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.
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 vii

Chapter 1 GETTING STARTED 1
1 Unpacking
2 A Close Look
3 Preparing Cable and Clamp Assembly
4 Attaching Internal Cable to SSC

Chapter 2 PRINTER MODE 5
5 Preparing the SSC for Printer Mode
6 Setting the Switches
7 Commonly Used Settings
8 Baud Rate
9 Data Format and Parity
10 Carriage Return Delay
11 Line Width and Video On/Off
12 Generate <LF> Out
13 Special Switches
14 Installation Procedure
15 External Cable and Connector
16 Using the SSC in Printer Mode
17 With a Printer
18 With a Terminal
19 Printer Mode Commands
20 Command Formats
21 The Command Character
22 Printer Mode Command Summary
23 Commands That Change Switch Settings
24 Baud Rate--<n>B
25 Data Format--<n>D
26 Parity--<n>P
27 Set Time Delay--<n>C, <n>L, <n>F
28 Generate <CR> On Column Overflow--<C
29 Generate <LF> Out--<L <E/D>
30 Mask (Suppress) <LF>--<M <E/D>
31 Reset the SSC--<R

ii SUPER SERIAL CARD

CONTENTS iii
Chapter 3

COMMUNICATIONS MODE

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

Appendix A

FIRMWARE

49 Pascal 1.1 Firmware Protocol
50 I/O Routine Entry Points
51 Device Identification
52 SSC Firmware Memory Usage
53 Zero Page Locations
54 Scratchpad RAM Locations
55 Peripheral Card I/O Space
56 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 PRA Emulation POKEs
95 Other Emulation Mode Differences
PREFACE

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.

CHAPTER 1
GETTING STARTED

This chapter takes you through the first steps of getting acquainted with your Super Serial Card (SSC). After unpacking the SSC and examining it, you will assemble the short internal cable (if it is not already assembled) that connects the 10-pin cable socket on the SSC to the 25-pin socket at the back of the Apple II case.

UNPACKING

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 edge connector. It is important not to touch the gold fingers on this connector: they must make a clean electrical contact in the Apple II connector slot when you install the SSC (Chapter 2 or Chapter 3).

- The cable socket. 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](image)

PREPARING CABLE AND CLAMP ASSEMBLY

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

![Figure 1-3. Components of Internal Cable and Clamp Assembly](image)

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 10-pin end of the cable: the wires come out of the SIDE of the connector—the same side as the raised "key" in the plastic (Figure 1-3). Hold the connector so
the wires are on the side away from the SSC, and insert the
connector firmly into the cable socket along the right edge of the
SSC. The raised "key" should slide into the groove in the cable
socket (Figure 1-4).

If the cable is now jammed between the 10-pin cable socket and
the SSC board, the connector is plugged in backwards. Unplug
the connector and reconnect it so that the cable is on the side
AWAY from the SSC (Figure 1-5).

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

Figure 1-5. Internal Cable Attached Correctly to SSC

CHAPTER 2
PRINTER MODE

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

PREPARING THE SSC FOR PRINTER MODE

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

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

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.

Figure 2-1. SSC Set for Printer Mode
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 560
SW1: OFF OFF OFF ON OFF ON SW2: ON ON * * OFF OFF OFF
Paper Tiger Printer Mode, HW Hndshk, 9600 baud, 1 stop bit, ** width
IDS SW1: -- -- ON ON OFF OFF SW2: -- -- OFF --
SSC/IDS pins: 3/3, 7/7, 28/28; all IDS jumpers removed

NEC 5510
SW1: OFF ON OFF OFF ON OFF OFF SW2: ON ON * * OFF OFF OFF
Spinwriter Printer Mode, ETO/Ack, 1200 baud, 1 stop bit, ** line width
NEC switches: OFF ON OFF OFF OFF OFF ON
SSC/NEC pins: 2/2, 3/3, 7/7, 28/68; 465 tied on NEC end
May need keystore to force first ETX after power-up.

NEC 5510
SW1: OFF ON OFF OFF ON OFF OFF SW2: ON ON * * OFF OFF OFF
Spinwriter Printer Mode, hardware handshaking, rest same as above
NEC switches: OFF ON OFF OFF OFF OFF ON
SSC/NEC pins: 3/3, 6/68, 7/7, 28/28; 465 NOT tied

Qume SW1: OFF ON OFF OFF ON SW2: ON OFF * * OFF OFF OFF
Sprint 5 Printer Mode, HW Hndshk, 1200 baud, 1 stop bit, ** width
Qume switches: 1200 baud, no modem; pins: 3, 4, 7, 28
Qume asserts RTS and DTR only when ready to receive data

Qume SW1: OFF OFF OFF OFF OFF 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 baud rate. Since the Apple II can usually send and receive information faster than what is connected to it, the simplest way to determine the baud rate is to consult the user manual for the device you will connect. Find out what rate is the fastest the device can handle (up to 19,200 baud). Once you know this, you are ready to set the baud rate switches on the SSC.

<table>
<thead>
<tr>
<th>Baud</th>
<th>SW1-1</th>
<th>SW1-2</th>
<th>SW1-3</th>
<th>SW1-4</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>ON</td>
<td>ON</td>
<td>ON</td>
<td>OFF</td>
</tr>
<tr>
<td>75</td>
<td>ON</td>
<td>ON</td>
<td>OFF</td>
<td>ON</td>
</tr>
<tr>
<td>110*</td>
<td>ON</td>
<td>ON</td>
<td>OFF</td>
<td>ON</td>
</tr>
<tr>
<td>135**</td>
<td>ON</td>
<td>OFF</td>
<td>ON</td>
<td>OFF</td>
</tr>
<tr>
<td>150</td>
<td>OFF</td>
<td>OFF</td>
<td>OFF</td>
<td>ON</td>
</tr>
<tr>
<td>300</td>
<td>OFF</td>
<td>OFF</td>
<td>OFF</td>
<td>ON</td>
</tr>
<tr>
<td>600</td>
<td>OFF</td>
<td>OFF</td>
<td>OFF</td>
<td>OFF</td>
</tr>
</tbody>
</table>

(* 109.92) (**) 134.58

Table 2-2. Baud Rate Switch Settings

DATA FORMAT AND PARITY

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

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

For Printer Mode, the only aspect of the data format you can change with switch settings is whether to send one stop bit or two. If you set the baud rate switches to 50, 75 or 110 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 <n>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 (CCHR). 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 <Esc> 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 40, the Apple II video screen is turned on, since it too can display 40 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.

<table>
<thead>
<tr>
<th>Line Width</th>
<th>Video Screen</th>
<th>SW2-3</th>
<th>SW2-4</th>
</tr>
</thead>
<tbody>
<tr>
<td>40 char/line</td>
<td>on</td>
<td>ON</td>
<td>ON</td>
</tr>
<tr>
<td>72 char/line</td>
<td>off</td>
<td>ON</td>
<td>OFF</td>
</tr>
<tr>
<td>80 char/line</td>
<td>off</td>
<td>OFF</td>
<td>ON</td>
</tr>
<tr>
<td>132 char/line</td>
<td>off</td>
<td>OFF</td>
<td>OFF</td>
</tr>
</tbody>
</table>

Table 2-3. Line Width and Video Switch Settings

The switch settings that turn off the Apple II video screen take effect only after P&F 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 (<LF>) after a carriage return (<CR>). 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 28, 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 connector slots. The connector slots are numbered from 0 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.

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 #1).
- 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 #1).
- Under Pascal, boot the Apple II and then use the terminal as the input/output console (with the SSC in slot #1).
- 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 <n> is given in each command description (next section). The choice of Enable or Disable is indicated as <E/D>.

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:

**Input BASIC:**

```
PRINT "#command" 'embedded <CTRL-I>
```

**Applesoft BASIC:**

```
PRINT CHR$(9): "command"
```

**Pascal:**

```
WRITELN (CHR(9), 'command');
```

**PRINTER MODE COMMAND SUMMARY**

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

This command overwrites 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>D4B<RETURN>.

<table>
<thead>
<tr>
<th><code>&lt;n&gt;</code></th>
<th>SSC Baud Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>0</code></td>
<td>use SW1-1 to SW1-4</td>
</tr>
<tr>
<td><code>1</code></td>
<td>50</td>
</tr>
<tr>
<td><code>2</code></td>
<td>75</td>
</tr>
<tr>
<td><code>3</code></td>
<td>100.92 (110)</td>
</tr>
<tr>
<td><code>4</code></td>
<td>134.58 (135)</td>
</tr>
<tr>
<td><code>5</code></td>
<td>150</td>
</tr>
<tr>
<td><code>6</code></td>
<td>300</td>
</tr>
<tr>
<td><code>7</code></td>
<td>600</td>
</tr>
</tbody>
</table>

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 `<n>`. For example, `<CTRL-I>D2D<RETURN>` causes the SSC to transmit each character in the form: one start bit (always transmitted), six data bits, and one stop bit.

<table>
<thead>
<tr>
<th><code>&lt;n&gt;</code></th>
<th>Data Bits</th>
<th>Stop Bits</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>0</code></td>
<td>8</td>
<td>1</td>
</tr>
<tr>
<td><code>1</code></td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td><code>2</code></td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td><code>3</code></td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td><code>4</code></td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td><code>5</code></td>
<td>7</td>
<td>2</td>
</tr>
<tr>
<td><code>6</code></td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td><code>7</code></td>
<td>5</td>
<td>2</td>
</tr>
</tbody>
</table>

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 2-6).

<table>
<thead>
<tr>
<th><code>&lt;n&gt;</code></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>0</code></td>
<td>odd parity</td>
</tr>
<tr>
<td><code>1</code></td>
<td>even parity</td>
</tr>
<tr>
<td><code>2</code></td>
<td>MARK (parity bit always 1)</td>
</tr>
<tr>
<td><code>3</code></td>
<td>SPACE (parity bit always 0)</td>
</tr>
</tbody>
</table>

Table 2-7. Parity Selections

### Table 2-4. Printer Mode Commands

<table>
<thead>
<tr>
<th>Format</th>
<th>Command Name</th>
<th>Values</th>
<th>Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;n&gt;</code></td>
<td>Baud Rate</td>
<td>0-15</td>
<td>see Table 2-5</td>
</tr>
<tr>
<td><code>&lt;n&gt;</code></td>
<td><code>&lt;CR&gt;</code> Delay</td>
<td>0</td>
<td>no delay</td>
</tr>
<tr>
<td></td>
<td><code>1</code></td>
<td>32 milliseconds</td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>2</code></td>
<td>250 milliseconds</td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>3</code></td>
<td>2 seconds</td>
<td></td>
</tr>
<tr>
<td><code>&lt;n&gt;</code></td>
<td>Data Format</td>
<td>0</td>
<td>8 data bits, 1 stop bit</td>
</tr>
<tr>
<td></td>
<td><code>1</code></td>
<td>7 data bits, 1 stop bit</td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>2</code></td>
<td>6 data bits, 1 stop bit</td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>3</code></td>
<td>5 data bits, 1 stop bit</td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>4</code></td>
<td>8 data bits, 2 stop bits</td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>5</code></td>
<td>7 data bits, 2 stop bits</td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>6</code></td>
<td>6 data bits, 2 stop bits</td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>7</code></td>
<td>5 data bits, 2 stop bits</td>
<td></td>
</tr>
<tr>
<td><code>&lt;n&gt;</code></td>
<td><code>&lt;FF&gt;</code> Delay</td>
<td>0</td>
<td>no delay (default)</td>
</tr>
<tr>
<td></td>
<td><code>1</code></td>
<td>32 milliseconds</td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>2</code></td>
<td>250 milliseconds</td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>3</code></td>
<td>2 seconds</td>
<td></td>
</tr>
<tr>
<td><code>&lt;n&gt;</code></td>
<td><code>&lt;LF&gt;</code> Delay</td>
<td>0</td>
<td>no delay (default)</td>
</tr>
<tr>
<td></td>
<td><code>1</code></td>
<td>32 milliseconds</td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>2</code></td>
<td>250 milliseconds</td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>3</code></td>
<td>2 seconds</td>
<td></td>
</tr>
<tr>
<td><code>&lt;n&gt;</code></td>
<td>Parity</td>
<td>0, 2, 4, 6</td>
<td>no parity (default = 0)</td>
</tr>
<tr>
<td></td>
<td><code>1</code></td>
<td>odd parity</td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>3</code></td>
<td>even parity</td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>5</code></td>
<td>MARK (parity bit always 1)</td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>7</code></td>
<td>SPACE (parity bit always 0)</td>
<td></td>
</tr>
<tr>
<td><code>&lt;n&gt;</code></td>
<td>Translate</td>
<td>0</td>
<td>change LC to UC (default)</td>
</tr>
<tr>
<td></td>
<td><code>1</code></td>
<td>leave LC (possible garbage)</td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>2</code></td>
<td>LC to UC inverse; leave UC</td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>3</code></td>
<td>LC to UC; LC to inverse</td>
<td></td>
</tr>
<tr>
<td><code>&lt;n&gt;</code></td>
<td>Lowercase (LC)</td>
<td>0</td>
<td>auto-&lt;CR&gt; at column's end</td>
</tr>
<tr>
<td><code>&lt;n&gt;</code></td>
<td>Column Overflow</td>
<td>1</td>
<td>reset SSC + PR/® and IN®</td>
</tr>
<tr>
<td><code>&lt;n&gt;</code></td>
<td>Reset the SSC</td>
<td>2</td>
<td>ignore all <code>&lt;CTRL&gt;</code> commands</td>
</tr>
<tr>
<td><code>&lt;n&gt;</code></td>
<td>Sap <code>&lt;CTRL&gt;</code></td>
<td>3</td>
<td></td>
</tr>
<tr>
<td><code>&lt;n&gt;</code></td>
<td>Find Keyboard</td>
<td>E or D</td>
<td>accept keyboard entries</td>
</tr>
<tr>
<td><code>&lt;n&gt;</code></td>
<td>Generate <code>&lt;LF&gt;</code> Out</td>
<td>E or D</td>
<td>send <code>&lt;LF&gt;</code> out after <code>&lt;CR&gt;</code></td>
</tr>
<tr>
<td><code>&lt;n&gt;</code></td>
<td>Mask <code>&lt;LF&gt;</code> In</td>
<td>E or D</td>
<td>drop <code>&lt;LF&gt;</code> in after <code>&lt;CR&gt;</code></td>
</tr>
<tr>
<td><code>&lt;n&gt;</code></td>
<td>Tab in BASIC</td>
<td>E or D</td>
<td>recognize BASIC tabs</td>
</tr>
<tr>
<td><code>&lt;n&gt;</code></td>
<td>XOFF Recognition</td>
<td>E or D</td>
<td>detect XOFF; await XON</td>
</tr>
<tr>
<td><code>&lt;n&gt;</code></td>
<td>Not supported by Pascal.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2-4. Printer Mode Commands
Parity to Use

\( \emptyset, 2, 4 \text{ or } 6 \) none (default value)
1 odd parity (odd total number of ones)
3 even parity (even total number of ones)
5 MARK parity (parity bit always 1)
7 SPACE parity (parity bit always \( \emptyset \))

Table 2-7. Parity Selections

For example, type \(<CTRRL-D>1F<RETURN>\) to cause the SSC to transmit and check for odd parity. Odd parity means that the high bit of every character is \( \emptyset \) 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-\((n)C, (n)L, (n)F\)

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 printer head enough time to return to the left side of the page so it is ready to continue printing.

The \(<n>C\) 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 \(<n>L\) 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 \(<n>F\) 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).

\( \emptyset \) none
1 32 milliseconds
2 250 milliseconds (1/4 second)
3 2 seconds

Table 2-8. Time Delay Selections

Consult the user manual for the printer to find out how much time it takes to move its print head and platen, and so to determine an appropriate set of values for these three delays. The idea is to have at least enough time for the printer parts to move the required distance, but not so much time that overall printing speed is slowed down drastically. A typical set for a very slow printer would be \(<CTRRL-D>2C<RETURN>, <CTRRL-D>2L<RETURN>, <CTRRL-D>3F<RETURN>\); 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 \(<CTRRL-D>C<RETURN>\) 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 \(<LF>\) Out-L_\(<E/D>\)

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

Mask (Suppress) \(<LF>\) In-M_\(<E/D>\)

If you type \(<CTRRL-D>M<RETURN>\), the SSC will suppress any incoming linefeed character that immediately follows a carriage return character.

Reset the SSC-R

Typing \(<CTRRL-D>R<RETURN>\) has the same effect as sending a PR\# and an IN\# 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 "translate" 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

0  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.

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

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

The only way to reinstate command recognition after the Z command is to reinitialize the SSC, or clear the high-order bit at location $5F84+s$ (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 \(<\text{CTRL}-I\text{X}\><\text{RETURN}>\) 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—\(<E/D>\)
If you type in \(<\text{CTRL}-I\text{E}\><\text{RETURN}>\), the BASIC horizontal position counter is left equal to the column count. All TABs work, including back-tabs. TABs beyond column 40 require a POKE to location 36, as usual. Commas only work as far as column 40, and BASIC programs will be listed in 40-column format.
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

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.

<table>
<thead>
<tr>
<th>Application</th>
<th>Switch Settings, Cable Connections, Other Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apple II</td>
<td>SW1: ON OFF OFF ON ON ON SW2: ON ON ** OFF OFF OFF</td>
</tr>
<tr>
<td>via modem</td>
<td>Comm Mode, 300 baud, 8 data, 1 stop, ** parity</td>
</tr>
<tr>
<td></td>
<td>If using SSC in each Apple, set both the same; for local</td>
</tr>
<tr>
<td></td>
<td>connection, second jumper block points toward TERMINAL.</td>
</tr>
<tr>
<td>Apple III</td>
<td>SW1: ON OFF OFF ON ON ON SW2: ON ON ** OFF OFF OFF</td>
</tr>
<tr>
<td>via modem</td>
<td>Comm Mode, 300 baud, 8 data, 1 stop, ** parity</td>
</tr>
<tr>
<td></td>
<td>Set Apple III RS-232-C Device Control Block to same</td>
</tr>
<tr>
<td></td>
<td>values (See Apple III Standard Device Drivers manual).</td>
</tr>
<tr>
<td>Printer</td>
<td>SW1: ON OFF OFF ON ON ON SW2: ON OFF ** OFF OFF OFF</td>
</tr>
<tr>
<td>via modem</td>
<td>Comm Mode, 300 baud, 7 data, 1 stop, ** parity</td>
</tr>
<tr>
<td></td>
<td>Baud rate is limited by modem and transmission lines;</td>
</tr>
<tr>
<td></td>
<td>some modems can also use 1200 baud; SW1-7 is always ON,</td>
</tr>
<tr>
<td></td>
<td>and SW2-7 is always OFF;CTS hookup is at remote modem.</td>
</tr>
<tr>
<td>Dow Jones</td>
<td>SW1: ON OFF OFF ON ON ON SW2: ON OFF OFF OFF OFF</td>
</tr>
<tr>
<td>News and Quotes</td>
<td>Comm Mode, 300 baud, 7 data, 1 stop, no parity</td>
</tr>
<tr>
<td>Reporter</td>
<td>Use T command for Terminal Mode operation.</td>
</tr>
</tbody>
</table>

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 baud rate. Since the Apple II can usually send and receive information faster than what is connected to it, the simplest way to determine the 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.

<table>
<thead>
<tr>
<th>Baud</th>
<th>SW1-1</th>
<th>SW1-2</th>
<th>SW1-3</th>
<th>SW1-4</th>
<th>SW2-1</th>
<th>SW2-2</th>
<th>SW2-3</th>
<th>SW2-4</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>ON</td>
<td>ON</td>
<td>ON</td>
<td>OFF</td>
<td>1200</td>
<td>OFF</td>
<td>ON</td>
<td>ON</td>
</tr>
<tr>
<td>75</td>
<td>ON</td>
<td>ON</td>
<td>OFF</td>
<td>ON</td>
<td>1800</td>
<td>OFF</td>
<td>ON</td>
<td>OFF</td>
</tr>
<tr>
<td>110*</td>
<td>ON</td>
<td>ON</td>
<td>OFF</td>
<td>OFF</td>
<td>2400</td>
<td>OFF</td>
<td>ON</td>
<td>OFF</td>
</tr>
<tr>
<td>135**</td>
<td>ON</td>
<td>OFF</td>
<td>ON</td>
<td>ON</td>
<td>3600</td>
<td>OFF</td>
<td>OFF</td>
<td>ON</td>
</tr>
<tr>
<td>150</td>
<td>ON</td>
<td>OFF</td>
<td>OFF</td>
<td>ON</td>
<td>4800</td>
<td>OFF</td>
<td>OFF</td>
<td>ON</td>
</tr>
<tr>
<td>300</td>
<td>ON</td>
<td>OFF</td>
<td>OFF</td>
<td>OFF</td>
<td>7200</td>
<td>OFF</td>
<td>OFF</td>
<td>ON</td>
</tr>
<tr>
<td>600</td>
<td>OFF</td>
<td>OFF</td>
<td>OFF</td>
<td>OFF</td>
<td>9600</td>
<td>OFF</td>
<td>OFF</td>
<td>OFF</td>
</tr>
</tbody>
</table>

(* 109.92) (** 134.58)

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

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

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

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

<table>
<thead>
<tr>
<th>Stop Bits</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>SW2-1</td>
<td>ON</td>
<td>OFF</td>
</tr>
<tr>
<td>SW2-2</td>
<td>8</td>
<td>7</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Data Bits</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>SW2-3</td>
<td>none</td>
<td>odd</td>
</tr>
<tr>
<td>SW2-4</td>
<td>--</td>
<td>ON</td>
</tr>
</tbody>
</table>

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.

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

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

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

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

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

USING SSC IN COMMUNICATIONS MODE
Communications Mode allows you to use the SSC with a modem, connected to a remote device (such as a remote printer, terminal, or other computer). After installing the SSC, you can control its operation from a BASIC, Pascal or assembly-language program, or even directly from the keyboard. To use the SSC in Communications Mode, do the following:

- Make sure the jumper block points toward MODEM.
- 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, LD to Disable automatic generation of linefeed characters)

The allowable range of <n> is given in each command description below. The choice of Enable or Disable is written as E/D.

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 Enable is the same as E E. The SSC ignores invalid commands.

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

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

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

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

Integer BASIC: PRINT """command" "embed+ed <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.

<table>
<thead>
<tr>
<th>Format</th>
<th>Command Name</th>
<th>Values</th>
<th>Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;n&gt;B</td>
<td>Baud Rate</td>
<td>0 - 15</td>
<td>see Table 3-5</td>
</tr>
<tr>
<td>&lt;n&gt;C</td>
<td>&lt;CR&gt; Delay</td>
<td>1</td>
<td>no delay</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>32 milliseconds</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
<td>250 milliseconds(1/4 s)</td>
</tr>
<tr>
<td>&lt;n&gt;D</td>
<td>Data Format</td>
<td>1</td>
<td>8 data bits, 1 stop bit</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>7 data bits, 1 stop bit</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
<td>6 data bits, 1 stop bit</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
<td>5 data bits, 1 stop bit</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5</td>
<td>4 data bits, 2 stop bits</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6</td>
<td>3 data bits, 2 stop bits</td>
</tr>
<tr>
<td></td>
<td></td>
<td>7</td>
<td>2 data bits, 2 stop bits</td>
</tr>
<tr>
<td>&lt;n&gt;F</td>
<td>&lt;FF&gt; Delay</td>
<td>1</td>
<td>no delay (default)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>32 milliseconds</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
<td>250 milliseconds (1/4 s)</td>
</tr>
<tr>
<td>&lt;n&gt;L</td>
<td>&lt;LF&gt; Delay</td>
<td>1</td>
<td>no delay (default)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>32 milliseconds</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
<td>250 milliseconds (1/4 s)</td>
</tr>
<tr>
<td>&lt;n&gt;P</td>
<td>Parity</td>
<td>0,2,4,6</td>
<td>no parity (default = 0)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>odd parity</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
<td>even parity</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5</td>
<td>MARK (parity bit always 1)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>7</td>
<td>SPACE (parity bit always 0)</td>
</tr>
<tr>
<td>&lt;n&gt;S</td>
<td>Screen Slot</td>
<td>0 - 7</td>
<td>chain SSC output to slot n</td>
</tr>
<tr>
<td>&lt;n&gt;T</td>
<td>Translate</td>
<td>0</td>
<td>change all LC to UC</td>
</tr>
<tr>
<td></td>
<td>Lowercase (LC)</td>
<td>1</td>
<td>leave LC (possible garbage)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>LC to UC inverse; leave LC</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
<td>LC to UC; UC to inverse</td>
</tr>
<tr>
<td>B</td>
<td>Break</td>
<td>1</td>
<td>transmit 233 ms BREAK</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>SW reset + PR# and IN#</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
<td>(see Terminal Mode section)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
<td>ignore all &lt;CTRL&gt; commands</td>
</tr>
<tr>
<td>*</td>
<td>* E&lt;GD</td>
<td>Echo</td>
<td>E or D</td>
</tr>
<tr>
<td>*</td>
<td>F&lt;GD</td>
<td>Find Keyboard</td>
<td>E or D</td>
</tr>
<tr>
<td>*</td>
<td>L&lt;GD</td>
<td>Generate &lt;LF&gt; Out</td>
<td>E or D</td>
</tr>
<tr>
<td>*</td>
<td>M&lt;GD</td>
<td>Mask &lt;LF&gt; In</td>
<td>E or D</td>
</tr>
<tr>
<td>*</td>
<td>X&lt;GD</td>
<td>XOFF Recognition</td>
<td>E or D</td>
</tr>
<tr>
<td>*</td>
<td>* Not supported by Pascal.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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—\(<n>8\)B

This command overrides the physical settings of switches SW1-1 to SW1-4 on the SSC. For example, to change the rate to 9600 baud, type <CTRL>-A+16<RETURN>.

<table>
<thead>
<tr>
<th>&lt;n&gt;</th>
<th>SSC Baud Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>use SW1-1 to SW1-4</td>
</tr>
<tr>
<td>1</td>
<td>50</td>
</tr>
<tr>
<td>2</td>
<td>75</td>
</tr>
<tr>
<td>3</td>
<td>109.92 (110)</td>
</tr>
<tr>
<td>4</td>
<td>134.58 (135)</td>
</tr>
<tr>
<td>5</td>
<td>150</td>
</tr>
<tr>
<td>6</td>
<td>300</td>
</tr>
<tr>
<td>7</td>
<td>600</td>
</tr>
</tbody>
</table>

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 <n>. For example, typing <CTRL>-A+3D<RETURN> causes the SSC to transmit each character in the form: one start bit (always transmitted), five data bits, and one stop bit.

<table>
<thead>
<tr>
<th>&lt;n&gt;</th>
<th>Data Bits</th>
<th>Stop Bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>8</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>2 (1 with &lt;n&gt;P options 4 through 7)</td>
</tr>
<tr>
<td>5</td>
<td>7</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td>5</td>
<td>2 (1-1/2 with &lt;n&gt;P options 0 through 3)</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>(&lt;n&gt;)</th>
<th>Parity to Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>0, 2, 4 or 6</td>
<td>none</td>
</tr>
<tr>
<td>1</td>
<td>odd parity (odd number of 1's)</td>
</tr>
<tr>
<td>3</td>
<td>even parity (even number of 1's)</td>
</tr>
<tr>
<td>5</td>
<td>MARK parity (parity bit always 1)</td>
</tr>
<tr>
<td>7</td>
<td>SPACE parity (parity bit always 0)</td>
</tr>
</tbody>
</table>

For example, type <CTRL>-A+1P<RETURN> to cause the SSC to transmit and check for odd parity. Odd parity means that the high bit of every character is 0 if there is 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—\(<E/D>\)

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

Mask (Suppress) \(<LF>\) In—\(<M>E/D>\)

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

Reset the SSC—R

Typing <CTRL>-A+R<RETURN> has the same effect as sending a FR# and an IN# 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 \(\text{CR}\), \(\text{LF}\) and \(\text{FF}\) 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 GnD command routes video output to a selected slot, and the E command suppresses display (echo) of characters on the screen.
Set Time Delays—\(<n>\)C, \(<n>\)L, \(<n>\)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 \(<n>\)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 \(<n>\)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 \(<n>\)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).

<table>
<thead>
<tr>
<th>(&lt;n&gt;)m</th>
<th>Time Delay</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\emptyset)</td>
<td>none</td>
</tr>
<tr>
<td>1</td>
<td>32 milliseconds</td>
</tr>
<tr>
<td>2</td>
<td>250 milliseconds (1/4 second)</td>
</tr>
<tr>
<td>3</td>
<td>2 seconds</td>
</tr>
</tbody>
</table>

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 \(<n>\)T command, four options are available:

- Change all lowercase characters to uppercase before passing them to a BASIC program or to the video screen. This is what the Apple II monitor does to lowercase.
- Pass along all lowercase characters unchanged. The appearance of the lowercase characters on the Apple II screen is undefined (garbage).
- Display lowercase characters as uppercase inverted 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 \(<\text{CTRL}-A>\)X.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 $5F8+4$ (with the SSC in slot a).

Find Keyboard—F_\(<\text{E/D}>\)

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

XOFF Recognition—X_\(<\text{E/D}>\)

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 \(<\text{CTRL}-A>\)X.DRETURN>.  

32 SUPER SERIAL CARD

COMMUNICATIONS MODE 33
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 #0, the Apple II video screen.) This allows "chaining" of the SSC to another card slot, such as an 80-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 C0C0 space; this is the case for all currently available 80-column-display cards for the Apple II.

For example, let's say you have the SSC in slot #2 with a remote terminal connected to it, and an 80-column-display card in slot #3. Type <CTRL-A>35<RETURN> to cause the data from the remote terminal to be chained through the card in slot #3, so that it is displayed on the Apple II in 80-column format. (Not available in Pascal.)

Echo Characters on the Screen-E_<E/D>
For the Apple II, as for most computers, displaying (echoing) a character on the video screen is a separate step from receiving it from the keyboard, though we tend to think of these as one step, as on a typewriter. For example, if you type in <CTRL-A> E<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 <ESC> once (if necessary) to place the keyboard in lowercase mode. Then press <ESC> 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 <ESC> followed by 9.

Table 3-10. Special ASCII Character Generation

<table>
<thead>
<tr>
<th>ESC followed by</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>generates:</td>
<td>FS</td>
<td>US</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>ESC RUB</td>
</tr>
<tr>
<td>or in hexadecimal:</td>
<td>9C</td>
<td>9F</td>
<td>DB</td>
<td>DC</td>
<td>DF</td>
<td>FB</td>
<td>FC</td>
<td>FE</td>
<td>9B</td>
<td>FF</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Format</th>
<th>Command Name</th>
<th>Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Enter Terminal Mode</td>
<td>Go into Terminal Mode.</td>
</tr>
<tr>
<td>2</td>
<td>Transmit a Break Signal</td>
<td>Send a 233-millisecond BREAK (signoff) signal.</td>
</tr>
<tr>
<td>* E_&lt;E/D&gt;</td>
<td>Echo Enable/Disable</td>
<td>Default E D (full-duplex); use X E for half-duplex.</td>
</tr>
<tr>
<td>S_&lt;E/D&gt;</td>
<td>Special Characters Enable/Disable</td>
<td>Default S E; allows/deactivates generation of lowercase and special characters (Table 3-10).</td>
</tr>
<tr>
<td>* Cn&gt;T</td>
<td>Translate Lowercase Characters</td>
<td>Determine treatment of incoming lowercase characters.</td>
</tr>
<tr>
<td>* X_&lt;E/D&gt;</td>
<td>XOFF Recognition Enable/Disable</td>
<td>Default X E; in Terminal Mode, X E makes SSC detect &lt;CTRL-E&gt; and &lt;CTRL-T&gt; (remote-control OFF &amp; ON, respectively), but not &lt;CTRL-S&gt;.</td>
</tr>
<tr>
<td>0</td>
<td>Quit (Exit from) Terminal Mode</td>
<td>Return to normal Communications Mode operation.</td>
</tr>
</tbody>
</table>

* Fully described earlier in this chapter.

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 ECHO E(nable) command.
Transmit a Break Signal—B
Typing &lt;CTRL-A&gt;&lt;RETURN&gt; causes the SSC to transmit a
233-millisecond break signal, recognized by most time-sharing
systems as a signoff.

Special Characters—$&lt;E/D&gt;
Typing &lt;CTRL-A&gt;$ &lt;RETURN&gt; causes the SSC to interpret &lt;ESC&gt;&lt;n&gt;
pairs as special characters, allowing a keyboard in this way to
generate all possible ASCII characters. If you type &lt;CTRL-A&gt;$
&lt;RETURN&gt;, the SSC will treat the &lt;ESC&gt; key like any other key.

Quit (Exit from) Terminal Mode—Q
Type &lt;CTRL-A&gt;&lt;Q&gt;&lt;RETURN&gt; 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 &amp; Quotes Reporter. This program assumes that the SSC
is set for Communications Mode and that the jumper block is pointing
toward MODEN. Neither of these conditions can be changed by
software. This program also assumes that the SSC is in slot #1 and
that you want to chain I/O to an 80-column card in slot #3; these
conditions you can change via software. To change this integer
BASIC program to an AppleSoft program, substitute CHR$(5) for D8 and
CHR$(2) for A8, and leave out program lines 40 and 42.

10 REM *****************************************************************************
20 REM * THIS PROGRAM SETS THE SSC FOR DOW JONES *
30 REM **************************************************************************
40 DS="\": REM TYPE &lt;CTRL-D&gt; ESCAPE CHARACTER BETWEEN QUOTES
42 AS="": REM TYPE &lt;CTRL-A&gt; COMMAND CHARACTER BETWEEN QUOTES
50 PRINT DS;"FR01": REM SSC IS IN SLOT #1;
52 PRINT AS;"6 BAUD": REM SET BAUD RATE TO 300;
54 PRINT AS;"1 DATA": REM DATA FORMAT OF 7 DATA, 1 STOP
56 PRINT AS;"7 PARITY": REM AND NO PARITY;
58 PRINT AS;"LF DISABLE": REM NO &lt;LF&gt; GENERATION AFTER &lt;CR&gt;.
60 PRINT AS;"3 SLOTCBN": REM CHAIN TO CARD IN SLOT #3
62 PRINT AS;"TERM MODE": REM AND ENTER TERMINAL MODE.
70 REM *****************************************************************************
72 REM * NOW YOU SHOULD BE IN TERMINAL MODE, GETTING THE *
74 REM # INFO YOU NEED FROM THE DOW JONES SERVICE. WHEN *
76 REM * FINISHED, EXIT WITH THE &lt;CTRL-A&gt;&lt;Q&gt; (QUIT COMMAND). *
78 REM *****************************************************************************
100 REM Q (QUIT COMMAND SENDS CONTROL BACK TO THIS PROGRAM:
110 PRINT AS;"RESET": REM RESET SWITCH-SELECTED OPTIONS
120 END

CHAPTER 4
HOW THE SSC 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 for the 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 serial data communication. Let's
consider communication first, then data, and then serial data and
data transfer.

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

Information moving from here to there (out of the Apple) is called
output; information moving from there to here (into the Apple) is
called input.

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.

<table>
<thead>
<tr>
<th>System</th>
<th>Digits</th>
<th>Using</th>
<th>You can represent</th>
</tr>
</thead>
<tbody>
<tr>
<td>decimal</td>
<td>0 - 9</td>
<td>1</td>
<td>ten items (0 through 9)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>one hundred (0 through 99)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
<td>one thousand (0 through 999)</td>
</tr>
<tr>
<td>binary</td>
<td>0 and 1</td>
<td>1</td>
<td>two items (0 or 1)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>four (0, 1, 10 or 11)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
<td>eight (0 through 11)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
<td>sixteen (0 through 111)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5</td>
<td>thirty-two (0 through 1111)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6</td>
<td>sixty-four (0 through 11111)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>7</td>
<td>one hundred twenty-eight</td>
</tr>
<tr>
<td></td>
<td></td>
<td>8</td>
<td>two hundred fifty-six, etc.</td>
</tr>
</tbody>
</table>

Table 4-1. Binary and Decimal Digits and Quantities

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

Throughout the world, post, telegraph, telex, and wire services use 5-bit and 6-bit code sets, even though so few bits cannot represent a very large selection of items. Meanwhile, computers have a penchant for sending each other streams of 8-bit codes for 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 eight-bit processor because the basic unit of data it uses and moves around internally is an eight-bit byte. The Apple II has sets of eight lines interconnecting its various internal parts, so it can move around all eight bits at the same time. Since the bits travel together like eight cars side by side on an eight-lane highway, data in the Apple II is called parallel data, and data movements within the Apple II are called parallel data transfers (Figure 4-1).

SERIAL DATA FOR LONG DISTANCES

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

DATA CONVERSION

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

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

![Figure 4-4. RS-232-C Serial Data Format](image)

What is this mysterious parity bit all about? It is an optional extra bit set to 0 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 0 (called space parity) or to 1 (MARK parity).

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

RS-232-C SIGNALS

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

<table>
<thead>
<tr>
<th>Signal Description</th>
<th>Abbrev.</th>
<th>Similar to</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Terminal Ready</td>
<td>DTR</td>
<td>you pick up the phone</td>
</tr>
<tr>
<td>Data Set Ready</td>
<td>DSR</td>
<td>the phone is working</td>
</tr>
<tr>
<td>Request To Send</td>
<td>RTS</td>
<td>you want to talk</td>
</tr>
<tr>
<td>Clear To Send</td>
<td>CTS</td>
<td>the phone has established a connection and the person at the other end is ready to listen</td>
</tr>
<tr>
<td>Transmit Data, not Request To Send</td>
<td>TxD</td>
<td>you've finished talking and are ready to listen or to hang up</td>
</tr>
<tr>
<td>not Clear To Send</td>
<td>CTS</td>
<td>the phone has sent your words and is ready for your next request to send a message</td>
</tr>
<tr>
<td>not Data Terminal Rdy</td>
<td>DTR</td>
<td>you hang up</td>
</tr>
</tbody>
</table>

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).

<table>
<thead>
<tr>
<th>Signal Description</th>
<th>Abbrev.</th>
<th>Similar to</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ring Indicator</td>
<td>RI</td>
<td>the phone rings (optional)</td>
</tr>
<tr>
<td>Data Set Ready</td>
<td>DSR</td>
<td>you pick up the phone; it works</td>
</tr>
<tr>
<td>Data Carrier Detect</td>
<td>DCD</td>
<td>you hear background noise</td>
</tr>
<tr>
<td>Receive Data</td>
<td>RxD</td>
<td>you hear what is said</td>
</tr>
<tr>
<td>not Data Set Ready</td>
<td>DSR</td>
<td>the other party has hung up</td>
</tr>
</tbody>
</table>

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

Modems

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

What is a modem? The name is short for Modulator/DeModulator. As a modulator it takes electrical signals from a computer or printer (or other device) that it is connected to, and turns them into musical tones over a telephone line. As a demodulator it takes the musical tones it detects on a telephone line and turns them back into electrical signals for use by the printer or computer (or other device) that it is connected to. It also handles the RS-232-C control signals to and from that device (Figure 4-5).
By convention, the calling ( originate) modem produces a fairly high tone (let's say 1 kHz) as the background or carrier signal that it sends; it then modulates (changes) that tone to 0 to 10 mV to mean 0 and 1 to mean 1. Meanwhile, the called (answer) modem plays a lower tone, HI, as a carrier signal, and modulates that tone to RE to indicate 0 or FA to indicate 1. 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" (transmit register) of one DTE gets routed to the "earpiece" (receive register) of the other. A null modem simply crosses those two wires (Figure 4-6).

To simulate the opposite signal exchanges that modems would perform, the null modem interconnected 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; CTS 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 4-8 illustrates the chief configurations possible with the Super Serial Card and how to set them up.

<table>
<thead>
<tr>
<th>Normal</th>
<th>( \text{input device} )</th>
<th>( \text{output device} )</th>
<th>both input &amp; output device</th>
<th>( \text{CPU} )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>keyboard, etc.</td>
<td>printer, plotter, typesetter, etc.</td>
<td>terminal, another Apple, time-sharing service, etc.</td>
<td>Apple Monitor, BASIC Monitor, Pascal Operating System, etc.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PR#s</th>
<th>input device</th>
<th>slot s</th>
<th>output device</th>
<th>( \text{CPU} )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Apple II Keyboard</td>
<td>SSC</td>
<td></td>
<td>Apple II screen</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>IN#s</th>
<th>input device</th>
<th>slot s</th>
<th>output device</th>
<th>( \text{CPU} )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Apple II Keyboard</td>
<td>SSC</td>
<td></td>
<td>Apple II screen</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PR#s and IN#s</th>
<th>input device</th>
<th>slot s</th>
<th>output device</th>
<th>( \text{CPU} )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Apple II Keyboard</td>
<td>SSC</td>
<td></td>
<td>Apple II screen</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Remote input/output (M=Modem)</th>
<th>input device</th>
<th>slot s</th>
<th>output device</th>
<th>( \text{CPU} )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Apple II Keyboard</td>
<td>M</td>
<td>M</td>
<td>Apple II screen</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Terminal Mode</th>
<th>input device</th>
<th>slot s</th>
<th>output device</th>
<th>( \text{CPU} )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Apple II Keyboard</td>
<td>SSC</td>
<td></td>
<td>Apple II screen</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Terminal Mode, chain to slot x</th>
<th>input device</th>
<th>slot s</th>
<th>output device</th>
<th>( \text{CPU} )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Apple II Keyboard</td>
<td>SSC</td>
<td></td>
<td>Apple II screen</td>
</tr>
</tbody>
</table>

---

**THEORY OF OPERATION**

This section explains the SSC's overall theory of operation, but not the internal workings of each IC chip. If you would like such information, it is best to obtain specifications from the IC manufacturers. The most complex component is the ACIA, which is a Synertek 6531 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 10-pin header (7B) to connect the SSC to a DB-25 jack via a short internal cable (discussed in Appendix C)

*Figure 4-9. Overall Block Diagram of the SSC*
ADDRESSING AND CONTROL LOGIC

The twelve address lines (A0 - A11) from the Apple II provide all the necessary $C000$ addressing on the SSC. Control logic at 1B, 1C, 2C and 3C, plus the signals RESET, DEVICE SELECT, I/O SELECT, and I/O STROBE, ensure the routing of signals to the appropriate addresses.

The SSC follows the Apple II protocol in its use of the $C800$ address space. An LS279 (1B) serves as a NAND gate, a pair of inverters, and a set-reset latch. The latch is set by access to the $C8xx$ space, and is reset by access to the $CPxx$ 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 $C800$ - $CFFF$ address space. However, an LS00 (2C) and an LS32 (3C) remap the addresses from the range $C900$ - $C9FF$ to the range $CP00$ - $CPFF$, since the $CPxx$ addresses are unusable. (Access to them disables use of the $C800$ 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, 20 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 $0$ volts. The remaining seven blocks are in the I/O Expansion ROM Space, selected when I/O STROBE from the Apple II drops to $0$ volts.

<table>
<thead>
<tr>
<th>SSC ROM/RAM Addresses</th>
<th>Become Apple II Addresses</th>
</tr>
</thead>
<tbody>
<tr>
<td>$0700 - 07FF$</td>
<td>$C900 - C9FF$</td>
</tr>
<tr>
<td>$0800 - 09FF$</td>
<td>$C900 - C9FF$</td>
</tr>
<tr>
<td>$1000 - 11FF$</td>
<td>$C900 - C9FF$</td>
</tr>
<tr>
<td>$2000 - 22FF$</td>
<td>$2000 - 22FF$</td>
</tr>
<tr>
<td>$3000 - 33FF$</td>
<td>$3000 - 33FF$</td>
</tr>
<tr>
<td>$4000 - 44FF$</td>
<td>$4000 - 44FF$</td>
</tr>
<tr>
<td>$5000 - 55FF$</td>
<td>$5000 - 55FF$</td>
</tr>
<tr>
<td>$6000 - 66FF$</td>
<td>$6000 - 66FF$</td>
</tr>
</tbody>
</table>

Table 4-4. SSC Address Remapping

Registers in Peripheral I/O Space

Whenever DEVICE SELECT drops to $0$ volts, the Apple II is addressing the SSC's Peripheral I/O Space (the sixteen bytes starting at $C000 + s0$). This signal is combined logically with address lines A8 through A3 to select one of the six registers that reside in that space (Table 4-5).

<table>
<thead>
<tr>
<th>Chip selected</th>
<th>Address (+s0)</th>
<th>Purpose of register</th>
</tr>
</thead>
<tbody>
<tr>
<td>LS365 (2B)</td>
<td>$C081$</td>
<td>store state of SW1 (1A) (read)</td>
</tr>
<tr>
<td>LS365 (3B)</td>
<td>$C082$</td>
<td>store state of SW2 (2A) (read)</td>
</tr>
<tr>
<td>ACTA (4-5A)</td>
<td>$C088$</td>
<td>receive (read), transmit (write)</td>
</tr>
<tr>
<td>ACTA (4-5A)</td>
<td>$C089$</td>
<td>status (read), reset (write)</td>
</tr>
<tr>
<td>ACTA (4-5A)</td>
<td>$C08A$</td>
<td>command (read and write)</td>
</tr>
<tr>
<td>ACTA (4-5A)</td>
<td>$C08B$</td>
<td>control (read and write)</td>
</tr>
</tbody>
</table>

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 RD-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 ACTA to the Apple II. (The Apple II, however, currently does not support interrupts.) Setting switches SW1-7 ON and SW2-7 OFF connect D8-25 pin 8 (DCD) to the DCD input of the ACTA. Setting SW1-7 OFF and SW2-7 ON splices pin 19, Secondary Clear To Send (SCCTS), onto the DCD input of the ACTA when the jumper block is in the TERMINAL position.

The ACTA has two pins used to select one of its four registers. While address lines A2 and A3 select the chip, A0 and A1 select the actual register. The SSC firmware reads and writes ACTA register contents; these registers are discussed in detail in Appendix A.

THE ACIA

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

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

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

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

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

APPENDIX A
FIRMWARE

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

PASCAL 1.1 FIRMWARE PROTOCOL
The old Apple II Serial Interface Card (SIC) ran under Pascal 1.0 with three direct firmware entry points, one for each of the three I/O functions it supported:

<table>
<thead>
<tr>
<th>Address</th>
<th>Contains</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C890</td>
<td>initialization routine entry point</td>
</tr>
<tr>
<td>$C84D</td>
<td>read routine entry point</td>
</tr>
<tr>
<td>$C9AA</td>
<td>write routine entry point</td>
</tr>
</tbody>
</table>

New peripheral cards can be "accepted" into the Pascal 1.0 system by appearing to be a SIC; that is, with the same three entry points and with $38 at $C8B3 and $18 at $C8B7 (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 $Cs\&0 address space (a being the slot number where the peripheral card is installed). This space was chosen instead of the $Cs\$0 space, since under BASIC protocol the $Cs\&0 space is required, while the $Cs\$0 space is optional.

The branch table locations that Pascal 1.1 uses are:

<table>
<thead>
<tr>
<th>Address</th>
<th>Contains</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Cs&amp;0</td>
<td>initialization routine offset (required)</td>
</tr>
<tr>
<td>$Cs&amp;E</td>
<td>read routine offset (required)</td>
</tr>
<tr>
<td>$Cs&amp;F</td>
<td>write routine offset (required)</td>
</tr>
<tr>
<td>$Cs11</td>
<td>status routine offset (required)</td>
</tr>
<tr>
<td>$Cs10</td>
<td>$00 if optional offsets follow; non-zero if not</td>
</tr>
<tr>
<td>$Cs12</td>
<td>control routine offset (optional)</td>
</tr>
<tr>
<td>$Cs13</td>
<td>interrupt handling routine offset (optional)</td>
</tr>
</tbody>
</table>

Notice that $Cs11 contains $00 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 $Cs11 contains a (non-zero) firmware instruction.) Apple II Pascal 1.0 and 1.1 do not support control and interrupt requests, but such requests may be implemented in future versions of the Pascal B10S and other future Apple II operating systems.

Here are the entry point addresses, and the contents of the 6582 registers on entry to and on exit from Pascal 1.1 I/O routines:

<table>
<thead>
<tr>
<th>Addr.</th>
<th>Offset for</th>
<th>X Register</th>
<th>Y Register</th>
<th>A Register</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Cs&amp;0</td>
<td>Initialization</td>
<td>$Cs</td>
<td>$00</td>
<td>(unchanged)</td>
</tr>
<tr>
<td></td>
<td>On entry</td>
<td>$Cs</td>
<td>$00</td>
<td>(unchanged)</td>
</tr>
<tr>
<td></td>
<td>On exit</td>
<td>$Cs</td>
<td>$00</td>
<td>(unchanged)</td>
</tr>
<tr>
<td>$Cs&amp;E</td>
<td>Read</td>
<td>$Cs</td>
<td>$00</td>
<td>(unchanged)</td>
</tr>
<tr>
<td></td>
<td>On entry</td>
<td>$Cs</td>
<td>$00</td>
<td>(unchanged)</td>
</tr>
<tr>
<td></td>
<td>On exit</td>
<td>$Cs</td>
<td>$00</td>
<td>(unchanged)</td>
</tr>
<tr>
<td>$Cs&amp;F</td>
<td>Write</td>
<td>$Cs</td>
<td>$00</td>
<td>(unchanged)</td>
</tr>
<tr>
<td></td>
<td>On entry</td>
<td>$Cs</td>
<td>$00</td>
<td>(unchanged)</td>
</tr>
<tr>
<td></td>
<td>On exit</td>
<td>$Cs</td>
<td>$00</td>
<td>(unchanged)</td>
</tr>
<tr>
<td>$Cs11</td>
<td>Status</td>
<td>$Cs</td>
<td>$00</td>
<td>request (0 or 1)</td>
</tr>
<tr>
<td></td>
<td>On entry</td>
<td>$Cs</td>
<td>$00</td>
<td>(changed)</td>
</tr>
<tr>
<td></td>
<td>On exit</td>
<td>$Cs</td>
<td>$00</td>
<td>(changed)</td>
</tr>
</tbody>
</table>

Notes:
- Request code 0 means, "Are you ready to accept output?"
- Request code 1 means, "Do you have input ready?"
- On exit, the reply to the status request is in the carry bit: carry clear means "No"; carry set means "Yes."

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\&0 ROM space. The identifiers are listed in Table A-2.

<table>
<thead>
<tr>
<th>Address</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Cs&amp;5</td>
<td>$38 (like the old Serial Interface Card)</td>
</tr>
<tr>
<td>$Cs&amp;7</td>
<td>$18 (like the old Serial Interface Card)</td>
</tr>
<tr>
<td>$Cs&amp;8</td>
<td>$81 (the Generic Signature of new FW cards)</td>
</tr>
<tr>
<td>$Cs&amp;C</td>
<td>$01 (the Device Signature; see below)</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Digit</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>$0</td>
<td>reserved</td>
</tr>
<tr>
<td>$1</td>
<td>printer</td>
</tr>
<tr>
<td>$2</td>
<td>joystick or other X-Y input device</td>
</tr>
<tr>
<td>$3</td>
<td>serial or parallel I/O card</td>
</tr>
<tr>
<td>$4</td>
<td>modem</td>
</tr>
<tr>
<td>$5</td>
<td>sound or speech device</td>
</tr>
<tr>
<td>$6</td>
<td>clock</td>
</tr>
<tr>
<td>$7</td>
<td>mass storage device</td>
</tr>
<tr>
<td>$8</td>
<td>$0-column card</td>
</tr>
<tr>
<td>$9</td>
<td>network or bus interface</td>
</tr>
<tr>
<td>$A</td>
<td>special purpose (none of the above)</td>
</tr>
<tr>
<td>$B-F</td>
<td>reserved for future expansion</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Addresses</th>
<th>Name of area</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>$0000-$00FF</td>
<td>Page Zero</td>
<td>Monitor pointers, I/O hooks, and temporary storage (Table A-5)</td>
</tr>
<tr>
<td>$0400-$07xx</td>
<td>Peripheral Slot</td>
<td>Locations (8 per slot) in Apple's Scratchpad RAM (Table A-6)</td>
</tr>
<tr>
<td></td>
<td>(selected locations)</td>
<td></td>
</tr>
<tr>
<td>$0C00-$0CFF</td>
<td>Peripheral Card</td>
<td>Locations (16 per slot) for general I/O; SSC uses 6 bytes (Table A-7)</td>
</tr>
<tr>
<td>$0C00-$0CF</td>
<td>I/O Space</td>
<td>I/O; SSC uses 6 bytes (Table A-7)</td>
</tr>
<tr>
<td>$CS00-$CFFF</td>
<td>Peripheral Card</td>
<td>One 256-byte page reserved for card in slot 3, first page of SSC FW</td>
</tr>
<tr>
<td></td>
<td>ROM Space</td>
<td></td>
</tr>
<tr>
<td>$CB00-$CFFF</td>
<td>Expansion ROM</td>
<td>Eight 256-byte pages reserved for a 2K ROM or FRM; SSC maps its FW onto $CB00-$CFFF (Table A-4)</td>
</tr>
</tbody>
</table>

Table A-4. Memory Usage Map

### Zero Page Locations

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

<table>
<thead>
<tr>
<th>Address</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>* $24</td>
<td>CH</td>
<td>Monitor pointer to current position of cursor on screen</td>
</tr>
<tr>
<td>$26</td>
<td>SLOT16</td>
<td>Usually (slot# x 16); that is, $60</td>
</tr>
<tr>
<td>$27</td>
<td>CHARACTER</td>
<td>Input or output character</td>
</tr>
<tr>
<td>* $28</td>
<td>BASL</td>
<td>Monitor pointer to current screen line</td>
</tr>
<tr>
<td>$2A</td>
<td>ZPTMP1</td>
<td>Temporary storage (various uses)</td>
</tr>
<tr>
<td>$2B</td>
<td>ZPTMP2</td>
<td>Temporary storage (various uses)</td>
</tr>
<tr>
<td>$35</td>
<td>ZPTMP</td>
<td>Temporary storage (various uses)</td>
</tr>
<tr>
<td>* $36</td>
<td>CSW1</td>
<td>BASIC output hook (not for Pascal)</td>
</tr>
<tr>
<td>* $37</td>
<td>CSW2</td>
<td>(high byte of CSW)</td>
</tr>
<tr>
<td>* $38</td>
<td>KSW</td>
<td>BASIC input hook (not for Pascal)</td>
</tr>
<tr>
<td>* $39</td>
<td>KSW2</td>
<td>(high byte of KSW)</td>
</tr>
<tr>
<td>* $4E</td>
<td>RNDL</td>
<td>random number location, updated when looking for a keypress (not used when initialized by Pascal)</td>
</tr>
</tbody>
</table>

* 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.

<table>
<thead>
<tr>
<th>Address</th>
<th>Field name</th>
<th>Bit(s)</th>
<th>Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$0478+08</td>
<td>DELAYLG</td>
<td>0 - 1</td>
<td>&lt;F key&gt; delay selection&lt;br&gt;2 - 3 &lt;L key&gt; delay selection&lt;br&gt;4 - 5 &lt;CR&gt; delay selection&lt;br&gt;6 - 7 Translate option</td>
</tr>
<tr>
<td>$04F8+08</td>
<td>HANDSHAKE</td>
<td>0 - 7</td>
<td>Buffer count for handshake (PRA Mode)&lt;br&gt;Accumulator for FW's command processor</td>
</tr>
<tr>
<td>$0578+08</td>
<td>STATELG</td>
<td>0 - 2</td>
<td>Command mode when not 0 (Printer and Communications Modes only)&lt;br&gt;0 - 4 Enquire character (PRA Mode); dfl ETX&lt;br&gt;3 - 5 Slot to chain to (Communications Mode) Set to 1 after lowercase input character 7 Terminal Mode when 1 (Comm Mode) Enable &lt;CR&gt; gen. when 1 (other 3 modes)</td>
</tr>
<tr>
<td>$05F8+08</td>
<td>CMDBYTE</td>
<td>0 - 6</td>
<td>Printer Mode default is &lt;CTRL-D&gt;&lt;Comm Mode default is &lt;CTRL-A&gt;&lt;Set to 1 to Zap control commands</td>
</tr>
<tr>
<td>$0678+08</td>
<td>STSBYTE</td>
<td>Status and IRESULT byte (Appendix F)</td>
<td></td>
</tr>
<tr>
<td>$06F8+08</td>
<td>CHNBYTE</td>
<td>0 - 2</td>
<td>Current Apple screen slot (Comm Mode); when slot = 0, chaining is enabled 3 - 7 $Ch00 space entry point (Comm Mode)</td>
</tr>
<tr>
<td>$0778+08</td>
<td>PWDBYTE</td>
<td>0 - 7</td>
<td>Current printer width (other modes); for listing compensation, auto&lt;CR&gt;</td>
</tr>
<tr>
<td>$07F8+08</td>
<td>BUFBYTE</td>
<td>0 - 6</td>
<td>One-byte input buffer (Comm Mode); used in conjunction with XOFF recognition 7 Set to 1 when buffer full (Comm Mode)</td>
</tr>
<tr>
<td>$08F8+08</td>
<td>COLBYTE</td>
<td>0 - 7</td>
<td>Current-column counter for tabbing, etc. (other 3 modes)</td>
</tr>
<tr>
<td>$0F78+08</td>
<td>MISCFLG</td>
<td>0 - 10</td>
<td>Generate &lt;LF&gt; after &lt;CR&gt; when 1 &lt;Printer Mode when 0; Comm Mode when 1 Keyboard input enabled when 1 3 &lt;CTRL-S&gt; (XOFF), &lt;CTRL-R&gt; and &lt;CTRL-T&gt; input checking when 1 4 Pascal Op Sys when 1; BASIC when 0 5 Discard &lt;LF&gt; input when 1 6 Enable lowercase and special character generation when 1 (Comm Mode) 7 Tabbing option on when 1 (Printer Mode) 7 Echo output to Apple screen when 1</td>
</tr>
</tbody>
</table>

Table A-6. Scratchpad RAM Locations Used by SSC
PERIPHERAL CARD I/O SPACE

There are 16 bytes of I/O space allocated to each slot in the Apple II. Each set begins at address $C880 + (slot x 16); for example, if the SSC is in slot 3, its group of bytes extends from $C880 to $C8BF. Table A-7 interprets the 6 bytes the SSC uses.

<table>
<thead>
<tr>
<th>Address</th>
<th>Register</th>
<th>Bit(s)</th>
<th>Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C881+00</td>
<td>DIPSM1</td>
<td>0</td>
<td>SWI-6 is OFF when 1, ON when 0</td>
</tr>
<tr>
<td></td>
<td>(SW1-x)</td>
<td>1</td>
<td>SWI-5 is OFF when 1, ON when 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4-7</td>
<td>same as above for SW1-4 through SW1-1</td>
</tr>
<tr>
<td>$C882+00</td>
<td>DIPSM2</td>
<td>0</td>
<td>Clear To Send (CTS) is true (+) when 0</td>
</tr>
<tr>
<td></td>
<td>(SW2-x)</td>
<td>1-3</td>
<td>same as above for SW2-5 through SW2-3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5-7</td>
<td>same as above for SW2-2 &amp; SW1-2</td>
</tr>
<tr>
<td>$C888+00</td>
<td>TDREG</td>
<td>0-7</td>
<td>ACIA Transmit Register (write)</td>
</tr>
<tr>
<td>$C889+00</td>
<td>RDREG</td>
<td>0-7</td>
<td>ACIA Receive Register (read)</td>
</tr>
<tr>
<td>$C88A+00</td>
<td>STATUS</td>
<td>0</td>
<td>Parity error detected when 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>Framing error detected when 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>Overrun detected when 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
<td>ACIA Receive Register full when 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
<td>ACIA Transmit Register empty when 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5</td>
<td>Data Carrier Detect (DCD) true when 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6</td>
<td>Data Set Ready (DSR) true when 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>7</td>
<td>Interrupt (IRQ) has occurred when 1</td>
</tr>
<tr>
<td>$C88B+00</td>
<td>COMMAND</td>
<td>0</td>
<td>ACIA Command Register (read/write)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>Data Terminal Ready (DTR): enable (1) or disable (0) receiver and all interrupts</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2-3</td>
<td>Control transmit interrupt, Request To Send (RTS) level, and transmitter</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
<td>When 0, normal mode for receiver; when 1, echo mode (but bits 2 and 3 must be 0)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5-7</td>
<td>Control parity: values: Table 2-7</td>
</tr>
<tr>
<td>$C88C+00</td>
<td>CONTROL</td>
<td>0-3</td>
<td>Baud rate: $B = 16 times external clock; $1 - $F = decimal in Table 2-5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
<td>When 1, use baud rate generator; when 0, use external clock (not supported)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5-6</td>
<td>Number of data bits: 8 (bit 5 and 6 = 0), 6 (5 = 1, 6 = 0), 4 (5 = 0, 6 = 1) or 5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>7</td>
<td>(bit 5 and 6 both = 1)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Number of stop bits: 1 (bit 7 = 0); if bit 7 = 1, then 1-1/2 (with 5 data bits, no parity), 1 (8 data plus parity) or 2</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>Addr.</th>
<th>Routine</th>
<th>X Register</th>
<th>Y Register</th>
<th>A Register</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C800</td>
<td>Initialization</td>
<td>$Cs</td>
<td>$s0</td>
<td>anything</td>
</tr>
<tr>
<td>On entry</td>
<td></td>
<td>$Cs</td>
<td>$s0</td>
<td>(unchanged)</td>
</tr>
</tbody>
</table>

Notes: $C800 space is enabled. Firmware initializes SSC to default values plus SW1 and SW2 selections.

<table>
<thead>
<tr>
<th>Addr.</th>
<th>Routine</th>
<th>X Register</th>
<th>Y Register</th>
<th>A Register</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C805</td>
<td>Read</td>
<td>$Cs</td>
<td>$s0</td>
<td>anything</td>
</tr>
<tr>
<td>On entry</td>
<td></td>
<td>$Cs</td>
<td>$s0</td>
<td></td>
</tr>
<tr>
<td>On exit</td>
<td></td>
<td>$Cs</td>
<td>$s0</td>
<td></td>
</tr>
</tbody>
</table>

Notes: $C805 space is enabled. Character in from ACIA or keyboard is returned in the A Register.

<table>
<thead>
<tr>
<th>Addr.</th>
<th>Routine</th>
<th>X Register</th>
<th>Y Register</th>
<th>A Register</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C80F</td>
<td>Write</td>
<td>$Cs</td>
<td>$s0</td>
<td>char. out</td>
</tr>
<tr>
<td>On entry</td>
<td></td>
<td>$Cs</td>
<td>$s0</td>
<td></td>
</tr>
<tr>
<td>On exit</td>
<td></td>
<td>$Cs</td>
<td>$s0</td>
<td>(changed)</td>
</tr>
</tbody>
</table>

Notes: $C80F space is enabled. The byte in the A Register is sent out through the ACIA.

<table>
<thead>
<tr>
<th>Addr.</th>
<th>Routine</th>
<th>X Register</th>
<th>Y Register</th>
<th>A Register</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C9A</td>
<td>Write</td>
<td>$Cs</td>
<td>$s0</td>
<td>character out</td>
</tr>
<tr>
<td>On entry</td>
<td></td>
<td>$Cs</td>
<td>$s0</td>
<td>(changed)</td>
</tr>
</tbody>
</table>

Notes: $C9A space is enabled. Output character is transmitted through the ACIA. Pascal posts error code to 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 1.0, Pascal 1.1 enters these routines using indirect addressing.

<table>
<thead>
<tr>
<th>Addr.</th>
<th>Offset for</th>
<th>Value</th>
<th>X Register</th>
<th>Y Register</th>
<th>A Register</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C90D</td>
<td>Initialization</td>
<td>$Cs0E</td>
<td>$Cs</td>
<td>$s0</td>
<td>anything</td>
</tr>
<tr>
<td>On entry</td>
<td></td>
<td>$Cs</td>
<td>$s0</td>
<td>(changed)</td>
<td></td>
</tr>
</tbody>
</table>

Notes: $C90D space is enabled. Firmware initializes SSC to default values plus SW1 and SW2 selections.

<table>
<thead>
<tr>
<th>Addr.</th>
<th>Offset for</th>
<th>Value</th>
<th>X Register</th>
<th>Y Register</th>
<th>A Register</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C90E</td>
<td>Read</td>
<td>$Cs94</td>
<td>$Cs</td>
<td>$s0</td>
<td>anything</td>
</tr>
<tr>
<td>On entry</td>
<td></td>
<td>$Cs</td>
<td>$s0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>On exit</td>
<td></td>
<td>error code</td>
<td>$Cs</td>
<td>char. in</td>
<td></td>
</tr>
</tbody>
</table>

Notes: $C90E space is enabled. Character in from ACIA or keyboard is returned in the A Register.

<table>
<thead>
<tr>
<th>Addr.</th>
<th>Offset for</th>
<th>Value</th>
<th>X Register</th>
<th>Y Register</th>
<th>A Register</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C90F</td>
<td>Write</td>
<td>$Cs97</td>
<td>$Cs</td>
<td>$s0</td>
<td>char. out</td>
</tr>
<tr>
<td>On entry</td>
<td></td>
<td>$Cs</td>
<td>$s0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>On exit</td>
<td></td>
<td>error code</td>
<td>$Cs</td>
<td>(changed)</td>
<td></td>
</tr>
</tbody>
</table>

Notes: $C90F space is enabled. The byte in the A Register is sent out through the ACIA.

<table>
<thead>
<tr>
<th>Addr.</th>
<th>Offset for</th>
<th>Value</th>
<th>X Register</th>
<th>Y Register</th>
<th>A Register</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C910</td>
<td>Status</td>
<td>$Cs9A</td>
<td>$Cs</td>
<td>$s0</td>
<td>request (0 or 1)</td>
</tr>
<tr>
<td>On entry</td>
<td></td>
<td>$Cs</td>
<td>$s0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>On exit</td>
<td></td>
<td>error code</td>
<td>$s0</td>
<td>error code</td>
<td></td>
</tr>
</tbody>
</table>

Notes: $C910 space is enabled. Request = 0 asks ACIA whether it is ready to transmit another byte; request = 1 asks ACIA whether it has an input character available. On exit, carry bit = 0 for Yes or 1 for No.

Table A-11. Pascal 1.1 Offsets Used by SSC

OTHER SPECIAL FIRMWARE LOCATIONS

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

<table>
<thead>
<tr>
<th>Address</th>
<th>Value</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>$CA05</td>
<td>$38</td>
<td>Pascal serial/firmware card identifier (as well as BASIC input entry point)</td>
</tr>
<tr>
<td>$CA07</td>
<td>$18</td>
<td>Pascal serial/firmware card identifier (as well as BASIC output entry point)</td>
</tr>
<tr>
<td>$CA08</td>
<td>$21</td>
<td>Pascal 1.1 generic signature byte ($21 = firmware card)</td>
</tr>
<tr>
<td>$CA0C</td>
<td>$31</td>
<td>Pascal 1.1 Device Signature byte ($31 = serial or parallel I/O card #1)</td>
</tr>
<tr>
<td>$CA11</td>
<td>$85</td>
<td>Pascal 1.1 optional routines flag (nonzero value = not supported)</td>
</tr>
<tr>
<td>$CAFF</td>
<td>$08</td>
<td>Firmware revision level</td>
</tr>
</tbody>
</table>

Table A-12. SSC Special Firmware Locations
### SSC Firmware Listings

<table>
<thead>
<tr>
<th>Address</th>
<th>Instruction</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0001</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>0002</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>0003</td>
<td>4</td>
<td>Apple II SSC Firmware</td>
</tr>
<tr>
<td>0004</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>0005</td>
<td>6</td>
<td>By Larry Kenyon</td>
</tr>
<tr>
<td>0006</td>
<td>7</td>
<td>January 1981</td>
</tr>
<tr>
<td>0007</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>0008</td>
<td>9</td>
<td>(C) Copyright 1981 by Apple Computer, Inc.</td>
</tr>
<tr>
<td>0009</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>0010</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>0011</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>0012</td>
<td>13</td>
<td>Variable Definitions</td>
</tr>
<tr>
<td>0013</td>
<td>14</td>
<td></td>
</tr>
<tr>
<td>0014</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>0015</td>
<td>16</td>
<td></td>
</tr>
<tr>
<td>0016</td>
<td>17</td>
<td>Zero Page Equations</td>
</tr>
<tr>
<td>0017</td>
<td>18</td>
<td></td>
</tr>
<tr>
<td>0018</td>
<td>19</td>
<td>Ch EQU $24</td>
</tr>
<tr>
<td>0019</td>
<td>20</td>
<td>Slot EQU $26</td>
</tr>
<tr>
<td>0020</td>
<td>21</td>
<td>Character EQU $27</td>
</tr>
<tr>
<td>0021</td>
<td>22</td>
<td>Basic EQU $28</td>
</tr>
<tr>
<td>0022</td>
<td>23</td>
<td>Eptmp EQU $35</td>
</tr>
<tr>
<td>0023</td>
<td>24</td>
<td>Eptmp1 EQU $2A</td>
</tr>
<tr>
<td>0024</td>
<td>25</td>
<td>Eptmp2 EQU $2B</td>
</tr>
<tr>
<td>0025</td>
<td>26</td>
<td>Cswl EQU $36</td>
</tr>
<tr>
<td>0026</td>
<td>27</td>
<td>Cswh EQU $37</td>
</tr>
<tr>
<td>0027</td>
<td>28</td>
<td>Kiwl EQU $38</td>
</tr>
<tr>
<td>0028</td>
<td>29</td>
<td>Kwh EQU $39</td>
</tr>
<tr>
<td>0029</td>
<td>30</td>
<td>All EQU $3C</td>
</tr>
<tr>
<td>002A</td>
<td>31</td>
<td>Random Number Seed</td>
</tr>
<tr>
<td>002B</td>
<td>32</td>
<td>Randh EQU $4F</td>
</tr>
<tr>
<td>002C</td>
<td>33</td>
<td></td>
</tr>
<tr>
<td>002D</td>
<td>34</td>
<td>General Equations</td>
</tr>
<tr>
<td>002E</td>
<td>35</td>
<td></td>
</tr>
<tr>
<td>002F</td>
<td>36</td>
<td>Stack EQU $100</td>
</tr>
<tr>
<td>0030</td>
<td>37</td>
<td>Inbbuf EQU $200</td>
</tr>
<tr>
<td>0031</td>
<td>38</td>
<td>Kbd EQU $CO00</td>
</tr>
<tr>
<td>0032</td>
<td>39</td>
<td>Kbdstrb EQU $CO10</td>
</tr>
<tr>
<td>0033</td>
<td>40</td>
<td>Romoff EQU $CFFC</td>
</tr>
<tr>
<td>0034</td>
<td>41</td>
<td></td>
</tr>
<tr>
<td>0035</td>
<td>42</td>
<td>SSC Card Addresses</td>
</tr>
<tr>
<td>0036</td>
<td>43</td>
<td></td>
</tr>
</tbody>
</table>

---

**Notes:**
- Addressing: 0000 to 003F.
- Instructions are in hexadecimal.
- Equations and constants are provided for various parts of the firmware.
- This firmware listing is for the SSC (Super Serial Card) device.
CA07:89 89 CO 60  LDA STRREG,Y
CA0A:29 70 61  AND #70
CA0C:C9 10 62  CMP $10 ;RQU IF TDR EMPTY, DCD, DSR, & CTS
CA0E:60 63  RTS
CA0F:  64 *
CA10:65 65  *****************************************************
CA11:66 * GENERAL INPUT ROUTINE *
CA12:67 67  INPUT JSR SRLN
CA13:68 68 INPUT JSR SRLN
CA14:69 69 SCC NOPUSH!
CA15:70  70 *
CA16:71 71  LDA RDRREG,Y ;GET THE ACTA INPUT
CA17:72 72  ORA #80 ;SET HI BIT FOR BASIC
CA18:73 73  CMP #$8A ;LINEREAD?
CA19:74 74  BNE INPUT2
CA20:75 *
CA21:76 76  TAX
CA22:77 77  LDA MISCFLG,X ;SEE IF WE SHOULD EAT IT
CA23:78 78  AND #$20
CA24:79 79  BNE MISPUSH. ;IF SO, JUST KEEP IT A SECRET
CA25:80  80  TPA
CA26:81 81 *
CA27:82 82 INPUT2 SEC ;INDICATE DATA
CA28:83 83  RTS
CA29:84 84 *
CA2A:85 85 NOINPUT CLC ;CARRY CLEAR FOR NO INPUT
CA2B:86 86  INPUT1 RTS
CA2C:87 87 *
CA2D:88 88  *****************************************************
CA2E:89 89 * GENERAL OUTPUT ROUTINE *
CA2F:90 90 *
CA30:91 91 *
CA31:92 92 * START OF COMMAND CHECK ROUTINE
CA32:93 93 *
CA33:94 94 CMDSCKX LOY SLOT16
CA34:95 95  LDA DISPM1,Y
CA35:96 96  LSR A
CA36:97 97  BCS NOSMD ;DON'T WORRY ABOUT CMD SEQ FOR SIC
CA37:98 98  LDA STATEFLG,X
CA38:99 99  AND #$07 ;ARE WE IN A COMMAND SEQUENCE?
CA39:00 100  BNE ESCCHECK
CA3A:01 101  JSR CMDSPROC ;IF SO, GOTO COMMAND CENTRAL
CA3B:02 102  TPA
CA3C:03 103 RTS
CA3D:04 104 *
CA3E:05 105  ESCCHECK LDA CHARACTER
CA3F:06 106  AND #$70 ;IGNORE HIGH BIT
CA40:07 107  CMP CMDSFIL,T,X ;IS THIS BEGINNING OF A CMD SEQUENCE?
CA41:08 108  BNE XOFPUK
CA42:09 109  LDA STATEFLG,X ;START UP COMMAND MODES
CA43:0A 110  SEC ;INDUCES COMMAND
CA44:0B 111  RTS
CA45:0C 112 *
CA46:0D 113  XOFPUK LDA MISCFLG,X ;ISE XON ENABLED?
CA47:0E 114  AND #$0A
CA48:0F 115  BNE ESCCHECK ;SKIP THIS IF NOT
CA49:10 116 *
CA4A:11 117  JSR INPUT ;ANY INPUT?
CBF4:6A  234  ROR A
CBF5:90  07  235  SBC OUTPUTEND
CBF7:9A  8A  236  LDA #58A
CBF9:85  27  237  STA CHARACTER ;LINE FEED
CBFB:4C  6B C8  238  JMP OUTPUT2 ;(DON'T ECHO IT)
CBFB:60  239  OUTPUTEND RTS
CBFF:7  240  *
CBFF:801  241  DLVTBL DEB $001 ;32 MSE
CC00:08  242  DFB $08 ;1/4 SEC
CC01:40  243  DFB $40 ; 2 SEC
CC02:  244  ***********************
CC02:  245  * ACIA OUTPUT ROUTINE *
CC03:  246  ***********************
CC02:  247  ACIAOUT JR SHOUT ;READY FOR OUTPUT!
CC05:00  248  BNE ACIACOUNT
CC07:98  249  TYA
CC08:09  250  ORA #$89 ;PREPARE TO ADDRESS ACIA,
CC0A:98  251  TAY ;CAUSING 6502 FALSE READ TO OCCUR
CC0B:52  252  LDA CHARACTER ;ON PAGE SHP (AVOIDING ROR READ)
CC0D:99  253  STA SHFFT,Y ;HERE YOU ARE ACIA
CC10:60  254  RTS
CC11:  255  *
CC12:  256  ***********************
CC11:  257  RESTORE CURSOR (NOT FOR PASCAL) *
CC12:  258  ***********************
CC11:  259  ***********************
CC11:  260  RESTORE PHA ;SAVE NEW CHARACTER
CC12:  261  LDY CH
CC14:05  262  LDA CHARACTER ;OLD CHARACTER
CC16:98  263  STA (MASK),Y
CC18:60  264  PLA
CC19:  265  *
CC1A:9C  266  CMP #$95 ;SCREEN PICK?
CC1B:00  267  BNE RESTORE
CC1C:52  268  LDA CHARACTER ;IF SQ, USE SCREEN CHAR
CC1C:99  269  CMP #$20 ;INVERSE?
CC21:8C  270  BCS RESTORE
CC23:20  271  JSR GETKEY ;REVERSE THE TRANSLATION
CC25:59  272  EORothyA
CC29:57  273  RESTORED STA CHARACTER
CC29:60  274  RTS
CC32:  275  *
CC32:  276  CHN SSC.UTIL
CC3C:  277  ***********************
CC3C:  278  2 *************************
CC3C:  279  3  +APPLE II SSC FIRMWARE *
CC3C:  280  5  *
CC3C:  281  6  + BY LARRY KENTON *
CC3C:  282  7  *
CC3C:  283  8  +JANUARY 1981 +
CC3C:  284  9  *
CC3C:  285  10  + (C) COPYRIGHT 1981 BY APPLE COMPUTER, INC. *
CC3C:  286  11  *
CC3C:  287  12  ***********************
CC3C:  288  13  *
CC3C:  289  14  +UTILITY ROUTINES *
CC3C:  290  15  *
CC3C:  291  16  ***********************
CC3C:  292  17  +PASCAL-BASIC KEYBOARD FETCH *
CC3C:  293  18  ***********************
CC3C:  294  19  CKBXD CLC ;RETURN CARRY CLEAR FOR NO DATA
CC3C:  295  20  LDA MISCLGL,X
CC3C:  296  21  AND #$04 ;ANSWER IF KEYBOARD IS DISABLED
CC3C:  297  22  BNE CKBXDIT
CC3C:  298  23  *
CC3C:  299  24  CKBXK1 LDA KEO
CC3C:  300  25  BPL CKBXDT
CC3C:  301  26  STA KBMTR
CC3C:  302  27  SEC ;INDICATE DATA
CC3C:  303  28  CKBXKIT RTS
CC3C:  304  29  ***********************
CC3C:  305  30  GET A CHAR FROM KEYBOARD FOR BASIC ONLY *
CC3C:  306  31  ***********************
CC3C:  307  32  GETKBD INC NNLL ;MIX UP RANDOM # SEED
CC3C:  308  33  BNE GETKBD1 ;FOR BASIC
CC3C:  309  34  INC NNLL
CC3C:  310  35  GETKBD JSR CKBXD ;KEYBOARD FETCH ROUTINE
CC3C:  311  36  CLI ;INDICATE NO ESCAPE SEQUENCE
CC3C:  312  37  BCC CKBXKEX ;EXIT IF NO KEY PRESS
CC3C:  313  38  JSR RESTORE ;DO BASIC CURSED DUTY
CC3C:  314  39  AND #$7F
CC3C:  40  40  CMP CMBYTE,X ;IS IT THE START OF A COMMAND?
CC3C:  41  41  BNE GETKBD ;IF NOT, EXIT INDICATING DATA
CC3C:  42  42  LDA SLO'T16
CC3C:  43  43  BNE GETKBD ;ONLY DO CMD ESC FOR PCC, SIC MODES
CC3C:  44  44  JR A
CC3C:  45  45  BCS GETKBD0
CC3C:  46  46  ***********************
CC3C:  47  47  +KEYBOARD ESCAPE HANDLER *
CC3C:  48  48  ***********************
CC3C:  49  49  BCS:10A 0A 49 BNE:0E 5A ;FIRST PRINT A PROMPT
CC3C:  50  50  LDA PROMPTIN
CC3C:  51  51  STA CHARACTER
CC3C:  52  52  TYA
CC3C:  53  53  PHA
CC3C:  54  54  JSR SCREENOUT ;ALWAYS SEND TO SCREEN
CC3C:  55  55  PLA
CC3C:  56  56  TAY
CC3C:  57  57  TAY
CC3C:  58  58  BPL PROMPTLOOP
CC3C:  59  59  *
CC60:1A9 01   60  LOA  #1 ;START OUT IN COMMAND STATE
CC62:20 78 CC 61  JSR  SETSTATE
CC61: 62 *
CC62:20 34 CC 63  G3TCMD JSR  C3KRD1 ;WAIT FOR KEYBOARD CHARACTER
CC62:10 FB 64  BFL  G3TCMD
CC71:09 88 65  CMP  #50B ;BACKSPACE
CC79:00 F1 66  BEEQ  KRODSC ;IF SO, THEN START OVER
CC79:00 F7 67  STA  CHARACTER
CC76:95 27 68 *
CC76:91 20 A3 CC 69  JSR  SCREENOUT!
CC80:20 1A C3 70  JSR  CMDSEQCN ;JUMP THRU CMD INTEPRETER
CC81: 71 *
CC82:00 06 04 72  LOA  STATEFLG,Y ;ARE WE DONE?
CC82:29 07 73  AND  #507 ;GET STATEFLG
CC83:00 88 74  BNE  GETCMD ;IF NOT, GO AGAIN
CC84: 75 *
CC84:40 80 76  LOA  #50D ;FORCE BACK A CARRIAGE RETURN
CC85:15 27 77  STA  CHARACTER
CC86:2C 58 FF 78  BIT  IORTS ;INDICATE THAT A CMD SEQ HAS OCCURRED
CC91:38 79  GETKRODSC SEC ;INDICATE SUCCESS
CC92:70 80  RTS
CC93: 81 *
CC93: 82 *
CC93:4A C3 D3 83  PRINTBLS ASC *;CSS  SLPPA*
CC96:13 00 C5 84  DPS  #5B
CC99:CC 00 D0 85 *
CC9C: 86  * ******************************************
CC9C: 87 * ROUTINE TO PRINT A CHARACTER ON THE CURRENT DISPLAY *
CC9C: 88  ******************************************
CC9C:30 38 07 89  SCREENOUT LOA  MISCFLG,X
CCAC:10 13 90  BPL  NOUT ;IF SCREEN DISABLED
CC9A: 91 *
CC9A:BD 38 07 92  SCREENOUT LOA  MISCFLG,X ;ENTRY AFTER ECHO CHECK
CC9E:29 02 93  AND  #502 ;IF IT ISN'T CIC MODE,
CC9E:FO 9D 94  BEEQ  ASCREEN ;ALWAYS USE THE APPLE SCREEN
CC9F:8D 88 80 95  LOA  STATEFLG,X ;CURRENT SCREEN = APPLE SCREEN?
CC9A:29 38 96  AND  #538
CC9C:FO 96 97  BEEQ  ASCREEN ;SLT  = APPLESCREEN
CC9B: 98 *
CC9B:18 8A 99  TXA ;JUMP TO CHAR SPACE
CC9B:48 100  PHA
CC9B:4A AF 101  LOA  #5;SENDCL-1 ;TO VECTOR TO THE PERIPHERAL
CC9C:48 102  PHA ;IN THE CHAIN SLOT
CC9D:60 103 NOUT  RTS
CC9F: 104 *
CC9F: 105 *APPLE 40-COL SCREEN DRIVER
CCB7: 106 *
CCB7:70 00 CC 107  ASCREEN JSR  GETKROD ;SET THE TRANSLATE OPTIONS
CCB8:09 80 108  ORA  #50B ;SET HIGH BIT OF CHAR
CCB9:09 80 109  CMP  #50D ;LOWERCASE
CCB7:90 06 110  BCC  TESTLETTER
CCD0:4F 03 CC 111  BOR  UCMASK,Y ;DO LOWERCASE TRIP
CCD3:4C #6 PD 112  TOSCREEN JMP  VINDOUT ;ALL REGS ARE PRESERVED
CCD6: 113 *
CCD6: 114 * IF UPPERCASE, WE ONLY MAP LETTERS

(listings continued on next page)
CD49:00 116   LDY $51
CD49:03 117   DELATSET AND DELATFLG,X ;DON'T DISTURB THE OTHER FLAGS
CD49:05 118   STA ZPMP1
CD50:00 06 119   LDA PARAMETER,X
CD51:29 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
CD59:00 FC 125   BNE ROTATE ;PREPARE IT TO OR INTO THE FLAGS
CD58:1 126   *
CD58:05 2A 127   ORA ZPMP1
CD59:00 88 128   STA DELATFLG,X
CD60:60 129   RTS
CD61: 130   *
CD61:29 07 131   SLOTCMD AND #$07 ;SET SLOT COMMAND
CD61:0A 132   ASL A
CD64:0A 133   ASL A
CD65:0A 134   ASL A
CD66:85 2A 135   STA ZPMP1
CD66:0A 136   ASL A
CD69:C5 26 137   CMP SLOT16 ;MAKE SURE WE DON'T SET IT
CD6B:00 00 138   BEQ SLOTCMD01 ;TO OUR OWN SLOT
CD6D:00 B0 04 139   LDA STATEFLG,X
CD70:29 C7 140   AND #$07 ;PUT NEW SLOT NUMBER IN RUTS 3-5
CD72:05 2A 141   ORA ZPMP1 ;OF CMBYREET,X
CD74:90 B8 04 142   STA STATEFLG,X
CD77:AF 00 143   STA $00 ;STORE ZERO INTO
CD79:90 38 06 144   STA CMBYTE,X ;SLOT OFFSET (SET TO CMDO ENTRY)
CD7C:16 145   ASL SLOTCMD1
CD7D: 146   *
CD7D:0F 09 147   BAUDCMD AND #$0F ;SET NEW BAUD RATE
CD7F:00 07 148   BNE BAUCMD2
CD81:BE 81 C0 149   BAUCMD1 LDA DISP1,Y ;ZERO PARM = RELOAD FROM SWITCHES
CD84:4A 150   LSR A
CD85:4A 151   LSR A
CD86:4A 152   LSR A
CD88:10 153   LSR A
CD89:09 10 154   BAUDCMD2 ORA $010 ;SET INT. BAUD RATE GENERATOR
CD8A:85 2A 155   STA ZPMP1
CD8C:80 156   LDA #$ED
CD8D:85 2B 157   CTLRESET STA ZPMP2
CD90:9B BB 158   STA CLRREG,Y
CD93:25 2B 159   AND ZPMP2
CD95:05 2A 160   ORA ZPMP1
CD97:99 88 C0 161   STA CLRREG,Y
CD9A:60 162   RTS
CD9A: 163   *
CD9A:BB 164   PARMCMD DEY ;TRICK: SO CLRREG,Y ACTUALLY ADDRESSES THE COMMAND REG
CD9C: 165   *
CD9C: 166   *
CD9D:0A 167   DATACMD ASL A ;SET NRW # OF DATA RITS
CD9D:0A 168   ASL A
CD9D:6A 169   ASL A
CD9F:0A 170   ASL A
CDA0:0A 171   ASL A
CDA1:85 2A 172   DATACMD STA ZPMP1
CDA3:09 1F 173   LDA #$1F
CDA5:00 E7 174   BNE CTLRESET ;ALWAYS
CDA7: 175   *
CDA7:1E B8 04 176   TERMSCMD ASL STAFLG,X ;SET TERMINAL MODE
CDAB:38 177   SEC
CDAD:80 10 178   BCS (CMD1) ;ALWAYS
CD4D: 179   *
CD4D:99 89 C0 180   RESETCMD STA RESET,Y ;DROP RTS, DTR
CD80:20 93 F8 181   JSK SETSCR ;PB#0
CD83:29 89 FE 182   JSK SETWDO ;PB#0
CD86:AF 08 183   LDX ISLOT
CD89:1E B8 04 184   QUITCMD ASL STAFLG,X ;CLEAR TERMINAL MODE
CD9C:18 185   CLC
CD9D:7E B8 04 186   QMDO STAFLG,X
CDCC:60 187   RTS
CDCC: 188   *
CDCC:89 6A C0 189   BREAKCMD LDA CMREG,Y ;SEND BREAK SIGNAL
CDC1:48 190   PHA ;FOR 233 MILISECONDS
CDC5:09 0C 191   ORA #$00
CDCC:19 9C 192   STA CMREG,Y
CDCA:A9 E9 193   LDA #$233 ;DELAY FOR 234 MICROSEC.
CDCC:20 C4 CA 194   JSK WAITMS
CDCC:68 195   PLA ;RESTORE OLD COMMAND REG CONTENTS
CDD0:9A 8C 196   STA CMREG,Y
CDD3:60 197   RTS
CD40: 198   *
CD40:28 199   ICMD LDA #$528
CD69:3D 38 06 200   STA PDVALUE,X ;SET PRINTER WIDTH TO 40
CD99:69 60 201   LDA #$580
CD88:1D 38 07 202   ORA MISCFLG,X ;SET SCREEN ECHO
CD96:00 05 203   BNE KOMD2 ;ALWAYS
CD9B: 204   *
CD9B:AF FE 205   KOMD LDA #$7FE ;RESET THE LS GENERATE FLAG
CD51:3D 38 07 206   KOMD1 AND MISCFLG,X
CD51:50 38 07 207   KOMD2 STA MISCFLG,X
CDE8:60 208   RTS
CDE9: 209   *
CDEF:C9 28 210   NCMD CMP $40 ;$40
CDEF:90 211   BCC SCMRTS ;IF NOT, JUST EXIT
CDEF:90 38 06 212   STA PDISYR,X ;SET NEI PRINTER WIDTH
CDEF:A9 3F 213   LDA #$3F ;DISABLE SCREEN, SET LISTING MODE
CDEF:00 EE 214   BNE KOMD1 ;ALWAYS
CDEF: 215   *
CDEF:4E 38 05 216   SCMDSL ASL CMDBTY,X ;DISABLE COMMAND RECOGNITION
CDFF:7E 217   SEC
CDFF:7E 38 05 218   ROR CMBYTE,X
CDFF:B0 219   SCMRTS RTS
CDFF:C1 220   *
CDFF:C1 221   ********************************************
CDFF:C1 222   * VECTOR ACCORDING TO COMMAND STATE *
CDFF:C1 223   ********************************************
CDFF:C1 224   CMDCMD TABLE ;A-REG COMMAND STATE
CDFF:D5 27 225   LDA CHARACTER
CDFF:29 7F 226   AND #$7F
CD31: 227   *
CD31:CC 20 228   CMP #$20 ;SKIP SPACES FOR ALL MODES
CD33:0D 09 229   BNE CMDCMD2
CD35:03 230   CFI #$03 ;EXCEPT MODE 3
CD37:00 01 231   BSC CMDCMD2
CE09:60  232 RTS
CE09:69  04  233 CMDPROC1 LDA #24
CE09:6D  00  234 * BNE SETSTATE ;<ALWAYS>
CE10:  235 *
CE10:69 C9  00  236 CMDPROC2 CMP #200 ;CARRIAGE RETURN?
CE11:00  12  237 BNE CMDPROC4
CE11:20  79 CE  238 JSR ZEROSTATE ;ABORT FOR STATES 0-5, EXIT FOR 6,7
CE15:00  07  239 CPY #207 ;IN STATE 7 WE VECTOR TO THE PROC
CE17:00  01  240 BEQ CMDPROC3 ;
CE17:69  00  241 RTS ;OTHERWISE, JUST EXIT
CE1A:  242 *
CE1A:A9 C6  243 CMDPROC3 LDA #20D ;ALL PROCS MUST START IN PAGE 20D
CE1C:48  244 * PMA
CE1D:BD  38 04  245 LDA PARAMETER,X
CE20:48  246 PMA
CE21:44  26  247 LDY SLOT16 ;NEEDED BY BREAK CMD
CE23:60  248 RTS
CE24:  249 *
CE24:85  35  250 CMDPROC4 STA ZPTMP
CE26:A9 CE  251 LDA #2CE ;ALL ROUTINES MUST START
CE28:48  252 PMA ;IN PAGE 0CE
CE29:BD  30 CE  253 LDA STATETR,Y
CE30:48  254 PMA
CE32:AD  35  255 LDA ZPTMP
CE3F:60  256 RTS ;RTS TO COMMAND PROCEDURE
CE50:  257 *
CE50:  258 * NOW THE STATE ROUTINES
CE59:  259 *
CE5A:  260 ********************************************
CE6A:  261 * STATE BRANCH TABLE *
CE69:  262 ********************************************
CE70:AD  263 STATETR DBF >STATETR-1 ;BAD STATE
CE71:37  264 DFB >STATETR-1 ;CMD> SEEN
CE72:61  265 DFB >STATETR-1 ;ACUMULATE PARAMETER
CE73:89  266 DFB >CDONE-1 ;SKIP UNTIL SPACE
CE74:8A  267 DFB >STATETR-1 ;E/D SOMETHING
CE75:AH  268 DFB >STATETR-1 ;ILLEGAL STATE
CE76:DA  269 DFB >CDONE-1 ;SKIP UNTIL CR
CE77:89  270 DFB >CDONE-1 ;SKIP UNTIL CR THEN DO CMD
CE81:  271 ********************************************
CE89:  272 * COMMAND STEPS 1 *
CE8A:  273 ********************************************
CE90:0D  38 05  274 STATE1 CMP CMDBYTE,X ;IS IT <CMD?>
CE90:0D  06  275 BNE CSTATETR
CE90:DE  08 04  276 BNE STATEFLG,X ;GET STATE BACK TO ZERO
CE9A:4C  02 CC  277 JMP ACIOUT ;OUTPUT <CMD> IF SO
CE9A:  278 *
CE9B:C9  30  279 CSTATETR CMP #30 ;>0?
CE9B:90  00  280 BCC CSTATETR
CE9B:7C  3A  281 CMP #3A ;<97
CE9B:90  09  282 BCS CSTATETR
CE9B:29  OF  283 AND #20F ;IT'S A NUMBER
CE9C:00  38 04  284 STA PARAMETER,X
CE9C:20  02  285 LDA #2
CE9C:20  27  286 BNE SETSTATE ;<ALWAYS> SET MODE 2 AND RETURN
CE9C:  287 *
CE9C:49 C9  20  288 CSTATETR CMP #20 ;IS IT A CONTROL CHAR?
CE9E:BD  06  289 BCS CSTATEFLG
CE9F:90  38 05  290 STA CMDBYTE,X ;SET NEW COMMAND CHARACTER
CE9F:4C  79 CE  291 JMP ZEROSTATE ;RESET STATE TO ZERO
CE9F:  292 *
CE9F:A9  00  293 CSTATETR LDX #0 ;USE COMMAND TABLE
CE9F:FD  40  294 BNE CMDSEARCH ;<ALWAYS>
CE9F:FD  00  295 ********************************************
CEA0:21  296 * COMMAND STATE 2: ACCUMULATE PARAMETER *
CEA0:21  297 ********************************************
CEA2:49  30  298 CSTATE2 EOR #20 ;CONVERT #30-239 TO 0-9
CEA2:49  09  299 CMP #$A ;0-9?
CEA2:BD  00  300 BNE CSTATE2A
CEA2:BD  00  301 LDX #$A ;IT'S A NUMBER, 80 ADD
CEA2:7D  38 04  302 ACCLOO KM:ACCU Parameter,Y ;IT TO 10*PARAMETER
CEA2:BB  303 DEY
CEA2:BD  304 BNE ACCLOO
CEA2:9D  38 04  305 BNE CMDSEARCH ;<ALWAYS>
CEA2:FD  15  306 BNE COOME ;<ALWAYS>
CEA5:  307 *
CEA5:40  28  308 CSTATE2A LDX $0400-1 ;CMDCHAR ;USE COMMAND TABLE
CEA5:DD  36  309 BNE CMDSEARCH ;<ALWAYS>
CEA5:FD  00  310 ********************************************
CEA7:  311 * SET COMMAND STATE *
CEA7:  312 ********************************************
CEA7:9D  00  313 ZEROSTATE LDA #0
CEA7:BD  00  314 SETSTATE STA ZPTMP
CEA7:DE  08 07  315 LDY MSLOT
CEA8:BD  08 04  316 LDA STATEFLG,X
CEA8:39 F8  317 AND #$FF
CEA8:05  2A  318 ORA ZPTMP1
CEA8:79  06 04  319 STA STATEFLG,X
CEA9:60  320 DCDN RTS
CEA9:  321 ********************************************
CEAA:  322 * COMMAND STATE 4 (E/D) *
CEAA:  323 ********************************************
CEBB:8A  324 CSTATE4 TAY ;E/D -> Y-RGS
CEBB:BD  38 04  325 LDA PARAMETER,X
CEBB:CD  44  326 CPY #$44 ;(ISABLE?)
CEBB:0F  09  327 BNE CSTATE4A
CEBB:3C  45  328 CPY #$45 ;(ENABLE)
CEBB:5D  11  329 BNE STATER ;IF NOT, IGNORE THIS COMMAND
CEBB:79  38 07  330 ORA MISCFLG,X ;SET FLAG
CEBB:AD  05  331 BNE CSTATE4B ;<ALWAYS>
CEBB:49 FF  332 CSTATE4A BOR #$FF ;INVERT FOR DISABLE
CEBB:3D  38 07  333 AND MISCFLG,X ;RESET FLAG
CEBB:9D  38 07  334 CSTATE4A STA MISCFLG,X
CEBB:  335 ********************************************
CEC4:  336 * ESCAPE TO STATE 6 *
CEC4:  337 ********************************************
CEC4:A9  06  338 SETSTATE6 LDA #6
CEC4:DD  00  339 BNE SETSTATE ;<ALWAYS>
CEC5:A9  20  340 STATEFLG LDA #33 ;CODE FOR RAD COMMAND
CEC5:AD  08 05  341 STA STATEFLG,X
CEC5:DD  05  342 BNE SETSTATE6 ;<ALWAYS>
CEC8:  343 ********************************************
CEC8:  344 * TABLE DRIVEN COMMAND EXECUTOR *
CEC8:  345 ********************************************
CEC8:EF  88 CC  346 CMDSRCH LDA CMCHIL,Y ;GET CANDIDATE CHARACTER
CEC8:FD  04  347 BEQ STATER ;<ZERO MARKS THE END OF A SUBTABLE
### 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 P8A 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 existing 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 (AGIA hardware generates only those 15)
- data formats other than 5-8 data bits and 1, 1-2/2 stop bits (AGIA characteristic; other formats rarely used anyway)

- (ESC) and (ESCO) 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 SIC 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 LF after CR; no translation of lowercase to uppercase; and no 1/4-second delay after CR. The PEEKs and POKEs:

POKE 49339, 243 (49291 + 3*16; 3 + 2*16)
POKE 49338, 187 (49290 + 3*16; p = 187)
POKE 2843, 132 (plug in magic number)
POKE 11477, 64 (plug in magic number)

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

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

Then to set 5 data and 2 stop bits, use <CTRL-L>D<RETURN>; for even parity, use <CTRL-L>D3<RETURN>; to leave lowercase alone, use <CTRL-L>D17<RETURN>. You can use commands to change baud rate, etc.

<table>
<thead>
<tr>
<th>Selection</th>
<th>SSC switches and settings</th>
<th>P8 Serial Card</th>
<th>P8A Serial Card</th>
</tr>
</thead>
<tbody>
<tr>
<td>P8 Mode:</td>
<td>SW1-5 ON,</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SW1-6 OFF</td>
<td></td>
<td></td>
</tr>
<tr>
<td>P8A Mode:</td>
<td>SW1-5 OFF,</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SW1-6 OFF</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Baud Rate:</th>
<th>SW1-1 to SW1-4 same as Printer Mode</th>
<th>P8 11444s,r</th>
<th>P8 49291+s16,r</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>(not available)</td>
<td>1 dec/$01 hex</td>
<td>1 dec/$01 hex</td>
</tr>
<tr>
<td>75</td>
<td>0 dec/$00 hex</td>
<td>2 dec/$02 hex</td>
<td>2 dec/$02 hex</td>
</tr>
<tr>
<td>110</td>
<td>176 dec/$00 hex</td>
<td>3 dec/$03 hex</td>
<td>3 dec/$03 hex</td>
</tr>
<tr>
<td>135</td>
<td>144 dec/$00 hex</td>
<td>4 dec/$04 hex</td>
<td>4 dec/$04 hex</td>
</tr>
<tr>
<td>150</td>
<td>128 dec/$00 hex</td>
<td>5 dec/$05 hex</td>
<td>5 dec/$05 hex</td>
</tr>
<tr>
<td>3600</td>
<td>64 dec/$00 hex</td>
<td>6 dec/$06 hex</td>
<td>6 dec/$06 hex</td>
</tr>
<tr>
<td>6000</td>
<td>32 dec/$00 hex</td>
<td>7 dec/$07 hex</td>
<td>7 dec/$07 hex</td>
</tr>
<tr>
<td>12000</td>
<td>16 dec/$00 hex</td>
<td>8 dec/$08 hex</td>
<td>8 dec/$08 hex</td>
</tr>
<tr>
<td>18000</td>
<td>11 dec/$00 hex</td>
<td>9 dec/$09 hex</td>
<td>9 dec/$09 hex</td>
</tr>
<tr>
<td>24000</td>
<td>8 dec/$00 hex</td>
<td>10 dec/$0A hex</td>
<td>11 dec/$0A hex</td>
</tr>
<tr>
<td>36000</td>
<td>4 dec/$00 hex</td>
<td>12 dec/$0B hex</td>
<td>12 dec/$0B hex</td>
</tr>
<tr>
<td>48000</td>
<td>2 dec/$00 hex</td>
<td>13 dec/$0C hex</td>
<td>13 dec/$0C hex</td>
</tr>
<tr>
<td>72000</td>
<td>1 dec/$00 hex</td>
<td>14 dec/$0D hex</td>
<td>14 dec/$0D hex</td>
</tr>
<tr>
<td>96000</td>
<td></td>
<td>15 dec/$0E hex</td>
<td>15 dec/$0E hex</td>
</tr>
<tr>
<td>192000</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Data Format:</th>
<th>SW2-1 ON</th>
<th>SW2-1 OFF</th>
</tr>
</thead>
<tbody>
<tr>
<td>8 data, 1 stop</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7 data, 1 stop</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6 data, 1 stop</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5 data, 1 stop</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4 data, 2 stop</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3 data, 2 stop</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2 data, 2 stop</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 data, 2 stop</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*add 1 if*

p = 1 or 0

<table>
<thead>
<tr>
<th>Parity:</th>
<th>SW2-1 ON</th>
<th>SW2-1 OFF</th>
</tr>
</thead>
<tbody>
<tr>
<td>none</td>
<td></td>
<td></td>
</tr>
<tr>
<td>odd</td>
<td></td>
<td></td>
</tr>
<tr>
<td>even</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MARK</td>
<td>(not available)</td>
<td>(not available)</td>
</tr>
<tr>
<td>SPACE</td>
<td>(not available)</td>
<td>(not available)</td>
</tr>
</tbody>
</table>

*Table B-1. SIC Switch Settings, PEEKs and POKEs, Part I*
<table>
<thead>
<tr>
<th>Selection</th>
<th>SSC switches and settings</th>
<th>P8 Serial Card</th>
<th>Super Serial Card</th>
</tr>
</thead>
<tbody>
<tr>
<td>Line Width:</td>
<td>SW2-3 &amp; SW2-4, same as Printer Mode</td>
<td>POKEx 1784+, r 1 to 255; for no &lt;CR&gt;, r = 0</td>
<td>POKEx 1784+, r 4 to 255; for no &lt;CR&gt;, POKEx 1440+, POKEx 1400+, (old value + 128)</td>
</tr>
<tr>
<td>Video/Generate &lt;LF&gt;/ Translate/ &lt;CR&gt; Delay:</td>
<td>SW2-3 &amp; SW2-4 (no switch)</td>
<td>V = Video on? G = Gen. &lt;LF&gt;? T = LC to UC? D = Dly 1/4 s? POKEx 2040+, r =</td>
<td>dec hex V G T D 4 $84 Y N N Y 5 $85 Y Y Y Y 36 $24 Y N Y N 37 $25 Y Y Y N 68 $44 Y N Y N 69 $45 Y Y Y N 100 $66 Y N N N 181 $65 Y Y N N 132 $84 N N Y Y 133 $85 N Y Y Y 164 $A4 N N N Y 165 $A5 N Y N N 196 $C4 N N N N 197 $C5 N Y N N 228 $E4 N N N N 229 $E5 N N N N</td>
</tr>
</tbody>
</table>

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

P8A EMULATION POKEs

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

1) The <CR> delay switch now determines whether an ETX/ACK handshake is performed after each <CR> that is transmitted. The corresponding RAM bit was not the same as the P8 <CR> delay bit, but was kept in bit 2 of location 1400+. For SSC emulation, the control is the same as the <CR> delay bit as noted above (in location 1144+).

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+ 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 POKEx 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 POKEx to location 1400+. For example, to change the enquire character to ENQ (ASCII 5), which is used by many RS-232 devices, use this POKEx: POKEx 1400+, 5. Note that this also disables the automatic generation of carriage returns. Actually, any character between 0 and 31 can be used, although only 3 and 5 are used much.

OTHER EMULATION MODE DIFFERENCES

If your old programs do not work 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 handshaking 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 RTS 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 SCEx00 (for example, by a P8Exs or INExs). The SSC is only reinitialized 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 ACTs' 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 <CR> after n characters (if automatic <CR> generation is enabled via the C command (Chapter 2); n can be any value from 40 through 255.

LINE WIDTH 40 AND VIDEO ON—<CTRL-I>:1

This command turns on the Apple II video screen and sets the line width to 40.

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 <LF> after <CR>.

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 2-3/4" x 7" (68.8 mm x 177.8 mm)
Weight 3 oz. (90 g), approximately
Cables required Internal cable from 18-pin header on SSC to DB-25 connector on case of Apple II (supplied); shielded RS-232-C cable to external device (not supplied)

Controls 2 blocks of 7 switches each, set by user before installation

Special Tools None required

ENVIRONMENT

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

SPECIAL CIRCUITS

SY6551 Asynchronous Communications Interface Adapter
2316 Read Only Memory (2,048 by 8 bits) with SSC firmware
The SSC has the usual power supply bypassing capacitors
APPLE II SLOT LOCATION

BASIC programs any slot except slot #4
APPLESOFT programs any slot except slot #0
PASCAL programs slot #1 for use with printer, etc.

SOFTWARE COMPATIBILITY

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

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

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

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

CONNECTOR PIN ASSIGNMENTS

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

<table>
<thead>
<tr>
<th>10-pin DB-25 Header</th>
<th>Connector</th>
<th>Signal name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>Frame Ground</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>Transmit Data (TXD)</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>Receive Data (RXD)</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>Request To Send (RTS)</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>Clear To Send (CTS)</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>Data Set Ready (DSR)</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>Secondary Clear To Send (SCCTS)</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>Signal Ground</td>
</tr>
<tr>
<td>9</td>
<td>20</td>
<td>Data Terminal Ready (DTR)</td>
</tr>
<tr>
<td>10</td>
<td>8</td>
<td>Data Carrier Detect (DCD)</td>
</tr>
</tbody>
</table>

Table C-1. Connector Pin Assignments

JUMPER BLOCK WIRING

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

Note that all RS-232-C signals on the SSC use negative-true logic; that is, they are true (asserted) at 0 volts and false at +5 volts.

<table>
<thead>
<tr>
<th>Signal at SSC</th>
<th>MODEM position (pin)</th>
<th>TERMINAL position (pin)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transmit Data</td>
<td>Transmit Data (2)</td>
<td>Receive Data (3)</td>
</tr>
<tr>
<td>Receive Data</td>
<td>Receive Data (3)</td>
<td>Transmit Data (2)</td>
</tr>
<tr>
<td>Request To Send</td>
<td>Request To Send (4)</td>
<td>Data Carrier Detect (8)</td>
</tr>
<tr>
<td>Clear To Send</td>
<td>Clear To Send (5)</td>
<td>Data Carrier Detect (8)</td>
</tr>
<tr>
<td>Data Set Ready</td>
<td>Data Set Ready (6)</td>
<td>Data Terminal Ready (20)</td>
</tr>
<tr>
<td>Data Terminal Ready</td>
<td>Data Term. Ready (20)</td>
<td>Data Set Ready (6)</td>
</tr>
<tr>
<td>Data Carrier Detect</td>
<td>Data Carrier Detect (8)</td>
<td>Request To Send (4)</td>
</tr>
<tr>
<td>Data Carrier Detect</td>
<td>Data Carrier Detect (8)</td>
<td>Clear To Send (5)*</td>
</tr>
</tbody>
</table>

*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
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 0 (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 (HARK parity; BASIC); for example, 11001001 for the letter M.
- 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-N>, which means "hold down CTRL and SHIFT while pressing N."). But <ESC>9 means "type ESC, THEN type 9" because the 9 is outside the angle brackets.
7-BIT LOW LOW HI HI ASCII CHAR INTERPRETATION NORMAL TERMINAL

<table>
<thead>
<tr>
<th>BINARY</th>
<th>DEC</th>
<th>HEX</th>
<th>DEC</th>
<th>HEX</th>
<th>CHAR</th>
<th>INTERPRETATION</th>
<th>NORMAL TERMINAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>00000</td>
<td>028</td>
<td>1E</td>
<td>156</td>
<td>9C</td>
<td>FS</td>
<td>n/a</td>
<td>&lt;ESC&gt;1</td>
</tr>
<tr>
<td>00001</td>
<td>129</td>
<td>81</td>
<td>SOH</td>
<td>Start of Header</td>
<td>0</td>
<td>1</td>
<td>Ctrl-A</td>
</tr>
<tr>
<td>00002</td>
<td>130</td>
<td>82</td>
<td>STX</td>
<td>Start of Text</td>
<td>1</td>
<td>2</td>
<td>Ctrl-B</td>
</tr>
<tr>
<td>00003</td>
<td>131</td>
<td>83</td>
<td>ETX</td>
<td>End of Text</td>
<td>2</td>
<td>3</td>
<td>Ctrl-C</td>
</tr>
<tr>
<td>00004</td>
<td>132</td>
<td>84</td>
<td>EOT</td>
<td>End of Transm.</td>
<td>3</td>
<td>4</td>
<td>Ctrl-D</td>
</tr>
<tr>
<td>00005</td>
<td>133</td>
<td>85</td>
<td>ENQ</td>
<td>Enquiry</td>
<td>4</td>
<td>5</td>
<td>Ctrl-E</td>
</tr>
<tr>
<td>00006</td>
<td>134</td>
<td>86</td>
<td>ACK</td>
<td>Acknowledge</td>
<td>6</td>
<td>7</td>
<td>Ctrl-F</td>
</tr>
<tr>
<td>00007</td>
<td>135</td>
<td>87</td>
<td>BEL</td>
<td>Bell</td>
<td>8</td>
<td>9</td>
<td>Ctrl-G</td>
</tr>
<tr>
<td>00008</td>
<td>136</td>
<td>88</td>
<td>BS</td>
<td>Backspace</td>
<td>10</td>
<td>11</td>
<td>Ctrl-H</td>
</tr>
<tr>
<td>00009</td>
<td>137</td>
<td>89</td>
<td>HT</td>
<td>Horizontal Tab</td>
<td>12</td>
<td>13</td>
<td>Ctrl-I</td>
</tr>
<tr>
<td>00010</td>
<td>138</td>
<td>8A</td>
<td>LF</td>
<td>Linefeed</td>
<td>14</td>
<td>15</td>
<td>Ctrl-J</td>
</tr>
<tr>
<td>00011</td>
<td>139</td>
<td>8B</td>
<td>VT</td>
<td>Vertical Tab</td>
<td>16</td>
<td>17</td>
<td>Ctrl-K</td>
</tr>
<tr>
<td>00012</td>
<td>140</td>
<td>8C</td>
<td>FF</td>
<td>Form Feed</td>
<td>18</td>
<td>19</td>
<td>Ctrl-L</td>
</tr>
<tr>
<td>00013</td>
<td>141</td>
<td>8D</td>
<td>CR</td>
<td>Carriage Return</td>
<td>20</td>
<td>21</td>
<td>Ctrl-M</td>
</tr>
<tr>
<td>00014</td>
<td>142</td>
<td>8E</td>
<td>SO</td>
<td>Start of Text</td>
<td>22</td>
<td>23</td>
<td>Ctrl-N</td>
</tr>
<tr>
<td>00015</td>
<td>143</td>
<td>8F</td>
<td>SI</td>
<td>Shift In</td>
<td>24</td>
<td>25</td>
<td>Ctrl-O</td>
</tr>
<tr>
<td>00016</td>
<td>144</td>
<td>90</td>
<td>DLE</td>
<td>Data Link Escape</td>
<td>26</td>
<td>27</td>
<td>Ctrl-P</td>
</tr>
<tr>
<td>00017</td>
<td>145</td>
<td>91</td>
<td>DC1</td>
<td>Device Control 1</td>
<td>28</td>
<td>29</td>
<td>Ctrl-Q</td>
</tr>
<tr>
<td>00018</td>
<td>146</td>
<td>92</td>
<td>DC2</td>
<td>Device Control 2</td>
<td>30</td>
<td>31</td>
<td>Ctrl-R</td>
</tr>
<tr>
<td>00019</td>
<td>147</td>
<td>93</td>
<td>DC3</td>
<td>Device Control 3</td>
<td>32</td>
<td>33</td>
<td>Ctrl-S</td>
</tr>
<tr>
<td>00020</td>
<td>148</td>
<td>94</td>
<td>DC4</td>
<td>Device Control 4</td>
<td>34</td>
<td>35</td>
<td>Ctrl-T</td>
</tr>
<tr>
<td>00021</td>
<td>149</td>
<td>95</td>
<td>NAK</td>
<td>Negative Acknowledge</td>
<td>36</td>
<td>37</td>
<td>Ctrl-U</td>
</tr>
<tr>
<td>00022</td>
<td>150</td>
<td>96</td>
<td>SYN</td>
<td>Synchronization</td>
<td>38</td>
<td>39</td>
<td>Ctrl-V</td>
</tr>
<tr>
<td>00023</td>
<td>151</td>
<td>97</td>
<td>ETB</td>
<td>End of Text Blk.</td>
<td>40</td>
<td>41</td>
<td>Ctrl-W</td>
</tr>
<tr>
<td>00024</td>
<td>152</td>
<td>98</td>
<td>CAN</td>
<td>Cancel</td>
<td>42</td>
<td>43</td>
<td>Ctrl-X</td>
</tr>
<tr>
<td>00025</td>
<td>153</td>
<td>99</td>
<td>EM</td>
<td>End of Medium</td>
<td>44</td>
<td>45</td>
<td>Ctrl-Y</td>
</tr>
<tr>
<td>00026</td>
<td>154</td>
<td>A0</td>
<td>SUB</td>
<td>Substitute</td>
<td>46</td>
<td>47</td>
<td>Ctrl-Z</td>
</tr>
<tr>
<td>00027</td>
<td>155</td>
<td>A1</td>
<td>ESC</td>
<td>Escape</td>
<td>48</td>
<td>49</td>
<td>&lt;ESC&gt;</td>
</tr>
</tbody>
</table>

1. Normal Command character in Communication Mode.
2. Used in ETX/ACK protocol (SIC CPA Emulation Mode).
3. Used in ENQ/ACK protocol (SIC CPA Emulation Mode).
4. Used in ETX/MX/EX or ENQ/Ack protocol (SIC CPA Emulation Mode).
5. Or use key.
7. Or use CRETURN key.
8. XON in XON/XOFF protocol (usually in Communication Mode).
9. Remote-control command to Exit from Terminal Mode.
10. XOFF in XON/XOFF protocol (usually in Communication Mode).
11. Remote-control command to Enter Terminal Mode.
12. Or use key.
13. Use the ESC key to generate the Escape character with the normal Apple II keyboard. In Terminal Mode, use <ESC>0.
APPENDIX E
TROUBLESHOOTING HINTS

This appendix contains two tables designed to help you diagnose problems that may 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 (DCRs) 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.

15. Use Closing Quote (39). For high value, use CHR$(96), etc.
<table>
<thead>
<tr>
<th>Problem</th>
<th>Symptom</th>
<th>Possible Cause</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>no data transfer</td>
<td>no sign of any communication at all</td>
<td>cable wires not connected OK; jumper block facing wrong way</td>
<td>check all cable connections, then pin assignments; try reversing jumper block</td>
</tr>
<tr>
<td>characters garbled</td>
<td>jh2 3g2$Q</td>
<td>wrong baud rate</td>
<td>change SW2-1 to SW1-4 or use CnB command</td>
</tr>
<tr>
<td></td>
<td></td>
<td>wrong data format</td>
<td>change SW2-1 (and SW2-2 in Comm Mode) or use CnD command to change format</td>
</tr>
<tr>
<td></td>
<td></td>
<td>other device is off, out of paper, etc., off-line</td>
<td>turn on device, remedy its problems, put it on-line</td>
</tr>
<tr>
<td>paper not advancing</td>
<td>one line of smudge</td>
<td>printer needs line feeds from SCC</td>
<td>turn SW2-5 ON or use Linefeed E(nable command)</td>
</tr>
<tr>
<td>printer is skipping lines</td>
<td>lines look like this</td>
<td>printer and SCC both generating CLEF after CRO</td>
<td>turn off SW2-5 in Printer Mode, or use Linefeed D(isable command)</td>
</tr>
<tr>
<td>missing characters</td>
<td>msg characters</td>
<td>device buffer is overflowing</td>
<td>If device supports full RS-232-C handshaking, ensure all required cable wires are connected</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>If device supports only ETX/JACK, set SCR PEA Mode</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>If device supports XON/XOFF, set Printer Mode and use X(OFF Enable cmd or set Comm Mode</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>If device supports none of these, set delays with CnD, CnL and CnF cmd</td>
</tr>
<tr>
<td>device sticks at line’s end</td>
<td>one long OK line, smudge at right end</td>
<td>device doesn’t generate own CRO, and isn’t getting enough from Apple</td>
<td>use SCR P8 Mode and CnD command, or Printer Mode and C command plus appropriate SW-2-3 and SW-4</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>have software send CRO before right margin</td>
</tr>
</tbody>
</table>

Table E-1. Problems Detected at the Device

<table>
<thead>
<tr>
<th>Problem</th>
<th>Symptom</th>
<th>Possible Cause</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apple has occasional</td>
<td>it works one minute &amp; not next</td>
<td>ACIA interrupting</td>
<td>make sure that interrupt switch SW-2-6 is OFF</td>
</tr>
<tr>
<td>bad times</td>
<td></td>
<td>the Apple when DCD or DSR changes</td>
<td></td>
</tr>
<tr>
<td>Apple not working</td>
<td>dead kybd and screen</td>
<td>SSC in slot 3 under Pascal</td>
<td>Pascal expects external terminal to run the show</td>
</tr>
<tr>
<td>Apple kybd seems off</td>
<td>keystrokes all lost</td>
<td>echo off; keyboard zapped; INP not 0</td>
<td>use E(cho E(nable cmd; unzap with POKE; INP)</td>
</tr>
<tr>
<td>screen seems off</td>
<td>nothing typed is displayed</td>
<td>device not echoing</td>
<td>in Comm or Terminal Mode, use E(cho E(nable; in SIC or Printer Mode, use l command or SW-2-3 &amp; 4 ON</td>
</tr>
<tr>
<td>screen is seeing double</td>
<td>eevemwry ttwhfingg ttwvixccee</td>
<td>device &amp; SCC both echoing to Apple</td>
<td>use E(cho D(isable cmd in Comm Mode or use CnD cmd in Printer Mode</td>
</tr>
<tr>
<td>screen is spacing double</td>
<td>lines look like this</td>
<td>device generating and sending CLEF after CRO</td>
<td>use M(ask Enable command to remove extra linefeeds</td>
</tr>
<tr>
<td>forced uppercase display</td>
<td>lowerace beCOMES UPPERCASE</td>
<td>Apple monitor changing letters in GRTLINE routine</td>
<td>use CnD command to allow lowercase to pass through (not possible in Pascal)</td>
</tr>
<tr>
<td>Apple misses some characters</td>
<td>pple sees one racters t the beginning of lines</td>
<td>screen scrolling too slowly, or BASIC or Pascal program running too slowly, and so ACIA overrun</td>
<td>turn off screen (CnN or SW-2-3 &amp; 4 in Pctr Mode); reduce scroll window; use assembly language or faster program routines; use lower baud rate (300 vs. 1200); use CnC, CnL or CnX commands; in Comm Mode, chain (CnS cmd) to 80-column card with its own scrolling hardware</td>
</tr>
</tbody>
</table>

Table E-2. Problems Detected at the Apple
APPENDIX F
ERROR CODES

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

§678+s

Status Byte (STSBYTE)

<table>
<thead>
<tr>
<th>Bit</th>
<th>&quot;1&quot; Means</th>
<th>&quot;0&quot; Means</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Parity Error occurred</td>
<td>No Parity Error occurred</td>
</tr>
<tr>
<td>1</td>
<td>Framing Error occurred</td>
<td>No Framing Error occurred</td>
</tr>
<tr>
<td>2</td>
<td>Overrun occurred</td>
<td>No Overrun occurred</td>
</tr>
<tr>
<td>3</td>
<td>Carrier lost</td>
<td>Carrier present</td>
</tr>
<tr>
<td>5</td>
<td>Error occurred</td>
<td>No error occurred</td>
</tr>
</tbody>
</table>

Table F-1. STSBYTE Bit Definitions

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

Bits 0, 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 0, 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.
**GLOSSARY**

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

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

**American Standard Code for Information Interchange (ASCII):** A standard defining the code 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 (Syntek 6531 or equivalent) that converts data from parallel to serial form and vice versa, and handles serial transmission and reception and RS-232-C signals, under the control of internal registers set and changed by SSC firmware.

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

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

**Bit:** A binary digit, either a 0 or a 1.

---

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

<table>
<thead>
<tr>
<th>BASIC PEEK $678+$$</th>
<th>Carrier</th>
<th>Framing</th>
<th>Parity</th>
</tr>
</thead>
<tbody>
<tr>
<td>or Pascal IORESULT</td>
<td>Lost</td>
<td>Overrun</td>
<td>Error</td>
</tr>
<tr>
<td>∅</td>
<td>(no error)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>no</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>33</td>
<td>no</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>34</td>
<td>no</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>35</td>
<td>no</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>36</td>
<td>no</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>37</td>
<td>yes</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>38</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>39</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>40</td>
<td>yes</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>41</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>42</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>43</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>44</td>
<td>yes</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>45</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>46</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>47</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
</tbody>
</table>

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.
BREAK: A 0.233 second SPACE (Ø) 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 $C064h±8h) 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 $C088h±8h) that stores data format and baud rate selections.

Daisy Chaining: A method of passing incoming signals and data from one peripheral connector slot to another, such as from the SSC slot to a slot containing an 80-column-display card.

Data Bit: With the SSC, one of 5 to 8 bits representing a character.

Data Carrier Detect (DCD): An RS-232-C signal from a DCE to a DTE (such as the Apple II) indicating that a communication connection has been established. The SSC's internal circuits hold DCD false until the external device sets DCD true.

Data Communication Equipment (DCE): As defined by the RS-232-C standard, any device that transmits or receives information. Usually this is a modem. However, when a Modem Eliminator is used, the Apple II looks like a DCE to the other device, and the other device looks like a DCE to the Apple.

Data Conversion: Changing of data from parallel to serial form or from serial to parallel form.

Data Format: The form in which data is stored, manipulated or transferred. Serial data transmitted and received by the SSC has a data format of: one start bit, 5 to 8 data bits, an optional parity bit, and one, one and a half, or two stop bits.

Data Set Ready (DSR): An RS-232-C signal from a DCE to a DTE indicating that the DCE has established a connection.

Data Terminal Equipment (DTE): As defined by the RS-232-C standard, any device that generates or absorbs information, thus acting as a terminator 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 P&A Emulation Mode).

Even Parity: Use of an extra bit set to 0 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 (1000001) has two 1 bits; for even parity, the transmitting device appends an eighth bit equal to 0 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 1 (FRM) of its Status Register to 1. The ACIA checks and records each framing error separately: if the next character is OK, the FRM bit is cleared.

Full Duplex: Capable of simultaneous two-way communications.

Half Duplex: Capable of communications in one direction at a time.

Handshake: A kind of communication protocol in which the receiving device, when it has successfully gotten a character or block of characters, sends back an acknowledging signal, thereby triggering the next transmission.

Hardware: The actual physical switches, wires, chips, PC boards, and so on, of a computer system.

Header: A cable connector mounted on a PC board.

Hexadecimal: A numbering system that uses 16 digits; usually these are represented by the ten decimal digits, 0 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 away) 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 0 or 1 (that is, 0 or 1 times 2 to the 0 power).

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

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

Low-order Bit: See Least Significant Bit.

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

Mode: Manner of operating. The SSC can operate in one of four chief modes, depending on the settings of switches SW1-5 and SW1-6: Printer Mode, Communications Mode, SIC P&A Emulation Mode, and SIC P&A 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 $0$ or $1$ times $2$ to the power one less than the total number of bits in the binary number. For example, in the binary number $10000$, the $1$ represents $1$ times $2$ to the fourth power, or sixteen.

Odd Parity: Use of an extra bit set to $0$ or $1$ as necessary to make the total number of 1 bits an odd number. For example, the 7-bit ASCII code for the letter A ($1000001$) has two 1 bits; for odd parity, the transmitting device appends an eighth bit equal to 1, making the total number of 1 bits odd. The receiving device can check for transmission errors by counting 1 bits.

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 $0$ (PAR) of its Status Register to $1$.

Peripheral Connector Slot: One of eight 56-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 14-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 $00000+0$) 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 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 $0$ volts.

Scratchpad RAM: Eight locations in the Apple’s memory reserved for each of the 8 peripheral connector slots (64 bytes in all).


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.
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
5. Figure 1-5 Internal Cable Attached Correctly to SSC

Chapter 2
PRINTER MODE

5. Figure 2-1 SSC Set for Printer Mode
10. Figure 2-2 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
26. Figure 3-2 SSC in Slot #2 and Clamp Assembly in Notch
22. Table 3-1 Commonly Used Switch Settings for Communications Mode
23. Table 3-2 Baud Rate Switch Settings
23. Table 3-3 Data Format Selections
29. Table 3-4 Summary of Communications Mode Commands
30. Table 3-5 Baud Rate Selections
30. Table 3-6 Data Format Selections
31. Table 3-7 Parity Selections
32. Table 3-8 Time Delay Selections
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  Serial Data Transfer
39  Figure 4-3  Parallel-to-Serial Data Conversion
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
44  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

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

Appendix B

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

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

Appendix E

TROUBLESHOOTING HINTS

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

Appendix F

ERROR CODES

109  Table F-1  STSBYTE Bit Definitions
110  Table F-2  Error Codes and Bits
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 65, 47
bus
   address 45
   data 45, 48
   parallel 48
byte 38
   device signature 51
C

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

122 SUPER SERIAL CARD

INDEX 123
D

data 37
ASCII 24
binary 24
bit 7, 15, 23, 24, 30, 40, 93
bus 45, 48
bus buffer 45
conversion format 7, 23, 28, 30, 40, 47, 91, 92, 93
remote 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)
DC-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

E
F command 31, 34, 35
echoing 34, 35
echo connector 2
EIA 40
80-column-display 34
Electronic Industries Association (see EIA)
emulation modes 91-96
old Serial Interface Card 91
PB 92
SIC 92, 102
enable 12, 28
enquire character 95
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
ENQ/ACK 102
<ESC> 34, 35, 101-104
<ESCL 91
<ESCU 91
ETX 91, 94, 95, 102, 106
EOT/ACK 91, 94, 102, 106
Expansion REM, SSC 46, 52
even parity 16, 31, 40, 92, 93
external cable 1, 10, 26

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

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

I
IN# 14
IDS 560 Paper Tiger 6
indirect addressing 96
input 37
data 48, 96
installation, SSC 9
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

124 SUPER SERIAL CARD

INDEX 125
modem 22, 26, 27, 41, 42, 105
answer 42
eliminator 5, 42, 43, 45, 48, 99, 105
null 42, 48
originate 42
MODEM 5, 21, 27, 48, 99
modulator 41
monitor
Apple II 11
ROM entry points 55

NEC 5510 Spimwriter 6
parity 31
null modem 42, 48

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

P
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
1/O card 21, 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
even codes 109, 110
Pascal 1.0 49
tentry points 55, 56
interrupts 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
Y 50, 55-57
remote 37
device 19, 22, 26, 27, 96
terminial 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, 103
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 compatibility 98
SPACE parity 16, 31, 40, 101
special characters, ASCII 101
circuits, SIC 97
specifications, SIC 97-98
specify screen slot 34
SIC 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
equipment 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 1/0 space 52
photo 1
physical characteristics 97
registers in 1/0 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 STRBYTE)
register, ACIA 109
status/reset register 45
stop bit 7, 15, 23, 24, 30, 40, 92, 94
STRBYTE 109
summary
Communications Mode commands 28-29
Printer Mode commands 14
Terminal Mode commands 35
Super Serial Card (see SIC)
suppress characters
control 33
keyboard 18, 33
CFLP 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
SW 48, 54, 56, 57, 92, 93
SW-1 through SW-4 2, 7, 15, 23, 31, 92, 93, 106
SW-5 and SW-6 2, 5, 21, 92, 93, 102, 115
SW-7 2, 8, 24, 48, 92, 99
SW-2 3, 22, 48, 54, 56, 57, 92, 93, 94
SW-2-1 2, 7, 15, 23-24, 31, 92, 93, 94, 106
SW-2-2 2, 7, 16, 23-24, 31, 92, 94, 106
SW-2-3 and SW-2-4 2, 8, 23, 92, 94, 106, 107
T
T command 35
tab 19
terminal
connecting 8
local 11
unintelligent 34, 35
TERMINAL 5, 11, 21, 42, 48, 95, 99, 101, 105
Terminal Mode 22, 34-36, 101-104
bypass 96
change to 96
commands 35
enter 35, 102
example 36
quit 36, 102
time of operation, SSC 45
time delay
selection 16, 32
time-sharing 34, 36
translate 94
comm 17, 31, 91
lowercase 18
transmit
break signal 36
interface 45
register 42
transmit/receive register 45
Transmitted Data (see TXD)
troubleshooting 105-107
at Apple II 107
at device 106
TXD 41, 42, 43, 98, 99
U
unintelligent terminal 34, 35, 36
uppercase 14, 18, 29, 33, 34, 91, 92, 102
V
video on/off 8, 92, 94, 96
W
wiring, jumper block 99
X
X command 17, 31, 35, 106
X register 50, 55-57
XOFF 35, 102
recognition 18, 33
XON 102
XOFF/XON 17, 19, 31, 33, 96, 102
Y
Y register 50, 55-57
Z
Z command 17, 18, 31
zero-page locations, SSC 52