TO UPPERCASE?
BINEND1 ;IF SO, LET THE MONITOR DO IT
C8D3:29 C0
                42
                           AND
C8D5:F0 OE
                43
                           BEQ
C8D7: A5 27
                44
                           LDA
                                CHARACTER ; IF NOT, SET FLAG IF
                               #$E0 ; THIS IS A LOWERCASE CHAR
BINEND1 ; FOR INPUT BUFFER CORRECTION
C8D9: C9 E0
                45
                           CMP
C8DB:90 08
                46
                           BCC
C8DD: BD B8 04
                          LDA STATEFLG, X ; (CIRCUMVENT APPLE MONITOR)
                47
C8E0:09 40
                48
                           ORA #$40
C8E2:9D B8 04
                          STA STATEFLG. X
                49
CBE5:
                50 *
C8E5:28
                51 BINEND1 PLP
C8E6:F0 D0
                           BEQ BASICEXIT ; BRANCH IF NOT CIC MODE
C8E8: DO CB
                53
                           BNE
                               CICEXIT
                                         ; <ALWAYS> CHECK TO SEE IF WE
CSEA:
                54 *
                                 ENTERED TERM MODE (VIA KYBD ESCAPE
C8EA: 20 FF CA
                55 BINACIA JSR
                                INPUT
                                         ;ACIA DATA?
C8ED: 90 DC
                56
                           BCC BINKBD
C8EF: 20 11 CC
                57 BINACIA1 JSR RESTORE ; DO BASIC CURSED DUTY
C8F2:28
                58
                          PLP
C8F3:08
                59
                           PHP
                                         GET CIC MODE INDICATOR
```

```
C8F4:F0 DA
                60
                          BEQ BINEND
                                          ;SKIP IF NOT CIC MODE
C8F6: 20 D1 C9
                61
                           JSR CKINPUT ; LOOK FOR INPUT STREAM SPECIAL CHARS
C8F9:4C D0 C8
                62
                           JMP
                                BINEND
C8FC:
                63 ******************
C8FC:
                64 * SIC, PPC BASIC OUTPUT ROUTINE *
CSFC:
                65 *****************
C8FC: 20 1A CB
                66 SEROUT JSR CMDSEQCK ; CHECK FOR A COMMAND SEQUENCE
C8FF:BO B7
                67
                           BCS BASICEXIT ; BRANCH IF WE WERE IN COMMAND MODE
C901:A5 27
                                CHARACTER ; SAVE CHAR ON STACK
                68
                           LDA
C903:48
                69
                           PHA
C904:BD 38 07
                70
                          LDA
                                MISCFLG, X ; IF VIDEO OR TABBING ENABLED,
C907:29 CO
                71
                           AND
                                #SCO
                                          ; DON'T MESS WITH THE CURSOR
C909:D0 16
                72
                           BNE
                                TABCHECK
C90B:
C90B:A5 24
                           LDA
                                CH
                                          ;CHECK FOR COMMA TABBING
;IF CH=O, THERE WAS NO TAB OR COMMA
C90D: FO 42
                75
                           BEQ
                                NOTAB
C90F: C9 08
                76
                           CMP
                                #8
                                          ;INTEGER BASIC COMMA?
C911:F0 04
                77
                           BEQ
                                COMMA
C913:C9 10
                78
                           CMP
                                #16
                                         ; APPLESOFT COMMA?
C915:D0 0A
                79
                           BNE
                                TABCHECK
C917:09 FO
                80 COMMA
                           ORA
                                #SFO
C919:3D B8 06
                81
                           AND
                                COLBYTE, X ; SET COL TO PREVIOUS TAB
C91C:18
                82
                           CLC
C91D:65 24
                           ADC CH
                83
                                        ; THEN INCREMENT TO NEXT TAB
C91F:85 24
                84
                           STA CH
C921:
                85 *
C921:
                86 *
C921:BD B8 06
                87 TABCHECK LDA COLBYTE, X
C924:C5 24
                88
                           CMP
                                CH
                                         ; IS TABBING NEEDED?
C926:F0 29
                89
                           BEQ NOTAB
                                          ; IF EQUAL THEN NO TAB NEEDED
C928: A9 A0
                90
                           LDA
                               #SAO
                                          ; SPACE FOR FORWARD TAB
C92A:90 08
                91
                           BCC
                               TAB1
C92C:BD 38 07
                92
                           LDA
                                MISCFLG, X ; DON'T BACKSPACE UNLESS TABBING
C92F: 0A
                93
                           ASL
                               A
                                         ; OPTION IS ENABLED
C930:10 1F
                94
                           BPL NOTAB
C932:A9 88
                95
                          LDA #$88
                                          ; BACKSPACE FOR BACKTAB
C934:85 27
                96 TAB1
                           STA
                               CHARACTER
C936:2C 58 FF
                97
                           BIT IORTS
                                          ;SET V=1 TO INDICATE TABBING
C939:08
                98
                           PHP
                                          ; SAVE TABBING INDICATOR
C93A:70 OC
                99
                           BVS TAB2
                                          ; <ALWAYS> AROUND BATCH MOVE ENTRY
C93C: EA
               100
                           NOP
C93D:
               101 ***************
               102 * SHORT BATCH MOVE:
C93D:
               103 * LOCATE AT $C93D FOR
104 * COMPATIBILITY WITH
105 * SIC P8 BLOCK MOVE.
C93D:
C93D:
C93D:
C93D:
               106 ***************
C93D: 2C 58 FF
               107 BATCHIN BIT IORTS
C940:50
               108
                          DFB $50
                                          ; DUMMY BVC
C941:B8
               109 BATCHOUT CLV
                                          ; V=O FOR OUTPUT ENTRY
C942:AE F8 07
               110 LDX MSLOT
C945:4C EF C9
                          JMP BATCHIO
C948:
               112 *************
               113 * BURP . . .
C948:
C948:
               114 *************
C948:20 B5 C9 115 TAB2 JSR ADJUST
                                          ; ADJUST COLUMN COUNT
C94B: 20 6B CB 116
                          JSR OUTPUT?
                                          ;DON'T GO TO SCREEN WHEN TABBING
```

JMP FORCECR

; SHARE SOME CODE. . .

C94E:4C 68 C9 117

```
C951:
                            118 *
             C951:68
                            119 NOTAB
                                        PLA
             C952:88
                            120
                                        CLV
             C953:08
                            121
                                        PHP
                                                        ;SAVE 'NO TAB' INDICATION
             C954:85 27
                            122 NOTAB1
                                        STA CHARACTER ; (FORCE CR REENTRY)
             C956:48
                            123
                                        PHA
             C957: 20 68 CB 124
                                        JSR OUTPUT1
                                                        ; ENTER AFTER CMD SEQ CHECK
             C95A:20 B5 C9
                            125
                                        JSR
                                             ADJUST
             C95D:68
                            126
                                        PLA
             C95E:49 8D
                            127
                                        EOR
                                             #$8D
                                                        ; WAS IT A CR?
             C960:0A
                            128
                                        ASL.
             C961:D0 05
                            129
                                        BNE FORCECE
             C963:9D B8 06
                            130
                                        STA COLBYTE, X ; IF SO, RESET COLUMN TO 0
             C966:85 24
                            131
                                        STA
             C968:
                            132 *
             C968: BD B8 04
                            133 FORCECR LOA
                                             STATEFLG, X ; FORCE CR DISABLED?
             C96B:10 0D
                                        BPL SEREND
                            1.34
             C96D: BD 38 06
                            135
                                             PWDBYTE, X ; FORCE CR IF LIMIT REACHED
                                        LDA
            C970:F0 08
                            136
                                        BEQ
                                                       ; (FOR P8 POKE COMPATIBILITY)
                                             SEREND
FPE
            C972:18
C973:FD B8 06
                                        CLC
                            138
                                        SBC
                                             COLBYTE, X
            C976: A9 8D
                            139
                                        LDA
                                             #$8D
             C978:90 DA
                            140
                                        BCC
                                             NOTAB1
                                                       ; BRANCH TO FORCE CR
             C97A:
            C97A:28
                            142 SEREND
            C97B:70 A4
                            143
                                        BVS TABCHECK ; BRANCH IF TABBING
            C97D:
                            144 *
            C97D: BD 38 07
                            145
                                        T-DA
                                             SEREND2 ; WHEN VIDEO IS ON COLBYTE, X
                                             MISCFLG, X ; DON'T MESS WITH CURSOR
            C980:30 16
C982:BC B8 06
                            146
                                        BMI
                            147
                                        LDY
            C985: 0A
                            148
                                        ASI.
            C986: 30 OE
                            149
                                        BMI
                                            SETCH
                                                       ;SET CH TO VALUE OF COL FOR TABBING
            C988:98
                            150
                                        TYA
                                        LDY
            C989: AO OO
                            151
             C98B:38
                            152
            C98C:FD 38 06
                            153
                                        SBC
                                             PWDBYTE, X ;
            C98F:C9 F8
                            154
                                        CMP
                                             #$FB
                                                       ; WITHIN 8 CHARS OF PWIDTH?
            C991:90 03
                            155
                                        BCC
                                             SETCH
            C993:69 27
                            156
                                        ADC
                                             #$27
                                                       ; IF SO, ADJUST TO WITHIN 8 OF 40
- 186
            C995: A8
                            157
                                        TAY
            C996:84 24
                            158 SETCH
                                       STY CH
            C998:
                            159 *
            C998:4C B8 C8
                            160 SEREND2 JMP BASICEXIT ; THAT'S ALL
            C998:
                            161 *
            C99B:
                            162 ****************
            C99B:
                            163 * PASCAL ENTRY ROUTINE
            C99B:
                            164 **************
            C99B:8E F8 07
                            165 PENTRY STX MSLOT
            C99E:84 26
                            166
                                        STY
                                            SLOT16
            C9A0: A9 00
                            167
                                        LDA #0
            C9A2:9D B8 05
                            168
                                        STA STSBYTE, X
            C9A5:60
                            169
                                       RTS
            C9A6:
                            170 *
            C9A6:
            C9A6:
                            172 * SIC MODE PRINTER WIDTH TABLE *
            C9A6:
                            173 ***********************
            C9A6: 29
                            174 PWDTBL DFB $29 ;40 COLUMNS
            C9A7:48
                            175
                                       DFB $48
                                                       ;72 COLUMNS
```

```
176 DFB $50 ;80 COLUMNS
C9A8:50
                                        ;132 COLUMNS
C9A9:84
                          DFB $84
              177
               178 *****************
C9AA:
               179 * PASCAL WRITE ROUTINE *
C9AA:
              180 * (DOUBLES AS PASCAL *
C9AA:
              181 * 1.0 ENTRY POINTY *
182 * -MUST BE AT $C9AA- *
C9AA:
C9AA:
              183 **************
C9AA:
              184 PASCALWRITE STA CHARACTER
C9AA:85 27
C9AC:20 9B C9 185 JSR PENTRY
                         JSR OUTPUT
C9AF: 20 63 CB 186
                         JMP PASEXIT ; LOAD X-REG WITH ERROR BYTE & RTS
              187
C982:4C A3 C8
              188 *
C985:
               189 *****************
C9B5:
               190 * COLUMN ADJUST ROUTINE *
C9B5:
              C9B5:
C9B5:
              193 ADJUST LDA CHARACTER
C9B5: A5 27
                               #$08 ;BACKSPACE?
                         EOR
              194
C9B7:49 08
                          ASL A
              195
C9B9:0A
                         BEQ DECRCOL ; IF SO, DECREMENT COLUMN
               196 BEQ DECRCOL ; IF SO, DECREMENT CO
197 BOR #SEE ;DELETE? (SFF, RUB)
198 BNE CTRLTST
C9BA:F0 04
C9BC:49 EE
C9BE:D0 09
C9C0:DE B8 06 199 DECRCOL DEC COLBYTE, X ; DECREMENT COLUMN COUNT
               200 BPL ADJRTS
C9C3:10 03
                          STA COLBYTE, X ; DON'T ALLOW TO GO BELOW O
C9C5:9D B8 06 201
               202 ADJRTS RTS
203 CTRLTST CMP #SCO ;DON'T INCREMENT COLUMN COUNT FOR
204 BCS ADJRTS ; CONTROL CHARACTERS
205 INC COLBYTE, X
 C9C8:60
 C9C9:C9 C0
 C9CB: BO FB
 C9CD:FE B8 06 205
                         RTS
               206
 C9D0:60
               207 *********************
 C9D1:
               208 * ROUTINE TO PROCESS SPECIAL INPUT CHARS *
 C9D1:
               209 *********************
 C9D1:
 C9D1:BD 38 07 210 CKINPUT LDA MISCFLG,X
                                         ; INPUT CTL CHARS ENABLED?
                         AND #508
BEQ CIEND
 C9D4:29 08
               211
               212
 C9D6:F0 16
                213 *
 C9D8:
                           LDA STATEFLG, X
 C9D8:BD B8 04 214
                           LDY CHARACTER
                215
 C9DB: A4 27
                                         ;CTL-T?
                           CPY #$94
                216
 C9DD: C0 94
                           BNE CKINPUT1
                217
 C9DF:D0 04
                                         ;SET TERMINAL MODE
                           ORA #$80
                218
 C9E1:09 80
                           BNE CKINPUTZ ; < ALWAYS>
                219
 C9E3:D0 06
                220 *
 C9E5:
                                         ; CONTROL-R?
                221 CKINPUT1 CPY #$92
  C9E5:C0 92
                222 BNE CIEND
223 AND #$7F
  C9E7:D0 05
                                         RESET TERMINAL MODE
  C9E9: 29 7F
                224 CKINPUT2 STA STATEFLG, X
  C9EB:9D B8 04
                225 CIEND RTS
  C9EE: 60
                226 *
```

C9EF:

```
CHN SSC. TERM
           C9EF:
                            1 *********************
           C9EF:
           C9EF:
                           3 * APPLE II SSC FIRMWARE
           C9EF:
                           4 *
                           5 * BY LARRY KENYON
           C9EF:
           C9EF:
                           6 *
            C9EF:
                                  -APRIL 1981-
           C9EF:
           C9EF:
                           9 * (C) COPYRIGHT 1981 BY APPLE COMPUTER, INC. *
                          10 *
           C9EF:
                          11 ****************************
           C9EF:
           C9EF:
                          12 * SHORT BLOCK MOVE *
           C9EF±
                           13 **************
120
            C9EF:8A
                          14 BATCHIO TXA
           C9F0:0A
                                   ASL A
                          15
            C9F1:0A
                                    ASL A
                          16
           C9F2:0A
                                    ASL A
           C9F3:0A
                                    ASL A
           C9P4:85 26
                                    STA SLOT16
                          19
mile
           C9F6:A9 00
                           20
                                    LDA #0
           C9F8:9D B8 05
                          21
                                    STA STSBYTE, X ; ZERO ERROR INDICATION
           C9FB:70 OF
                           22
                                    BVS MOVIN
                          23 *
           C9FD:
           C9FD: A0 00
                          24 MOVOUT LDY #0
           C9PF:B1 3C
                                    LDA (A1L), Y ;GET BUFFER DATA
STA CHARACTER
                          25
            CA01:85 27
                          26
            CA03:20 02 CC
                                    JSR ACIAOUT ; SEND IT OUT THE ACIA
           CA06:20 BA FC
                          28
                                    JSR NXTA1
           CA09:90 F2
                          29
                                    BCC MOVOUT
           CAOB: 60
                                    RTS
           CAOC:
           CAOC: 20 D2 CA
                          32 MOVIN
                                    JSR SRIN
                           33
           CAOF: 90 FB
                                     BCC
                                         MOVIN
           CA11: B9 88 CO
                          34
                                    LDA
                                         RDREG, Y
           CA14: AO 00
                           35
                                    LDY
                                         #0
                                    STA (A1L), Y ; PUT ACIA DATA INTO BUFFER
           CA16:91 3C
                          36
           CA18:20 BA FC
                          37
                                    JSR NXTA1
           CA1B:90 EF
                          38
                                    BCC MOVIN
           CA1D: 60
39
                                    RTS
           CA1E:
                          40 *
           CA1E:
                          41 *********************
           CA1E:
           CA1E:
                          43 * TERMINAL MODE ROUTINES
           CA1E:
                          44 *
                          45 *******************
           CA1E:
           CA1E: BD B8 04
                          46 CHECKTERM LDA STATEFLG, X ; HAVE WE ENTERED TERMINAL MODE?
           CA21:10 31
                          47
                                    BPL TERMRTS ; IF NOT, A SIMPLE RTS WILL DO. . .
                          48 *
           CA23:
           CA23:
                          49 * WE ENTER THE WORLD OF TERMINAL MODE
           CA23:
                          50 *
           CA23:A9 02
                          51 TERMMODE LDA #S02
                                                  ;START IN SHIFT-LOCK STATE
                                   PHA
LDA #$7F
           CA25:48
                          52
                                                  ; SHIFT STATE IS SAVED ON STACK
           CA26: A9 7F
                          53
           CA28:20 E2 CD
                                                ; RESET ECHO (DEFAULT TO FULL DUP)
                          54
                                  JSR KCMD1
                          55 *
           CA2B:
           CA2B: A4 24
                          56 TERMNEXT LDY CH
           CA2D: B1 28
                                    LDA (BASL), Y
```

```
STA CHARACTER ; SAVE SCREEN CHARACTER
CA2F:85 27
                59 TERMNEXT1 LDA #507 ; IMPLEMENT A FLASHING UNDERLINE
CA31:A9 07
                                          ; FOR A CURSOR
CA33:25 4F
                60
                           AND RNDH
                           BNE TERMNEXT3
CA35:D0 10
                61
CA37: A4 24
                62
                           LDY CH
CA39: A9 DF
                63
                           LDA #SDF
                           CMP (BASL), Y ; IS UNDERLINE ON THE SCREEN?
BNE TERMNEXT2 ; IF NOT, PUT IT THERE
CA3B:D1 28
                64
CA3D: DO 02
                65
                           LDA CHARACTER ; OTHERWISE USE TRUE SCREEN CHAR
CA3F: A5 27
                66
                67 TERMNEXT2 STA (BASL), Y
CA41:91 28
                                         ; MAKE IT FLASH, BUT
                          INC RNDH
                68
CA43: E6 4F
                           INC RNDH
                                           NOT TOO SLOW AND NOT TOO FAST
CA45: E6 4F
                69
                70 *
CA47:
                71 TERMNEXT3 LDA STATEFLG, X ; ARE WE STILL IN TERM MODE?
CA47: BD B8 04
                         BMI TERMACIAIN ; IF SO, GO CHECK ACIA
CA4A:30 09
CA4C:
                                           ; ALWAYS REPLACE OUR CURSOR
                 74 TERMEXIT JSR RESTORE
CA4C:20 11 CC
                      PLA
                                           CLEAN UP THE STACK
CA4F:68
                                           ; RETURN A <CR> TO COVER UP
CA50: A9 8D
                           LDA #$8D
                76
CA52:85 27
                            STA
                                CHARACTER
                 78 TERMRTS RTS
CA54:60
CA55:
                 79 *
                                          ;ACIA INPUT?
CA55: 20 FF CA
                80 TERMACIAIN JSR INPUT
                           BCC TERMKBDIN ; IF NOT, GO CHECK KEYBOARD
CA58:90 OC
                81
                           JSR RESTORE ;RESTORE CURSOR, INPUT->CHARACTER
JSR CKINPUT ;CHECK FOR CTL-T, CTL-R
CA5A: 20 11 CC
                82
CA5D: 20 D1 C9
                83
                           JSR SCREENOUT1 ; INPUT->SCREEN ALWAYS
CA60: 20 A3 CC
                84
                           JMP TERMNEXT ;
CA63:4C 2B CA
                 85
                 86 *
CA66:
CA66: 20 3E CC
                 87 TERMKBDIN JSR GETKBD ; KEYPRESS?
CA69:90 C6
                 88
                           BCC TERMNEXT1 ; SKIP IF NOT
                            BVS TERMNEXT ; BRANCH IF WE DID A KBD ESCAPE SEQ.
CA6B: 70 BE
                 89
                            LDA
                                 MISCFLG, X ; SHIFTING ENABLED?
CA6D: BD 38 07
                 90
CA70:0A
                            ASL A
                 91
CA71:10 22
                 92
                            BPL TERMSEND1
                                           ; RECOVER TERMSTATE
CA73:68
                 93
                            PLA
CA74: A8
                 94
                            TAY
                            LDA CHARACTER
CA75: A5 27
                 95
                                         ;1 = SHIFT LETTERS, XLATE NUMBERS
                            CPY #1
CA77:CO 01
                 96
                            BEQ TERMCAP
CA79:F0 20
                 97
                                 TERMLOCK ; 2 MEANS CAPS LOCK MODE
CA7B:B0 34
                 98
CA7D:
                 99 *
                100 TERMNORM CMP #$9B
CA7D: C9 98
                                           ; ESC?
                101
                           BNE TERMLETTER
 CA7F:D0 06
                102 *
 CA81:
                                           :INCREMENT STATE
 CA81:C8
                103 TERMING INY
 CA82:98
                104 TERMINC1 TYA
                                           ; PUT BACK ON STACK
 CA83:48
                105
                            PHA
                            JMP TERMNEXT
 CA84:4C 2B CA
                106
                107 *
 CA87:
 CA87: C9 C1
                108 TERMLETTER CMP #$C1
                                           1<A?
                            BCC TERMSEND
 CA89:90 08
                109
                                           1>27
 CASB: C9 DB
                110
                            CMP #SDB
                            BCS TERMSEND
 CA8D: BO 04
                111
                                           ; IT'S A LETTER SO TRANSLATE TO LC
                            ORA #$20
 CASF: 09 20
                112
 CA91:85 27
                113
                            STA CHARACTER
                114 *
 CA93:
```

CA93:98

115 TERMSEND TYA

```
PUT STATE BACK ON STACK
CA94:48
               116
                          PHA
CA95:20 68 CB 117 TERMSEND1 JSR OUTPUT1
                                         GO OUTPUT
CA98: 4C 2B CA 118
CA98: 119 *
                          JMP TERMNEXT
CA9B: C9 9B
                                          TWO ESCAPES?
               120 TERMCAP CMP
                               #$9B
CA9D: FD E2
               121
                          BEQ
                               TERMINC
CA9F:C9 BO
               122
                          CMP
                               #$B0
                                          ;<0?
CAA1:90 OA
                          BCC
                               TERMCAP1
               123
                                          ;>COLON?
CAA3:C9 BB
                          CMP #$BB
               124
CAA5: BO 06
               125
                          BCS TERMCAP1
CAA7:
               126 *
CAA7:
               127 * ESC (NUMBER) SO TRANSLATE INTO MISSING ASCII CHAR
CAA7:
               128 *
CAA7: A8
CAA8: B9 09 CA
                           LDA TRANSLATE-$BO, Y
               130
                           STA CHARACTER
CAAB:85 27
CAAD:A0 00
               131 STA CHAR.
132 TERMCAP1 LDY #0
                          P1 LDY #0 ;BACK TO STATE 0
BEQ TERMSEND ;<ALWAYS>
CAAF: FO E2
               133
CAB1:
               134 *
CAB1: C9 9B
               135 TERMLOCK CMP #$9B
                                         ; ESC?
CAB3:DO DE
                          BNE TERMSEND
CAB5: A0 00
               137
                           LDY #0
CAB7: FO C9
               138
                           BEQ TERMINC1; <ALWAYS>
CAB9:
               139 *
               140 ***************
CAR9:
               141 * TRANSLATE TABLE
CAB9:
               142 *****************
CAB9:
               143 TRANSLATE DFB $9B
CAB9:9B
                                         ; ESC
                          DFB $9C
CABA:9C
                                          ;FS
               144
CABB: 9F
                           DFB $9F
               145
                                          ;US
CABC:DB
                           DFB $DB
                                          LEFT BRACKET
               146
CABD: DC
               147
                           DFB $DC
                                          ;LEFT SLASH
CABE: DF
               148
                           DFB $DF
                                          ; UNDERSCORE
CABF: FB
               149
                           DFB SFB
                                          ;LEFT ENCLOSE
CACO: FC
                           DFB SFC
                                          ; VERTICAL BAR
               150
                                          ; RIGHT ENCLOSE
CAC1:FD
                           DFB $FD
               151
CAC2: FE
               152
                           DFB SFE
                                          ; TILDE
CAC3: FF
               153
                           DFB $FF
                                          ; RUB
CAC4:
               154 *
CAC4:
                           CHN SSC.CORE
               155
```

and PE

```
2 **********************
CAC4:
CAC4:
                4 * APPLE II SSC FIRMWARE
CAC4:
                6 * BY LARRY KENYON 7 *
CAC4:
CAC4:
CAC4:
CAC4:
                8 *
                      -JANUARY 1981-
CAC4:
CAC4:
               10 * (C) COPYRIGHT 1981 BY APPLE COMPUTER, INC. *
CAC4:
                11 *
CAC4:
                12 *******************************
CAC4:
                13 *
CAC4:
                14 * CORE SUBROUTINES
CAC4:
               15 .
                16 ********************
CAC4:
               17 *********************
CAC4:
                18 * GENERAL PURPOSE WAIT ROUTINE *
CAC4:
CAC4:
                19 ********************
                20 *
CAC4:
                21 * WAITMS WAITS FOR [A-REG] MILLISECONDS (256 IF A-REG=0)
CAC4:
CAC4:
               24 WAITMS1 DEX ;<DON'T LET THIS LOOP CROSS A PAGE>
25 BNE WAITMS1 ;5 MICROSECOND LOOP
26 SEC
27 SBC #01
28 BNE WAITMS
29 LDX MSLOT
30 RTS
                22 *
CAC4: A2 CA
CAC6:CA
CAC7:DO FD
CAC9:38
CACA: E9 01
CACC:DO F6
CACE: AE F8 07
CAD1:60
                31 *********************
CAD2:
                32 * ACIA STATUS REGISTER READ ROUTINES *
CAD2:
                33 ***************************
CAD2:
                34 *
CAD2:
                35 * SRIN USED TO CHECK ACIA INPUT STATUS
CAD2:
                36 *
CAD2:
                37 SRIN LDY SLOT16
CAD2: A4 26
                                        ;SLOT16=$NO
                        LDA STREG,Y
PHA
AND #$20
LSR A
LSR A
CAD4: B9 89 CO
                38
CAD7:48
                39
                                        ;DCD?
;AN ERROR IF NOT
CAD8: 29 20
CADA: 4A
                41
CADB: 4A
                42
                         STA ZPTEMP
PLA
CADC:85 35
               43
CADE: 68
                44
CADF: 29 OF
                          AND #SOF
                45
                                        ;SET CARRY IF RDR FULL, ELSE CLEAR
                          CMP #$08
CAE1: C9 08
               46
               47 BCC SRIN1
48 AND #$07 ;PE, FE, OVR VALID ONLY WHEN RDR=1
49 BCS SRIN2 ;<ALWAYS>
50 SRIN1 LDA ZPTEMP
CAE3:90 04
CAE5: 29 07
CAE7: BO 02
CAE9: A5 35
                                        ;GET DCD ERROR BIT
;BRANCH IF NO ERRORS FOUND
                51 SRIN2 ORA ZPTEMP
CAEB: 05 35
                52
CAED: FO 05
                           BEQ SRIN3
CAEF:09 20
                53
                           ORA
                                #$20
                                          ; ELSE SET BIT 5 TO OFFSET FOR PASCAL
                           STA STSBYTE, X ; AND SAVE IN STATUS TEMP
CAF1:9D B8 05
                54
CAF4:60
                                          ;CY=1 MEANS DATA IS AVAILABLE
                55 SRIN3 RTS
CAF5:
                56 *
                57 * SROUT CHECKS IF TOR IS EMPTY + HARDWARE HANDSHAKE IS OK
CAF5:
 CAF5:
```

CAF5: A4 26

59 SROUT LDY SLOT16

```
CAF7: B9 89 C0 60
                                   LDA STREG, Y
           CAFA: 29 70
                          61
                                    AND #$70
           CAFC:C9 10
                                    CMP #$10
                                                 ; EQU IF TOR EMPTY, DCD, DSR, & CTS
                          62
           CAPE: 60
                          63
                                    RTS
           CAFF:
                          64 *
           CAFF:
                          65 ****************
           CAFF:
                          66 * GENERAL INPUT ROUTINE *
           CAFF:
                          67 ***************
           CAFF: 20 D2 CA
                          68 INPUT JSR SRIN
           CB02:90 15
                                    BCC
                          69
                                         NOINPUT1
           CB04:
                          70 *
           CB04:B9 88 CO
                          71
                                   LDA RDREG, Y ;GET THE ACIA INPUT
                                    ORA #$80
           CB07:09 80
                                                  SET HI BIT FOR BASIC
           CB09:C9 8A
                          73
                                    CMP #$8A
                                                  ;LINEFEED?
           CBOB: DO 09
                          74
                                    BNE INPUT2
                          75 *
           CBOD:
           CBOD: A8
                          76
           CBOE: BD 38 07
                          77
                                    LDA MISCFLG, X ; SEE IF WE SHOULD EAT IT
           CB11:29 20
                           78
                                    AND
                                         #520
100
           CB13:DO 03
                          79
                                    BNE NOINPUT ; IF SO, JUST KEEP IT A SECRET
           CB15:98
                          80
                                    TYA
                          81 *
           CB16:
           CB16:38
                          B2 INPUT2 SEC
                                                  ; INDICATE DATA
           CB17:60
                                    RTS
                          83
           CB18:
                          84 *
           CB18:18
                          85 NOINPUT CLC
                                                  ; CARRY CLEAR FOR NO INPUT
           CB19:60
                          86 NOINPUT1 RTS
           CB1A:
                          87 *
           CB1A:
                          88 ***************
                          89 * GENERAL OUTPUT ROUTINE *
           CB1A:
                          90 ****************
           CB1A:
           CB1A:
                          91 *
           CB1A:
                          92 * START OF COMMAND CHECK ROUTINE
           CB1A:
                          93 *
           CB1A: A4 26
                          94 CMDSEQCK LDY SLOT16
                               LDA DIPSW1,Y
           CB1C:B9 81 CO
                          95
           CB1F:4A
                          96
           CB20:B0 36
                          97
                                    BCS NOCMD
                                                  ;DON'T WORRY ABOUT CMD SEQ FOR SIC
           CB22:BD B8 04
                          98
                                    LDA STATEFLG, X
           CB25:29 07
                          99
                                    AND #$07
                                                  ; ARE WE IN A COMMAND SEQUENCE?
mobre
           CB27:F0 05
                          100
                                    BEQ ESCCHECK
           CB29:20 FC CD
                         101
                                    JSR CMDPROC ; IF SO, GOTO COMMAND CENTRAL
           CB2C:38
                         102
                                    SEC
                                                  ; INDICATE COMMAND
           CB2D: 60
                         103
                                    RTS
                         104 *
           CB2E:
           CB2E: A5 27
                         105 ESCCHECK LDA CHARACTER
           CB30: 29 7F
                         106
                                    AND #S7F
                                                  ; IGNORE HIGH BIT
           CB32:DD 38 05 107
                                    CMP CMDBYTE, X ; IS THIS BEGINNING OF A CMD SEQ?
           CB35:D0 05
                         108
                                    BNE XOFFCK
           CB37:FE B8 04
                         109
                                    INC STATEFLG, X ; START UP COMMAND MODES
           CB3A:38
                                                  ; INDICATE COMMAND
                         110
                                    SEC
100
           CB3B:60
                         111
                                    RTS
           CB3C:
                          112 *
           CB3C:BD 38 07
                         113 XOFFCK LDA MISCFLG, X ; IS XON ENABLED?
           CB3F:29 08
                         114
                                     AND #$08
           CB41:F0 15
                         115
                                                 ;SKIP THIS IF NOT
                                    BEQ NOCMD
           CB43:
           CB43:20 FF CA 117
                                    JSR INPUT ; ANY INPUT?
```

```
BCC NOCMD
                                         ; IF NOT, GO OUTPUT
              118
CB46:90 10
                                         ; IS IT AN XOFF?
                          CMP
                               #$93
CB48: C9 93
              119
                               XONWAIT ; IF SO, GO WAIT FOR ANOTHER INPUT
CB4A:FO OE
               120
                          BEO
CB4C:48
               121
                          PHA
                               MISCFLG, X ; CIC MODE?
CB4D: BD 38 07
               122
                          T.DA
                          LSR
CB50:4A
               123
                               A
                           LSR A
CB51:4A
               124
                           PLA
               125
CB52:68
CB53:90 04
                           BCC
               126
                               BUFBYTE, X ; IF SO, WE HAVE A BUFFER
                           STA
CB55:9D B8 06
               127
               128 NOCMD
                                        ; INDICATE NOT A CMD SEQ
CB58:18
               129 ANRTS
                           RTS
CB59:60
               130 *
CB5A:
                                GETCHAR ;GET ACIA/KBD DATA
CB5A:20 AA C8
               131 XONWAIT JSR
                                          IS IT AN XON?
CB5D: C9 91
               132
                          CMP #$91
                                         ; IF NOT, WAIT
               133
                           BNE
                               XONWAIT
CB5F:D0 F9
                                          ;OTHERWISE, INDICATE NOT A CMD SEQ
CB61:18
               134
                           CLC
                                          ; AND RETURN
CB62:60
               135
                          RTS
               136 *******************************
CB63:
               CB63:
CB63:
               139 OUTPUT JSR CMDSEQCK
                        BCS ANRTS
                                         ;DON'T OUTPUT COMMAND SEQUENCES
CB66: BO F1
               140
               141 *
CB68:
               142 OUTPUT1 JSR SCREENOUT
CB68:20 9E CC
                143 *
CB6B:
                144 OUTPUT2 LDY SLOT16
CB6B: A4 26
                           LDA DIPSW1, Y
CB6D: B9 81 CO
               145
 CB70:4A
                           LSR
                           BCC OUTPUT3 ;SKIP ETX/ACK FOR NATIVE MODES
 CB71:90 4E
                147
                           LSR A
 CB73:4A
                148
                           BCC OUTPUT3 ; BRANCH IF NOT P8A EMULATION
 CB74:90 4B
                149
                150 *
 CB76:
                151 **************
 CB76:
                152 * PSA ETX/ACK STUFF*
 CB76:
                153 *************
 CB76:
                154 * AFTER 148 CHARACTERS BUT NOT WITHIN AN ESCAPE SEQUENCE
 CB76:
                155 * OF UP TO 5 CHARACTERS, THE HANDSHAKE IS PERFORMED
156 * (WILL DELAY UNTIL 'NOT ESC' AND THEN 4 MORE CHARS
 CB76:
 CB76:
 CB76:
                157 * OR UNTIL AN 'ESC')
 CB76:
                159 PRAGUTI LDA CHARACTER ; SAVE CHAR ON STACK
 CB76: A5 27
                           PHA
 CB78:48
                160
                                 HANDSHKE, X ; CHAR COUNT FOR BUFFER FULL
                           LDA
 CB79: BD 38 04
                161
                                          ; IF <103 THEN 153 CHARS IN BUFFER
                            CMP
                                 #103
 CB7C:C9 67
                162
                            BCC
 CB7E:90 10
                163
                                          ; IF >=108 THEN LESS THAN 149 CHARS
                            CMP
                                 #108
 CB80: C9 6C
                164
                                          ; SO NO HANDSHAKE IS NEEDED YET
                            BCS
                                 P8AOUT2
 CB82:B0 22
                165
                                           ;SETS CARRY IF 107 (149 SENT)
                            CMP
                                 #107
 CB84:C9 6B
                166
                167
 CB86:68
                168
                            PHA
 CB87:48
                            EOR
                                 #$9B
                                           : ESC?
                169
 CB88: 49 9B
                                           ; IGNORE HI-BIT
                            AND
                                 #$7F
 CB8A: 29 7F
                170
                                           COUNT AS 1 OF 5 IF NOT 'ESC'
                            BNE
                                 P8AOUT2
 CB8C:D0 18
                                          ;DON'T COUNT IF 149TH CHAR IS 'ESC'
                            BCS
                                 PSAOUT3
 CB8E:B0 19
                 173 *
                            LDA STATEFLG, X ; SEND QUERY CHAR TO PRINTER
 CB90: BD B8 04
                174 ETX
                                           ; (DEFAULT IS ETX)
                            AND #$1F
 CB93:29 1F
                175
```

```
CB95:09 80
                           176
                                       ORA #$80
            CB97:85 27
                           177
                                            CHARACTER
                                       STA
            CB99: 20 02 CC
                           178
                                       JSR
                                            ACIAOUT
            CB9C: 20 AA C8
                           179 ACK
                                       JSR
                                            GETCHAR
                                                      GET ACIA/KBD DATA
            CB9F:49 86
                                            #$86 ;ACK?
ETX ;IF NOT ACK, REPEAT HANDSHAKE
                           180
                                       EOR
            CBA1:DO ED
                           181
                                       BNE
                                            ETX
            CBA3:9D 38 04
                           182
                                            HANDSHKE, X ; INIT CHAR COUNT TO 255
                                       STA
            CBA6:
                           183 *
            CBA6: DE 38 04
                           184 PSAOUT2 DEC
                                            HANDSHKE, X
            CBA9:68
                           185 PSAOUT3 PLA
                                                      GET REAL CHAR TO OUTPUT
  175
            CBAA:85 27
                           186
                                       STA
                                            CHARACTER
            CBAC: 49 8D
                           187
                                       EOR
                                            #$8D
                                                      ; IF CR AND CR DELAY MODE
            CBAE: OA
                           188
                                       ASL
 -198
            CBAP: DO OA
                                       BNE
                                            PSAOUT4
                                                      ; THEN FAKE CHAR COUNT TO LESS THAN
            CBB1: BD B8 03
                           190
                                       LDA
                                            DELAYFLG, X ; 48 TO FORCE HANDSHAKE ON NEXT
            CBB4: 29 30
                           191
                                       AND
                                            #$30
                                                     ; CHARACTER OUT
  150
            CBB6: FO 03
                           192
                                       BEQ
                                            PSAOUT4
            CBB8:9D 38 04
                           193
                                       STA
                                            HANDSHKE, X
            CBBB:
                           194 *
 -150
            CBBB:20 02 CC
CBBE:4C EA CB
                           195 P8AOUT4 JSR
                                            ACIAOUT
                           196
                                       JMP.
                                            LEGEN
                                                      (SKIP DELAYS)
            CBC1:
                           197 **************
            CBC1:
                           198 * AND BACK TO NORMAL OUTPUT *
            CBC1:
                           199 ******************
            CBC1:20 02 CC
                           200 OUTPUT3 JSR ACIAOUT ; OUTPUT THE CHARACTER
 155
            CBC4:
                           201 *
            CBC4:
                           202 * NOW CHECK FOR CR, LF, AND FF DELAYS
            CBC4:
                           203 *
            CBC4:0A
                           204
                                       ASL A
            CBC5: A8
                           205
                                       TAY
           CBC6: BD B8 03
                           206
                                       LDA DELAYFLG, X ;GET DELAY PLAGS
           CBC9:C0 18
                           207
                                       CPY #$18
                                                     FORM FEED?
           CBCB: FO OC
                           208
                                       BEO
                                            OUTDLY1
           CBCD: 4A
                           209
                                       LSR A
           CBCE: 4A
                           210
                                       LSR
                                                      ; RIGHT JUSTIFY LF DELAY
           CBCF:CO 14
#514
                           211
                                       CPY
                                                      ;LINE FEED?
           CBD1:F0 06
                           212
                                       BEO
                                            OUTDLY1
           CBD3:4A
                           213
                                       LSR
           CBD4:4A
                           214
                                       LSR
                                                      RIGHT JUSTIFY CR DELAY
           CBD5:CO 1A
                                       CPY
                                            #$1A
                                                      :CARRIAGE RETURN?
           CBD7: DO 25
                           216
                                       BNE
                                            OUTPUTEND
           CBD9: 29 03
                           217 OUTDLY1 AND
                                            #$03
                                                     JUST WANT LOWEST 2 BITS
           CBDB: FO OD
                           218
                                       BEQ
                                                     ; NO DELAY INDICATED
           CBDD: AR
                           219
                                       TAY
           CBDE: B9 FE CB
                           220
                                       LDA
                                            DLYTBL-1, Y
           CBE1: A8
                           221
                                       TAY
                                                     ;DELAY IN 32 MSEC INCREMENTS
           CBE2: A9 20
                           222 OUTDLYLP LDA #32
           CBE4: 20 C4 CA
                                      JSR WAITMS
                           223
           CBE7:88
                           224
                                      DEV
           CBE8: DO F8
                           225
                                      BNE OUTDLYLP
           CBEA:
                           226 *
           CBEA:
                           227 * CHECK ON LF GENERATION OPTION
           CBEA:
                           228 *
           CBEA: A5 27
                           229 LFGEN
                                      LDA CHARACTER
           CBEC: OA
                           230
                                       ASL A
           CBED: C9 1A
                           231
                                       CMP
                                            #$1A
                                                     ; CARRIAGE RETURN?
           CBEF: DO OD
                           232
                                       BNE
                                            OUTPUTEND
           CBF1:BD 38 07 233
                                       LDA MISCFLG, X ; IS LF GENERATE ENABLED?
```

```
CBF4:6A
             234
CBF5:90 07
                         BCC OUTPUTEND
              235
                         LDA #$8A
CBF7: A9 8A
              236
                         STA CHARACTER ; LINE FEED
CBF9:85 27
              237
CBFB: 4C 6B CB 238
                         JMP OUTPUT2 ; (DON'T ECHO IT)
CBFE:60
              239 OUTPUTEND RTS
              240 *
CBFF:
                                       ;32 MSEC
              241 DLYTBL DFB $01
CBFF:01
              242
                         DFB $08
                                       ;1/4 SEC
CC00:08
CC01:40
              243
                         DFB $40
                                        : 2 SEC
              244 ******************
CC02:
              245 * ACIA OUTPUT ROUTINE *
CC02:
              246 ******************
CC02:
CC02:20 F5 CA
              247 ACIAOUT JSR SROUT ; READY FOR OUTPUT?
                     BNE ACIAOUT
CC05:D0 FB
              248
CC07:98
              249
                                      ; PREPARE TO ADDRESS ACIA,
CC08:09 89
              250
                         ORA #$89
CCOA: A8
              251
                         TAY
                                       ; CAUSING 6502 FALSE READ TO OCCUR
CCOB: A5 27
                         LDA CHARACTER ; ON PAGE $BF (AVOIDING RDR READ)
              252
CCOD: 99 FF BF 253
                         STA $BFFF,Y ;HERE YOU ARE ACIA
CC10:60
              254
                         RTS
CC11:
              255 *
CC11:
              256 *********************
CC11:
              257 * RESTORE CURSOR (NOT FOR PASCAL) *
CC11:
              258 * (A-REG SHOULD CONTAIN NEW CHAR) *
              259 *******************
CC11:
CC11:48
              260 RESTORE PHA ; SAVE NEW CHARACTER
CC12: A4 24
              261
                         LDY CH
                         LDA CHARACTER ;OLD CHARACTER
CC14:A5 27
CC16:91 28
              262
                         STA (BASL), Y
              263
CC18:68
              264
                         PLA
CC19:
              265 *
CC19:C9 95
                         CMP #$95
                                       ;SCREEN PICK?
              266
                         BNE RESTOREND
CC1B:DO OC
              267
CC1D: A5 27
                          LDA CHARACTER ; IF SO, USE SCREEN CHAR
CC1F:C9 20
              269
                         CMP #$20
                                       ; INVERSE?
                         BCS RESTOREND
CC21:B0 06
               270
                         JSR GETXLATE ; REVERSE THE TRANSLATION EOR REVMASK, Y
CC23:20 DF CC
              271
CC26:59 DB CC
               272
CC29:85 27
               273 RESTOREND STA CHARACTER
CC2B:60
               274
                         RTS
               275 *
CC2C:
                        CHN SSC.UTIL
CC2C:
              276
```

```
CC2C:
          CC2C:
           CC2C:
                          4 * APPLE II SSC FIRMWARE
           CC2C:
                          5 *
           CC2C:
                          6 * BY LARRY KENYON
11.078
           CC2C:
           CC2C:
                                -JANUARY 1981-
1-d/0
           CC2C:
           CC2C:
                          10 * (C) COPYRIGHT 1981 BY APPLE COMPUTER, INC. *
           CC2C:
                          11 *
19
          CC2C:
                          12 ****************************
           CC2C:
                          13 *
           CC2C:
                         14 * UTILITY ROUTINES
100
           CC2C:
                         15 *
           CC2C:
                         16 ******************
           CC2C:
                          17 * PASCAL-BASIC KEYBOARD FETCH *
- 10
          CC2C:
                          18 ********************
                                   CLC ; RETURN CARRY CLEAR FOR NO DATA LDA MISCFLG, X
                          19 CKKBD CLC
           CC2C:18
           CC2D: BD 38 07
                         20
                                   AND #$04 ;ANSWER NO IF KEYBOARD IS DISABLED BEQ CKKBDXIT
- 19
           CC30:29 04
                          21
           CC32:F0 09
                          22
                          23 *
          CC34:
19
           CC34:AD 00 CO
                          24 CKKBD1 LDA KBD
           CC37:10 04
                          25
                                   BPL CKKBDXIT
           CC39:8D 10 CO
                                   STA KBDSTRB
- 15
           CC3C:38
                                    SEC
                                                 ; INDICATE DATA
           CC3D:60
                          28 CKKBDXIT RTS
          CC3E:
                          29 **********************
-- 25
          CC3E:
                         30 * GET A CHAR FROM KEYBOARD FOR BASIC ONLY *
           CC3E:
                          31 ***********************
           CC3E: E6 4E
                          32 GETKBD INC RNDL
                                                ;MIX UP RANDOM # SEED
CC40:D0 02
                         33
                                   BNE GETKBD1 ; FOR BASIC
          CC42:E6 4F
                                    INC RNDH
          CC44:20 2C CC
                                                 ; KEYBOARD FETCH ROUTINE
- 65
                          35 GETKBD1 JSR CKKBD
          CC47: B8
                         36 CLV
                                                 ; INDICATE NO ESCAPE SEQUENCE
                                   BCC CKKBDXIT ; EXIT IF NO KEY PRESS
JSR RESTORE ; DO BASIC CURSED DUTY
          CC48:90 F3
                         37
- 08
          CC4A:20 11 CC
                         38
          CC4D: 29 7F
                          39
                                   AND #$7F
          CC4F:DD 38 05
                                   CMP CMDBYTE, X ; IS IT THE START OF A COMMAND?
                         40
          CC52:D0 3D
                                   BNE GETKBOONE ; IF NOT, EXIT INDICATING DATA
- 45
                         41
          CC54:A4 26
                          42
                                   LDY SLOT16
          CC56:89 81 CO
                                   LDA DIPSW1,Y ;ONLY DO CMD ESC FOR PPC, SIC MODES
-16
          CC59:4A
                         44
                                   LSR A
          CC5A:B0 35
                         45
                                  BCS GETKBDONE
          CC5C:
                         46 *****************
- 198
          CC5C:
                         47 * KEYBOARD ESCAPE HANDLER *
          CC5C:
                         48 ****************
           CC5C: AO OA
                         49 KBDESC LDY #$A ;FIRST PRINT A PROMPT
-
          CC5E:B9 93 CC
                         50 PROMPTLOOP LDA PROMPTBL, Y
          CC61:85 27
                                 STA CHARACTER
CC63:98
                          52
                                   TYA
          CC64:48
                                   PHA
          CC65:20 A3 CC
                          54
                                   JSR SCREENOUT1 ; ALWAYS SEND TO SCREEN
          CC68:68
-08
                         55
                                   PLA
          CC69: A8
                          56
                                   TAY
          CC6A:88
                          57
                                   DEY
          CC68:10 F1
                                   BPL PROMPTLOOP
-29
          CC6D:
                         59 *
```

- 100

-88

-

MATERIAL PROPERTY.

```
LDA #1
                                         START OUT IN COMMAND STATE 1
CC6D: A9 01
                60
                61
                           JSR SETOSTATE
CC6F: 20 7B CE
                62 *
CC72:
                                          ; WAIT FOR KEYBOARD CHARACTER
                           JSR CKKBD1
                63 GETCMD
CC72:20 34 CC
                           BPL GETCMD
CC75:10 FB
                64
                                          ; BACKSPACE?
CC77:C9 88
                65
                           CMP
                               #$88
                                          ; IF SO, THEN START OVER
                           BEO KBDESC
CC79:F0 E1
                66
                               CHARACTER
                 67
                           STA
CC7B:85 27
                 68 *
CC7D:
                                SCREENOUT1
                 69
CC7D: 20 A3 CC
                           JSR CMDSEQCK ; PUMP THRU CMD INTERPRETER
CC80:20 1A CB
                 71 *
CCB3:
                            LDA STATEFLG, X ; ARE WE DONE?
CC83: BD B8 04
                 72
                                 #$07
 CC86:29 07
                 73
                            AND
                            BNE GETCMD
                                           ; IF NOT, GO AGAIN
 CC88:D0 E8
                 74
                                #S8D ;FORCE BACK A CARRIAGE RETURN CHARACTER
                 75 *
CC8A:
                            LDA
                 76
77
 CCBA: A9 8D
                            STA
 CC8C:85 27
                                           ; INDICATE THAT A CMD SEQ HAS OCCURRED
                            BIT IORTS
 CC8E: 2C 58 FF
                 78
                                           ; INDICATE SUCCESS
                 79 GETKBDONE SEC
 CC91:38
 CC92:60
                 80
                            RTS
 CC93:
                 81 *
                 82 *
 CC93:
                 83 PROMPTBL ASC ":CSS
                                           ELPPA"
 CC93: BA C3 D3
 CC96:D3 A0 C5
 CC99:CC DO DO
 CC9C:C1
                            DFB $8D
 CC9D: 8D
                 84
                 85 *
 CC9E:
 CC9E:
                 87 * ROUTINE TO PRINT A CHARACTER ON THE CURRENT DISPLAY *
 CC9E:
                  88 ******************************
 CC9E:
                  89 SCREENOUT LDA MISCFLG, X
 CC9E:BD 38 07
                                         ; IF SCREEN DISABLED
                            BPL NOOUT
                  90
 CCA1:10 13
                 91 *
 CCA3:
                  92 SCREENOUT1 LDA MISCFLG, X ; ENTRY AFTER ECHO CHECK
 CCA3: BD 38 07
                                 #$02 ;IF IT ISN'T CIC MODE,
ASCREEN ;ALWAYS USE THE APPLE SCREEN
                            AND #$02
 CCA6: 29 02
                  93
                             BEO
  CCA8: FO OD
                  94
                             LDA STATEFLG, X ; CURRENT SCREEN = APPLE SCREEN?
  CCAA: BD B8 04
                  95
                             AND #$38
  CCAD: 29 38
                  96
                            BEQ ASCREEN ; SLOT 0= APPLE SCREEN
  CCAF:F0 06
                  97
                  98 *
  CCB1:
                                             JUMP TO CNOO SPACE
                             TXA
  CCB1:8A
                  99
                             PHA
  CCB2:48
                 100
                                 #>SENDCD-1 ; TO VECTOR TO THE PERIPHERAL
                             LDA
                 101
  CCB3: A9 AF
                                             ; IN THE CHAIN SLOT
                             PHA
                 102
  CCB5:48
                 103 NOOUT
                            RTS
  CCB6:60
                 104 *
  CCB7:
                 105 * APPLE 40-COL SCREEN DRIVER
  CCB7:
                  106 *
  CCB7:
                 107 ASCREEN JSR GETXLATE ;GET THE TRANSLATE OPTIONS
  CCB7:20 DF CC
                                            ;SET HIGH BIT OF CHAR
                             ORA #$80
  CCBA: 09 80
                 108
                                  #SEO
                                             ; LOWERCASE?
                             CMP
                  109
  CCBC: C9 E0
                                  TESTLETTER
                             BCC
                  110
  CCBE: 90 06
                             EOR LCMASK, Y ; DO LOWERCASE TRIP
  CCC0:59 D3 CC
                  111
                                            ;ALL REGS ARE PRESERVED
                 112 TOSCREEN JMP VIDOUT
  CCC3:4C F6 FD
                  113 *
  CCC6:
                  114 * IF UPPERCASE, WE ONLY MAP LETTERS
  CCC6:
```

```
CCC6:
               115 *
               116 TESTLETTER CMP #$C1
117 BCC TOSCREEN
118 CMP #$DB
CCC6: C9 C1
CCC8:90 F9
               117
CCCA: C9 DB
               118
                                           1>2?
CCCC: BO F5
                            BCS TOSCREEN
                119
CCCE:59 D7 CC 120
                           EOR UCMASK, Y
CCD1:90 FO
                121
                           BCC TOSCREEN ; <ALWAYS>
CCD3:
                122 *
CCD3:
                123 * MASKS FOR CASE TRANSLATION
CCD3:20 00 E0 124 LCMASK DFB $20,800,8E0,$20
CCD6:20
CCD7:00 00 00 125 UCMASK DFB $00,$00,$00,$CO
CCDA:CO
CCDB:00 00 E0 126 REVMASK DPB $00,500,$E0,$C0
CCDE: CO
CCDF:
                127 *
CCDF:BD B8 03 128 GETXLATE LDA DELAYFLG,X ;TRANSLATE OPTIONS IN B6-B7 CCE2:2A 129 ROL A
CCE3: 2A
                            ROL A
                130
                            ROL A
AND #$03
CCE4: 2A
                131
CCE5:29 03
                132
                            TAY
CCE7: A8
                133
                            LDA CHARACTER
CCE8: A5 27
                134
CCEA:60
                            RTS
                135
CCEB:
                136 *
```

(listings continued on next page)

-

- 10

FE

-55

- 12

```
138 CHN SSC.CMD
CCEB:
               1 ************
CCEB:
CCEB:
CCEB:
               3 * APPLE II SSC FIRMWARE
CCEB:
              5 * BY LARRY KENYON
CCEB:
CCEB:
               6 *
7 *
CCEB:
                   -JANUARY 1981-
CCEB:
              B *
CCEB:
               9 * (C) COPYRIGHT 1981 BY APPLE COMPUTER, INC. *
              10 *
CCEB:
              11 ******************************
CCEB:
CCEB:
              12 *
              13 * SSC COMMAND PROCESSOR
CCEB:
CCEB:
              14 *
              15 *********************
CCEB:
              16 ********************************
CCEB:
              17 * COMMAND TABLE (USED BY COMMAND PROCESSER ROUTINE *
              18 ***************
CCEB:
              19 CMDTBL DFB $42
20 DFB $67
                                     ;B(REAK)
CCEB: 42
CCEC:67
              20
CCED: CO
                        DFB >BREAKCMD-1
CCEE:54
              22
                       DPB $54 ;T(ERMINAL)
                                     ;CIC
CCEF: 47
              23
                        DFB $47
                                                 NS=7
                       DFB >TERMCMD+1
CCF0: A6
              24
CCF1:43
              25
                        DFB $43 ;C(R GENERATE)
CCF2:87
              26
                        DFB S87
                                     ; PPC
                                                 NS=7
                        DFB >TERMCMD-1
CCF3: A6
              27
CCF4:51
                        DFB $51 ;Q(UIT)
              28
CCF5:47
                        DFB $47
                                     ;CIC
                                                 NS=7
              29
CCF6:B8
              30
                        DFB >QUITCMD-1
CCF7:52
                        DFB
                           $52 ;R(ESET)
CCF8:C7
                        DFB
                            $C7
                                                 NS=7
CCF9: AC
                        DFB
                            >RESETCMD-1
CCFA:5A
                        DFB
                           $5A ;Z COMMAND
                                     ;CIC PPC PAS
                                                 NS=7
CCFB: E7
              35
                        DFB SE7
CCFC:F3
                        DFB >ZCMD-1
              36
                                     ;I COMMAND
CCFD: 49
              37
                        DFB $49
CCFE:90
              38
                        DFB
                            $90
CCFF:D3
              39
                        DFB >ICMD-1
                        DFB $4B
                                     ; K COMMAND
CD00:4B
              40
                                     ; PPC
                        DFB $90
                                                 NS=0
CD01:90
              42
                        DFB >KCMD-1
CD02:DF
CD03:
              43 *
                                     ; E(CHO)
                        DFB $45
CD03:45
              44
                        DFB $43
                                     CIC
CD04:43
              45
                        DFB
CD05:80
                            $80
CD06:46
                        DFB
                           $46
                                     ;F(ROMKYBD)
                        DFB $E3
                                     ;CIC PPC PAS NS=3
CD07: E3
CD08:04
              49
                        DFB
                           504
CD09:4C
              50
                        DFB $4C
                                     ; L(F GENERATE)
CDOA: E3
              51
                        DFB SE3
                                     ;CIC PPC PAS NS=3
                        DFB $01
CD0B: 01
              52
CD0C:58
                        DFB $58
                                     ;X(OFF)
              53
                                     ;CIC PPC PAS NS=3
CDOD: E3
                        DFB $E3
                        DFB $08
CD0E:08
CDOF:54
                        DFB $54
                                     ;T(ABBING)
              56
                                     ; PPC
CD10:83
              57
                       DFB $83
                                                 NS=3
```

```
CD11:40
                           58
                                      DFB $40
           CD12:53
                                                      ;S(HIFTING)
                           59
                                       DFB
                                            $53
           CD13:43
                                       DFB
                           60
                                            $43
                                                      CIC
                                                                    NS=3
           CD14:40
                                       DFB
                                            $40
           CD15:4D
                                       DFB
                            62
                                            $40
                                                      ;M(UNCH LF)
           CD16: E3
                                       DFB
                                                      ;CIC PPC PAS NS=3
                            63
                                            SE3
           CD17:20
                                       DFB
                            64
                                            $20
           CD18:
                            65 *
           CD18:00
                                       DFB
                                                      ; END OF FIRST PART MARKER
                            66
                            67 *
           CD19:
           CD19:42
                            68 CMDTBL1 DFB
                                            $42
                                                      ;B(AUD)
           CD1A:F6
                            69
                                       DFB
                                                      ;CIC PPC PAS NS=6
           CD1B:7C
                            70
                                       DFB
                                            >BAUDCMD-1
                            71
72
           CD1C:50
                                       DFB
                                            $50
                                                      ;P(ARITY)
 100
           CD1D: F6
                                       DFB
                                            SF6
                                                      ;CIC PPC PAS NS=6
           CD1E: 9A
                                       DFB
                            73
                                            >PARITYCMD-1
           CD1F:44
                            74
                                       DFB
                                                      D(ATA)
                                            $44
                            75
           CD20:F6
                                       DFB
                                                      ;CIC PPC PAS NS=6
                                            SF6
           CD21:9B
                            76
                                       DFB
                                            >DATACMD-1
           CD22:46
                            77
                                       DFB
                                            $46
                                                      F(F DELAY)
 125
                            78
                                       DFB
                                            SF6
                                                      CIC PPC PAS NS=6
           CD24:46
                            79
                                       DFB
                                            >FFCMD-1
           CD25:4C
                            80
                                       DFB
                                            $4C
                                                      ;L(F DELAY)
           CD26:F6
                                       DFB
                                                      CIC PPC PAS NS=6
                            81
                                            SF6
           CD27:40
                            82
                                       DFB
                                            >LFCMD-1
           CD28:43
                            83
                                       DFB
                                            $43
                                                       C(R DELAY)
           CD29:F6
                            84
                                       DFB
                                            $F6
                                                      CIC PPC PAS NS=6
           CD2A:3A
                            85
                                       DFB
                                            >CRCMD-1
           CD2B:54
                            86
                                       DFB
                                            $54
                                                      ;T(RANSLATE)
           CD2C:D6
                            87
                                       DFB
                                            SD6
                                                      CIC PPC
                                                                    NS=6
           CD2D: 34
                            88
                                       DFB
                                            >TRANCMD-1
           CD2E: 4E
                            89
                                       DFB
                                            S4E
                                                      ; N COMMAND
           CD2F:90
                            90
                                       DFB
                                                                    NS=0
                                            $90
                                                           PPC
           CD30: E8
                            91
                                       DFB
                                            >NCMD-1
           CD31:53
                            92
                                       DFB
                                                      ;S(CREENSLOT)
                                            $53
           CD32:56
                            93
                                       DFB
                                                                    NS=6
                                                      :CIC
                                            $56
           CD33:60
                            94
                                       DFB
                                            >SSLOTCMD-1
                            95 *
           CD34:
           CD34:00
                                      DFB $00
                                                      ; END OF TABLE MARKER
                            96
           CD35:
                            97 *
           CD35:
                            98 *************
man All
           CD35:
                            99 * COMMAND ROUTINES
                           100 * (CALLED BY PARSER) *
           CD35:
           CD35:
                           101 * (MUST START IN
                           CD35:
           CD35:
           CD35: A9 3F
                           104 TRANCMD LDA #S3F
                                                      ;SET SCREEN TRANSLATE OPTIONS
           CD37: AO 07
                                      LDY
                           105
                                            #S7
           CD39:D0 10
                           106
                                       BNE
                                            DELAYSET ; < ALWAYS>
           CD3B:A9 CF
                           107 CRCMD
                                            #$CF
                                      LDA
                                                      SET CR DELAY
           CD3D: A0 05
                           108
                                       LDY
                                            #$5
           CD3F:DO OA
                           109
                                       BNE
                                            DELAYSET
                                                     ; <ALWAYS>
           CD41:
                           110 *
           CD41:A9 F3
                           111 LFCMD
                                       LDA
                                            #$F3
                                                      SET LF DELAY
           CD43:A0 03
                           112
                                       LDY
                                            #$3
           CD45:D0 04
                           113
                                            DELAYSET ; < ALWAYS>
                                       BNE
                           114 *
           CD47:
           CD47: A9 FC
                           115 FFCMD
                                      LDA #SFC
                                                      SET FF DELAY
```

```
LDY #$1
CD49: AO 01
               116
CD48:3D 88 03 117 DELAYSET AND DELAYPLG, X ; DON'T DISTURB THE OTHER FLAGS
                           STA
                                ZPTMP1
               118
CD4E:85 2A
                                PARAMETER, X
CD50:BD 38 04
               119
                           AND
                                #$03
                                          JUST USE TWO BITS
               120
CD53:29 03
                           CLC
CD55:18
               121
                                          ;ONCE FOR FUN
CD56:6A
               122
                           ROR
                                          ; CHANGE DIRECTIONS
               123 ROTATE
                           ROL A
CD57:2A
CD58:88
               124
                           DEY
                           BNE ROTATE
                                          ; PREPARE IT TO OR INTO THE FLAGS
CD59:D0 FC
               125
               126 *
CD5B:
                           ORA ZPTMP1
CD5B:05 2A
                127
CD5D: 9D B8 03
                           STA
                                DELAYFLG, X
               128
                129
                           RTS
CD60:60
                130 *
CD61:
                                         ;SET SLOT COMMAND
CD61:29 07
                131 SSLOTCMD AND #$7
CD63:0A
                132
                           ASL A
CD64:0A
                133
                            ASL
                                A
                134
                            ASL A
CD65:0A
CD66:85 2A
                135
                            STA
                                ZPTMP1
CD68:0A
                136
                            ASL
                                SLOT16
                                          ; MAKE SURE WE DON'T SET IT
                            CMP
CD69:C5 26
                137
                            BEQ SSLOTCMD1 ; TO OUR OWN SLOT
CD6B: FO OF
                138
CD6D: BD B8 04
                            LDA STATEFLG, X
                139
                                 #$C7
                                          ; PUT NEW SLOT NUMBER IN BITS 3-5
                            AND
 CD70:29 C7
                140
                            ORA ZPTMP1
                                           ; OF CMDBYTE, X
CD72:05 2A
                141
                           STA STATEFLG, X
CD74: 9D B8 04
                142
                                          ;STORE ZERO INTO
                            LDA #0
 CD77:A9 00
                143
                            STA CHNBYTE, X ; SLOT OFFSET (SET TO CNOO ENTRY)
 CD79:9D 38 06
                144
                145 SSLOTCMD1 RTS
 CD7C:60
                146 *
 CD7D:
                                          ;SET NEW BAUD RATE
 CD7D: 29 OF
                147 BAUDCMD AND #$OF
                           BNE BAUDCMD2
 CD7F:D0 07
                149 BAUDCMD1 LDA DIPSW1,Y ;ZERO PARM = RELOAD FROM SWITCHES
 CD81:B9 81 CO
                            LSR A
                150
 CD84:4A
 CD85:4A
                151
                            LSR
 CD86:4A
                152
                            T.S.R.
                153
                            LSR
 CD87:4A
                                           ;SET INT. BAUD RATE GENERATOR
                154 BAUDCMD2 ORA #$10
 CD88:09 10
                           STA ZPTMP1
 CD8A:85 2A
                155
                            LDA
                                 #SEO
 CD8C: A9 E0
                156
                157 CTLREGSET STA ZPTMP2
 CD8E:85 2B
                158
                           LDA CTLREG, Y
 CD90: B9 8B CO
 CD93:25 2B
                159
                            AND
                                 ZPTMP2
                160
                            ORA ZPTMP1
 CD95:05 2A
 CD97:99 8B CO
                161
                            STA CTLREG, Y
                162
                           RTS
 CD9A:60
 CD9B:
                163 *
                                            ;TRICK: SO CTLREG, Y ACTUALLY
 CD9B:88
                164 PARITYCMD DEY
                                           ADDRESSES THE COMMAND REG
 CD9C:
                165 *
 CD9C:
                166 *
                                           ;SET NEW # OF DATA BITS
 CD9C:OA
                167 DATACMD ASL A
                            ASL A
 CD9D: OA
                168
                169
                            ASL
 CD9F:OA
                 170
                            ASL
 CDAO: OA
                171
                            ASL
                172 DATACMD1 STA ZPTMP1
 CDA1:85 2A
                            LDA #S1F
 CDA3: A9 1F
                173
```

```
CDA5: DO E7
                          174
                                      BNE CTLREGSET ; < ALWAYS>
           CDA7:
                           175 *
           CDA7:1E B8 04
                          176 TERMCMD ASL STATEFLG, X ; SET TERMINAL MODE
                               SEC
           CDAA: 3B
                           177
           CDAB: BO 10
                           178
                                      BCS
                                           CCMD1
                                                     ; <ALWAYS>
           CDAD:
                           179 *
           CDAD: 99 89 CO
                          180 RESETCMD STA RESET, Y
                                                     ; DROP RTS, DTR
           CDB0:20 93 FE
                          181
                                   JSR SETSCR
                                                     ; PR#0
           CDB3:20 89 FE
                          182
                                      JISR
                                           SETKBD
                                                     ; IN#0
           CDB6: AE F8 07
                          183
                                      LDX MSLOT
           CDB9:1E B8 04
                          184 QUITCMD ASL
                                           STATEFLG, X ; CLEAR TERMINAL MODE
           CDBC:18
                          185
                                      CLC
           CDBD: 7E B8 04
                          186 QCMD1
                                      ROR
                                           STATEFLG, X
           CDC0:60
                          187
                                      RTS
                           188 *
           CDC1:
           CDC1: B9 8A CO
                          189 BREAKCMD LDA CMDREG, Y ; SEND BREAK SIGNAL
CDC4:48
                           190
                                      PHA
                                                      ; FOR 233 MILLISECONDS
           CDC5:09 OC
CDC7:99 8A CO
                           191
                                      ORA
                                            #soc
                                      STA CMDREG, Y
                          192
           CDCA: A9 E9
                                      LDA
                                           #233
                           193
                                                     ;DELAY FOR 233 MICROSEC.
           CDCC: 20 C4 CA
                                      JSR WAITMS
                          194
           CDCF:68
                           195
                                      PLA
                                                      ; RESTORE OLD COMMAND REG CONTENTS
           CDD0:99 8A CO
                          196
                                      STA CMDREG, Y
           CDD3:60
           CDD4:
           CDD4: A9 28
                          199 ICMD
                                      LDA
           CDD6: 9D 38 06
                                      STA PWDBYTE, X ; SET PRINTER WIDTH TO 40
           CDD9: A9 80
                          201
                                      LDA
           CDDB: 1D 38 07
                          202
                                      ORA
                                           MISCFLG, X ; SET SCREEN ECHO
           CDDE:DO 05
                          203
                                      BNE
                                           KCMD2 ; <ALWAYS>
           CDE0:
                          204 *
                                           #$FE ;RESET THE LF GENERATE FLAG
           CDEO: A9 FE
                           205 KCMD
                                      LDA
           CDE2: 3D 38 07
                          206 KCMD1
                                      AND
                                           MISCFLG, X
           CDE5:9D 38 07 207 KCMD2
                                      STA
           CDE8:60
                          208
                                      RTS
           CDE9:
                          209 *
           CDE9: C9 28
                          210 NCMD
                                      CMP
                                            #40
                                                     ;>=40?
           CDEB: 90 OE
                          211
                                      BCC
                                           ZCMDRTS ; IF NOT, JUST EXIT
           CDED: 9D 38 06
                                           PWDBYTE, X ; SET NEW PRINTER WIDTH
                          212
                                      STA
           CDFO: A9 3F
                          213
                                      LDA
                                           #$3F
                                                    ;DISABLE SCREEN, SET LISTING MODE
           CDF2:DO EE
                          214
                                      BNE
                                           KCMD1
                                                     ; <ALWAYS>
                          215 *
           CDF4:
           CDF4:1E 38 05
                                      ASL CMDBYTE, X ; DISABLE COMMAND RECOGNITION SEC
                          216 ZCMD
           CDF7:38
           CDF8:7E 38 05
                          218
                                      ROR
                                           CMDBYTE, X
           CDFB:60
                          219 ZCMDRTS RTS
           CDFC:
                          220 *
           CDFC:
                           221 ***********************
           CDFC:
                          222 * VECTOR ACCORDING TO COMMAND STATE *
                          223 **************************
           CDFC:
           CDFC: A8
                          224 CMDPROC TAY
                                                     ; A-REG=COMMAND STATE
           CDFD: A5 27
                                      LDA CHARACTER
                          225
           CDFF: 29 7F
                          226
                                      AND #$7F
           CEO1:
                           227 *
           CE01:C9 20
                          228
                                      CMP #$20
                                                     ; SKIP SPACES FOR ALL MODES
           CE03:D0 09
                          229
                                      BNE CMDPROC2
           CE05:C0 03
                          230
                                      CPY
                                           #$3
                                                     ; EXCEPT MODE 3
           CE07:FO 01
                          231
                                      BEQ CMDPROC1
```

```
CE09:60
              232
              233 CMDPROC1 LDA #$4
CEOA: A9 04
              234
                          BNE SETOSTATE ; < ALWAYS>
CEOC:DO 6D
              235 *
CEOE:
               236 CMDPROC2 CMP #$0D
                                         ; CARRIAGE RETURN?
CEOE: C9 OD
                     BNE CMDPROC4 ;
CE10:D0 12
               237
                               ZEROSTATE ; ABORT FOR STATES 0-5, EXIT FOR 6,7
CE12:20 79 CE 238
                          JSR
                                         ; IN STATE 7 WE VECTOR TO THE PROC
               239
                          CPY
                               #$07
CE15:C0 07
                               CMDPROC3 ;
CE17:F0 01
               240
                          BEQ
                                          ;OTHERWISE, JUST EXIT
CE19:60
               241
                          RTS
CE1A:
               242 *
               243 CMDPROC3 LDA #$CD ;ALL PROCS MUST START IN PAGE $CD
CE1A: A9 CD
                           PHA
CE1C: 48
               244
                           LOA PARAMETER, X
CE1D: BD 38 04 245
                           PHA
CE20:48
               246
                           LDY
                               SLOT16
                                         ; NEEDED BY BREAK CMD
CE21:A4 26
               247
CE23:60
               248
                          RTS
               249 *
CE24:
               250 CMDPROC4 STA ZPTEMP
CE24:85 35
                      LDA
                               #$CE
                                         ;ALL ROUTINES MUST START
CE26: A9 CE
               251
                                          ; IN PAGE SCE
               252
CE28:48
                           LDA STATETBL, Y
CE29:B9 30 CE
               253
               254
                           PHA
CE2C:48
                           LDA ZPTEMP
CE2D: A5 35
               255
                                         RTS TO COMMAND PROCEDURE
                           RTS
CE2F:60
               256
               257 *
CE30:
               258 * NOW THE STATE ROUTINES
CE30:
               259 *
CE30:
CE30:
               260 **************
               261 * STATE BRANCH TABLE *
CE30:
               262 **************
CE30:
CE30: A7
               263 STATETBL DFB >STATERR-1 ; BAD STATE
                         DFB >CSTATE1-1 ; < CMD> SEEN
 CE31:37
               264
                           DFB >CSTATE2-1 ;ACCUMULATE PARAMETER
               265
CE32:61
                         DFB >CDONE-1 ;SKIP UNTIL SPACE
DFB >CSTATE4-1 ;E/D SOMETHING
               266
CE33:89
CE34:8A
               267
                           OFB >STATERR-1 ; ILLEGAL STATE
 CE35: A7
               268
                         DFB >CDONE-1 ;SKIP UNTIL CR
DFB >CDONE-1 ;SKIP UNTIL CR THEN DO CMD
               269
 CE36:89
 CE37:89
                270
                271 ***********
 CE38:
                272 * COMMAND STATE 1 *
                273 ************
 CE38:
 CE38:DD 38 05 274 CSTATE1 CMP CMDBYTE, X ; IS IT < CMD>?
                       BNE CSTATEIA
 CE3B:D0 06
CE3D:DE B8 04
                275
                           DEC STATEFLG, X ; SET STATE BACK TO ZERO
               276
                           JMP ACIAOUT ;OUTPUT <CMD> IF SO
 CE40:4C 02 CC 277
                278 *
 CE43:
 CE43:C9 30
                279 CSTATE1A CMP #$30
                280 BCC CSTATE1B
 CE45:90 0D
                            #$3A ;<#9?
BCS CSTATE1B
 CE47:C9 3A
                281
                282
                                           :IT'S A NUMBER
 CE4B: 29 OF
                283
                            AND #SOF
                            STA PARAMETER, X
 CE4D: 9D 38 04
                284
                            LDA #2
 CE50: A9 02
                285
                            BNE SETOSTATE ; <ALWAYS> SET MODE 2 AND RETURN
 CE52:DO 27
                286
                287 *
 CE54:
                                          ; IS IT A CONTROL CHAR?
                288 CSTATE1B CMP #$20
 CE54:C9 20
                           BCS CSTATE1C
```

CE56: BO 06

```
CE58:9D 38 05 290
                                    STA CMDBYTE, X ; SET NEW COMMAND CHARACTER
The last last last last last
          CE5B:4C 79 CE 291
                                    JMP ZEROSTATE ; RESET STATE TO ZERO
                         292 *
          CE5E: AO 00
                         293 CSTATE1C LDY #0
                                                  ;USE COMMAND TABLE
                         CE60: FO 4D
          CE62:
          CE62:
                         296 * COMMAND STATE 2: ACCUMULATE PARAMETER *
          CE62:
                         297 *************************
          CE62:49 30
                         298 CSTATE2 EOR #$30 ;CONVERT $30-$39 TO 0-9
          CE64: C9 OA
                         299
                                    CMP #$A
                                                  ;0-97
                                    BCS CSTATE2A
          CE66: BO OD
                         300
          CE68: AO OA
                         301
                                    LDY
                                        #SA
                                                  ; IT'S A NUMBER, SO ADD
                         302 ACCLOOP ADC PARAMETER, X ; IT TO 10*PARAMETER
          CE6A:7D 38 04
          CE6D: 88
                         303
                                    DEY
          CE6E: DO FA
                         304
                                    BNE ACCLOOP
                                    STA PARAMETER, X
          CE70:9D 38 04
                         305
          CE73:FO 15
                         306
                                    BEQ CDONE
                                                  ; <ALWAYS>
                         307 *
          CE75:
          CE75: A0 2E
                         308 CSTATE2A LDY #CMDTBL1-CMDTBL ; USE COMMAND TABLE
          CE77:DO 36
                         309
          CE79:
| N | N |
          CE79:
                         311 * SET COMMAND STATE *
          CE79:
                         312 *************
          CE79: A9 00
                         313 ZEROSTATE LDA #0
          CE7B: 85 2A
                         314 SETOSTATE STA ZPTMP1
                                   LDX MSLOT
          CE7D: AE F8 07
                        315
          CE80: BD B8 04
                         316
                                    LDA STATEFLG, X
10
          CE83:29 F8
                         317
                                    AND #$F8
          CE85:05 2A
                         318
                                    ORA ZPTMP1
          CE87: 9D B8 04
                         319
                                    STA STATEFLG, X
- 10
                         320 CDONE RTS
          CEBA: 60
                         321 ****************
          CE8B:
          CE8B:
                         322 * COMMAND STATE 4 (E/D) *
- 10
          CE8B:
                         323 *****************
          CEBB: AS
                         324 CSTATE4 TAY
                                                  ; E/D -> Y-REG
          CE8C: BD 38 04
                                    LDA PARAMETER, X
                         325
- 10
          CEBF: CO 44
                         326
                                    CPY #$44
                                                  ;D(ISABLE)?
          CE91:FO 09
                         327
                                    BEQ CSTATE4A
                                    . #$45 ;E(NABLE)?
BNE STATERR ;IF MOUND
          CE93:CO 45
                         328
                                    CPY #S45
CE95: DO 11
                         329
                                                  ; IF NOT, IGNORE THIS COMMAND
          CE97:1D 38 D7
                        330
                                    ORA MISCFLG, X ; SET FLAG
                                    BNE CSTATE4B ; <ALWAYS>
          CE9A:DO 05
                         331
                         332 CSTATE4A EOR #SFF
          CE9C: 49 FF
                                                  ; INVERT FOR DISABLE
          CE9E:3D 38 07
                        333
                                   AND MISCFLG, X ; RESET FLAG
          CEA1:9D 38 07 334 CSTATE4B STA MISCFLG, X
          CEA4:
                         335 *************
          CEA4:
                         336 * ESCAPE TO STATE 6 *
          CEA4:
                         337 **************
          CEA4: A9 06
                         338 SETSTATE6 LDA #6
          CEA6:DO D3
                         339 BNE SETOSTATE ;<ALWAYS>
340 STATERR LDA #32 ;CODE FOR
                                                  ; CODE FOR BAD COMMAND
          CEAA:9D BB 05
                         341
                             STA STSBYTE, X
          CEAD: DO F5
                                    BNE SETSTATE6 ; < ALWAYS>
          CEAF:
                         343 **********************
          CEAF:
                         344 * TABLE DRIVEN COMMAND PROCESSOR *
                         345 ******************
          CEAF:
          CEAF: B9 EB CC
                        346 CMDSEARCH LDA CMDTBL, Y ; GET CANDIDATE CHARACTER
          CEB2:FO F4
                        347
                                    BEO STATERR ; A ZERO MARKS THE END OF A SUBTABLE
```

-

-

```
CMP ZPTEMP
                                          ; MATCH?
CEB4: C5 35
               348
                           BEQ CMDMATCH
CEB6: PO 05
               349
CEB8: C8
               350
                            INY
                351 CMDSEARCH1 INY
                                           ; REENTRY FOR WRONG MODES
CEBA: C8
               352
                           INY
                                            ; ENTRY LENGTH = 3
               353
                           BNE CMDSEARCH ; < ALWAYS>
CEBB: DO F2
CEBD:
               354 *
CEBD: C8
                355 CMDMATCH INY
CEBE: B9 EB CC
               356
                           LDA CMDTBL, Y
                           STA ZPTMP1
CEC1:85 2A
                357
CEC3:29 20
                358
                            AND
                                          ; CHECK PASCAL ENABLE
                                 #$20
                            BNE CMDMATCH1 ; IT'S ON SO DONT CHECK P-BIT
CEC5: DO 07
                359
                           LDA MISCFLG, X ;OFF SO MAKE SURE
CEC7: BD 38 07
                360
                                          ; THAT WE AREN'T IN PASCAL
CECA: 29 10
                361
                           AND #$10
                           BNE CMDSEARCH1 ; BRANCH IF WE ARE
CECC: DO EB
                362
               363 *
CECE:
CECE: BD 38 07
                364 CMDMATCH1 LDA MISCFLG, X ; GET CIC/PPC BIT
                                          ;SHIFT CIC/PPC MODE BIT TO CARRY
CED1:4A
                           LSR A
                365
                            LSR A
CED2:4A
                366
CED3:24 2A
                367
                            BIT
                                ZPTMP1
                                           ; PPC->N CIC->V
                            BCS CMDMATCH2 ; BRANCH IF CIC MODE
CED5: BO 04
                368
                           BPL CMDSEARCH1 ; NOT OK FOR PPC
BMI CMDEXEC ; AND OK
CED7:10 E0
                369
CED9:30 02
                370
CEDB:50 DC
                371 CMDMATCH2 BVC CMDSEARCH1 ; NOT OK FOR CIC
CEDD:
                                         ; RETRIEVE TABLE MODE BYTE
                373 CMDEXEC LDA ZPTMP1
CEDD: A5 2A
                374
                            PHA
CEDF: 48
CEE0: 29 07
                375
                            AND
                                 #$07
                            JSR SETOSTATE ; SET NEXT STATE
CEE2: 20 7B CE
                376
                377
                            INY
CEE5: CB
CEE6:68
                378
                            PLA
CEE7: 29 10
                379
                            AND
                                 CMDEXEC1 ; IF BIT 4 IS SET, VECTOR TO ROUTINE
CEE9: DO 07
                380
                            BNE
CEEB: B9 EB CC
                381
                            LDA
                                CMDTBL, Y
CEEE:9D 38 04
                382
                            STA
                                 PARAMETER, X
CEF1:60
                383
                            RTS
CEF2:
                384 *
                385 CMDEXEC1 LDA #SCD ; ROUTINES MUST BE IN PAGE SCD
CEF2: A9 CD
CEF4:48
                386
                           PHA
CEF5: B9 EB CC
                            LDA CMDTBL, Y
                387
                            PHA
CEF8:48
                388
CEF9: A4 26
                            LDY SLOTI6
                389
                            LDA PARAMETER, X ; LOT OF ROUTINES NEED THIS
CEFB: BD 38 04
                390
                            RTS
CEFE:60
                391
                392 *
 CEFF:
 CEFF:00
                393
                            DFB $00
SYMBOL TABLE
               SORTED BY SYMBOL
                                                              ?CB9C ACK
                      CE6A ACCLOOP
                                          CCD2 ACTAOUT
                                                              CCB7 ASCREEN
                      C9B5 ADJUST
                                         CB59 ANRTS
 C9C8 ADJRTS
                                         2C93D BATCHIN
                                                               C9EF BATCHIO
 C8B8 BASICEXIT
                        28 BASL
                                                               CD88 BAUDCMD2
                                          CD81 BAUDCMD1
                      CD7D BAUDCMD
 7C941 BATCHOUT
                                                               C8E5 BINEND1
                                          CSEA BINACIA
 C711 BENTRY
                      CREF BINACIA1
                                                               C8CB BINKBD
                                         2C700 BINIT
                      C745 BINITI
 CRDO BINEND
                      C77C BOUTPUT1
                                          C767 BOUTPUT
                                                               C78B BOUTPUT2
 CARF BINPUT
                                                               24 CH
C8B5 CICEXIT
                                          CESA CDONE
                      0688 BUFBYTE
 CDC1 BREAKCMD
    27 CHARACTER
                      CA1E CHECKTERM
                                          0638 CHNBYTE
                                                               C9EB CKINPUT2
  C9EE CIEND
                      C9D1 CKINPUT
                                          C9E5 CKINPUT1
```

```
0538 CMDBYTE
                    CC2C CKKBD
                                        CC34 CKKBD1
CC3D CKKBDXIT
                                                             CEBD CMDMATCH
                                        CECE CMDMATCH1
                    CEDD CMDEXEC
CEP2 CMDEXEC1
                                                             CE1A CMDPROC3
                                        CEOE CMDPROC2
CEDB CMDMATCH2
                    CEOA CMDPROC1
                                                             CEAF CMDSEARCH
CE24 CMDPROC4
                    CDFC CMDPROC
                                        COSA CMDREG
                                                             CCEB CMDTBL
                                        CD19 CMDTBL1
CEB9 CMDSEARCH1
                    CB1A CMDSEQCK
                                                             CD3B CRCMD
                                         FDED COUT
06B8 COLBYTE
                    C917 COMMA
                                                             CE38 CSTATE
                                         CESE CSTATEIC
                    CE54 CSTATE1B
CE43 CSTATE1A
                                                             CEA1 CSTATE4B
                                         CE9C CSTATE4A
CE75 CSTATE2A
                    CE62 CSTATE2
                                                             COSB CTLREG
                                          36 CSWL
                      37 CSWH
CE8B CSTATE4
                                         CDA1 DATACMD1
                                                             CD9C DATACMD
                    C9C9 CTRLTST
CDSE CTLREGSET
                                         CD4B DELAYSET
                                                             CO81 DIPSW1
C9CO DECRCOL
                    03B8 DELAYFLG
                                         CB2E ESCCHECK
                                                             CB90 ETX
CO82 DIPSW2
                    CBFF DLYTBL
                                                             C751 FROMOUT
                                         C754 PROMIN
                    C968 FORCECR
CD47 FFCMD
                                         CC72 GETCMD
                                                             CC3E GETKBD
                    CRAA GETCHAR
CRRA GETCHART
                                                             0438 HANDSHKE
                    CC91 GETKBDONE
                                         CCDF GETXLATE
CC44 GETKBD1
                                         0200 INBUFF
                                                             CROS INIT1
                    C705 IENTRY
CDD4 ICMD
                                                             C83F INIT2B
                                         C83C INIT2A
C827 INITIA
                    C835 INIT2
                                                            2C879 INITACIA
                                         C872 INITS
C857 INIT3
                    C864 INIT4
                                                             CAFF INPUT
                    C88F INITACIA2
C882 INITACIA1
                                         CB16 INPUT2
                                                              CC5C KBDESC
FF58 IORTS
                    CO10 KBDSTRB
                                         C000 KBD
                                                               39 KSWH
                                         CDEO KCMD
                    CDE5 KCMD2
CDE2 KCMD1
                                                              CBEA LFGEN
                    CCD3 LCMASK
                                         CD41 LFCMD
  38 KSWL
                                                              07F8 MSLOT
                                         C9FD MOVOUT
0738 MISCFLG
                    CAOC MOVIN
                                                              CB18 NOINPUT
                                         CB19 NOINPUT1
CDE9 NCMD
                    CB58 NOCMD
                                                              C951 NOTAB
                                         C954 NOTAB1
CCB6 NOOUT
                    C75C NORMIO
                                                              CBE2 OUTDLYLP
                                         CBD9 OUTDLY1
                     C707 OENTRY
FCBA NXTA1
                                                              CBC1 OUTPUT3
CB68 OUTPUT1
                     CB6B OUTPUT2
                                         CB63 OUTPUT
                                                              CBA9 P8AOUT3
                                         CBA6 P8AOUT2
                    ?CB76 P8AOUT1
CBFE OUTPUTEND
                                         CD9B PARITYCMD
                                                              C800 PASCALINIT
                     0438 PARAMETER
CBBB P8AOUT4
                     C89B PASCALREAD
                                         C9AA PASCALWRITE
                                                              CSA3 PASEXIT
CR9E PASCALREAD1
                                        7C84D PREADO
                                                              C794 PREAD
                     C78E PINIT
COOR PENTRY
                                                              C79A PSTATUS
                     CC5E PROMPTLOOP
                                         C7A8 PSTATIN
CC93 PROMPTBL
                                                              C797 PWRITE
                     0638 PWDBYTE
                                         C9A6 PWDTBL
C7AB PSTATUS 2
                                                              COS9 RESET
CDBD QCMD1
                     CDB9 QUITCMD
                                         COSS RDREG
                                                              C7EE RESTORHOOK
                                         CC29 RESTOREND
CDAD RESETCMD
                     CC11 RESTORE
                                                              CFFF ROMSOFF
                                           4E RNDL
CCDB REVMASK
                      4F RNDH
                                                              CCA3 SCREENOUT1
                                         CC9E SCREENOUT
CD57 ROTATE
                     C7B2 SAVEHOOK
                                                              CBFC SEROUT
C7BO SENDCD
                     C998 SEREND2
                                         C97A SEREND
                                         CE7B SETOSTATE
                                                              FE93 SETSCR
                     FE89 SETKBD
 C996 SETCH
                                                              CAEB SRIN2
                                         CAE9 SRIN1
CEA4 SETSTATE6
                       26 SLOT16
                                         CAF5 SROUT
                                                              CD7C SSLOTCMD1
                     CAF4 SRIN3
 CAD2 SRIN
                                                              CEAS STATERR
                                         04B8 STATEFLG
 CD61 SSLOTCMD
                     0100 STACK
                                          05B8 STSBYTE
                                                              C934 TAB1
                     COSS STREG
 CE30 STATETBL
                                         CO88 TOREG
                                                              CASS TERMACIAIN
                     C921 TABCHECK
 C948 TAB2
                                                             ?CA4C TERMEXIT
 CAAD TERMCAP1
                     CA9B TERMCAP
                                         CDA7 TERMCMD
                                                              CAS7 TERMLETTER
                     CA81 TERMINC
                                          CA66 TERMKBDIN
 CA82 TERMINC1
                                                              CA31 TERMNEXT1
                    ?CA23 TERMMODE
                                          CA2B TERMNEXT
 CAB1 TERMLOCK
                                         PCA7D TERMNORM
                                                              CA54 TERMRTS
 CA41 TERMNEXT2
                     CA47 TERMNEXT3
                                                              CCC3 TOSCREEN
                                         CCC6 TESTLETTER
 CA93 TERMSEND
                     CA95 TERMS END1
                                                              FDF6 VIDOUT
                     CAB9 TRANSLATE
                                          CCD7 UCMASK
 CD35 TRANCMD
                                                              CB5A XONWAIT
                                          CB3C XOFFCk
 CAC4 WAITMS
                     CAC6 WAITMS1
                                                                35 ZPTEMP
                                          CE79 ZEROSTATE
                     CDF4 ZCMD
 CDFB ZCMDRTS
   2A ZPTMP1
                       2B ZPTMP2
                SORTED BY ADDRESS
SYMBOL TABLE
                                                                28 BASL
                                            27 CHARACTER
   24 CH
                        26 SLOT16
                                                                36 CSWL
                                            35 ZPTEMP
                        2B ZPTMP2
   2A ZPTMP1
                                                                3C A1L
                                            39 KSWH
   37 CSWH
                        38 KSWL
                                          0100 STACK
                                                              0200 INBUFF
   4E RNDL
                       4F RNDH
                                                              04B8 STATEFLG
                                          0438 PARAMETER
```

-08

-100

100

-/88

In. A

District Ball

- K

0388 DELAYFLG

0438 HANDSHKE

0538	CMDBYTE	0588	STSBYTE	0638	PWDBYTE	0639	CHNBYTE
0688	COLBYTE		BUFBYTE		MISCFLG		MSLOT
C000	KBD		KBDSTRB		DIPSW1		DIPSW2
2C088	TOREG		RDREG		STREG		RESET
	CMDREG		CTLREG				
	OENTRY		BENTRY		BINIT		IENTRY
	PROMIN		NORMIO		BINIT!		FROMOUT
	BOUTPUT2		PINIT		BOUTPUT		BOUTPUT1
	PSTATUS				PREAD		PWRITE
	SAVEHOOK		PSTATIN		PSTATUS 2		SENDCD
	INIT1A		RESTORHOOK		PASCALINIT		INITI
	PREADO		INIT2		INIT2A		INIT2B
			INIT3		INIT4		INIT5
	INITACIA		INITACIAT		INITACIA2		PASCALREAD
	PASCALREAD1		PASEXIT		GETCHAR	C8B4	GETCHAR1
	CICEXIT		BASICEXIT		BINPUT	CBCB	BINKBD
	BINEND		BINEND1		BINACIA	CSEF	BINACIA1
	SEROUT		COMMA	C921	TABCHECK	C934	TAB1
?C93D	BATCHIN	?C941	BATCHOUT	C948	TAB2	C951	NOTAB
C954	NOTAB1	C968	FORCECR	C97A	SEREND	C996	SETCH
C998	SEREND2	C99B	PENTRY	C9A6	PWDTBL	C9AA	PASCALWRITE
	ADJUST	C9C0	DECRCOL	C9C8	ADJRTS	C9C9	CTRLTST
	CKINPUT	C9E5	CKINPUT1	C9EB	CKINPUT2		CIEND
C9EF	BATCHIO	C9FD	MOVOUT	CAOC	MOVIN	CALE	CHECKTERM
?CA23	TERMMODE	CA2B	TERMNEXT	CA31	TERMNEXT1	CA41	TERMNEXT2
CA47	TERMNEXT3	?CA4C	TERMEXIT		TERMRTS		TERMACIAIN
CA66	TERMKBDIN	?CA7D	TERMNORM		TERMINC		TERMINC1
CA87	TERMLETTER	CA93	TERMSEND		TERMS END1		TERMCAP
CAAD	TERMCAP1	CAB1	TERMLOCK		TRANSLATE		WAITMS
CAC6	WAITMS1	CAD2			SRIN1		SRIN2
CAF4	SRIN3		SROUT		INPUT		INPUT2
	NOINPUT		NOINPUT1		CMDSEOCK	1000	
	XOFFCK		NOCMD		ANRTS		ESCCHECK
	OUTPUT		OUTPUT1		OUTPUT2		XONWAIT
CB90		?CB9C			P8AOUT2		PBAOUT1
	PSAOUT4						P8AOUT3
	LFGEN		OUTPUT3		OUTDLY1		OUTDLYLP
	RESTORE		OUTPUTEND		DLYTBL		ACIAOUT
	CKKBDXIT		RESTOREND		CKKBD		CKKBD1
	PROMPTLOOP		GETKBD		GETKBD1		KBDESC
	SCREENOUT		GETCMD		GETKBDONE		PROMPTBL
	TOSCREEN		SCREENOUT1		NOOUT		ASCREEN
	REVMASK		TESTLETTER		LCMASK		UCMASK
			GETXLATE		CMDTBL	CD19	CMDTBL1
	TRANCMD		CRCMD		LFCMD	CD47	FFCMD
	DELAYSET		ROTATE	CD61	SSLOTCMD	CD7C	SSLOTCMD1
	BAUDCMD		BAUDCMD1	CD88	BAUDCMD2	CDSE	CTLREGSET
	PARITYCMD		DATACMD		DATACMD1	CDA7	TERMCMD
	RESETCMD		QUITCMD	CDBD	QCMD1	CDC1	BREAKCMD
CDD4		CDEO		CDE2	KCMD1	CDE5	KCMD2
CDE9		CDF 4		CDFB	ZCMDRTS	CDFC	CMDPROC
	CMDPROC1		CMDPROC2		CMDPROC3	CE24	CMDPROC4
	STATETBL		CSTATE1		CSTATE1A		CSTATE1B
	CSTATE1C		CSTATE2		ACCLOOP	CE75	CSTATE2A
	ZEROSTATE		SETOSTATE	CE8A	CDONE	CE88	CSTATE4
	CSTATE4A		CSTATE4B		SETSTATE6	CEA8	STATERR
	CMDSEARCH		CMDS EARCH1	CEBD	CMDMATCH	CECE	CMDMATCH1
	CMDMATCH2	CEDD	CMDEXEC	CEF 2	CMDEXEC1	CFFF	ROMSOFF
	NXTA1	FDED		PDF6	VIDOUT	FE89	SETKBD
FE93	SETSCR	FF58	IORTS				

# APPLE INTERFACE CARD EMULATION

- 64

- Est

15

-8

The SSC emulates both the P8 and the P8A versions of the Apple II Serial Interface Card (SIC), although the SSC is not completely POKE-compatible with either. In addition, the SSC supports several Apple II Communications Card and Parallel Card software commands.

# OLD SERIAL INTERFACE CARD EMULATION

The SSC replaces the P8 and P8A versions of the Apple II Serial Interface Card (SIC) and it has two switch-selectable modes to emulate them, as explained below. However, because of firmware space limitations, the SSC does not support all functions of the older interface cards, and various POKE locations are different. This section explains these functional differences.

It is best to use Printer Mode rather than one of the emulation modes, except under these circumstances:

- if you have extensive existent applications that use PEEKs and POKEs to modify SIC operating characteristics
- if you need SIC P8A mode's ETX/ACK (or other-character/ACK) handshaking capabilities

What the SSC does NOT support that the old SIC does:

- P8 SIC block moves
- baud rates other than the 15 listed in the various baud rate tables in this manual (ACIA hardware generates only those 15)
- data formats other than 5 8 data bits and 1, 1-1/2 or 2 stop bits (ACIA characteristic; other formats rarely used anyway)
- <ESC>U and <ESC>L commands for upper and lowercase (but SSC's Translate command offers more options; POKEs also available)
- current-loop operation

To run the SSC in emulation of the old Apple II Serial Interface Card (SIC), prepare and install the SSC the same way as for Printer Mode (Chapters 1 and 2), with the following exceptions:

- Set mode switches SW1-5 ON and SW1-6 OFF to emulate the old SIC with a P8 ROM.
- Set mode switches SW1-5 OFF and SW1-6 OFF to emulate the old SIC with a P8A ROM.
- Install the SSC in whatever slot the old SIC was installed in for the application involved.
- Follow the instructions given in the next sections if the application program did PEEKs and POKEs.

### **P8 EMULATION POKES**

Changing SIC parameters was done either by setting the seven switches located on the card, or by POKEing the SIC slot RAM locations where this configuration data was stored. BASIC programs that talked through the old SIC may be used with the new SSC; however, if the program POKEs at these slot RAM locations, those POKEs must be changed to be compatible with the SSC's use of the RAM. The P8 and P8A ROMs differ slightly in their use of these RAM locations. Tables B-1 and B-2 show the transformation for P8 mode; additional differences for P8A mode are noted in the following section. Other POKE possibilities are described in Appendix A.

In the tables, the letter s stands for the slot number (1-7) in which the SSC is installed; the other letters are used as variables whose values are noted in the table (sometimes further down).

There is no claim that making these changes is simple. In fact, whenever possible it is best to use Printer Mode and its software commands to change SSC operating variables.

Here is an example of how to use the tables: let's say that the SSC is in slot #3. You want: a baud rate of 110; data format of 5 data bits and 2 stop bits, even parity; line width of 40 with video on, no automatic  $\langle LF \rangle$  after  $\langle CR \rangle$ ; no translation of lowercase to uppercase; and no 1/4-second delay after  $\langle CR \rangle$ . The PEEKs and POKEs:

```
POKE 49339, 243 (49291 + 3*16; 3 + 240)

POKE 49338, 107 (49290 + 3*16; p = 107)

POKE 2043, 132 (plug in magic number)

POKE 1147, 64 (plug in magic number)
```

The same thing in Printer Mode with appropriate switch settings is:

```
SW1-1 to SW1-7: ON ON OFF OFF OFF ON ON SW2-1 to SW2-7: -- OFF ON ON OFF OFF OFF
```

Then to set 5 data and 2 stop bits, use <CTRL-I>7D<RETURN>; for even parity, use <CTRL-I>3P<RETURN>; to leave lowercase alone, use <CTRL-I>1T<RETURN>. You can use commands to change baud rate, etc.

	SSC switches	PEEKs and POKES to use for				
Selection	and settings	P8 Serial Card	Super Serial Card			
P8 Mode: P8A Mode:	SW1-5 ON, SW1-6 OFF SW1-5 OFF, SW1-6 OFF					
50 75 110 135 150 300 600 1200 1200 1800 4800 4800 7200	SW1-1 to SW1-4 same as Printer Mode	POKE 1144+s,r r = (not available) Ø dec/\$ØØ hex 176 dec/\$BØ hex 144 dec/\$9Ø hex 128 dec/\$8Ø hex 64 dec/\$4Ø hex 32 dec/\$2Ø hex 16 dec/\$1Ø hex 11 dec/\$ØB hex 5 dec/\$Ø5 hex 4 dec/\$Ø2 hex (not available) 2 dec/\$Ø2 hex 1 dec/\$Ø1 hex	POKE 49291+s*16,r r = b + d; b = 1 dec/\$\%1 hex 2 dec/\$\%2 hex 3 dec/\$\%3 hex 4 dec/\$\%4 hex 5 dec/\$\%5 hex 6 dec/\$\%6 hex 7 dec/\$\%7 hex 8 dec/\$\%8 hex 9 dec/\$\%9 hex 1\% dec/\$\%9 hex 11 dec/\$\%0 hex 12 dec/\$\%0 hex 13 dec/\$\%0 hex 14 dec/\$\%0 hex 15 dec/\$\%0 hex			
Data Format:  8 data,1 stop 7 data,1 stop 6 data,1 stop 5 data,1 stop 8 data,2 stop 7 data,2 stop 6 data,2 stop 6 data,2 stop 5 data,2 stop	SW2-1 ON SW2-1 OFF	POKE 1912+s,r POKE 1272+s,t r = 9; t = 1* r = 8; t = 1* r = 7; t = 1* r = 6; t = 1* r = 9; t = 2* r = 8; t = 2* r = 7; t = 2* r = 6; t = 2* add 1 if p = 1 or Ø	(to get r above, add d to b) d = 16 dec/\$10 hex 48 dec/#30 hex 80 dec/\$50 hex 112 dec/\$70 hex 144 dec/\$90 hex 176 dec/\$80 hex 208 dec/\$D0 hex 240 dec/\$F0 hex			
Parity: none odd even MARK SPACE		POKE 1400+s,p p = 2 p = 1 p = 0 (not available) (not available)	POKE 4929Ø+s*16,1 p = 11 (\$ØB hex; p = 43 (\$2B hex; p = 107 (\$6B hex; (not available) (not available)			

-/E

Table B-1. SIC Switch Settings, PEEKs and POKEs, Part I

	SSC switches and settings	PEEKs and POK	ES to use for	
Selection		P8 Serial Card	Super Serial Card	
Line Width:	SW2-3 & SW2-4, same as Printer Mode	POKE 1784+s,r r=1 to 255; for no ⟨CR⟩,r=∅	POKE 1784+s,r r=40 to 255; for no <cr>, PEEK 1400+s, POKE 1400+s, (old value + 128)</cr>	
Video/ Generate <lf>/ Translate/ <cr> Delay:</cr></lf>	SW2-3 & SW2-4 SW2-5 (no switch) SW2-2 (all switches same as in Printer Mode)	V = Video on? G = Gen. < LF>? T = LC to UC? D = Dly 1/4 s? POKE 2040+s,r r=  dec hex V G T Y Y 5 \$05 Y Y Y Y 36 \$24 Y N N Y 37 \$25 Y Y N Y 68 \$44 Y N Y N 100 \$64 Y N N N 101 \$65 Y Y N N 132 \$84 N N Y Y 133 \$85 N Y Y Y 164 \$A4 N N N Y 165 \$A5 N Y N Y 196 \$C4 N N Y N 197 \$C5 N Y Y N 228 \$E4 N N N 229 \$E5 N Y N N	V = Video on? G = Gen. <lf>? POKE 2040+s,r r= dec hex V G 4 \$04 N N 5 \$05 N Y 132 \$84 Y N 133 \$85 Y Y T = LC to UC? D = Dly 1/4 s? POKE 1144+s,r r = dec hex T D 0 \$00 Y N 16 \$10 Y Y 64 \$40 N N 80 \$50 N Y</lf>	

Table B-2. SIC Switch Settings, PEEKs and POKEs, Part II

## **P8A EMULATION POKES**

The P8A ROM differs from the P8 ROM in several ways:

- 1) The  $\langle \text{CR} \rangle$  delay switch now determines whether an ETX/ACK handshake is performed after each  $\langle \text{CR} \rangle$  that is transmitted. The corresponding RAM bit was not the same as the P8  $\langle \text{CR} \rangle$  delay bit, but was kept in bit 2 of location 1400+s. For SSC emulation, the control is the same as the  $\langle \text{CR} \rangle$  delay bit as noted above (in location 1144+s).
- 2) The number of stop bits was always 2; for SSC P8A mode this is configured via switch SW2-1 and can also be set via software by POKEing location 4929 as noted above.
- 3) The printer width information was kept in the same location that the P8 ROM kept the number of stop bits; the P8 printer width byte was zeroed to avoid automatic generation of carriage returns. The SSC P8A emulation code keeps the printer width information in the

same place as for P8 emulation and uses the high-order bit at location 1400+s to control automatic generation of carriage returns.

- 4) Lowercase input is enabled by default for the P8A ROM; in P8A emulation, however, it is enabled by the POKE shown in Table B-2.
- 5) In contrast to the P8 ROM, the P8A ROM and the SSC do not support batch moves.

41 A1 A1

| N | N |

-

-

- 1

N. A.

-

1

-

B

-

A LA LA

-45

- 100

100

6) The enquire character for the SIC P8A ROM was ETX (ASCII 3); for SSC P8A mode, this can be changed to another control character by a POKE to location 1400+s. For example, to change the enquire character to ENQ (ASCII 5), which is used by many RS-232 devices, use this POKE: POKE 1400+s,5. Note that this also disables the automatic generation of carriage returns. Actually, any character between  $\emptyset$  and 31 can be used, although only 3 and 5 are used much.

### OTHER EMULATION MODE DIFFERENCES

If your old programs, written to control one of the old Serial Interface Card ROMs, still don't work after you've followed all this handy advice, then read on.

The SSC always monitors the RS-232-C handshake lines to determine whether or not the device is ready to accept data. If your device fails to assert one of these lines, the SSC will wait patiently forever.

When the arrow on the jumper block is pointing toward TERMINAL, your device sees DCD and DSR asserted as soon as the SSC is initialized, and the SSC sees CTS whenever the device sends RTS. If the device does not assert both RTS and DTR, the SSC will assume it is not ready to receive data. This can be used as a hardware handshake to prevent buffer overflow at the device (e.g., when your printer runs out of paper it can stop asserting one of these lines and the SSC will wait while you put in more paper). If you do not connect these lines, the SSC will always treat them as if they were asserted.

The Serial Interface Card tied RTS to CTS, and DTR to DCD and DSR; if your RS-232 device depended upon this, you may want to make a special connector which does this.

Your device may have depended upon the half-duplex nature of the SIC. The ACIA on the SSC is able to send and receive at the same time and is always configured to do so.

The SIC was initialized each time it was called at location CSOO (for example, by a PR#s or IN#s). The SSC is only reintialized after the ACIA has been reset (either by resetting the Apple or by exiting from Printer or Communication Mode via a Reset command).

# OLD COMMUNICATIONS CARD COMMANDS

The SSC supports all the functions supported by the old Apple II Communications Interface Card (CIC), although the two ACIAs' registers are not the same on a bit-by-bit level. The SSC also supports the CIC commands: <CTRL-T>, <CTRL-R>, and <CTRL-S>.

### SWITCH TO TERMINAL MODE—(CTRL-T)

In Communication Mode, the SSC is initialized to recognize the remote-control command <CTRL-T> arriving in the stream of incoming data. This character causes the SSC to enter Terminal Mode (the same as the T(erminal command (Chapter 3). You can disable <CTRL-T> recognition by issuing an X(OFF D(isable command.

### BYPASS TERMINAL MODE—(CTRL-R)

When the SSC is in Terminal Mode and X(OFF E(nable (the default in this mode) is in effect, the SSC recognizes the remote control command <CTRL-R> arriving in the input data stream, and responds by bypassing (exiting from) Terminal Mode. This is the same as the Q(uit Terminal Mode command (Chapter 3).

### XOFF-(CTRL-S)

The SSC interprets  $\langle \text{CTRL-S} \rangle$  as the ASCII XOFF character. When it receives  $\langle \text{CTRL-S} \rangle$  from a remote device, it stops transmitting data until it receives an XON character from that device.

# PARALLEL CARD COMMANDS

The SSC is not hardware compatible with the Apple II Parallel Cards. However, for the sake of compatibility with software written for parallel interface applications, the SSC supports the following commands. You do not need to follow these commands with <RETURN>.

### LINE WIDTH n AND VIDEO OFF-(CTRL-I)(n)N

This command turns off the Apple II video screen and generates a  $\langle CR \rangle$  after n characters (if automatic  $\langle CR \rangle$  generation is enabled via the C command (Chapter 2); n can be any value from 40 through 255.

# LINE WIDTH 40 AND VIDEO ON-(CTRL-I)I

This command turns on the Apple II video screen and sets the line width to  $4\emptyset$ .

### DISABLE AUTOMATIC LINEFEED-(CTRL-I)K

This command has the same effect as L(inefeed D(isable (Chapter 2): it turns off automatic generation of  $\langle LF \rangle$  after  $\langle CR \rangle$ .

# APPENDIX C SPECIFICATIONS AND SCHEMATICS

This appendix contains the SSC specifications, connector pin assignments, jumper block wiring, and a schematic diagram. Use the schematic diagram with the Theory of Operation section in Chapter 4.

## SSC SPECIFICATIONS

PHYSICAL CHARACTERISTICS

Dimensions Weight

Cables required

Controls

Special Tools

ENVIRONMENT

Operating temperature Storage temperature Operating relative humidity Storage relative humidity 40°F to 95°F (5°C to 35°C) -40°F to 122°F (-40°C to 50°C) 5% to 95% (noncondensing) 5% to 95% (noncondensing)

2-3/4" x 7" (68.8 mm x 177.8 mm)

3 oz. (90 gm), approximately internal cable from 10-pin header

on SSC to DB-25 connector on case of Apple II (supplied); shielded RS-232-C cable to external device

2 blocks of 7 switches each, set

by user before installation

SPECIAL CIRCUITS

SY6551 2316 Asynchronous Communications Interface Adapter Read Only Memory (2, $\emptyset$ 48 by 8 bits) with SSC firmware The SSC has the usual power supply bypassing capacitors

(not supplied)

none required

### APPLE II SLOT LOCATION

BASIC programs	any slot except slot #Ø
APPLESOFT programs	any slot except slot #Ø
PASCAL programs	slot #1 for use with printer, etc.
Contract Description	slot #2 for use with modem
	slot #3 for use with terminal

1

### SOFTWARE COMPATIBILITY

The SSC is compatible with the following languages and operating systems:

Integer BASIC	DOS 3.2	Pascal 1.0	6502 Assembler
Applesoft BASIC	DOS 3.3	Pascal 1.1	

Under BASIC, input sent to the SSC at high baud rates may be lost, since the SSC can only buffer two characters at a time and BASIC may not be fast enough to read characters before they are overlaid.

In any software environment, characters may be lost when sent to the video screen in scrolling mode at greater than 300 baud. There are at least three solutions to this problem: lower the baud rate to 300 baud; reduce the scrolling window size (using 2 fewer lines already makes 1200 baud possible), or use an 80-column card with automatic hardware scrolling.

# **CONNECTOR PIN ASSIGNMENTS**

Table C-1 lists the signals assigned to the connector pins on the 10-pin header at location 7B on the SSC, and the corresponding pins on the DB-25 connector that you attach to the back of the Apple II case.

10-pin Header	DB-25 Connector	Signal name	3-25
1	1	Frame Ground	1
2	2	Transmit Data (TXD)	. 14
3	3	Receive Data (RXD)	
4	4	Request To Send (RTS)	•
5	5	Clear To Send (CTS)	
6	6	Data Set Ready (DSR)	
7	19	Secondary Clear To Send (SCTS)	•
8	7	Signal Ground .	:
9	20	Data Terminal Ready (DTR)	
1Ø	8	Data Carrier Detect (DCD) 13	)25

Table C-1. Connector Pin Assignments

#### JUMPER BLOCK WIRING

Table C-2 lists the signals that the jumper block connects to the SSC when the arrow points toward the word MODEM and when it points toward the word TERMINAL. In the latter case, the jumper block acts as a modem eliminator.

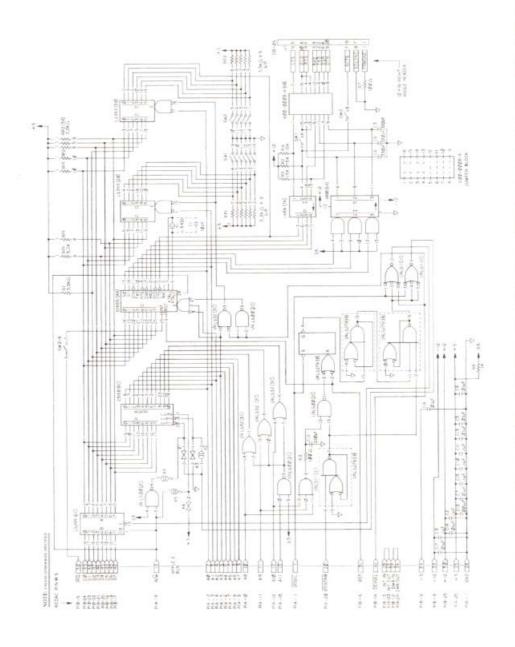
Note that all RS-232-C signals on the SSC use negative-true logic; that is, they are true (asserted) at Ø volts and false at +5 volts.

Signal at SSC	MODEM position (pin)	TERMINAL position (pin)
Transmit Data	Transmit Data (2)	Receive Data (3)
Receive Data	Receive Data (3)	Transmit Data (2)
Request To Send	Request To Send (4)	Data Carrier Detect (8)
Clear To Send	Clear To Send (5)	Data Carrier Detect (8)
Data Set Ready	Data Set Ready (6)	Data Terminal Ready (20)
Data Terminal Ready	Data Term. Ready (20)	
Data Carrier Detect	Data Carrier Detect (8)	
Data Carrier Detect	Data Carrier Detect (8)	

\*When SW1-7 is OFF and SW2-7 is ON, the jumper block in the TERMINAL position connects Data Carrier Detect on the SSC to Secondary Clear To Send on the DB-25 connector.

Table C-2. Jumper Block Wiring

## **SCHEMATIC DIAGRAM**



# ASCII CODE TABLE

The table below shows the entire ASCII character set, and how to generate each character. Not all characters are available directly from the Apple II keyboard. However, in Terminal Mode (Chapter 3) you can generate all of the lowercase and special ASCII characters not accessible directly from the Apple II keyboard.

Here is how to interpret this table:

- The BINARY column has the 7-bit code for each ASCII character.
- The LOW DEC column gives the decimal equivalent of the 7-bit binary value. This value is the same if the binary code has 8 bits and the high-order bit is Ø (SPACE parity; Pascal).
- The LOW HEX column gives the corresponding hexadecimal value.
- The HI DEC column gives the decimal equivalent of the 7-bit binary value if a high-order bit equal to 1 is appended to it (MARK parity; BASIC); for example, 11001000 for the letter H.
- The HI HEX column gives the corresponding hexadecimal value.
- · The ASCII CHAR column gives the ASCII character name.
- The INTERPRETATION column spells out the meaning of special symbols and abbreviations where necessary.
- The WHAT TO TYPE column indicates what keystrokes generate the ASCII character from the NORMAL (unaided) Apple II keyboard, and from the TERMINAL Mode (firmware assisted) keyboard. Characters not accessible are labeled "n/a." The numbers between columns refer to footnotes.
- Angle brackets enclose the names of single keys (like <ESC> for the ESC key), or enclose keystrokes involving more than one key (like <CTRL-SHIFT-M>, which means "hold down CTRL and SHIFT while pressing M.") But <ESC>9 means "type ESC, THEN type 9" because the 9 is outside the angle brackets.

To put the SSC in Terminal Mode, set SW1-5 and SW1-6 both ON; then use the T command or the remote-control <CTRL-T> command. When the SSC first enters Terminal Mode, the keyboard is locked in uppercase. Press <ESC> once for lowercase. This also prepares the SSC for the special <ESC>-plus-number keystrokes. Press <ESC> twice in a row to lock the keyboard in uppercase again.

Bio

The same

No.

Bild

7-BIT	LOW	LOW	HI	HI	ASCII			TY PE
BINARY	DEC	HEX	DEC	HEX	CHAR	INTERPRETATION	NORMAL	TERMINAL
9999999	Ø	ØØ	128	8Ø	NUL	Blank (null)	<ctrl-@></ctrl-@>	
00000001	1	Ø1	129	81	SOH	Start of Header	CTRL-G>	1
00000010	2	02	130	82	STX	Start of Text	<ctrl-b></ctrl-b>	1
0000011	3	Ø3	131	83	ETX	End of Text	<ctrl-c></ctrl-c>	2
0000100	4	04	132	84	EOT	End of Transm.	(CTRL-D)	6
0000101	5	05	133	85	ENQ	Enquiry	CTRL-E>	3
0000110	6	06	134	86	ACK	Acknowledge	<ctrl-f></ctrl-f>	4
0000111	7	Ø7	135	87	BEL	Bell	(CTRL-G)	77
0001000	8	Ø8	136	88	BS	Backspace	(CTRL-II)	5
0001001	9	09	137	89	HT	Horizontal Tab	(CTRL-I)	6
0001010	10	ØA	138	8A	LF	Linefeed	(CTRL-J)	Ü
0001011	11	ØB	139	8B	VT	Vertical Tab	CTRL-K>	
0001100	12	ØC	140	8C	FF	Form Feed	(CTRL-L)	
0001101	13	ØD	141	8D	CR	Carriage Return	CTRL-M>	7
0001110	14	ØE	142	8E	SO	Shift Out	CTRL-N>	1
ØØØ1111	15	ØF	143	8F	SI	Shift In	CTRL-0>	
0010000	16	10	144	90	DLE	Data Link Escape		
0010001	17	11	145	91	DC1	Device Control 1		8
0010010	18	12	146	92	DG2	Device Control 2		9
0010011	19	13	147	93	DC3	Device Control 3	(CTRL-S)	10
0010100	20	14	148	94	DC4	Device Control 4		11
0010101	21	15	149	95	NAK	Neg. Acknowledge		12
0010110	22	16	150	96	SYN	Synchronization	<ctrl-v></ctrl-v>	**
ØØ1Ø111	23	17	151	97	ETB	End of Text Blk.		
0011000	24	18	152	98	CAN	Cancel	CTRL-X>	
0011001	25	19	153	99	EM	End of Medium	<ctrl-y></ctrl-y>	
0011010	26	1A	154	9A	SUB	Substitute	<ctrl-z></ctrl-z>	
ØØ11Ø11	27	1 B	155	9B	ESC	Escape	(ESC)	13 <esc>Ø</esc>

- 1. Normal command character in Communication Mode.
- Used in ETX/ACK protocol (SIC P8A Emulation Mode).
- 3. Used in ENQ/ACK protocol (SIC P8A Emulation Mode).
- 4. Used in ETX/ACK or ENQ/ACK protocol (SIC P8A Emulation Mode).
- Or use ← key.
- 6. Normal Command character in Printer Mode.
- 7. Or use <RETURN> key.
- 8. XON in XON/XOFF protocol (usually in Communication Mode).
- 9. Remote-control command to Exit from Terminal Mode.
- 10. XOFF in XON/XOFF protocol (usually in Communication Mode).
- 11. Remote-control command to Enter Terminal Mode.
- 12. Or use → key.
- 13. Use the ESC key to generate the Escape character with the normal Apple II keyboard. In Terminal Mode, use <ESC>Ø.

7-BIT BINARY	LOW	LOW	HI DEC	HI HEX	ASCI I CHAR	INTERPRETATION	WHAT TO TYPE NORMAL TERMINAL
ØØ111ØØ	28	1C	156	9C	FS	File Separator	n/a <esc>1</esc>
0011101	29	1 D	157	9D	GS	Croup Separator	<ctrl-shift-m></ctrl-shift-m>
ØØ1111Ø	3Ø	1E	158	9E	RS	Record Separator	<ctrl-shift-n></ctrl-shift-n>
ØØ11111	31	1 F	159	9F	US	Unit Separator	n/a <esc>2</esc>
ØlØØØØØ	32	20	16Ø	AØ	SP	Space	spacebar
0100001	33	21	161	A1	1	37/3-038	!
Ø1ØØØ1Ø	34	22	162	A2	11		11
0100011	35	23	163	A3	Ø.		#
0100100	36	24	164	A4	\$		\$ %
0100101	37	25	165	A5	%		%
0100110	38	26	166	A6	δr		&
Ø1ØØ111	39	27	167	A7		Closing Quote	
0101000	40	28	168	A8	(		(
0101001	41	29	169	A9	)		)
0101010	42	2A	170	AA	*		*
Ø1Ø1Ø11	43	2B	171	AB	+		+
0101100	44	2C	172	AC	,	Comma	,
0101101	45	2D	173	AD	_	Hyphen	<u> </u>
Ø1Ø111Ø	46	2E	174	AE		Period	
0101111	47	2F	175	AF	/		/
Ø11ØØØØ	48	30	176	BØ	Ø		Ø
Ø11ØØØ1	49	31	177	B1	1		1
Ø11ØØ1Ø	50	32	178	B2	2		2
Ø11ØØ11	51	33	179	В3	3		3
Ø11Ø1ØØ	52	34	18Ø	B4	4		4
Ø11Ø1Ø1	53	35	181	B5	5		5
Ø11Ø11Ø	54	36	182	В6	6		6
Ø11Ø111	55	37	183	B7	7		7
0111000	56	38	184	B8	8		8
Ø111ØØ1	57	39	185	В9	9		9
Ø111Ø1Ø	58	3A	186	BA	:		:
Ø111Ø11	59	3B	187	BB	;		;
Ø1111ØØ	60	3C	188	BC	<		<
Ø1111Ø1	61	3D	189	BD	=		=
Ø11111Ø	62	3E	190	BE	>		>
Ø111111	63	3F	191	BF	?		?
1000000	64	40	192	CØ	0		6
1000001	65	41	193	C1	A		A
1000010	66	42	194	C2	В		В
1000011	67	43	195	C3	C		C
1000100	68	44	196	C4	D		D
1000101	69	45	197	C5	E		E
1000110	70	46	198	C6	F		F
1000111	71	47	199	C7	G		G
1001000	72	48	200	C8	H		H
1001001	73	49	201	C9	I		I
1001010	74	4A	202	CA	J		J
1001011	75	4B	203	CB	K		K
1001100	76	4C	204	CC	L		L
1001101	77	4D	205	CD	M		M
1001110	78	4E	206	CE	N		N

-11

7-BIT	LOW	LOW	HI	HI	ASCII		WHAT TO T	
BINARY	DEC	HEX	DEC	HEX	CHAR	INTERPRETATION	NORMAL	TERMINAL
1001111	79	4F	207	CF	0		0	
1001111	8Ø	5Ø	208	DØ	P		P	
	81	51	209	D1	Q		Q	
1010001	82	52	210	D2	R		R	
1010010	83	53	211	D3	S		S	
1010011	84	54	212	D4	T		T	
1010100	85	55	213	D5	Ü		U	
1010101	86	56	214	D6	V		V	
1010110		57	215	D7	W		W	
1010111	87	58	216	D8	X		X	
1011000	88	59	217	D9	Y		Y	
1011001			218	DA	Z		Z	
1011010	9Ø 91	5A 5B	219	DB	Ï	Opening Bracket	n/a	<esc>3</esc>
1011011		5C	220	DC	1	Reverse Slant	n/a	<esc>4</esc>
1011100	92			DD	1	Closing Bracket	<shift-m></shift-m>	E.
1011101	93	5D	221	DE		Circumflex	^	
1011110	94	5E 5F	223	DF		Underline	n/a	<esc>5</esc>
10111111	95		224	EØ	7	Opening Quote	n/a	15
1100000		60	225	E1	а	opening quoes	n/a	a
1100001	97	61		E2	b		n/a	b
1100010		62	226	E3			n/a	c
1100011	99	63	227		C		n/a	d
1100100		64	228	E4	d		n/a	e
1100101		65	229	E5	e		n/a	£
1100110		66	230	E6	f		n/a	g
1100111		67	231	E7	g		n/a	ĥ
1101000		68	232	E8	h		n/a	í
1101001		69	233	E9	i		n/a	j
1101010		6A	234	EA	j		n/a	k
1101011		6 B	235	EB	k 1		n/a	1
1101100		6C	236	EC ED			n/a	m
1101101		6D	237		m		n/a	n
11011110		6E	238	EE	n		n/a	0
1101111		6F	239	EF	0		n/a	P
1110000		70	24Ø 241	FØ F1	P		n/a	q
1110001		71	242	F2	P		n/a	r
1110010		72		F3	r		n/a	s
111001		73	243	F4	S		n/a	t
1110100		74	244		t		n/a	u
111010		75	245	F5	u		n/a	v
111011		76	246		V		n/a	W
111011		77	247	F7	W		n/a	x
111100		78	248		x		n/a	у
111100		79	249		У		n/a	z
111101		7A	250		Z	Occasing Proces	n/a	<esc>6</esc>
111101		7 B	251	FB	{	Opening Brace	n/a	(ESC)
111110			252		1	Vertical Line	n/a n/a	(ESC)
111110			253		}	Closing Brace		(ESC)
111111	0 126	7 E	254	FE	1000	Overline (Tilde	/ 11/11	Station.

<sup>15.</sup> Use Closing Quote (39). For high value, use CHR\$(96), etc.

## APPENDIX E TROUBLESHOOTING HINTS

This appendix contains two tables designed to help you diagnose problems that can occur when using the SSC to communicate with an RS-232-C device. The device can be a printer, or a plotter, or terminal, or another computer, or some other Data Terminal Equipment (DTE), and it can be connected either directly, or via a modem or some other Data Communication Equipment (DCE). Whenever two DTEs are connected together, there must be TWO modems (DCEs) or ONE modem eliminator (such as the jumper block when it points toward the word TERMINAL) between them.

When diagnosing problems, remember that there are many variables involved in the communications connection:

- · the Apple II and its keyboard, screen, and software
- the SSC, the slot it is in, its switch settings (especially mode selection), its jumper block, cable, and software commands
- the external cable, with some number of wires (enough wires?) connected to pins (all the correct pins?) at each end
- · possibly two modems connected by low-grade telephone lines, plus another cable from the remote modem to the remote device
- an RS-232-C device at the other end, with its own switch settings and needs (such as paper, ribbon, AC power...)

As you can see, making all these components work together correctly is no mean feat. If there are problems, the easiest way to resolve them is to start with very simple, sure communication between the Apple and the device. Once you have established basic communication (even if the characters are garbled), further troubleshooting becomes much easier. Be patient and methodical.

Trouble usually has characteristics visible on the Apple II screen (Table E-1), or at the device (Table E-2). If your troubleshooting efforts fail, consult your Apple dealer -- but first record all the variables (as outlined above) and the symptoms you observed.

Problem	Symptom	Possible Cause	Solution
no data transfer	no sign of any commu- nication at all	cable wires not connected OK; jumper block facing wrong way	check all cable connections, then pin assignments; try reversing jumper block
characters garbled	jh2 3g%\$Q	wrong baud rate	change SW1-1 TO SW1-4 or use <n>B command</n>
		wrong data format	change SW2-1 (and SW2-2 in Comm Mode) or use <n>D command to change format</n>
		other device is off, out of paper, etc., off-line	turn on device, remedy its problems, put it on-line
paper not advancing	one line of smudge	printer needs line feeds from SSC	turn SW2-5 ON or use L(inefeed E(nable command
printer is skipping lines	lines look	printer and SSC both generating <pre><lf> after <cr></cr></lf></pre>	turn off SW2-5 in Printer Mode, or use L(inefeed D(isable command
missing characters	mssig caractrs	device buffer is overflowing	if device supports full RS-232-C handshaking, en- sure all required cable wires are connected if device supports only ETX/ACK, set SIC P8A Mode
			if device supports XON/ XOFF, set Printer Mode and use X(OFF E(nable cmd or set Comm Mode
			if device supports none of these, set delays with <n>C, <n>L and <n>F cmds</n></n></n>
device sticks at line's end going nuts	one long OK line, smudge at right end	device doesn't generate own <cr>, and isn't getting enough from Apple</cr>	use SIC P8 Mode and <n>N command, or Printer Mode and C command plus appro- priate SW2-3 and SW2-4</n>
			have software send <cr> before right margin</cr>

Table E-1. Problems Detected at the Device

Miles

Made

Problem	Symptom	Possible Cause	Solution
Apple has occasional bad times	it works one minute & not next	ACIA interrupting the Apple when DCD or DSR changes	make sure that interrupt switch SW2-6 is OFF
Apple not working	dead kybd and screen	SSC in slot #3 under Pascal	Pascal expects external terminal to run the show
Apple kybd seems off	keystrokes all lost	echo off; keyboard zapped; IN# not Ø	use E(cho E(nable cmd; unzap with POKE; IN#Ø
screen seems off	nothing typed is displayed	device not echoing (half duplex) or ACIA not sending to screen	in Comm or Terminal Mode, use E(cho E(nable; in SIC or Printer Mode, use I command or SW2-3 & -4 ON
screen is seeing double	eevveerryy tthhiinngg ttwwiiccee	device & SSC both echoing to Apple (full duplex)	use E(cho D(isable cmd in Comm Mode or use <n>N cmd in Printer Mode</n>
screen is spacing double	lines look	device generating and sending (LF) after (CR)	use M(ask E(nable command to remove extra linefeeds
forced uppercase display	lowercase beCOMES UPPERCASE	Apple monitor changing letters in GETLINE routine	use <n>T command to allow lowercase to pass through (not possible in Pascal)</n>
Apple misses some characters at the beginning of lines	pple sses ome racters t the bgnning lines	screen scrolling too slowly, or BASIC or Pascal program running too slowly, and so ACIA overruns	turn off screen ( <n>N or SW2-3 &amp; -4 in Prtr Mode); reduce scroll window; use assembly language or faster program routines; use lower baud rate (300 vs. 1200); use <n>C, <n>L or <n>F commands; in Comm Mode, chain (<n>S cmd) to 80-column card with its own scrolling hardware</n></n></n></n></n>

Table E-2. Problems Detected at the Apple

# APPENDIX F ERROR CODES

The SSC uses I/O scratchpad address \$678+s (s is the number of the slot that the SSC is in) to record status after a read operation. The firmware calls this byte STSBYTE. Table F-1 lists the bit definitions of this byte:

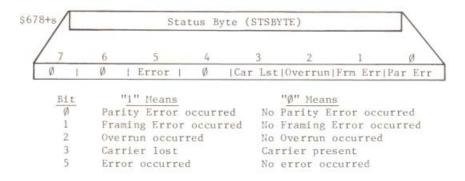


Table F-1. STSBYTE Bit Definitions

The terms Parity Error, Framing Error and Overrun are defined in the Glossary.

Bits  $\emptyset$ ,1, and 2 are the same as the corresponding three bits of the ACIA Status Register (Appendix A). Bit 3 indicates whether or not the Data Carrier Detect (DCD; Chapter 4) signal went false at any time during the receive operation. Bit 5 is set if any of the other bits are set, as an overall error indicator. If bit 5 is the only bit set, an unrecognized command was detected. If all bits are  $\emptyset$ , no error occurred.

In BASIC, you can check this status byte via a PEEK \$678+s (s is the SSC slot), and reset it with a POKE command at the same location.

In Pascal, the IORESULT function returns the error code value.



Any character--including the carriage return at the end of a WRITELN statement--will cause posting of a new value in IORESULT.

Table F-2 shows the possible combinations of error bits correspond to these decimal error codes.

BASIC PEEK \$678+s or Pascal IORESULT	Carrier Lost	Overrun	Framing Error	Parity Error
Ø		(no er	ror)	
32		(illegal	command)	
33	no	no	no	yes
34	no	no	yes	no
35	no	no	yes	yes
36	no	yes	no	no
37	no	yes	no	yes
38	no	yes	yes	no
39	no	yes	yes	yes
40	yes	no	no	no
41	yes	no	no	yes
42	yes	no	yes	no
43	yes	no	yes	yes
44	yes	yes	no	no
45	yes	yes	no	yes
46	yes	yes	yes	no
47	yes	yes	yes	yes
10001	-	0.50		

Table F-2. Error Codes and Bits

These error codes begin with the number 32 to avoid conflicting with previously defined and documented system error codes.

## **GLOSSARY**

To avoid lengthy or repetitive definitions, many terms used in one definition are themselves defined elsewhere in this glossary. Also for the sake of brevity, terms and expressions are spelled out, with their abbreviations immediately after them. In a glossary of this size, the reader will have little difficulty locating abbreviations.

ACK: An ASCII character (decimal 6; Appendix D) sent from a device to the Apple II in response to an ETX or ENQ character in SIC P8A Emulation Mode.

American Standard Code for Information Interchange (ASCII): A standard defining the codes to represent a 128-element character set (Appendix D) in a fixed way for devices of different manufacturers. It is the standard for digital communication over telephone lines.

Asserted: Made true (positive in positive-true logic; negative in negative-true logic). Usually refers to electrical signals, like the RS-232-C signal Clear To Send, etc.

Asynchronous: Having a variable time interval between characters.

Asynchronous Communications Interface Adapter (ACIA): In the SSC, a single chip (Synertek 6551 or equivalent) that converts data from parallel to serial form and vice versa, and handles serial transmission and reception and RS-232-C signals, under the control of internal registers set and changed by SSC firmware.

Baud: A unit of signalling speed equal to the number of discrete conditions or signal events per second. With the SSC, for example, using a data format of 1 start bit, 7 data bits, 1 parity bit and 1 stop bit (10 bits in all), 300 baud is approximately equal to 30 characters per second.

Binary: A number system with two digits, "0" and "1," with each digit position moving from right to left representing a successive power of two. For example, 1 represents decimal 1; 10 represents 2; 100 represents 4; 1000 represents 8, etc.

Bit: A BInary digiT, either a Ø or a 1.

-FIS

- BREAK: A  $\emptyset$ .233 second SPACE ( $\emptyset$ ) signal sent over a communication line to interrupt the sender. This signal is often used to end a session with a timesharing service.
- Carriage Return (CR): An ASCII character (decimal 13; Appendix D) that ordinarily causes a printer or display screen to place the subsequent character on the left margin. On a manual typewriter, this movement is combined with linefeed (the advancement of the paper to the next line). With computers, carriage return and linefeed are separate, causing hair-raising problems for the user.

- Carrier: The background signal on a communication channel that is modified to "carry" the information. Under RS-232-C, the carrier signal is equivalent to a continuous MARK or 1; a transition to ∅ then represents a start bit.
- Character: Any symbol that has a widely understood meaning. In the ASCII code, letters, numbers, punctuation marks, and so on, are all characters (Appendix D).
- Chip: A tiny wafer of silicon, with conductive metallic impurities, that has layers of microscopic circuits etched on it.
- Clear To Send (CTS): An RS-232-C signal from a DCE to a DTE that the SSC keeps false until the DCE makes it true, indicating that all circuits are ready to transfer data.
- Command Character: An ASCII character, usually <CTRL-A> or <CTRL-I> (Appendix D), that causes the SSC firmware to interpret subsequent characters as a command.
- Command Register: An ACIA location (at hexadecimal address \$C08A+s0) that stores parity type and RS-232-C signal characteristics.
- Communications Interface Card (CIC): An Apple II interface card designed to connect the Apple II to a device via a DCE.
- Communications Mode: An operating state in which the SSC is prepared to exchange data and signals with a DCE.
- Control Character: Any character generated by holding down the key marked CTRL while pressing some other key.
- Control Register: An ACIA location (at hexadecimal address \$C08B+s0) that stores data format and baud rate selections.
- Daisy Chaining: A method of passing incoming signals and data from one peripheral connector slot to another, such as from the SSC slot to a slot containing an 80-column-display card.
- Data Bit: With the SSC, one of 5 to 8 bits representing a character.

- Data Carrier Detect (DCD): An RS-232-C signal from a DCE to a DTE (such as the Apple II) indicating that a communication connection has been established. The SSC's internal circuits hold DCD false until the external device sets DCD true.
- Data Communication Equipment (DCE): As defined by the RS-232-C standard, any device that transmits or receives information. Usually this is a modem. However, when a Modem Eliminator is used, the Apple II looks like a DCE to the other device, and the other device looks like a DCE to the Apple.
- Data Conversion: Changing of data from parallel to serial form or from serial to parallel form.
- Data Format: The form in which data is stored, manipulated or transferred. Serial data transmitted and received by the SSC has a data format of: one start bit, 5 to 8 data bits, an optional parity bit, and one, one and a half, or two stop bits.
- Data Set Ready (DSR): An RS-232-C signal from a DCE to a DTE indicating that the DCE has established a connection.
- Data Terminal Equipment (DTE): As defined by the RS-232-C standard, any device that generates or absorbs information, thus acting as a terminus of a communication connection.
- Data Terminal Ready (DTR): An RS-232-C signal from a DTE to a DCE indicating a readiness to transmit or receive data.
- Default Value: A value that is assumed or set in the absence of explicit instructions otherwise.
- Device: A piece of equipment; usually a printer, plotter, terminal or computer. When the jumper block is in the MODEM position, the SSC expects the device to be a DCE (such as a modem).
- Echo: To send an input character to a video screen, printer, or other output device. On a typewriter, what we strike on the keyboard appears on the page in the same step. With a computer, these two steps are controlled separately.
- Electromagnetic Interference (EMI): Electrical or magnetic signals or noise that disturbs the operation of radio or television receivers. For example, a hair dryer often creates EMI that fuzzes up the picture on a nearby television set.

-8

-64

- Emulation Mode: A manner of operating in which one computer or interface imitates another. For example, in SIC P8 Emulation Mode, the SSC acts very much like an Apple II Serial Interface Card with the P8 version of firmware.
- ENQ: An ASCII character (decimal 5; Appendix D) used in the ENQ/ACK protocol (SIC P8A Emulation Mode).

- ETX: An ASCII character (decimal 3; Appendix D) used in the ETX/ACK protocol (SIC P8A Emulation Mode).
- Even Parity: Use of an extra bit set to Ø or l as necessary to make the total number of l bits an even number. For example, the 7-bit ASCII code for the letter A (1000001) has two l bits; for even parity, the transmitting device appends an eighth bit equal to Ø so that the total number of l bits remains even. The receiving device can count l bits as a way of checking for transmission errors.
- False: Zero or negative voltage in positive-true logic; positive voltage in negative-true logic. Absence of an arbitrary signal or condition.
- Firmware (FW): Software that resides in ROM and so is relatively unchangeable (firm) compared to software in RAM.
- Form Feed (FF): An ASCII character (decimal 12; Appendix D) that causes a printer or other paper-handling device to advance to the top of the next page.
- Framing Error (FRM): Absence of the expected stop bit(s) on a received character. The ACIA records this error by setting bit 1 (FRM) of its Status Register to 1. The ACIA checks and records each framing error separately: if the next character is OK, the FRM bit is cleared.

- Full Duplex: Capable of simultaneous two-way communications.
- Half Duplex: Capable of communications in one direction at a time.
- Handshake: A kind of communication protocol in which the receiving device, when it has successfully gotten a character or block of characters, sends back an acknowledging signal, thereby triggering the next transmission.
- Hardware: The actual physical switches, wires, chips, PC boards, and so on, of a computer system.
- Header: A cable connector mounted on a PC board.
- Hexadecimal: A numbering system that uses 16 digits; usually these are represented by the ten decimal digits, Ø through 9, plus the letters A through F (A representing decimal ten, F representing decimal fifteen, etc.). Each hexadecimal digit can represent a string of four binary digits.
- High-order Bit: See Most Significant Bit.
- Initialization: The process of setting up initial values and conditions. In the SSC, the firmware finds out the switch positions and the current operating system, and uses these

findings to initialize both the ACIA registers and the Scratchpad RAM locations for the slot the SSC is in.

Input: Data that flows from the outside world into the Apple II.

Interface: Some combination of hardware, firmware and software that makes possible the useful connection of two otherwise incompatible pieces of equipment.

Interrupt: A special control signal from an external source that
 diverts the Apple II from the program it is executing to a
 specific routine that handles the condition (such as a printer
 gone awry) that caused the interrupt.

Jumper Block: In the SSC, a plastic plug with pins connected in such a way that it passes RS-232-C signals between the SSC and the external device either unchanged (MODEM position) or permuted in the manner of a Modem Eliminator (TERMINAL position).

Least Significant Bit (LSB): The right-hand bit of a binary number as written down; its positional value is  $\emptyset$  or 1 (that is,  $\emptyset$  or 1 times 2 to the  $\emptyset$  power).

Linefeed (LF): An ASCII character (decimal 10; Appendix D) that ordinarily causes a printer or video display to advance to the next line.

Local: Nearby; capable of direct connection using wires only.

Low-order Bit: See Least Significant Bit.

MARK Parity: A bit of value 1 appended to the high-order end of a binary number for transmission. The receiving device can then check for errors by looking for this value on each character.

Mode: Manner of operating. The SSC can operate in one of four chief modes, depending on the settings of switches SW1-5 and SW1-6: Printer Mode, Communications Mode, SIC P8 Emulation Mode, and SIC P8A Emulation Mode.

Modem: MOdulator/DEModulator; a DCE device that connects a DTE to communications lines. As used with the SSC, a device that exchanges RS-232-C signals with the ACIA to establish a communications connection, and then either converts data from RS-232-C voltages to RS-232-C tones for transmission, or performs the opposite conversion on received data.

Modem Eliminator: The physical crossing of wires that replaces a pair of modems for direct connection of two pieces of RS-232-C Data Terminal Equipment. In the SSC, the jumper block serves this purpose when installed in the TERMINAL position.

- Most Significant Bit (MSB): The leftmost bit of a binary number as written down. This bit represents Ø or 1 times 2 to the power one less than the total number of bits in the binary number. For example, in the binary number 10000, the 1 represents 1 times 2 to the fourth power, or sixteen.
- Odd Parity: Use of an extra bit set to Ø or 1 as necessary to make the total number of 1 bits an odd number. For example, the 7-bit ASCII code for the letter A (1000001) has two 1 bits; for odd parity, the transmitting device appends an eighth bit equal to 1, making the total number of 1 bits odd. The receiving device can check for transmission errors by counting 1 bits.

No.

N.o.

E.

N.o.

Mi.o

MA.

Blad

ML.

- Output: Data that flows from the Apple II to an external device.
- Overrun (OVR): A condition that occurs when the Apple II processor does not retrieve a received character from the Receive Data Register before the subsequent character arrives. The ACIA automatically sets bit 2 (OVR) of its Status Register; subsequent characters are lost. The Receive Data Register contains the last valid data word received.
- P8: One of two types of Programmable ROM (PROM) installed in the Apple II Serial Interface Card. This PROM performed batch moves, but had no provision for software handshaking.
- P8A: One of two types of Programmable ROM (PROM) installed in the Apple II Serial Interface Card. This PROM provided the ENQ/ACK software handshaking required by several types of printers.
- Parallel Interface: A connection between two devices where there is a separate wire for each bit of a character, so that an entire character can be transferred in a single instant.
- Parity: Maintenance of a sameness of level or count, usually the count of 1 bits in each character, for error checking. In the SSC, the ACIA has a register that stores the type of parity selected (none, odd, even, MARK or SPACE). It automatically generates the parity bit when transmitting, and both checks and discards parity bits appended to received characters.
- Parity Error (PAR): Absence of the correct parity bit value in a received character. The ACIA records this error by setting bit  $\emptyset$  (PAR) of its Status Register to 1.
- Peripheral Connector Slot: One of eight 50-pin slots inside the Apple II case near the back. Within certain restrictions, each slot can contain add-on memory, an adapter for 80-column display, or an interface to an external device.
- Polarized Header: On the SSC, a 10-pin female connector for the internal cable; this connector has a slot on one side that receives a "key" on the cable's male connector.

- Printed Circuit (PC) Board: A sheet of stiff nonconductive material with one or more thin layers of metal bonded to it. Unwanted areas of this metal are etched away, leaving the paths of the desired circuits. Electronic components can then be soldered to the board. Small PC boards are also called cards.
- Printer Mode: An operating state in which the SSC is prepared to exchange data and signals with another DTE (such as a printer).
- Protocol: A predefined exchange of control signals between devices enabling them to prepare for coordinated data transfer.
- Radio Frequency Interference (RFI): Electromagnetic interference occurring at frequencies used for radio communications.
- Random Access Memory (RAM): A series of storage locations that can be accessed directly (by means of horizontal and vertical coordinates) for both reading and writing.
- Read Only Memory (ROM): A series of storage locations that can be read but cannot be written to; this protects the programs and data in the ROM from alteration or destruction.
- Receive Data Register: A read-only register in the ACIA (at hexadecimal location \$CØ88+sØ) that stores the most recent character successfully received.
- Remote: Too distant for direct connection via wires or cables only.
- Request To Send (RTS): An RS-232-C signal from a DTE to a DCE to prepare the DCE for data transmission.
- Ring Indicator (RI): An optional RS-232-C signal from a DCE to a DTE that indicates the arrival of a call.
- RS-232-C: A standard created by the Electronic Industries
  Association (EIA) to allow devices of different manufacturers
  to exchange serial data--particularly via telephone lines. The
  ACIA in the SSC implements all the required primary RS-232-C
  signals. These signals are true when at Ø volts.
- Scratchpad RAM: Eight locations in the Apple's memory reserved for each of the 8 peripheral connector slots (64 bytes in all).
- Secondary Clear To Send (SCTS): A secondary RS-232-C signal that some printers use instead of Clear To Send.
- Serial Interface: A connection in which all the bits of a character are sent along a single wire one after the other.
- Serial Interface Card (SIC): An Apple II product designed to connect an RS-232-C device directly to the Apple II.

- SIC Emulation Mode: A state of operation in which the SSC imitates an Apple II Serial Interface Card.
- SPACE Parity: A bit of value Ø appended to a binary number for transmission. The receiving device can look for this value on each character as a means of error checking.
- Start Bit: A transition from a MARK signal to a SPACE signal for one bit-time, indicating that the next string of bits represents a character.
- Status Register: An ACIA register (hexadecimal location \$CØ89+sØ) that stores the state of two of the RS-232-C signals and of the Transmit and Receive Data Registers, as well as the outcome of the most recent character transfer.

Bil.

Biles

- Stop Bit: A MARK signal following a string of data bits to indicate the end of a character.
- Super Serial Card (SSC): The interface card described in this manual. It is called "super" because it can simultaneously transmit and receive data in one of 35 formats at any of 15 speeds, honor several software protocols, communicate directly with either DTE or DCE, change operating characteristics in response to software commands, and dovetail with the chief operating environments offered with the Apple II.
- Terminal: An input/output device, usually made up of a keyboard and video display and sometimes including its own printer and magnetic storage devices, that can act as a separate and even remote site for data transfer with a computer system.
- Terminal Mode: An operating state of the SSC in which the firmware bypasses the Apple II's central processor, and makes the Apple act as a simple terminal capable of generating all of the ASCII characters.
- Transmit Data Register: A write-only register in the ACIA (at hexadecimal location \$C088+s0) that holds the current character to be transmitted.
- True: Positive voltage in positive-true logic; zero or negative voltage in negative-true logic. Assertion of an arbitrary signal or condition.
- XOFF: An ASCII character (decimal 19; Appendix D) sent by a receiving device to a transmitting device to halt transmission of characters.
- XON: An ASCII character (decimal 17; Appendix D) used in the XON/XOFF protocol as a go-ahead character from the receiving device to the sending device after an XOFF has been sent to halt transmission.

## FIGURES AND TABLES

#### Chapter 1

#### GETTING STARTED

- Figure 1-1 Photo of the Super Serial Card
- 2 Figure 1-2 Line Drawing of the SSC
- Figure 1-3 Components of the Internal Cable and Clamp Assembly Sliding the "Key" into the Groove
- 4 Figure 1-4
- Figure 1-5 Internal Cable Attached Correctly to SSC

#### Chapter 2

- dill

-

#### PRINTER MODE

- SSC Set for Printer Mode Figure 2-2 SSC in Slot #1 and Clamp Assembly in Notch 10
  - Commonly Used Switch Settings for Printer Mode
- Table 2-2 Baud Rate Switch Settings
- 8 Table 2-3 Line Width and Video Switch Settings
- 14 Table 2-4 Printer Mode Commands
- 15 Table 2-5 Baud Rate Selections
- 15 Table 2-6 Data Format Selections
- 16 Table 2-7 Parity Selections
- 16 Table 2-8 Time Delay Selections
- 18 Table 2-9 Lowercase Character Displays

#### Chapter 3

#### COMMUNICATIONS MODE

- Figure 3-1 SSC Set for Communications Mode
- 26 Figure 3-2 SSC in Slot #2 and Clamp Assembly in Notch
- 22 Table 3-1 Commonly Used Switch Settings for Communications Mode
- 23 Table 3-2 Baud Rate Switch Settings
- 23 Table 3-3 Data Format Selections
- 29 Table 3-4 Summary of Communications Mode Commands
- 30 Table 3-5 Baud Rate Selections
- 30 Table 3-6 Data Format Selections
- 31 Table 3-7 Parity Selections
- 32 Table 3-8 Time Delay Selections
- Lowercase Character Displays 33 Table 3-9
- Table 3-10 Special ASCII Character Generation 35
- Table 3-11 Terminal Mode Commands 35

#### Chapter 4

## HOW THE SSC WORKS

```
Parallel Data Transfer
      Figure 4-1
39
                  Serial Data Transfer
      Figure 4-2
39
                  Parallel-to-Serial Data Conversion
      Figure 4-3
39
                  RS-232-C Serial Data Format
      Figure 4-4
40
                  An RS-232-C Setup with Modems
      Figure 4-5
42
                 An RS-232-C Setup with a Modem Eliminator
      Figure 4-6
43
      Figure 4-7
                  SSC Operating Modes
43
                  SSC Configurations
44
      Figure 4-8
                  Overall Block Diagram of the SSC
      Figure 4-9
45
                  Binary and Decimal Digits and Quantities
      Table 4-1
38
                  RS-232-C Signals as Interpreted by the Sender
      Table 4-2
41
                  RS-232-C Signals as Interpreted by the Receiver
41
      Table 4-3
                  SSC Address Remapping
      Table 4-4
46
                  Registers in SSC Peripheral I/O Space
      Table 4-5
```

## Appendix A

47

### FIRMWARE

50	Table A-1	I/O Routine Offsets and Registers under Pascal 1.1
51	Table A-2	Bytes Used for Device Identification
51	Table A-3	Device Class Digit
52	Table A-4	Memory Usage Map
52	Table A-5	Zero-Page Locations Used by SSC
53	Table A-6	Scratchpad RAM Locations Used by SSC
54	Table A-7	SSC Registers in Peripheral Card I/O Space
55	Table A-8	Monitor ROM Entry Points Used by SSC
55	Table A-9	BASIC Entry Points Used by SSC
56	Table A-10	Pascal 1.0 Entry Points Used by SSC
57	Table A-11	Pascal 1.1 Offsets Used by SSC
57	Table A-12	SSC Special Firmware Locations

#### Appendix B

#### INTERFACE CARD **EMULATION** APPLE

```
SIC Switch Settings, PEEKs and POKEs, Part I
93
      Table B-1
                 SIC Switch Settings, PEEKs and POKEs, Part II
94
      Table B-2
```

#### Appendix C

### SPECIFICATIONS AND SCHEMATICS

98 Table C-1 Connector Pin Assignments 99 Table C-2 Jumper Block Wiring

#### Appendix E

#### TROUBLESHOOTING HINTS

106 Table E-1 Problems Detected at the Device 107 Table E-2 Problems Detected at the Apple

#### Appendix F

#### **ERROR CODES**

 $\begin{array}{cccc} 109 & & Table \ F-l & STSBYTE \ Bit \ Definitions \\ 110 & & Table \ F-2 & Error \ Codes \ and \ Bits \end{array}$ 

INDEX	binary/decimal representation
	bit 7, 23, 37
A	data 30, 40, 93 parity 40
	start 40
A register 50	stop 30, 40, 92, 94
ACIA 45, 47, 48, 91, 95, 96,	block diagram, SSC 45
function 47	moves 91
registers 49, 54	branch table locations,
status register 109	Pascal 1.1 50
ACK 91, 94, 102, 106	break signal 36
address	buffer 45, 47
bus 45	bus
lines 46	address 45
mapping 46	data 45, 48
space 46	parallel 48
addressing logic 45, 46	byte 38
answer modem 42	device signature 51
Apple II Communications Card (see CIC)	
Apple II Communications	C
Interface Card (see CIC)	
Apple II Parallel Cards 96	cable
Apple II Serial Interface Card	connector 10, 26
(see SIC)	external 1, 10, 26
APPLE SSC: 8, 12, 27	internal 1, 9, 24, 45
Applesoft 36, 98	socket 2
ASCII 24, 34, 38, 95, 96	capitalize 34
code table 101-104	carriage return delay 7, 17,
special characters 35, 36,	94, 95, 96
101	carrier
assembly language 11, 27, 98	lost 110
Asynchronous Communications	signal 42
Interface Adapter (see ACIA)	chaining 34
	character
B	displays 33
В	enquire 95
	characters
B command 35, 36	echo on screen 34
BASIC 8, 11, 12, 17, 18, 19,	lowercase 34, 102
27, 31, 33, 36, 92, 101,	uppercase 34, 102
104	capitalize 34
entry points 55	CIC 36, 91, 96
error codes 109, 110	clamp assembly 2, 3
protocol 50	Clear To Send (see CTS)
batch moves 95	code 38
baud rate 6, 15, 22, 30	column overflow 17
selection 15, 30	command
switch settings 7, 23, 47,	character
91, 92, 93, 98 binary 37, 101	change 13, 28
data 24, 38	Communications Mode 28,

Printer Mode 12, 102 send 12	D
data format 15	data 37
format 12, 27	ASCII 24
Communications Mode 27-28	binary 24
register 45	bit 7, 15, 23, 24, 30, 40,
remote-control 102	93
reset 15, 95	bus 45, 48
set time delay 16	bus buffer 45
summary, Terminal Mode 35	conversion 39, 47
translate 91	format 7, 23, 28, 30, 40, 47, 91, 92, 93
commands CIC 96	command 15
Communications Mode 27,	RS-232-C 40
29, 102	selection 15, 23, 30
Parallel Card 96	input 48, 96
Printer Mode 11, 14	output 48
Terminal Mode 22, 35	serial 48
communications 37	terminal 40
mode 21-33, 91, 95, 96	transfer 47
Communications Card,	Data Carrier Detect (see DCD)
Apple II (see CIC)	Data Communication Equipment
Communications Mode 22, 26	(see DCE)
command	Data Set Ready (see DSR)
character 28	Data Terminal Equipment (see
summary 28-29	DTE)
summary 20-29	Data Terminal Ready (see DTR)
commands 27 control character 27	DB-25 connector 10, 26, 97, 99
	pin assignments 98
components SSC 45	DCD 41, 42, 43, 47, 48, 95,
configurations, SSC 44	
connector	98, 99 DCE 41, 48, 105
pin assignments 98	
slots 9, 25, 98	decimal error codes 110
control	default parity 16
character	demodulator 41
communications mode 27	device
suppress 18, 33	class digit 51
logic 45, 46	identification 51
register 45	remote 19, 96
<pre><cr> 7, 17, 94, 95, 96, 102, 107</cr></pre>	signature byte 51 DEVICE SELECT 46, 47
<ctrl-a> 27, 102</ctrl-a>	disable 12, 28
<ctrl-i> 8, 12, 102</ctrl-i>	DOS 8, 11, 27
<ctrl-r> 28, 35, 96, 102</ctrl-r>	Dow Jones News & Quotes
<ctrl-s> 28, 96, 102</ctrl-s>	Reporter 22, 36
<ctrl-shift-m> 101, 103</ctrl-shift-m>	setup program for 36
<ctrl-shift-n> 103</ctrl-shift-n>	DSR 41, 42, 43, 48, 95, 98,
<ctrl-t> 28, 35, 96, 102</ctrl-t>	99
CTS 41, 42, 43, 47, 48, 95,	DTE 41, 42, 48, 105
98, 99	DTR 41. 42, 43, 95, 98, 99

Marie

Miles

Black

Ni.o

Ella d

Marie Control

Bild

Mile

Bild

Billion

dist.	E	locations, special 57 memory usage 53
m.d	E command 31, 34, 35 echoing 34, 35	protocol, Pascal 1.1 49 SSC driver 46
in a	edge connector 2 EIA 40	form feed delay 14, 16, 17, 29, 32
	80-column-display 34 Electronic Industries	format, data 92, 93 framing error 109, 110
and a	Association (see EIA) emulation	full-duplex 35
and the same of	modes 91-96 old Serial Interface Card 91 P8 92	G
otto.	SIC 92, 102 enable 12, 28	generate ASCII characters 35
dist	enquire character 95 enter Terminal Mode 35 entry points	<pre><cr> 17, 95, 96 <lf> 8, 17, 24, 31, 92, 94 generic signature 51</lf></cr></pre>
artical line	BASIC 55 hardware 49	ground 10, 26, 42, 43
	I/O routine 50 monitor ROM 55	Н
	Pascal 1.0 55, 56 Pascal 1.1 55, 56 SSC 55	half-duplex 35, 36, 95 handshake 94, 95
	environment, SSC 97 error	handshaking 19 hardware 96
	codes 109-110 BASIC 110 Pascal 110	entry points 49 handshaking 19 hexadecimal 101
-	framing 109, 110 parity 109, 110	Parameter Williams
and a	ENQ 95, 102 ENQ/ACK 102	TV# 1/
	<esc> 34, 35, 101-104 <esc>L 91 <esc>U 91</esc></esc></esc>	IN# 14 IDS 560 Paper Tiger 6 indirect addressing 56
	ETX 91, 94, 95, 102, 106 ETX/ACK 91, 94, 102, 106	input 37 data 48, 96
	Expansion ROM, SSC 46, 52 even parity 16, 31, 40, 92, 93	installation, SSC 9 Integer BASIC 36
	external cable 1, 10, 26	interface receive 45
end.	F	internal
-	F command 17, 31 FCC rules 10, 26	cable 1, 9, 24, 45 attachment 3 clamp assembly 2, 3
and a	<pre><ff> 14, 16, 17, 29, 32 firmware 34, 47, 49, 109</ff></pre>	shielding 10, 26 interrupt
-	entry points, Pascal 1.0 (SIC) 49 interrupt handling 50	handling 50 request 47
=	listings 57-96	Pascal 1.0 50

I/O routine entry points 50 scratchpad address 109 SELECT 46 space peripheral card 52 registers in 54 STROBE 46 IORESULT 109	modem 22, 26, 27, 41, 42, 105 answer 42 eliminator 5, 42, 43, 45, 48, 99, 105 null 42, 48 originate 42 MODEM 5, 21, 27, 48, 99 modulator 41 monitor Apple II 11 ROM entry points 55	
3	N	_
jumper block 2, 5, 11, 21, 27, 42, 45, 48, 95, 99, 105 wiring 99	NEC 5510 Spinwriter 6 no parity 31 null modem 42, 48	
V		-
K	0	1000
keyboard 11, 18, 27, 33, 101, 102	odd parity 16, 31, 40 offsets, Pascal 1.1 57	=
L	originate 40 modem 42	
<pre><lf> 8, 17, 24, 31, 91, 92, 94, 102</lf></pre>	oscillator 45, 47 output 37 data 48	-
line width 8, 92, 94, 96 linefeed (see <lf>) listings, firmware 57-96</lf>	override switch settings 27 overrun 109, 110	
local 37 lowercase 14, 18, 29, 32-33, 34, 91, 92, 93, 95, 101,	P	Bin
102 character displays 18	P8 91-95 emulation POKEs 92	
М	ROM 92, 94, 95 P8A 91 mode 6, 102, 106	_
	ROM 92, 94, 95	Bat.
MARK parity 16, 31, 40, 101 mask <lf> in 17, 31 memory usage map, SSC 53</lf>	parallel bus 48 card commands 96	BELL
mode communications 21-33, 26,	data 37, 38-39 I/O card 52	1
91, 96, 102 emulation 91-96 P8 93	parity 7, 15, 23, 30, 47, 92, 93 bit 7, 23, 24, 40	100
P8A 93-94, 102, 106 printer 5-19, 91, 92, 93-94,	error 109, 110 even 40	in the second
102 SSC operating 43 terminal 22, 34-36, 96, 101,	MARK 40, 101 odd 40 selection 16, 31	
102 126 SUPER SERIAL CARD		
100 A		666

SPACE 40	receive
Pascal 11, 12, 27	interface 45
error codes 109, 110	register 42
Pascal 1.0 49	Received Data (see RXD)
entry points 55, 56	register
interrupt requests 50	command 45, 54
Pascal 1.1 49	control 45, 54
branch table locations 50	receive 42
device	status/reset 45, 54
class digit 51	transmit 42, 54
identification 51	transmit/receive 45
signature byte 51	registers
entry points 55, 56	A 50, 55-57
firmware protocol 49	ACIA 49, 54
I/O routines 50	Pascal 1.1 50
offsets 50	peripheral I/O space 47
offsets 57	switch settings 45
PEEK 91, 92, 93-94, 109	X 50, 55-57
peripheral I/O space 49	Y 50, 55-57
physical characteristics, SSC	remote 37
97	device 19, 22, 26, 27, 96
pin assignments 98	terminal 34
POKE 19, 91, 92, 93-94, 95	remote-control commands 102
power supply 9, 25	Request To Send (see RTS)
PR# 14	reset 27
preparing SSC, printer mode 5	command 15, 95
printer(s) 6	SSC 17, 31
local 11	RESET 46
using 11	<pre><return> 12, 27, 102 Pine Indicator (PI) (1)</return></pre>
Printer Mode 5-19, 91, 92, 93-94, 95	Ring Indicator (RI) 41 ROM 45
command character 12, 102	SSC Expansion 46, 52
commands 11	ROM/RAM
summary 13-14	address mapping 46
data format 7, 15	space 46
switch settings 6 using 11	RS-232-C 40, 43, 47, 48, 95,
protocol 102	signals 40-43, 49, 98-99
BASIC 50	RTS 41, 42, 43, 95, 98, 99
Pascal 1.1 firmware 49	RXD 41, 42, 43, 98, 99
Q	S
W-7557	
Q command 35, 36	schematic, SSC 45, 100
Qume 6	scratchpad RAM locations 53
quit Terminal Mode 36, 102	SCTS 98, 99
	Secondary Clear To Send (see
D.	SCTS)
R	selection
	baud rate 15, 30
RAM 45, 49, 92	data format 15, 23, 30
scratchpad locations 53	parity 16, 31
	time delay 16, 32

serial	using	E-
data 37, 39, 40, 48	Communications Mode 27	-
interface card 95	printer 11	
setting switches 6, 22, 92	Printer Mode 11	E.
settings, switch 6, 22, 93-94	terminal 11	
shielding, internal 10, 26	zero-page locations 52	-
SIC 91, 92	start bit 7, 23, 40	Minute Control
emulation mode 92, 102	status	
firmware entry points 49	byte (see STSBYTE)	1000
P8A mode 102, 106	register, ACIA 109	
switch settings 93, 94	status/reset register 45	-
6502 registers 49, 50, 55-57	stop bit 7, 15, 23, 24, 30,	Mary Control
slew rate 48	40, 92, 94	
software compatability 98	STSBYTE 109	100
SPACE parity 16, 31, 40, 101	summary	_
special	Communications Mode commands	-
characters, ASCII 101	28-29	
circuits, SSC 97	Printer Mode commands 14	
specifications, SSC 97-98	Terminal Mode commands 35	No.
specify screen slot 34	Super Serial Card (see SSC)	
SSC 1, 91, 95, 96, 102	suppress	
block diagram 45	characters	Sec.
cable connector 10	control 33	
preparation 21	keyboard 18, 33	El .
configurations 44	<lf> in 17, 31</lf>	
connect	switch settings	-
printer 7	baud rate 7, 23	Milita
terminal 8	change 14, 30	0.00
controlling 11	Communications Mode 22	-
driver firmware 46	line width 8	-
emulation 94	override 12, 27	
entry points 50, 55-57	Printer Mode 6	100
environment 97	SIC 93-94	
Expansion ROM 46, 52	SSC 93-94	process.
firmware memory usage 53	video 8	-
installation 9, 92	switches 2	
lowercase 18	Communications Mode 21-22	100
main components 45	setting 22, 92	
operating modes 43	SSC 45	100
peripheral card I/O space 52	state of 47	-
photo 1	SW1 48, 54, 56, 57, 92, 93	
physical characteristics 97	SWI-1 through SWI-4 2, 7,	\$86
registers in I/O space 54	15, 23, 31, 92, 93, 106	
reset 17, 31	SW1-5 and SW1-6 2, 5, 21,	-
RS-232-C signals 99	92, 93, 102, 115	-
schematic 45, 100	SW1-7 2, 8, 24, 48, 92, 99	
scratchpad RAM locations 53	SW2 6, 22, 48, 54, 56, 57,	16.0
slot location 98	92, 93, 94	
special circuits 97	SW2-1 2, 7, 15, 23-24, 31,	A1000
specifications 97-98	92, 93, 94, 106	Marie Control
switch settings 93-94	SW2-2 2, 7, 16, 23-24, 31,	
theory of operation 45	92, 94, 106	Table 1
unpacking 1	SW2-3 and SW2-4 2, 8, 23,	
	92, 94, 106, 107	and a

```
SW2-5 2, 8, 17, 24, 32, 92, 93, 106
     SW2-6 2, 8, 24, 48, 92,
                                        video on/off 8, 92, 94, 96
       107
     SW2-7 2, 8, 24, 48, 92,
                                        W
                                        wiring, jumper block 99
T
                                        X
T command 35
tab 19
                                        X command 17, 31, 35, 106
X register 50, 55-57
terminal
  connecting 8
  local 11
                                        XOFF 35, 102
unintelligent 34, 35
TERMINAL 5, 11, 21, 42, 48,
95, 99, 101, 105
Terminal Mode 22, 34-36, 101-
                                          recognition 18, 33
                                        XON 102
                                        XON/XOFF 17, 19, 31, 33, 96,
                                          102
    104
  bypass 96
  change to 96
  commands 35
enter 35, 102
                                        Y register 50, 55-57
  example 36
quit 36, 102
theory of operation, SSC 45
time delay
                                        Z command 17, 18, 31
selection 16, 32
time-sharing 34, 36
                                        zero-page locations, SSC 52
translate 94
  command 17, 31, 91
  lowercase 18
transmit
 break signal 36
  interface 45
  register 42
transmit/receive register 45
Transmitted Data (see TXD)
troubleshooting 105-107
  at Apple II 107
at device 106
TXD 41, 42, 43, 98, 99
U
unintelligent terminal 34, 35,
 36
uppercase 14, 18, 29, 33, 34, 91, 92, 102
```

and to

and a

and the last

-

and the



10260 Bandley Drive Cupertino, California 95014 (408) 996-1010

030-0270-A