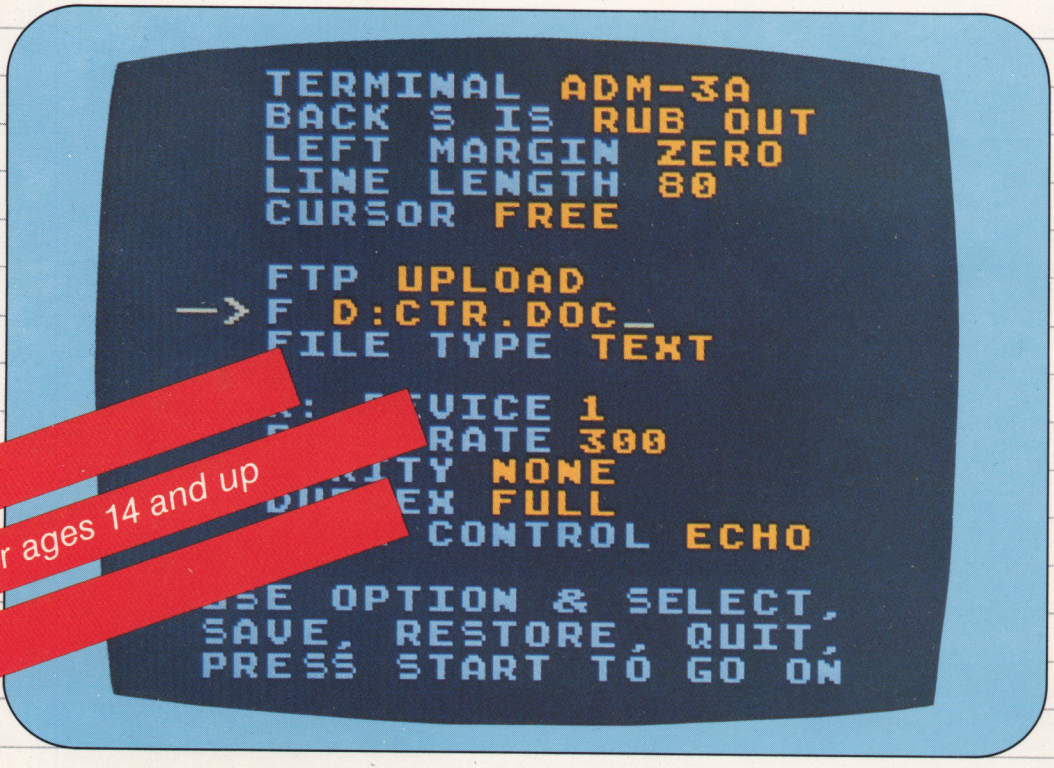
CHAMELEON

SYSTEMS/TELECOMMUNICATIONS

Convert your computer into a GLASS TTY, ADM-3A, or VT-52 terminal



For ages 14 and up

CONSUMER-WRITTEN PROGRAMS FOR

ATARI®

HOME COMPUTERS

APX

ATARI Program Exchange

Printed in U.S.A.

CHAMELEON CRT TERMINAL EMULATOR

by

John H. Palevich

Distributed By

The ATARI Program Exchange
P.O. Box 3705
Santa Clara, CA 95055

To request an APX Product Catalog, write to the address above, or call toll-free:

800/538-1862 (outside California)
800/672-1850 (within California)

Or call our Sales number, 408/727-5603

Trademarks of Atari

ATARI is a registered trademark of Atari, Inc. The following are trademarks of Atari, Inc: 400, 410, 800, 810, 820, 822, 825, 830, 850, 1200XL.

Contents

Figures

Figure

Tables

Table

Introduction

Overview

Chameleon lets you convert your ATARI Home Computer into a computer terminal. It emulates four terminals: "GLASS TTY" (that is, a video terminal with the same features as a printing terminal), Lear Siegler's ADM-3A, Digital Equipment Corporation's VT-52, and a Test terminal, useful for debugging. For Glass, ADM-3A, and VT-52, *Chameleon* supports tabs, backspace, line feed, form feed, and a visual bell signal. For the GLASS TTY, it supports word-wrapping. For the ADM-3A and VT-52, it also supports cursor addressing and basic editing features. It doesn't support the special graphics character set of the VT-52 or some rarely used ADM-3A features.

Chameleon's major features are:

- a 24-line, 40-column to 134-column character display (displaying 40 columns at a time) with a lock cursor feature for automatic horizontal scrolling
- a buffer that saves up to 9 previous pages of text, depending on the amount of available RAM
- continuous communication at baud rates of 75 to 1200 and non-continuous (requires flow control) at rates of 2400 to 9600 baud
- selectable input/output parity
- a BREAK signal
- the full ASCII standard character set
- selectable duplex
- the ability to copy text from the screen and the previous pages buffer to a file
- the ability to transfer files between an ATARI Computer and most time sharing systems
- the ability to transfer files between an ATARI Computer and other microcomputers using the CP/M-standard MODEM protocol

Programmers with a disk-based computer system, 48K of RAM, and the ATARI MACRO Assembler™ can define new terminal types with *Chameleon*; the diskette version includes the source code.

Required accessories

ATARI 830 Acoustic Modem or an RS-232C device

ATARI 850 Interface Module

- Cassette version

  24K RAM
  ATARI 410 Program Recorder

- Diskette version

  32K RAM
  ATARI 810 Disk Drive

Optional accessories

ATARI MACRO Assembler

Printer or any other input/output device

Contacting the author

Users wishing to contact the author about *Chameleon* may write to him at:

Apt. F-211
175 Calvert Drive
Cupertino, CA 95014

Attaching an RS-232C device to the interface module

Chameleon talks to port 1 of the ATARI 850 Interface Module. (This is the default behavior — you can use the R:DEVICE menu item to have *Chameleon* talk to any one of the four ports.) Only "Transmit Data", "Receive Data", and ground need be connected, but "Data Terminal Ready" and "Request to Send" are set high (+12V = true) when *Chameleon* starts as a convenience to devices requiring these signals. *Chameleon* ignores all other signals.

Loading *Chameleon* into computer memory

If you have the cassette version of Chameleon:

1. Remove any cartridge from the cartridge slot of your computer.

2. Make sure all your equipment is turned off, including your disk drive. Connect your ATARI 830 Acoustic Modem to your ATARI 850 Interface Module or to your RS-232C device. Set your modem controls according to the requirements of the host computer you're contacting.

3. Turn on your interface module or RS-232C device (consult the operating manual for your RS-232C device for any special instructions).

4. Connect your program recorder to the interface module and to a wall outlet.

5. While holding down the START key on the computer console, turn on your computer. The computer will beep. Also turn on your TV screen.

6. Insert the *Chameleon* cassette in the cassette holder of your program recorder, press REWIND until the tape rewinds completely, and then press PLAY to prepare the program recorder for loading the program.

7. Press the RETURN key on your computer keyboard. The tape will load into computer memory (RAM); then it will stop and the interface module will automatically load its software into RAM.

8. Dial the host computer and, at the high-pitched tone, place the phone receiver in the modem.

If you have the diskette version of Chameleon:

1. Remove any cartridge from the slot of your computer.

2. Make sure all your equipment is turned off. Connect your ATARI 830 Acoustic Modem to your ATARI 850 Interface Module or to your RS-232C device. Set your modem controls according to the requirements of the host computer you're contacting.

3. Turn on your interface module or RS-232C device (consult the operating manual for your RS-232C device for any special instructions).

4. Connect your disk drive to the interface module and turn on your drive.

5. When the BUSY light goes out, open the disk drive door and insert the *Chameleon* diskette with the label in the lower right-hand corner nearest to you.

6. Turn on your computer and TV set and wait until *Chameleon*'s information screen appears.

7. Dial the host computer and, at the high-pitched tone, place the phone receiver in the modem.

Demonstration mode

If you don't have an ATARI 850 Interface Module, you can still try out *Chameleon* by booting it without the interface module turned on. If you have the cassette version, *Chameleon* simply informs you that the "R:" device is not installed. If you have the diskette version, *Chameleon* asks you to:

```
Turn on 850 and press a key,
or RETURN for Demo mode.
```

If you've simply forgotten to turn on your interface module, then turn it on now and press the SPACE bar. If you really don't have an interface module, then press RETURN.

Chameleon's information screen will have an extra line in it

```
"R:" device not installed
```

and *Chameleon* won't be able to talk to any other system.

Chameleon's information screen

The following information screen displays until you press the SPACE bar.

```
Chameleon TTY Emulator

Disk (or Cassette) Version 3.1

(C) 1982 John Howard Palevich

Press SPACE BAR for the Menu
(or other text, see below)
```

Figure 1. CRT Terminal Emulator Information Screen

Terminal emulator menu keys

After pressing the SPACE bar in response to the information screen, the terminal emulator menu screen displays in double-width, colored characters. A dark grey arrow (-►) points to the item you've selected. The item's name is blue and the item's value is yellow. The menu screen looks approximately as follows (the initial settings are underlined).

```
►TERMINAL GLASS TTY
 BACK S IS CTRL H
 LEFT MARGIN TWO
 LINE LENGTH 82
 CURSOR FREE

 FTP NONE
 F P:
 FILE TYPE TEXT

 R: DEVICE 1
 BAUD RATE 300
 PARITY OFF
 DUPLEX FULL
 FLOW CONTROL ^S/^Q

 USE OPTION & SELECT
 SAVE, RESTORE, QUIT, (disk version only)
 PRESS START TO GO ON
```

Figure 2. Terminal Emulator Menu Screen

Selecting an item—OPTION key

Each time you press the OPTION key, the dark grey arrow moves down one line so that you can select the next item in the menu. You return to the first menu item after reaching the last one. You can also reverse the order by holding down the SHIFT key while pressing the OPTION key, which causes the arrow to move to the previous menu item. With this method, you return to the last menu item after reaching the first one.

Selecting a value for an item — SELECT key

Each time you press the SELECT key, a new value displays for the current menu item. Again, to reverse the order, hold down the SHIFT key while pressing the SELECT key, which causes the previous value to display.

Entering a file name

The "F" line (line 7) of the menu displays the file currently selected for file transfer operations. When you press the SELECT or SHIFT-SELECT keys, a dark grey cursor (-) replaces the first character of the old file name. Type in your new file name, using the BACK S key to correct any errors. Press RETURN to remove the cursor.

If you always keep FTP (the File Transfer Protocol) at NONE, then you can safely leave this line alone, even if you don't have a P: (printer) device.

Moving to terminal mode — START key

Press the START key to move from the Terminal Emulator Menu screen to terminal mode.

If you enabled FTP, then *Chameleon* attempts to open the file you specified in the F line. If this operation fails (because, for example, you specified a nonexistent file), *Chameleon* prints FILE OPEN ERROR # # # where "# # #" is an error code. FTP will be disabled until you return to the Terminal Emulator Menu and correct the problem. See the section on FTP for more information.

If *Chameleon* can't talk to the R: (interface module) device, it prints the message CAN'T HACK RS-232C # # # where "# # #" is an error code, which you can decode by referring to the table in the FTP section. *Chameleon* stops functioning at this point. Turn off both your computer and your interface module, and then check the connections as well as the instructions under "Getting started."

Most times, you'll have no problems. *Chameleon* simply prints

```
NO ERRORS IN SETUP
```

and puts you into terminal mode.

Recalling the Terminal Emulator Menu — OPTION key

When you're in terminal mode, you can recall the Terminal Emulator Menu by pressing the OPTION key. The grey arrow will point to the last menu item you selected. Characters sent to you while you're in Terminal Emulator Menu mode will be lost.

If you enabled FTP, *Chameleon* closes your file when you recall the menu. You should then disable FTP; if you don't, *Chameleon* reopens your file when you exit menu mode, and, in the case of downloading or screen dump, you'll lose all the data you put into the file. This result applies to storage devices only; printers have already printed the information and so may be reopened.

Menu items

Terminal type

Use the first item to select the kind of terminal you want to emulate. Your choices are:

GLASS

This value simulates a typical printing terminal with eight column tabs, line feed, form feed, backspace, carriage return, and a visual bell with some computer installations. In addition, words typed at the end of a line wrap onto the next line, making it easier to read text formatted for very long lines. GLASS TTY is sophisticated enough to be used with many computer installations. However, it isn't sufficient for time-sharing computers capable of sending special characters for cursor control, character editing, and so on. Select one of the video terminals to communicate with such computers.

VT-52

This value simulates the DEC VT-52 terminal, except for the special graphics character set. *Chameleon* doesn't support the Enable Graphics and Disable Graphics commands. Normal mode, lowercase characters print in place of graphics. The alternate keypad mode also isn't supported. However, you may type sequences from the following table to emulate the keypad:

TABLE 1. VT-52 Alternate Keypad Mode Sequences

Key	Keypad numeric mode	Keypad application mode
0	0	ESC ? p
1	1	ESC ? q
2	2	ESC ? r
3	3	ESC ? s
4	4	ESC ? t
5	5	ESC ? u
6	6	ESC ? v
7	7	ESC ? w
8	8	ESC ? x
9	9	ESC ? y
-	- (dash)	ESC ? m
,	, (comma)	ESC ? l
.	. (period)	ESC ? n
ENTER	Carriage Return	ESC ? M
up	ESC A	ESC A
down	ESC B	ESC B
right	ESC C	ESC C
left	ESC D	ESC D
blue	ESC P	ESC P
red	ESC Q	ESC Q
grey	ESC R	ESC R

ADM3A

This value simulates the Lear Siegler ADM-3A terminal. It supports Option A (24 lines) and Option No. 1 (lowercase characters). However, it won't beep when the cursor crosses the seventy-second character position.

The Internal Configuration Control Switches are set as shown in Table 2.

Table 2. Internal Configuration Control Switches for ADM-3A

No.	Left set name	Setting	May be modified by
1	12 line/24 line	24 line	Reassembly
2	50 hz/60 hz	60 hz	N/A
3	Disable Clr Scrn	Clr Scrn	Reassembly
4	Disable Kb Lock	Disable	Reassembly
5	UC Disp/U/L Disp	U/L Disp	Reassembly
6	Space-Adv	Space	Reassembly

No.	Right set name	Setting	May be modified by
1	EOT-OFF	OFF	Reassembly
2	ETX-OFF	OFF	Reassembly
3	CODE - SEC CHAN	CODE (N/A)	Reassembly
4	202-OFF	OFF	Reassembly
5	103-OFF	103	Reassembly
6	LOCAL-OFF	OFF	Reassembly
7	CUR CTL-OFF	CUR CTL	Reassembly

Keyboard Dip Switch Settings:

No.	Name	Settings	May be modified by
L1-L7 & M1-M4	Baud Rates	300	Menu (baud rate)
	Note that Chameleon can't run at 19200 baud		
M5	HDX-FDX	FDX	Menu (duplex)
M6	RS-232C CL	RS-232C	Reassembly
M7	Auto NL-OFF	Wrap	Reassembly
H1	LC EN-UC	UC	SHIFT-lock
H2	PAR ODD/ EVEN	None	Menu (I/O parity)
H3	DATA 7/8	None	Menu (I/O parity)
H4	STOP 1/2	1	Reassembly
H5	PARITY-INH	None	Menu (I/O parity)
H6	Bit 8 0/1	None	Menu (I/O parity)

TEST

Characters sent to the screen while *Chameleon* is in TEST mode are printed as is. None of the control characters (even carriage return) is treated specially. When the cursor reaches the right margin, it moves down to the next line. When the cursor reaches the bottom of the screen, it scrolls the whole screen upward one line. Thus, the last several thousand characters sent to *Chameleon* will be visible on the screen.

This mode is useful for debugging new operating systems because it lets you find all padding characters (like nulls and rub outs) or other control characters the operating system might be sending you.

BACK S IS

This option controls which character is sent to the other computer when you type the BACK S key. The choices are:

CTRL H

This choice sends an ASCII BS (8) when you type the BACK S key. To get rub out, type SHIFT and BACK S.

RUB OUT

This choice sends an ASCII RUB (127) when you type the BACK S key. To get ctrl-h you type CTRL and H.

LEFT MARGIN

Many TV sets overscan, making the left two character positions of the screen unreadable. This option lets you select your left margin.

Your choices are:

ZERO

This choice displays a full forty characters across.

TWO

This choice gives you a two-character margin (like BASIC) so that you can use *Chameleon* even if the left edge of your screen is clipped off.

LINE LENGTH

You can adjust *Chameleon*'s line length to suit your preferences. The choices are:

40
42
80
82
132
134

Notice that for each of the three popular lengths (40, 80, and 132) there is another choice two characters longer. These "odd" lengths (42, 82, and 134) are designed to be used with a left margin of two, but you may use any line length with any left margin.

CURSOR

This line controls the behavior of the cursor. A cursor can behave in two ways:

FREE

The cursor is free to travel off the screen.

ON SCREEN

The cursor is forced to remain on the screen at all times. ON SCREEN is useful for editing text.

FTP (File Transfer Protocol)

This item controls what type of file transfer *Chameleon* will do. See the section on FTP for more information. The choices are:

NONE

No file transfer selected

SNAPSHOT

Transfer characters from the screen to a device on the ATARI Computer System

UPLOAD

Transfer characters from an ATARI Computer device up to the RS-232C device

DOWNLOAD

Transfer characters from the RS-232C device down to a device on the ATARI Computer System

MODEM UP

Transfer characters from an ATARI Computer device up to the RS-232C device using the MODEM2 protocol

MODEM DOWN

Transfer characters from the RS-232C device down to a device on the ATARI Computer System, using the MODEM2 protocol

F (file name)

The file transfer modes need a file name. If you disable FTP, you needn't consider this line. Pressing the SELECT or SHIFT-SELECT keys for this item causes the grey cursor (-) to appear. Type in the name of the file you want to use. Use the BACK S key to correct typing mistakes. Press RETURN to send your file name to the computer. See the section on FTP for more information.

FILE TYPE

Use this item to control how end-of-line and tab characters are sent and received during uploading and downloading. The choices are:

TEXT

This choice tries to map ASCII Carriage Return/Line Feed to ATASCII End of Line, and ASCII TAB to ATASCII TAB. During uploading, EOL is converted to Carriage Return. During downloading, Carriage Return is converted to EOL and all line feeds are ignored (i.e., not sent to the FTP file). Use this mode with your ASCII Computer System.

BINARY

In this choice, all characters are sent and received as is. Use this mode with another ATARI Computer or with non-text files.

R:Device

Use this item to select which of the four RS-232C ports *Chameleon* will use for communication. Most people use only port 1, but the choices are:

1
2
3
4

BAUD RATE

Use this item to select the baud (data transmission) rate. The values are in bits per second. Consult the manual or User Services for the system you're using for appropriate baud rates. Continuous transmission choices are:

75
110
134.5
150
300
600
1200
1800
2400
4800

Noncontinuous transmission is available at:

9600

PARITY

Use this item to select the kind of parity check sent to *Chameleon* from the host computer and from *Chameleon* to the host computer. Consult the manual or User Services for the system you're communicating with for the appropriate value. The values are:

NONE

Send characters with parity bit 0

ODD

Number of bits must be odd

EVEN

Number of bits must be even

ON

Send characters with parity bit 1

DUPLEX

Use this item to control where the characters you type are sent. The choices are:

FULL

Keyboard characters go to RS-232C device

LOCAL

Keyboard characters go to the screen

HALF

Keyboard characters sent to both RS-232C device and screen; use this to talk to systems requiring half duplex

FLOW CONTROL

This item selects the way in which *Chameleon* controls the flow of data between itself and the other computer. Flow control is used only at high (more than 1200) baud rates and during file transfer. The choices are:

NONE

The other computer has no means of flow control. When NONE is selected, the FTP modes UPLOAD and DOWNLOAD, and baud rates greater than 1200 will lose characters.

^S/^Q

The other computer will stop sending when it receives a control-s and resume sending when it receives a control-q. When FTP is in UPLOAD mode, *Chameleon* will pause on a ^S/ ^Q pair sent by the other computer.

ECHO

Echo acts just like NONE, except during UPLOAD, when it waits for the character sent to the other computer to come back (or "echo"). This mode is designed to allow text uploading into slow time-sharing systems that don't support the ^S/ ^Q protocol. If a character doesn't echo within one quarter of a second, then the uploading continues anyway.

DELAY

DELAY acts just like NONE, except during UPLOAD, when it forces a short (1/16 sec.) pause between characters to allow slow computers to process the characters.

SLOW

Slow acts just like NONE, except during UPLOAD, when it forces a long (1/4 sec.) pause between characters to allow very slow computers to process the characters.

Saving and restoring menu settings

The diskette version of *Chameleon* can save the current menu item settings in a diskette file called LIZARD31.OPT. You can save current menu settings any time you're in the menu. To do so, insert a DOS-II formatted writable diskette in drive 1 and press the S key. Every menu item except the F (file name) will be saved into the file D:LIZARD31.OPT.

Saved menu settings may be restored any time you are in the menu. Just insert a DOS-II formatted diskette containing the LIZARD31.OPT file into drive 1 and press the R key. *Chameleon* will try to read the saved options out of D:LIZARD31.OPT. If the file doesn't exist, or can't be read, then *Chameleon* uses its standard defaults. If the file is read correctly, then menu items (except for the file name) are set to the stored values.

Chameleon automatically tries to read in the stored defaults when it is booted. This makes it easy to customize *Chameleon* to your application.

Example: Settings for communicating with compuServe

If you want to call CompuServe, or MicroNET through CompuServe, your settings typically would be as follows:

Item 1—TTY type	→ ADM3A or VT-52 (if you want cursor control)
Item 2—Back S is	→ CTRL H
Item 3—Left Margin	→ TWO
Item 4—Line Length	→ 82
Item 5—Cursor	→ FREE
Item 6—FTP	→ NONE
Item 7—F	→ P:
Item 8—File Type	→ TEXT
Item 9—R: Device	→ 1
Item 10-Baud rate	→ 300
Item 11-Parity	→ EVEN
Item 12-Duplex	→ FULL
Item 13-Flow Control	→ ^S/^Q

Control codes

While you're using *Chameleon*, some of the ATARI control characters won't do what they normally do. For example, the CAPS key works as it usually does, but the ATARI key has been disabled. As another example, the RETURN key repositions the cursor (the light green square indicating where the next transmitted character will display on the screen) at the beginning of the current line, not at the beginning of the next line. You'll need to use CTRL-J to move the cursor to the next line.

See the tables at the end of this manual for information about what each character and control code means while you're using *Chameleon*.

Vertical and horizontal scrolling

Most time sharing system programs format text for 80-column wide printing terminals. Since the ATARI Computer's screen is only 40 characters wide, and only 24 lines high, you can't see all the text at one time.

Your ATARI Computer can, however, keep much more text in memory than it can show on the screen. *Chameleon* uses this ability to give you most of the benefits of an 80-column screen, as well as most of the benefits of a printing terminal.

In essence, when you're in the terminal mode, *Chameleon* shows you a 24 x 40 character window onto a much larger "virtual" screen. This virtual screen is as wide as the value of the LINE LENGTH setting on the menu, that is, between 40 and 134 characters wide, and as long as it can be and still fit in your ATARI Computer's memory. The virtual screen is always between 24 and 255 lines long, depending on the amount of memory your ATARI Computer has, the FTP mode, and the LINE LENGTH.

You move around the virtual screen with the START, SELECT, and SHIFT keys. To try horizontal and vertical scrolling, use the menu to set TTY TYPE to GLASS, CURSOR to FREE, DUPLEX to LOCAL, FTP to NONE, and LINE LENGTH to 80. Press START to enter the terminal mode.

Vertical scrolling—SELECT key

If you type some text and then press the RETURN key repeatedly, you'll soon notice that the cursor moves down to the bottom line of the screen and remains there, while the text you've typed in scrolls off the top of the screen. If you press and hold the SELECT key, the screen scrolls down and the text you typed returns to view. As this happens, the cursor changes to an arrow pointing down to its "true" position. As you continue to hold down the SELECT key, the text you typed continues to scroll down, and if you have enough memory, it even disappears from view off the bottom of the screen.

Eventually, though, you'll bump into the upper end of the virtual screen. Press and hold the RETURN key, and in a little while the text you typed will scroll off the top of the virtual screen, too. When it scrolls off the top of the virtual screen, it has been erased from the computer's memory and can never be retrieved. To return to the bottom of the virtual screen, press the SHIFT + SELECT keys and hold them down. The screen will scroll up and bring the cursor back into view.

Horizontal scrolling—START and SHIFT + START keys

Type a sentence (such as, "The quick brown fox jumped over the . . . ") until you reach the right-hand side of the line. If you continue to type, you notice that the cursor has become an arrow pointing right, off the edge of the screen, toward where the next character should display. Press the START key and watch the characters scroll leftward, until the screen is blank and the cursor has changed back into a small square on the left-hand edge of the screen. You've just used the START key to scroll, character by character, to the rightmost 40 columns of the 80-column display.

To return to the first 40 columns, press SHIFT + START. You can use START and SHIFT + START to center the cursor on whatever column you wish. When the cursor is positioned in its true location, it displays as a square. When it is located in a screen area other than its true location (i.e., in the previous page buffer or on columns across the screen), it displays as an arrow pointing toward its true position.

Combining vertical and horizontal scrolling

You can use the SELECT and SHIFT + SELECT keys and the START and SHIFT + START keys to position yourself anywhere on the virtual screen. If characters are typed by you or sent by the remote computer while the cursor is off the screen, they display in the correct spot, but you won't see them until you return to the portion of the screen that contains the cursor.

Locking the cursor for faster data entry

Set the CURSOR item of the menu to ON SCREEN. This causes *Chameleon* to keep the cursor on the screen at all times. With this setting, if characters sent to the screen cause the cursor to move off the screen, *Chameleon* scrolls automatically to keep the cursor square on the screen. This feature is useful when you have a lot of information to type in. You can type faster when *Chameleon* keeps track of the cursor on the screen, because you don't have to stop every few words to reposition the cursor with the START and SHIFT + START keys.

For long printouts, many people prefer to control scrolling themselves (CURSOR LOCK OFF), since they can then move the screen around to read the text at their own reading rate, as well as back up to reread a section.

FTP and the virtual screen

UPLOAD, DOWNLOAD, MODEM UP, and MODEM DOWN all work more smoothly when there is a large amount of memory to buffer text. When FTP is set to any of these modes, the virtual screen is reduced to 24 lines (the height of the actual screen). When FTP is set to NONE or SNAPSHOT, *Chameleon* doesn't need to buffer text, and so the virtual screen again expands to fill all of memory.

When is the virtual screen cleared?

The virtual screen is cleared of all the text it contains whenever its size or shape is altered. In particular the screen clears if:

- you change from one of the buffered FTP modes to one of the unbuffered FTP modes
- or you change the LINE LENGTH

The most important case where the virtual screen will NOT clear is when you change FTP from NONE to SNAPSHOT, which makes it easy to copy a portion of your virtual screen to a file.

SYSTEM RESET and BREAK keys

If you press the SYSTEM RESET key, you'll have to turn off your ATARI Computer and turn it on again to regain control.

Chameleon has disabled the computer console's BREAK key in menu or terminal mode because it's so easy to press by mistake and because the interface module software would stop sending data in that case.

When a file I/O is in progress (for example, when you're entering or leaving menu mode, or during FTP mode), you can use the BREAK key to abort the I/O.

Sending a "break" to the host computer

You'll sometimes want to send a "break" signal to the host computer—to tell it to stop sending data. To do so, press the SHIFT + ATARI keys. Each time you press these keys, *Chameleon* sends a half-second of break signal to the other computer.

File Transfer Protocols (FTP)

Terms

"File Transfer Protocol" (FTP) is a generic term for any method of moving data from one place to another, for instance, between two ATARI Computers or between an ATARI Computer and a large time sharing system. "Uploading" refers to moving data from your ATARI Computer to some other computer. "Downloading" refers to moving data from another computer into your ATARI Computer.

There are many ways to upload and download information. *Chameleon* uses one that works with almost any time sharing computer without any special programs on the other end. If this isn't the case, contact the systems staff and ask about getting "XON/XOFF flow control" installed (or write a program to simulate such a feature). "XON/XOFF flow control" refers to the process whereby a system stops printing characters when you (or *Chameleon*) send it an ASCII ^S (for XOFF), and then waits for you (or *Chameleon*) to send an ASCII ^Q (for XON), when it continues sending characters.

Chameleon uses flow control to tell the system to stop sending characters for a moment, so that it can turn off the RS-232C port and save the characters it has collected to the file you specified. Thus, if your system doesn't have flow control, you can't download more characters at a time than *Chameleon* has room for in its internal buffer. This internal buffer varies, depending on the size of your computer, from about 300 to 32,000 bytes.

When one *Chameleon* is uploading a file to another *Chameleon*, the first one stops sending characters when it gets a ^S and it restarts when it receives a ^Q. Thus, two ATARI Computer owners who each own *Chameleon* can transfer uncopyrighted files to one another over phone lines.

Types of devices that can upload

You can upload from any device from which you can read, except for K: (keyboard), S: (TV monitor), and E: (Screen Editor) devices, which *Chameleon* uses or which would interfere with *Chameleon*. Examples are:

C:	Cassette tape recorder file
D1:HILOGAME.BAS	Disk file

Types of devices that can download

You can download to any device that you can write to, except for S: (TV monitor) and E: (Screen Editor) devices, which would interfere with *Chameleon*. Examples are:

P:	Printer
C:	Cassette
D1:DATAFILE.TXT	Disk file

Special problems with the program recorder

It's fairly well known that cassettes need a nondata block of 128 characters at the beginning of the tape so that the tape can be opened long before it's read or written to. Since *Chameleon* opens FTP files when you leave menu mode, it has to read or write the first block of data at that time.

The most likely use of FTP with cassettes would be to upload or download programs saved using the LIST command. So as not to lose the first 128 characters of these programs, follow these steps.

For BASIC programs

1. Type in the direct mode command:

   OPEN#1,8,0,"C:":FOR I=1TO128:PUT#1,155:NEXT I:LIST"C:"

   and press the RETURN key.

2. The computer will beep twice. Press the RETURN key.

3. The computer will write one block of data onto the cassette.

4. The computer will beep twice. Press the RETURN key.

5. Then the computer will write your program to the tape.

When you upload this file, *Chameleon* will read in and ignore the first block of data, but it will get all of your program.

For Assembler Editor programs

1. REN ⟨RETURN⟩

   01 ;4567891123456789212345678931234567894123456789512345678961 2
   ⟨RETURN⟩
   02 ;4567891123456789212345678931234567894123456789512345678961 2
   ⟨RETURN⟩
   LIST#C: ⟨RETURN⟩

2. The computer will beep twice. Press the RETURN key.

3. The computer will then save your program with the two new comment lines in the first block of data.

When *Chameleon* uploads your program, it will read in the two comments and ignore them. Then it will get all of your program.

Entering a cassette downloading or screen-dumped program

BASIC programs

1. Type ENTER"C:" and press the RETURN key.

2. The computer will beep once. Press the RETURN key.

3. The cassette program will load in and BASIC will simply ignore the 128 ⟨EOL⟩ s that *Chameleon* wrote into the first block.

Assembler Editor programs

1. Type ENTER#C: and press the RETURN key.

2. The computer will beep once. Press the RETURN key.

3. The cassette program will load in, but the Assembler Editor will beep 128 (count 'em) times, once for each EOL it sees in the first block. It will also beep once for each blank line included in a SNAPSHOT. While this is admittedly annoying, it works!

Uploading to a time sharing system

To upload a file to a time sharing system, follow these steps.

1. Establish a connection with the system, log in, and run your favorite editor.

2. Go into text entry mode. If the mode requires line numbers, make sure your file has them. If a certain sequence of characters will stop text entry, check that your file doesn't contain these characters. Similarly, make sure your file has no control characters that might abort the editor and return you to the top level.

3. Press the OPTION key to go into *Chameleon*'s menu mode. Change the "F" menu item to the file you want to upload. Set FILE TYPE to TEXT. Set FTP to UPLOAD.

4. Press the START key. *Chameleon* will open your file for reading and return to CRT TTY mode. If *Chameleon* displays errors, fix them and try again. When *Chameleon* displays the NO ERRORS IN SETUP message, you can continue.

5. Press the SHIFT + OPTION keys to start the uploading.

6. *Chameleon* continues to upload until one of three events occurs:

   a. It comes to the end of the file, whereupon it prints QUITTING TO TTY MODE and returns you to terminal mode.

   b. You type any character on the keyboard, whereupon it prints QUITTING TO TTY MODE and returns you to terminal mode.

   c. A file I/O error occurs, whereupon it prints FILE I/O ERROR WAS :#, where "#" represents a single-colored character decodable in the table in Appendix 1 (File I/O Error Codes), and returns you to terminal mode.

7. Once you're back in terminal mode, you can type the special characters that tell your system's text editor you're finished inputting characters. That's it.

Notes

Older time sharing systems often assume terminals will type only a few characters a second, and some—such as the author's Unix system—will crash if characters are sent continuously at 2400 baud. If you find that you lose characters when you upload, use the FLOW CONTROL item on the menu to slow down uploading to a rate your system can handle.

There are often more efficient ways to save characters from the terminal than with a text editor. In many operating systems, your characters can be sent directly to a file (ask a systems staff member if you're unfamiliar with how to do this). This kind of procedure lets the system save your characters faster (and so lose fewer of them).

When FILE TYPE is TEXT, *Chameleon* sends a carriage return after each line, which may cause the lines to write over one another on the display, but it's usually the correct way to enter text to a time sharing system. Use a specially prepared file and the FILE TYPE BINARY mode if your text editor requires something else.

Downloading from a time sharing system

The steps for downloading a file are as follows.

1. Type all but the last character, which is typically pressing the RETURN key, of the command to send the file to your terminal.

2. Press the OPTION key to go to *Chameleon*'s menu mode.

3. Select F and enter the file name. Select the appropriate FILE TYPE option (probably TEXT), and set FTP to DOWNLOAD.

4. Press the START key. If *Chameleon* displays a file error message, fix the problem and try again.

5. When you have successfully set up the FTP, press SHIFT + OPTION. The prompt "TYPE CHAR TO SEND..." appears. Now type the last character in your command (probably RETURN).

6. Downloading will begin. It continues until one of two events occurs:

   a. you type another character, whereupon *Chameleon* types QUITTING TO TTY MODE and returns you to terminal mode, or

   b. a file I/O error occurs. *Chameleon* displays a FILE I/O ERROR WAS:# message, where "#" is a one-character code decodable in the FILE I/O ERROR CODE TABLE in Appendix 1.

7. The download file isn't closed until you return to the menu, so you can download different system files into the same ATARI Computer file. If you do, they'll be placed one after the other, in the order you download them.

Notes

Some operating systems use characters like TAB (ASCII 9) to compress the number of spaces in a file. There's usually a way to tell the system not to do so; otherwise, you'll have to go through the downloaded file and manually convert the tabs to the appropriate number of spaces.

When a ^S is sent to the time sharing system, it must stop sending data within 100 characters. If it doesn't, you get the message DA BUFFER OVERFLOWED and you lose all the characters sent from that point to when the system finally does stop sending data.

Taking a photograph of the screen

Use the SNAPSHOT option of the FTP mode to copy part or all of the virtual TV screen to an ATARI file. The steps are as follows.

1. Press the OPTION key to go to *Chameleon*'s menu mode.

2. Select FTP SNAPSHOT, and set F to the file name you want to dump to. If you want each line of the text to end in an ATARI EOL, then set FILE TYPE to TEXT. If you want each line to end in a CR/LF pair, set the FILE TYPE to BINARY.

3. Press the START key to go to terminal mode. If *Chameleon* displays any error message, correct the problem and try again.

4. When you're ready to send a copy of part of the virtual screen contents to your specified file, press the SHIFT + OPTION keys.

5. *Chameleon* displays the message MARK TOP OF REGION. Use the SELECT key and the SHIFT + SELECT keys to scroll the virtual screen up and down until the first line you want to copy is at the top of the TV screen. Then press the SPACE BAR.

6. *Chameleon* displays the message MARK BOTTOM OF REGION. Use the SELECT key and the SHIFT + SELECT keys to scroll the virtual screen up and down until the last line you want to copy is at the bottom of the TV screen. Press the SPACE BAR.

7. *Chameleon* displays the message DUMPING SCREEN 4 YOU and saves the screen to your specified device. Only the characters from the left edge of the screen to the rightmost nonblank character of each line will be saved.

Notes

If, in the middle of marking the region to dump, you decide not to dump that text, you can type Q to quit the snapshot. *Chameleon* says OK THEN, I WON'T DUMP and returns you to terminal mode.

If any error occurs, *Chameleon* displays a FILE I/O ERROR WAS :# message, where "#" represents a coded error decodable in the FILE I/O ERROR CODE TABLE in Appendix 1 and returns you to terminal mode. Otherwise, it displays the message QUITTING TO TTY MODE and returns you to terminal mode.

If no errors occur, you can continue to copy portions of the virtual screen (presumably new data has been written to it) by pressing the SHIFT + OPTION keys. One copy will be written to your specified device each time you press these keys. *Chameleon* ignores any characters sent to it while it's dumping the screen.

Until you return to *Chameleon*'s menu, the ftp file isn't really closed. Thus, some data might not yet have been written out to your device. Always return to the menu before using the SYSTEM RESET key or before turning off your computer to end your session with *Chameleon*, so that the operating system can update its buffer and save all your data.

Transferring a file between two ATARI Home Computers

The steps for transferring information from one ATARI Home Computer to another are as follows.

1. Establish a telephone connection between the two computers. One agrees to be the sender, the other the receiver. Each places his or her phone receiver in the modem cups and loads *Chameleon* into RAM.

2. The receiver sets up her system to DOWNLOAD with FILE TYPE BINARY.

3. The sender sets up his system to UPLOAD with FILE TYPE BINARY and the file to be sent specified for F (line 8 of the menu).

4. The receiver presses the SHIFT + OPTION keys and types an asterisk (*).

5. When the sender sees the asterisk, he presses the SHIFT + OPTION keys.

6. Transfer of the file should then begin. When it finishes, the receiver types any character to stop, thanks the sender, presses the OPTION key to go to menu mode (which also closes the downloaded file), and disables FTP.

7. Both parties hang up their phones.

Notes

Both parties should exercise good judgment in transferring only files whose copyright allows for such transfer.

The flow control FTP used by *Chameleon* is extremely general, but also extremely trusting. If you have poor quality communication lines, some characters will probably be lost or changed during transmission. For a text file, these occurrences are of little consequence. For a binary file, however, it's very important that all characters transmit exactly as is. For these files, make sure to set INPUT PARITY and OUTPUT PARITY to NONE.

For serious use of FTP over low-grade lines, you really want a checksummed file transfer protocol, like the MODEM protocol described in the next chapter.

MODEM FTP

Introduction

MODEM is a public domain protocol for transferring files between many kinds of computers. It was designed by Ward Christensen for use with CP/M microcomputers. Over the years it has become the standard file transfer protocol for personal computers because:

- it defines a standard format for text files
- it detects and corrects errors in transmission, and
- it can transfer all eight bits of a binary file.

Chameleon uses MODEM protocol when the FTP item reads MODEM UP or MODEM DOWN. MODEM UP sends files from the ATARI Computer to the remote computer, and MODEM DOWN receives files the remote computer sends to the ATARI Computer.

What kinds of files can be MODEMed

The MODEM protocol is designed around the CP/M file format, which is much simpler than the ATARI DOS-II file format. On a CP/M system, text and binary files are stored in exactly the same way, and all files are a multiple of 128 characters long. Text files end at the first ^Z in the file, while binary files start loading at 100 hex and end on a 128-byte boundary.

The ATARI DOS-II file format, on the other hand, stores files as any number of bytes, with special "magic number" bytes at the start of the file. The ATARI DOS-II file system is "smarter" than the CP/M file system. While this is a good thing for ATARI Computer users, it does mean that some ATARI files have no CP/M counterpart.

Chameleon tries its best to convert ATARI files to and from CP/M files, but there are some restrictions. Chameleon's MODEM **can**:

- send files
- receive files
- send ATARI DOS-II binary files
- receive ATARI DOS-II binaray files

Chameleon's version of the MODEM protocol **can't**:

- send a text file with a control-z in it (this is a CP/M problem)
- receive a CP/M binary file (this is also a CP/M problem)

Since CP/M binary files contain machine language for the 8080/Z80 microprocessor, which the 6502-based ATARI can't execute anyway, these restrictions aren't very important.

MODEM screen messages

The MODEM protocol divides the file being transferred into small parts, called "blocks". Each block is sent with a number (from 0 to 255, starting with 1 and wrapping from 255 to 0) and some other characters that help the receiver detect any errors that might have occurred in transmission. If there are no errors, an ACK (short for "acknowledge") code is sent to the sender. If there are errors, then an NAK (short for "negative-acknowledge") is sent, and the sender will retransmit the block. More details of the protocol can be found in the MODEM appendix.

While a MODEM file transfer is in progress, *Chameleon* displays a status message telling you how the transfer is progressing. It's general format is:

For MODEM UP

GOT ACK 4 BLOCK *###* — if block *###* was sent without error

GOT NAK 4 BLOCK *###* — if block *###* was sent incorrectly

For MODEM DOWN

SENT ACK 4 BLOCK *###* — if block *###* received without error

SENT NAK 4 BLOCK *###* — if block *###* was received incorrectly

Figure 3. MODEM FTP Status Messages

If there are more than ten errors in a row, *Chameleon* gives up and returns to terminal mode, informing the user, ABORT:TOO MANY NAKS.

MODEM uploading

To upload a file to another computer using the MODEM protocol, follow these steps.

1. Establish a connection with the system, log in, and run that system's remote-MODEM program. Specify that the remote system is going to RECEIVE the file.

2. Press the OPTION key to go into *Chameleon*'s menu mode. Set the "F" item to the file you want to upload. Change FILE TYPE to the type of the file you want to upload. Set FTP to MODEM UP.

3. Press the START key. *Chameleon* will open your file for reading and return to CRT TTY mode. If *Chameleon* displays errors, fix them and try again. When *Chameleon* displays the NO ERRORS IN SETUP message, you can continue.

4. Press the SHIFT + OPTION keys to start the MODEM uploading.

5. *Chameleon* continues to upload until one of the following events occurs.

   a. It comes to the end of the file, whereupon it prints QUITTING TO TTY MODE and returns you to terminal mode.

   b. You type any character on the keyboard, whereupon it prints QUITTING TO TTY MODE and returns you to terminal mode.

   c. A file I/O error occurs, whereupon it prints FILE I/O ERROR WAS : # # #, where "# # #" represents an error code, and returns you to terminal mode.

   d. It fails to transmit a block more than ten times, whereupon it prints ABORT: TOO MANY NAKS and returns you to terminal mode.

6. The remote MODEM program knows when the file transmission is over, so it will automatically save the file and exit.

MODEM downloading

The steps for downloading a file with the MODEM protocol are as follows.

1. Run the remote system's remote MODEM program, asking it to SEND the file you want.

2. Press the OPTION key to go to *Chameleon*'s menu mode.

3. Select F and enter the file name. Select the appropriate FILE TYPE option (probably TEXT) and set FTP to MODEM DOWN.

4. Press the START key. If *Chameleon* displays a file error message, fix the problem and try again.

5. When you have successfully set up the FTP, press SHIFT + OPTION.

6. Downloading will begin. It will continue until one of the following events occurs.

   a. You type another character, whereupon *Chameleon* types QUITTING TO TTY MODE and returns you to terminal mode.

   b. A file I/O error occurs. *Chameleon* displays a FILE I/O ERROR WAS: # # # message, where "# # #" is an OS error code.

   c. *Chameleon* receives ten bad blocks in a row, in which case it displays ABORT: TOO MANY NAKS and returns you to terminal mode.

   d. The file transfers successfully, in which case *Chameleon* prints QUITTING TTY MODE and returns you terminal mode.

7. The download file isn't closed until you return to the menu, so it's a good idea to return to the menu and set the FTP item to NONE. That way your file will be saved *and* you won't accidentally reopen it.

MODEMing a file between two ATARI Home Computers

The steps for transferring information from one ATARI Home Computer to another using MODEM protocol are as follows.

1. Establish a telephone connection between the two computers. One agrees to be the sender, the other the receiver. Each places his or her phone receiver in the modem cups and loads *Chameleon* into RAM.

2. The receiver sets up her system with FTP as MODEM DOWN, FILE TYPE as BINARY and F as the name of the file to receive.

3. The sender sets up his system with FTP as MODEM UP, FILE TYPE as BINARY, and F as the name of the file to be sent.

4. The receiver types "Ready to receive file" and presses the SHIFT + OPTION keys.

5. When the sender sees that the receiver is ready, he presses the SHIFT + OPTION keys.

6. Transfer of the file should then begin. When it finishes, the receiver should press OPTION to enter the menu (this saves the last of the file), set FTP to NONE, and press START to return to TTY mode.

7. Both parties hang up their phones.

ATARI <—> CP/M file conversion

This section is for advanced users interested in the internal format of the ATARI MODEM file format — details of the actual MODEM protocol are in the MODEM appendix.

File type text

On MODEM UP, ATARI text files are converted into CP/M text files as follows:

1. All EOLs ($9b) are converted into CR($0d)/LF($0a) pairs

2. All ATARI-TABs ($7f) are converted into ASCII-TABs ($09)

3. The end of the file is padded out to a 128-byte boundary with control-Zs. If the file happens to end on a 128-byte boundary, an additional 128-byte block of control-Zs is sent.

On MODEM DOWN, CP/M text files are converted into ATARI text files as follows:

1. All CRs ($0d) are converted into EOLs ($9b)

2. All LFs ($0a) are removed from the file

3. All ASCII TABs ($09) are converted into ATARI TABS ($127)

4. The trailing control-Zs are removed from the file

File type binary

On MODEM UP, ATARI DOS-II files are converted into CP/M files as follows: The file is sent unaltered, except that the last block is padded out to a 128-byte boundary and the number of valid bytes in that block is inserted into the very last byte of that block.

If the ATARI DOS-II file ends on a 128-byte boundary, a junk block is appended with a "valid byte count" of zero.

On MODEM DOWN, CP/M binary files are assumed to be ATARI binary files that were uploaded by another *Chameleon* user. Thus, the last byte of the last block is used to determine how many of the bytes in the last block are actually valid. This means that an actual CP/M binary file would have some of its last bytes deleted, and is the source of *Chameleon*'s inability to receive CP/M binary files. Big deal. If your application demands the use of CP/M files, you might want to write a simple CP/M utility to post-pend 128 zero bytes to any particular file, or you could reassemble *Chameleon* to fit your requirements.

Customizing *Chameleon*

You can reassemble *Chameleon* if you want to customize the program to fit special needs. To modify *Chameleon*, you need:

- At least 48K RAM
- DOS II
- At least one ATARI 810 Disk Drive
- ATARI MACRO Assembler
- Some 6502 assembly language programming experience

The source for *Chameleon* resides in the following files:

D:EASYFTP.MAC
D:LIZARD.MAC
D:MENU.MAC
D:MODEMFTP.MAC
D:SCREEN.MAC
D:TERMINAL.MAC

To change a terminal type definition, you probably need to modify only TERMINAL. The code is commented, but it will be rough going . . . lots of luck!

With the new version of the source written out to the diskette (it's wise to save a copy of the original *Chameleon* in case you make a horrendous mistake), invoke AMAC and give it the command line D:LIZARD.MAC <RETURN>. AMAC should then assemble a file called D:LIZARD.OBJ which you can rename AUTORUN.SYS.

Chameleon contains the code of the traditional AUTORUN.SYS RS-232C loader file, so you don't need that file.

Known bugs and limitations

Any program as large as *Chameleon* is bound to have a few bugs lurking in its more remote corners — but I used it for a year prior to publishing it through APX without too many problems. Here's an informal list of the "rough edges" of *Chameleon*. You're welcome to try to fix them.

1. If you have a diskette version of *Chameleon*, then typing Q usually gets you back to DOS. Occasionally you will simply get a black screen. Turn off your computer and turn it on again.

2. If you SELECT the F line of the menu, the cursor will usually appear in the right place. Occasionally it will appear several lines lower in the menu. When this happens, simply press RETURN, and then SELECT the F line again.

3. MODEM UP and MODEM DOWN do not use the memory to buffer data — this slows down disk-based MODEM FTPing considerably. The only reason for not buffering was to keep cassette-based users from exceeding the 100 second time-out inherent in the MODEM protocol. Disk users who never plan to use the cassette may wish to change this by modifying and reassembling the MODEMFTP file.

4. You can't get a disk directory, delete, or rename files from within *Chameleon*.

Character and Control Codes

Following are the character and control codes for each terminal type *Chameleon* emulates.

Table 3. The Keyboard

Type	To get	ASCII Code
CTRL-COMMA	^	0
CTRL- letter	^<letter>	1-26
TAB	^ I (TAB)	9
RETURN	^ M (CR)	10
ESC	^ [(ESC)	27
CTRL-UP ARROW	^\	28
CTRL-DOWN ARROW	^]	29
CTRL-LEFT ARROW	^^	30
CTRL-RIGHT ARROW	^ -	31
CTRL-PERIOD	))	96
CTRL-SEMICOLON	~	123
SHIFT-CLEAR	{	125
CTRL-DELETE	}	126
BACK S	RUB OUT	127

The following characters are typed exactly as they are in ATASCII:

```
SPACE  !  "  #  $  %  &  '  (  *
       -  .  /  0  1  2  3  4  5
       6  7  8  9  :  <  =  >  ?  @  +  A  B  C  D  E  F  G  H  I
       J  K  L  M  N  O  P  Q  R
       S  T  U  V  W  X  Y  Z  [
       ]        .  a  b  c  d  e
       f  g  h  i  j  k  l  m  n
       o  p  q  r  s  t  u  v  w
       x  y  z  |  ,  :  ;  (  )
```

When you have OUTPUT PARITY set to NONE, the ATARI key controls whether the high bit is ON or OFF (most Operating Systems don't care).

Characters such as CTRL-4 or CTRL-SHIFT-<any key>, which aren't legal in ATASCII (or ASCII), cause *Chameleon* to lock until you type a legal character.

Control codes

TEST terminal

The TEST terminal type simply prints all characters it receives, including control codes. Bytes with the high bit set appear in inverse video. Control characters appear as their ATASCII equivalents (e.g., ASCII nulls show up as little hearts and CTRL-P as a little club).

GLASS terminal

The GLASS TTY terminal type prints all characters except ASCII 0-31 and ASCII 127. Most control codes are ignored. The following, however, are recognized:

Table 4. Control Codes for GLASS TTY

ASCII name	Decimal value	Action
^ G	7	Flash visual bell
^ H	8	Backspace cursor without erasing character under cursor
^ I	9	Tab to next 8-column tab stop
^ J	10	Move cursor down one line
^ L	12	Clear screen, home cursor
^ M	13	Move cursor to beginning of line

When the cursor is at the end of a line and a character is typed, the word and the cursor are "wrapped" to the next line.

ADM-3A terminal

The ADM-3A terminal prints the same characters as the GLASS TTY, but it handles a different set of control characters. They are as follows:

Table 5. Control Characters for ADM-3A

ASCII name	Decimal value	Action
^ G	7	Flash visual bell
^ H	8	Move cursor left one space; don't erase character under cursor
^ J	10	Move cursor down one line
^ K	11	Move cursor up one line
^ L	12	Move cursor right one character
^ M	13	Move cursor to beginning of current line
^ Z	26	Clear screen, home cursor
^ [+ < Y+32 ><X+32 >	27, etc.	Set cursor to (X,Y)
^^	30	Home cursor

VT-52 Terminal

The VT-52 terminal prints the same characters as the GLASS TTY. It reacts to control characters the same as the GLASS, except when ESC (decimal 27) is received. *Chameleon* will treat the next character received specially: if it isn't in the following table, *Chameleon* ignores the character; if it is in the table, the action is as follows.

Table 6. Control Codes for VT-52

Letter following ESC	Decimal value	Action
A	65	Move cursor up one line
B	66	Move cursor down one line
C	67	Move cursor right one column
D	68	Move cursor left one column
H	72	Home cursor to (0,0) (top left of screen)
I	73	Reverse line feed—moves cursor up one line. Top of screen scrolls down one line.
J	74	Clear from cursor to end of screen
K	75	Clear from cursor to end of line
Y <Y+32><X+32>	89 <Y+32> < X+32>	Move cursor to (X,Y)
Z	90	Sends ESC/Z to R device

Appendix 1

File I/O Error Codes

69 — *Chameleon* can't ftp from/to the E device

75 — *Chameleon* can't ftp from/to the K device

83 — *Chameleon* can't ftp from/to the S device

128 — BREAK abort

130 — Nonexistent device

136 — EOF

138 — Device timeout

139 — Device NAK

140 — Serial bus

142 — Overrun

143 — Checksum

144 — Device done

145 — Read after write compare error

160 — Disk number error

162 — Disk full

163 — Unrecoverable error

165 — File name error

167 — File locked

170 — File not found

Appendix 2

MODEM2 Protocol Specification

The following is the "official" specification of the MODEM2 file transfer protocol. Both the protocol and this appendix are in the public domain.

Modem protocol overview

1/1/82 by Ward Christensen. I will maintain a master copy of this. Please pass on changes or suggestions via CBBS/Chicago at (312) 545-8086, or by voice at (312) 849-6279.

Note. This does not include things which I am not familiar with, such as the CRC option implemented by John Mahr.

Definitions

<soh> 01H
<eot> 04H
<ack> 05H
<nak> 15H
<can> 18H

Transmission medium level protocol

Asynchronous, 8 data bits, no parity, one stop bit.

The protocol imposes no restrictions on the contents of the data being transmitted. No control characters are looked for in the 128-byte data message. Absolutely any kind of data may be sent - binary, ASCII, etc. The protocol has not formally been adopted to a 7-bit environment for the transmission of ASCII only (or unpacked-hex) data, although it could be simply by having both ends agree to AND the protocol-dependent data with 7F hex before validating it. I specifically am referring to the checksum, and the block numbers and their ones complement.

Those wishing to maintain compatibility of the CP/M file structure, i.e., to allow modeming ASCII files to or from CP/M systems should follow this data format:

- ASCII tabs used (09H); tabs set every 8
- Lines terminated by CR/LF (0DH 0AH)
- End-of-file indicated by ^ Z, 1AH (one or more)
- Data is variable length, i.e., should be considered stream of data bytes, broken into 128-byte chunks purely for the purpose of transmission.
- A CP/M "peculiarity": If the data ends exactly on a 128-byte boundary, i.e., CR in 127 and LF in 128, a subsequent sector containing the ^ Z EOF character(s) is optional, but is preferred. Some utilities or user programs still do not handle EOF without ^ Zs.
- The last block sent is no different from others; i.e., there is no "short block".

Message block level protocol

Each block of the transfer looks like:

<SOH> <blk #> <255-blk #> <—128 data bytes—> <cksum>

in which:

<SOH>	01 hex
<blk #>	binary number, starts at 01 increments by 1, and wraps 0FFH to 00H (not to 01)
<255-blk #>	blk # after going thru 8080 "CMA" instr, i.e., each bit complemented in the 8-bit block number. Formally, this is the "ones complement."
<cksum>	the sum of the data bytes only. Toss any carry.

File level protocol

Common to both sender and receiver

All errors are retried 10 times. For versions running with an operator (i.e., NOT with XMODEM), a message is typed after 10 errors asking the operator whether to "retry or quit".

Some versions of the protocol use <can> , ASCII ^ X, to cancel transmission. This was never adopted as a standard, as having a single "abort" character makes the transmission susceptible to false termination due to an <ack> <nak> or <soh> being corrupted into a <can> and canceling transmission.

The protocol may be considered "receiver driven"; that is, the sender need not automatically retransmit, although it does in the current implementations.

Receive program considerations

The receiver has a 10-second timeout. It sends a <nak> every time it times out. The receiver's first timeout, which sends a <nak> , signals the transmitter to start. Optionally, the receiver could send a <nak> immediately, in case the sender was ready. This would save the initial 10-second timeout. However, the receiver *must* continue to timeout every 10 seconds in case the sender wasn't ready.

Once into a receiving block, the receiver goes into a one-second timeout for each character and the checksum. If the receiver wishes to <nak> a block for any reason (e.g., invalid header, timeout receiving data), it must wait for the line to clear. See "programming tips" for ideas.

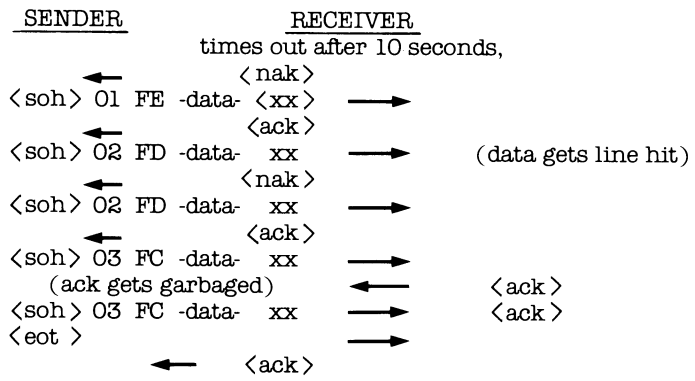
Synchronizing: If a valid block number is received, it will be: (1) the expected one, in which case everything is fine; or (2) a repeat of the previously received block. This should be considered OK, and only indicates that the receiver's <ack> got glitched, and the sender retransmitted. Any other block number indicates a fatal loss of synchronization, such as the rare case of the sender getting a line-glitch that looked like an <ack> . Abort the transmission, sending a <can> .

Sending program considerations

While waiting for transmission to begin, the sender has only a single very long timeout, say one minute. In the current protocol, the sender has a 10-second timeout before retrying. I suggest *not* doing this, and letting the protocol be completely receiver-driven. This will be compatible with existing programs. When the sender has no more data, it sends an <eot> , and awaits an <ack> , resending the <eot> if it doesn't get one. Again, the protocol could be receiver-driven, with the sender only having the high-level 1-minute timeout to abort.

Data flow example including error recovery

Here's a sample of the data flow, sending a 3-block message. It includes the two most common line hits - a garbaged block, and an <ack> reply getting garbaged. <xx> represents the checksum byte.



Programming tips

The character-receiver subroutine should be called with a parameter specifying the number of seconds to wait. The receiver should first call it with a time of 10, then <nak> and try again, 10 times.

After receiving the <soh> , the receiver should call the character receive subroutine with a 1-second timeout, for the remainder of the message and the <cksum> . Since they are sent as a continuous stream, timing out of this implies a serious like glitch that caused, say, 127 characters to be seen instead of 128.

When the receiver wishes to < nak > , it should call a "PURGE" subroutine, to wait for the line to clear. Recall the sender tosses any characters in its UART buffer immediately upon completing sending a block, to ensure no glitches were misinterpreted.

The most common technique is for "PURGE" to call the character receive subroutine, specifying a 1-second timeout, and looping back to PURGE until a timeout occurs. The < nak > is then sent, ensuring the other end will see it.

You may wish to add code recommended by John Mahr to your character receive routine to set an error flag if the UART shows framing error, or overrun. This will help catch a few more glitches the most common of which is a hit in the high bits of the byte in two consecutive bytes. The <cksum> comes out OK since counting in 1-byte produces the same result of adding 80H + 80H as with adding 00H + 00H.

APX
ATARI Program Exchange
P.O. Box 3705
Santa Clara, CA 95055

Review Form

We're interested in your experiences with APX programs and documentation, both favorable and unfavorable. Many of our authors are eager to improve their programs if they know what you want. And, of course, we want to know about any bugs that slipped by us, so that the author can fix them. We also want to know whether our instructions are meeting your needs. You are our best source for suggesting improvements! Please help us by taking a moment to fill in this review sheet. Fold the sheet in thirds and seal it so that the address on the bottom of the back becomes the envelope front. Thank you for helping us!

1. Name and APX number of program.

2. If you have problems using the program, please describe them here.

3. What do you especially like about this program?

4. What do you think the program's weaknesses are?

5. How can the catalog description be more accurate or comprehensive?

6. On a scale of 1 to 10, 1 being "poor" and 10 being "excellent", please rate the following aspects of this program:

   _____ Easy to use
   _____ User-oriented (e.g., menus, prompts, clear language)
   _____ Enjoyable
   _____ Self-instructive
   _____ Use (non-game programs)
   _____ Imaginative graphics and sound

7. Describe any technical errors you found in the user instructions (please give page numbers).

8. What did you especially like about the user instructions?

9. What revisions or additions would improve these instructions?

10. On a scale of 1 to 10, 1 representing "poor" and 10 representing "excellent", how would you rate the user instructions and why?

11. Other comments about the program or user instructions:

From

STAMP

APX

ATARI Program Exchange
P.O. Box 3705
Santa Clara, CA 95055

[seal here]

CHAMELEON CRT TERMINAL EMULATOR

by John H. Palevich

- **Turn your ATARI Computer into a computer terminal**
- **Upload and download files**
- **Communicate with another ATARI Computer or with larger computer systems**

Chameleon gives you the best of both worlds — the instant, exclusive availability of a personal computer, and the added power of a network or larger computer system. *Chameleon* converts your ATARI Computer into a computer terminal emulating three widely used types, GLASS TTY, ADM-3A, and VT-52. The terminal emulator supports tabs, line feed, form feed, and a visual bell signal. For the ADM-3A and VT-52, it also supports cursor addressing and basic editing features.

Chameleon's major features are: (1) program uploading and downloading, using up to nine pages of memory as a buffer, (2) the ability to print all the buffer or to store it on diskette, (3) support of the CP/M standard MODEM protocol, (4) the full ASCII standard character set, (5) a 134-column (maximum), 24-line character display (displaying 40 columns at a time) with a lock cursor feature for automatic horizontal scrolling, (6) continuous transmission at baud rates of 75 to 1200 and non-continuous at baud rates of 2400 to 9600, (7) fine scrolling at some baud rates, and (8) selectable RS-232C ports and input/output parity. Assembly language programmers with 48K of memory, a disk drive, and the ATARI MACRO Assembler can even define new terminal types (the diskette version contains the source code).

Requires:

- ATARI 830™ Acoustic Modem or an RS-232C device
- ATARI 850™ Interface Module

Cassette (APX-10058)

- ATARI 410™ Program Recorder
- 24K RAM

Diskette (APX-20058)

- ATARI 810™ Disk Drive
- 32K RAM

Optional:

- ATARI printer or equivalent printer
- ATARI MACRO Assembler

About the author

JOHN PALEVICH

John Palevich of Cupertino, California, designed *Chameleon CRT Terminal Emulator* for a practical reason. When he was a student at Massachusettes Institute of Technology, he wanted to use the school's computer from his own room, rather than walk to the terminals located elsewhere on campus. It makes sense when you think about New England winters! He named his program "Chameleon" because it changes an ATARI Home Computer into a terminal. John is the author of several systems programs available through APX, including an implementation of the SMALL C language and a boot tape development system. Upon his graduation from M.I.T., John joined the Corporate Research Division of Atari in 1983.