

Apple II

# **Super Serial Card**

Installation and Operating Manual





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

030-0270-A

#### 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.

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**

## 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.

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

## PREFACE

2

6

G

### Chapter 1 GETTING STARTED

1 Unpacking

- 2 A Close Look
- 2 Preparing Cable and Clamp Assembly
- 3 Attaching Internal Cable to SSC

## Chapter 2 PRINTER 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
11	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 Overflow(</cr>
17	Generate <lf> OutL <e d=""></e></lf>
17	Mask (Suppress) <lf> InM <e d=""></e></lf>
17	Reset the SSCR

vii

5

- 17 Other Commands
- 18 Translate Lowercase Characters--<n>T
- 18 Zap (Suppress) Control Characters--Z 18
- Find Keyboard--F <E/D> 18 XOFF Recognition--X <E/D>
- 19 Tab in BASIC--T <E/D>

## Chapter 3 COMMUNICATIONS MODE

- 21 Preparing the SSC for Communications Mode 22 Setting the Switches
- 22 Commonly Used Settings
- 22 Baud Rate
- 23 Data Format and Parity
- 24 Generate <LF> Out
- 24 Special Switches
- 24 Installation Procedure
- 26 External Cable and Connector
- 26 Using the SSC in Communications Mode
- 27 Communications Mode Commands
- 27 Command Formats
- 28 The Command Character
- 28 Communications Mode Command Summary
- 30 Commands That Change Switch Settings
- 30 Baud Rate--<n>B
- 30 Data Format--<n>D
- 30 Parity--<n>P
- 31 Generate <LF> Out--L <E/D>
- 31 Mask (Suppress) <LF> In--M <E/D>
- 31 Reset the SSC--R 31 Other Commands
- 32
- Set Time Delays--<n>C, <n>L, <n>F 32 Translate Lowercase Characters--<n>T
- 33 Zap (Suppress) Control Characters--Z
- 33 Find Keyboard--F <E/D>
- 33 XOFF Recognition--X <E/D>
- Specify Screen Slot--<n>S 34
- 34 Echo Characters on the Screen--E <E/D>
- 34 Terminal Mode
- 35 Terminal Mode Commands
- 35 Enter Terminal Mode--T
- 36 Transmit a Break Signal--B
- 36 Special Characters--S <E/D> 36 Ouit (Exit from) Terminal Mode--Q
- 36 A Terminal Mode Example

## Chapter 4 HOW THE SSC WORKS

37 Serial Data Communication 38 Parallel Data in the Apple II 39 Serial Data for Long Distances 39 Data Conversion 40 RS-232-C Data Formats 40 RS-232-C Signals 41 Modems 42 Modem Eliminators 43 SSC Modes and Configurations 45 Theory of Operation 46 Addressing and Control Logic 46 ROM/RAM Space 47 Registers in Peripheral I/O Space 47 The ACIA 48 Data Input and Output 48 Data Bus 48 Jumper Block

## Appendix A **FIRMWARE**

21

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
54	Peripheral Card I/O Space
55	SSC Entry Points
55	Monitor ROM Entry Points
55	BASIC Entry Points
56	Pascal 1.0 Entry Points
56	Pascal 1.1 Entry Points
57	Other Special Firmware Locations
58	SSC Firmware Listings

## Appendix B **APPLE INTERFACE CARD EMULATION**

91 Old Serial Interface Card Emulation

- 92 P8 Emulation POKEs
- 94 P8A Emulation POKEs 95
  - Other Emulation Mode Differences

37

49

91

- 96 Old Communications Card Commands
- 96 Switch to Terminal Mode--<CTRL-T>
- 96 Bypass Terminal Mode--<CTRL-R>
- 96 XOFF--<CTRL-S>
- 96 Parallel Card Commands
- 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

## Appendix C SPECIFICATIONS AND SCHEMATICS

97 SSC Specifications

- 98 Connector Pin Assignments
- 99 Jumper Block Wiring
- 100 Schematic Diagram

## Appendix D ASCII CODE TABLE

Appendix E105Appendix F109ERROR CODES109GLOSSARY111FIGURES AND TABLES119INDEX123

## PREFACE

97

101

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.

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 SSC.

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.

# 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

a second

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).
- 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 <u>edge connector</u>. 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.



1 11

(JIII

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).

(A)

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





# CHAPTER 2 PRINTER MODE

S

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

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.

#### **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

- IDS 56Ø SW1: OFF OFF OFF ON OFF ON ON SW2: ON ON \* \* OFF OFF OFF Paper Tiger Printer Mode, HW Hndshk, 96ØØ baud, 1 stop bit, \*\* width IDS SW1: - - - ON ON OFF OFF SW2: - - - OFF - -SSC/IDS pins: 3/3, 7/7, 2Ø/2Ø; all IDS jumpers removed
- NEC 551Ø Spinwriter SW1: OFF ON ON ON OFF OFF OFF SW2: ON ON \* \* OFF OFF ON P8A Mode, ETX/ACK, 12ØØ baud, 1 stop bit, \*\* line width NEC switches: OFF ON OFF OFF OFF OFF ON ON SSC/NEC pins: 2/2, 3/3, 7/7, 2Ø/6&8; 4&5 tied on NEC end May need keystroke to force first ETX after power-up.

NEC 551Ø Spinwriter Spinwriter Spinwriter Sul: 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 SW1: OFF ON ON ON OFF ON ON SW2: ON 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 SW1: OFF OFF OFF ON OFF ON ON SW2: ON OFF \* \* OFF OFF OFF Sprint 9/35 Printer Mode, HW Hndshk, 9600 baud, 1 stop bit, \*\* width 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 <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 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
50	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	36ØØ	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		(**	134.5	8)	19200	OFF	OFF	OFF	OFF

Table 2-2. Baud Rate Switch Settings

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

0

The SSC sends each character (such as a "3" or an "F" or a Carriage Return) as a string of zeroes and ones (<u>bits</u>). The way it can send a character in Printer Mode, using switch settings, is this:

- first a single <u>start bit</u> 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 (<CR>). 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  $\langle n \rangle C$  command described later in this chapter.

#### 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	on	ON	ON
72 char/line	off	ON	OFF
8Ø char/line	off	OFF	ON
132 char/line	off	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  $2\emptyset$ , Clear To Send, set SW1-7 OFF and SW2-7 ON.

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

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 <u>connector slots</u>. 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

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

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.

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 <CTRL-I> to <CTRL-something else>, type <CTRL-I><CTRL-something else>. For example, to change the command character to <CTRL-W>, type <CTRL-I><CTRL-W>. To change back, type <CTRL-W><CTRL-I>. No <RETURN> is required after either of these commands.

The command character <CTRL-I> is ASCII code 9. Here is how to generate this character in BASIC and Pascal:

Integer BASIC:	PRINT "*command" *embedded <ctrl-i></ctrl-i>
Applesoft BASIC: Pascal:	PRINT CHR\$(9): "command"
rascal:	WRITELN (CHR(9), 'command');

### PRINTER MODE COMMAND SUMMARY

11

11

1

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.

	Die Prickly fart o Selfraden robertin Selfraden folgen Selfraden Soler	

0

Forma		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 25Ø milliseconds (1/4 s) 2 seconds
<n>L</n>	<lf> Delay</lf>	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>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 L_<e M_<e * T_<e X_<e * Not</e </e </e </e </e 	/D> Generate <lf> Out /D&gt; Mask <lf> In /D&gt; Tab in BASIC</lf></lf>	E or D 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>

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

1

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	1Ø	2400
3	109.92 (110)	11	36ØØ
4	134.58 (135)	12	48ØØ
5	15Ø	13	7200
6	300	14	96ØØ
7	6ØØ	15	192ØØ

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,  $\langle CTRL-I \rangle 2D \langle RETURN \rangle$  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	
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 Ø 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).

<u><n>=</n></u>	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) MARK parity (parity bit always 1)
7	SPACE parity (parity bit always Ø)

#### Table 2-7. Parity Selections

For example, type <CTRL-I>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 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).

Time Delay		
none		
32 milliseconds		
25Ø milliseconds	(1/4	second)
2 seconds		
	none 32 milliseconds 25Ø milliseconds	none 32 milliseconds 25Ø milliseconds (1/4



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.

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 $\langle LF \rangle$ Out-L\_ $\langle E/D \rangle$

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  $\langle \text{CTRL-I} \rangle$ L E $\langle \text{RETURN} \rangle$  to cause your printer to print listings or double-spaced manuscripts for editing.

#### Mask (Suppress) $\langle LF \rangle$ In–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

0

Typing  $\langle CTRL-I \rangle R \langle RETURN \rangle$  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 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

- 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).
- 2 Display lowercase characters as uppercase inverse characters (that is, as black characters on a white background).
- 3 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.

 $\bigcirc$ 

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- $F_{E/D}$

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

#### Tab in BASIC-T\_(E/D)

0

If you type in  $\langle \text{CTRL-I} \rangle$ T E $\langle \text{RETURN} \rangle$ , 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. Ponsione toly we have Characteria and the second

in Statest Sole, fail, digitar simularia an as sole will with an

(that the the black derivation on a white hadgrough).

These inversions the contents, he principle inchanged, the planter is a principle of the planter is a principle of the planter is a structure of the planter is a planter in the planter in the planter in the planter is a planter in the planter in the planter in the planter is a planter in the planter

Table 2-3. Lowerches Character Simplays

If you denie the Z domard described here, all function contains on contains (1912, is well's, if fin vita contast contains if a contains the protocold that the USS can statistic for contains characters.

The set of the set of

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

ten est provert derestar prints form disvestion be bayeredes alle Note committee for example, you can the bait in the commendate of proverse. Tollowed by a rearrie cher complexity data control to the bares the Sidy tablesed by A to batter in the compress Default is F B

KOFF Recognition-X. (E/D). Trainer Gitte-Dis Econtribute constant the First to back for any ROFF. For General 19, Aroundits T) character consider trap a dedice of technol 10. The Dis, and to reference to be by Walting trapped and a dedice of technol.

## CHAPTER 3 COMMUNICATIONS MODE

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

-

1.0

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.

#### **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.

Application	Switch	Settings,	Cable	Connections,	Other	Information
-------------	--------	-----------	-------	--------------	-------	-------------

Apple II	SW1: ON OFF OFF ON ON ON ON SW2: ON ON * * OFF OFF OFF
via modem	Comm Mode, 300 baud, 8 data, 1 stop, * * parity
	If using SSC in each Apple, set both the same; for local
	connection, second jumper block points toward TERMINAL.

Apple III SW1: ON OFF OFF ON ON ON ON SW2: ON ON \* \* 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
SW1: ON OFF OFF ON ON ON ON SW2: ON OFF \*\* OFF OFF OFF Comm Mode, 300 baud, 7 data, 1 stop, \*\* parity Baud rate is limited by modem and transmission lines; some modems can also use 1200 baud; SW1-7 is always ON, and SW2-7 is always OFF; SCTS hookup is at remote modem.

Dow JonesSW1: ON OFF OFF OFF OFF ON ON ON ON SW2: ON OFF - ON OFF OFF OFFNews andComm Mode, 3ØØ baud, 7 data, 1 stop, no parityQuotesSample program at end of this chapter sets same traits.ReporterUse 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
50	ON	ON	ON	OFF	1200	OFF	ON	ON	ON
75	ON	ON	OFF	ON	18ØØ	OFF	ON	ON	OFF
11Ø*	ON	ON	OFF	OFF	2400	OFF	ON	OFF	ON
135**	ON	OFF	ON	ON	36ØØ	OFF	ON	OFF	OFF
15Ø	ON	OFF	ON	OFF	48ØØ	OFF	OFF	ON	ON
300	ON	OFF	OFF	ON	72ØØ	OFF	OFF	ON	OFF
600	ON	OFF	OFF	OFF	96ØØ	OFF	OFF	OFF	ON
(* 109	.92)	(**	134.5	8)	192ØØ	OFF	OFF	OFF	OFF

Table 3-2. Baud Rate Switch Settings

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

0

The SSC sends each character (such as a "7" or an "H" or a "?") as a string of zeroes and ones (<u>bits</u>). 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 Bits	SW2-1	Data Bits	SW2-2	Parity 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.

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.

(B)

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.



1.00

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 <u>connector slots</u>. 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 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.

M

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.
- 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)

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 <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, 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>(ro <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  $<\!\!\text{CTRL-A}\!\!>$  is ASCII code 1. Here is how to generate this character in BASIC and Pascal:

Integer BASIC:	PRINT "*command" *embedded <ctrl-a></ctrl-a>
Applesoft BASIC:	PRINT CHR\$(2): "command"
Pascal:	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>	<n>B Baud Rate</n>		see Table 3-5
<n>C <cr> Delay</cr></n>		Ø 1 2 3	no delay 32 milliseconds 25Ø milliseconds (1/4 s) 2 seconds
<n>D Data Format</n>		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
<n>F</n>	<ff> Delay</ff>	Ø 1 2 3	no delay (default) 32 milliseconds 25Ø milliseconds (1/4 s) 2 seconds
<n>L</n>	<lf> Delay</lf>	Ø 1 2 3	no delay (default) 32 milliseconds 25Ø milliseconds (1/4 s) 2 seconds
<n>P Parity</n>		Ø,2,4,6 1 3 5 7	no parity (default = ØP) odd parity even parity MARK (parity bit always l) SPACE (parity bit always Ø)
* <n>S</n>	Screen Slot	Ø-7	chain SSC output to slot n
* <n>T</n>	* <n>T Translate Lowercase (LC)</n>		change all LC to UC leave LC (possible garbage) LC to UC inverse; leave UC LC to UC; UC to inverse
B Break * R Reset the SSC * T Terminal Mode Z Zap <ctrl></ctrl>		e per la co	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 suppo</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 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>

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  $96\emptyset\emptyset$  baud, type  $\langle CTRL-A \rangle 14B \langle RETURN \rangle$ .

<n>=</n>	SSC Baud Rate	<u><n>=</n></u>	SSC Baud Rate
ø	use SW1-1 to SW1-4	8	1200
1	50	9	18ØØ
2	75	1Ø	2400
3	109.92 (110)	11	36ØØ
4	134.58 (135)	12	48ØØ
5	150	13	7200
6	300	14	96ØØ
7	600	15	192ØØ

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  $\langle CTRL-A \rangle$ 3D  $\langle RETURN \rangle$  causes the SSC to transmit each character in the form: one start bit (always transmitted), five data bits, and one stop bit.

8	1
7	1 1 (Tab)
6	1
5	1
8	2 (1 with $\langle n \rangle P$ options 4 through 7)
7	2
6	2
5	2 (1-1/2 with $\langle n \rangle P$ options Ø through 3)
	o 7 6 5 8 7 6 5

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:

Ø, 2, 4 or 6	none
1	odd parity (odd number of 1's)
3	even parity (even number of 1's)
5	MARK parity (parity bit always 1)
7	SPACE parity (parity bit always Ø)

Table 3-7. Parity Selections

For example, type  $\langle CTRL-A \rangle IP \langle RETURN \rangle$  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) (LF) In-M\_(E/D)

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

#### Reset the SSC-R

Typing  $\langle CTRL-A \rangle R \langle RETURN \rangle$  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 in this subsection control the handling of characters and of the video screen. Three commands control timed delays following transmission of  $\langle CR \rangle$ ,  $\langle LF \rangle$  and  $\langle FF \rangle$  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.

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	25Ø 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:

 $\langle n \rangle =$  What to Do with Lowercase Characters

- 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.

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)

2

3

1

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  $\#\emptyset$ , the Apple II video screen.) This allows "chaining" of the SSC to another card slot, such as an  $8\emptyset$ -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 Cs $\emptyset\emptyset$  space; this is the case for all currently available  $8\emptyset$ -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  $8\emptyset$ -column-display card in slot #3. Type  $\langle CTRL-A \rangle 3S \langle RETURN \rangle$  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  $8\emptyset$ -column format. (Not available in Pascal.)

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

For the Apple II, as for most computers, displaying (<u>echoing</u>) 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.

## **TERMINAL MODE**

Under Communication Mode, 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	}	~	ESC	RUB
or in hexadecimal:	9C	9F	DB	DC	DF	FB	FC	FD	FE	9B	FF

Table 3-10. 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		
Т	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	e Default E D (full-duplex); use D for half-duplex.		
S_ <e d=""></e>	Special Characters Enable/Disable	Default S E; allows/defeats generation of lowercase and special characters (Table 3-1Ø).		
* <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 M 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 escribed earlier in th	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  $8\emptyset$ -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  $4\emptyset$  and 42.

```
20 REM * THIS PROGRAM SETS UP THE SSC FOR DOW JONES
40 DS="": REM TYPE <CTRL-D> ESCAPE CHARACTER BETWEEN QUOTES
42 AS="": REM TYPE <CTRL-A> COMMAND CHARACTER BETWEEN QUOTES
50 PRINT DS; "PR#1": REM SSC IS IN SLOT #1:
52 PRINT AS;"6 BAUD": REM SET BAUD RATE TO 300;
54 PRINT A$;"1 DATA":
                   REM DATA FORMAT OF 7 DATA, 1 STOP
56 PRINT AS;"Ø PARITY": REM AND NO PARITY;
58 PRINT A$; "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.
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>O(UIT COMMAND.
100 REM O(UIT COMMAND SENDS CONTROL BACK TO THIS PROGRAM:
110 PRINT AS; "RESET":
                   REM RESET SWITCH-SELECTED OPTIONS
12Ø END
```

## CHAPTER 4 HOW THE SCC WORKS

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.

<u>Communication</u> 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.

Data 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)
		2	one hundred (Ø through 99)
		3	one thousand ( $\emptyset$ through 999)
binary	Ø and 1	1	two items (Ø or 1)
		2	four (Ø, 1, 1Ø or 11)
		3	eight (Ø through 111)
		4	sixteen (Ø through 1111)
		5	thirty-two (Ø 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 <u>characters</u>: 52 for the upper and lowercase alphabet,  $1\emptyset$  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 them.

#### PARALLEL DATA IN THE APPLE II

The Apple II is called an <u>eight-bit processor</u> because the basic unit of data it uses and moves around internally is an eight-bit <u>byte</u>. 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).



#### 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 <u>serial</u> data transfer (Figure 4-2).





#### DATA CONVERSION

111

-

Changing parallel data to serial data or vice versa is called <u>data</u> <u>conversion</u> (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 <u>start bit</u> at the beginning, an optional <u>parity bit</u> after the five to eight data bits, and finally one or two <u>stop bits</u> at the end (Figure 4-4). This is the data format that most RS-232-C devices use.





What is this mysterious <u>parity bit</u> 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 l in Figure 4-4 is 5, an odd number (and the parity bit is l), 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 <u>originate</u>. 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: <u>not</u> 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 <u>demodulator</u> 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 (<u>originate</u>) modem produces a fairly high tone (let's say LA) as the background or <u>carrier</u> signal that it sends; it then modulates (changes) that tone to SO to mean  $\emptyset$  and TI to mean 1. Meanwhile, the called (<u>answer</u>) modem plays a lower tone, MI, as a carrier signal, and modulates that tone to RE to indicate  $\emptyset$  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" (<u>transmit</u> <u>register</u>) of one DTE gets routed to the "earpiece" (<u>receive</u> <u>register</u>) 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

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



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

1

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
- 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 (1A, 2A)
- two registers for reading the switch settings (2B, 3B)
- an Asynchronous Communications Interface Adapter (ACIA; 4-5A) with its internal registers: status/reset register control register transmit/receive data register command register
- 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 lØ-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  $C\emptyset\emptyset\emptyset$  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 \$C800address 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.

#### **ROM/RAM Space**

The 2K ROM (4B-5C) containing the SSC driver firmware resides in the CR(0) = CFFF address space. However, an LS(0) (2C) and an LS(2) (3C) remap the addresses from the range CS(0) = CFFF to the range CF(0) = CFFF, since the CFXX addresses are unusable. (Access to them disables use of the CR(0) 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 Addresse
\$Ø7ØØ - \$Ø7FF	\$Cs∅Ø − \$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



#### Registers in Peripheral I/O Space

Whenever DEVICE SELECT drops to  $\emptyset$  volts, the Apple II is addressing the SSC's Peripheral I/O Space (the sixteen bytes starting at  $SC\emptyset \otimes \emptyset + s\emptyset$ ). 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).

Address(+s∅)	Purpose of register
\$CØ81	store state of SW1 (1A) (read)
\$CØ82	store state of SW2 (2A) and state of CTS (read)
\$CØ88	receive (read), transmit (write)
\$CØ89	status (read), reset (write)
\$CØ8A	command (read and write)
\$CØ8B	control (read and write)
	\$CØ81 \$CØ82 \$CØ88 \$CØ89 \$CØ8A

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, A $\emptyset$  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 MCl489 at 7A converts the incoming serial data from RS-232-C to TTL voltage levels. The MCl488 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	initialization routine entry point
\$C84D	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 818 at cs07 (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 CsØØ address space (s being the slot number where the peripheral card is installed). This space was chosen instead of the CsØØ space, since under BASIC protocol the CsØØ space is required, while the CsØØ 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)
\$Cs1Ø	status routine offset (required)
\$Cs11	\$00 if optional offsets follow; non-zero if not
\$Cs12	control routine offset (optional)
\$Cs13	interrupt handling routine offset (optional)

Notice that \$Csll contains \$ØØ 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.Ø 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  $65\emptyset2$  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)
		error code	(unenangeu)	(unenangeu)
\$CsØE	Read On entry On exit	\$Cs error code	\$sØ (unchanged)	character read
\$CsØF	Write	\$Cs	\$sØ	char. to write
	On entry On exit	error code		(unchanged)
\$Cs1Ø	Status On entry On exit	\$Cs error code	\$sØ (changed)	request (Ø or 1) (unchanged)
Notes:	Request code Ø Request code 1	means, "Are y means, "Do yo ply to the st	you ready to a ou have input tatus request	ccept output?" ready?" 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  $Cs\emptyset$  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)				
\$CsØC	\$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		
şø	reserved		
\$1	printer		
\$2 \$3	joystick or other X-Y input device		
\$3	serial or parallel I/O card		
\$4 \$5	modem		
\$5	sound or speech device		
\$6	clock		
\$7	mass storage device		
\$7 \$8	8Ø-column card		
\$9 \$A	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.

Addresses	Name of area	Contents
\$ØØØ <b>0-</b> \$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 \$Ø4 through \$Ø7. 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/0; 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 \$C8ØØ-\$CEFF (Table 4-4)

Table A-4. Memory Usage Map

#### **ZERO PAGE LOCATIONS**

Name

Address

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

Description

-	the second se	the second s	
*	\$24	CH	Monitor pointer to current position
	\$26	SLOT16	of cursor on screen 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	ZP TMP2	Temporary storage (various uses)
	\$35	ZPTEMP	Temporary storage (various uses)
*	\$36	CSWL	BASIC output hook (not for Pascal)
*	\$37	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)

\* Not used when Pascal initializes SSC.

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

#### SCRATCHPAD RAM LOCATIONS

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

Field name	Bit(s)	Interpretation
DELAYFLG	Ø - 1	<ff> delay selection</ff>
	2 - 3	<lf> delay selection</lf>
	4 - 5	<cr> delay selection</cr>
	6 - 7	Translate option
HANDSHKE	Ø – 7	Buffer count for handshake (P8A Mode)
PARAMETER	Ø – 7	Accumulator for FW's command processor
STATEFLG	Ø - 2	Command mode when not $\emptyset$ (Printer and Communications Modes only)
	0 - 4	Enquire character (P8A Mode); dflt ETX
		Slot to chain to (Communications Mode)
		Set to 1 after lowercase input characte
	7	Terminal Mode when 1 (Comm Mode)
	7	Enable <cr> gen. when 1 (other 3 modes)</cr>
CMDBYTE	Ø – 6	Printer Mode default is <ctrl-i>;</ctrl-i>
		Comm Mode default is <ctrl-a></ctrl-a>
	7	Set to 1 to Zap control commands
STSBYTE		Status and IORESULT byte (Appendix F)
CHNBYTE	Ø – 2	Current Apple screen slot (Comm Mode);
		when slot = $\emptyset$ , chaining is enabled
		\$CsØØ space entry point (Comm Mode)
PWDBYTE	Ø – 7	Current printer width (other modes); for listing compensation, auto- <cr></cr>
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)
MISCELC	Ø	Generate <lf> after <cr> when 1</cr></lf>
11100110	1	Printer Mode when $\emptyset$ ; Comm Mode when 1
		Keyboard input enabled when 1
		<pre><crrl-s> (XOFF), <ctrl-r> and <ctrl-t></ctrl-t></ctrl-r></crrl-s></pre>
	,	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)
	DELAYFLG HANDSHKE PARAMETER STATEFLG CMDBYTE STSBYTE CHNBYTE PWDBYTE BUFBYTE	DELAYFLG       Ø = 1         2 = 3         4 = 5         6 = 7         HANDSHKE       Ø = 7         PARAMETER       Ø = 7         STATEFLG       Ø = 2         Ø = 4       3 = 5         6 7       7         STATEFLG       Ø = 6         7       7         CMDBYTE       Ø = 6         7       7         STSBYTE       7         PWDBYTE       Ø = 6         COLBYTE       Ø = 6         COLBYTE       Ø = 7         MI SCFLG       Ø         1       2         3       4

Table A-6. Scratchpad RAM Locations Used by SSC

-

5 7

#### **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 CØ80 + (slot x 16); for example, if the SSC is in slot 3, its group of bytes extends from COB0 to COBF. Table A-7 interprets the 6 bytes the SSC uses.

ddress	Register	Bit(s)	Interpretation
\$CØ81 <b>+</b> 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 <b>+</b> 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	\$ 1 2 3 4 3 6 3 6 6 7 1 8 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 (Ø) receiver and all interrupts When 1, allow STATUS bit 3 to cause IRQ Control transmit interrupt, Request To Send (RTS) level, and transmitter When Ø, normal mode for receiver; when I echo mode (but bits 2 and 3 must be Ø) Control parity (values: Table 2-7)
\$CØ8B+sØ	CONTROL	Ø – 3 4 5 – 6 7	ACIA Control Register (read/write) Baud rate: $\$ \emptyset = 16$ times external clock; \$ 1 - \$ F = 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

## SSC ENTRY POINTS

-

100

-1

5 7

-

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.

Name	Description
COUT	sends a character to output hook (chaining) used for chaining
SETKBD	sets KSW to point to keyboard (reset)
SETSCR IORTS VIDOUT	sets CSW to point to Apple screen (reset) known position of an RTS instruction sends a character to the Apple screen
	COUT SETKBD SETSCR

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

#### **BASIC ENTRY POINTS**

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:

n exit and/or KSW po ister is outpu point to \$CsØ ut n entry n exit	(unchanged) pints to \$CsØØ. at unless KSW po 0. anything (unchanged)		character in the A and CSW does anything
and/or KSW po ister is outpu point to \$CsØ ut n entry n exit	ints to \$CsØØ. at unless KSW po Ø. anything (unchanged)	The character pints to \$CsØØ a anything	in the A and CSW does anything
n entry n exit	(unchanged)		
racter in is f	rom ACIA or key	yboard.	
put n entry n exit			
			(changed)
Table A-9.	BASIC Entry Po	oints Used by SS	C
		Compressioner (	
	n exit racter out is	n exit (unchanged) racter out is transmitted th	, , , , , , , , , , , , , , , , , , , ,

### PASCAL 1.0 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.

Addr.	Routine	X Register	Y Register	A Register
\$C8ØØ	Initialization On entry	\$Cs	ŞsØ	anything
	On exit	\$Cs	\$sØ	(unchanged)
Notes:	\$C8ØØ space is values plus SW1	enabled. Firmwa and SW2 select:	are initializes ions.	SSC to default
\$C84D	Read			
	On entry On exit	\$Cs \$Cs	\$sØ \$Cs	anything character in
Notes:	\$C8ØØ space is in the A Regist	enabled. Pasca er and location	l returns ACIA \$678+s with hi	or keyboard data gh bit cleared.
\$C9AA	Write			
		\$Cs error code		character out (changed)
Notes:	\$C8ØØ space is through the ACI	enabled. Outpu A. Pascal post	t character is s error code to	transmitted IORESULT.

Table A-10. Pascal 1.0 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. $\emptyset$ , Pascal 1.1 enters these routines using indirect addressing.

Addr.	Offset for	Value	X Register	Y Register	A Register
\$CsØD	Initializati On entry On exit	on \$(Cs)8E	\$Cs \$ØØ	\$5Ø \$5Ø	anything (changed)
Notes:	\$C8ØØ space values plus				SSC to default
\$CsØE	Read On entry On exit	\$(Cs)94	\$Cs	\$sØ	anything char, in
Notes:					A or keyboard
\$CsØF	Write On entry On exit	\$(Cn)97	\$Cs error code	\$s∅ \$Cs	char, out (changed)
Notes:	\$C8ØØ space out through				ister is sent
\$Cs1Ø	Status On entry On exit	\$(Cs)9A	\$Cs error code	\$sØ \$sØ	request (Ø or 1) error code
Notes:	\$C8ØØ space 'ready to tra	nsmit anoth put charact	Request = er byte; red	Ø asks ACIA quest = 1 as	whether it is ks ACIA whether carry bit = Ø

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	\$Ø1	Pascal 1.1 generic signature byte (\$01 = firmware card)
\$CsØC	\$31	Pascal 1.1 Device Signature byte (\$31 = serial or parallel I/O card #1)
\$Cs11	\$85	Pascal 1.1 optional routines flag (nonzero value = not supported)
\$CsFF	\$Ø8	Firmware revision level

Table A-12. SSC Special Firmware Locations

## **SSC FIRMWARE LISTINGS**

0000:	2 *********************	*******
0000:	3 *	*
0000:	4 * APPLE II SSC FIRMWARE	*
0000:	5 *	*
0000:	6 * BY LARRY KENYON	*
0000:	7 * -JANUARY 1981-	******
0000:	8 *	And Stands and Annual * months
0000:	9 * (C) COPYRIGHT 1981 BY 7	APPLE COMPUTER, INC. *
0000:	10 *	*
0000:	11 ************************************	* * * * * * * * * * * * * * * * * * * *
0000:	12 *	*
0000:	13 * VARIABLE DEFINITIONS	*
0000:	14 *	
0000:	15 ****************	****
0000:	16 ************	
0000:	17 * ZERO PAGE EOUS *	
0000:	18 *************	
0024:	19 CH EQU \$24 ;0	CURSOR HORIZONTAL POSITION
0026:		SAVE \$NO TO FREE UP Y-REG
0027:		OUTPUT, SCREEN AND INPUT CHARS
0028:		BASE SCREEN ADDRESS POINTER
0035:		WORKHORSE TEMPORARY
002A:		WHEN ZPTEMP ISN'T ENOUGH
002B:		TEMPORARIES, TEMPORARIES!
0036:		CHAR OUT VECTOR
0037:	27 CSWH EQU \$37	
0038:		CHAR IN VECTOR
0039:	29 KSWH EOU \$39	
003C:		BATCH MOVE POINTER
004E:		RANDOM NUMBER SEED
004F:	32 RNDH EQU \$4F	
0000:	33 ************	
0000:	34 * GENERAL EQUATES *	
0000:	35 ************	
0100:	36 STACK EQU \$100 ;	SYSTEM STACK BLOCK
0200:	37 INBUFF EQU \$200 ;	SYSTEM INPUT BUFFER
C000:	38 KBD EQU \$C000 ;	KEYBOARD INPUT
C010:	39 KBDSTRB EQU \$C010 ;	KEYBOARD CLEAR
CFFF:	40 ROMSOFF EQU \$CFFF ;	DISABLES CO-RES. \$C800 ROMS
0000:	41 ***************	
0000:	42 * SSC CARD ADDRESSES *	
0000:	43 **************	
C081:		(+\$NO) DIPSWITCH BLOCK 1
C082:		(+\$NO) DIPSWITCH BLOCK 2
C088:	~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~	(+\$NO) TRANSMIT DATA REG (WRITE)
C088:	-	(+\$NO) READ DATA REG (READ)
C089:	~ .	(+\$NO) STATUS REGISTER (READ)
C089:	49 RESET EQU \$C089 ;	(+\$NO) SOFTWARE RESET (WRITE)
C08A:		(+\$NO) COMMAND REGISTER (R/W)
C08B:	51 CTLREG EQU \$C08B ;	(+\$NO) CONTROL REGISTER (R/W)

0000: 0000: 0000: 0000: 3 0000: 0000: 0000: 0000: 0000: 0000: 0000: 0000: 0000: 0000: 0000: -----0000: 0000: 0000: 0000: 0538: 0438: 0438: 04B8: 0000: 0000: 0000: 0000: 0000: 0000: 0000: Ξ 0000: 0000: 0000: 0000: 0000: 0000: 03B8: 0000: 0000: 0000: 0000: 0000: 0000: 0000: 0000: 0000: 0000: 0000: 0000: 05B8: 0638: 06B8: 0738: 0000: 0000: 0000: 0000: 0000: 

Car.

S.C. 

Sel. 

1

5

5

E.

.

1

Can.

6

S.C.

	**************************************
55	
56	* DIPSW1 S1 S2 S3 S4 Z Z S5 S6 (LEFT DIPSWITCH)
	* (S1-S4 USED FOR BAUD RATE, S5-S6 FOR FIRMWARE MODE)
	* DIPSW2 S1 Z S2 Z S3 S4 S5 CTS (RIGHT DIPSWITCH)
	* STREG INT DSR DCD TDR RDR OVR FE PE
63 64 65	* CTLREG STB << WL >> CK << BAUD RATE >>
	* CMDREG < <parity>&gt; ECH &lt;<xmit>&gt; RE DTR</xmit></parity>
	~ ************************************
	**********
	* SCREEN VARIABLES: PPC AND SIC MODES *
	CMDBYTE EQU \$5F8-\$C0 ;HOLDS COMMAND CHARACTER (PPC & CIC
73	HANDSHKE EQU \$4F8-\$C0 ;SIC 98A CHAR COUNTER FOR ETX/ACK PARAMETER EQU \$4F8-\$C0 ;ACCUMULATOR FOR CMD PARAMETER
	STATEFLG EQU \$578-\$C0 ;
	* B7=CR GEN ENB FLAG B6=AFTER LC INPUT FLG
	* B2-B0=COMMAND INTERPRETER STATES
	* 0 0 0 IDLE
	* 0 0 1 CMD CHAR RECEIVED
	* 0 1 0 COLLECT <n> UNTIL CHAR THEN DO COMMAND</n>
	* 0 1 1 SKIP UNTIL SPACE, THEN GOTO STATE 4
	* 1 0 0 E/D COMMANDS
	* 1 0 1 UNUSED
	* 1 1 0 WAIT UNTIL CR THEN SET STATE TO ZERO
	* 1 1 1 WAIT UNTIL CR THEN DO PROC INDICATED BY PARM
	* (B4-B0 DETERMINE ENQUIRE CHAR FOR P8A MODE)
	DELAYFLG EQU \$478-\$CO
	* B7-B6=SCREEN TRANSLATION OPTIONS
	* 0 0 LC->UC
92	* 0 1 NO TRANSLATION
93	* 1 0 LC->UC INVERSE
94	* 1 1 LC->UC, UC->UC INVERSE
95	* (1-3 WILL ALLOW LC CHARS TO PASS THRU MONITOR)
96	* A construction of the Management of the Annual State of the State
97	* B5-B4=CR DELAY 0 0 = NO DELAY
98	* B3-B2=LF DELAY 0 1 = 32 MILLISEC
99	* $B1-B0=FF$ DELAY 1 0 = 1/4 SEC
00	* 1 1 = 2 SEC
01	*
	STSBYTE EQU \$678-\$CO ;STATUS/IORESULT/INPUT BYTE PWDBYTE EQU \$6F8-\$CO ;PRINTER (FORMAT) WIDTH
04	COLBYTE EQU \$778-\$CO ;COLUMN POSITION COUNTER
	MISCFLG EQU \$7F8-\$C0 ;
	* B7=ECHO BIT B6=TABBING OPTION ENABLE
	* B5=LINEFEED EAT B4=PASCAL/BASIC FLAG
0R	* B3=XOFF ENB FLAG B2=KEYBOARD ENB
00	

0000:						****
0000:						T INDEPENDENT) *
0000:		114	******	*****	*******	******
07F8:				EQU		;BUFFER FOR HI SLOT ADDR (\$CN)
0000:		116	******	*****	*******	****
0000:		117	* SCREE	N VARI	ABLES: CIC	MODE *
0000:		118	******	*****	*******	****
0000:		119	*			
0000:		120	* STATE	FLG: H	37=TERMINAL	MODE FLAG
0000:		121	*	B3-E	5=CHAIN SL	TO
0000:		122	*			
0638:				EOU	\$6F8-\$C0	;CURRENT OUTPUT SCREEN (\$CN00 ENTRY)
0000:		124		~~		(
0000:		0.77.0		=CN00	ENTRY	
0000:		126				
06B8:				FOII	\$778-\$00	;BUFFER FOR ONE
0000:		128		220	\$110-\$00	
0000:		128				INPUT BYTE: HIGH BIT IS SET
0000:		130				WHEN BUFFER IS FULL
0000:		1.7.7				RE-MEDN HODE CUITEM THE
0000:		131	* MISCF	LG:		B6=TERM MODE SHIFT ENB
				01.00	UNDINDING	10 DESTURN FOR SHE AND AND
0000:				SLOT	VARIABLES	AS DEFINED FOR PPC AND SIC MODES
0000:		134				
0000:					********	
0000:		1.2.2		12221222	BROUTINES *	
0000:		137	******	*****	*******	
FDED:		138	COUT	EQU	\$FDED	;CHARACTER OUT (THRU CSW)
FE89:		139	SETKBD	EQU	\$FE89	;SETS KSW TO APPLE KEYBOARD
FF58:		140	IORTS	EQU	\$FF58	;KNOWN "RTS" LOCATION
FCBA:		141	NXTA1	EQU	\$FCBA	; INCREMENT A1H, L AND CMP TO A2H, L
FE93:		142	SETSCR	EOU	\$FE93	;SETS CSW TO APPLE SCREEN
FDF6:		143				;OUTPUT A CHAR TO APPLE SCREEN
0000:		144		~	SSC.CN00	
0000:		1	******			****
0000:		2	*			
0000:				TT SS	SC FIRMWARE	* * · · · · · · · · · · · · · · · · · ·
0000:			*			*
0000:		-		TADDY	KENYON	
0000:			*	LARKI	RENION	
0000:		7		ANUADS	( 1981-	*****
0000:		8		ANUAR	1981-	+
0000:				OBVBT		* APPLE COMPUTER, INC. *
		10	10 A. C.	OFIRIO	aur 1361 B)	AFFLE COMPUTER, INC. *
0000:		1.53				
0000:		10.00		*****	*********	*********
0000:		12			The second second	*
0000:			* CN00	SPACE	CODE	*
0000:		14				
0000:						*****
	T OBJE				SSC.DCLS.OF	ВЈО
C700:		16		ORG	\$C700	
C700:		17				
C700:2C 5			BINIT			;SET THE V-FLAG
C703:70 0	С	19		BVS		; <always></always>
C705:38		20	IENTRY	SEC		;BASIC INPUT ENTRY
C706:90		21		DFB	\$90	;OPCODE FOR BCC
C707:18		22	OENTRY	CLC		;BASIC OUTPUT ENTRY
C708:B8		23		CLV		
C709:50 0	6	24		BVC	BENTRY	; <always> SKIP AROUND PASCAL 1.1 ENTRY</always>
and the second second						Antheorem and a constraint and a star start start and a start of the start of the start of the start of the star

10

5

C708:01			25		DFB	\$01	;GENERIC SIGNATURE BYTE
C70C:31			26		DFB	\$31	;DEVICE SIGNATURE BYTE
C70D:8E			27		DFB	>PINIT	
C70E:94			28		DFB	>PREAD	
C70F:97			29		DFB	>PWRITE	
C710:9A			30		DFB	>PSTATUS	
C711:85	27		31	BENTRY	STA	CHARACTER	
C713:86			32		STX	ZPTEMP	; INPUT BUFFER INDEX
C715:8A			33		TXA		SAVE X AND Y REGS ON STACK
C716:48			34		PHA		JOHND X HAD I REGS ON STACK
C717:98			35		TYA		
C718:48			36		PHA		
C719:08			37		PHP		;SAVE ENTRY FLAGS
C71A:78			38		SEI		NO RUPTS DURING SLOT DETERMINATION
C71B:8D	FF	CF	39		STA	ROMSOFF	;SWITCH OUT OTHER \$C800 ROMS
C71E:20			40		JSR	IORTS	yourrow our orman peopo none
C721:BA			41		TSX		TOUTTON ON ANY ANY ANY ANY ANY ANY ANY ANY ANY AN
C722:BD	00	01	42		LDA	STACK, X	;RECOVER \$CN
C725:8D			43		STA	MSLOT	
C728:AA			44		TAX		;X-REG WILL GENERALLY BE \$CN
C729:0A			45		ASL	A	The first oppression of the
C72A:0A			46		ASL	A	;DETERMINE SNO
C72B:0A			47		ASL	A	/ Diminizing Que
C72C:0A			48		ASL	A	
C72D:85	26		49		STA	SLOT16	
C72F:A8			50		TAY	010110	;Y-REG WILL GENERALLY BE \$NO
C730:28			51		PLP		RESTORE RUPTS
C731:50	29		52		BVC	NORMIO	
C733:			53	*			
C733:					TNT	TIALIZATION	
C733:			55		****	THUTTON	
C733:1E	38	05	56		ASL	CMDBYTE, X	ALWAYS ENABLE COMMANDS
C736:5E	38	05	57		LSR	CMDBYTE, X	Jumilie States Countration
C739:B9	8A	CO	58		LDA	CMDREG, Y	JUST HAD A POWER-ON OR PROGRAM RESE
C73C:29	1F		59		AND	#\$1F	JOODI HAD A FOREN-ON ON FROMAN RESE
C73E:D0			60		BNE	BINIT1	
C740:A9	1000		61		LDA	#SEF	; IF SO, GO JOIN INIT IN PROGRESS
C742:20		CB	62		JSR	INIT1	, IL SO, GO JOIN INIT IN PROGRESS
C745:	~~		63	*	UUK		
C745:E4	37			BINIT1	CPX	CSWH	
C747:D0			65		BNE	FROMIN	
C749:A9			66		LDA	#>OENTRY	
C74B:C5	36		67		CMP	CSWL	; IF CSW IS ALREADY POINTING TO OENTR
C74D:F0	05		68		BEO	FROMIN	; THEN WE MUST HAVE COME FROM KSW
C74F:85	36		69		STA	CSWL	; OTHERWISE, SET CSW TO OENTRY
C751:18	28		70	FROMOUT			; INDICATE WE ARE CALLED FOR OUTPUT
C752:90	08		71		BCC	NORMIO	; <always></always>
C754:E4			72	FROMIN	CPX	KSWH	MAKE SURE KSW POINTS HERE
C756:D0			73	Contraction of the state	BNE	FROMOUT	i i i i i i i i i i i i i i i i i i i
C758:A9	05		74		LDA	#>IENTRY	AND
C75A:85	10170		75		STA	KSWL	SET UP KSW (NOTE CARRY SET FROM CP)
C75C:	-		76	*	a arts		Joss of Ron (note CARAT SET FROM CP)
C75C:			77		1 100		BASIC I/O ROUTINE
C75C:			78		. 10	AFFROPRIATE	PHOTO I/O KOOIINE
C75C: BD	38	07		NORMIO	LDA	MISCELG Y	SEPARATE CIC MODE FROM OTHERS
		1996	80		AND	#\$02	;NOT ZERO FOR CIC MODE
C75F:29	02		80				
C75F:29 C761:08	02		81		PHP	#902	;SAVE CIC MODE INDICATION
C764:4C	BF	C8	83		JMP	BINPUT	
----------	------------	----	-----	----------	--------	-------------	------------------------------------
C767:			84				
C767:BD	<b>B</b> 8	04		BOUTPUT		STATEFLG, X	CHECK FOR AFTER LOWERCASE INPUT
C76A:48			86		PHA		
C76B:0A			87			A	STRACT STREET, STREET, ST.
C76C:10			88		BPL		;SKIP IF NOT
C76E:A6			89		LDX	ZPTEMP	
C770:A5			90		LDA	CHARACTER	
C772:09			91		ORA	#\$20	
C774:9D		02	92		STA		RESTORE LOWERCASE IN BUFFER
C777:85			93		STA		; AND FOR OUTPUT ECHO
C779:AE	F8	07	94		LDX	MSLOT	
C77C:68				BOUTPUT1			
C77D:29			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 MODE
C78B:			102	*			
C78B:4C	FC	C8		BOUTPUT2			
C78E:			104	******	*****	********	*****
C78E:			105				*
C78E:			106	* NEW H	PASCAI	INTERFACE	E ENTRIES *
C78E:			107				*
C78E:			108	******	*****	********	*****
C78E:20	00	C8	109	PINIT	JSR	PASCALINI	F ;
C791:A2	00		110		LDX	#0	;NO ERROR POSSIBLE
C793:60			111		RTS		
C794:4C	9B	C8	112	PREAD	JMP	PASCALREAD	);
C797:4C	AA	C9	113	PWRITE	JMP	PASCALWRIT	FE ;
C79A:			114	*			
C79A:			115	* NEW PA	ASCAL	STATUS REG	QUEST
C79A:			116	*			
C79A:			117	* A-REG=	=0 ->	READY FOR	OUTPUT?
C79A:			118	* A-REG=	=1 ->	HAS INPUT	BEEN RECEIVED?
C79A:			119	*			
C79A:4A			120	PSTATUS	LSR	A	;SAVE REQUEST TYPE IN CARRY
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	
C7A5:18			125		CLC		
C7A6:90	03		126		BCC	PSTATUS2	;CARRY CLEAR FOR NOT READY
C7A8:			127	*			
C7A8:20	D2	CA	128	PSTATIN	JSR	SRIN	;SETS CARRY CORRECTLY
C7AB:BD	BR	05	129	PSTATUS	2 LDA	STSBYTE, X	GET ERROR FLAGS
C7AE: AA			130		TAX		NAMES AND ADDRESS OF TAXABLE PARTY
C7AF:60			131		RTS		
C7B0:				******		********	*****
C7B0:			133	* ROUTI	NE TO	SEND A CH	ARACTER TO ANOTHER CARD *
C7B0:			134	******	****	********	******
C7B0:A2	03		135	SENDCD	LDX	#3	
C7B2:B5	36	į.	136	SAVEHOO	K LDA	CSWL,X	
C7B4:48			137		PHA		
C785:CA			138		DEX		
C7B6:10	FA		139		BPL	SAVEHOOK	
C7B8:			140				

13				
	C7B8:	141 * NOW PUT CA	RD ADDRESS IN HOOK	
	C7B8:	142 *		
	C7B8: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	#SCO FORM SCN	
	C7CA:85 37	152 STA	CSWH	
	C7CC:	153 *	Conti	
		154 * OUTPUT TO	THE DEPT DUEDAT	
	C7CC: C7CC:	155 *	THE PERIPHERAD	
	C7CC:8A	156 TXA	;SAVE SCN	
	C7CD: 48	157 PHA	JORAT SCH	
	C7CE: A5 27	158 LDA	CHARACTER	
		159 PHA	CHARACIER	
-	C7D0:48		4400	WE UT DIE ON
	C7D1:09 80	160 ORA	#\$80 ;80 COL BOARDS WA	NT HI-BIT ON
-	C7D3:20 ED FD	161 JSR	COUT	
	C7D6:	162 *		
	C7D6:		RE EVERYTHING THE OTHER CARD	MAY HAVE CLOBBERED
-	C7D6:	164 *		
-	C7D6:68	165 PLA		
	C7D7:85 27	166 STA	CHARACTER	
-	C7D9:68	167 PLA		
	C7DA:8D F8 07	168 STA	MSLOT	
	C7DD: AA	169 TAX		
-	C7DE: OA	170 ASL	A	
-	C7DF:OA	171 ASL	A	
	C7E0:0A	172 ASL	A	
	C7E1:0A	173 ASL	A	
-	C7E2:85 26	174 STA	SLOT16	
	C7E4:8D FF CF	175 STA	ROMSOFF	
	C7E7:	176 *		
-	C7E7:	177 * PUT BACK (	CSWL INTO CHNBYTE	
	C7E7:	178 *		
	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		
	C7EF:95 36	184 STA	CSWL, X	
	C7F1:E8	185 INX		
-	C7F2:E0 04	186 CPX	#4	
	C7F4:90 F8	187 BCC	RESTORHOOK	
	C7F6:	188 *		
-	C7F6:AE F8 07	189 LDX	MSLOT	
	C7F9:60		19701	
		190 RTS		
	C7FA:	191 *		
	C7FA:C1 D0 D0	192 ASC	"APPLE"	
	C7FD:CC C5			
-	C7FF:08	193 DFB	\$8	
	C800:	194 *		
1				
-				

6 3

6 1

6 1

C800:			196		CHN	SSC.C800	
C800:			1	******	*****	********	*****
C800:			2				water and the second second second second
C800:					TT SS	C FIRMWAR	*
C800:			4		22 00	o i fiamma	
C800:			5		ADDV	KENYON	
				* DII	JARRI	KENION	
C800:			6 7			1001	*
C800:				* -01	ANUARI	1981-	*****
C800:			~		DUDTO		A NOT E CONDUCTO THE
C800:			-		JPIRIC	HT 1981 B	Y APPLE COMPUTER, INC. *
C800:			10	*			*
C800:					*****	********	*****
C800:			14	*			*
C800:					SPACE:	HIGH LEVI	EL STUFF *
C800:			14				*
C800:							*****
C800:						INIT ENTRY	
C800:			17	******	*****	********	***
N	EXT	OBJ	ECT F	ILE NAM	E IS S	SSC.DCLS.O	BJ1
C800:			18		ORG	\$C800	
C800:20	9B	C9	19	PASCALI	NIT JS	SR PENTRY	; PASCAL 1.0 INITIALIZATION ENTRY
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	B8	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	
C81A: B9	82	CO	29		LDA	DIPSW2,Y	;SET LF GEN OPTION FROM D2-S5
C81D:85			30		STA	ZPTMP2	SAVE FOR LATER
C81F:4A			31		LSR	A	:S5-> CARRY
C820:4A			32		LSR	A	; IF S5=ON=O THEN LEAVE MISCFLG ALONE
C821:90			33		BCC	INIT1A	and the second
C823:68			34		PLA		;OTHERWISE, MAKE SURE LF GEN
C824:29			35		AND	#SFE	; ENABLE IS RESET
C826:48			36		PHA		a state to prove the state state of the state
C827:B8			37	INIT1A	CLV		V WILL BE CLEAR FOR CIC MODE
C828:B9	81	CO	38		LDA	DIPSW1,Y	second in St. 100
C82B:4A			39		LSR	A	;SIC MODES SET CARRY
C82C:B0			40		BCS	INIT2	BRANCH FOR SIC MODES
C82E: 4A			41		LSR	A	CONTRACT OF ADDRESS OF
C82F:BC			42		BCS	INIT2B	PPC MODE BRANCH
C831:A9			43		LDA	#\$01	;CTL-A
C833:D0			44		BNE	INIT5	; (ALWAYS) CIC MODE BRANCH
C835:DC	. 30		44		DITE		L'ANNUARY and HARMAN
C835:47				INIT2	LSR	A	SET CARRY FOR P8A
C835:44			40	THTTT	LDA	#\$03	SET ETX AS DEFAULT INQUIRY CHAR
C836: AS			47		BCS	INIT2A	BRANCH FOR P8A
			40		LDA	#\$80	FOR P8 SET AUTO CR GEN
C83A:A9				TNITOOA			
C83C:91				INIT2A	STA	STATEFLG,	
C83F:20			51	INIT2B	BIT	IORTS	;SET V-FLAG FOR PPC, SIC MODES
C842:A			52 53		LDA AND	ZPTMP2 #\$20	SET CR DELAY
C844:29							
C846:49	-		54		EOR	#\$20	;SO 1=ENB, 0=DISABLE
C848:91	) R8	03	55 56		STA	DELAITLG,	X ; FROM D2-S2
C84B:			20				

13								
	C84B:70	OA		57				; <always> BRANCH AROUND PASCAL</always>
	C84D:			1000			********	
	C84D:						READ ENTRY	THE REPORT OF A CONTRACT OF A
	C84D:						r \$C84D)	CONTRACT SOMEONICAST IN 1994 AND DECEMBER
-	C84D:			~ .			********	
	C84D:20				PREADO	JSR		;DO PASCAL 1.1 READ
	C850:AE			63				;MODIFY FOR 1.0
	C853:9D	B8	05	64			STSBYTE, X	;CHARACTER READ
	C856:60			65		RTS		
	C857:			66	******	*****	********	The supersone subscription a provide the laboration
	C857:						WERE WE???	
	C857:			68	******	*****	*******	* Proce as Arrow ware * NC1 C
	C857:			69	*			
	C857:A5	2B		70	INIT3	LDA		; PPC, SIC MODES USE SWITCHES
-	C859:4A			71		LSR	A	; TO SET PWIDTH, CR DELAY
	C85A:4A			72		LSR	A	INAL THE AT CHARTE & DEA
-	C85B:29	03		73		AND	#\$03	
	C85D: A8			74		TAY		
	C85E:F0	04		75		BEQ	INIT4	
-	C860:			76	*			
	C860:68			77		PLA		;RESET VIDEO ENABLE FOR PWIDTH#40
	C861:29	7F		78		AND	#\$7F	
-	C863:48			79		PHA		
	C864:			80	*			
-	C864:B9	A6	C9	81	INIT4	LDA	PWDTBL, Y	
-	C867:9D			82		STA	PWDBYTE, X	
_	C86A:A4			83		LDY	SLOT16	
-	C86C:			84	*			
100	C86C:68			85		PLA		CLEAR CIC BIT IN FUTURE MISCFLG
-	C86D: 29			86		AND	#\$95	; (AND TABBING, XOFF AND LF EAT BITS)
1	C86F:48			87		PHA		The second s
	C870:A9	09		88		LDA	#\$09	;CTL-I
-	C872:			89	*			
-	C872:9D	38	05	90	INIT5	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 F	OR TH	E ACIA INIT	TIALIZATION ROUTINE
	C879:			95	*			
-	C879:A5	2B		96	INITACI	A LDA	ZPTMP2	;DIPSW2
	C87B:48			97		PHA		
	C87C:29	AO		98		AND	#\$AO	;DATA BIT OPTIONS FOR CIC MODE
	C87E:50			99		BVC		BRANCH FOR CIC MODE
	C880:29	80		100		AND	#\$80	;8 DATA, 1 OR 2 STOP FOR SIC, PPC
	C882:20	A1	CD	101	INITACI	LA1 JS	R DATACMD1	SET CONTROL REG
-	C885:20			102		JSR		SET DIPSWITCH BAUD RATE
	C888:68	1		103		PLA		
	C889:29	00		104		AND	#\$0C	; PARITY OPTIONS FOR CIC MODE
-	C88B:50			105		BVC		BRANCH FOR CIC MODE
	C88D: A9			106		LDA	#\$0	DISABLE PARITY FOR SIC, PPC MODES
-	C88F:0A				INITAC			
-	C890:0A			108		ASL	A	
	C891:0A			109		ASL	A	
	C892:09			110		ORA	#\$0B	
-	C894:99			111		STA	CMDREG, Y	
	C897:B9			112			RDREG, Y	THROW OUT THE STRANGE STUFF
		00	00	112			ADAEG, I	THE OT THE STRANGE STOLE
3		1		14.0				
3	C89A:60 C89B:	)		113		RTS	*********	

C89B:			115	* PASCAL READ ROUTINE *
C89B:			116	*****
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
C8A1:29			119	AND #\$7F ;CLEAR HIGH BIT FOR PASCAL
C8A3:AC	F8	07	120	PASEXIT LDY MSLOT
				LDX STSBYTE, Y ; ERROR STATUS-> X-REG
C8A9:60			122	
CBAA:			123	******
CBAA:			124	* GETCHAR ROUTINE WAITS FOR *
CBAA:				* THE NEXT CHAR FROM EITHER *
CBAA:				* THE ACIA OR KEYBOARD (IF *
CBAA:			127	* ENABLED). USED BY PASCAL *
CBAA:			128	* READ ROUTINE, XON WAIT, *
CBAA:			129	* AND ACK WAIT. DATA IS RE- *
C8AA:			130	* TURNED IN THE A-REGISTER *
C8AA:			131	*****
C8AA:20	FF	CA	132	GETCHAR JSR INPUT ;ACIA DATA?
C8AD: BO	05		133	BCS GETCHAR1
C8AF:20	2C	CC	134	JSR CKKBD ;KEYBOARD INPUT?
C8B2:90	F6		135	BCC GETCHAR
C8B4:60			136	GETCHAR1 RTS ; EXIT WHEN WE HAVE SOMETHING
C8B5:			137	Aura Amb
C8B5:			138	CHN SSC.HILEV

6

2:3

-

C8B5:	2 *****
C8B5:	3 *
C8B5:	4 * APPLE II SSC FIRMWARE *
C8B5:	5 *
C8B5:	6 * BY LARRY KENYON *
C8B5:	7 *
C8B5:	8 * -FEBRUARY 1981- **********
C8B5:	9 *
C8B5:	10 * (C) COPYRIGHT 1981 BY APPLE COMPUTER, INC. *
C8B5:	11 *
C8B5:	12 *************
C8B5:	13 * *
C8B5:	14 * CIC, SIC, PPC MODE HIGH-LEVEL *
C8B5:	15 * *
C8B5:	16 ********************************
C8B5:	17 * CIC EXIT ROUTINE *
C8B5:	18 ************************************
C8B5:20 1E CA	19 CICEXIT JSR CHECKTERM ; SEE IF WE'VE ENTERED TERMINAL MODE
C8B8:	20 ***************
C8B8:	21 * BASIC EXIT ROUTINE *
C8B8:	22 ***********
C8B8:68	23 BASICEXIT PLA
C8B9:A8	24 TAY
C8BA:68	25 PLA
C8BB: AA	26 TAX
C8BC:A5 27	27 LDA CHARACTER
C8BE:60	28 RTS
C8BF:	29 ******
C8BF:	30 * BASIC INPUT ROUTINE *
C8BF:	31 **********
C8BF:F0 29	32 BINPUT BEO BINACIA ; BRANCH IF NOT CIC MODE
C8C1:BD B8 06	33 LDA BUFBYTE, X ; INPUT BUFFER FULL?
C8C4:10 05	34 BPL BINKBD
C8C6:5E B8 06	35 LSR BUFBYTE, X ;RESET BUFFER FULL
C8C9:D0 24	36 BNE BINACIA1 ; <always></always>
C8CB:	37 *
C8CB:20 3E CC	38 BINKBD JSR GETKBD ;KEYBOARD DATA?
C8CE:90 1A	39 BCC BINACIA
C8D0:	40 *
C8D0: BD B8 03	41 BINEND LDA DELAYFLG, X
C8D3:29 C0	
C8D5:F0 0E	43 BEQ BINEND1 ; IF SO, LET THE MONITOR DO IT
C8D7:A5 27	44 LDA CHARACTER ; IF NOT, SET FLAG IF
C8D9:C9 E0	45 CMP #\$E0 ; THIS IS A LOWERCASE CHAR
C8DB:90 08	46 BCC BINEND1 ; FOR INPUT BUFFER CORRECTION
C8DD: BD B8 04	47 LDA STATEFLG, X ; (CIRCUMVENT APPLE MONITOR)
C8E0:09 40	48 ORA #\$40
C8E2:9D B8 04	49 STA STATEFLG, X
C8E5:	50 *
C8E5:28	51 BINENDI PLP
C8E6:F0 D0	52 BEQ BASICEXIT ; BRANCH IF NOT CIC MODE
C8E8:D0 CB	53 BNE CICEXIT ; <always> CHECK TO SEE IF WE</always>
C8EA:	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

C	8F4:F0	DA		60		BEQ	BINEND	;SKIP IF NOT CIC MODE
C	8F6:20	D1	C9	61		JSR	CKINPUT	;LOOK FOR INPUT STREAM SPECIAL CHARS
C	8F9:4C	DO	C8	62		JMP	BINEND	; and the state of
Ø	SFC:			63	*******	*****	********	*****
c	SFC:			64	* SIC. F	PC BA	SIC OUTPUT	ROUTINE *
C	SFC:							****
	8FC:20	14	CB		SEROUT	JSR	CMDSEOCK	CHECK FOR A COMMAND SEQUENCE
	8FF: B0		0.0	67	OBROOM	BCS		
7		-						; BRANCH IF WE WERE IN COMMAND MODE
	901:A5	21		68 69		LDA PHA	CHARACTER	;SAVE CHAR ON STACK
			~ ~					
	904:BD		07	70		LDA	20/10/10/10 miles	; IF VIDEO OR TABBING ENABLED,
	907:29			71		AND	#\$C0	; DON'T MESS WITH THE CURSOR
	909:D0	16		72	13	BNE	TABCHECK	
	:90B:			73	*			
	90B:A5			74		LDA	CH	;CHECK FOR COMMA TABBING
1	:90D:F0	42		75		BEQ	NOTAB	; IF CH=0, THERE WAS NO TAB OR COMMA
	:90F:C9	10.01		76		CMP	#8	;INTEGER BASIC COMMA?
1	911:F0	04		77		BEQ	COMMA	
	:913:C9	10		78		CMP	#16	;APPLESOFT COMMA?
	:915:D0	OA		79		BNE	TABCHECK	
Ć	917:09	FO		80	COMMA	ORA	#\$F0	
C	:919:3D	B8	06	81		AND	COLBYTE, X	;SET COL TO PREVIOUS TAB
1	:91C:18			82		CLC		
ζ	91D:65	24		83		ADC	CH	;THEN INCREMENT TO NEXT TAB
c	291F:85	24		84		STA	СН	
c	:921:			85	*			
	2921:			86	*			
(	:921:BD	B8	06	87	TABCHECK	C LDA	COLBYTE, X	
	C924:C5			88		CMP	СН	; IS TABBING NEEDED?
ĺ	926:F0	29		89		BEQ	NOTAB	; IF EQUAL THEN NO TAB NEEDED
¢	928:A9	AO		90		LDA	#\$AO	; SPACE FOR FORWARD TAB
0	292A:90	08		91		BCC	TAB1	
¢	92C:BD	38	07	92		LDA	MISCFLG, X	DON'T BACKSPACE UNLESS TABBING
0	292F:0A			93		ASL	A	; OPTION IS ENABLED
0	2930:10	1F		94		BPL	NOTAB	ACCANESE MINE AND AN
c	2932:A9	88		95		LDA	#\$88	BACKSPACE FOR BACKTAB
	2934:85			96	TAB1	STA	CHARACTER	
	C936:2C		FF	97	INDI	BIT	IORTS	;SET V=1 TO INDICATE TABBING
	2939:08	50		98		PHP		SAVE TABBING INDICATOR
	2939:08	~~		99		BVS	TAB2	; (ALWAYS) AROUND BATCH MOVE ENTRY
	C93C:EA	a		100		NOP	IADZ	(ALWAIS) AROUND BAICH HOVE ENTRI
							********	
	C93D:				* SHORT			The second
	C93D:							dare sono in the set of the sonor
	C93D:			103			\$C93D FOR	the state of the second s
	C93D:						LITY WITH	ATTACK AND A TO A CARACTER AND AN ADDRESS
	C93D:						OCK MOVE.	
	C93D:	FC	-				*********	
	C93D: 2C	58	F.F.		BATCHIN		IORTS	5-18-04-11 - 5-19-2
	0940:50			108		DFB	\$50	;DUMMY BVC
	C941:B8				BATCHOU			; V=0 FOR OUTPUT ENTRY
	C942:AE			110		LDX	MSLOT	
	C945:4C	EF	C9	111		JMP	BATCHIO	
	C948:			1.1.44			*******	
	C948:				* BURP		*	
	C948:						*******	
	C948:20	B5	C9	115	TAB2	JSR	ADJUST	; ADJUST COLUMN COUNT
	C94B:20			116		JSR JMP	OUTPUT2 FORCECR	;DON'T GO TO SCREEN WHEN TABBING ;SHARE SOME CODE

-

6 m

1 3

3

3

6 7

611

5

.

-

10

3

-

3

and the second second second							
951:			118				
951:68			119	NOTAB	PLA		
952:B8			120		CLV		
953:08			121		PHP		;SAVE 'NO TAB' INDICATION
954:85	27		122	NOTAB1	STA	CHARACTER	; (FORCE CR REENTRY)
956:48			123		PHA		
957:20	68	CB	124		JSR	OUTPUT1	;ENTER AFTER CMD SEQ CHECK
95A:20			125		JSR	ADJUST	
95D:68			126		PLA		
95E:49	8D		127		EOR	#\$8D	;WAS IT A CR?
960:0A			128		ASL	A	
961:D0	05		129		BNE	FORCECR	
963:9D		06	130		STA	COLBYTE, X	; IF SO, RESET COLUMN TO O
966:85			131		STA	CH	
968:			132	*			
968: BD	BR	04		FORCECR	LDA	STATEFLG, )	FORCE CR DISABLED?
96B:10			134		BPL	SEREND	anangeneren er er er bet ter er er tet
96D: BD		06	135		LDA		FORCE CR IF LIMIT REACHED
960: BD		00	136		BEQ	SEREND	; (FOR P8 POKE COMPATIBILITY)
972:18	00		137		CLC		191.00
973:FD	BR	06	138		SBC	COLBYTE, X	
976:A9		00	139		LDA	#\$8D	
978:90			140		BCC	NOTAB1	;BRANCH TO FORCE CR
97A:	Da		141	*	500	ito artis i	services and approximation and the second
97A:28				SEREND	PLP		
97B:70	A4		143		BVS	TABCHECK	; BRANCH IF TABBING
97D:			144	*			
97D: BD	38	07	145		LDA	MISCFLG, X	;DON'T MESS WITH CURSOR
980:30		120.00	146		BMI	SEREND2	; WHEN VIDEO IS ON
982:BC		06	147		LDY	COLBYTE, X	The state of the second s
985:0A			148		ASL	A	
:986:30	OE		149		BMI	SETCH	;SET CH TO VALUE OF COL FOR TABBING
988:98			150		TYA		internets on mitting a new contraction
989:A0	00		151		LDY	#0	
98B:38	00		152		SEC	π0	
:98C:FD	38	06	153		SBC	PWDBYTE, X	· Series Series (1) Series (1) Series (1)
98F:C9		00	154		CMP	#SF8	WITHIN 8 CHARS OF PWIDTH?
991:90	10000		155		BCC	SETCH	
993:69			156		ADC	#\$27	; IF SO, ADJUST TO WITHIN 8 OF 40
995:A8			157		TAY	11 4 to 1	nenanane ver
996:84	24			SETCH	STY	СН	
998:	24		159		J.1	Chi	
998:4C	BR	CB		SEREND2	TMD	BASICEXIT	THAT'S ALL
99B:	20	00	161	100 100 100 100 100 100 100 100 100 100	Unit	DHOTOOLT	Junt 5 mil
			100				
99B:						*******	
99B:			163			RY ROUTINE	
	-		1000				
998:8E					STX	MSLOT SLOT16	
99E:84			166				
			167		LDA	#0	
C9A2:9D		05	168		STA	STSBYTE, X	
C9A5:60			169		RTS		
C9A6:			170			********	
						PRINTER WID	
C9A6:							
C9A6: C9A6: C9A6: C9A6: 29			173			\$29	

C9A8:50			176	1	DFB	\$50	;80 COLUMNS
C9A9:84			177		DFB	\$84	;132 COLUMNS
C9AA:			178	*******	*****	********	*
C9AA:			179	* PASCAL	WRIT	TE ROUTINE	*
C9AA:			180	* (DOUBL	ES AS	PASCAL	*
C9AA:			181	* 1.0 E	NTRY	POINT)	*
C9AA:						T SCOAA-	*
C9AA:						*********	
C9AA:85	27					STA CHARACT	
C9AC:20	-	00	185		JSR	PENTRY	
					JSR	OUTPUT	
C9AF:20			186		JSR		LOAD X-REG WITH ERROR BYTE & RTS
C9B2:4C	A3	C8	187		JMP	PASEXIT	LOAD X-REG WITH ERROR BITE & RTS
C9B5:			188				AAAATOO MAAATAA TOTT OLI OLI OLI OLI OLI
C9B5:						********	
C9B5:			190	* COLUMN	ADJU	JST ROUTINE	
C9B5:						MODES ONLY	
C9B5:			192	******	****	********	******
C9B5:A5	27		193		LDA		
C9B7:49	08		194		EOR	#\$08	; BACKSPACE?
C9B9:0A			195		ASL	A	
C9BA:FO	04		196		BEQ	DECRCOL	; IF SO, DECREMENT COLUMN
C9BC:49	EE		197		EOR	#\$EE	;DELETE? (\$FF, RUB)
C9BE:DO	09		198		BNE	CTRLTST	
C9C0:DE	<b>B8</b>	06	199	DECRCOL	DEC	COLBYTE, X	;DECREMENT COLUMN COUNT
C9C3:10			200		BPL	ADJRTS	
C9C5:9D		06	201		STA		;DON'T ALLOW TO GO BELOW O
C9C8:60	50	00	-		RTS	CODDITATA	Joon I maden ie de ballen e
C9C9:C9	co		_	CTRLTST		#\$CO	DON'T INCREMENT COLUMN COUNT FOR
C9CB:B0	1000		204	CINDIGI	BCS	ADJRTS	; CONTROL CHARACTERS
C9CD:FE			205		INC	COLBYTE, X	
C9D0:60	20		206		RTS		
C9D0:00				*******		*********	*****
C9D1:							PECIAL INPUT CHARS *
C9D1:							*****
C9D1:BD	20	07		CKINPUT		MISCFLG, X	
C9D4:29			211	CRIMPOL	AND		; INPUT CTL CHARS ENABLED?
C9D4:29			212		BEO	CIEND	, THEOT CID CHARG DATIONED.
C9D8:	10		213		PPÅ	CIBND	
	-	~	214		LDA	STATEFLG,	v
C9D8:BD	1226						
C9DB:A4			215		LDY	CHARACTER	
C9DD:C0			216		CPY	#\$94	;CTL-T?
C9DF:D0			217		BNE	CKINPUT1	
C9E1:09			218		ORA		;SET TERMINAL MODE
C9E3:D0	06		219		BNE	CKINPUT2	; <always></always>
C9E5:			220				
C9E5:C0	92		221	CKINPUTI		#\$92	;CONTROL-R?
C9E7:D0			222			CIEND	TOTOL AND A COLORADOR AND A COLORADOR
C9E9:29	7F		223		AND	#\$7F	;RESET TERMINAL MODE
C9EB:9D	B8	04	224	CKINPUT2	2 STA	STATEFLG,	X
C9EE:60			225	CIEND	RTS		
C9EF:			226	*			

CSPE:         228         CHN SSC.TERM           CSPE:         1           CSPE:         2         -           CSPE:         3         APPLE II SSC FIRMWARE         -           CSPE:         5         BY LARRY KENYON         -           CSPE:         5         BY LARRY KENYON         -           CSPE:         7         -APRIL 1981-         -           CSPE:         8         (C) COPYRIGHT 1981 BY APPLE COMPUTER, INC. *         -           CSPE:         10         -         -         -           CSPE:         11         -         -         -           CSPE:         12         SHORT DELOCK MOVE         -         -           CSPE:         13         -         -         -           CSPE:         14         BATCHIO TXA         -         -           CSPE:         15         ASL A         -         -           CSPF:0A         16         ASL A         -         -           CSPF:0A         0         20         LDA HO         -           CSPF:0A         0         20         LDA HO         -           CSPF:0A         25         LDA (ALL): <t< th=""><th></th><th></th><th></th><th></th><th></th></t<>					
Corr         2 *         *           Corr         3 * APPLE II SSC FIRMWARE         *           Corr         4 *         *           Corr         5 *         BY LARRY KENYON         *           Corr         5 *         BY LARRY KENYON         *           Corr         7 *         -APRIL 1981-         *           Corr         7 *         -APRIL 1981-         *           Corr         9 * (C) COPYRIGHT 1981 BY APPLE COMPUTER, INC. *         *           Corr         9 * (C) COPYRIGHT 1981 BY APPLE COMPUTER, INC. *         *           Corr         1 *         *         *           Corr         2 *         *         *           Corr         2 *         *         *           Corr         1 *         *         *           Corr         2 *         FVS	C9EF:			228	CHN SSC.TERM
CSUE: 3 * APPLE II SSC FIRMWARE * CSEF: 3 * APPLE II SSC FIRMWARE * CSEF: 4 * CSEF: 5 BY LARRY KENYON * CSEF: 6 * CSEF: 6 * CSEF: 7 * APRIL 1981- * CSEF: 8 * CC COPYRIGHT 1981 BY APPLE COMPUTER, INC. * CSEF: 10 * CSEF: 10 * CSEF: 11 ** CSEF: 12 * SHORT BLOCK MOVE * CSEF: 12 * SHORT BLOCK MOVE * CSEF: 13 ** CSEF: 14 ** CSEF: 14 ** CSEF: 15 *	C9EF:			1	*******
CSEP:       4 *       *         CSEP:       5 * BY LARRY KENYON       *         CSEP:       7 * -APRIL 1981-       ************************************	C9EF:			2	* 200000 / 1000 000 / 1000 * A 1000 000 00 00 00 00 00 00 000
CSPF:         5 *         BY LARRY KENYON         *           CSEF:         6 *         *           CSFF:         7 *         -APRIL 1981-         *           CSFF:         8 *         *         CSFF:           CSFF:         9 *         C) COPTRICHT 1981 BY APPLE COMPUTER, INC. *         *           CSFF:         10 *         *         *         *           CSFF:         11 ***********************************	C9EF:			3	* APPLE II SSC FIRMWARE *
CSPEF:         5 * BY LARRY KENYON         *           CSEF:         6 *         *           CSEF:         7 * -APRIL 1981-         *           CSEF:         8 *         *           CSEF:         9 * (C) COPTRIGHT 1981 BY APPLE COMPUTER, INC. *           CSEF:         10 *         *           CSEF:         11 ***********************************	C9EF:			4	· The second
C9EF:       6 *       *         C9EF:       7 * -APRIL 1981-       ************************************				5	* BY LARRY KENYON *
CSPF:       7 * -APRIL 1981-       ************************************				6	*
Cype:       8 *       *         Cype:       9 * (C) COPYRIGHT 1981 BY APPLE COMPUTER, INC. *         Cype:       11 ***********************************					* -APRIL 1981- **********
C9EF:       10 *       *         C9EF:       11 ***********************************					*
CSPE:       11         CSPE:       12       * SHORT BLOCK MOVE *         CSPE:       13       ************************************	C9EF:			9	* (C) COPYRIGHT 1981 BY APPLE COMPUTER, INC. *
CSPF:       12 * SHORT BLOCK MOVE *         CSPF:       13 ************************************	C9EF:			10	*
C9EF:       13       ************************************	C9EF:			11	*******
CSPE:BA       14 BATCHIO TXA         CSPE:BA       14 BATCHIO TXA         CSPE:BA       15 ASL A         CSPE:BA       17 ASL A         CSPE:BA       17 ASL A         CSPE:BA       18 ASL A         CSPE:BA       19 STA SLOTI6         CSPE:BA       00         CSPE:AD       00         CSPE:BA       25         LDA       (A1L).Y         CSPE:BA       30         CSPE:AD       02         CAD:BS       27         CS       SR         CAD:BS       27         CAD:BS       33         CAC:20 D2 CC       23         CADC:20 D2 CA       32       MOVIN         CADC:20 D2 CA       32       MOVIN         CAD:BS       35       LDP #0         CAL1:B9	C9EF:			12	* SHORT BLOCK MOVE *
C9FF:0A         14         BATCHIO         TXA           C9F0:0A         15         ASL         A           C9F0:0A         16         ASL         A           C9F2:0A         17         ASL         A           C9F2:0A         17         ASL         A           C9F2:0A         18         ASL         A           C9F2:0A         18         ASL         A           C9F2:0A         18         ASL         A           C9F2:0A         0         20         LDA         #O           C9F8:0D         B8 05         21         STA         STSBYTE,X         ; ZERO ERROR INDICATION           C9FD:AO         02         MOVOUT         LDA         #O         (SPF):AO         02         MOVOUT           C9FD:AO         02         MOVOUT         LDA         #O         (SPF):AO         02         #O           C9FD:AO         02         K         MOVOUT         LDA         #O         (SPF):AO         02         #O           C9FD:AO         02         CC         ZO         ACAD         ; SEND IT OUT THE ACIA         CAD           CAD0:20         CC         A3         MOVIN         JS	C9EF:			13	*******
C9F1:0A       16       ASL A         C9F2:0A       17       ASL A         C9F2:0A       18       ASL A         C9F3:0A       18       ASL A         C9F4:85       26       19       STA SLOTI6         C9F6:0A       00       20       LDA #0         C9F6:0A       00       20       LDA #0         C9F0:0F       22       BVS MOVIN       C9F0:0F         C9FD:0       23 *       C       GYDIAO         C9FD:10       23 *       C       C         C9FD:10       23 *       C       GYDIAO         C9FD:10       23 *       GYDIAO       GYDIAO         C9FD:10       23 *       GYDIAO       GYDIAO         C9F1:10       3C       C       MOVIN         CA03:20 C2       C2       GYDIAOUT       SEND IT OUT THE ACIA         CA06:20 BA FC       31 *       GADE:60       GO RTS         CA00:20 C2       C3 3       BCC MOVIN       GATAOUT         CA11:30 G				14	BATCHIO TXA
C9F2:0A       17       ASL A         C9F2:0A       18       ASL A         C9F3:0A       18       ASL A         C9F3:0B       00       20       LDA #0         C9F8:0D       B8 05       21       STA SLOT16         C9F8:0D       DF       21       STA STSFYTE,X;ZERO ERROR INDICATION         C9F8:0D       DF       22       BVS MOVIN         C9F0:       23 *       C       CPT:         C9F1:10       24       MOVUT LDY #0       C         C9F1:13       C2       SC       DA (ALL),Y; GET BUFFER DATA         CA01:25       Z       DA ACIAOUT; SEND IT OUT THE ACIA       CA06:20         CA06:20       BA FC       Z       MOVIN       CA0F:00         CA01:20       D2 CC       31 *       C       CA0F:00         CA11:189       88 CO       34       LDA RDREG,Y       CA14:10       CA11:10         CA11:19       88 CO       34       LDA RDREG,Y       CA14:10       CA14:10       GET         CA11:29       BA FC       37       JSR NXTA1       CA12:10       GET       SEC         CA11:29       BA FC       37       JSR NXTA1       CA14:10       GET       GET	C9F0:0A			15	ASL A
C9F3:0A       18       ASL A         C9F6:A9       0       STA SLOT16         C9F6:A9       00       2       DAA #0         C9F6:A9       0 F       22       BVS MOVIN         C9F0:A0       00       24       MOVOUT LDY #0         C9FD:A0       00       24       MOVOUT LDY #0         C9FD:A0       00       24       MOVOUT LDY #0         C9FF:B1       3C       25       LDA (ALL), Y ;GET BUFFER DATA         CA01:85       7       26       STA CHARACTER         CA03:20       02 CC       27       JSR ACLAOUT ;SEND IT OUT THE ACIA         CA06:20       BA FC       28       JSR NXTA1         CA06:20       D2 CC       29       BCC MOVOUT         CA06:20       D2 CA       3       BCC MOVIN         CA11:8:0       BS CO       34       LDA RDREG, Y         CA14:A0       00       35       LDY #0         CA16:91       3C       36       STA (A1L),Y ;PUT ACIA DATA INTO BUFFER	C9F1:0A			16	ASL A
C9F4:85 26       19       STA SLOT16         C9F6:90       20       LDA #0         C9F8:90       88 05       21       STA STSBYTE, X ; ZERO ERROR INDICATION         C9FB:70       0F       22       BVS MOVIN         C9FD:70       23 *         C9FD:70       24 MOVOUT LDY #0         C9FD:71       26         C37       ZARACTER         CA01:85 27       26         C30:20       D2 CC         CA03:20       02 CC         CA03:20       D2 CC         CA06:20       BA FC         CA00:20       D2 CA         32       MOVIN         CA0C:20       D2 CA         33       BCC MOVIN         CA11:89       BC         CA14:A0       00         35       LDY #0         CA16:91       3C         CA11:89       BFC         38       BCC MOVIN         CA16:91       3C         CA11:20       BFC         39       RTS         C				17	ASL A
C9F4:85 26       19       STA SLOT16         C9F6:90       20       LDA #0         C9F8:90       88 05       21       STA STSBYTE, X ; ZERO ERROR INDICATION         C9FB:70       0F       22       BVS MOVIN         C9FD:70       23 *         C9FD:70       24 MOVOUT LDY #0         C9FD:71       26         C37       ZARACTER         CA01:85 27       26         C30:20       D2 CC         CA03:20       02 CC         CA03:20       D2 CC         CA06:20       BA FC         CA00:20       D2 CA         32       MOVIN         CA0C:20       D2 CA         33       BCC MOVIN         CA11:89       BC         CA14:A0       00         35       LDY #0         CA16:91       3C         CA11:89       BFC         38       BCC MOVIN         CA16:91       3C         CA11:20       BFC         39       RTS         C					
C9F8:9D       B8 05       21       STA STSBYTE,X ; ZERO ERROR INDICATION         C9FD:       23       *         C9FD:       23 *         C9FD:       23 *         C9FD:       23 *         C9FD:       23 *         C9FD:       23 *         C9FD:A0 00       24 MOVOUT LDY #0         C9FD:A0 02       25 LDA (AIL),Y ;GET BUFFER DATA         CA01:85 27       26 STA CHARACTER         CA03:20 02 CC       27 JSR ACLAOUT ;SEND IT OUT THE ACIA         CA06:20 BA FC       28 JSR NXTA1         CA06:0 30 RTS       30 RTS         CA0C:       31 *         CA0C:20 D2 CA 32 MOVIN JSR SRIN         CA11:B9 88 CO       34 LDA RDREG,Y         CA14:A0 00       35 LDY #0         CA16:91 3C       36 STA (AIL),Y ;PUT ACIA DATA INTO BUFFER         CA18:20 BA FC       38 BCC MOVIN         CA18:100 F       38 B	C9F4:85 :	26		19	STA SLOT16
C9F8:9D       B8 05       21       STA STSBYTE,X ; ZERO ERROR INDICATION         C9FD:       23       *         C9FD:       23 *         C9FD:       23 *         C9FD:       23 *         C9FD:       23 *         C9FD:       23 *         C9FD:A0 00       24 MOVOUT LDY #0         C9FD:A0 02       25 LDA (AIL),Y ;GET BUFFER DATA         CA01:85 27       26 STA CHARACTER         CA03:20 02 CC       27 JSR ACLAOUT ;SEND IT OUT THE ACIA         CA06:20 BA FC       28 JSR NXTA1         CA06:0 30 RTS       30 RTS         CA0C:       31 *         CA0C:20 D2 CA 32 MOVIN JSR SRIN         CA11:B9 88 CO       34 LDA RDREG,Y         CA14:A0 00       35 LDY #0         CA16:91 3C       36 STA (AIL),Y ;PUT ACIA DATA INTO BUFFER         CA18:20 BA FC       38 BCC MOVIN         CA18:100 F       38 B				20	
C9FD:       23 *         C9FD:A0 00       24 MOVOUT LDY #0         C9FF:B1 3C       25 LDA (A1L),Y ;GET BUFFER DATA         CA01:85 27       26 STA CHARACTER         CA03:20 02 CC       27 JSR ACIAOUT ;SEND IT OUT THE ACIA         CA06:20 BA FC       28 JSR NXTA1         CA06:20 DA FC       29 BCC MOVOUT         CA08:60       30 RTS         CA00:20 D2 CA       31 *         CA00:20 D2 CA       32 MOVIN JSR SRIN         CA0C:20 D2 CA       32 MOVIN JSR SRIN         CA0C:20 D2 CA       32 MOVIN JSR SRIN         CA0C:20 D2 CA       32 MOVIN JSR SRIN         CA11:B9 88 C0       34 LDA RDREG,Y         CA16:91 3C       36 STA (A1L),Y ;PUT ACIA DATA INTO BUFFER         CA18:20 BA FC       37 JSR NXTA1         CA18:90 EF       38 BCC MOVIN         CA11:       40 *         CA12:       40 *         CA18:0       40 *         CA18:0       40 *         CA12:1:0<	C9F8:9D H	B8	05	21	STA STSBYTE, X ; ZERO ERROR INDICATION
C9FD:A0       00       24 MOVOUT       LDY #0         C9FF:B1       3C       25       LDA       (A1L),Y       ;GET BUFFER DATA         CA01:85       27       26       STA       CHARACTER         CA03:20       02 CC       27       JSR       ACIAOUT       ;SEND IT OUT THE ACIA         CA06:20       BA FC       28       JSR       NXTA1         CA06:20       D2 CC       29       BCC       MOVOUT         CA06:20       D2 CA       32       MOVIN       JSR         CA06:20       D2 CA       32       MOVIN       JSR         CA00:30       RTS       STA       CANC:20       D2 CA         CA00:40       05       LDY #0       CA14:40       CA14:40       STA         CA14:40       00       35       LDY #0       CA16:91       36       STA         CA16:91       3C       36       STA       (A1L),Y       ; PUT ACIA DATA INTO BUFFER         CA16:91       3C       36       STA       (A1L),Y       ; PUT ACIA DATA INTO BUFFER         CA16:91       3C       36       STA       (A1L),Y       ; PUT ACIA DATA INTO BUFFER         CA16:91       3C       37       JSR	C9FB:70 (	OF		22	BVS MOVIN
C9FF:B1 3C       25       LDA (A1L),Y       ;GET BUFFER DATA         CA01:85       27       26       STA CHARACTER         CA03:20       02 CC       27       JSR ACIAOUT       ;SEND IT OUT THE ACIA         CA06:20       BA FC       28       JSR NITA1         CA09:90       F2       29       BCC MOVOUT         CA08:60       30       RTS         CA00:20       D2 CA       32 MOVIN JSR SRIN         CA00:20       D2 CA       32 MOVIN JSR SRIN         CA00:20       D2 CA       32 MOVIN JSR SRIN         CA00:90       FB       33       BCC MOVIN         CA11:B9       88 CO       34       LDA RDREG, Y         CA14:A0       00       35       LDY #0         CA18:20       BA FC       37       JSR NXTA1         CA18:10       BF       38       BCC MOVIN         CA18:10       EF       38       BCC MOVIN         CA18:10       BF       38       BCC MOVIN         CA18:10       FF       38       BCC MOVIN         CA18:10       EF       38       BCC MOVIN         CA18:10       FF       38       BCC MOVIN         CA18:10       I <td>C9FD:</td> <td></td> <td></td> <td>23</td> <td>An anne of the second second build and a second sec</td>	C9FD:			23	An anne of the second second build and a second sec
CA01:85       27       26       STA       CHARACTER         CA03:20       02       CC       27       JSR       ACIAOUT       ; SEND IT OUT THE ACIA         CA06:20       BA       FC       28       JSR       NXTA1         CA06:20       BA       FC       29       BCC       MOVOUT         CA08:60       30       RTS         CA0C:       31       *         CA0C:20       D2       CA       32       MOVIN       JSR       SRIN         CA0C:20       D2       CA       32       MOVIN       JSR       SRIN         CA11:B9       88       CO       34       LDA       RDREG, Y         CA14:A0       00       35       LDY #0       CA16:91       3C       36       STA (A1L), Y       ;PUT ACIA DATA INTO BUFFER         CA16:91       3C       36       STA (A1L), Y       ;PUT ACIA DATA INTO BUFFER       CA16:91       38       BCC       MOVIN         CA16:91       3C       36       STA (A1L), Y       ;PUT ACIA DATA INTO BUFFER       CA16:10       STA       CA12:10       STA       CA12:10       STA       CA12:10       STA       STA       CA12:10       STA       STERMINAL MODE ROUTINES	C9FD: A0	00		24	MOVOUT LDY #0
CA03:20 02 CC       27       JSR ACIAOUT       ; SEND IT OUT THE ACIA         CA06:20 BA FC       28       JSR NXTA1         CA09:90 F2       29       BCC MOVOUT         CA08:60       30       RTS         CA0C:20 D2 CA       32 MOVIN       JSR SRIN         CA0F:90 FB       33       BCC MOVIN         CA14:A0 00       35       LDY #0         CA16:91 3C       36       STA (A1L),Y         CA16:91 3C       36       STA (A1L),Y         CA18:20 BA FC       37       JSR NXTA1         CA18:20 BA FC       34       TERMINAL         CA18:20 BA       40 *       *         CA18:20 BA       57       JSR NXTA1         CA18:20 BA       58       BCC MOVIN         CA12:40       *       *         CA12:41       *       *         CA12:40       * <td< td=""><td>C9FF:B1</td><td>3C</td><td></td><td>25</td><td>LDA (A1L), Y ;GET BUFFER DATA</td></td<>	C9FF:B1	3C		25	LDA (A1L), Y ;GET BUFFER DATA
CA06:20 BA FC       28       JSR NXTA1         CA09:90 F2       29       BCC MOVOUT         CA08:60       30       RTS         CA0C:       31       *         CA0C:20 D2 CA       32 MOVIN JSR SRIN         CA0F:90 FB       33       BCC MOVIN         CA11:B9 88 C0       34       LDA RDREG,Y         CA14:A0 00       35       LDY #0         CA16:91 3C       36       STA (A1L),Y ;PUT ACIA DATA INTO BUFFER         CA18:20 BA FC       37       JSR NXTA1         CA18:20 BA FC       39       RTS         CA12:       40       *         CA12:       40       *         CA12:       40       *         CA12:       41       *         CA12:       42       *         CA12:       43       * TERMINAL MODE ROUTINES *         CA12:       44       *         CA12:       44       *         CA12:       48 *	CA01:85	27		26	STA CHARACTER
CA09:90 F2       29       BCC MOVOUT         CA0B:60       30       RTS         CA0C:       31 *         CA0C:20 D2 CA       32 MOVIN JSR SRIN         CA0C:20 D2 CA       32 MOVIN JSR SRIN         CA0F:90 FB       33       BCC MOVIN         CA11:B9 88 CO       34       LDA RDREG,Y         CA14:A0 00       35       LDY #0         CA16:91 3C       36       STA (A1L),Y ;PUT ACIA DATA INTO BUFFER         CA18:20 BA FC       37       JSR NXTA1         CA11:60       39       RTS         CA12:       40 *          CA12:       40 *          CA12:       40 *          CA12:       41 ************************************	CA03:20 0	02	CC	27	JSR ACIAOUT ;SEND IT OUT THE ACIA
CAOB:60       30       RTS         CAOC:       31 *         CAOC:20       D2 CA       32 MOVIN         CAOF:90       FB       33       BCC         CAOF:90       FB       33       BCC         CAOF:90       FB       33       BCC         CAOF:90       FB       33       BCC         CAI1:19       88 CO       34       LDA         CA14:40       00       35       LDY #0         CA16:91       3C       36       STA (A1L),Y ;PUT ACIA DATA INTO BUFFER         CA18:20       BA FC       37       JSR NXTA1         CA18:190       EF       38       BCC MOVIN         CA16:0       39       RTS         CA11:       40       *         CA12:0       30       *         CA12:10       40       *         CA12:10       43       * TERMINAL MODE ROUTINES       *         CA12:10       44       *       *         CA12:10       31       47       BPL TERMETS ;IF NOT, A SIMPLE RTS WILL DO. • .         CA23:10       31       47       BPL TERMETS ;IF NOT, A SIMPLE RTS WILL DO. • .         CA23:10       32       48       * </td <td>CA06:20 H</td> <td>BA</td> <td>FC</td> <td>28</td> <td>JSR NXTA1</td>	CA06:20 H	BA	FC	28	JSR NXTA1
CAOC:       31 *         CAOC:20 D2 CA       32 MOVIN JSR SRIN         CAOF:90 FB       33 BCC MOVIN         CA14:10       34 LDA RDREG,Y         CA14:20 B8 CO       34 LDA RDREG,Y         CA14:30 OO       35 LDY #0         CA16:91 3C       36 STA (A1L),Y ;PUT ACIA DATA INTO BUFFER         CA18:20 BA FC       37 JSR NXTA1         CA18:90 EF       38 BCC MOVIN         CA16:91 3C       36 STA (A1L),Y ;PUT ACIA DATA INTO BUFFER         CA18:20 BA FC       37 JSR NXTA1         CA16:90 EF       38 BCC MOVIN         CA16:100 G       39 RTS         CA1E:       40 *         CA1E:       40 *         CA1E:       41 ************************************	CA09:90 H	F2		29	BCC MOVOUT
CAOC: 20       D2       CA       32       MOVIN       JSR       SRIN         CAOF: 90       FB       33       BCC       MOVIN         CA11: B9       88       C0       34       LDA       RDREG, Y         CA14: A0       00       35       LDY       #0         CA16: 91       3C       36       STA       (A1L), Y       ; PUT ACIA DATA INTO BUFFER         CA18: 20       BA       FC       37       JSR       NXTA1         CA18: 90       EF       38       BCC       MOVIN         CA10: 60       39       RTS         CA1E:       40       *         CA1E:       40       *         CA1E:       41       ************************************	CAOB:60			30	RTS
CAOF:90 FB       33       BCC MOVIN         CA11:B9 88 C0       34       LDA RDREG,Y         CA14:A0 00       35       LDY #0         CA16:91 3C       36       STA (All),Y ;PUT ACIA DATA INTO BUFFER         CA18:20 BA FC       37       JSR NXTA1         CA18:20 BA FC       39       RTS         CA16:00       39       RTS         CA1E:       40 *         CA1E:       40 *         CA1E:       41 ************************************	CAOC:			31	<ul> <li>Interview of a second se</li></ul>
CA11:B9 88 C0       34       LDA RDREG,Y         CA14:A0 00       35       LDY #0         CA16:91 3C       36       STA (A1L),Y ;PUT ACIA DATA INTO BUFFER         CA18:20 BA FC       37       JSR NXTA1         CA18:20 BA FC       37       JSR NXTA1         CA18:20 BA FC       39       RTS         CA12:       40 *       CMOVIN         CA1E:       40 *         CA1E:       40 *         CA1E:       42 *         42 *       *         CA1E:       43 * TERMINAL MODE ROUTINES *         CA1E:       44 *         CA1E:       45 ************************************			CA		
CA14:A0 00       35       LDY #0         CA16:91 3C       36       STA (A1L),Y       ;PUT ACIA DATA INTO BUFFER         CA18:20 BA FC       37       JSR NXTA1         CA18:20 BA FC       37       JSR NXTA1         CA18:20 BA FC       39       RTS         CA16:91 3C       39       RTS         CA10:60       39       RTS         CA1E:       40 *         CA1E:       40 *         CA1E:       41 **********************************         CA1E:       42 *         CA1E:       43 * TERMINAL MODE ROUTINES *         CA1E:       44 *         CA1E:       45 ************************************					
CA16:91 3C       36       STA (A1L),Y ;PUT ACIA DATA INTO BUFFER         CA18:20 BA FC       37       JSR NXTA1         CA18:90 EF       38       BCC MOVIN         CA10:60       39       RTS         CA1E:       40         CA1E:       40 *         CA1E:       41 ************************************			CO		
CA18:20 BA FC       37       JSR NXTA1         CA1B:90 EF       38       BCC MOVIN         CA1D:60       39       RTS         CA1E:       40       *         CA1E:       40         CA1E:       40         CA1E:       41         41       ************************************					
CA1B:90 EF       38       BCC MOVIN         CA1D:60       39       RTS         CA1E:       40 *         CA1E:       41         ************************************					
CA1D:60       39       RTS         CA1E:       40 *         CA1E:       41 ************************************			FC		
CA1E:40 *CA1E:41 ************************************		EF			
CA1E:41*********************************					
CA1E:42 **CA1E:43 * TERMINAL MODE ROUTINES *CA1E:44 *CA1E:44 *CA1E:45 ************************************					
CA1E:43 * TERMINAL MODE ROUTINES *CA1E:44 *CA1E:44 *CA1E:45 ************************************					
CA1E:       44 *       *         CA1E:       45 ************************************					
CA1E:45 ************************************					
CA1E:BDB80466CHECKTERMLDASTATEFLG, X; HAVEWEENTEREDTERMINALMODE?CA21:103147BPLTERMRTS; IFNOT, ASIMPLERTSWILLDOCA23:48*CA23:49* WEENTERTHE WORLD OFTERMINALMODECA23:50*CA23:A90251TERMMODELDA\$\$02; STARTINSHIFT-LOCKSTARECA25:4852PHA; SHIFTSTAREISSAVED ONSTACKCA26:A97F53LDA\$\$7FCA28:20E2CD54JSRKCMD1; RESETECHO(DEFAULTTOFULLDUP)					
CA21:10 31       47       BPL TERMENTS ; IF NOT, A SIMPLE RTS WILL DO         CA23:       48 *         CA23:       49 * WE ENTER THE WORLD OF TERMINAL MODE         CA23:       50 *         CA23:A9 02       51 TERMMODE LDA \$\$02 ; START IN SHIFT-LOCK STATE         CA25:48       52 PHA ; SHIFT STATE IS SAVED ON STACK         CA26:49 7F       53 LDA \$\$7F         CA28:20 E2 CD       54 JSR KCMD1 ; RESET ECHO (DEFAULT TO FULL DUP)			~ *		
CA23:       48 *         CA23:       49 * WE ENTER THE WORLD OF TERMINAL MODE         CA23:       50 *         CA23:A9 02       51 TERMMODE LDA #\$02 ;START IN SHIFT-LOCK STATE         CA25:48       52 PHA ;SHIFT STATE IS SAVED ON STACK         CA26:49 7F       53 LDA #\$7F         CA28:20 E2 CD       54 JSR KCMD1 ;RESET ECHO (DEFAULT TO FULL DUP)		10.00	04		
CA23:     49 * WE ENTER THE WORLD OF TERMINAL MODE       CA23:     50 *       CA23:A9 02     51 TERMMODE LDA #\$02       ;START IN SHIFT-LOCK STATE       CA25:48     52       CA26:49 7F     53       CA28:20 E2 CD     54       JSR KCMD1     ;RESET ECHO (DEFAULT TO FULL DUP)		31			
CA23:     50 *       CA23: A9 02     51 TERMMODE LDA #\$02     ;START IN SHIFT-LOCK STATE       CA25: 48     52     PHA     ;SHIFT STATE IS SAVED ON STACK       CA26: A9 7F     53     LDA #\$7F       CA28: 20 E2 CD     54     JSR KCMD1     ;RESET ECHO (DEFAULT TO FULL DUP)					
CA23:A9       02       51       TERMMODE LDA #\$02       ;START IN SHIFT-LOCK STATE         CA25:48       52       PHA       ;SHIFT STATE IS SAVED ON STACK         CA26:A9       7F       53       LDA #\$7F         CA28:20       E2       CD       54       JSR       KCMD1       ;RESET ECHO (DEFAULT TO FULL DUP)					
CA25:48         52         PHA         ; SHIFT STATE IS SAVED ON STACK           CA26:A9 7F         53         LDA #\$7F           CA28:20 E2 CD         54         JSR KCMD1         ; RESET ECHO (DEFAULT TO FULL DUP)		~~			
CA26:A9 7F     53     LDA #\$7F       CA28:20 E2 CD     54     JSR KCMD1     ;RESET ECHO (DEFAULT TO FULL DUP)		02			
CA28:20 E2 CD 54 JSR KCMD1 ;RESET ECHO (DEFAULT TO FULL DUP)		-			former owned as only on other
The set work i work i work to tobb bor			-		
CA2D: 35 *		E2	CD		fine as there ( but house to rough bor)
CA2B: A4 24 56 TERMNEXT LOY CH		24		10000	
		-			
CA2D: B1 28 57 LDA (BASL), Y	SHEDI DI	20		57	LUA (DADL),I

-3

-

3

E

. 3

.

51

3

3

60

-

=

CA2F:85 CA31:A9 CA33:25 CA35:D0 CA37:A4 CA39:A9 CA3B:D1 CA3D:D0 CA3F:A5 CA41:91	07 4F 10 24 DF 28		60 61	TERMNEXT1 A	LDA	#\$07	;SAVE SCREEN CHARACTER ;IMPLEMENT A FLASHING UNDERLINE ; FOR A CURSOR
CA33:25 CA35:D0 CA37:A4 CA39:A9 CA3B:D1 CA3D:D0 CA3F:A5	4F 10 24 DF 28		60 61	A			
CA35:D0 CA37:A4 CA39:A9 CA3B:D1 CA3D:D0 CA3F:A5	10 24 DF 28		61		ND	RNDH	- FOR & CURCOR
CA37:A4 CA39:A9 CA3B:D1 CA3D:D0 CA3F:A5	24 DF 28					TOTAT ALL	FOR A CORSOR
CA39:A9 CA3B:D1 CA3D:D0 CA3F:A5	DF 28		10000	В	NE	TERMNEXT3	
CA3B:D1 CA3D:D0 CA3F:A5	28		62	L	DY	CH	
CA3D:DO CA3F:A5	1000		63	L	DA	#\$DF	
CA3F:A5			64	C	MP	(BASL),Y	IS UNDERLINE ON THE SCREEN?
	02		65	В	NE	TERMNEXT2	; IF NOT, PUT IT THERE
CA41:91	27		66	L	DA	CHARACTER	;OTHERWISE USE TRUE SCREEN CHAR
	28		67	TERMNEXT2	STA	(BASL),Y	
CA43:E6	4F		68	I	NC	RNDH	;MAKE IT FLASH, BUT
CA45:E6	4F		69	I	NC	RNDH	;NOT TOO SLOW AND NOT TOO FAST
CA47:			70	*			
CA47:BD	B8	04	71	TERMNEXT3	LDA	STATEFLG,	X ;ARE WE STILL IN TERM MODE?
CA4A:30	09		72	В	MI	TERMACIAIN	; IF SO, GO CHECK ACIA
CA4C:			73	*			
CA4C:20	11	CC	74	TERMEXIT	JSR	RESTORE	;ALWAYS REPLACE OUR CURSOR
CA4F:68			75				CLEAN UP THE STACK
CA50: A9	8D		76	L	DA		RETURN A (CR) TO COVER UP
			77				
CA54:60			78			and the second second	
CA55:			79	*			
CA55:20	FF	CA	80	TERMACIAI	N JS	R INPUT	;ACIA INPUT?
CA58:90	OC		81	В	CC	TERMKBDIN	; IF NOT, GO CHECK KEYBOARD
CA5A:20	11	CC	82	J	SR		RESTORE CURSOR, INPUT->CHARACTER
CA5D: 20	D1	C9	83	J	SR		CHECK FOR CTL-T, CTL-R
							; INPUT->SCREEN ALWAYS
							;
	60	Ch	~~~	-	1.12	TERMINEAT	Automation total and an annual first state of the state
	3E	CC			JSR	GETKBD	; KEYPRESS?
							SKIP IF NOT
							BRANCH IF WE DID A KBD ESCAPE SEO.
	10000	07					SHIFTING ENABLED?
ALC	50	07	10.00			Contraction of the second	, on in the bandbab,
	22					Alexandra and and	
	-						RECOVER TERMSTATE
CA74:A8				-	1000		Proversion and a second account
	27					CHARACTER	
			96	1.1			;1 = SHIFT LETTERS, XLATE NUMBERS
							2 MEANS CAPS LOCK MODE
	- 4						
	98				CMP	#\$9B	; ESC?
						When a state of a transmission	
	00					I like	
					NY		;INCREMENT STATE
Eller aller Parts							, anonalista ornasi
							PUT BACK ON STACK
	28	CA				TERMNEXT	I DA DHOR ON DIAGR
	20	on				- MANITURA	
CONCEAL STREET	C1				ER CN	IP #SC1	; <a?< td=""></a?<>
22227070/2020/00	100		2.2.2.2.				;>Z?
1000 C 100 C 1000							
							; IT'S A LETTER SO TRANSLATE TO LC
						A CONTRACTOR OF A CONTRACTOR O	
	~ 1					- mana so a tall	
					TVA		
	CA4C: CA4C:20 CA4F:68 CA50:A9 CA52:85 CA52:85 CA55:20 CA55:20 CA55:20 CA53:20 CA53:20 CA53:20 CA53:20 CA62:20 CA62:20 CA62:20 CA62:20 CA62:20 CA62:20 CA62:20 CA62:00 CA65:20 CA65:20 CA65:20 CA65:20 CA65:20 CA65:20 CA65:20 CA65:20 CA65:20 CA65:20 CA71:10 CA71:10 CA71:10 CA71:10 CA71:68 CA71:48 CA71:48 CA71:48 CA71:68 CA71:20 CA71:00	CA4C: CA4C:20 11 CA4F:68 ////////////////////////////////////	CA4C: CA4C:20 11 CC CA4F:68 CA5C:A9 8D CA52:85 27 CA54:60 CA55:20 FF CA CA55:20 FF CA CA56:20 A3 CC CA5A:20 11 CC CA5A:20 11 CC CA5A:20 A3 CC CA6A:20 A3 CC CA66:20 A2 CC CA67:0 CC CA70:00 CC CA70:00 CC CA70:C0 01 CA70:F0 20 CA79:F0 20 CA79:F0 20 CA81: CS CA82:98 CA84:4C 2B CA CA81:CS CA82:98 CA84:4C 2B CA CA81:CS CA82:90 08 CA83:C9 DB CA88:C9 DB CA88:C9 DB CA88:C9 20 CA88:C9 20 CA93:C7 CA93:	CA4C:       73         CA4C:20       11       CC       74         CA4F:68       75         CA50:A9       8D       76         CA52:85       27       77         CA54:60       78         CA55:20       FF       CA         CA55:20       FF       CA         CA52:80       0C       81         CA53:90       0C       81         CA54:00       A3       CC         CA51:20       11       CC       82         CA52:20       11       CC       82         CA51:20       D1       C9       83         CA60:20       A3       CC       84         CA63:4C       2B       CA       86         CA66:20       3E       CC       87         CA66:B0       3E       79       92         CA73:68       93       04       94         CA71:10       22       92       92         CA73:68       93       04       94         CA79:F0       20       97       97         CA79:F0       20       97       05         CA79:F0       20       97	CA4C:       73 *         CA4C:20       11       CC       74       TERMEXIT         CA4F:68       75       FE         CA50:49       80       76       II         CA52:85       27       77       S         CA55:20       FF       CA       80       TERMACIAI         CA52:20       D1       C9       83       JJ         CA52:20       D1       C9       83       JJ         CA52:20       D1       C9       83       JJ         CA63:20       A       CC       84       JJ         CA66:20       3E       CC       85       J         CA66:10       B       89       E       C         CA66:20       3E       CC       87       TERMKBIN         CA66:20       3E       C2       J       C         CA66:20       3E       CC       87       F         CA66:20       3E	CA4C:       73 *         CA4C:20       11 CC       74 TERMEXIT JSR         CA4F:68       75       PLA         CA50:49       8D       76       LDA         CA52:85       27       77       STA         CA55:80       77       78 TERMETS RTS       CA52:85         CA55:20       FF       CA       80       TERMACTAIN JS         CA52:20       11 CC       82       JSR         CA51:20       D1 C9       83       JSR         CA52:20       D1 C9       83       JSR         CA51:20       D1 C9       83       JSR         CA51:20       D1 C9       83       JSR         CA61:20       A3 CC       84       JSR         CA66:20       32 CC       87       TERMKBDIN JSF         CA66:10       B6       *       CA66:20         CA66:20       32 CC       87       TERMKBDIN JSF         CA66:20       32 CC       87       TERMKBDIN JSF         CA66:20       32 CC       87       TERMKBDIN JSF         CA66:20       32 CP       92       BPL         CA73:68       93       PLA         CA71:10       22	CA4C:       73 *         CA4C:20       11 CC       74 TERMEXIT JSR RESTORE         CA4C:20       11 CC       74 TERMEXIT JSR RESTORE         CA4C:20       11 CC       74 TERMEXIT JSR RESTORE         CA50:49       8D       76       LDA #\$8D         CA52:85       27       77       STA CHARACTER         CA54:60       78 TERMETS RTS       CASS:20       FF CA       80 TERMACIAIN JSR INPUT         CA58:20       0C       81       BCC TERMEDIN       SR RESTORE         CA52:20       D1 C9       83       JSR CKINPUT         CA60:20       A3 CC       84       JSR SCREENOUT!         CA66:       28       JMP TERMNEXT       CA66:         CA66:       86 *        CA66:20         CA66:20       3E CC       87 TERMKBDIN JSR GETKBD       CA66:20         CA66:20       3E CC       87 TERMINCIN JSR GETKBD       CA66:20         CA66:20       3E CC       87 TERMINCIN JSR GETKBD       CA66:20

CA94:48			116	PHA	A		; PUT STATE BACK ON S	TACK	
CA95:20	68	CB	117	TERMSEND1	JSR	OUTPUT1	;GO OUTPUT		
CA98:4C	2B	CA	118	JMI	P	TERMNEXT			
CA9B:			119	*					
CA98:C9	9B		120	TERMCAP CMI	9	#\$9B	; TWO ESCAPES?		
CA9D: FO	E2		121	BEG	2	TERMINC			
CA9F:C9	во		122	CMI	P	#\$B0	;<0?		
CAA1:90	OA		123	BCC	2	TERMCAP1			
CAA3: C9	BB		124	CMI	P	#\$BB	;>COLON?		
CAA5: BO	06		125	BCS	S	TERMCAP1			
CAA7:			126	*					
CAA7:			127	* ESC <num< td=""><td>BER</td><td>&gt; SO TRANS</td><td>SLATE INTO MISSING AS</td><td>CII CHAR</td><td></td></num<>	BER	> SO TRANS	SLATE INTO MISSING AS	CII CHAR	
CAA7:			128	*					
CAA7:A8			129	TAT	Y				
CAA8: B9	09	CA	130	LDA	A	TRANSLATE	-\$B0,Y		
CAAB:85	27		131	ST	A	CHARACTER			
CAAD: AO	00		132	TERMCAP1 LI	DY	#0	;BACK TO STATE 0		
CAAF: FO	E2		133	BE	2	TERMSEND	; <always></always>		
CAB1:			134	*					
CAB1:C9	9B		135	TERMLOCK CI	MP	#\$9B	; ESC?		
CAB3:DO	DE		136	BN	Е	TERMS END			
CAB5:A0	00		137	LD	Y	#0			
CAB7:FO	C9		138	BE	Q	TERMINC1;	<always></always>		
CAB9:				*					
CAB9:			140	********	***	********	*****		
CAB9:			141	* TRANSLAT	E 1	FABLE	0.00 * 20 00 20 00 000		
CAB9:			142	********	***	********	****		
CAB9:9B			143	TRANSLATE	DFE	8 \$9B	;ESC		
CABA:9C			144			\$9C	;FS		
CABB:9F			145	DF		* * *	;US		
CABC:DB			146			\$DB	;LEFT BRACKET		
CABD: DC			147			\$DC	;LEFT SLASH		
CABE: DF			148	DF			;UNDERSCORE		
CABF:FB			149	DF	В		;LEFT ENCLOSE		
CACO:FC			150			\$FC	; VERTICAL BAR		
CAC1:FD			151	DF	В	ŞFD	;RIGHT ENCLOSE		
CAC2:FE			152	DF	в	\$FE	;TILDE		
CAC3:FF			153	DF	В	ŞFF	;RUB		
CAC4:			154	*					
CAC4:			155	CH	N	SSC.CORE			

-

CAC4:	2 ********	*****
CAC4:	3 *	
CAC4:	4 * APPLE II SSC FIRMWARE	
CAC4:	5 *	*
CAC4:	6 * BY LARRY KENYON	
CAC4:	7 *	SACHARY * N
CAC4:	8 * -JANUARY 1981-	*****
CAC4:	9 *	
CAC4:	10 * (C) COPYRIGHT 1981 BY	APPLE COMPUTER, INC. *
CAC4:	11 *	*
CAC4:	12 ****************	****
CAC4:	13 *	*
CAC4:	14 * CORE SUBROUTINES	* · · · · · · · · · · · · · · · · · · ·
CAC4:	15 *	a second of the side of the second second second
CAC4:	16 **************	*****
CAC4:	17 *************	****
CAC4:	18 * GENERAL PURPOSE WAIT	ROUTINE *
CAC4:	19 ***************	*****
CAC4:	20 *	
CAC4:	21 * WAITMS WAITS FOR [A-R	REG] MILLISECONDS (256 IF A-REG=0)
CAC4:	22 *	
CAC4: A2 CA	23 WAITMS LDX #202	
CAC6:CA		; <don't a="" cross="" let="" loop="" page="" this=""></don't>
CAC7:D0 FD	25 BNE WAITMS1	;5 MICROSECOND LOOP
CAC9:38	26 SEC	
CACA:E9 01	27 SBC #01	
CACC:DO F6	28 BNE WAITMS	
CACE: AE F8 07	29 LDX MSLOT	
CAD1:60	30 RTS	
CAD2:	31 ****************	
CAD2: CAD2:	31 ************************************	READ ROUTINES *
CAD2: CAD2: CAD2: CAD2:	31 ************************************	READ ROUTINES *
CAD2: CAD2: CAD2: CAD2: CAD2:	31 ************************************	READ ROUTINES *
CAD2: CAD2: CAD2: CAD2: CAD2: CAD2:	31 ************************************	READ ROUTINES *
CAD2: CAD2: CAD2: CAD2: CAD2: CAD2: CAD2: CAD2:	31 ************************************	READ ROUTINES * ***********************************
CAD2: CAD2: CAD2: CAD2: CAD2: CAD2: CAD2: CAD2: CAD2:A4 26	31       ************************************	READ ROUTINES *
CAD2: CAD2: CAD2: CAD2: CAD2: CAD2: CAD2: CAD2: CAD2:A4 26 CAD4:B9 89 CO	31       ************************************	READ ROUTINES * ***********************************
CAD2: CAD2: CAD2: CAD2: CAD2: CAD2: CAD2: CAD2: CAD2:A4 26 CAD4:B9 89 C0 CAD7:48	31 ************************************	READ ROUTINES * ***********************************
CAD2: CAD2: CAD2: CAD2: CAD2: CAD2: CAD2: CAD2: CAD2:A4 CAD2:A4 26 CAD4:B9 89 C0 CAD7:48 CAD8:29 20	31       ************************************	READ ROUTINES * ***********************************
CAD2: CAD2: CAD2: CAD2: CAD2: CAD2: CAD2: CAD2: CAD2:A4 26 CAD4:B9 89 C0 CAD7:48 CAD8:29 20 CADA:4A	31       ************************************	READ ROUTINES * ***********************************
CAD2: CAD2: CAD2: CAD2: CAD2: CAD2: CAD2: CAD2: CAD2:A4 26 CAD4:B9 89 C0 CAD4:B9 89 C0 CAD7:48 CAD8:29 20 CAD8:4A CADB:4A	31       ************************************	READ ROUTINES * ***********************************
CAD2: CAD2: CAD2: CAD2: CAD2: CAD2: CAD2: CAD2:A4 CAD4:B9 89 C0 CAD7:48 CAD8:29 20 CAD7:48 CAD8:29 20 CAD3:4A CAD8:4A CAD6:45 35	31       ************************************	READ ROUTINES * ***********************************
CAD2: CAD2: CAD2: CAD2: CAD2: CAD2: CAD2: CAD2: CAD2:A4 CAD2:A4 CAD3:A9 CAD3:A9 CAD3:A9 CAD3:AA CAD5:4A CAD5:4A CAD5:4A CAD5:4A	31       ************************************	READ ROUTINES * ***********************************
CAD2: CAD2: CAD2: CAD2: CAD2: CAD2: CAD2: CAD2: CAD2:A4 26 CAD4:B9 89 C0 CAD7:48 CAD8:29 20 CAD7:48 CAD8:29 20 CAD3:4A CAD8:4A CAD6:85 35 CAD6:68 CADF:29 0F	31       ************************************	READ ROUTINES * ***********************************
CAD2: CAD2: CAD2: CAD2: CAD2: CAD2: CAD2: CAD2: CAD2:A4 26 CAD4:B9 89 C0 CAD7:48 CAD8:29 20 CAD7:48 CAD8:29 20 CAD3:4A CAD8:4A CAD8:4A CAD6:45 35 CADC:85 35 CADC:868 CADF:29 0F CAE1:C9 08	31       ************************************	READ ROUTINES * ***********************************
CAD2: CAD2: CAD2: CAD2: CAD2: CAD2: CAD2: CAD2: CAD2:A4 26 CAD4:B9 89 C0 CAD4:B9 89 C0 CAD4:B9 89 C0 CAD7:48 CAD8:29 20 CAD3:4A CAD8:4A CAD8:4A CAD5:45 35 CAD5:45 35 CAD5:29 0F CAD5:29 0F CAE1:C9 08 CAE3:90 04	31       ************************************	READ ROUTINES * ***********************************
CAD2: CAD2: CAD2: CAD2: CAD2: CAD2: CAD2: CAD2: CAD2: CAD2: CAD2: CAD3: 4 CAD4: 4 CAD4: 4 CAD4: 4 CAD4: 4 CAD4: 4 CAD4: 4 CAD4: 4 CAD4: CAD4: 4 CAD4: CAD4: CAD4: CAD4: CAD4: CAD4: CAD4: CAD4: CAD4: CAD4: CAD4: CAD4: CAD4: CAD4: CAD4: CAD4: CAD4: CAD4: CAD4: CAD2: CAD4:	31       ************************************	READ ROUTINES * ***********************************
CAD2: CAD2: CAD2: CAD2: CAD2: CAD2: CAD2: CAD2: CAD2:A4 26 CAD4:B9 89 C0 CAD4:B9 89 C0 CAD4:B9 89 C0 CAD7:48 CAD8:29 20 CAD3:4A CAD8:4A CAD8:4A CAD5:45 35 CAD5:45 35 CAD5:29 0F CAD5:29 0F CAE1:C9 08 CAE3:90 04	31       ************************************	READ ROUTINES * ***********************************
CAD2: CAD2: CAD2: CAD2: CAD2: CAD2: CAD2: CAD2: CAD2: CAD2: CAD2: A CAD3: CAD3: CAD3: CAD3: CAD3: CAD3: CAD4: CAD3: CAD3: CAD3: CAD3: CAD4: CAD4: CAD3: CAD3: CAD4: CAD5: CAD5: CAD5: CAD5: CAD5: CAD5: CAD5: CAD5: CAD5: CAD5: CAD5: CAD5: CAD5: CAD5: CAD5: CAD5: CAD5: CAD3	31       ************************************	READ ROUTINES * ***********************************
CAD2: CAD2: CAD2: CAD2: CAD2: CAD2: CAD2: CAD2: CAD2: CAD2: CAD2: CAD2:A4 26 CAD4:B9 89 C0 CAD7:48 CAD8:29 20 CAD7:48 CAD8:29 20 CAD3:4A CAD5:4A CAD5:4A CAD5:45 35 CAD5:68 CAD7:29 0F CAE1:C9 08 CAE3:90 04 CAE5:29 07 CAE5:29 07 CAE5:80 02 CAE9:A5 35	31       ************************************	READ ROUTINES * ***********************************
CAD2: CAD2: CAD2: CAD2: CAD2: CAD2: CAD2: CAD2: CAD2:A4 26 CAD4:B9 89 C0 CAD7:48 CAD8:49 20 CAD7:48 CAD8:29 20 CAD3:44 CAD8:4A CAD8:4A CAD8:4A CAD8:4A CAD8:4A CAD6:29 0F CAD2:85 35 CAE1:C9 08 CAE3:90 04 CAE3:90 04 CAE3:29 07 CAE7:B0 02 CAE3:A5 35 CAE8:05 35	31       ************************************	READ ROUTINES * ***********************************
CAD2: CAD2: CAD2: CAD2: CAD2: CAD2: CAD2: CAD2: CAD2: CAD2:A4 26 CAD4:B9 89 C0 CAD4:B9 89 C0 CAD7:48 CAD8:29 20 CAD3:4A CAD8:4A CAD8:4A CAD8:4A CAD2:85 35 CAD6:62 CAD7:29 0F CAE1:C9 08 CAE3:90 04 CAE3:90 04 CAE3:90 04 CAE3:90 7 CAE3:90	31       ************************************	READ ROUTINES * ***********************************
CAD2: CAD2: CAD2: CAD2: CAD2: CAD2: CAD2: CAD2: CAD2: CAD2: CAD2: CAD2: AB CAD3: AB CAD3: AB CAD3: CAD	31       ************************************	READ ROUTINES * ***********************************
CAD2: CAD2: CAD2: CAD2: CAD2: CAD2: CAD2: CAD2: CAD2: CAD2: CAD2: CAD2: CAD2: CAD3:	31       ************************************	READ ROUTINES * ***********************************
CAD2: CAD2: CAD2: CAD2: CAD2: CAD2: CAD2: CAD2: CAD2: CAD2: CAD2: CAD2: CAD2: CAD2: CAD3: 48 CAD8: 48 CAD8: 48 CAD7: 48 CAD8: 48 CAD8: 48 CAD7: 48 CAD8: 48 CAD8: 48 CAD7: 48 CAD8: 48 CAD8: 48 CAD8: 48 CAD7: 48 CAD8: 48 CAE3: 49 CAE3: 48	31       ************************************	READ ROUTINES * ***********************************
CAD2: CAD2: CAD2: CAD2: CAD2: CAD2: CAD2: CAD2: CAD2: CAD2: CAD2: CAD2: CAD2: CAD3:	31       ************************************	READ ROUTINES * ***********************************

CAF7: B9	89	CO	60		LDA	STREG, Y	
CAFA:29	70		61		AND	#\$70	
CAFC:C9			62		CMP	#\$10	;EQU IF TDR EMPTY, DCD, DSR, & CT
CAFE:60			63		RTS		
CAFF:			64	*			
CAFF:			65	******	*****	*******	***
CAFF:			66	* GENERA	L INE	UT ROUTINE	*
CAFF:			67	******	*****	********	***
CAFF:20	D2	CA	68	INPUT	JSR	SRIN	
CB02:90			69		BCC	NOINPUT1	
CB04:			70	*			
CB04: B9	88	CO	71		LDA	RDREG, Y	;GET THE ACIA INPUT
CB07:09		00	72		ORA	#\$80	SET HI BIT FOR BASIC
CB09:C9			73		CMP	#\$8A	;LINEFEED?
CBOB:DO			74		BNE	INPUT2	/==
CBOD:	09		75	*	DNL	INFOIL	
			76		TAY		
CBOD: A8 CBOE: BD	38	07	77		LDA	MISCELG. Y	;SEE IF WE SHOULD EAT IT
CB0E: BD		01	78		AND	#\$20	Jose to the blocker with the
CB11:29 CB13:D0			79		BNE		; IF SO, JUST KEEP IT A SECRET
CB15:98	05		80		TYA	10111-01	, at boy over male it i blocket
CB15:98 CB16:			81	*	110		
				INPUT2	000		;INDICATE DATA
CB16:38			83	INPOIZ	RTS		FINDICALE DATA
CB17:60			83	+	RIS		
CB18:					ara		CARRY OF BAR BOR NO TURIN
CB18:18				NOINPUT			;CARRY CLEAR FOR NO INPUT
CB19:60					RIS		
CB1A:			87			********	
CB1A:			88				
			00	* CEMEDI	AT OT	TUTOD TUTOT	NIP *
CB1A:			12.2			TPUT ROUTIN	
CB1A:			90	******		TPUT ROUTIN	
CB1A: CB1A:			90 91	*******	****	********	
CB1A: CB1A: CB1A:			90 91 92	******** * * START	****		
CB1A: CB1A: CB1A: CB1A:	26		90 91 92 93	******** * * START *	OF C	***********	
CB1A: CB1A: CB1A: CB1A: CB1A:A4		CO	90 91 92 93 94	******** * * START	OF CO	SLOT16	
CB1A: CB1A: CB1A: CB1A: CB1A:A4 CB1C:B9	81	со	90 91 92 93 94 95	******** * * START *	OF CO	SLOT16 DIPSW1,Y	
CB1A: CB1A: CB1A: CB1A: CB1A:A4 CB1C:B9 CB1F:4A	81	со	90 91 92 93 94 95 96	******** * * START *	OF CO LDY LDA LSR	SLOT16 DIPSW1,Y A	**** CK ROUTINE
CB1A: CB1A: CB1A: CB1A: CB1A:A4 CB1C:B9 CB1F:4A CB20:B0	81 36		90 91 92 93 94 95 96 97	******** * * START *	OF CO LDY LDA LSR BCS	SLOT16 DIPSW1,Y A NOCMD	**** CK ROUTINE ;DON'T WORRY ABOUT CMD SEQ FOR SI
CB1A: CB1A: CB1A: CB1A: CB1A: A CB1C: B9 CB1F: 4A CB20: B0 CB22: BD	81 36 88		90 91 92 93 94 95 96 97 98	******** * * START *	OF CO LDY LDA LSR BCS LDA	SLOT16 DIPSW1,Y A NOCMD STATEFLG,J	**** CK ROUTINE ;DON'T WORRY ABOUT CMD SEQ FOR SI X
CB1A: CB1A: CB1A: CB1A: CB1A:A: CB1A:A4 CB1C:B9 CB1F:4A CB20:B0 CB22:BD CB22:29	81 36 88 07		90 91 92 93 94 95 96 97 98 98 99	******** * * START *	OF CO LDY LDA LSR BCS LDA AND	SLOT16 DIPSW1,Y A NOCMD STATEFLG,1 #\$07	**** CK ROUTINE ;DON'T WORRY ABOUT CMD SEQ FOR SI
CB1A: CB1A: CB1A: CB1A: CB1A: CB1A:A4 CB1C:B9 CB1F:4A CB20:B0 CB22:BD CB22:29 CB27:F0	81 36 88 07 05	04	90 91 92 93 94 95 96 97 98 99 100	******** * * START *	OF CO LDY LDA LSR BCS LDA AND BEQ	SLOT16 DIPSW1,Y A NOCMD STATEFLG,3 #\$07 ESCCHECK	**** CK ROUTINE ;DON'T WORRY ABOUT CMD SEQ FOR SI X ;ARE WE IN A COMMAND SEQUENCE?
CB1A: CB1A: CB1A: CB1A: CB1A: A CB1C:B9 CB1F:4A CB20:B0 CB22:BD CB22:29 CB25:29 CB27:F0 CB29:20	81 36 88 07 05 FC	04	90 91 92 93 94 95 96 97 98 99 100 101	******** * * START *	OF CC C LDY LDA LSR BCS LDA AND BEQ JSR	SLOT16 DIPSW1,Y A NOCMD STATEFLG,3 #\$07 ESCCHECK	**** CK ROUTINE ;DON'T WORRY ABOUT CMD SEQ FOR SI X ;ARE WE IN A COMMAND SEQUENCE? ;IF SO, GOTO COMMAND CENTRAL
CB1A: CB1A: CB1A: CB1A: CB1A: CB1A:A4 CB1C:B9 CB1F:4A CB20:B0 CB22:BD CB22:BD CB22:CB2 CB27:F0 CB29:20 CB2C:38	81 36 88 07 05 FC	04	90 91 92 93 94 95 96 97 98 99 100 101	******** * * START *	OF CC C LDY LDA LSR BCS LDA AND BEQ JSR SEC	SLOT16 DIPSW1,Y A NOCMD STATEFLG,3 #\$07 ESCCHECK	**** CK ROUTINE ;DON'T WORRY ABOUT CMD SEQ FOR SI X ;ARE WE IN A COMMAND SEQUENCE?
CB1A: CB1A: CB1A: CB1A: CB1A: CB1A:A4 CB1C:B9 CB1F:4A CB20:B0 CB22:BD CB22:BD CB22:CB CB27:F0 CB29:20 CB29:20 CB22:38 CB20:60	81 36 88 07 05 FC	04	90 91 92 93 94 95 96 97 98 99 100 101 102 103	* START * CMDSEQC	OF CC C LDY LDA LSR BCS LDA AND BEQ JSR	SLOT16 DIPSW1,Y A NOCMD STATEFLG,3 #\$07 ESCCHECK	**** CK ROUTINE ;DON'T WORRY ABOUT CMD SEQ FOR SI X ;ARE WE IN A COMMAND SEQUENCE? ;IF SO, GOTO COMMAND CENTRAL
CB1A: CB1A: CB1A: CB1A: CB1A:A4 CB1C:B9 CB1C:B9 CB1C:B9 CB2C:B0 CB22:B0 CB22:29 CB27:F0 CB22:20 CB22:38 CB2C:38 CB2C:38 CB2C:38	81 36 88 07 05 FC	04	90 91 92 93 94 95 96 97 98 99 100 101 102 103 104	******** * START * CMDSEQC	OF CC LDY LDA LSR BCS LDA AND BEQ JSR SEC RTS	SLOT16 DIPSW1,Y A NOCMD STATEFLG,: #\$07 ESCCHECK CMDPROC	**** CK ROUTINE ;DON'T WORRY ABOUT CMD SEQ FOR SI X ;ARE WE IN A COMMAND SEQUENCE? ;IF SO, GOTO COMMAND CENTRAL ;INDICATE COMMAND
CB1A: CB1A: CB1A: CB1A: CB1A: A: CB1C:B9 CB1C:B9 CB1C:B9 CB1C:B9 CB20:B0 CB22:B0 CB22:B0 CB22:CB2 CB22:CB2 CB20:G0 CB22:A5	81 36 88 07 05 FC 27	04	90 91 92 93 94 95 96 97 98 99 100 101 102 103 104	******** * START * CMDSEQC	OF CO C LDY LDA LSR BCS LDA AND BEQ JSR SEC RTS K LDA	SLOT16 DIPSW1,Y A NOCMD STATEFLG,I #\$07 ESCCHECK CMDPROC CHARACTER	**** CK ROUTINE ;DON'T WORRY ABOUT CMD SEQ FOR SI X ;ARE WE IN A COMMAND SEQUENCE? ;IF SO, GOTO COMMAND CENTRAL ;INDICATE COMMAND
CB1A: CB1A: CB1A: CB1A: CB1A: CB1A: A4 CB1C: B9 CB1F: 4A CB20: B0 CB22: B0 CB22: E0 CB22: 20 CB22: 20 CB22: 38 CB20: 60 CB22: CB22: A5 CB20: 29	81 36 88 07 05 FC 27 7F	04 CD	90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106	******** * START * CMDSEQC	OF CO C LDY LDA LSR BCS LDA AND BEQ JSR SEC RTS C LDA AND	CHARACTER #57F	**** CK ROUTINE ;DON'T WORRY ABOUT CMD SEQ FOR SI ; ;ARE WE IN A COMMAND SEQUENCE? ;IF SO, GOTO COMMAND CENTRAL ;INDICATE COMMAND ;IGNORE HIGH BIT
CB1A: CB1A: CB1A: CB1A: CB1A: CB1A: A: CB1C: B9 CB1F: 4A CB20: B0 CB22: B0 CB22: CB22: CB2	81 36 88 07 05 FC 27 7F 38	04 CD	90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107	******** * START * CMDSEQC	OF CC C LDY LDA LSR BCS LDA AND BEQ JSR SEC RTS K LDA AND CMP	CHARACTER #57F CMDBYTE, X	**** CK ROUTINE ;DON'T WORRY ABOUT CMD SEQ FOR SI X ;ARE WE IN A COMMAND SEQUENCE? ;IF SO, GOTO COMMAND CENTRAL ;INDICATE COMMAND
CB1A: CB1A: CB1A: CB1A: CB1A: CB1A: ACB1C:B9 CB1F:4A CB20:B0 CB22:BD CB22:BD CB22:CB27:F0 CB22:20 CB22:38 CB20:60 CB22: CB22:A5 CB22:A5 CB22:29 CB32:D0 CB35:D0	81 36 88 07 05 FC 27 7F 38 05	04 CD 05	90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107	******** * START * CMDSEQC	OF CC LDY LDA LSR BCS LDA AND BEQ JSR SEC RTS K LDA AND CMP BNE	CHARACTER #\$7F CMDBYT, X A NOCMD STATEFLG, 1 #\$07 ESCCHECK CMDPROC	**** CK ROUTINE ;DON'T WORRY ABOUT CMD SEQ FOR SI X ;ARE WE IN A COMMAND SEQUENCE? ;IF SO, GOTO COMMAND CENTRAL ;INDICATE COMMAND ;IGNORE HIGH BIT ;IS THIS BEGINNING OF A CMD SEQ?
CB1A: CB1A: CB1A: CB1A: CB1A: A4 CB1C: B9 CB1F: 4A CB20: B0 CB22: B0 CB22: 20 CB22: 20 CB22: 38 CB20: 60 CB22: 38 CB20: 60 CB22: CB30: 20 CB32: D0 CB37: FE	81 36 B8 07 05 FC 27 7F 38 05 B8	04 CD 05	90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108	******** * START * CMDSEQC	OF CC LDY LDA LSR BCS LDA AND BEQ JSR SEC RTS CLDA AND CMP BNE INC	CHARACTER #\$7F CMDBYT, X A NOCMD STATEFLG, 1 #\$07 ESCCHECK CMDPROC	**** CK ROUTINE ;DON'T WORRY ABOUT CMD SEQ FOR SI x ;ARE WE IN A COMMAND SEQUENCE? ;IF SO, GOTO COMMAND CENTRAL ;INDICATE COMMAND ;IGNORE HIGH BIT ;IS THIS BEGINNING OF A CMD SEQ? X ;START UP COMMAND MODES
CB1A: CB1A: CB1A: CB1A: CB1A: CB1A: A4 CB1C: B9 CB1F: 4A CB20: B0 CB22: B0 CB22: B0 CB22: 20 CB22: 38 CB20: 60 CB22: A5 CB20: 60 CB22: A5 CB30: 29 CB32: D0 CB32: D0 CB33: E0 CB33: S0	81 36 B8 07 05 FC 27 7F 38 05 B8	04 CD 05	90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110	******** * START * CMDSEQC	OF CC LDY LDA LSR BCS LDA AND BEQ JSR SEC RTS K LDA AND CMP BNE INC SEC	CHARACTER #\$7F CMDBYT, X A NOCMD STATEFLG, 1 #\$07 ESCCHECK CMDPROC	**** CK ROUTINE ;DON'T WORRY ABOUT CMD SEQ FOR SI X ;ARE WE IN A COMMAND SEQUENCE? ;IF SO, GOTO COMMAND CENTRAL ;INDICATE COMMAND ;IGNORE HIGH BIT ;IS THIS BEGINNING OF A CMD SEQ?
CB1A: CB1A: CB1A: CB1A: CB1A: CB1A: A4 CB1C: B9 CB1F: 4A CB20: B0 CB22: BD CB22: BD CB22: CB22: 20 CB22: 20 CB22: 30 CB22: 40 CB22: 40 CB23: 40 CB23: 40 CB23: 40 CB23: 40 CB23: 40 CB23: 40 CB23: 40 CB23: 40 CB33: 40 CB3	81 36 B8 07 05 FC 27 7F 38 05 B8	04 CD 05	90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110	* * START CMDSEQC:	OF CC LDY LDA LSR BCS LDA AND BEQ JSR SEC RTS CLDA AND CMP BNE INC	CHARACTER #\$7F CMDBYT, X A NOCMD STATEFLG, 1 #\$07 ESCCHECK CMDPROC	**** CK ROUTINE ;DON'T WORRY ABOUT CMD SEQ FOR SI x ;ARE WE IN A COMMAND SEQUENCE? ;IF SO, GOTO COMMAND CENTRAL ;INDICATE COMMAND ;IGNORE HIGH BIT ;IS THIS BEGINNING OF A CMD SEQ? X ;START UP COMMAND MODES
CB1A: CB1A: CB1A: CB1A: CB1A: CB1A: ACB20:B0 CB20:B0 CB22:B0 CB22:C0 CB22:20 CB22:20 CB22:38 CB20:60 CB22: CB22:5 CB30:29 CB32:D0 CB32:D0 CB33:FE CB33:68 CB33:60 CB33:30 CB33:30 CB33:30	81 36 B8 07 05 FC 27 7F 38 05 B8	04 CD 05 04	90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111	******** * START CMDSEQCP * ESCCHECI	OF CC LDY LDA LSR BCS LDA BCS LDA BEQ JSR SEC RTS K LDA AND CMP ENE INC SEC RTS	CMMAND CHEX SLOT16 DIPSW1,Y A NOCMD STATEFLG,I #\$07 ESSCHECK CMDPROC CHARACTER #\$7F CMDBYTE,X XOFPCK STATEFLG,I	**** CK ROUTINE ;DON'T WORRY ABOUT CMD SEQ FOR SI x ;ARE WE IN A COMMAND SEQUENCE? ;IF SO, GOTO COMMAND CENTRAL ;INDICATE COMMAND CENTRAL ;ISTHIS BEGINNING OF A CMD SEQ? X ;START UP COMMAND MODES ;INDICATE COMMAND
CB1A: CB1A: CB1A: CB1A: CB1A: CB1A: ACB20:B0 CB22:B0 CB22:B0 CB22:C0 CB22:C0 CB22:C0 CB22:C0 CB22:C0 CB22:C0 CB22:C0 CB22:C0 CB22:C0 CB22:C0 CB22:C0 CB22:C0 CB22:C0 CB32:D0 CB32:C0 C	81 36 88 07 05 FC 27 7F 38 05 88 38	04 CD 05 04	90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 111 112	******** * START CMDSEQCP * ESCCHECI	OF CC LDY LDA LSR BCS LDA AND BEQ JSR SEC RTS KLDA AND CMP ENE INC SEC RTS LDA	CMMAND CHEX SLOT16 DIPSW1,Y A NOCMD STATEFLG,3 #\$07 ESCCHECK CMDPROC CHARACTER #\$7F CMDPROC CHARACTER #\$7F CMDBYTE,X XOFPCK STATEFLG,3	**** CK ROUTINE ;DON'T WORRY ABOUT CMD SEQ FOR SI x ;ARE WE IN A COMMAND SEQUENCE? ;IF SO, GOTO COMMAND CENTRAL ;INDICATE COMMAND ;IGNORE HIGH BIT ;IS THIS BEGINNING OF A CMD SEQ? X ;START UP COMMAND MODES
CB1A: CB1A: CB1A: CB1A: CB1A: ACB20: CB1C: B9 CB1C: B9 CB20: B0 CB20: B0 CB20: B0 CB20: B0 CB20: B0 CB20: B0 CB20: B0 CB20: CB30: CB	81 36 88 07 05 FC 27 7F 38 05 88 38 08	04 CD 05 04	90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111	******** * START CMDSEQCP * ESCCHECI	OF CC LDY LDA LSR BCS LDA BCS LDA BEQ JSR SEC RTS K LDA AND CMP ENE INC SEC RTS	CMMAND CHEX SLOT16 DIPSW1,Y A NOCMD STATEFLG,I #\$07 ESSCHECK CMDPROC CHARACTER #\$7F CMDBYTE,X XOFPCK STATEFLG,I	**** CK ROUTINE ;DON'T WORRY ABOUT CMD SEQ FOR SI ;ARE WE IN A COMMAND SEQUENCE? ;IF SO, GOTO COMMAND CENTRAL ;INDICATE COMMAND ;IGNORE HIGH BIT ;IS THIS BEGINNING OF A CMD SEQ? X ;START UP COMMAND MODES ;INDICATE COMMAND ;IS XON ENABLED?
CB1A: CB1A: CB1A: CB1A: CB1A: CB1A: A4 CB1C: B9 CB1F: 4A CB20: B0 CB22: BD CB22: BD CB22: CB22: 20 CB22: 38 CB20: 60 CB22: 38 CB20: 60 CB22: 32 CB22: A5 CB30: 29 CB32: D0 CB37: FC CB33: 38 CB33: 60 CB37: 50 CB37: 50 CB3	81 36 88 07 05 FC 27 7F 38 05 88 38 08	04 CD 05 04	90 91 92 93 94 95 96 97 98 99 90 100 101 102 103 104 105 106 107 108 109 110 111 112 113	* * START CMDSEQCP * ESCCHECI	OF CC LDY LDA LSR BCS LDA AND BEQ JSR SEC RTS KLDA AND CMP ENE INC SEC RTS LDA	CMMAND CHEX SLOT16 DIPSW1,Y A NOCMD STATEFLG,3 #\$07 ESCCHECK CMDPROC CHARACTER #\$7F CMDPROC CHARACTER #\$7F CMDBYTE,X XOFPCK STATEFLG,3	**** CK ROUTINE ;DON'T WORRY ABOUT CMD SEQ FOR SI x ;ARE WE IN A COMMAND SEQUENCE? ;IF SO, GOTO COMMAND CENTRAL ;INDICATE COMMAND CENTRAL ;ISTHIS BEGINNING OF A CMD SEQ? X ;START UP COMMAND MODES ;INDICATE COMMAND
CB1A: CB1A: CB1A: CB1A: CB1A: ACB20: CB1C: B9 CB1C: B9 CB20: B0 CB20: B0 CB20: B0 CB20: B0 CB20: B0 CB20: B0 CB20: B0 CB20: CB30: CB	81 36 B8 07 05 FC 27 7F 38 05 B8 38 05 B8	04 CD 05 04	90 91 92 93 94 95 96 97 98 99 90 101 102 103 104 105 106 107 108 109 110 111 112 113 114	* * START CMDS EQC: * ESCCHECI * XOF FCK *	OF CC C LDY LDA LSR BCS LDA AND BEQ JSR SEC RTS CMP BNE SEC RTS LDA AND CMP ENE SEC RTS	CHARACTER #\$07 CHARACTER #\$07 CHARACTER #\$7F CMDBYTE, X XOFPCK STATEFLG, 1 #\$08 NOCMD	**** CK ROUTINE ;DON'T WORRY ABOUT CMD SEQ FOR SI ;ARE WE IN A COMMAND SEQUENCE? ;IF SO, GOTO COMMAND CENTRAL ;INDICATE COMMAND ;IGNORE HIGH BIT ;IS THIS BEGINNING OF A CMD SEQ? X ;START UP COMMAND MODES ;INDICATE COMMAND ;IS XON ENABLED?

10

11

6 3

-

3

3

-

CB46:90	10		118		BCC	NOCMD	; IF NOT, GO OUTPUT
CB48:C9	93		119		CMP	#\$93	; IS IT AN XOFF?
CB4A:FO	OE		120		BEO	XONWAIT	; IF SO, GO WAIT FOR ANOTHER INPUT
CB4C:48			121		PHA		
CB4D: BD	38	07	122		LDA	MISCFLG, X	CIC MODE?
CB50:4A		~ .	123		LSR	A	CARGE AND
CB51:4A			124		LSR	A	
CB52:68			125		PLA		
CB53:90	04		126		BCC	ANRTS	
CB55:90	1.7.7	06	127		STA		; IF SO, WE HAVE A BUFFER
CB55:9D CB58:18	DO	06		NOCMD	CLC	BUF BIIE, A	; INDICATE NOT A CMD SEO
CB58:18 CB59:60				ANRTS	RTS		FINDICATE NOT A CHD SEQ
CB59:60 CB5A:			130		RIS		
CB5A:20		C8		XONWAIT		GETCHAR	;GET ACIA/KBD DATA
CB5D:C9			132		CMP	#\$91	; IS IT AN XON?
CB5F:DO	F9		133		BNE	XONWAIT	; IF NOT, WAIT
CB61:18			134		CLC		;OTHERWISE, INDICATE NOT A CMD SEQ
CB62:60			135		RTS		; AND RETURN
CB63:			136	******	****	*******	****
CB63:			137	* NOW TH	HE OU	TPUT ROUTIN	NE YOU'VE BEEN WAITING FOR *
CB63:			138	******	****	********	******
CB63:20	1A	CB	139	OUTPUT	JSR	CMDSEQCK	
CB66: B0			140		BCS	ANRTS	;DON'T OUTPUT COMMAND SEQUENCES
CB68:			141	*			Energy Barrier Barrier
CB68:20	OF	CC		OUTPUT1	TOP	SCREENOUT	
CB68:20 CB6B:	96	cc	143		JOK	SCREENOUI	
CB6B:A4	26			OUTPUT2	LDY	SLOT16	
CB6D: B9			145	0011012	LDA	DIPSW1,Y	
CB70:4A	01	CU	146		LSR	A	
CB70:4A	4E		147		BCC	OUTPUT3	SKIP ETX/ACK FOR NATIVE MODES
CB73:4A	1000		148		LSR	A	, okii bik/nek tok militib nobbo
CB74:90			149		BCC	OUTPUT3	BRANCH IF NOT PSA EMULATION
CB74:90	40		150		Dec	OULFOLD	, bitation in and row interaction
				******			
CB76:							
CB76:				* P8A E			
CB76:						******	NOR HERITAL IN DOGLDB COOLDINGS
CB76:							BUT NOT WITHIN AN ESCAPE SEQUENCE
CB76:							S, THE HANDSHAKE IS PERFORMED
CB76:							OT ESC' AND THEN 4 MORE CHARS
CB76:					NTIL	AN 'ESC')	
CB76:			158				
CB76:A5				P8AOUT1		CHARACTER	;SAVE CHAR ON STACK
CB78:48			160		PHA		
CB79:BD			161		LDA		X ;CHAR COUNT FOR BUFFER FULL
CB7C:C9	-		162		CMP	#103	; IF <103 THEN 153 CHARS IN BUFFER
CB7E:90	10		163		BCC	ETX	
CB80:C9	6C		164		CMP	#108	; IF >=108 THEN LESS THAN 149 CHARS
CB82:B0	22		165		BCS	P8AOUT2	; SO NO HANDSHAKE IS NEEDED YET
CB84:C9	6B		166		CMP	#107	;SETS CARRY IF 107 (149 SENT)
CB86:68			167		PLA		AND AND DESCRIPTION AND A DESCRIPTION OF A
CB87:48			168		PHA		
CB88:49			169		EOR	#\$9B	;ESC?
CB8A: 29			170		AND	#\$7F	; IGNORE HI-BIT
			171		BNE	PSAOUT2	COUNT AS 1 OF 5 IF NOT 'ESC'
CB8C:DC					BCS	P8AOUT2 P8AOUT3	DON'T COUNT IF 149TH CHAR IS 'ESC'
CB8E:BC CB90:	19		172		DCS	FONDUI3	1000 1 00001 11 14510 0000 100
CB90: BE	D	0.04	0.000	ETX	LDA	STATEFIC	X ;SEND QUERY CHAR TO PRINTER
			175		AND	#\$1F	;(DEFAULT IS ETX)
CB93:29	9 12	50 C	1/5		AND	#\$11	(DEFNOLT TO GIV)

5

	CB95:09	80		176		ORA	#\$80	
	CB97:85			177		STA	CHARACTER	
	СВ99:20	02	CC	178		JSR	ACIAOUT	
	CB9C:20	AA	C8		ACK	JSR	GETCHAR	;GET ACIA/KBD DATA
	CB9F:49	86		180			#\$86	
	CBA1:DO	ED		181				; IF NOT ACK, REPEAT HANDSHAKE
	CBA3:9D	38	04	182		STA	HANDSHKE,	X ; INIT CHAR COUNT TO 255
	CBA6:			183	*			
	CBA6:DE							X
	CBA9:68				P8AOUT3			;GET REAL CHAR TO OUTPUT
	CBAA:85	27		186				
	CBAC:49	8D		187				; IF CR AND CR DELAY MODE
	CBAE: 0A			188				
-	CBAF:DO							; THEN FAKE CHAR COUNT TO LESS THAN
	CBB1:BD							X ; 48 TO FORCE HANDSHAKE ON NEXT
	CBB4:29							; CHARACTER OUT
-	CBB6:F0							
	CBB8:9D							X
	CBBB:				*			
_	CBBB:20							
-	CBBE:4C	EA	CB					;(SKIP DELAYS)
-	CBC1:						*******	
	CBC1:							UTPUT *
-	CBC1:		00					*****
-	CBC1:20		CC					;OUTPUT THE CHARACTER
	CBC4:			201				and Michael and An America, 422
-	CBC4:							, AND FF DELAYS
	CBC4:							
	CBC4:0A							
	CBC5:A8 CBC6:BD							
-			03					X ;GET DELAY FLAGS
	CBC9:C0 CBCB:F0			207 208			#\$18 OUTDLY1	;FORM FEED?
	CBCD: 4A							
_	CBCE: 4A							
	CBCF:CO			210		CDV	A #01 A	;RIGHT JUSTIFY LF DELAY ;LINE FEED?
	CBD1:F0			212		BEO	OUTDI.V1	JUNE FEED!
	CBD3:4A					LSR	A	
	CBD4:4A			214				RIGHT JUSTIFY CR DELAY
	CBD5:CO			215				CARRIAGE RETURN?
	CBD7:D0			216			OUTPUTEND	
-	CBD9:29							JUST WANT LOWEST 2 BITS
	CBDB:FO			218				;NO DELAY INDICATED
	CBDD: A8			219		TAY		THE PERIT TREATER
	CBDE: B9		CB	2008			DLYTBL-1,	Y
-	CBE1:A8							;DELAY IN 32 MSEC INCREMENTS
	CBE2: A9						#32	
-	CBE4:20				COLDBIDI		WAITMS	antisty count & blancer
	CBE7:88			224		DEY		
	CBE8:DO			225			OUTDLYLP	
	CBEA:	1.1.2		226		Sec. 11	COTONINE.	
-						ON LI	GENERATI	ON OPTION
	CBEA:					and all		
				228				
	CBEA:	27		228		LDA	CHARACTER	
	CBEA: CBEA:			229	LFGEN		CHARACTER A	
	CBEA: CBEA: CBEA:A5				LFGEN	ASL	A	
	CBEA: CBEA: CBEA:A5 CBEC:0A	1A		229 230 231	LFGEN	ASL CMP	A #S1A	;CARRIAGE RETURN?
	CBEA: CBEA: CBEA:A5 CBEC:OA CBED:C9	1A OD		229 230 231 232	LFGEN	ASL CMP BNE	A #\$1A OUTPUTEND	;CARRIAGE RETURN?

CBF4:6A			234		ROR	A	
CBF5:90	07		235		BCC	OUTPUTEND	
CBF7:A9	8A		236		LDA	#\$8A	
CBF9:85			237		STA	CHARACTER	;LINE FEED
CBFB:4C	6B	CB	238		JMP	OUTPUT2	; (DON'T ECHO IT)
CBFE:60			239	OUTPUTE	ND RT	S	
CBFF:			240	*			
CBFF:01			241	DLYTBL	DFB	\$01	;32 MSEC
CC00:08			242		DF B	\$08	;1/4 SEC
CC01:40			243			\$40	; 2 SEC
CC02:						********	
CC02:						T ROUTINE	* 5 F C C C C C C C C C C C C C C C C C C
CC02:			246	******	****	********	
CC02:20	F5	CA	247	ACIAOUT	JSR	SROUT	;READY FOR OUTPUT?
CC05:D0	FB	i no	248		BNE	ACIAOUT	
CC07:98			249		TYA		
CC08:09	89	0	250		ORA	#\$89	; PREPARE TO ADDRESS ACIA,
CCOA: A8			251		TAY		; CAUSING 6502 FALSE READ TO OCCUR
CCOB:A5	27	6	252		LDA		; ON PAGE \$BF (AVOIDING RDR READ)
CC0D:99	FF	BF	253		STA	\$BFFF,Y	;HERE YOU ARE ACIA
CC10:60	6		254		RTS		
CC11:			255				
CC11:						********	
CC11:			257	* RESTO	RE CU	JRSOR (NOT	FOR PASCAL) *
CC11:			258	* (A-RE	G SHO	OULD CONTAI	N NEW CHAR) *
CC11:			259	******	*****		*****
CC11:48	3		260	RESTORE	E PHA		;SAVE NEW CHARACTER
CC12: A4	24	1	261		LDY	CH	
CC14:A5	5 27	7	262		LDA		;OLD CHARACTER
CC16:91	28	3	263		STA	(BASL),Y	
CC18:68	3		264	10.40	PLA		
CC19:			265	*			
CC19:C9	9 9	5	266		CMP	#\$95	;SCREEN PICK?
CC1B:DO	0 00	3	267		BNE	RESTOREND	
CC1D: AS	5 2	7	268		LDA	CHARACTER	R ; IF SO, USE SCREEN CHAR
CC1F:CS	9 2	0	269	,	CMP	#\$20	; INVERSE?
CC21:B	0 0	6	270	)	BCS	RESTORENI	
CC23:20	D D	F CC	271		JSR		REVERSE THE TRANSLATION
CC26:5	9 D	B CC	272		EOR		
CC29:8	5 2	7	273	B RESTOR	END S	TA CHARACTI	ER
CC2B:6	0		274	1	RTS		
CC2C:			275	5 *			
CC2C:			276	5	CHN	SSC.UTIL	

			-				****
CC2C:			1.2		*****	*********	*********
CC2C:			2.77	*			
CC2C:					II S	SC FIRMWAR	5 *
CC2C:			5	*			A DECC. • IN COLUMN TO TO BE DE
CC2C:			6	* BY	LARRY	KENYON	A DECEMBER OF A
CC2C:			7	*			*
CC2C:			8	* -J	ANUAR	Y 1981-	*******
CC2C:			9	*			
CC2C:			10	* (C) C	OPYRI	GHT 1981 B	Y APPLE COMPUTER, INC. *
CC2C:			11	*			and the second s
CC2C:			12	******	*****	********	******
CC2C:			13	*			*
CC2C:			14	* UTILI	TY RO	UTINES	*
CC2C:			15	*			*
CC2C:			16	******	****	********	****
CC2C:			1000			IC KEYBOAR	
CC2C:						********	
				CKKBD	CLC		
CC2C:18 CC2D:BD	20	07	20	CARDD		MISCFLG, X	;RETURN CARRY CLEAR FOR NO DATA
		07					
CC30:29			21		AND		;ANSWER NO IF KEYBOARD IS DISABLED
CC32:F0	09		22		BEQ	CKKBDXIT	
CC34:			23				
CC34:AD		CO		CKKBD1	LDA	KBD	
CC37:10			25		BPL	CKKBDXIT	
CC39:8D	10	CO	26		STA	KBDSTRB	
CC3C:38			27		SEC		; INDICATE DATA
CC3D:60			28	CKKBDXI	T RTS		
CC3E:			29	******	*****	*******	*****
CC3E:			30	* GET A	CHAR	FROM KEYB	DARD 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		34		TNC	RNDH	,
CC44:20	2C	CC	35	GETKBD1	JSR		; KEYBOARD FETCH ROUTINE
CC47: B8			36		CLV	onnoo	;INDICATE NO ESCAPE SEQUENCE
CC48:90	F3		37			CRABDATE	;EXIT IF NO KEY PRESS
CC4A:20		cc	38		JSR	RESTORE	DO BASIC CURSED DUTY
CC4D: 29		cc	39		AND		TO BASIC CORSED DUTY
CC4F:DD		05	40		CMP	#\$7F	TO TH HUR CHARM OF A COMMANDS
CC52:D0		05	40				; IS IT THE START OF A COMMAND?
			100		BNE		; IF NOT, EXIT INDICATING DATA
CC54:A4 CC56:B9		00	42		LDY	SLOT16	ANT IL DO AND DOO 700
	01	0	43		LDA	DIPSW1,Y	;ONLY DO CMD ESC FOR PPC, SIC MODE
CC59:4A			44		LSR	A	
CC5A:BO	35		45		BCS	GETKBDONE	
CC5C:			46	******	****	*******	****
CC5C:			47	* KEYBO	ARD E	SCAPE HANDI	LER *
CC5C:			48	******	****	*******	****
CC5C:A0	OA		49	KBDESC	LDY	#\$A	FIRST PRINT A PROMPT
C5E: B9	93	CC	50	PROMPTL	DOP L	DA PROMPTBI	L.Y
CC61:85	27		51		STA	CHARACTER	.,.
CC63:98			52		TYA		
CC64:48			53		PHA		
CC65:20	AB	CC	54			SCREENOUT	1 ;ALWAYS SEND TO SCREEN
CC68:68			55		PLA	JCRUENOUT	, ADMAID DEND TO SCREEN
CC69:A8			56		TAY		
CC6A:88			57		DEY		
	F1		57		BPL	DD ON DET OO	And a state of the second s
			20		BPL	PROMPTLOOD	2
CC6B:10 CC6D:			59				

3

- -

3

. 3

8 3

3

3

-

-

0000.10	0.4		00		TDA		CONTRACT AN CONMAND COMMON
CC6D: A9			60		LDA	and the standard standard of	START OUT IN COMMAND STATE 1
CC6F:20	7B	CE	61		JSR	SETOSTATE	
CC72:			62				UNTE FOR UNUDADE OUNEACTIN
CC72:20		CC	1.00		JSR		;WAIT FOR KEYBOARD CHARACTER
CC75:10			64		BPL	GETCMD	
CC77:C9			65		CMP	#\$88	; BACKSPACE?
CC79:F0	E1		66		BEQ	KBDESC	; IF SO, THEN START OVER
CC7B:85	27		67		STA	CHARACTER	
CC7D:			68	*			
CC7D:20	A3	CC	69		JSR	SCREENOUT1	
CC80:20	1A	CB	70		JSR	CMDSEQCK	; PUMP THRU CMD INTERPRETER
CC83:			71	*			
CC83:BD	B8	04	72		LDA	STATEFLG, X	; ARE WE DONE?
CC86:29	07		73		AND	#\$07	
CC88:D0			74		BNE	GETCMD	; IF NOT, GO AGAIN
CC8A:			75	*			
CC8A: A9	8D		76		LDA	#\$8D	FORCE BACK A CARRIAGE RETURN
CC8C:85			77		STA	CHARACTER	
CC8E: 2C	58	FF	78		BIT	IORTS	; INDICATE THAT A CMD SEQ HAS OCCURRED
CC91:38			79	GETKBDON	IE SEC		; INDICATE SUCCESS
CC92:60			80		RTS		
CC93:			81	*			
CC93:			82	*			
CC93: BA	C3	D3		PROMPTBI	ASC	":CSS	ELPPA"
CC96:D3							
CC99:CC							
CC9C:C1	20	20					
CC9D: 8D			84		DFB	\$8D	
CC9E:			85	*		Ç OD	
			86	******	****	********	********
CC9E: CC9E:							
CC9E:			87	* ROUTIN	NE TO	PRINT A C	HARACTER ON THE CURRENT DISPLAY *
CC9E: CC9E:	38	07	87 88	* ROUTI	NE TO	PRINT A C	HARACTER ON THE CURRENT DISPLAY *
CC9E: CC9E: CC9E:BD			87 88	* ROUTIN	NE TO	PRINT A C	HARACTER ON THE CURRENT DISPLAY *
CC9E: CC9E: CC9E:BD CCA1:10			87 88 89	* ROUTIN	NE TO	PRINT A C	HARACTER ON THE CURRENT DISPLAY *
CC9E: CC9E: CC9E:BD CCA1:10 CCA3:	13		87 88 89 90 91	* ROUTIN ******* SCREENO	NE TO ***** UT LD BPL	PRINT A CI **************** A MISCFLG, NOOUT	HARACTER ON THE CURRENT DISPLAY * ***********************************
CC9E: CC9E: CC9E: BD CCA1:10 CCA3: CCA3: BD	13 38	07	87 88 89 90 91	* ROUTIN ******* SCREENO	NE TO ***** UT LD BPL UT1 L	PRINT A CI ************************************	HARACTER ON THE CURRENT DISPLAY *
CC9E: CC9E: BD CCA1:10 CCA3: CCA3: BD CCA6:29	13 38 02	07	87 88 89 90 91 92 93	* ROUTIN ******* SCREENO	NE TO ***** UT LD BPL UT1 L AND	PRINT A CI ************************************	HARACTER ON THE CURRENT DISPLAY * ***********************************
CC9E: CC9E: BD CCA1:10 CCA3: CCA3: BD CCA6:29 CCA8:F0	13 38 02 0D	07	87 88 89 90 91 92 93 94	* ROUTIN ******* SCREENO	NE TO ***** UT LD BPL UT1 L AND BEQ	PRINT A CU *********** A MISCFLG, NOOUT DA MISCFLG #\$02 ASCREEN	HARACTER ON THE CURRENT DISPLAY * ***********************************
CC9E: CC9E:BD CCA1:10 CCA3: CCA3:BD CCA6:29 CCA8:F0 CCAA:BD	13 38 02 0D 88	07	87 88 89 90 91 92 93 94 95	* ROUTIN ******* SCREENO	NE TO ***** UT LD BPL UT1 L AND BEQ LDA	PRINT A CU *********** A MISCFLG, NOOUT DA MISCFLG #\$02 ASCREEN STATEFLG,	HARACTER ON THE CURRENT DISPLAY * ***********************************
CC9E: CC9E:BD CCA1:10 CCA3:BD CCA3:BD CCA6:29 CCA8:F0 CCAA:BD CCAA:29	13 38 02 0D 88 38	07 04	87 88 89 90 91 92 93 94 95 96	* ROUTIN ******* SCREENO	NE TO ***** BPL UT1 L AND BEQ LDA AND	PRINT A CU *********** A MISCFLG, NOOUT DA MISCFLG #\$02 ASCREEN STATEFLG, #\$38	HARACTER ON THE CURRENT DISPLAY * ***********************************
CC9E: CC9E: CC9E:BD CCA1:10 CCA3:BD CCA6:29 CCA8:F0 CCAA:BD CCAA:ED CCAA:F0	13 38 02 0D 88 38	07 04	87 88 89 90 91 92 93 94 95 96 97	* ROUTIN ******* SCREENO	NE TO ***** UT LD BPL UT1 L AND BEQ LDA	PRINT A CU *********** A MISCFLG, NOOUT DA MISCFLG #\$02 ASCREEN STATEFLG,	HARACTER ON THE CURRENT DISPLAY * ***********************************
CC9E: CC9E: CC9E:BD CCA1:10 CCA3: CCA3:BD CCA6:29 CCA8:F0 CCAA:BD CCAD:29 CCAF:F0 CCB1:	13 38 02 0D 88 38 06	07 04	87 88 89 90 91 92 93 94 95 96 97 98	* ROUTIN ******* SCREENO	NE TO ***** BPL UT1 L AND BEQ LDA AND BEQ	PRINT A CU *********** A MISCFLG, NOOUT DA MISCFLG #\$02 ASCREEN STATEFLG, #\$38	HARACTER ON THE CURRENT DISPLAY * ***********************************
CC9E: CC9E: CC9E:BD CCA1:10 CCA3: CCA3:BD CCA6:29 CCA8:F0 CCAA:BD CCAA:BD CCAA:CAA:CAA:CAAAAAAAAAAAAAAAAAAAAAAAA	13 38 02 0D 88 38 06	07 04	87 88 89 90 91 92 93 94 95 94 95 96 97 98 99	* ROUTII ******* SCREENOU * SCREENOU	NE TO ***** BPL UT1 L AND BEQ LDA AND BEQ TXA	PRINT A CU *********** A MISCFLG, NOOUT DA MISCFLG #\$02 ASCREEN STATEFLG, #\$38	HARACTER ON THE CURRENT DISPLAY * ***********************************
CC9E: CC9E: CC9E:BD CCA1:10 CCA3: CCA3:BD CCA6:29 CCA8:F0 CCAA:BD CCAA:BD CCAA:29 CCAF:F0 CCB1:28 CCB1:8A CCB2:48	13 38 02 0D 88 38 06	07	87 88 89 90 91 92 93 94 95 96 97 98 99 100	* ROUTII ******* SCREENOU * SCREENOU	NE TO ***** BPL UT1 L AND BEQ LDA AND BEQ TXA PHA	PRINT A CL A MISCFLG, NOOUT DA MISCFLG #\$02 ASCREEN STATEFLG, #\$38 ASCREEN	HARACTER ON THE CURRENT DISPLAY * ***********************************
CC9E: CC9E: CC9E:BD CCA1:10 CCA3:BD CCA6:29 CCA8:F0 CCAA:BD CCAD:29 CCAF:F0 CCAF:F0 CCB1:29 CCAF:F0 CCB1:8A CCB2:48 CCB3:A9	13 38 02 0D 88 38 06 AF	07	87 88 89 90 91 92 93 94 95 96 97 98 99 100 101	* ROUTII ******* SCREENOU * SCREENOU	NE TO ***** BPL UT1 L AND BEQ LDA AND BEQ TXA PHA LDA	PRINT A CL A MISCFLG, NOOUT DA MISCFLG #\$02 ASCREEN STATEFLG, #\$38 ASCREEN	HARACTER ON THE CURRENT DISPLAY * ***********************************
CC9E: CC9E: CC9E:BD CCA1:10 CCA3:BD CCA6:29 CCA8:F0 CCA8:F0 CCA1:29 CCAF:F0 CCB1:8A CCB2:48 CCB2:48 CCB3:A9 CCB5:48	13 38 02 0D 88 38 06 AF	07	87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102	* ROUTIN ******* SCREENO * SCREENO	NE TO ***** UT LDD BPL UT1 L AND BEQ LDA AND BEQ TXA PHA LDA PHA	PRINT A CL A MISCFLG, NOOUT DA MISCFLG #\$02 ASCREEN STATEFLG, #\$38 ASCREEN	HARACTER ON THE CURRENT DISPLAY * ***********************************
CC9E: CC9E: CC9E:BD CCA1:10 CCA3:BD CCA6:29 CCA8:F0 CCA8:F0 CCA3:BD CCA6:29 CCA7:F0 CCB1:29 CCA7:F0 CCB1:A2 CCB2:48 CCB3:48 CCB6:60	13 38 02 0D 88 38 06 AF	07	87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103	* ROUTIN SCREENOU * SCREENOU * NOOUT	NE TO ***** BPL UT1 L AND BEQ LDA AND BEQ TXA PHA LDA	PRINT A CL A MISCFLG, NOOUT DA MISCFLG #\$02 ASCREEN STATEFLG, #\$38 ASCREEN	HARACTER ON THE CURRENT DISPLAY * ***********************************
CC9E: CC9E: CC9E:BD CCA1:10 CCA3: CCA3:BD CCA6:29 CCA8:F0 CCAA:BD CCAA:BD CCAA:CAB CCAA:CAB CCB1:AB CCB1:AB CCB3:A8 CCB3:A8 CCB5:48 CCB5:48 CCB5:48	13 38 02 0D 88 38 06 AF	07	87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104	* ROUTII ******* SCREENO * SCREENO * * NOOUT	NE TO ***** BPL UT1 L AND BEQ LDA AND BEQ TXA PHA LDA PHA RTS	PRINT A CL MISCPLG, NOOUT DA MISCPLG #\$02 ASCREEN STATEFLG, #\$38 ASCREEN #>SENDCD-	HARACTER ON THE CURRENT DISPLAY * ***********************************
CC9E: CC9E: CC9E:BD CCA1:10 CCA3: CCA3:BD CCA6:29 CCA8:F0 CCAA:BD CCAA:BD CCAA:GE CCA3:CCCA3: CCB1:SA CCB2:48 CCB5:48	13 38 02 0D 88 38 06 AF	07	87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104	* ROUTII ******* SCREENO * SCREENO * * NOOUT * *	NE TO ***** BPL UT1 L AND BEQ LDA AND BEQ TXA PHA LDA PHA RTS	PRINT A CL A MISCFLG, NOOUT DA MISCFLG #\$02 ASCREEN STATEFLG, #\$38 ASCREEN	HARACTER ON THE CURRENT DISPLAY * ***********************************
CC9E: CC9E:BD CCA1:10 CCA3:BD CCA6:29 CCA8:F0 CCA8:F0 CCA1:29 CCAF:F0 CCB1:SA8 CCB2:48 CCB2:48 CCB5:48 CCB5:48 CCB5:48 CCB5:48 CCB5:48 CCB5:48 CCB5:48 CCB7:CCB7:CCB7:CCB7:CCB7	13 38 02 0D 88 38 06 AF	07	87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105	* ROUTIN ******* SCREENO * SCREENO * * NOOUT * * APPLE *	NE TO ***** BPL UT1 L AND BEQ LDA AND BEQ TXA PHA LDA PHA RTS : 40-C	PRINT A Cl ********** A MISCFLG, NOOUT DA MISCFLG #\$02 ASCREEN STATEFLG, #538 ASCREEN #>SENDCD-	HARACTER ON THE CURRENT DISPLAY * ***********************************
CC9E: CC9E:BD CCA1:10 CCA3:BD CCA6:29 CCA8:F0 CCA8:F0 CCA2:29 CCA7:F0 CCB1:CCB1:8A CCB2:48 CCB2:48 CCB3:A9 CCB5:48 CCB5:48 CCB7:CCB7:CCB7:CCB7:CCB7:CCB7:CCB7:CCB7	13 38 02 0D 88 38 06 AF	07 04	87 88 88 90 91 92 93 94 95 96 97 98 99 90 101 102 103 104 105 106 107	* ROUTII ****** SCREENO * SCREENO * * NOOUT * * APPLE * ASCREEN	NE TO ***** BPL UT1 LD BPL UT1 L AND BEQ LDA AND BEQ TXA PHA LDA PHA RTS 40-C	PRINT A Cl ********* A MISCFLG, NOOUT DA MISCFLG #\$02 ASCREEN STATEFLG, #338 ASCREEN #>SENDCD- COL SCREEN GETXLATE	HARACTER ON THE CURRENT DISPLAY * ***********************************
CC9E: CC9E: CC9E:BD CCA1:10 CCA3: CCA3:BD CCA6:29 CCA8:F0 CCA4:BD CCA29 CCA7:F0 CCB1:A8 CCB2:48 CCB3:A9 CCB3:48 CCB3:4	13 38 02 0D 88 38 06 AF	07 04	87 88 89 90 91 92 93 94 95 96 97 98 99 90 100 101 102 103 104 105 106 107 108	* ROUTII ******* SCREENO * SCREENO * * NOOUT * * APPLE * ASCREEN	NE TO ****** UT LD. BPL AND BEQ LDA AND BEQ LDA AND BEQ TXA PHA RTS : 40-CC	PRINT A CU ********** A MISCFLG, NOOUT DA MISCFLG #\$02 ASCREEN STATEFLG, #\$38 ASCREEN #>SENDCD- POL SCREEN GETXLATE #\$80	HARACTER ON THE CURRENT DISPLAY * ***********************************
CC9E: CC9E: CC9E: BD CCA1:10 CCA3: CCA3: BD CCA6:29 CCA8:F0 CCAA: BD CCAA: BD CCAA: BD CCAA: CCAA: CCAA: CCAA: CCAA: CCB1: CCB1: CCB1: CCB1: CCB1: CCB1: CCB2: 48 CCB2: 48 CCB	13 38 02 0D 88 38 06 AF 0 0 D F 80 0 0 D F 80 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	07 04 cc	87 88 89 90 91 92 93 94 95 97 97 98 99 100 101 102 103 104 105 106 107 108	* ROUTII ******* SCREENO * SCREENO * * NOOUT * * APPLE * ASCREEN	NE TO ****** UT LD. BPL BPL LDA AND BEQ LDA AND BEQ TXA PHA LDA PHA RTS : 40-CC JSR ORA CMP	PRINT A CU ********* A MISCPLG, NOOUT DA MISCPLG #\$02 ASCREEN STATEFLG, #\$38 ASCREEN #>SENDCD- COL SCREEN GETXLATE #\$80 #\$E0	HARACTER ON THE CURRENT DISPLAY * ***********************************
CC9E: CC9E: CC9E:BD CCA1:10 CCA3:BD CCA6:29 CCA8:F0 CCA8:F0 CCA1:29 CCAF:F0 CCB1:8A CCB2:48 CCB3:A9 CCB5:48 CCB5:48 CCB5:48 CCB5:48 CCB5:48 CCB7:CCB7: CCB7:20 CCB7:20 CCB2:48 CCB3:49 CCB3:49 CCB3:49 CCB3:49 CCB3:49 CCB3:49 CCB3:49 CCB3:49 CCB3:40	13 38 02 0D 88 38 06 AF 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	07 04	87 88 89 90 91 92 93 94 95 96 97 97 98 99 100 101 102 103 104 105 106 107 108 109 110	* ROUTII ******* SCREENO * SCREENO * * NOOUT * * APPLE * ASCREEN	NE TO ****** BPL DT LD BPL LDA AND BEQ LDA AND BEQ TXA PHA RTS : 40-CC I JSR ORA CMP BCC	PRINT A Cl ********** A MISCFLG, *000UT DA MISCFLG #\$02 ASCREEN STATEFLG, #\$38 ASCREEN #>SENDCD- COL SCREEN GETXLATE #\$80 #\$20 TESTLETTE	HARACTER ON THE CURRENT DISPLAY * ***********************************
CC9E: CC9E: CC9E:BD CCA1:10 CCA3:BD CCA6:29 CCA8:F0 CCA8:F0 CCA1:29 CCAF:F0 CCB1: CCB1:8A CCB2:48 CCB2:48 CCB2:48 CCB5:48 CCB5:48 CCB7: CCB7: CCB7: CCB7: CCB7:20 CCB7	13 38 02 0D 88 38 06 AF 0 0 0 0 0 0 0 0 0 0 0 0 0	07 04 , , , , , , , , , , , , , , , , , ,	87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111	* ROUTII ****** SCREENO * SCREENO * * NOOUT * APPLE * ASCREEN	NE TO ****** UT LD. BPL BPL LDA AND BEQ LDA AND BEQ TXA PHA RTS : 40-CC I JSR ORA CMP BCC EOR	PRINT A Cl ********* A MISCFLG, NOOUT DA MISCFLG #\$02 ASCREEN STATEFLG, #338 ASCREEN #>SENDCD- COL SCREEN GETXLATE #\$80 #\$E0 TESTLETTE LCMASK,Y	HARACTER ON THE CURRENT DISPLAY * ***********************************
CC9E: CC9E: CC9E:BD CCA1:10 CCA3:BD CCA6:29 CCA8:F0 CCA8:F0 CCA3:BD CCA6:29 CCA8:F0 CCA1:29 CCA7:F0 CCB1: CCB1:8A CCB2:48 CCB3:A9 CCB5:48 CCB6:60 CCB7: CCB7: CCB7: CCB7:20 CCB7:20 CCB4:05 CCB6:95 CCB6:95 CCB6:95 CCCB2:40 CCB7:20 CCB2:40 CCB2:40 CCB2:40 CCB7:20 CCB2:40 CCB7:20 CCB7:20 CCB2:40 CCB7:20 CCB7 CCB7:20 CCB7 CCB7 CCB7 CCB7 CCB7 CCB7 CCB7 CCB	13 38 02 0D 88 38 06 AF 0 0 0 0 0 0 0 0 0 0 0 0 0	07 04 , , , , , , , , , , , , , , , , , ,	87 88 89 90 91 92 93 94 95 96 97 98 99 90 100 101 102 103 104 105 106 107 108 109 110 111 112	* ROUTII ******* SCREENO * SCREENO * * NOOUT * * ASCREEN	NE TO ****** UT LD. BPL BPL LDA AND BEQ LDA AND BEQ TXA PHA RTS : 40-CC I JSR ORA CMP BCC EOR	PRINT A Cl ********* A MISCFLG, NOOUT DA MISCFLG #\$02 ASCREEN STATEFLG, #338 ASCREEN #>SENDCD- COL SCREEN GETXLATE #\$80 #\$E0 TESTLETTE LCMASK,Y	HARACTER ON THE CURRENT DISPLAY * ***********************************
CC9E: CC9E: CC9E:BD CCA1:10 CCA3:BD CCA6:29 CCA8:F0 CCA8:F0 CCA1:29 CCAF:F0 CCB1: CCB1:8A CCB2:48 CCB2:48 CCB2:48 CCB5:48 CCB5:48 CCB7: CCB7: CCB7: CCB7: CCB7:20 CCB7	13 38 02 0D 88 38 06 AF 0 0 0 0 0 0 0 0 0 0 0 0 0	07 04 , , , , , , , , , , , , , , , , , ,	87 88 89 90 91 92 93 94 95 96 97 98 99 97 98 99 97 100 101 102 103 104 107 108 109 110 111 112	* ROUTII ******* SCREENO * SCREENO * * NOOUT * * ASCREEN * *	NE TO NE	PRINT A CL ********** A MISCFLG, ********** PA MISCFLG, *************** PA MISCFLG, ************************************	HARACTER ON THE CURRENT DISPLAY * ***********************************

CCC6:			115	*				
CCC6: C9	C1		116	TESTLETT	TER CI	MP #\$C1	; <a?< td=""><td></td></a?<>	
CCC8:90	F9		117		BCC	TOSCREEN		
CCCA:C9	DB		118		CMP	#\$DB	;>Z?	
CCCC: BO	F5		119		BCS	TOSCREEN		
CCCE:59	D7	CC	120		EOR	UCMASK, Y		
CCD1:90	FO		121		BCC	TOSCREEN	; <always></always>	
CCD3:			122	*				
CCD3:			123	* MASKS	FOR	CASE TRANS	LATION	
CCD3:20	00	EO	124	LCMASK	DFB	\$20,\$00,\$	E0,\$20	
CCD6:20								
CCD7:00 CCDA:CO	00	00	125	UCMASK	DFB	\$00,\$00,\$	00,\$C0	
CCDB:00 CCDE:C0	00	EO	126	REVMASK	DFB	\$00,\$00,\$	E0,\$C0	
CCDF:			127	*				1100 21
CCDF: BD CCE2: 2A	B8	03	128 129	GETXLATE	E LDA ROL	DELAYFLG,	X ;TRANSLATE OPTI	ONS IN B6-B7
						A		
CCE3:2A			130		ROL	1.1		
CCE4:2A			131		ROL	A		
CCE5:29	03		132		AND	#\$03		
CCE7:A8	-		133		TAY			
CCE8:A5	27		134		LDA	CHARACTER		
CCEA:60			135		RTS			
CCEB:			136	*				

(listings continued on next page)

-

-

3

3

2 3

1

-

CCEB:	138	CHN SSC	.CMD		
CCEB:			********	****	
CCEB:	2 *			*	
CCEB:		II SSC F	TRMWARE		
CCEB:	4 *	11 350 1.	INIMAND	*	
CCEB:		LARRY KEN	YON	*	
CCEB:	6 *		C.S.S.	*	
CCEB:		ANUARY 19	81-	****	****
CCEB:	8 *				*
CCEB:		OPYRIGHT	1981 BY APPL	E COMPUTE	ER. INC. *
CCEB:	10 *				*
CCEB:		*******	********	*******	*****
CCEB:	12 *			*	
CCEB:	13 * SSC 0	OMMAND PR	OCESSOR	*	
CCEB:	14 *			*	
CCEB:		********	********	*****	
CCEB:					****
CCEB:					CESSER ROUTINE *
CCEB:					*****
CCEB:42	19 CMDTBL	DFB \$42			
CCEC:67	20	DFB \$67		PAS	NS=7
CCED: CO	21	DFB >BR	EAKCMD-1		
CCEE:54	22	DFB \$54	;T(EF	RMINAL)	
CCEF:47	23	DFB \$47	;CIC		NS=7
CCF0:A6	24		RMCMD-1		
CCF1:43	25	DFB \$43	;C(R	GENERATE	)
CCF2:87	26	DFB \$87	;	PPC	NS=7
CCF3: A6	27	DFB >TE	RMCMD-1		
CCF4:51	28	DFB \$51	;Q(U)	IT)	
CCF5:47	29	DFB \$47	;CIC		NS=7
CCF6:B8	30	DFB >QU	IITCMD-1		
CCF7:52	31	DFB \$52	;R(E	SET)	
CCF8:C7	32	DFB \$C7	;CIC	PPC	NS=7
CCF9:AC	33	DFB >RE	SETCMD-1		
CCFA: 5A	34	DFB \$5A		OMMAND	
CCFB:E7	35	DFB \$E7	Carrier Carrier	PPC PAS	NS=7
CCFC:F3	36		CMD-1		
CCFD: 49	37	DFB \$49		OMMAND	
CCFE:90	38	DFB \$90	Realized and	PPC	NS=0
CCFF:D3	39		MD-1		
CD00:4B	40	DFB \$4E		OMMAND	NG-0
CD01:90	41	DFB \$90		PPC	NS=0
CD02:DF	42	DFB >KC	CMD-1		
CD03:	43 *				
CD03:45	44	DFB \$45			
CD04:43	45	DFB \$43			NS=3
CD05:80	46	DFB \$80			
CD06:46	47	DFB \$46	0	OMKYBD)	
CD07: E3	48	DFB \$E	52	PPC PAS	NS=3
CD08:04	49	DFB \$04		CENEDATE	
CD09:4C	50	DFB \$40 DFB \$E		GENERATE PPC PAS	NS=3
CDOA:E3 CDOB:01	51 52	DFB \$0		FFC FAS	10-0
CDOB:01 CDOC:58	52	DFB \$5		FF)	
CDOC:58 CDOD:E3	53	DFB \$50		PPC PAS	NS=3
CDOE:08	54	DFB \$0			an Us surviv
CD0F:54	56	DFB \$5		BBING)	
CD10:83	57	DFB \$8		PPC	NS=3
CD10:85	57	50 50			

CD11:40		58		DFB	\$40								
CD12:53		59		DFB	\$53	;S(H)	POT	NCI					
CD12:53 CD13:43		60		DFB	\$43	;CIC	LE LL	NG)	NS=3				
		61		DFB	\$40	icic			N9=2				
CD14:40		62		DFB	\$40 \$4D	;M(UN	ICH	TEN					
CD15:4D									10 2				
CD16:E3		63		DFB	\$E3	;010	PPC	PAS	NS=3				
CD17:20		64		DF B	\$20								
CD18:		65	*										
CD18:00		66		DFB	\$00	; END	OF.	FIRST	PART	MARKER			
CD19:		67	*										
CD19:42		68	CMDTBL1	DFB	\$42	; B(A							
CD1A:F6		69		DFB	\$F6		PPC	PAS	NS=6				
CD1B:7C		70		DFB	>BAUDCMD-1	1							
CD1C:50		71		DFB	\$50	;P(AI							
CD1D:F6		72		DFB	\$F6	;CIC	PPC	PAS	NS=6				
CD1E:9A		73		DFB	>PARITYCME	0-1							
CD1F:44		74		DFB	\$44	;D(A	(A)						
CD20:F6		75		DFB	\$F6		PPC	PAS	NS=6				
CD21:9B		76		DFB	>DATACMD-1	1							
CD22:46		77		DFB	\$46	;F(F	DEL	AY)					
CD23:F6		78		DFB	\$F6	;CIC	PPC	PAS	NS=6				
CD24:46		79		DFB	>FFCMD-1								
CD25:4C		80		DFB	\$4C	;L(F	DEL	AY)					
CD26:F6		81		DFB	\$F6			PAS	NS=6				
CD27:40		82		DFB	>LFCMD-1								
CD28:43		83		DFB	\$43	;C(R	DEL	AY)					
CD29:F6		84		DFB	\$F6	1.000		PAS	NS=6				
CD2A: 3A		85		DFB	>CRCMD-1	1010			10-0				
CD2B:54		86		DFB	\$54	;T(R)	ANGT	ATE)					
CD2C:D6		87		DFB	SD6			2	NS=6				
CD2D: 34		88		DFB	>TRANCMD-1		110	•	140-0				
CD2E:4E		89		DFB	\$4E	IN CO	OMMO	ND					
CD2F:90		90		DFB	\$90	;			NS=0				
CD30: E8		91		DFB	>NCMD=1	'	PPC		N9=0				
CD31:53							DER	in on					
CD31:55		92 93		DFB	\$53		KEEN	ISLOT)	10 0				
		22		DFB	\$56	;CIC			NS=6				
CD33:60		94		DFB	>SSLOTCMD-	-1							
CD34:		95	*			_							
CD34:00		96		DFB	\$00	;END	OF	TABLE	MARK	ER			
CD35:		97											
CD35:		0.00	******										
CD35:			* COMMAN										
CD35:					PARSER) *								
CD35:			* (MUST										
CD35:		102			) *								
CD35:		1000			********								
CD35: A9			TRANCMD		#\$3F	;SET	SCR	REEN TH	RANSL	ATE OPT	IONS		
CD37:A0		105		LDY	#\$7								
CD39:D0		106		BNE	DELAYSET	; <aln< td=""><td>AYS</td><td>5&gt;</td><td></td><td></td><td></td><td></td><td></td></aln<>	AYS	5>					
CD3B:A9			CRCMD	LDA	#\$CF	;SET	CR	DELAY					
CD3D: AO		108		LDY	#\$5								
CD3F:DO	AO	109		BNE	DELAYSET	; <aln< td=""><td>AYS</td><td>5&gt;</td><td></td><td></td><td></td><td></td><td></td></aln<>	AYS	5>					
CD41:		110	*										
CD41:A9	F3	111	LFCMD	LDA	#\$F3	;SET	LF	DELAY					
CD43:A0	03	112		LDY	#\$3								
CD45:D0		113		BNE	DELAYSET	; <al< td=""><td>AYS</td><td>5&gt;</td><td></td><td></td><td></td><td></td><td></td></al<>	AYS	5>					
CD47:		114	*					TOTO OF					
CD47:A9			FFCMD	LDA	#SFC	SET	FF	DELAY					
and the second s	and the second se			- and the second		1001		- Action of the					

0

6

- 3

. 3

. 3

CD49:A0 01		
	116 LDY	#\$1
CD4B:3D B8 03	117 DELAYSET AND	DELAYFLG, X ; DON'T DISTURB THE OTHER FLAGS
CD4E:85 2A	118 STA	ZPTMP1
CD50:BD 38 04	119 LDA	PARAMETER, X
CD53:29 03	120 AND	#\$03 ;JUST USE TWO BITS
CD55:18	121 CLC	
CD56:6A	122 ROR	A ;ONCE FOR FUN
CD57:2A	123 ROTATE ROL	A ;CHANGE DIRECTIONS
CD58:88	124 DEY	A Jenning Diribertono
CD59:D0 FC	125 BNE	ROTATE ; PREPARE IT TO OR INTO THE FLAGS
		NOTIFIE / FRANKER IN TO ON ANYO AND THREE
CD5B:	126 *	755101
CD5B:05 2A	127 ORA 128 STA	ZPTMP1 DELAYFLG, X
CD5D:9D B8 03		DELAIF LG, X
CD60:60	129 RTS	
CD61:	130 *	
CD61:29 07	131 SSLOTCMD AND	
CD63:0A	132 ASL	A
CD64:0A	133 ASL	A
CD65:0A	134 ASL	A
CD66:85 2A	135 STA	ZPTMP1
CD68:0A	136 ASL	A
CD69:C5 26	137 CMP	SLOT16 ; MAKE SURE WE DON'T SET IT
CD6B:F0 OF	138 BEQ	SSLOTCMD1 ; TO OUR OWN SLOT
CD6D: BD B8 04	139 LDA	STATEFLG, X
CD70:29 C7	140 AND	#\$C7 ; PUT NEW SLOT NUMBER IN BITS 3-5
CD72:05 2A	141 ORA	ZPTMP1 ; OF CMDBYTE, X
CD74:9D B8 04	142 STA	STATEFLG, X
CD77:A9 00	143 LDA	#0 ;STORE ZERO INTO
CD79:9D 38 06	145 IDA	CHNBYTE, X ;SLOT OFFSET (SET TO CNOO ENTRY)
CD79:90 38 08	145 SSLOTCMD1 RT	
and the second sec		
CD7D:	146 *	NAME OF A DAUG DAUG
CD7D: 29 OF	147 BAUDCMD AND	#\$OF ;SET NEW BAUD RATE BAUDCMD2
CD7F:D0 07	148 BNE	
CD81:B9 81 C0	149 BAUDCMD1 LDA	
CD84:4A	150 LSR	A
	150 LSR 151 LSR	A
CD84:4A	150         LSR           151         LSR           152         LSR	A A A
CD84:4A CD85:4A	150         LSR           151         LSR           152         LSR           153         LSR	A A A A
CD84:4A CD85:4A CD86:4A	150         LSR           151         LSR           152         LSR           153         LSR           154         BAUDCMD2	A A A A #\$10 ;SET INT. BAUD RATE GENERATOR
CD84:4A CD85:4A CD86:4A CD87:4A	150         LSR           151         LSR           152         LSR           153         LSR           154         BAUDCMD2           155         STA	A A A A #\$10 ;SET INT. BAUD RATE GENERATOR ZPTMP1
CD84:4A CD85:4A CD86:4A CD87:4A CD88:09 10 CD88:85 2A CD8C:A9 E0	150         LSR           151         LSR           152         LSR           153         LSR           154         BAUDCMD2           155         STA           156         LDA	A A A A #\$10 ;SET INT. BAUD RATE GENERATOR ZPTMP1 #\$E0
CD84:4A CD85:4A CD86:4A CD87:4A CD88:09 10 CD88:85 2A CD8C:A9 E0 CD8E:85 2B	150         LSR           151         LSR           152         LSR           153         LSR           154         BAUDCMD2           155         STA           156         LDA           157         CTLREGSET	A A A A #\$10 ;SET INT. BAUD RATE GENERATOR ZPTMP1 #\$E0 TA ZPTMP2
CD84:4A CD85:4A CD86:4A CD87:4A CD88:09 10 CD8A:85 2A CD8C:A9 E0 CD8E:85 2B CD90:B9 8B C0	150         LSR           151         LSR           152         LSR           153         LSR           154         BAUDCMD2           155         STA           156         LDA           157         CTLREGSET           158         LDA	A A A A #\$10 ;SET INT. BAUD RATE GENERATOR ZPTMP1 #\$E0 TA ZPTMP2 CTLREG, Y
CD84:4A CD85:4A CD86:4A CD87:4A CD88:09 10 CD88:85 2A CD8C:A9 E0 CD8E:85 2B	150         LSR           151         LSR           152         LSR           153         LSR           154         BAUDCMD2           155         STA           156         LDA           157         CTLREGSET	A A A A #\$10 ;SET INT. BAUD RATE GENERATOR ZPTMP1 #\$E0 TA ZPTMP2 CTLREG, Y
CD84:4A CD85:4A CD86:4A CD87:4A CD88:09 10 CD8A:85 2A CD8C:A9 E0 CD8E:85 2B CD90:B9 8B C0	150         LSR           151         LSR           152         LSR           153         LSR           154         BAUDCMD2           155         STA           156         LDA           157         CTLREGSET           158         LDA	A A A A A #\$10 ;SET INT. BAUD RATE GENERATOR ZPTMP1 #\$E0 TA ZPTMP2 CTLREG,Y ZPTMP2
CD84:4A CD85:4A CD86:4A CD87:4A CD88:09 10 CD8A:85 2A CD8C:A9 E0 CD8E:85 2B CD90:B9 8B C0 CD93:25 2B	150         LSR           151         LSR           152         LSR           153         LSR           154         BAUDCMD2 ORA           155         STA           156         LDA           157         CTLREGSET ST           158         LDA           159         AND	A A A A A #\$10 ;SET INT. BAUD RATE GENERATOR ZPTMP1 #\$E0 TTA ZPTMP2 CTLREG,Y ZPTMP2 ZPTMP1
CD84:4A CD85:4A CD86:4A CD87:4A CD87:4A CD88:09 10 CD8A:85 2A CD8C:A9 E0 CD8E:85 2B CD90:B9 8B C0 CD93:25 2B CD95:05 2A	150         LSR           151         LSR           152         LSR           153         LSR           154         BAUDCMD2 ORA           155         STA           156         LDA           157         CTLREGSET STA           158         LDA           159         AND           160         ORA	A A A A A #\$10 ;SET INT. BAUD RATE GENERATOR ZPTMP1 #\$E0 TA ZPTMP2 CTLREG,Y ZPTMP1 CTLREG,Y
CD84:4A CD85:4A CD86:4A CD86:4A CD88:09 10 CD88:85 2A CD8C:A9 E0 CD8E:85 2B CD90:B9 8B C0 CD93:25 2B CD95:05 2A CD97:99 8B C0	150         LSR           151         LSR           152         LSR           153         LSR           154         BAUDCMD2 ORA           155         STA           156         LDA           157         CTLREGSET ST           158         LDA           159         AND           160         ORA           161         STA	A A A A A #\$10 ;SET INT. BAUD RATE GENERATOR ZPTMP1 #\$E0 TA ZPTMP2 CTLREG,Y ZPTMP1 CTLREG,Y
CD84:4A CD85:4A CD86:4A CD86:4A CD88:09 10 CD88:85 2A CD8C:A9 E0 CD8E:85 2B CD90:B9 8B C0 CD93:25 2B CD95:05 2A CD97:99 8B C0 CD9A:60	150         LSR           151         LSR           152         LSR           153         LSR           154         BAUDCMD2 ORA           155         STA           156         LDA           157         CTLREGSET ST           158         LDA           159         AND           160         ORA           161         STA           162         RTS	A A A A #\$10 ;SET INT. BAUD RATE GENERATOR ZPTMP1 #\$E0 TA ZPTMP2 CTLREG,Y ZPTMP2 ZPTMP1 CTLREG,Y
CD84:4A CD85:4A CD86:4A CD86:4A CD87:4A CD88:09 10 CD88:85 2A CD8C:A9 E0 CD82:85 2B CD90:B9 8B C0 CD93:25 2B CD95:05 2A CD97:99 8B C0 CD9A:60 CD9B:	150         LSR           151         LSR           152         LSR           153         LSR           154         BAUDCMD2 ORP           155         STA           156         LDA           157         CTLREGSET ST           158         LDA           159         AND           160         ORA           161         STA           162         RTS           163 *	A A A A #\$10 ;SET INT. BAUD RATE GENERATOR ZPTMP1 #\$E0 TA ZPTMP2 CTLREG,Y ZPTMP2 ZPTMP1 CTLREG,Y
CD84:4A CD85:4A CD86:4A CD87:4A CD88:09 10 CD8A:85 2A CD8C:A9 E0 CD8E:85 2B CD90:B9 8B C0 CD93:25 2B CD95:05 2A CD97:99 8B C0 CD93:60 CD98: CD9B:88 CD9C:	150         LSR           151         LSR           152         LSR           153         LSR           154         BAUDCMD2 ORA           155         STA           156         LDA           157         CTLREGSET ST           158         LDA           159         AND           160         ORA           161         STA           162         RTS           163 *         164           164         PARITYCMD DI	A A A A A A #\$10 ;SET INT. BAUD RATE GENERATOR ZPTMP1 #\$E0 TA ZPTMP2 CTLREG,Y ZPTMP1 CTLREG,Y ;TRICK: SO CTLREG,Y ACTUALLY
CD84:4A CD85:4A CD86:4A CD87:4A CD87:4A CD88:09 10 CD88:85 2A CD8C:A9 E0 CD8E:85 2B CD90:B9 8B C0 CD93:25 2B CD95:05 2A CD97:99 8B C0 CD93:60 CD98: CD9B:888 CD9C: CD9C:	150         LSR           151         LSR           152         LSR           153         LSR           154         BAUDCMD2 ORA           155         STA           156         LDA           157         CTLREGSET ST           158         LDA           159         AND           160         ORA           161         STA           162         RTS           163 *         164           165 *         166 *	A A A A ststo ;set int. baud rate generator zptmp1 #seo ta zptmp2 ctlreg, y zptmp1 ctlreg, y ;trick: So ctlreg, y actually addresses the command reg
CD84:4A CD85:4A CD86:4A CD86:4A CD88:09 10 CD88:85 2A CD8C:A9 E0 CD8E:85 2B CD90:B9 8B C0 CD93:25 2B CD95:05 2A CD97:99 8B C0 CD94:60 CD98: CD98:88 CD9C: CD9C: CD9C: CD9C:0A	150         LSR           151         LSR           152         LSR           153         LSR           154         BAUDCMD2 ORA           155         STA           156         LDA           157         CTLREGSET STA           158         LDA           159         AND           160         ORA           161         STA           162         RTS           163<*	A A A A A \$10 ;SET INT. BAUD RATE GENERATOR ZPTMP1 #\$E0 TA ZPTMP2 CTLREG,Y ZPTMP1 CTLREG,Y EY ;TRICK: SO CTLREG,Y ACTUALLY ADDRESSES THE COMMAND REG A ;SET NEW # OF DATA BITS
CD84:4A CD85:4A CD86:4A CD87:4A CD87:4A CD88:09 10 CD88:85 2A CD8C:A9 E0 CD8E:85 2B CD90:B9 8B C0 CD93:25 2B CD95:05 2A CD97:99 8B C0 CD93:60 CD98: CD9B:888 CD9C: CD9C:	150         LSR           151         LSR           152         LSR           153         LSR           154         BAUDCMD2 ORA           155         STA           156         LDA           157         CTLREGSET ST           158         LDA           159         AND           160         ORA           161         STA           162         RTS           163 *         164           165 *         166 *           167         DATACMD ASL	A A A A A A A A A A A A A A A A A A A
CD84:4A CD85:4A CD86:4A CD87:4A CD87:4A CD88:09 10 CD8A:85 2A CD8C:A9 E0 CD8C:A9 E0 CD82:85 2B CD90:B9 8B C0 CD93:25 2B CD95:05 2A CD97:99 8B C0 CD94:60 CD98: CD98:88 CD9C: CD9C: CD9C: OA CD9D:0A	150         LSR           151         LSR           152         LSR           153         LSR           154         BAUDCMD2 ORA           155         STA           156         LDA           157         CTLREGSET STA           158         LDA           159         AND           160         ORA           161         STA           162         RTS           163 *         164           165 *         166 *           166 *         167           167         DATACMD ASL           168         ASL	A A A A A A A A A A A A A A A SET INT. BAUD RATE GENERATOR ZPTMP1 *\$E0 CTLREG,Y ZPTMP2 ZPTMP2 ZPTMP1 CTLREG,Y ;TRICK: SO CTLREG,Y ACTUALLY ADDRESSES THE COMMAND REG A ;SET NEW # OF DATA BITS A A
CD84:4A CD85:4A CD85:4A CD86:4A CD87:4A CD88:09 10 CD84:85 2A CD8C:A9 E0 CD92:52 2B CD93:25 2B CD93:25 2B CD93:25 2A CD97:99 8B C0 CD93:60 CD98:60 CD98:88 CD9C: CD9C: CD9C:0A CD9C:0A	150         LSR           151         LSR           152         LSR           153         LSR           154         BAUDCMD2 ORA           155         STA           156         LDA           157         CTLREGSET STA           158         LDA           159         AND           160         ORA           161         STA           162         RTS           163         *           164         PARITYCMD DD           165 *         166 *           167         DATACMD ASL           168         ASL           169         ASL	A A A A A #\$10 ;SET INT. BAUD RATE GENERATOR ZPTMP1 #\$E0 TA ZPTMP2 ZPTMP2 ZPTMP1 CTLREG,Y :TRICK: SO CTLREG,Y ACTUALLY ADDRESSES THE COMMAND REG A ;SET NEW # OF DATA BITS A A A
CD84:4A CD85:4A CD86:4A CD86:4A CD87:4A CD88:09 10 CD88:85 2A CD8C:A9 E0 CD8E:85 2B CD90:B9 8B C0 CD93:25 2B CD95:05 2A CD97:99 8B C0 CD98: CD98:88 CD9C: CD9B:88 CD9C: CD9C:0A CD9C:0A CD9D:0A CD9F:0A CD9F:0A CDA0:0A	150         LSR           151         LSR           152         LSR           153         LSR           154         BAUDCMD2 ORA           155         STA           156         LDA           157         CTLREGSET STA           158         LDA           159         AND           160         ORA           161         STA           165 *         164           165 *         166 *           167         DATACMD ASL           168         ASL           169         ASL           170         ASL	A A A A A ststo ;set int. baud rate generator zptmp1 #\$e0 TA zptmp2 ctlreg, y zptmp1 ctlreg, y ;trick: SO ctlreg, y actually addresses the command reg A ;set new # OF data Bits A A A A
CD84:4A CD85:4A CD86:4A CD87:4A CD88:09 10 CD8A:85 2A CD8C:A9 E0 CD8E:85 2B CD90:B9 8B C0 CD93:25 2B CD95:05 2A CD97:99 8B C0 CD93:60 CD9B: CD9B:88 CD9C: CD9C: CD9C:0A CD9D:0A CD9F:0A	150         LSR           151         LSR           152         LSR           153         LSR           154         BAUDCMD2 ORA           155         STA           156         LDA           157         CTLREGSET ST           158         LDA           159         AND           160         ORA           161         STA           162         RTS           163 *         166 *           166 *         166 *           167         DATACMD ASL           168         ASL           169         ASL           169         ASL           170         ASL           171         ASL	A A A A A A A A A A A A A A A A A A A

	CDA5:DO	E7		174		BNE	CTLREGSET	; <always></always>
	CDAJ: DO			175				
	CDA7:1E	B8	04	0000	TERMCMD	ASL	STATEFLG, >	SET TERMINAL MODE
	CDAA: 38			177		SEC		
	CDAB: BO	10		178		BCS	QCMD1	; <always></always>
	CDAD:			179	*		~	
	CDAD: 99	89	CO			STA	RESET. Y	;DROP RTS, DTR
	CDB0:20			181		JSR		
	CDB3:20			182		JSR		;IN#O
	CDB6:AE			183		LDX	MSLOT	/******
	CDB9:1E				QUITCMD			CLEAR TERMINAL MODE
	CDB9:1E		04	185	An orally a second second		STATE DO, A	CLEAR IENTINAL HODE
	CDBC:18 CDBD:7E		04		QCMD1		STATEFLG, X	,
		BB	04		(CHD)	RTS	STATE LG, A	
	CDC0:60			187 188		RIS		
	CDC1:	07	00				CHOREC V	;SEND BREAK SIGNAL
	CDC1: B9		0	190	BREAKCMI		CHURCHI, I	
	CDC4:48					PHA ORA	#\$0C	; FOR 233 MILLISECONDS
	CDC5:09 CDC7:99		co	191 192		STA	#SOC CMDREG,Y	
	CDC7:99 CDCA:A9		0	192		LDA		DELAY FOR 222 MICROSEC
	CDCC:20	1000	CA	193		JSR	#233 WAITMS	;DELAY FOR 233 MICROSEC.
	CDCF:68	-4	CA	194		PLA		PREMORE OF COMMAND DEC COMMAND
		0.5	00	0.5152				RESTORE OLD COMMAND REG CONTENTS
	CDD0:99		CO	196		STA	CMDREG, Y	
	CDD3:60			197		RTS		
	CDD4:	-		198			and the second s	
	CDD4:A9				ICMD	LDA	#\$28	
	CDD6:9D		06	200		STA	PWDBYTE, X	;SET PRINTER WIDTH TO 40
	CDD9:A9			201		LDA	#\$80	
	CDDB:1D		07	202		ORA	MISCFLG, X	;SET SCREEN ECHO
	CDDE:DO	05		203		BNE	KCMD2	; <always></always>
	CDE0:			204	*			
	CDE0:A9	FE		205	KCMD	LDA	#\$FE	;RESET THE LF GENERATE FLAG
1	CDE2:3D	38	07	206	KCMD1	AND	MISCFLG, X	
	CDE5:9D	38	07	207	KCMD2	STA	MISCFLG, X	
	CDE8:60			208		RTS		
	CDE9:			209	*			
1	CDE9:C9	28		210	NCMD	CMP	#40	;>=40?
1	CDEB:90	OE		211		BCC		; IF NOT, JUST EXIT
1	CDED: 9D	38	06	212				SET NEW PRINTER WIDTH
	CDF0:A9			213		LDA	#\$3F	;DISABLE SCREEN, SET LISTING MODE
	CDF2:D0			214		BNE		; <always></always>
	CDF4:			215	*	DITE	North I	120000207
	CDF4:1E	39	05		ZCMD	AGT	CMDBYTE Y	DISABLE COMMAND RECOGNITION
	CDF 4: 12		05	210	acred	SEC	CHUDILE, X	IDIGADLE COMMAND RECOGNITION
	CDF8:7E		05	218			CMDBYTE, X	
	CDFB:60	50	05		ZCMDRTS		CHUBITE, X	
	CDFC:			219		KID.		
	CDFC:					*****	********	****
	CDFC:							COMMAND STATE *
	CDFC:							UMMAND STATE *
	CDFC:A8				CMDPROC			;A-REG=COMMAND STATE
	CDFD: A5	27		224		LDA		A-REGECOMMAND STATE
	CDFF:29			225				
	CE01:	12				AND	#\$7F	
		20		227	~	-		
	CE01:C9			228		CMP		;SKIP SPACES FOR ALL MODES
	CE03:DO			229		BNE	CMDPROC2	Contraction of the second s
	CE05:C0			230		CPY	#\$3	;EXCEPT MODE 3
	CE07:FO	10		231		BEQ	CMDPROC1	

6:3

1 3

3

-

1

. .

1

З

CE09:60			232	RTS		
CEOA: A9				CMDPROC1 LDA		
CEOC:DO	6D		234	BNE	SETOSTATE	; <always></always>
CEOE:			235	*		
CEOE:C9			236	CMDPROC2 CMP	#\$0D	;CARRIAGE RETURN?
CE10:D0	12		237	BNE	CMDPROC4	1
CE12:20	79	CE	238	JSR	ZEROSTATE	;ABORT FOR STATES 0-5, EXIT FOR 6,7
CE15:C0	07		239	CPY	#\$07	; IN STATE 7 WE VECTOR TO THE PROC
CE17:F0	01		240	BEQ	CMDPROC 3	;
CE19:60			241	RTS		;OTHERWISE, JUST EXIT
CE1A:			242	*		
CE1A:A9	CD		243	CMDPROC3 LDA	#\$CD	;ALL PROCS MUST START IN PAGE \$CD
CE1C:48			244	PHA		
CE1D: BD	38	04	245	LDA	PARAMETER,	X
CE20:48			246	PHA		
CE21:A4	26		247	LDY	SLOT16	;NEEDED BY BREAK CMD
CE23:60			248	RTS		
CE24:			249	*		
CE24:85	35		250	CMDPROC4 STA	ZPTEMP	
CE26:A9			251	LDA		;ALL ROUTINES MUST START
CE28:48			252	PHA		; IN PAGE \$CE
CE29:B9	30	CE	253	LDA	STATETBL,	
CE2C:48			254	PHA		
CE2D: A5	35		255	LDA	ZPTEMP	
CE2F:60			256	RTS		RTS TO COMMAND PROCEDURE
CE30:			257	*		
CE30:				* NOW THE STA	TE ROUTIN	ES
CE30:			259	*		
CE30:			260	*********	********	
CE30:			261	* STATE BRAN	CH TABLE	
CE30:			262	*********	********	•
CE30:A7			263	STATETBL DFB	>STATERR-	I ; BAD STATE
CE31:37			264	DFB	>CSTATE1-	1 ; <cmd> SEEN</cmd>
CE32:61			265	DFB	>CSTATE2-	ACCUMULATE PARAMETER
CE33:89			266	DFB	>CDONE-1	SKIP UNTIL SPACE
CE34:8A			267	DFB	>CSTATE4-	1 ;E/D SOMETHING
CE35: A7			268	DFB		1 ;ILLEGAL STATE
CE36:89			269	DFB	>CDONE-1	
CE37:89			270			SKIP UNTIL CR THEN DO CMD
CE38:			271	*********		Sector and the sector of the sector is the sector
CE38:				* COMMAND ST		
CE38:			1000	******		
CE38:DD	38	05				;IS IT <cmd>?</cmd>
CE3B:DO			275	BNE	CSTATE1A	
CE3D:DE		04	276	DEC		X ;SET STATE BACK TO ZERO
CE40:4C			277	JMP		;OUTPUT <cmd> IF SO</cmd>
CE43:			278			
CE43:C9	30		2000	CSTATE1A CMP	#\$30	;>=0?
CE45:90			280	BCC	CSTATE1B	
CE47:C9			281	CMP	#\$3A	;<=9?
CE49:B0			282	BCS	CSTATE1B	
CE4B:29			283	AND	#SOF	IT'S A NUMBER
CE4D: 9D		04	284	STA	PARAMETER	
CE50: A9		2.2	285	LDA	#2	A REAL PROPERTY OF A REAL PROPER
CE52:D0			286	BNE		; <always> SET MODE 2 AND RETURN</always>
CE52:00	41		287			
CE54:C9	20			CSTATE1B CMP	#\$20	IS IT A CONTROL CHAR?
CE56:B0			289		CSTATE1C	

F

1

1

1

-

-

-

	CE58:9D	38	05	290	STA CMDBYTE, X ; SET NEW COMMAND CHARACTER
	CE5B:4C	79	CE	291	JMP ZEROSTATE ; RESET STATE TO ZERO
	CE5E:			292	*
£	CE5E:A0	00		293	CSTATEIC LDY #0 ;USE COMMAND TABLE
٩	CE60: F0	4D		294	BEQ CMDSEARCH ; <always></always>
	CE62:				*****
£	CE62:				* COMMAND STATE 2: ACCUMULATE PARAMETER *
۰.	CE62:				***************
	CE62:49				CSTATE2 EOR #\$30 ;CONVERT \$30-\$39 TO 0-9
	CE64:C9			299	CMP #\$A ;0-9?
	CE66:B0			300	BCS CSTATE2A
	CE68: A0		04	301	LDY #\$A ; IT'S A NUMBER, SO ADD ACCLOOP ADC PARAMETER, X ; IT TO 10*PARAMETER
	CE6A:7D CE6D:88	38	04	302	DEY
	CE6E:DO	EA		304	BNE ACCLOOP
1	CE70:9D		04	305	STA PARAMETER, X
•	CE73:F0		0.4	306	BEQ CDONE ; <always></always>
	CE75:			307	
	CE75:A0	2E		308	CSTATE2A LDY #CMDTBL1-CMDTBL ;USE COMMAND TABLE
	CE77:D0	36		309	BNE CMDSEARCH ; <always></always>
	CE79:			310	******
	CE79:				* SET COMMAND STATE *
	CE79:				*****
÷	CE79: A9				ZEROSTATE LDA #0
	CE7B:85		~ 7		SETOSTATE STA ZPTMP1
	CE7D: AE			315	
	CE80: BD		04	316	LDA STATEFLG, X
•	CE83:29			317	AND #\$F8
	CE85:05 CE87:9D		04	318 319	
	CE87:90 CE8A:60	DO	04		CDONE RTS
-	CE8B:				*****
1	CE8B:			322	* COMMAND STATE 4 (E/D) *
	CE8B:				*******
	CE8B:A8			324	CSTATE4 TAY ; E/D -> Y-REG
1	CE8C: BD	38	04	325	LDA PARAMETER, X
ъ.,	CE8F:CO	44		326	CPY #\$44 ;D(ISABLE)?
	CE91:F0	100		327	BEQ CSTATE4A
	CE93:C0			328	CPY #\$45 ;E(NABLE)?
1	CE95:D0			329	
	CE97:1D		07	330	
	CE9A:DO	1.7.77		331	BNE CSTATE4B ; <always></always>
	CE9C:49				CSTATE4A EOR #\$FF ; INVERT FOR DISABLE
	CE9E: 3D			333	
	CEA1:9D CEA4:	38	07		CSTATE4B STA MISCFLG,X *******************
	CEA4:			1000	
	CEA4:				* ESCAPE TO STATE 6 * *********************************
1	CEA4: A9	06			SETSTATE6 LDA #6
-	CEA6:DO			339	
1	CEA8: A9				STATERR LDA #32 ;CODE FOR BAD COMMAND
2.5	CEAA:9D	B8	05	341	
	CEAD: DO	F5		342	
2	CEAF:			343	*****
	CEAF:			344	* TABLE DRIVEN COMMAND PROCESSOR *
1	CEAF:				*****
A	CEAF: B9				CMDSEARCH LDA CMDTBL,Y ;GET CANDIDATE CHARACTER
	CEB2:F0	F4		347	BEQ STATERR ; A ZERO MARKS THE END OF A SUBTABLE

CENS. 100 5 349 BEQ CANAMACCH CEB9:08 351 CMDSEARCH1 INY ;PERMETY FOR WRONG MODES CEB9:08 351 CMDSEARCH1 INY ;PERMETY FOR WRONG MODES CEB0:08 352 INY ;PENTRY LENGTH = 3 CEB0:08 355 CMDMATCH INY ;PENTRY LENGTH = 3 CEB0:08 355 CMDMATCH INY CEB0:08 355 CMDMATCH INY CEB0:08 355 CMDMATCH INY CEB2:39 20 358 AND 4520 ;CHECK PASCAL ENABLE CEC:180 20 358 CMDMATCH INT SO NS DONT CHECK P-BIT CEC:100 20 358 AND 4520 ;CHECK PASCAL ENABLE CEC:00 20 358 AND 4520 ;CHECK PASCAL ENABLE CEC:00 20 361 AND 4510 ; THAT WE AREN'T IN PASCAL CEC:00 20 361 AND 4510 ; THAT WE AREN'T IN PASCAL CEC:100 20 363 CMDMATCH IDA MISCPLG,X ;GET CIC/PPC BIT CEC:100 20 363 CMMATCH IDA MISCPLG,X ;GET CIC/PPC BIT CEC:140 365 LSR A ;SHIFT CIC/PPC MODE BIT TO CARRY CED1:4A 365 LSR A ;SHIFT CIC/PPC MODE BIT TO CARRY CED1:4A 365 LSR A ;SHIFT CIC/PPC MODE BIT TO CARRY CED1:4A 365 LSR A ;SHIFT CIC/PPC MODE BIT TO CARRY CED1:4A 365 LSR A ;SHIFT CIC/PPC MODE BIT TO CARRY CED1:4A 365 LSR A ;SHIFT CIC/PPC MODE BIT TO CARRY CED1:4A 365 LSR A ;SHIFT CIC/PPC MODE BIT TO CARRY CED1:4A 365 LSR A ;SHIFT CIC/PPC MODE BIT TO CARRY CED1:43 374 CMDEXEC LDA ZPTMP1 ;PEC-N CIC->V CED5:80 0C 371 CMEMATCH2 BVC CMDSEARCH1 ;NOT OK FOR PPC CED1:372 * CED1:43 374 FHA CED3:20 77 375 AND #S07 CEED:20 77 375 AND #S07 CEED:20 77 375 AND #S07 CEED:20 77 375 AND #S10 ; CEED:48 378 FLA CED1:48 374 FHA CEED:49 30 04 382 STA PARAMETER,X CEED:48 98 EC C3 381 LDA CMDTEL,Y CEEE:68 378 FLA CEED:49 38 04 382 STA PARAMETER,X CEEE:99 38 04 382 STA PARAMETER,X CEEE:99 38 04 382 STA PARAMETER,X CEEE:99 38 04 393 CHD SCM CMDTEL,Y CEEE:99 38 04 390 LDA SLOTI6 CEEF:60 391 RTS CEFF: 322 * CEFF:60 393 DFB \$00 SYMBOL TABLE SORTED BY SYMBOL 3C ALL CE6A ACCLOOP CC02 ACIAOUT 7CB9C ACK CB6B ABICCM1 CB6B ADINCIN C96B BAUCCM02 CACH BACKCMD C07B BAUCCMD C081 BAUNCIN C96F BACTCH02 CACH BACKCMD C77C BUTTOT C776 BUTTOT C778				13 00010	
CEB9:CG 351 CMDSEARCH1 INY ;PERMERY FOR WRONG MODES CEB9:CG 351 CMDSEARCH1 INY ;PERMERY FOR WRONG MODES (EB0:C3 352 INY ;PITRY LENGTH = 3 CEB0:C3 353 BRE CMDSEARCH ;ALMAYS) CEB0:354 * CEB0:354 * CEB0:354 * CEB0:355 CMDMATCH INY CEE0:352 20 358 AND 4520 ;CHECK PASCAL ENABLE CEC3:29 20 368 AND 4510 ; THAT WE ARE SURE CECA:29 10 361 AND 4510 ; THAT WE ARE SURE CECA:29 10 361 AND 4510 ; THAT WE ARE SURE CECA:29 10 361 AND 4510 ; THAT WE ARE SURE CECA:29 10 361 AND 4510 ; THAT WE ARE SURE CECA:29 10 364 CMDMATCH1 LDA MISCPLG,X ;GET CLC/PPC BIT CECA:29 10 364 CMDMATCH1 LDA MISCPLG,X ;GET CLC/PC BIT CECD:42 30 40 368 BCS CMDMATCH2 JBRANCH IF WE ARE CED3:24 2A 367 BIT ZIPMPI ;PEC->N CIC->V CED3:24 2A 369 BPL CMDSEARCH1 ;NOT OK FOR PC CED3:24 2A 367 BIT CMDEXEC ;AND OK CEDD:43 23 70 BMI CMDEXEC ;AND OK FOR PC CED7:10 E0 369 BPL CMDSEARCH1 ;NOT OK FOR CIC CED7:10 E0 369 BPL CMDSEARCH1 ;NOT OK FOR CIC CEDF:48 374 PHA CEED:48 374 PHA CEED:48 374 PHA CEED:48 374 PHA CEED:48 374 PHA CEED:48 374 PHA CEED:49 C3 370 CMDEXEC LDA ZPMP1 ;RETRIEVE TABLE MODE BYTE CEED:48 374 PHA CEED:48 374 PHA CEED:49 C3 370 AND #S10 ; CEED:48 374 PHA CEED:49 C3 370 AND #S10 ; CEED:48 374 PHA CEED:49 C3 370 AND #S10 ; CEED:49 C3 370 AND #S10 ; CEED:48 374 PHA CEED:49 C3 380 PHA CEED:49 C3 390 LDA PARAMETER,X CEED:48 388 PHA CEED:49 C3 390 LDA PARAMETER,X ;LOT OF ROUTINES NEED THIS CEEF:48 388 PHA CEFF:40 391 AFS CEFF:40 391 AFS CEFF:40 393 DFB S00 SYMBOL TABLE SORTED BY SYMBOL 3C ALL CEGA ACCLOOP CC2 ACIAOUT 7CB9C ACK CG8F ADJTPS C985 ADJUST CB95 ANTON CG8F BATCHIN C968 ADJTPS C985 ADJUST CB9 ANTON CG8F BATCHIN C960 ADJTPS C985 AD	CEB4:C5 35				
CEDENCE 351 CMDSEARCH1 INY ; PERTRY FOR WRONG MODES CEBA:CS 352 INY ; ENTRY LENGTH = 3 CEBB:D 2 353 BBE CMDSEARCH ; ALMANS> CEBD: 354 * CEBD: 354 * CEBD: 355 CMDMATCH INY CEBA:B9 EB CC 355 CMDMATCH INY CED: 85 2A 357 STA 2FM4P1 CEC3:29 20 358 AND #320 ; CHECK PASCAL ENABLE CEC: 30 359 BBE CMDMATCH ; IT'S ON 30 DONT CHECK P-BIT CEC3:29 10 361 AND #310 ; THAT WE AREN'T IN PASCAL CEC: B 362 CMDMATCH LDA MISCPLG, X ; OFF SO MAKE SURE CEC: B 363 * CEC: B 362 CMDMATCH LDA MISCPLG, X ; OFF SO MAKE SURE CEC: B 363 * CEC: B 364 CMDMATCH LDA MISCPLG, X ; GET CIC/PPC BIT CED 1: 4A 365 LSR A ; SHIFT CIC/PPC NDE BIT TO CARRY CED 3: 42 A 367 BIT ZPMP1 ; PPC->N CIC->V CED 3: 42 A 367 BIT ZPMP1 ; PPC->N CIC->V CED 3: 42 A 367 BIT ZPMP1 ; PPC->N CIC->V CED 3: 42 CM 368 BCS CMDMATCH ; BRANCH IF CIC MODE BIT TO CARRY CED 3: 42 CM 368 BCS CMDMATCH ; MOT OK FOR CIC CED 3: 32 * CED 1: 372 * CED 2: 373 CMDEXEC LDA ZPMM1 ; RETRIEVE TABLE MODE BYTE CED 2: 375 AND #\$10 ; CEE 30 01 399 AND #\$10 ; CEE 30 02 317 LDA CMDTEL, Y CEE 30 03 04 BNE CMDEXEC1 LDA #SCD ; ROUTINES MUST BE IN PAGE SCD CEF 30 03 04 BAS CMDEXEC1 LDA #SCD ; ROUTINES MUST BE IN PAGE SCD CEF 30 03 04 DA RARAMETER, X ;LOT OF ROUTINES NEED THIS CEF 30 03 04 DA RARAMETER, X ;LOT OF ROUTINES NEED THIS CEF 30 03 04 DA RARAMETER, X ;LOT OF ROUTINES NEED THIS CEF 30 03 05 DA SME DIA ST CE3 9 ANTTS CC37 ASCREEN CEF 10 33 0 FB 500 SYMEOL TABLE SORTED BY SYMBOL 3C ALL CE6A ACCLOOP CC02 ACIAOUT 7CB9C ACK CEB BINKTD CHS BANCTA1 CBF BINKTD CHS BINTTD					
CEBA:CG         352         TNY         PENTRY LENGTH = 3           CEBA:CG         352         TNY         PENTRY LENGTH = 3           CEBD:         353         NE CMDSEARCH ; (ALWAYS)           CEBD:         355         CMDMATCH INY           CEBR:B9 EB CC 356         LA CMDTBL, Y           CEC1:155         2A         357           STA ZPTMP1         STA ZPTMP1           CEC3:29         358         AND #\$20           CEC1:30         360         LAA MISCPLG, X; JOPF SO MARE SURE           CECA:29         361         AND #\$10         , THAT WE AREN'T IN PASCAL           CECA:29         363         AND #\$10         , THAT WE AREN'T IN PASCAL           CECA:29         364         CMDMATCH1 JAM MISCPLG, X; GET CIC/PPC BIT           CECE:         363         SEX A ;SHIFT CIC/PPC MODE BIT TO CARRY           CED1:4A         366         LSR A ;SHIFT CIC/PPC MODE BIT TO CARRY           CED1:4A         366         LSR A ;SHIFT CIC/PPC MODE BIT TO CARRY           CED3:10         237         CMDMATCH2 BWC CMDSEARCH1 ;NOT OK FOR PPC           CED5:10         371 <cmomatch2 ;not="" bwc="" cic<="" cmdsearch1="" for="" ok="" td="">           CED1:43         374         PHA           CEED1:52         375         AND #\$507</cmomatch2>					
CEBB:D0 F2         353         BNE CMDSEARCH ; <always>           CEBD:         354 *           CEBD:6         355 CMEMATCH INY           CEBD:8         354 *           CES:10:         354 *           CED:6         355 CMEMATCH INY           CES:10:         354 *           CEC:11:5:         24           CES:10:         358           CES:10:         358           CEC:11:5:         20           SEC         358           CEC:11:5:         20           SEC         358           CEC:11:5:         20           SEC         361           CEC:11:5:         363 *           CEC:11:4:         365           CEC:11:5:         33 *           CEC:11:4:         365           CEC:11:5:         33 *           CEC:11:4:         365           CED:1:4:         365           CED:1:4:         365           CED:1:4:         366           CEC:1:1:4:         366           CEC:1:1:4:         366           CEC:1:1:4:         367           CED:1:5:2:2:4:         363           CED:1:5:2:2:2:4:         373<!--</td--><td></td><td></td><td></td><td></td><td></td></always>					
CEBD:         354 *           CEBD:CB         355 CMDMATCH INY           CEBD:165 2A         357         STA ZPTMP1           CEC1:29 20         358         AND #\$20         ;CHECK PASCAL ENABLE           CEC3:29 20         358         AND #\$20         ;CHECK PASCAL ENABLE           CEC3:29 20         358         AND #\$10         ;ITYS ON SO DONT CHECK P-BIT           CEC:100 07         359         BRE CMMATCH ;JTYS ON SO DONT CHECK P-BIT           CEC:100 08         362         BRE CMDERARCH ;JENACH IP WE AREN'T IN PASCAL           CECCE:         33 *           CECE:         33 *           CECE:         364           CECE:         365           LSR A         ;SHIPT CLC/PEC MODE BIT TO CARRY           CED1:4A         365           CED3:20 04         366           DSC CMDMATCH1 LDA MISCPLG,X ;GET CLC/PEC MODE BIT TO CARRY           CED3:20 02         370           CED3:20 02         370           CED3:20 02         370           CED1:4A         365           CED3:20 02         370           CED3:20 02         370           CED3:20 02         370           CED3:20 02         370           CED3:20					3
CEBD;C8         355         CMDMATCH INY           CEBD;C8         355         CMDMATCH INY           CEC1:85         23         357         STA ZPTMPI           CEC3:10         00         358         AND #\$20         ;CHECK PASCAL ENABLE           CEC3:10         03         07         360         LDA MISCFLG,X;OFF SO MAKE SURE           CEC3:10         03         07         AND #\$10         ; THAT WE AREN'T IN PASCAL           CEC4:29         10         361         AND #\$10         ; THAT WE AREN'T IN PASCAL           CEC2:10         361         AND #\$10         ; THAT WE AREN'T IN PASCAL           CEC2:24         365         LSR A         ; SHIFT CIC/PPC BIT           CED1:40         365         LSR A         ; SHIFT CIC/PPC MODE BIT TO CARRY           CED2:44         366         LSR A         ; SHIFT CIC/PPC MODE BIT TO CARRY           CED3:24         24         367         BIT 2PTMP1 ; PPC->N CIC->V           CED5:10         04         368         BCS         CMDMATCH ; BRANCH IF CIC MODE           CED7:10         03         BNI CMDEXEC ; JAND OK         CIC         CED8:10           CED1:30         2370         EMEXEC CMDEXECC1 ; JOT OK FOR CIC         CED9:13 <t< td=""><td>CEBB:D0 F2</td><td></td><td>BNE CMDSEARCH</td><td>;<always></always></td><td></td></t<>	CEBB:D0 F2		BNE CMDSEARCH	; <always></always>	
CEBE:B9         EB         CC         356         LDA         CMOTEL,Y           CEC1:B5         2A         357         STA         ZPTMP1           CEC3:B5         20         358         AND         #\$20         ;CHECK PASCAL ENABLE           CEC3:D0         07         359         ENE         CMMATCH1         JIT'S ON SO DONT CHECK PABLT           CEC7:B0         80         7         66         LDA         MISCFLG(X, 10PF SO MAKE SURE           CEC2:D0         EB         361         AND #\$10         ; THAT WE AREN'T IN PASCAL           CEC2:D1         363         *         CEC2:D1         363         *           CEC2:24.4         366         LSR A         ; SHETC CLOPPC MODE BIT TO CARRY           CED3:24         2A         367         BIT ZETWP1         ; PPC->N CIC->V           CED5:80         04         368         BCS CMMATCH2 ; BRANCH IP CIC MODE           CED1:43         2A         371         CMMATCH2 BWC CMOSEARCH1 ; NOT OK FOR CIC           CED1:45         2A         373         CMDEXEC LDA         ZPTMP1           CED1:45         2A         373         CMDEXEC LDA         ZPTMP1           CED1:45         2A         374         PIA <td>CEBD:</td> <td>354 *</td> <td></td> <td></td> <td></td>	CEBD:	354 *			
CEC1:85 2A       357       STA       ZPTMP1         CEC3:10       0358       AND       \$20       ;CHECK PASCAL ENABLE         CEC5:10       0360       LDA       MISCFLG,X ;OFF SO MAKE SURE         CECA:29       01       361       AND       \$10         CECA:29       01       361       AND       \$10       ;THAY WE AREN'T IN PASCAL         CECA:29       01       361       AND       \$10       ;THAY WE AREN'T IN PASCAL         CECA:29       363       *       GECEND       363       *         CECE:80       364       CHEMATCH1 LDA MISCFLG,X ;GET CIC/PPC MDE BIT TO CARRY       GED         CED2:4A       366       LSR A       ;SHIFT CIC/PPC MDE BIT TO CARRY         CED2:4A       366       LSR A       ;GED CIC/PC MODE BIT TO CARRY         CED2:4A       366       LSR A       ;GED CIC/PC MODE BIT TO CARRY         CED3:40       2.4       367       BIL CMDEXEC ; #AND OK       CCC         CED1:520       372       *       CED:4       374       PHA         CED1:520       373       CMDEXEC LDA       ZPTMP1       ;PETRIEVE TABLE MODE BYTE         CED2:40       CBS CB       376       JAD       #507         C	CEBD:C8	355 CMDMATCH	INY		
CEC3:12       20       358       AND       #20       ;CHECK PASCAL ENABLE         CEC3:10       07       359       ENE       CMDMATCH1 ;IT'S ON SO DONT CHECK P-BIT         CEC7:10       380       7       60       LDA       MISCFLG,X, JOPF SO MAKE SURE         CECA:12       10       361       AND       #\$10       ; THAT WE AREN'T IN PASCAL         CECC:10       B8       67       364       CMDMATCH1 ;BRANCH IF WE ARE         CECE:       363       *       CEC1006       B0         CED1:4A       365       LSR       A       ;SHIFT CIC/PPC MODE BIT TO CARRY         CED3:24       366       LSR       A       ;SHIFT CIC/PPC MODE BIT TO CARRY         CED3:24       366       LSR       A       ;SHIFT CIC/PPC MODE BIT TO CARRY         CED3:24       366       LSR       A       ;SHIFT CIC/PPC MODE BIT TO CARRY         CED3:24       366       LSR       A       ;SHIFT CIC/PPC MODE BIT TO CARRY         CED3:30       04       368       BCS       CMDATCH2 ;BRANCH I; OT CIC->V         CED5:50       04       368       BCS       CMDATCH2 ;BRANCH I; OT CIC CONDE         CED0:35       27       TOMDATCH2 BVC CMDSEARCH1 ;NOT OK FOR CIC       CED1:48	CEBE: B9 EB CC	356	LDA CMDTBL, Y		
CECS:D0       07       359       ENE       COMMANCH1 jIT'S ON SO DONT CHECK P-BIT         CEC7:BD       36 07       160       LDA       MISCFLG,X jOFP SO MAKE SURE         CECA:29 10       361       AND       #\$10       , THAY WE AREN'T IN PASCAL         CECC:100       EB       362       ENE       CMDSEARCH1 ;BRANCH IP WE ARE         CECE:       363 *         CECE:       363 *         CECE:       363 *         CED2:4A       366       LSR A         CED1:4A       365       LSR A         CED2:4A       366       LSR A         CED3:24       2A       367       EIT 2FMP1 ;PC->N CIC->V         CED3:30       02       370       BMI CMDEXEC ;AND OK         CED5:40       04       368       BCS CMDMATCH2 ;BRANCH IP CIC MODE         CED1:0       G2       370       BMI CMDEXEC ;AND OK       FOR CIC         CED1:3       72 *       CED1:4A       373 CMDEXEC LDA ZPTMP1 ;PETRIEVE TABLE MODE BYTE       CEES:20         CED2:30       73       375       AND #\$01 ;       ;       CEES:20         CED2:30       73       375       AND #\$01 ;       ;       CEES:20         CEES:20       76       JSR<	CEC1:85 2A	357	STA ZPTMP1		
CEC7: DD 38 07 560       LDA MISCFLG,X; OFF SO MAKE SURE         CECA: 29 10       361       AND #\$10       ; THAT WE AREN'T IN PASCAL         CECC: DD 361       AND #\$10       ; THAT WE AREN'T IN PASCAL         CECC: DD 360       364       CONSEARCH1; BRANCH IF WE ARE         CECE: DD 38 07 364 (MOMATCH1 LDA MISCFLG,X; GET CLC/PFC BIT       CECERED 38 07 364 (MOMATCH1 LDA MISCFLG,X; GET CLC/PFC BIT         CED1: 4A       365       LSR A       ;SHIFT CLC/PFC MODE BIT TO CARRY         CED2: 42       365       LSR A       ;SHIFT CLC/PFC MODE BIT TO CARRY         CED3: 42 2A       365       BK       CS CMOMATCH2; BRANCH IF VEC MODE BIT TO CARRY         CED3: 42 2A       365       BMI CMDEXEC: ;RANCH IF OIC MODE       CED9: 30 02         CED1: 4A       364       BMI CMDEXEC: ;AND OK       CED0: 372 *         CED0: A5 2A       373       CMDEXEC LDA ZPTMP1 ;RETRIEVE TABLE MODE BYTE         CEED: 43       374       PHA       PHA         CEED: 43       374       PHA       PHA         CEED: 43       374       PHA       PHA         CED0: 45       2A       375       AND #\$07       CEED:         CEED: 43       374       PHA       PHA       PHA         CEED: 44       374       PHA	CEC3:29 20	358	AND #\$20	;CHECK PASCAL ENA	BLE
CECA:29 10       361       AND #\$10       ; THAT WE AREN'T IN PASCAL         CECC:       363       *         CECCE:       363       *         CECE:       363       *         CED:       366       LSR A         SHIP CIC/PPC BIT       CED:       *         CED:       807       366         CED:       807       366         CED:       807       SEC         CED:       803       *         CED:       369       BPL         CED:       803       *         CED:       370       EMCMATCH:       PRACH IP CIC MODE         CED:       371       CMCMATCH:       PRACH IP CIC MODE         CED:       372       *       CED:       373         CED:       373       CMDEXEC IDA       ZPTMP1       // RETRIEVE TABLE MODE BYTE         CEED:29       07       375       AND #\$10       ;         CEED:29       07       380       ENE CMDEXEC1 ; PR HET 4 IS SET, VECTOR TO ROUTINE </td <td>CEC5:D0 07</td> <td>359</td> <td>BNE CMDMATCH1</td> <td>;IT'S ON SO DONT</td> <td>CHECK P-BIT</td>	CEC5:D0 07	359	BNE CMDMATCH1	;IT'S ON SO DONT	CHECK P-BIT
CECC:D0       EB       362       ENE       CMDSEARCH1 ; BRANCH IF WE ARE         CECE:E       363 *         CECE:ED       360 *         CED1:4A       365       LSR A ; SHIFT CIC/PPC BIT         CED1:4A       366       LSR A ; SHIFT CIC/PPC MODE BIT TO CARRY         CED2:24 A       366       LSR A ; SHIFT CIC/PPC MODE BIT TO CARRY         CED2:24 A       367       BIT ZPTMP1 ; PPC->N CIC->V         CED2:32 42 A       367       BIT ZPTMP1 ; PPC->N CIC->V         CED2:30 02       370       BMI CMDEXEC ; RANCH IF CIC MODE         CED1:37       CMMARCH2 BVC CMDSEARCH1 ; NOT OK FOR CIC         CED0:37       371       CMDEXEC ; AND oK         CED0:35       2A       373       CMDEXEC 1DA ZPTMP1 ; RETRIEVE TABLE MODE BYTE         CEED:43       374       PHA         CEED:39       373       CMDEXEC1 LDA ZPTMP1 ; RETRIEVE TABLE MODE BYTE         CEED:43       374       PHA         CEED:43       374       PHA         CEED:29       07       375       AND #\$10 ;         CEES:20       07       370       BND         CEES:20       380       BA       CMDEXEC1 ; JF BIT 4 IS SET, VECTOR TO ROUTINE         CEES:48       380       BA	CEC7: BD 38 07	360			
CECE:       363 *         CECE:       36 07 364 CMDMATCH1 LDA MISCFLG,X ;GET CIC/PPC BIT         CED1:4A 365 LSR A ;SHIFT CIC/PPC MODE BIT TO CARRY         CED2:4A 366 LSR A         CED3:24 2A 367 BIT ZPTMP1 ;PC->N CIC->V         CED5:8D 004 368 BCS CMDMATCH2 ;PRANCH IF CIC MODE         CED7:10 E0 369 BPL CMDSEARCH1 ;NOT OK FOR PC         CED9:30 02 370 BMI CMDEXEC ;AND OK         CED1:5 DC 371 CMDMATCH2 BVC CMDSEARCH1 ;NOT OK FOR CIC         CED0: 372 *         CED0:5 20 73 75 AND #\$07         CEE2:20 77 B CE 376 JSR SETOSTATE ;SET NEXT STATE         CEE5:68 377 INY         CEE8:50 77 G CE 376 JSR SETOSTATE ;SET NEXT STATE         CEE8:68 378 PLA         CEE8:90 38 04 382 STA PARAMETER,X         CEE1:60 383 RTS         CEE7: 384 *         CEF7: 392 A         CEF7: 392 HA         CEF7:       392 *         CEF7:	CECA:29 10	361	AND #\$10	; THAT WE AREN'T	IN PASCAL
CECE:ED 38 07       364 CMDMATCH1 LDA MISCFLG, X ; GET CIC/PPC BIT         CED1:4A       365       LSR A       ;SHIFT CIC/PPC MODE BIT TO CARRY         CED2:4A       366       LSR A       ;SHIFT CIC/PPC MODE BIT TO CARRY         CED2:4A       366       LSR A       ;SHIFT CIC/PPC MODE BIT TO CARRY         CED3:24 2A       367       BIT ZPTMP1       ;PRC->N CIC->V         CED3:10       60       BSP C MDSEARCH1; NOT OK FOR PPC       CED3:10         CED3:30       02       370       BMI CMDEXEC ;AND OK       CEC         CED5:10       60       372 *       CED5:10 CK       371 CMDMATCH2 BVC CMDSEARCH1; NOT OK FOR CIC         CED5:20       77       AND #\$07       CEE0:20 OT 375       AND #\$07       CEE0:20 OT 375         CEE0:20       73       AND #\$10       ;       CEE5:20 OT 379       AND #\$10       ;         CEE5:20       07       379       AND #\$10       ;       CEE6:68       378       PLA         CEE5:21:20       10       379       AND #\$10       ;       CEE9:10       380       RES       CMDEXEC1 ; IF BIT 4 IS SET, VECTOR TO ROUTINE         CEE5:29       18 04       382       STA PARAMETER, X       CEE7:10       383       RTS         CEF2:49       C	CECC:DO EB	362	BNE CMDSEARCH	1 ; BRANCH IF WE AR	E
CED1:4A       365       LSR A       ;SHIFT CIC/PPC MODE BIT TO CARRY         CED2:4A       366       LSR A       ;SHIFT CIC/PPC MODE BIT TO CARRY         CED3:24 2A       367       BIT ZPTMP1       ;PPC->N CIC->V         CED5:80 04       368       BCS CMDMATCH2       ;BRANCH IF CIC MODE         CED7:10 E0       369       BPL CMDSEARCH1       ;NOT OK FOR PPC         CED9:30 02       370       BMI CMDEREC       ;AND OK         CED9:48       374       PHA         CED0:43       373       CMDEXEC LDA       ZPTMP1         CED0:43       373       CMDEXEC LDA       ZPTMP1         CED0:43       374       PHA         CEE0:29 07       375       AND #\$07         CEE2:20 70 B CE 376       JSR SETOSTATE ;SET NEXT STATE         CEE5:68       377       INY         CEE1:29 10       379       AND #\$10         CEE1:29 10       379       AND #\$10         CEE1:60       381       LDA       CMDTEXEX, X         CEE1:60       383       RTS         CEE1:60       383       RTS         CEF2:384       386       PHA         CEF2:49       385       CMDTEXEX, X; LOT OF ROUTINES NEED THIS	CECE:	363 *			
CED2:4A       366       LSR A         CED3:80 04       367       BIT ZPTMP1 ;PC->N CIC->V         CED3:80 04       368       BCS CMOMATCH2 ;BRANCH IF CIC MODE         CED7:10 E0       369       BPL CMDSEARCH1 ;NOT OK FOR PPC         CED8:80 02       370       BMI CMDEXEC ;AND OK         CED8:50 02       371       CMDMATCH2 BVC CMDSEARCH1 ;NOT OK FOR CIC         CED0:       372 *         CED0:A5 2A       373 CMDEXEC LDA ZPTMP1 ;RETRIEVE TABLE MODE BYTE         CED1:A5 2A       373 CMDEXEC LDA ZPTMP1 ;RETRIEVE TABLE MODE BYTE         CEE0:29 07       375       AND #\$07         CEE5:68       377       INY         CEE5:68       378       PLA         CEE7:29       10       379       AND #\$10 ;         CEE8:00 07       380       BNE CMDEXEC1 ;IF BIT 4 IS SET, VECTOR TO ROUTINE         CEE8:189 EB CC 381       LDA CMDTBL, Y         CEF2:49 CD       383       RTS         CEF2:49 CD       386       PHA         CEF2:89 CB C3 87       LDA CMDTBL, Y         CEF3:89 EB CC 387       LDA CMDTBL, Y         CEF3:48       386       PHA         CEF9:43 26       390       LDA PARAMETER, X ;LOT OF ROUTINES NEED THIS         CEF9:43	CECE: BD 38 07	364 CMDMATCH	1 LDA MISCFLG,	X ;GET CIC/PPC BIT	
CED3:24 2A       367       BIT ZPTMP1 ;PPC->N CIC->V         CED3:24 2A       367       BIT ZPTMP1 ;PPC->N CIC->V         CED5:80 04       368       BCS CMDMATCH2 ;BRANCH IF CIC MODE         CED7:10 E0       369       BPL CMDSEARCH1 ;NOT OK FOR PPC         CED3:30 02       370       BMI CMDEXEC ;AND OK         CCD0:       372 *         CCDD:       372 *         CCD0:A5 2A       373 CMDEXEC LDA ZPTMP1 ;RETRIEVE TABLE MODE BYTE         CED2:20 7B CE 376       JSR SETOSTATE ;SET NEXT STATE         CEE2:20 7B CE 376       JSR SETOSTATE ;SET NEXT STATE         CEE5:68       377       INY         CEE6:29 07       379       AND #\$10 ;         CEE6:30       371       MOBEXEC1 ;IF BIT 4 IS SET, VECTOR TO ROUTINE         CEE6:48       376       PLA         CEE6:29 10       379       AND #\$10 ;         CEE6:30       804       382       STA PARAMETER,X         CEF1:60       383       RTS         CEF2:39 CD       385 CMDEXEC1 LDA #SCD ;ROUTINES MUST BE IN PAGE SCD         CEF4:48       386       PHA         CEF5:19 EB CC 387       LDA CMDTEL,Y         CEF8:48       388       PHA         CEF9:19 38 04 390       LDA PARAMETER,X ;LOT	CED1:4A	365	LSR A	;SHIFT CIC/PPC MO	DE BIT TO CARRY
CED5:B0 04       368       BCS CMDMATCH2 ;BRANCH IP CIC MODE         CED5:B0 04       369       BLL CMDSEARCH1 ;NOT OK FOR PPC         CED9:30 02       370       BMI CMDEXEC ;AND OK         CED8:50 DC       371 CMDMATCH2 BVC CMDSEARCH1 ;NOT OK FOR CIC         CED0:372 *         CED0:45 2A       373 CMDEXEC LDA ZPTMP1 ;RETRIEVE TABLE MODE BYTE         CED0:45 2A       373 CMDEXEC LDA ZPTMP1 ;RETRIEVE TABLE MODE BYTE         CED0:45 2A       373 CMDEXEC LDA ZPTMP1 ;RETRIEVE TABLE MODE BYTE         CED0:45 2A       373 CMDEXEC LDA ZPTMP1 ;RETRIEVE TABLE MODE BYTE         CED0:45 2A       373 CMDEXEC LDA ZPTMP1 ;RETRIEVE TABLE MODE BYTE         CED1:45 2A       373 CMDEXEC LDA ZPTMP1 ;RETRIEVE TABLE MODE BYTE         CED2:40 77 375       AND #\$10 ;         CEE5:68       377 INY         CEE5:19 01 379       AND #\$10 ;         CEE9:100 7380       BNE CMDEXEC1 ;IF BIT 4 IS SET, VECTOR TO ROUTINE         CEE9:100 380 4362       STA PARAMETER, X         CEF1:60 381       LDA CMDTBL, Y         CEF2:89 EB CC 387       LDA CMDTBL, Y         CEF9:80 38 04 390       LDA PARAMETER, X ;LOT OF ROUTINES NEED THIS         CEF9:10 393       DFB \$00         SYMBOL TABLE       SORTED BY SYMBOL         3C A1L       CE6A ACCLOOP       CC02 ACIAOUT<	CED2:4A	366	LSR A		
CED7:10 E0       369       BPL CMDSEARCH1 ;NOT OK FOR PPC         CED9:30 02       370       BHI CMDEXEC ;AND OK         CEDB:50 DC       371       CMDMATCH2 BVC CMDSEARCH1 ;NOT OK FOR CIC         CEDD:372 *       CRDDA52 A       373 CMDEXEC LDA ZPTMP1 ;RETRIEVE TABLE MODE BYTE         CED:43       374       PHA         CED:20 7       375       AND #\$07         CEE6:29 07       375       AND #\$10 ;         CEE6:68       377       INY         CEE6:68       378       PLA         CEE7:29 10       379       AND #\$10 ;         CEE9:100 7       380       BNE CMDEXEC1 ;IF BIT 4 IS SET, VECTOR TO ROUTINE         CEE8:9D 38 04 382       STA PARAMETER,X         CEF2:304 *       CEF2:304 *         CEF2:49 CD       385 CMDEXEC1 LDA #SCD ;ROUTINES MUST BE IN PAGE SCD         CEF8:48       386       PHA         CEF9:44 26       389       LDY SLOT16         CEF8:48       386       PHA         CEF9:50       393       DFB \$00         SYMBOL TABLE       SORTED BY SYMBOL         3C A1L       CE6A ACCLOOP       CC02 ACIAOUT       7CB9C ACK         CEF8:60       393       DFB \$00         SYMBOL TABLE       SORT	CED3:24 2A	367	BIT ZPTMP1	; PPC->N CIC->V	
CED7:10 E0       369       BPL CMDSEARCH1 ;NOT OK FOR PPC         CEDB:30 02       370       BHI CMDEXEC ;AND OK         CEDB:50 DC       371       CMDMATCH2 BVC CMDSEARCH1 ;NOT OK FOR CIC         CEDD:372 *       CCDD:A5 2A       373 CMDEXEC LDA ZPTMP1 ;RETRIEVE TABLE MODE BYTE         CED:48       374       PHA         CED:29 07       375       AND #\$07         CEE6:29 07       375       AND #\$10 ;         CEE6:68       377       INY         CEE6:68       378       PLA         CEE7:29 10       379       AND #\$10 ;         CEE9:100 7       380       BNE CMDEXEC1 ; IF BIT 4 IS SET, VECTOR TO ROUTINE         CEE8:90 38 04 382       STA PARAMETER,X         CEF2:3 04 *       CEF2:384 *         CEF2:49 CD       385 CMDEXEC1 LDA #SCD ;ROUTINES MUST BE IN PAGE \$CD         CEF8:48       386       PHA         CEF9:49 CD       380 LDA PARAMETER,X ;LOT OF ROUTINES NEED THIS         CEF8:48       386       PHA         CEF9:50       393       DFB \$00         SYMBOL TABLE       SORTED BY SYMBOL         SC A1L       CE6A ACCLOOP       CC02 ACIAOUT       ?CB9C ACK         CEF9:60       393       DFB \$00       CSTA ASCREEN			BCS CMDMATCH2	; BRANCH IF CIC MO	DE
CED9:30 02       370       BMI CMDEXEC ;AND OK         CED9:30 02       371 CMDMATCH2 BVC CMDSEARCH1;NOT OK FOR CIC         CED0:372 *       CCD0145 2A       373 CMDEXEC LDA ZPTMP1;RETRIEVE TABLE MODE BYTE         CED0:45 2A       373 CMDEXEC LDA ZPTMP1;RETRIEVE TABLE MODE BYTE         CED0:70 70 75 AND #\$07       CEE2:20 77 ATE       SETOSTATE;SET NEXT STATE         CEE2:20 78 CE 376 JSR SETOSTATE;SET NEXT STATE       CEE6:68       378 PLA         CEE7:29 10 379 AND #\$10 ;       CEE6:00 07 380       BNE CMDEXEC1 ;IF BIT 4 IS SET, VECTOR TO ROUTINE         CEE9:00 07 380       BNE CMDEXEC1 ;IF BIT 4 IS SET, VECTOR TO ROUTINE       CEE9:10 383 RTS         CEE9:00 38 04 382       STA PARAMETER,X       CEF1:60 383 RTS         CEF2:384 *       CEF2:384 *       CEF9:89 CD 386 CMDEXEC1 LDA #\$CD ;ROUTINES MUST BE IN PAGE SCD         CEF4:48 386 PHA       CEF9:10 391 RTS       CEF9:10 S0116         CEF9:10 391 RTS       CEF9:392 *       CEF7:392 *         CEFF:0 391 RTS       CG95 ADJUST CB59 ANRTS       CCB7 ASCREEN         CG88 BASICEXIT 28 BASL ?C93D BATCHIN C98F BATCHIO       C968 ADJUST CB59 ANRTS       CCB7 ASCREEN         C3C ALL       CE6A ACCLOOP       C022 ACIAOUT       ?CB9C ACK         CGF95 ADJUST CB59 ANRTS       CCB7 ASCREEN       CG85 BADJUST CB59 ANRTS       CCB7 ASCREEN         C3					
CEDB: 50 DC       371 CMDMATCH2 BVC CMDSEARCH1 ;NOT OK FOR CIC         CEDD:       372 *         CEDD:A5 2A       373 CMDEXEC LDA ZPTMP1 ;RETRIEVE TABLE MODE BYTE         CEDF:48       374       PHA         CEE0:29 07       375       AND #\$07         CEE2:20 7B CE 376       JSR SETOSTATE ;SET NEXT STATE         CEE5:68       377       INV         CEE6:68       378       PLA         CEE7:19 10       379       AND #\$10 ;         CEE9:00 07       380       BNE CMDEXEC1 ;IF BIT 4 IS SET, VECTOR TO ROUTINE         CEE8:90 38 04       382       STA PARAMETER,X         CEF2:       384 *       CEF2:         CEF2:39 CD       385 CMDEXEC1 LDA #\$CD ;ROUTINES MUST BE IN PAGE \$CD         CEF4:48       386       PHA         CEF9:44       386       PHA         CEF9:44       389       LDA CMDTEL,Y         CEF8:10 39 0 LDA PARAMETER,X ;LOT OF ROUTINES NEED THIS       CEF7:         CEF7:       392 *					
CEDD:       372 *         CEDD:A5 2A       373 CMDEXEC LDA       ZPTMP1       ;RETRIEVE TABLE MODE BYTE         CEDF:48       374       PHA         CED:30 07       375       AND       #\$07         CEE2:20 7B CE       376       JSR       SETOSTATE ;SET NEXT STATE         CEE5:C8       377       INY         CEE6:68       378       PLA         CEE7:29 10       379       AND       #\$10         CEE8:89       EB CC       281       LDA       CMDEXEC1       JF BIT 4 IS SET, VECTOR TO ROUTINE         CEE8:89       EB CC       381       LDA       CMDTBL, Y       CEEF:2:       384 *         CEF2:49       CD       385       CMDEXEC1 LDA #SCD       ;ROUTINES MUST BE IN PAGE SCD         CEF2:49       CD       385       CMDEXEC1 LDA #SCD       ;ROUTINES MUST BE IN PAGE SCD         CEF2:48       386       PHA       CEF9:80       280 4       390       LDA CMDTBL, Y         CEF2:80       287       LDA       CMDTRL, Y       CEF8:80 380 4       390       LDA PARAMETER, X ;LOT OF ROUTINES NEED THIS         CEF7:0       391       RTS       CEF7:       392 *       CEF7:       392 *         CEF7:0       393					
CEDD:A5       2A       373       CMDEXEC       LDA       ZPTMP1       ;RETRIEVE TABLE MODE BYTE         CED:48       374       PHA         CEE0:29       07       375       AND       #\$07         CEE2:20       7B       CE       376       JAND       #\$07         CEE2:20       7B       CE       376       JAND       #\$10       ;         CEE5:C8       377       INY       CEE6:68       378       PLA         CEE7:20       10       379       AND       #\$10       ;         CEE8:90       38       04       382       STA       PARAMETER, X         CEF1:00       383       RTS       CEF2:       384 *       CEF2:       384 *         CEF2:39       CB       376       PHA       CEF3:48       386       PHA         CEF2:49       CD       385       CMDEXEC1       LDA       #SCD       ;ROUTINES MUST BE IN PAGE SCD         CEF4:48       386       PHA       CEF3:89       EB       CC       387       LDA         CEF9:A0       J0       LDA       CMDTBL, Y       CEF6:80       391       RTS         CEF9:10       393       DFB       \$00			12 Die cinconine		
CEDF:48       374       PHA         CEED:29       07       375       AND       #\$07         CEE2:20       7B CE       376       JSR       SETOSTATE ;SET NEXT STATE         CEE5:20       7B CE       376       JSR       SETOSTATE ;SET NEXT STATE         CEE5:20       7B CE       376       PLA         CEE7:29       10       379       AND       #\$10       ;         CEE9:20       07       380       BNE       CMDEXEC1 ; IF BIT 4 IS SET, VECTOR TO ROUTINE         CEE8:39       EB CC       381       LDA       CMDTBL, Y         CEE1:60       383       RTS         CEF2:39       284 *       .         CEF2:48       386       PHA         CEF9:189       EB CC       387       LDA         CEF9:180       38 04       390       LDA       #SCD ;ROUTINES MUST BE IN PAGE \$CD         CEF9:143       388       PHA       .       .       .         CEF9:142       380       JDA       CMDTBL, Y       .       .         CEF9:142       380       JDA       PARAMETER, X ;LOT OF ROUTINES NEED THIS       .         CEF9:142       380       JDA       PARAMETER, X ;LOT OF ROUTINES N			TDA ZOTMD1	.RETRIEVE TABLE M	ODE BYTE
CEE0:29 07       375       AND #\$07         CEE2:20 7B CE       376       JSR       SETOSTATE ; SET NEXT STATE         CEE2:20 7B CE       377       INY         CEE6:68       377       INY         CEE7:29 10       379       AND #\$10       ;         CEE7:29 10       379       AND #\$10       ;         CEE7:29 10       379       AND #\$10       ;         CEE8:89 EB CC       381       LDA       CMDTBL, Y         CEE9:00 38 04 382       STA PARAMETER,X       CEE1:60       383         CEF2:       384 *       CEF2:384       SEC         CEF2:48       386       PHA       CEF5:89       EB CC 387       LDA CMDTBL,Y         CEF2:48       386       PHA       CEF5:89       EB CC 387       LDA CMDTBL,Y         CEF8:48       388       PHA       CEF5:89       EB CC 387       LDA CMDTBL,Y         CEF8:48       388       PHA       CEF5:9330       JD PB \$L00'       FOT ROUTINES NEED THIS         CEF5:89       EB CC 391       RTS       CEF5:60       391       RTS         CEFF:00       393       DFB \$00       SUMBOL       CAL       CACK         CSYMBOL TABLE       SORTED BY SYMBOL <td>A REAL PROPERTY OF A REAL PROPER</td> <td></td> <td></td> <td>, and the second second</td> <td></td>	A REAL PROPERTY OF A REAL PROPER			, and the second second	
CEE2:20 78 CE 376 JSR SETOSTATE ;SET NEXT STATE CEE5:C8 377 INY CEE6:68 378 PLA CEE7:29 10 379 AND #\$10 ; CEE9:D0 07 380 BNE CMDEXEC1 ;IF BIT 4 IS SET, VECTOR TO ROUTINE CEE9:D38 04 382 STA PARAMETER,X CEE1:60 383 RTS CEE2: 384 * CEF2: 392 * CEF2: 392 * CEFF: 392 * CEFF: 392 * CEFF: 392 * CEFF: 393 DFB \$00 SYMBOL TABLE SORTED BY SYMBOL 3C A1L CE6A ACCLOOP CC02 ACIAOUT ?CB9C ACK C9C8 ADJRTS C9F5 ADJUST CE59 ANRTS CCB7 ASCREEN CB88 BASICEXIT 28 BASL ?C93D BATCHIN C9EF BARCHIO C304 BATCHOUT CD7D BAUCMD CD81 BAUDCMD1 CD88 BAUDCMD2 C711 BENTRY C8EF BINACIA1 C8EA BINACIA C8E5 BINEND1 C800 BINEND C745 BINIT1 ?C700 BINIT C8C6 BINKBD C88F BINPUT C77C BOUTPUT1 C767 BOUTPUT C788 BOUTPUT2 CDC1 BREAKCMD 0688 BUFPYTE CE8A CDONE 24 CH 27 CHARACTER CA1E CHECKTERM 0638 CHNBYTE C885 CICRXIT					
CEE5:C8 377 INY CEE6:68 378 PLA CEE7:29 10 379 AND #\$10 ; CEE9:D0 07 380 BNE CMDEXEC1 ;IF BIT 4 IS SET, VECTOR TO ROUTINE CEE8:B9 EB CC 381 LDA CMDTBL, Y CEEE:9D 38 04 382 STA PARAMETER, X CEF1:60 383 RTS CEF2: 384 * CEF2:A9 CD 385 CMDEXEC1 LDA #\$CD ;ROUTINES MUST BE IN PAGE \$CD CEF4:48 386 PHA CEF5:B9 EB CC 387 LDA CMDTBL, Y CEF8:48 386 PHA CEF9:A4 26 389 LDY SLOT16 CEF8:E0 391 RTS CEF7: 392 * CEFF: 392 * CEFF: 392 * CEFF: 392 * CEFF: 392 * CEFF: 393 DFB \$00 SYMBOL TABLE SORTED BY SYMBOL 3C A1L CE6A ACCLOOP CC02 ACIAOUT ?CB9C ACK C968 ADJRTS C9B5 ADJUST CB59 ANRTS CCB7 ASCREEN CB88 BASICEXIT 28 BASL ?C30 BATCHIN C9EF BATCHIO C264 BASICEXIT 28 BASL ?C30 BATCHIN C9EF BATCHIO ?C411 BATCHOUT CD7D BAUDCMD CD81 BAUDCMD1 CD88 BAUDCMD2 C711 BENTRY C3EF BINACIA1 C3EA BINACIA C4E5 BINED1 C300 BINEND C745 BINIT1 ?C700 BINIT C38C BINKED C86F BINPUT C77C BOUTPUT1 C767 BOUTPUT C78B BOUTPUT2 CDC1 BREAKCMD 06B8 BUFBTFE CE8A CDONE 24 CH 27 CHARACTER CAIE CHECKTERM 0638 CHNBYTE CB55 CICEXIT				SET NEXT STATE	
CEE6:68       378       PIA         CEE6:68       378       PIA         CEE7:29       10       379       AND       #\$10       ;         CEE9:D0       07       380       BNE       CMDEXEC1       ; IF       BIT 4 IS SET, VECTOR TO ROUTINE         CEEB:B9       EB CC       381       LDA       CMDTBL, Y       CEER:P0       SET       AND         CEEP:160       383       RTS       CEF2:       384 *       CEF2:       384 *         CEF2:3       384       *       CEF2:3       384 *       CEF5:B9       EB CC       387       LDA       #\$CD       ;ROUTINES MUST BE IN PAGE \$CD         CEF4:48       386       PHA       CEF5:B9       EB CC       387       LDA       CMDTBL, Y         CEF8:48       388       PHA       CEF7:       392       SIOTED BY SLOT16       CEFF:60       391       RTS         CEFF:0       392       *       CEFF:00       393       DFB \$00       SYMBOL         SC A1L       CE6A ACCLOOP       CC02 ACIAOUT       ?CB9C ACK       CCB7 ASCREEN         C988 BASICEXIT       28 BAS       C993D BATCHIN       C9EF BATCHIO       CD8 BAUCMD2         C911 BENTRY       C38EF BINACIA1 </td <td>and the second s</td> <td></td> <td></td> <td></td> <td></td>	and the second s				
CEE7:29 10 379 AND #\$10 ; CEE7:29 10 379 AND #\$10 ; CEE9:D0 07 380 BNE CMDEXEC1 ;IF BIT 4 IS SET, VECTOR TO ROUTINE CEEB:B9 EB CC 381 LDA CMDTBL,Y CEEE:90 38 04 382 STA PARAMETER,X CEF1:60 383 RTS CEF2: 384 * CEF2:A9 CD 385 CMDEXEC1 LDA #\$CD ;ROUTINES MUST BE IN PAGE \$CD CEF4:48 386 PHA CEF5:B9 EB CC 387 LDA CMDTBL,Y CEF8:48 388 PHA CEF9:A4 26 389 LDY SLOT16 CEF8:48 388 PHA CEF9:A4 26 389 LDY SLOT16 CEFF: 392 * CEFF: 392 * CEFF: 392 * CEFF: 392 * CEFF: 392 * CEFF: 392 * CEFF: 28 ALL CE6A ACCLOOP CC02 ACIAOUT ?CB9C ACK C9C8 ADJRTS C9B5 ADJUST CB59 ANRTS CCB7 ASCREEN C888 BASICEXIT 28 BASL ?C93D BATCHIN C9EF BATCHIO C9C94 BATCHOUT CD7D BAUDCMD CD81 BAUDCMD1 CD88 BAUDCMD2 C711 BENTRY C8EF BINACIA1 C8EA BINACIA C8E5 BINEND1 C8D0 BINEND C745 BINT1 ?C70C BOUTPUT C78B BOUTPUT2 CDC1 BREAKCMD 06B8 BUFBYTE CE8A CDONE 24 CH 27 CHARACTER CAIE CHECKTERM 0638 CHNBYTE C8B5 CICEXIT					
CEE9:D0 07380BNECMDEXEC1; IFBIT 4 IS SET, VECTOR TO ROUTINECEEB:B9EB CC381LDACMDTBL, YCEEE:9D38 04382STAPARAMETER, XCEF1:60383RTSCEF2:384 *CEF2:384 *CEF2:385CMDEXEC1 LDA #SCD;ROUTINES MUST BE IN PAGE \$CDCEF4:48386PHACEF5:B9EB CC387LDACMDTBL, YCEF8:48388PHACEF9:A426389LDYSLOT16CEF8:48380JDAPARAMETER, X ;LOT OF ROUTINES NEED THISCEFF:60391RTSCEFF:00393DFB \$00SYMBOL TABLESORTED BY SYMBOL3CA1LCE6A ACCLOOPCC02 ACIAOUT?CB9C ACKC9C8ADJRTSC9B5 ADJUSTCB59 ANRTSCCB7 ASCREENC888BASICEXIT28BASL?C93D BATCHINC9EF BATCHIO?C941BATCHOUTCD7D BAUDCMDCD81 BAUDCMD1CD88 BAUDCMD2C711BENTRYC8EF BINACIA1C8EA BINACIAC8E5 BINEND1C8D0BINENDC745 BINT1?C700 BINITC8C8 BINKBDC8BF BINPUTC77C BOUTPUT1C767 BOUTPUTC78B BOUTPUT2CDC1BREAKCMD06B8 BUFBYTECE8A CDONE24 CH27CHARACTERCA1E CHECKTERM0638 CHNBYTEC8B5 CICEXIT	CEE7: 29 10		AND #\$10		
CEEB:B9 EB CC 381 LDA CMDTBL,Y CEEE:90 38 04 382 STA PARAMETER,X CEF1:60 383 RTS CEF2: 384 * CEF2: 384 * CEF2:A9 CD 385 CMDEXEC1 LDA #\$CD ;ROUTINES MUST BE IN PAGE \$CD CEF4:48 386 PHA CEF5:B9 EB CC 387 LDA CMDTBL,Y CEF8:48 388 PHA CEF9:A4 26 389 LDY SLOT16 CEF8:E0 38 04 390 LDA PARAMETER,X ;LOT OF ROUTINES NEED THIS CEFF: B0 38 04 390 LDA PARAMETER,X ;LOT OF ROUTINES NEED THIS CEFF: 392 * CEFF: 392 * CEFF: 392 * CEFF: 00 393 DFB \$00 SYMBOL TABLE SORTED BY SYMBOL 3C A1L CE6A ACCLOOP CC02 ACIAOUT ?CB9C ACK C9C8 ADJRTS C9B5 ADJUST CB59 ANRTS CCB7 ASCREEN C888 BASICEXIT 28 BASL ?C93D BATCHIN C9EF BATCHIO ?C941 BATCHOUT CD7D BAUDCMD CD81 BAUDCMD1 CD88 BAUDCMD2 C711 BENTRY C8EF BINACIA1 C8EA BINACIA C8E5 BINEND1 C8D0 BINEND C745 BINTT1 ?C700 BINIT C8CB BINKBD C8BF BINPUT C77C BOUTPUT1 C767 BOUTPUT C78B BOUTPUT2 CDC1 BREAKCMD 06B8 BUFBYTE CE8A CDONE 24 CH 27 CHARACTER CA1E CHECKTERM 0638 CHNBYTE C8B5 CICEXIT		202.5			VECTOR TO ROUTINE
CEEE:9D 38 04 382 STA PARAMETER,X CEF1:60 383 RTS CEF2: 384 * CEF2: 384 * CEF2:A9 CD 385 CMDEXEC1 LDA #SCD ;ROUTINES MUST BE IN PAGE \$CD CEF4:48 386 PHA CEF5:B9 EB CC 387 LDA CMDTBL,Y CEF8:48 388 PHA CEF9:A4 26 389 LDY SLOT16 CEFB:B0 38 04 390 LDA PARAMETER,X ;LOT OF ROUTINES NEED THIS CEFF: 392 * CEFF: 392 * CEFF: 392 * CEFF:00 393 DFB \$00 SYMBOL TABLE SORTED BY SYMBOL 3C A1L CE6A ACCLOOP CC02 ACIAOUT ?CB9C ACK C9C8 ADJRTS C9B5 ADJUST CB59 ANRTS CCB7 ASCREEN C888 BASICEXIT 28 BASL ?C93D BATCHIN C9EF BATCHIO 2C911 BENTRY C8EF BINACIA1 C8EA BINACIA C9E5 BINEND1 C8D0 BINEND C745 BINIT1 ?C700 BINIT C8C8 BINKBD C8BF BINPUT C77C BOUTPUT1 C767 BOUTPUT C78B BOUTPUT2 CDC1 BREAKCMD 06B8 BUFBYTE CE8A CDONE 24 CH 27 CHARACTER CAIE CHECKTERM 0638 CHNBYTE C8B5 CICEXIT					
CEF1:60 383 RTS CEF2: 384 * CEF2: 384 * CEF2: 384 * CEF2: A9 CD 385 CMDEXEC1 LDA #SCD ;ROUTINES MUST BE IN PAGE \$CD CEF4:48 386 PHA CEF5:B9 EB CC 387 LDA CMDTBL,Y CEF8:48 388 PHA CEF9:A4 26 389 LDY SLOT16 CEF8:BD 38 04 390 LDA PARAMETER,X ;LOT OF ROUTINES NEED THIS CEF7: 392 * CEFF: 392 * CEFF: 392 * CEFF: 00 393 DFB \$00 SYMBOL TABLE SORTED BY SYMBOL 3C A1L CE6A ACCLOOP CC02 ACIAOUT ?CB9C ACK C9C8 ADJRTS C9B5 ADJUST CB59 ANRTS CCB7 ASCREEN C8B8 BASICEXIT 28 BASL ?C93D BATCHIN C9EF BATCHIO ?C941 BATCHOUT CD7D BAUDCMD CD81 BAUDCMD1 CD88 BAUDCMD2 C711 BENTRY C8EF BINACIA1 C8EA BINACIA C8E5 BINEND1 C8D0 BINEND C745 BINIT1 ?C700 BINIT C858 BINET C8BF BINPUT C77C BOUTPUT1 C767 BOUTPUT C0C1 BREAKCMD 0688 BUFBYTE CE8A CDONE 24 CH 27 CHARACTER CAIE CHECKTERM 0638 CHNBYTE C8B5 CICEXIT					
CEP2: 384 * CEP2: 384 * CEP2: A9 CD 385 CMDEXEC1 LDA #\$CD ;ROUTINES MUST BE IN PAGE \$CD CEP4:48 386 PHA CEF5: B9 EB CC 387 LDA CMDTBL,Y CEF8:48 388 PHA CEF9: A4 26 389 LDY SLOT16 CEF8: BD 38 04 390 LDA PARAMETER,X ;LOT OF ROUTINES NEED THIS CEFE: 60 391 RTS CEFF: 392 * CEFF: 392 * CEFF: 00 393 DFB \$00 SYMBOL TABLE SORTED BY SYMBOL 3C A1L CE6A ACCLOOP CC02 ACIAOUT ?CB9C ACK C9C8 ADJRTS C9B5 ADJUST CB59 ANRTS CCB7 ASCREEN C888 BASICEXIT 28 BASL ?C93D BATCHIN C9EF BATCHIO ?C941 BATCHOUT CD7D BAUDCMD CD81 BAUDCMD1 CD88 BAUDCMD2 C711 BENTRY C8EF BINACIA1 C8EA BINACIA C8E5 BINEND1 C800 BINEND C745 BINT1 ?C700 BINIT C86B BINEND C86F BINPUT C77C BOUTPUT1 C767 BOUTPUT C78B BOUTPUT2 CDC1 BREAKCMD 0688 BUFBYTE CE8A CDONE 24 CH 27 CHARACTER CAIE CHECKTERM 0638 CHNBYTE C8B5 CICEXIT				and the second se	
CEF2:A9 CD 385 CMDEXEC1 LDA #SCD ;ROUTINES MUST BE IN PAGE \$CD CEF4:48 386 PHA CEF5:B9 EB CC 387 LDA CMDTBL,Y CEF8:48 388 PHA CEF9:A4 26 389 LDY SLOT16 CEF8:48 390 LDA PARAMETER,X ;LOT OF ROUTINES NEED THIS CEF8:60 391 RTS CEFF: 392 * CEFF: 392 * CEFF: 392 * CEFF:00 393 DFB \$00 SYMBOL TABLE SORTED BY SYMBOL 3C A1L CE6A ACCLOOP CC02 ACIAOUT ?CB9C ACK C9C8 ADJRTS C9B5 ADJUST CB59 ANRTS CCB7 ASCREEN C888 BASICEXIT 28 BASL ?C93D BATCHIN C9EF BATCHIO ?C941 BATCHOUT CD7D BAUDCMD CD81 BAUDCMD1 CD88 BAUDCMD2 C711 BENTRY C8EF BINACIA1 C8EA BINACIA C8E5 BINEND1 C800 BINEND C745 BINIT1 ?C700 BINIT C78B BOUTPUT2 CDC1 BREAKCMD 06B8 BUFBYTE CE8A CDONE 24 CH 27 CHARACTER CAIE CHECKTERM 0638 CHNBYTE C8B5 CICEXIT					
CEP4:48 386 PHA CEF5:B9 EB CC 387 LDA CMDTBL,Y CEF8:48 388 PHA CEF9:A4 26 389 LDY SLOT16 CEF8:60 391 RTS CEFF:60 391 RTS CEFF: 392 * CEFF:00 393 DFB \$00 SYMBOL TABLE SORTED BY SYMBOL 3C A1L CE6A ACCLOOP CC02 ACIAOUT ?CB9C ACK C9C8 ADJRTS C9B5 ADJUST CB59 ANRTS CCB7 ASCREEN C8B8 BASICEXIT 28 BASL ?C93D BATCHIN C9EF BATCHIO ?C941 BATCHOUT CD7D BAUDCMD CD81 BAUDCMD1 CD88 BAUDCMD2 C711 BENTRY C8EF BINACIA1 C8EA BINACIA C8E5 BINEND1 C8D0 BINEND C745 BINTT1 ?C700 BINIT C8CB BINKBD C8BF BINPUT C77C BOUTPUT1 C767 BOUTPUT C78B BOUTPUT2 CDC1 BREAKCMD 06B8 BUFBYTE CE8A CDONE 24 CH 27 CHARACTER CA1E CHECKTERM 0638 CHNBYTE C8B5 CICEXIT	and the second		LDA #SCD	ROUTINES MUST BE	IN PAGE SCD
CEF5:B9 EB CC 387 LDA CMDTBL,Y CEF8:48 388 PHA CEF9:A4 26 389 LDY SLOT16 CEF9:A4 26 389 LDY SLOT16 CEF8:ED 38 04 390 LDA PARAMETER,X ;LOT OF ROUTINES NEED THIS CEFF:60 391 RTS CEFF: 392 * CEFF:00 393 DFB \$00 SYMBOL TABLE SORTED BY SYMBOL 3C A1L CE6A ACCLOOP CC02 ACIAOUT ?CB9C ACK C9C8 ADJRTS C9B5 ADJUST CB59 ANRTS CCB7 ASCREEN C888 BASICEXIT 28 BASL ?C93D BATCHIN C9EF BATCHIO ?C911 BENTRY CBEF BINACIA1 CBEA BINACIA C9E5 BINEND1 C711 BENTRY C8EF BINACIA1 CBEA BINACIA C8E5 BINEND1 C800 BINEND C745 BINT1 ?C700 BINIT C8C8 BINKBD C8BF BINPUT C77C BOUTPUT1 C767 BOUTPUT C78B BOUTPUT2 CDC1 BREAKCMD 06B8 BUFBYTE CE8A CDONE 24 CH 27 CHARACTER CAIE CHECKTERM 0638 CHNBYTE C8B5 CICEXIT					
CEF8:48388PHACEF9:A426389LDYSLOT16CEF8:BD3804390LDAPARAMETER, X ;LOT OF ROUTINES NEED THISCEF8:60391RTSCEFF:392 *CEFF:00393DFB \$00SYMBOLTABLESORTED BY SYMBOL3CA1LCE6AACCLOOPCC02ACIAOUTC9C8ADJRTSC9B5C8B8BASICEXIT28BASL?C93DPATCHOUTCD7DBAUDCMDCD81BAUCCMDCD81BANCIAC8EFBINENDC745C8BFBINPUTC77CBOUTPUTIC68FBINPUTC77CBOUTPUTIC77CBAUSTEC88FCB85C11BREAKCMDC685CONE24CH27CHARACTERCA1ECHECKTERM0638CHNBYTEC885CICEXIT					
CEF9:A4 26389LDYSLOT16CEFB:BD 38 04390LDAPARAMETER, X ; LOT OF ROUTINES NEED THISCEFE:60391RTSCEFF:392 *CEFF:00393DFB \$00SYMBOL TABLESORTED BY SYMBOL3C A1LCE6A ACCLOOPCC02 ACIAOUTC9C8 ADJRTSC9B5 ADJUSTCB59 ANRTSC8B8 BASICEXIT28 BASL?C93D BATCHIN2C941 BATCHOUTCD7D BAUDCMDCD81 BAUDCMD1CB00 BINENDC745 BINIT1?C70C BOUTPUTC8D8 BINENDC745 BINIT1?C70C BOUTPUTC8D6 BINENDC745 BINIT1?C70C BOUTPUTC8D7 BINPUTC77C BOUTPUT1C767 BOUTPUTC0C1 BREAKCMD0688 BUFBYTECE8A CDONE24 CH27 CHARACTERCA1E CHECKTERM0638 CHNBYTEC8B5 CICEXIT			and the second		
CEFB:BD 38 04 390       LDA PARAMETER,X ;LOT OF ROUTINES NEED THIS         CEFE:60       391       RTS         CEFF:       392 *         CEFF:00       393       DFB \$00         SYMBOL TABLE       SORTED BY SYMBOL         3C A1L       CE6A ACCLOOP       CC02 ACIAOUT       ?CB9C ACK         C9C8 ADJRTS       C9B5 ADJUST       CB59 ANRTS       CCB7 ASCREEN         C8B8 BASICEXIT       28 BASL       ?C93D BATCHIN       C9EF BATCHIO         ?C911 BATCHOUT       CD7D BAUDCMD       CD81 BAUDCMD1       CD88 BAUDCMD2         C711 BENTRY       C8EF BINACIA1       C8EA BINACIA       C8E5 BINEND1         C8D0 BINEND       C745 BINIT1       ?C700 BINIT       C8CB BINKBD         C8BF BINPUT       C77C BOUTPUT1       C767 BOUTPUT       C78B BOUTPUT2         CDC1 BREAKCMD       0688 UFBYTE       CE8A CDONE       24 CH         27 CHARACTER       CA1E CHECKTERM       0638 CHNBYTE       C8B5 CICEXIT					
CEFE:60       391       RTS         CEFF:       392 *         CEFF:00       393       DFB \$00         SYMBOL TABLE       SORTED BY SYMBOL         3C A1L       CE6A ACCLOOP       CC02 ACIAOUT       ?CB9C ACK         C9C8 ADJRTS       C9B5 ADJUST       CB59 ANRTS       CCB7 ASCREEN         C8B8 BASICEXIT       28 BASL       ?C93D BATCHIN       C9EF BATCHIO         ?C911 BATCHOUT       CD7D BAUDCMD       CD81 BAUDCMD1       CD88 BAUDCMD2         C711 BENTRY       C8EF BINACIA1       C8EA BINACIA       C8E5 BINEND1         C8D0 BINEND       C745 BINIT1       ?C700 BINIT       C8CB BINKBD         C8BF BINPUT       C77C BOUTPUT1       C767 BOUTPUT       C78B BOUTPUT2         CDC1 BREAKCMD       0688 UFBYTE       CE8A CDONE       24 CH         27 CHARACTER       CA1E CHECKTERM       0638 CHNBYTE       C8B5 CICEXIT				R.X :LOT OF ROUTINE	S NEED THIS
CEFF: 392 * CEFF:00 393 DFB \$00 SYMBOL TABLE SORTED BY SYMBOL 3C A1L CE6A ACCLOOP CC02 ACIAOUT ?CB9C ACK C9C8 ADJRTS C9B5 ADJUST CB59 ANRTS CCB7 ASCREEN C8B8 BASICEXIT 28 BASL ?C93D BATCHIN C9EF BATCHIO ?C941 BATCHOUT CD7D BAUDCMD CD81 BAUDCMD1 CD88 BAUDCMD2 C711 BENTRY C8EF BINACIA1 C8EA BINACIA C8E5 BINEND1 C8D0 BINEND C745 BINIT1 ?C700 BINIT C8CB BINKBD C8BF BINPUT C77C BOUTPUT1 C76F BOUTPUT C78B BOUTPUT2 CDC1 BREAKCMD 06B8 BUFBYTE CE8A CDONE 24 CH 27 CHARACTER CA1E CHECKTERM 0638 CHNBYTE C8B5 CICEXIT					
CEFF:00393DFB \$00SYMBOL TABLESORTED BY SYMBOL3C A1LCE6A ACCLOOPCC02 ACIAOUT?CB9C ACKC9C8 ADJRTSC9B5 ADJUSTCB59 ANRTSCCB7 ASCREENC8B8 BASICEXIT28 BASL?C93D BATCHINC9EF BATCHIO?C941 BATCHOUTCD7D BAUDCMDCD81 BAUDCMD1CD88 BAUDCMD2C711 BENTRYC8EF BINACIA1C8EA BINACIAC8E5 BINEND1C8D0 BINENDC745 BINIT1?C700 BINITC8CB BINKBDC8BF BINPUTC77C BOUTPUT1C767 BOUTPUTC78B BOUTPUT2CDC1 BREAKCMD06B8 BUFBYTECE8A CDONE24 CH27 CHARACTERCA1E CHECKTERM0638 CHNBYTEC8B5 CICEXIT		The second s	N10		
SYMBOL TABLE       SORTED BY SYMBOL         3C A1L       CE6A ACCLOOP       CC02 ACIAOUT       ?CB9C ACK         C9C8 ADJRTS       C9B5 ADJUST       CB59 ANRTS       CCB7 ASCREEN         C888 BASICEXIT       28 BASL       ?C93D BATCHIN       C9EF BATCHIO         ?C941 BATCHOUT       CD7D BAUDCMD       CD81 BAUDCMD1       CD88 BAUDCMD2         C711 BENTRY       C8EF BINACIA1       C8EA BINACIA       C8E5 BINEND1         C8D0 BINEND       C745 BINIT1       ?C700 BINIT       C8CB BINKBD         C8BF BINPUT       C77C BOUTPUT1       C767 BOUTPUT       C78B BOUTPUT2         CDC1 BREAKCMD       06B8 BUFBYTE       CE8A CDONE       24 CH         27 CHARACTER       CA1E CHECKTERM       0638 CHNBYTE       C8B5 CICEXIT			DFB \$00		
3C A1LCE6A ACCLOOPCC02 ACIAOUT?CB9C ACKC9C8 ADJRTSC9B5 ADJUSTCB59 ANRTSCCB7 ASCREENC8B8 BASICEXIT28 BASL?C93D BATCHINC9EF BATCHIO?C911 BATCHOUTCD7D BAUDCMDCD81 BAUDCMD1CD88 BAUDCMD2C711 BENTRYC8EF BINACIA1C8EA BINACIAC8E5 BINEND1C8D0 BINENDC745 BINIT1?C700 BINITC8CB BINKBDC8BF BINPUTC77C BOUTPUT1C767 BOUTPUTC78B BOUTPUT2CDC1 BREAKCMD06B8 BUFBYTECE8A CDONE24 CH27 CHARACTERCA1E CHECKTERM0638 CHNBYTEC8B5 CICEXIT	CDFF:00	555			
SC ALLCDBCACCBCACCBCASC CCB7ASCREENC9C8 ADJRTSC9B5 ADJUSTCB59 ANRTSCCB7ASCREENC8B8 BASICEXIT28 BASL?C93D BATCHINC9EF BATCHIO?C941 BATCHOUTCD7D BAUDCMDCD81 BAUDCMD1CD88 BAUDCMD2C711 BENTRYC8EF BINACIA1C8EA BINACIAC8E5 BINEND1C8D0 BINENDC745 BINIT1?C700 BINITC8CB BINKBDC8BF BINPUTC77C BOUTPUT1C767 BOUTPUT2C78B BOUTPUT2CDC1 BREAKCMD06B8 BUFBYTECE8A CDONE24 CH27 CHARACTERCA1E CHECKTERM0638 CHNBYTEC8B5 CICEXIT	SYMBOL TABLE	SORTED BY	SYMBOL		
SC ALLCDBCACDBCACBCAC9C8 ADJRTSC9B5 ADJUSTCB59 ANRTSCCB7 ASCREENC8B8 BASICEXIT28 BASL?C93D BATCHINC9EF BATCHIO?C941 BATCHOUTCD7D BAUDCMDCD81 BAUDCMD1CD88 BAUDCMD2C711 BENTRYC8EF BINACIA1C8EA BINACIAC8E5 BINEND1C8D0 BINENDC745 BINIT1?C700 BINITC8CB BINKBDC8BF BINPUTC77C BOUTPUT1C767 BOUTPUTC78B BOUTPUT2CDC1 BREAKCMD06B8 BUFBYTECEBA CDONE24 CH27 CHARACTERCA1E CHECKTERM0638 CHNBYTEC8BF CICEXIT				and the second second	
C9C3 ADJRTSC9D1C9D1C9D1C9D1C8B8 BASICEXIT28 BASI?C93D BATCHINC9EF BATCHIO?C941 BATCHOUTCD7D BAUDCMDCD81 BAUDCMD1CD88 BAUDCMD2C711 BENTRYC8EF BINACIA1C8EA BINACIAC8E5 BINEND1C8D0 BINENDC745 BINIT1?C700 BINITC8EB BINEDDC8BF BINPUTC77C BOUTPUT1C767 BOUTPUTC78B BOUTPUT2CDC1 BREAKCMD06B8 BUFBYTECE8A CDONE24 CH27 CHARACTERCA1E CHECKTERM0638 CHNBYTEC8B5 CICEXIT	3C A1L				
C898 BASICEATI     CD7D BAUDCMD     CD81 BAUDCMD1     CD88 BAUDCMD2       ?C941 BATCHOUT     CD7D BAUDCMD     CD81 BAUDCMD1     CD88 BAUDCMD2       C711 BENTRY     C8EF BINACIA1     C8EA BINACIA     C8E5 BINEND1       C8D0 BINEND     C745 BINIT1     ?C700 BINIT     C8CB BINKBD       C8BF BINPUT     C77C BOUTPUT1     C767 BOUTPUT     C78B BOUTPUT2       CDC1 BREAKCMD     0688 UFBYTE     CEAR CDONE     24 CH       27 CHARACTER     CA1E CHECKTERM     0638 CHNBYTE     C885 CICEXIT	C9C8 ADJRTS				
C711BENTRYC8EFBINACIA1C8EABINACIAC8E5BINEND1C800BINENDC745BINT11?C700BINITC8CBBINKBDC8BFBINPUTC77CBOUTPUT1C767BOUTPUTC78BBOUTPUT2CDC1BREAKCMD06B8BUFBYTECE8ACDONE24CH27CHARACTERCA1ECHECKTERM0638CHNBYTEC885CICEXIT					
CALLBENKRIC745BINITI?C700BINITC8CBBINKBDC8DFBINPUTC77CBOUTPUT1C767BOUTPUTC78BBOUTPUT2CDC1BREAKCMD06B8BUFBYTECE8ACDONE24CH27CHARACTERCA1ECHECKTERM0638CHNBYTEC885CICEXIT	?C941 BATCHOU				
C800     BINEND     C77C     BOUTPUT1     C767     BOUTPUT2       C8BF     BINPUT     C77C     BOUTPUT1     C767     BOUTPUT2       CDC1     BREAKCMD     0688     BUFBYTE     CE8A     CDONE     24     CH       27     CHARACTER     CA1E     CHECKTERM     0638     CHNBYTE     C885     CICEXIT	C711 BENTRY				
CDC1 BREAKCMD     06B8 BUFBYTE     CE8A CDONE     24 CH       27 CHARACTER     CA1E CHECKTERM     0638 CHNBYTE     C8B5 CICEXIT	C8D0 BINEND		Panaa .		
27 CHARACTER CA1E CHECKTERM 0638 CHNBYTE C885 CICEXIT	C8BF BINPUT				
27 CHARACTER CHIEFE CHIEFE					
					C9EB CKINPUT2
C9EE CIEND C9D1 CKINPUT C9E5 CKINPUT1 C9EB CKINPUT2	C9EE CIEND	C9D1	CKINPUT	C9E5 CKINPUT1	CAUD CVIMPOIS

CC3D	CKKBDXIT	cc	:2C	CKKBD
CEF2	CMDEXEC1		DD	CMDEXEC
CEDB	CMDMATCH2		AOS	CMDPROC1
CE24	CMDPROC4	CD	FC	CMDPROC
CEB9	CMDSEARCH1	CE	31A	CMDSEQCK
06B8	COLBYTE		17	COMMA
CE43			54	CSTATE1B
CE75	CSTATE2A	CE	62	CSTATE2
CE8B	CSTATE4		37	CSWH
CD8E	CTLREGSET	CS	009	CTRLTST
C9C0	DECRCOL	03	B8	DELAYFLG
C082	DIPSW2	CE	BFF	DLYTBL
CD47	FFCMD	CS	68	FORCECR
C8B4	GETCHAR1	CE	AA	GETCHAR
CC44	GETKBD1		:91	GETKBDONE
CDD4	ICMD	C7	705	IENTRY
C827	INIT1A	CE	335	INIT2
C857	INIT3		864	INIT4
C882	INITACIA1		88F	INITACIA2
FF58	IORTS			KBDSTRB
CDE2	KCMD1		)E5	KCMD2
38	KSWL		CD3	LCMASK
0738	MISCFLG		10C	MOVIN
CDE9	NCMD		358	NOCMD
CCB6	NOOUT		75C	NORMIO
FCBA	NXTA1		07	OENTRY
CB68	OUTPUT1		36B	OUTPUT2
CBFE CBBB	OUTPUTEND P8AOUT4		138	P8AOUT1 PARAMETER
2C89E	PASCALREAD		138 198	PARAMETER
C99B	PENTRY		78E	PINIT
CC93	PROMPTBL		SE	PROMPTLOOP
C7AB	PSTATUS 2			PWDBYTE
CDBD	OCMD1		)B9	OUITCMD
CDAD	RESETCMD		211	RESTORE
CCDB	REVMASK		4F	RNDH
CD57	ROTATE	CT	7B2	SAVEHOOK
С7ВО	SENDCD	CS	998	SEREND2
C996	SETCH	FI	89	SETKBD
CEA4	SETSTATE6		26	SLOT16
CAD2	SRIN	CF	AF4	SRIN3
CD61	SSLOTCMD		00	STACK
CE30	STATETBL		989	
C948	TAB2		921	TABCHECK
CAAD	TERMCAP1		19B	TERMCAP
CA82 CAB1	TERMINC1		181	TERMINC
CA41	TERMLOCK		123	TERMMODE
CA93	TERMNEXT2 TERMSEND		447	TERMNEXT3 TERMSEND1
CD35	TRANCMD			
CAC4	WAITMS		AB9	TRANSLATE
CDFB	ZCMDRTS		AC6 DF4	WAITMS1 ZCMD
2A	ZPTMP1	CL	2B	ZPTMP2
			20	DE THE 2
SYMBO	L TABLE	SORTED	BY	ADDRESS
			-	
24	СН		26	SLOT16
2A	ZPTMP1		2B	
37	CSWH			KSWL
4E	RNDL			RNDH
0388	DELAYFLG	04	138	HANDSHKE
				ALCON DE LE COMPANY

1 

Se. . 

-

51.

-

1 

-----1 100

ALC: N

1

and a

1.11

.

1

1.20 1

-

	CC34	CKKBD1	0538	CMDBYTE
	CECE	CMDMATCH1	CEBD	CMDMATCH
	CEOE	CMDPROC2	CE1A	CMDPROC 3
	C08A	CMDREG	CEAF	CMDSEARCH
	CD19	CMDTBL1	CCEB	CMDTBL
	FDED	COUT	CD3B	CRCMD
	CE5E	CSTATE1C	CE38	CSTATE1
	CE9C	CSTATE4A	CEA1	CSTATE4B
		CSWL		CTLREG
		DATACMD1	CD9C	
		DELAYSET		DIPSW1
		ESCCHECK	CB90	
		FROMIN	C751	FROMOUT
		GETCMD		GETKBD
	CCDF	GETXLATE	0438	
		INBUFF		INIT1
	C83C	INIT2A		INIT2B
		INIT5		INITACIA
		INPUT2		INPUT
	C000			KBDESC
	CDEO		39	KSWH
	CD41		CBEA	LFGEN
	C9FD	MOVOUT	07F8	MSLOT
	CB19	NOINPUT1	CB18	NOINPUT
	C954	NOTAB1	C951	NOTAB
	CBD9	OUTDLY1	CBE2	OUTDLYLP
	CB63	OUTPUT	CBC1	OUTPUT3
		P8AOUT2		P8AOUT3
		PARITYCMD		PASCALINIT
D	C9AA	PASCALWRITE	C8A3	PASEXIT
	?C84D	PREADO	C794	
P	C7A8	PSTATIN		PSTATUS
		PWDTBL	C797	PWRITE
		RDREG		RESET
		RESTOREND	C7EE	RESTORHOOK
		RNDL	CFFF	
		SCREENOUT	CCA3	SCREENOUTI
	C97A			SEROUT
		SETOSTATE	FE93	
		SRIN1		SRIN2
		SROUT		SSLOTCMD1
		STATEFLG		STATERR
		STSBYTE TDREG	C934	
	2C088 CDA7	TERMCMD	CA55	
			?CA4C	
	CA66	TERMKBDIN	CA87	
	CA2B		CA31	
	?CA7D			TERMRTS
	CCC6			TOSCREEN
	CCD7		FDF6	
		XOFFCK	CB5A	XONWAIT
	CE79	ZEROSTATE	35	ZPTEMP
	27	CHARACTER		BASL
	35	ZPTEMP	36	CSWL
	39	KSWH	3C	A1L
	0100	STACK	0200	INBUFF
	0438	PARAMETER	04B8	STATEFLG

0200 INBUFF 04B8 STATEFLG

0538		
	CMDBYTE	
06B8	COLBYTE	
C000	KBD	
?C088	TDREG	
C08A	CMDREG	
C707	OENTRY	
C754	FROMIN	
C78B	BOUTPUT2	
C79A	PSTATUS	
C7B2	SAVEHOOK	
C827	INIT1A	
?C84D	PREADO	
?C879	INITACIA	
?C89E	PASCALREAD1	
C8B5	CICEXIT	
C8D0	BINEND	
C8FC	SEROUT	
?C93D	BATCHIN	
C954	NOTAB1	
C998	SEREND2	
C9B5	ADJUST	
C9D1	CKINPUT	
C9EF	BATCHIO	
?CA23	TERMMODE	
CA47	TERMNEXT3	
CA66	TERMKBDIN	
CA87	TERMLETTER	
CAAD	TERMCAP1	
CAC6	WAITMS1	
CAF4	SRIN3	
CB18	NOINPUT	
CB3C	XOFFCK	
CB63	OUTPUT	
	ETX	
CB90		
CB90 CBBB	P8AOUT4	
CBBB CBEA CC11	P8AOUT4	
CBBB CBEA	P8AOUT4 LFGEN	
CBBB CBEA CC11 CC3D CC5E	P8AOUT4 LFGEN RESTORE CKKBDXIT PROMPTLOOP	
CBBB CBEA CC11 CC3D CC5E CC9E	P8AOUT4 LFGEN RESTORE CKKBDXIT PROMPTLOOP SCREENOUT	
CBBB CBEA CC11 CC3D CC5E CC9E CC23	P8AOUT4 LFGEN RESTORE CKKBDXIT PROMPTLOOP SCREENOUT TOSCREEN	
CBBB CBEA CC11 CC3D CC5E CC9E CC23 CCDB	P8AOUT4 LFGEN RESTORE CKKBDXIT PROMPTLOOP SCREENOUT TOSCREEN REVMASK	
CBBB CBEA CC11 CC3D CC5E CC9E CCC3 CCDB CD35	P8AOUT4 LFGEN RESTORE CKKBDXIT PROMPTLOOP SCREENOUT TOSCREEN REVMASK TRANCMD	
CBBB CBEA CC11 CC3D CC5E CC9E CC23 CCDB	P8AOUT4 LFGEN RESTORE CKKBDXIT PROMPTLOOP SCREENOUT TOSCREEN REVMASK	
CBBB CBEA CC11 CC3D CC5E CC9E CC23 CCDB CD35 CD4B CD7D	P8AOUT4 LFGEN RESTORE CKKBDXIT PROMPTLOOP SCREENOUT TOSCREEN REVMASK TRANCMD DELAYSET BAUDCMD	
CBBB CBEA CC11 CC3D CC5E CC9E CC23 CCDB CD35 CD4B CD7D CD9B	P8AOUT4 LFGEN RESTORE CKKBDXIT PROMPTLOOP SCREENOUT TOSCREEN REVMASK TRANCMD DELAYSET BAUDCMD PARITYCMD	
CBBB CBEA CC11 CC3D CC5E CC9E CC23 CCDB CD35 CD4B CD7D CD9B CDAD	P8AOUT4 LFGEN RESTORE CKKBDXIT PROMPTLOOP SCREENOUT TOSCREEN REVMASK TRANCMD DELAYSET BAUDCMD PARITYCMD RESETCMD	
CBBB CBEA CC11 CC3D CC5E CC9E CC23 CCDB CD35 CD4B CD7D CD9B CDAD	P8AOUT4 LFGEN RESTORE CKKBDXIT PROMPTLOOP SCREENOUT TOSCREEN REVMASK TRANCMD DELAYSET BAUDCMD PARITYCMD RESETCMD ICMD	
CBBB CBEA CC11 CC3D CC5E CC9E CC9E CC3 CCDB CD35 CD4B CD7D CD9B CDAD CD24 CD24	P8AOUT4 LFGEN RESTORE CKKBDXIT PROMPTLOOP SCREENOUT TOSCREEN REVMASK TRANCMD DELAYSET BAUDCMD PARITYCMD RESETCMD ICMD NCMD	
CBBB CBEA CC11 CC3D CC5E CC9E CC2B CCDB CD35 CD4B CD7D CD9B CDAD CDD4 CD24 CD29 CE0A	P8AOUT4 LFGEN RESTORE CKKBDXIT PROMPTLOOP SCREENOUT TOSCREEN REVMASK TRANCMD DELAYSET BAUDCMD PARITYCMD RESETCMD ICMD NCMD CMDPROC1	
CBBB CBEA CC11 CC3D CC5E CC9E CC0B CD35 CD4B CD7D CD4B CD4D CD4A CDE9A CDE0A CD24 CD24 CD24 CD24 CD24 CD24	P8AOUT4 LFGEN RESTORE CKKBDXIT PROMPTLOOP SCREENOUT TOSCREEN REVMASK TRANCMD DELAYSET BAUDCMD PARITYCMD RESETCMD ICMD NCMD CMDPROC1 STATETBL	
CBBB CBEA CC11 CC3D CC5E CC9E CC2B CD35 CD4B CD7D CD4B CD4D CD4A CDE9A CC5E CC93 CD43 CD40 CD42 CD40 CD54 CD54 CD54 CD54 CD54 CD55 CD54 CD55 CD55	P8AOUT4 LFGEN RESTORE CKKBDXIT PROMPTLOOP SCREENOUT TOSCREEN REVMASK TRANCMD DELAYSET BAUDCMD PARITYCMD RESETCMD ICMD NCMD CMDPROC1 STATETBL CSTATE1C	
CBBB CBEA CC11 CC3D CC5E CC9E CC29E CC23 CCDB CD35 CD4B CD4B CD4D CD4A CD59 CD4A CD59 CD4A CD59 CD4A CC52E CC79 CC79E	P8AOUT4 LFGEN RESTORE CKKBDXIT PROMPTLOOP SCREENOUT TOSCREEN REVMASK TRANCMD DELAYSET BAUDCMD PARITYCMD RESETCMD ICMD NCMD CMDPROC1 STATETBL CSTATE1C ZEROSTATE	
CBBB CBEA CC11 CC3D CC5E CC9E CC23 CCDB CD4B CD4B CD7D CD9B CD4D CD9B CD4D CD29 CD42 CD29 CD42 CD29 CD42 CD29 CD42 CD29 CD42 CD29 CD42 CC22 CC22 CC22 CC22 CC22 CC22 CC22	P8AOUT4 LFGEN RESTORE CKKBDXIT PROMPTLOOP SCREENOUT TOSCREEN REVMASK TRANCMD DELAYSET BAUDCMD PARITYCMD RESETCMD ICMD NCMD CMDPROC1 STATETBL CSTATETC ZEROSTATE CSTATE4A	
CBBB CBEA CC11 CC3D CC5E CC9E CC35 CC0B CD35 CD4B CD7D CD9B CD4D CD4 CD4 CD29 CE0A CE30 CE5E CE7C CE7C CE9C CE9C	P8AOUT4 LFGEN RESTORE CKKBDXIT PROMPTLOOP SCREENOUT TOSCREEN REVMASK TRANCMD DELAYSET BAUDCMD PARITYCMD RESETCMD ICMD NCMD CMDPROC1 STATETBL CSTATETC ZEROSTATE CSTATE1C ZEROSTATE CSTATE4A CMDSEARCH	
CBBB CBEA CC11 CC3D CC5E CC9E CC35 CC4B CD4B CD4B CD4B CD4D CD4 CD4 CD4 CD4 CD4 CD4 CD4 CD5 CE4A CE5E CE79 CE9C CE4F CE5B	P8AOUT4 LFGEN RESTORE CKKBDXIT PROMPTLOOP SCREENOUT TOSCREEN REVMASK TRANCMD DELAYSET BAUDCMD PARITYCMD RESETCMD ICMD NCMD CMDPROC1 STATETBL CSTATETC ZEROSTATE CSTATE1C ZEROSTATE CSTATE4A CMDSEARCH CMDMATCH2	
CBBB CBEA CC11 CC3D CC5E CC9E CC35 CC0B CD35 CD4B CD7D CD9B CD4D CD4 CD4 CD29 CE0A CE30 CE5E CE7C CE7C CE9C CE9C	P8AOUT4 LFGEN RESTORE CKKBDXIT PROMPTLOOP SCREENOUT TOSCREEN REVMASK TRANCMD DELAYSET BAUDCMD PARITYCMD RESETCMD ICMD NCMD CMDPROC1 STATETBL CSTATETC ZEROSTATE CSTATE1C ZEROSTATE CSTATE4A CMDSEARCH	

	STSBYTE	0638	PWDBYTE	
06B8		0738	MISCFLG	
C010	KBDSTRB	C081	DIPSW1	
C088		C089	STREG	
	CTLREG	?C700	BINIT	
C711	BENTRY	C745	BINIT1	
C75C	NORMIO	C767	BOUTPUT	
C78E	PINIT	C794	PREAD	
C7A8	PSTATIN	C7AB	PSTATUS 2	
C7EE	RESTORHOOK	C800	PASCALINIT	
C835	INIT2	C83C	INIT2A	
C857	INIT3	C864	INIT4	
C882	INITACIA1	C88F	INITACIA2	
	PASEXIT	CBAA	GETCHAR	
C8B8	BASICEXIT	C8BF	BINPUT	
C8E5	BINEND1	CSEA	BINACIA	
C917	COMMA	C921	TABCHECK	
?C941	BATCHOUT	C948	TAB2	
C968	FORCECR	C97A	SEREND	
C99B	PENTRY	C9A6	PWDTBL	
C9C0	DECRCOL	C9C8	ADJRTS	
C9E5	CKINPUT1	C9EB	CKINPUT2	
C9FD	MOVOUT	CAOC	MOVIN	
CA2B	TERMNEXT	CA31	TERMNEXT1	
?CA4C	TERMEXIT	CA54	TERMRTS	
?CA7D	TERMNORM	CA81	TERMINC	
CA93	TERMSEND	CA95	TERMS END1	
CAB1	TERMLOCK	CAB9	TRANSLATE	
CAD2	SRIN	CAE9	SRIN1	
	SROUT	CAFF	INPUT	
CB19	NOINPUT1	CB1A	CMDSEQCK	
CB58	NOCMD	CB1A CB59	ANRTS	
CB58		CB59 CB6B	OUTPUT2	
?CB9C	ACK	CBA6	P8AOUT2	
CBC1	OUTPUT3	CBA6 CBD9		
			OUTDLY1	
CC29	OUTPUTEND	CBFF CC2C	DLYTBL CKKBD	
	GETKBD	CC44	GETKBD1	
CC72	GETCMD			
CCA3	SCREENOUT1	CC91 CCB6	GETKBDONE NOOUT	
CCC6	TESTLETTER	CCD3	LCMASK	
CCDF	GETXLATE	CCEB	CMDTBL	
	CRCMD	CD41	LFCMD	
	ROTATE	CD61	SSLOTCMD	
CD81	BAUDCMD1	CD88	BAUDCMD2	
CD9C	DATACMD	CDA1	DATACMD1	
CDB9	QUITCMD	CDBD	QCMD1	
CDEO	KCMD	CDE2	KCMD1	
CDF4	ZCMD	CDFB	ZCMDRTS	
CEOE	CMDPROC 2	CE1A	CMDPROC3	
CE38	CSTATE1	CE43	CSTATE1A	
CE62	Contraction of the second second	CE6A	ACCLOOP	
CE7B	SETOSTATE	CE8A	CDONE	
CEA1	CSTATE4B	CEA4	SETSTATE6	
CEB9		CEBD		
CEDD	CMDEXEC	CEF2	CMDEXEC1	
FDED	COUT	FDF 6	VIDOUT	
FF58	IORTS			

0638 CHNBYTE 07F8 MSLOT CO82 DIPSW2 CO89 RESET C705 IENTRY C751 FROMOUT C77C BOUTPUT1 C797 PWRITE C7B0 SENDCD C805 INIT1 C83F INIT2B C872 INIT5 C89B PASCALREAD C8B4 GETCHAR1 C8CB BINKBD CREF BINACTA C934 TAB1 C951 NOTAB C996 SETCH C9AA PASCALWRITE C9C9 CTRLTST C9EE CIEND CA1E CHECKTERM CA41 TERMNEXT2 CA55 TERMACIAIN CA82 TERMINC1 CA9B TERMCAP CAC4 WAITMS CAER SRIN2 CB16 INPUT2 CB2E ESCCHECK CB5A XONWATT 2CB76 PSAOUT1 CBA9 P8AOUT3 CBE2 OUTDLYLP CC02 ACIAOUT CC34 CKKBD1 CC5C KBDESC CC93 PROMPTBL CCB7 ASCREEN CCD7 UCMASK CD19 CMDTBL1 CD47 FFCMD CD7C SSLOTCMD1 CD8E CTLREGSET CDA7 TERMCMD CDC1 BREAKCMD CDE5 KCMD2 CDFC CMDPROC CE24 CMDPROC4 CE54 CSTATE1B CE75 CSTATE2A CE8B CSTATE4 CEA8 STATERR CECE CMDMATCH CFFF ROMSOFF FE89 SETKBD

.

100

. 100

100

-

1

1 100

1 

1

Max 1 100

1 1

# APPENDIX B **APPLE INTERFACE** CARD EMULATION

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 +	24Ø)	
POKE	49338,	1Ø7	(4929Ø	+	3*16;	p =	1Ø7)	
POKE	2043,	132	(plug :	in	magic	numb	er)	
POKE	1147, 6	64	(plug :	in	magic	numb	per)	

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		123 444		
Baud Rate: 50 75 110 135 150 300 600 1200 1200 1800 2400 3600 4800 7200 9600 19200	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 32 dec/\$2Ø hex 16 dec/\$1Ø hex 11 dec/\$ØB hex 8 dec/\$Ø8 hex 5 dec/\$Ø5 hex 4 dec/\$Ø4 hex (not available) 2 dec/\$Ø1 hex	POKE $49291+s*16$ , r = b + d; b = 1 dec/\$ $\emptyset$ 1 hex 2 dec/\$ $\emptyset$ 2 hex 3 dec/\$ $\emptyset$ 3 hex 4 dec/\$ $\emptyset$ 4 hex 5 dec/\$ $\emptyset$ 6 hex 7 dec/\$ $\emptyset$ 6 hex 7 dec/\$ $\emptyset$ 6 hex 9 dec/\$ $\emptyset$ 8 hex 9 dec/\$ $\emptyset$ 8 hex 10 dec/\$ $\emptyset$ 8 hex 11 dec/\$ $\emptyset$ 8 hex 12 dec/\$ $\emptyset$ 6 hex 13 dec/\$ $\emptyset$ 6 hex 13 dec/\$ $\emptyset$ 6 hex 14 dec/\$ $\emptyset$ 6 hex 15 dec/\$ $\emptyset$ 7 hex		
Data Format: 8 data,l stop 7 data,l stop 6 data,l stop 5 data,l stop 8 data,2 stop 7 data,2 stop 6 data,2 stop 5 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 = 6; t = 1* r = 6; t = 1* r = 9; t = 2* r = 8; t = 2* r = 7; t = 2* r = 6; t = 2* r = 6; t = 2* r = 6; t = 2*	(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/\$B0 hex 208 dec/\$D0 hex 240 dec/\$F0 hex		
Parity: none odd even MARK SPACE		POKE 14ØØ+s,p p = 2 p = 1 p = Ø (not available) (not available)	POKE 4929Ø+s*16 p = 11 (\$ØB her p = 43 (\$2B her p = 1Ø7 (\$6B her (not available) (not available)		

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

	SSC switches	PEEKs and POK	ES to use for
Selection	and settings	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=Ø</cr>	POKE 1784+s,r r=4Ø to 255; for no <cr>, PEEK 14ØØ+s, POKE 14ØØ+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	$ \begin{array}{l} V = Video \ on? \\ G = Gen. ? \\ T = LC \ to \ UC? \\ D = Dly \ 1/4 \ s? \\ POKE \ 2\emptyset 4\emptyset + s, r \\ r = \\ \hline \frac{dec}{4} \frac{hex}{\$\emptyset 4} \frac{V}{Y} \frac{G}{N} \frac{T}{Y} \frac{D}{Y} \end{array} $	$V = Video \text{ on?}$ $G = Gen. \langle LF \rangle?$ $POKE 2\emptyset 4\emptyset + s, r$ $r =$ $\frac{dec}{4} \frac{hex}{\$ \emptyset 4} \frac{V}{N} \frac{G}{N}$
	Printer Mode)	5 \$Ø5 Y Y Y Y 36 \$24 Y N N Y 37 \$25 Y Y N Y 68 \$44 Y N Y N 69 \$45 Y Y Y N 1ØØ \$64 Y N N N	5 \$ 05 N Y $132 $ 84 Y N$ $133 $ 85 Y Y$ $T = LC to UC?$
		1Ø1 \$65 Y Y N N 132 \$84 N N Y Y 133 \$85 N Y Y Y 164 \$A4 N N N Y	D = Dly 1/4 s? POKE 1144+s,r r =
		165 \$A5 N Y N Y 196 \$C4 N N Y N 197 \$C5 N Y Y N	Ø \$ØØ Y N
	ing pangla lagangan na malan lagangan	228 \$E4 N N N N 229 \$E5 N Y N N	

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  $\langle CR \rangle$  delay switch now determines whether an ETX/ACK handshake is performed after each  $\langle CR \rangle$  that is transmitted. The corresponding RAM bit was not the same as the P8  $\langle CR \rangle$  delay bit, but was kept in bit 2 of location 1400+s. For SSC emulation, the control is the same as the  $\langle CR \rangle$  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.

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  $14\emptyset\emptyset$ +s. For example, to change the enquire character to ENQ (ASCII 5), which is used by many RS-232 devices, use this POKE: POKE  $14\emptyset\emptyset$ +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**

----

-

MERT I MAR

ALL I

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 CSØ (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 <CTRL-S> as the ASCII XOFF character. When it receives <CTRL-S> 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 4 $\emptyset$  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  $4 \, \emptyset_{\, \bullet}$ 

#### 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

2-3/4" x 7" (68.8 mm x 177.8 mm) 3 oz. (9Ø gm), approximately internal cable from 1Ø-pin header on SSC to DB-25 connector on case of Apple II (supplied); shielded RS-232-C cable to external device (not supplied) 2 blocks of 7 switches each, set by user before installation none 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)

SPECIAL CIRCUITS

SY6551 2316

Asynchronous Communications Interface Adapter Read Only Memory (2,048 by 8 bits) with SSC firmware The SSC has the usual power supply bypassing capacitors BASIC programs APPLESOFT programs PASCAL programs any slot except slot #Ø any slot except slot #Ø slot #1 for use with printer, etc. slot #2 for use with modem slot #3 for use with terminal

#### SOFTWARE COMPATIBILITY

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

Integer BASIC	DOS 3.2	Pascal 1.Ø	65Ø2 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  $3\emptyset\emptyset$  baud. There are at least three solutions to this problem: lower the baud rate to  $3\emptyset\emptyset$  baud; reduce the scrolling window size (using 2 fewer lines already makes  $12\emptyset\emptyset$  baud possible), or use an  $8\emptyset$ -column card with automatic hardware scrolling.

# **CONNECTOR PIN ASSIGNMENTS**

Table C-1 lists the signals assigned to the connector pins on the  $1\emptyset$ -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.

lØ−pin Header	DB-25 Connector	Signal name	
1	1	Frame Ground	DB-25
2	2	Transmit Data (TXD)	$\left[ \cdot \cdot \right]$
3	3	Receive Data (RXD)	·:
4	4	Request To Send (RTS)	1: •
5	5	Clear To Send (CTS)	
6	6	Data Set Ready (DSR)	· :
7	19	Secondary Clear To Send (SCTS)	
8	7	Signal Ground	• :
9	2Ø	Data Terminal Ready (DTR)	1
1Ø	8	Data Carrier Detect (DCD) 13	:.)
Tab	10 (-1 (0)	nnector Pin Assignments	0

JUMPER BLOCK WIRING

100

-

-

1

THE PARTY OF

NII

 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  $\emptyset$  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 (2Ø)	Data Set Ready (6)
Data Carrier Detect	Data Carrier Detect (8)	Request To Send (4)
Data Carrier Detect	Data Carrier Detect (8)	Clear To Send (5)*

\*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



# APPENDIX D 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.

7-BIT	LOW	LOW	HI	HI	ASCII	THERRORDER	WHAT TO T	
BINARY	DEC	HEX	DEC	HEX	CHAR	INTERPRETATION	NORMAL	TERMINAL
ØØØØØØØ	ø	ØØ	128	8Ø	NUL	Blank (null)	<ctrl-@></ctrl-@>	
0000001	1	ØI	129	81	SOH	Start of Header	<ctrl-a></ctrl-a>	1
0000010	2	Ø2	130	82	STX	Start of Text	<ctrl-b></ctrl-b>	
ØØØØØ11	3	Ø3	131	83	ETX	End of Text	<ctrl-c></ctrl-c>	2
0000100	4	Ø4	132	84	EOT	End of Transm.	<ctrl-d></ctrl-d>	
0000101	5	Ø5	133	85	ENQ	Enguiry	<ctrl-e></ctrl-e>	3
0000110	6	Ø6	134	86	ACK	Acknowledge	<ctrl-f></ctrl-f>	4
0000111	7	07	135	87	BEL	Bell	<ctrl-g></ctrl-g>	
0001000	8	Ø8	136	88	BS	Backspace	<ctrl-h></ctrl-h>	5
ØØØ1ØØ1	9	Ø9	137	89	HT	Horizontal Tab	<ctrl-i></ctrl-i>	6
0001010	10	ØA	138	8A	LF	Linefeed	<ctrl-j></ctrl-j>	
ØØØ1Ø11	11	ØB	139	8B	VT	Vertical Tab	<ctrl-k></ctrl-k>	
ØØØ11ØØ	12	ØC	14Ø	8C	FF	Form Feed	<ctrl-l></ctrl-l>	
ØØØ11Ø1	13	ØD	141	8D	CR	Carriage Return	<ctrl-m></ctrl-m>	7
0001110	14	ØE	142	8E	SO	Shift Out	<ctrl-n></ctrl-n>	
ØØØ1111	15	ØF	143	8F	SI	Shift In	<ctrl-0></ctrl-0>	
0010000	16	1Ø	144	9Ø	DLE	Data Link Escape	<ctrl-p></ctrl-p>	
ØØ1ØØØ1	17	11	145	91	DC1	Device Control 1	<ctrl-q></ctrl-q>	8
ØØ1ØØ1Ø	18	12	146	92	DC2	Device Control 2	<ctrl-r></ctrl-r>	9
ØØ1ØØ11	19	13	147	93	DC3	Device Control 3	<ctrl-s></ctrl-s>	1Ø
ØØ1Ø1ØØ	2Ø	14	148	94	DC4	Device Control 4	<ctrl-t></ctrl-t>	11
ØØ1Ø1Ø1	21	15	149	95	NAK	Neg. Acknowledge	<ctrl-u></ctrl-u>	12
ØØ1Ø11Ø	22	16	15Ø	96	SYN	Synchronization	<ctrl-v></ctrl-v>	
ØØ1Ø111	23	17	151	97	ETB	End of Text Blk.	<ctrl-w></ctrl-w>	
ØØ11ØØØ	24	18	152	98	CAN	Cance1	<ctrl-x></ctrl-x>	
ØØ11ØØ1	25	19	153	99	EM	End of Medium	<ctrl-y></ctrl-y>	
ØØ11Ø1Ø	26	1A	154	9A	SUB	Substitute	<ctrl-z></ctrl-z>	
ØØ11Ø11	27	1 B	155	9B	ESC	Escape	<esc></esc>	13 <esc></esc>

1. Normal command character in Communication Mode.

Used in ETX/ACK protocol (SIC P8A Emulation Mode).
 Used in ENQ/ACK protocol (SIC P8A Emulation Mode).

4. Used in ETX/ACK or ENQ/ACK protocol (SIC P8A Emulation Mode).

- 5. 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.

 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 DEC	LOW HEX	HI DEC	HI HEX	ASCII CHAR	INTERPRETATION	WHAT TO TYP NORMAL T	E ERMINA
1	ØØ111ØØ	28	10	156	90	FS	File Separator	n/a	<esc></esc>
	ØØ111Ø1	29	1D	157	9D	GS	Group Separator	<ctrl-shift< td=""><td>-M&gt;</td></ctrl-shift<>	-M>
	ØØ1111Ø	3Ø	1E	158	9E	RS	Record Separator	<ctrl-shift< td=""><td>-N&gt;</td></ctrl-shift<>	-N>
	ØØ11111	31	1F	159	9F	US	Unit Separator	n/a	(ESC>
	0100000	32	20	16Ø	AØ	SP	Space	spacebar	
	Ø1ØØØØ1	33	21	161	A1	1	· •	1 1 1	
	0100010	34	22	162	A2			н	
	Ø1ØØØ11	35	23	163	A3	#		#	
	0100100	36	24	164	A4	Ş		Ş	
	Ø1ØØ1Ø1	37	25	165	A5	%		%	
	Ø1ØØ11Ø	38	26	166	A6	&		&	
	Ø1ØØ111	39	27	167	A7	,	Closing Quote	110 · 160 · 120	
		40	28	168	A8	(	oroorug daore	(	
	Ø1Ø1ØØØ Ø1Ø1ØØ1	41 41	29	169	A9	;		)	
				17Ø	AA	*		*	
	Ø1Ø1Ø1Ø	42	2A	170		+		+	
	Ø1Ø1Ø11	43	2B	171	AB AC	T	Comma		
	Ø1Ø11ØØ	44	2C 2D	172	AD	2	Hyphen	,	
	Ø1Ø11Ø1	45		174	AE		Period	10.00	
	Ø1Ø111Ø	46	2E	174	AE	i	reriou	;	
	Ø1Ø1111	47	2F					Ø	
	Ø11ØØØØ	48	30	176	BØ	ø		Ø 1	
	Ø11ØØØ1	49	31	177	B1	1		2	
	Ø11ØØ1Ø	50	32	178	B2	2		3	
	Ø11ØØ11	51	33	179	B3	3		4	
	Ø11Ø1ØØ	52	34	18Ø	B4	4		5	
	Ø11Ø1Ø1	53	35	181	B5	5		6	
	Ø11Ø11Ø	54	36	182	B6	6		7	
	Ø11Ø111	55	37	183	B7	7		8	
	Ø111ØØØ	56	38	184	B8	8		8	
	Ø111ØØ1	57	39	185	B9	9		9	
	Ø111Ø1Ø		3A	186	BA	:		:	
	Ø111Ø11	59	3B	187	BB	;		;	
	Ø1111ØØ		3C	188	BC	<		<	
	Ø1111Ø1	61	3D	189	BD	=		=	
	Ø11111Ø		3E	19Ø	BE	>		>	
	Ø111111	63	3F	191	BF	?		?	
	1000000		4Ø	192	CØ	Q		0	
	1000001	65	41	193	C1	A		A	
	1000010		42	194	C2	В		В	
	1000011	67	43	195	C3	С		C	
	1000100		44	196	C4	D		D	
	1000101		45	197	C5	Е		Е	
	1000110		46	198		F		F	
	1000111		47	199	C7	G			
	1001000		48	2ØØ	C8	Н		п	
	1001001	73	49	2Ø1	C9	I		I	
	1001010		4A	2Ø2	CA	J		J	
	1001011		4B	2Ø3	CB	K		14	
	1001100			2Ø4		L	inden inden inden		
	1001101		4D	2Ø5	CD	М			
	1001110	1 78	4E	2Ø6	CE	N		N	

7-BIT	LOW	LOW	HI	HI	ASCII		WHAT TO	
BINARY	DEC	HEX	DEC	HEX	CHAR	INTERPRETATION	NORMAL	TERMINAL
1001111	79	4F	2Ø7	CF	0		0	
010000	80	50	208	DØ	P		P	
1Ø1ØØØ1	81	51	209	D1	0		0	
010010	82	52	210	D2	R		R	
1010011	83	53	211	D3	S		S	
1010100	84	54	212	D4	T		T	
01010101	85	55	213	D5	Ů		U	
010101	86	56	214	D6	v		V	
1010111	87	57	214	D7	W		W	
IØ11ØØØ	88	58	215	D8	X		w X	
1Ø11ØØ1	89	59	210	D0	A Y		A Y	
	9Ø	59 5A	217		-		Z	
	1.			DA	Z	Orandara Burghash		(100)
1011011	91	5B	219	DB	[	Opening Bracket	n/a	<esc></esc>
1Ø111ØØ	92	5C	220	DC	1	Reverse Slant	n/a	<esc>4</esc>
1Ø111Ø1	93	5D	221	DD	]	Closing Bracket	<shift-m< td=""><td>-</td></shift-m<>	-
1Ø1111Ø	94	5E	222	DE	20	Circumflex		(200)
LØ11111	95	5F	223	DF	-	Underline	n/a	<esc>5</esc>
1100000	96	6Ø	224	EØ		Opening Quote	n/a	15
1100001	97	61	225	E1	a		n/a	a
1100010	98	62	226	E2	b		n/a	b
1100011	99	63	227	E3	С		n/a	С
1100100		64	228	E4	d		n/a	d
11ØØ1Ø1	1Ø1	65	229	E5	е		n/a	e
11ØØ11Ø		66	23Ø	E6	f		n/a	f
11ØØ111	1Ø3	67	231	E7	g		n/a	g
11Ø1ØØØ	1Ø4	68	232	E8	h		n/a	h
11Ø1ØØ1	1Ø5	69	233	E9	i		n/a	i
11Ø1Ø1Ø	1Ø6	6A	234	EA	j		n/a	j
11Ø1Ø11	1Ø7	6B	235	EB	k		n/a	k
11Ø11ØØ	1Ø8	6C	236	EC	1		n/a	1
11Ø11Ø1	1Ø9	6D	237	ED	m		n/a	m
11Ø111Ø	11Ø	6E	238	EE	n		n/a	n
11Ø1111	111	6F	239	EF	0		n/a	0
111ØØØØ		7Ø	24Ø	FØ	р		n/a	р
111ØØØ1	113	71	241	F1	q		n/a	p
111ØØ1Ø		72	242	F2	r		n/a	r
111ØØ11	115	73	243	F3	S		n/a	S
111Ø1ØØ	116	74	244	F4	t		n/a	t
111Ø1Ø1	117	75	245	F5	u		n/a	u
111Ø11Ø	118	76	246	F6	v		n/a	v
111Ø111	119	77	247	F7	w		n/a	W
1111000	12Ø	78	248	F8	x		n/a	х
1111001	121	79	249	F9	у		n/a	у
1111Ø1Ø	122	7A	25Ø	FA	z		n/a	Z
1111011		7B	251	FB	{	Opening Brace	n/a	<esc>6</esc>
1111100		7C	252	FC	i	Vertical Line	n/a	<esc>7</esc>
1111101		7D	253	FD	}	Closing Brace	n/a	<esc>8</esc>
1111110		7E	254	FE	~	Overline (Tilde)	n/a	<esc></esc>
1111111		7F	255		DEL	Delete/Rubout	n/a	<esc>:</esc>

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

# APPENDIX E TROUBLESHOOTING HINTS

PIL -

FIL I

FILE

1 1

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 connec- tions, then pin assign- ments; 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 like this	printer and SSC both generating <lf> after <cr></cr></lf>	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
		in sadaun anna dalar No sadaun anna dalar	if device supports only ETX/ACK, set SIC P8A Mode
		energen by Low-on on the censter words	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>
	Suna P. Suck	(1.e. (S-1.e. dat) =	have software send <cr> before right margin</cr>

Ce :

6

6

1

-

FI

E

E

3

IL I

1

3

3

Table E-1. Problems Detected at the Device

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 SI 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 i Comm Mode or use <n>N cm in Printer Mode</n>
screen is spacing double	lines look like this	device generating and sending <lf> after <cr></cr></lf>	use M(ask E(nable comman to remove extra linefeed
forced uppercase display	lowercase beCOMES UPPERCASE	Apple monitor changing letters in GETLINE routine	use <n>T command to allo lowercase to pass throug (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; us assembly language or fas ter program routines; us lower baud rate (300 vs. l200); use <n>C, <n>L or <n>F commands; in Comm Mode, chain (<n>S cmd) t 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

10

ALL IN

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-l lists the bit definitions of this byte:





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.

 $\bigcirc$ 

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

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.

bit, and a scotting wave as the untransmitting three bits of the Status believes (Apprindix AL, SIT 3 fedicates whether of not bats Corrier Detect (GCD), GARREY () signal wort fains at any morting the recal or operation. SIC 3 to set () any of the other are sai, as a overall avior inficator. If bit its the only of, at meaning was detected. If all hirs are (), and contracts

 you can block this stabus byte via a 2010 (50/6+2 (a in the b), and rear if whith a PORE company as the same local tura.

#### the sources and an article reader with

# GLOSSARY

1

----

1.00

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, "Ø" and "l," with each digit position moving from right to left representing a successive power of two. For example, l represents decimal l; 1Ø represents 2; 1ØØ represents 4; 1ØØØ represents 8, etc.

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

- 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 Ø 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 SCØ8A+sØ) 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  $C(\emptyset B+s\emptyset)$  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.

EN I IN

(# 1 H

100 1 100

. .

IL IN

11 1

.....

1.1

- 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.
- 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  $\emptyset$  or 1 as necessary to make the total number of 1 bits an even number. For example, the 7-bit ASCII code for the letter A ( $1\emptyset\emptyset\emptyset\emptyset\emptyset1$ ) has two 1 bits; for even parity, the transmitting device appends an eighth bit equal to  $\emptyset$  so that the total number of 1 bits remains even. The receiving device can count 1 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 l (FRM) of its Status Register to l. 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 Ø or l (that is, Ø or l times 2 to the Ø 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.

E11

-

ALL IN

- 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 l times 2 to the power one less than the total number of bits in the binary number. For example, in the binary number 1ØØØØ, the l represents l 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 (1ØØØØ1) 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.

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.
- 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 \$CØ88+sØ) 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**

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

# Chapter 2 PRINTER MODE

5	Figure 2-1	SSC Set for Printer Mode
10		SSC in Slot #1 and Clamp Assembly in Notch
6	Table 2-1	Commonly Used Switch Settings for Printer Mode
7	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

	21	Figure 3-1	SSC Set for Communications Mode
1	26		SSC in Slot #2 and Clamp Assembly in Notch
9	22	Table 3-1	Commonly Used Switch Settings for Communications Mode
	23	Table 3-2	Baud Rate Switch Settings
4	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
•	33	Table 3-9	Lowercase Character Displays
	35	Table 3-10	Special ASCII Character Generation
	35	Table 3-11	Terminal Mode Commands

# Chapter 4 HOW THE SSC WORKS

39	Figure 4-1	Parallel Data Transfer
39	Figure 4-2	
39	0	
40	Figure 4-4	RS-232-C Serial Data Format
42	Figure 4-5	An RS-232-C Setup with Modems
43	Figure 4-6	An RS-232-C Setup with a Modem Eliminator
43	Figure 4-7	SSC Operating Modes
44	Figure 4-8	SSC Configurations
45	Figure 4-9	Overall Block Diagram of the SSC
38	Table 4-1	Binary and Decimal Digits and Quantities
41	Table 4-2	RS-232-C Signals as Interpreted by the Sender
41	Table 4-3	RS-232-C Signals as Interpreted by the Receiver
46	Table 4-4	SSC Address Remapping
47	Table 4-5	Registers in SSC Peripheral I/O Space

# Appendix A FIRMWARE

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

# Appendix B

# APPLE INTERFACE CARD EMULATION

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

# Appendix C SPECIFICATIONS AND SCHEMATICS

Table C-1Connector Pin AssignmentsTable C-2Jumper Block Wiring

# Appendix E TROUBLESHOOTING HINTS

106Table E-1Problems Detected at the Device107Table E-2Problems Detected at the Apple

# Appendix F ERROR CODES

109 T 110 T

1

11100

1120

-

and a

-

CO.

ALC: N

12 15

-

-

-

1

7

98

99

Table F-1STSBYTE Bit DefinitionsTable F-2Error Codes and Bits

120 SUPER SERIAL CARD



# INDEX

A

# 

A register 50 ACIA 45, 47, 48, 91, 95, 96, 97 function 47 registers 49, 54 status register 109 ACK 91, 94, 102, 106 address bus 45 lines 46 mapping 46 space 46 addressing logic 45, 46 answer modem 42 Apple II Communications Card (see CIC) Apple II Communications Interface Card (see CIC) Apple II Parallel Cards 96 Apple II Serial Interface Card (see SIC) APPLE SSC: 8, 12, 27 Applesoft 36, 98 ASCII 24, 34, 38, 95, 96 code table 101-104 special characters 35, 36, 101 assembly language 11, 27, 98 Asynchronous Communications Interface Adapter (see ACIA)

#### B

B command 35, 36 BASIC 8, 11, 12, 17, 18, 19, 27, 31, 33, 36, 92, 101, 104 entry points 55 error codes 109, 110 protocol 50 batch moves 95 baud rate 6, 15, 22, 30 selection 15, 30 switch settings 7, 23, 47, 91, 92, 93, 98 binary 37, 101 data 24, 38 binary/decimal representation 38 bit 7, 23, 37 data 30, 40, 93 parity 40 start 40 stop 30, 40, 92, 94 block diagram, SSC 45 moves 91 branch table locations, Pascal 1.1 50 break signal 36 buffer 45, 47 bus address 45 data 45, 48 parallel 48 byte 38 device signature 51

# С

cable connector 10, 26 external 1, 10, 26 internal 1, 9, 24, 45 socket 2 capitalize 34 carriage return delay 7, 17, 94, 95, 96 carrier lost 110 signal 42 chaining 34 character displays 33 enquire 95 characters echo on screen 34 lowercase 34, 102 uppercase 34, 102 capitalize 34 CIC 36, 91, 96 clamp assembly 2, 3 Clear To Send (see CTS) code 38 column overflow 17 command character change 13, 28 Communications Mode 28, 102

#### send 12 data format 15 format 12, 27 Communications Mode 27-28 register 45 remote-control 102 reset 15, 95 set time delay 16 summary, Terminal Mode 35 translate 91 commands CIC 96 Communications Mode 27, 29, 102 Parallel Card 96 Printer Mode 11, 14 Terminal Mode 22, 35 communications 37 mode 21-33, 91, 95, 96 Communications Card, Apple II (see CIC) Communications Mode 22, 26 command character 28 summary 28-29 commands 27 control character 27 components SSC 45 configurations, SSC 44 connector pin assignments 98 slots 9, 25, 98 control character communications mode 27 suppress 18, 33 logic 45, 46 register 45 <CR> 7, 17, 94, 95, 96, 102, 107 <CTRL-A> 27, 102 <CTRL-I> 8, 12, 102 <CTRL-R> 28, 35, 96, 102 <CTRL-S> 28, 96, 102 <CTRL-SHIFT-M> 101, 103 <CTRL-SHIFT-N> 103 <CTRL-T> 28, 35, 96, 102 CTS 41, 42, 43, 47, 48, 95, 98, 99

Printer Mode 12, 102

data 37 ASCII 24 binary 24 bit 7, 15, 23, 24, 30, 40, 93 bus 45, 48 bus buffer 45 conversion 39, 47 format 7, 23, 28, 30, 40. 47, 91, 92, 93 command 15 RS-232-C 40 selection 15, 23, 30 input 48, 96 output 48 serial 48 terminal 40 transfer 47 Data Carrier Detect (see DCD) Data Communication Equipment (see DCE) Data Set Ready (see DSR) Data Terminal Equipment (see DTE) Data Terminal Ready (see DTR) DB-25 connector 10, 26, 97, 99 pin assignments 98 DCD 41, 42, 43, 47, 48, 95, 98, 99 DCE 41, 48, 105 decimal error codes 110 default parity 16 demodulator 41 device class digit 51 identification 51 remote 19, 96 signature byte 51 DEVICE SELECT 46, 47 disable 12, 28 DOS 8, 11, 27 Dow Jones News & Quotes Reporter 22, 36 setup program for 36 DSR 41, 42, 43, 48, 95, 98, 99 DTE 41, 42, 48, 105 DTR 41. 42, 43, 95, 98, 99

D

E E command 31, 34, 35 echoing 34, 35 edge connector 2 EIA 40 80-column-display 34 Electronic Industries Association (see EIA) emulation modes 91-96 EL C. HER old Serial Interface Card 91 P8 92 SIC 92, 102 enable 12, 28 enquire character 95 11. I. III enter Terminal Mode 35 entry points BASIC 55 hardware 49 I/O routine 50 monitor ROM 55 Pascal 1.0 55, 56 Pascal 1.1 55, 56 SSC 55 environment, SSC 97 error codes 109-110 BASIC 110 Pascal 110 framing 109, 110 parity 109, 110 ENQ 95, 102 ENO/ACK 102 <ESC> 34, 35, 101-104 <ESC>L 91 <ESC>U 91 ETX 91, 94, 95, 102, 106 ETX/ACK 91, 94, 102, 106 Expansion ROM, SSC 46, 52 even parity 16, 31, 40, 92, 93 external cable 1, 10, 26 F F command 17, 31 FCC rules 10, 26 <FF> 14, 16, 17, 29, 32 firmware 34, 47, 49, 109 entry points, Pascal 1.0 (SIC) 49

interrupt handling 50

listings 57-96

CT. IN

#### locations, special 57 memory usage 53 protocol, Pascal 1.1 49 SSC driver 46 form feed delay 14, 16, 17, 29, 32 format, data 92, 93 framing error 109, 110 full-duplex 35

# G

generate ASCII characters 35 <CR> 17, 95, 96 <LF> 8, 17, 24, 31, 92, 94 generic signature 51 ground 10, 26, 42, 43

#### Н

half-duplex 35, 36, 95 handshake 94, 95 handshaking 19 hardware 96 entry points 49 handshaking 19 hexadecimal 101

#### 1

IN# 14 IDS 560 Paper Tiger 6 indirect addressing 56 input 37 data 48, 96 9 installation, SSC Integer BASIC 36 interface receive 45 transmit 45 internal cable 1, 9, 24, 45 attachment 3 clamp assembly 2, 3 shielding 10, 26 interrupt handling 50 request 47 Pascal 1.0 50

INDEX 125

I/O
routine entry points 50
scratchpad address 109
SELECT 46
space
peripheral card 52
registers in 54
STROBE 46
IORESULT 109

#### J

jumper block 2, 5, 11, 21, 27, 42, 45, 48, 95, 99, 105 wiring 99

#### K

keyboard 11, 18, 27, 33, 101, 102

#### L

<LF> 8, 17, 24, 31, 91, 92, 94, 102 line width 8, 92, 94, 96 linefeed (see <LF>) listings, firmware 57-96 local 37 lowercase 14, 18, 29, 32-33, 34, 91, 92, 93, 95, 101, 102 character displays 18

#### M

MARK parity 16, 31, 40, 101
mask <LF> in 17, 31
memory usage map, SSC 53
mode
communications 21-33, 26,
91, 96, 102
emulation 91-96
P8 93
P8A 93-94, 102, 106
printer 5-19, 91, 92, 93-94,
102
SSC operating 43
terminal 22, 34-36, 96, 101,
102

### Ν

NEC 5510 Spinwriter 6 no parity 31 null modem 42, 48

### 0

odd parity 16, 31, 40 offsets, Pascal 1.1 57 originate 40 modem 42 oscillator 45, 47 output 37 data 48 override switch settings 27 overrun 109, 110

#### Ρ

P8 91-95 emulation POKEs 92 ROM 92, 94, 95 P8A 91 mode 6, 102, 106 ROM 92, 94, 95 parallel bus 48 card commands 96 data 37, 38-39 I/O card 52 parity 7, 15, 23, 30, 47, 92, 93 bit 7, 23, 24, 40 error 109, 110 even 40 MARK 40, 101 odd 40 selection 16, 31



SPACE 40 Pascal 11, 12, 27 error codes 109, 110 Pascal 1.0 49 entry points 55, 56 interrupt requests 50 Pascal 1.1 49 branch table locations 50 device class digit 51 identification 51 signature byte 51 entry points 55, 56 firmware protocol 49 I/O routines 50 offsets 50 offsets 57 PEEK 91, 92, 93-94, 109 peripheral I/O space 49 physical characteristics, SSC 97 pin assignments 98 POKE 19, 91, 92, 93-94, 95 power supply 9, 25 PR# 14 preparing SSC, printer mode 5 printer(s) 6 local 11 using 11 Printer Mode 5-19, 91, 92, 93-94, 95 command character 12, 102 commands 11 summary 13-14 data format 7, 15 switch settings 6 using 11 protocol 102 BASIC 50 Pascal 1.1 firmware 49

### Q

Q command 35, 36 Qume 6 quit Terminal Mode 36, 102

#### R

RAM 45, 49, 92 scratchpad locations 53 receive interface 45 register 42 Received Data (see RXD) register command 45, 54 control 45, 54 receive 42 status/reset 45, 54 transmit 42, 54 transmit/receive 45 registers A 50, 55-57 ACIA 49, 54 Pascal 1.1 50 peripheral I/O space 47 switch settings 45 x 50, 55-57 ¥ 50, 55-57 remote 37 device 19, 22, 26, 27, 96 terminal 34 remote-control commands 102 Request To Send (see RTS) reset 27 command 15, 95 SSC 17, 31 RESET 46 <RETURN> 12, 27, 102 Ring Indicator (RI) 41 ROM 45 SSC Expansion 46, 52 ROM/RAM address mapping 46 space 46 RS-232-C 40, 43, 47, 48, 95, 105 signals 40-43, 49, 98-99 RTS 41, 42, 43, 95, 98, 99 RXD 41, 42, 43, 98, 99

## S

schematic, SSC 45, 100 scratchpad RAM locations 53 SCTS 98, 99 Secondary Clear To Send (see SCTS) selection baud rate 15, 30 data format 15, 23, 30 parity 16, 31 time delay 16, 32

serial data 37, 39, 40, 48 interface card 95 setting switches 6, 22, 92 settings, switch 6, 22, 93-94 shielding, internal 10, 26 SIC 91, 92 emulation mode 92, 102 firmware entry points 49 P8A mode 102, 106 switch settings 93, 94 6502 registers 49, 50, 55-57 slew rate 48 software compatability 98 SPACE parity 16, 31, 40, 101 special characters, ASCII 101 circuits, SSC 97 specifications, SSC 97-98 specify screen slot 34 SSC 1, 91, 95, 96, 102 block diagram 45 cable connector 10 preparation 21 configurations 44 connect printer 7 terminal 8 controlling 11 driver firmware 46 emulation 94 entry points 50, 55-57 environment 97 Expansion ROM 46, 52 firmware memory usage 53 installation 9, 92 lowercase 18 main components 45 operating modes 43 peripheral card I/O space 52 photo 1 physical characteristics 97 registers in I/O space 54 reset 17, 31 RS-232-C signals 99 schematic 45, 100 scratchpad RAM locations 53 slot location 98 special circuits 97 specifications 97-98 switch settings 93-94 theory of operation 45 unpacking 1

using Communications Mode 27 printer 11 Printer Mode 11 terminal 11 zero-page locations 52 start bit 7, 23, 40 status byte (see STSBYTE) register, ACIA 109 status/reset register 45 stop bit 7, 15, 23, 24, 30, 40, 92, 94 STSBYTE 109 summary Communications Mode commands 28-29 Printer Mode commands 14 Terminal Mode commands 35 Super Serial Card (see SSC) suppress characters control 33 keyboard 18, 33 <LF> in 17, 31 switch settings baud rate 7, 23 change 14, 30 Communications Mode 22 line width 8 override 12, 27 Printer Mode 6 SIC 93-94 SSC 93-94 video 8 switches 2 Communications Mode 21-22 setting 22, 92 SSC 45 state of 47 SW1 48, 54, 56, 57, 92, 93 SW1-1 through SW1-4 2, 7, 15, 23, 31, 92, 93, 106 SW1-5 and SW1-6 2, 5, 21, 92, 93, 102, 115 SW1-7 2, 8, 24, 48, 92, 99 SW2 6, 22, 48, 54, 56, 57, 92, 93, 94 SW2-1 2, 7, 15, 23-24, 31, 92, 93, 94, 106 SW2-2 2, 7, 16, 23-24, 31, 92, 94, 106 SW2-3 and SW2-4 2, 8, 23, 92, 94, 106, 107

CI III 99 1 1 1 1 11 1 10 1 1 10 tab 19 terminal ----11 1 10 104 ----EL L 1 I II HELL STREET State of Concession, Name Real Property lies U 36

SW2-5 2, 8, 17, 24, 32, 92, 93, 106 SW2-6 2, 8, 24, 48, 92, 107 SW2-7 2, 8, 24, 48, 92,

T command 35 connecting 8 local 11 unintelligent 34, 35 TERMINAL 5, 11, 21, 42, 48, 95, 99, 101, 105 Terminal Mode 22, 34-36, 101bypass 96 change to 96 commands 35 enter 35, 102 example 36 quit 36, 102 theory of operation, SSC 45 time delay selection 16, 32 time-sharing 34, 36 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

unintelligent terminal 34, 35, 36 uppercase 14, 18, 29, 33, 34, 91, 92, 102 V

video on/off 8, 92, 94, 96

#### W

wiring, jumper block 99

# Х

X command 17, 31, 35, 106 X register 50, 55-57 XOFF 35, 102 recognition 18, 33 XON 102 XON/XOFF 17, 19, 31, 33, 96, 102

## Y

Y register 50, 55-57

### Ζ

Z command 17, 18, 31 zero-page locations, SSC 52