



ACE80  
ACE80



# Eighty Column Screen Display for Atari Computers

## Technical Notes



**INTRODUCTION..... 3**

**ACE-80'S PRESENCE ..... 3**

**ACE-80 AND RAM..... 3**

**ACE-80'S CURSOR..... 4**

**ACE-80, START AND SELECT ..... 4**

**ACE-80'S USER ROUTINES..... 5**

**UTILIZING ACE-80 ..... 6**



## Introduction

ACE-80 is designed to be an eighty-column replacement for the screen editor built into the Atari's operating system (the OS). As such, it works very well with programs that call the OS for all their screen I/O. However, many useful programs bypass the OS and use their own screen handlers, usually to gain speed improvements. Other programs may make proper use of the OS but might conflict with ACE-80's memory requirements. These notes are for those who would attempt to alter such incompatible programs to make use of ACE-80.

## ACE-80's Presence

On power-up, ACE-80 hides a watchdog routine in page one of RAM. This routine looks at every OPEN E: or OPEN S: call to the OS and activates the eighty-column display when appropriate. SYSTEM RESET would kill the watchdog if ACE-80 didn't put its initialization address in the CASINI vector and set bit 1 of the BOOT? flag.

Some DOSes carelessly clear bit 1 of BOOT? when they set bit 0. Such a DOS can be patched.

Replace any code such as:

```
LDA #501  
STA $09
```

with something like:

```
LDA $09  
ORA #501  
STA $09
```

to allow ACE-80 to survive SYSTEM RESET.

## ACE-80 and RAM

The watchdog routine resides in locations \$0100 through \$0146 and must be undisturbed if ACE-80 is to remain available.

When and only when the eighty-column display is active, ACE-80 claims the top 12,688 bytes of RAM. In a system with no left cartridge and BASIC disabled, RAM from \$8E70 through \$BFFF is taken. If a left cartridge is present, or if BASIC is enabled, then RAM from \$6E70 through \$9FFF is taken. ACE-80 sets the OS variable MEMTOP (\$02E5-\$02E6) with the correct value.



Some programs don't look to MEMTOP for the top of available RAM. They must be patched. BASIC does look there and adjusts its program space accordingly.

When ACE-80 is told to OPEN an eighty-column display, it looks to the OS variable APPMHI (\$000E-\$000F) for the bottom of free memory. If there is not enough room for the display, it returns error #147.

### ACE-80's Cursor

ACE-80 normally displays its cursor only when the E: device is waiting for input. Some programs have their own keyboard input routines. To display the cursor, write \$C0 to the ANTIC register NMIEN (\$D40E). Be sure to turn the cursor off again before starting any serial I/O. To turn it off, write \$40 to NMIEN.

### ACE-80, START and SELECT

ACE-80 pays attention to the ACE-80, START and SELECT

ACE-80 pays attention to the START and SELECT buttons only when the E: device is waiting for input. The buttons can be examined at other times by calling the user routine mentioned in the next section. Otherwise, the program can control the screen colors and cursor mode itself.

ACE-80 variable FLASH (\$0120) controls the cursor mode. A value greater than \$80 gives a steady cursor, \$00 gives an invisible cursor, and other values provide different flashing cursors. The value \$20 gives the default flashing cursor.

ACE-80 variable BKCOLR (\$0121) holds the screen background color for the next OPEN call. CHCOLR (\$0122) holds the character color. The values are the usual Atari color codes. and **[SELECT]** buttons only when the E: device is waiting for input. The buttons can be examined at



other times by calling the user routine mentioned in the next section. Otherwise, the program can control the screen colors and cursor mode itself.

ACE-80 variable FLASH (\$0120) controls the cursor mode. A value greater than \$80 gives a steady cursor, \$00 gives an invisible cursor, and other values provide different flashing cursors. The value \$20 gives the default flashing cursor.

ACE-80 variable BKCOLR (\$0121) holds the screen background color for the next OPEN call. CHCOLR (\$0122) holds the character color. The values are the usual Atari color codes.

## ACE-80's User Routines

A few of ACE-80's routines are available to programs that bypass the OS for screen I/O. They provide a minimal-function eighty-column screen interface and are intended for programs that rewrite the entire portion of the screen that changes. In other words, they provide no special functions such as insert line, delete character, and so on. The only functions available are: initialize screen, print character at a certain location, and scroll screen up. One further routine processes the START and SELECT buttons. These routines are callable via JSR, and they destroy the contents of the A, X, and Y registers. Entry parameters, if any, are shown.

INIT (\$9400) sets up the eighty-column display and clears the screen. It does not enable the cursor. INIT has no entry parameters.

FCHLIN (\$9403) prepares pointers for a subsequent call to PRTCH2. On entry, the A register contains the number of the row in which a character is to be printed (0 is the top row; 23 is the bottom).

PRTCH2 (\$9406) prints a character on the screen at the row and column specified. FCHLIN must be called first with the row number. On entry to PRTCH2, Y contains the column number (0 is



the leftmost column; 79 is the rightmost). A contains the ATASCII code of the character to be printed. X must contain zero.

FCHLIN and PRTCH2 should be used together as in this example:

```
LDA #11 ;row number
JSR FCHLIN
LDY #39 ;column number
LDA #$A1 ;ATASCII
LDX #0
JSR PRTCH2
```

The example prints a reverse video exclamation point in the middle of the screen.

CLRLIN (\$9409) clears the row specified in the A register.

SCROLU (\$940C) scrolls the screen up one line. A contains a row number, and SCROLU scrolls up all the rows below that row and moves that row to the bottom.

This example scrolls the entire screen and produces a blank line at the bottom:

```
LDA #0
JSR CLRLIN
LDA #0
JSR SCROLU
```

CHKSWS (\$940F) reads the START and SELECT buttons and executes their functions. There are no entry parameters. Programs should call CHKSWS regularly to retain the color and cursor adjustment feature.

## Utilizing ACE-80

The best way for an application to use ACE-80's features is to follow OS protocols and address the E: device. As a substitute for the forty-column screen editor, ACE-80 is actually more than twice as fast as the OS routines. Plenty of documentation exists for interfacing with OS devices. The ultimate authority is the Operating System Manual in the Atari Technical Reference Notes.

Armed with that knowledge and the details in these notes, the programmer should be able to develop or upgrade sophisticated software for the Atari home computers. If the reader requires further technical information about ACE-80, they should contact Vintage Computer Center with specific questions.