Coleco Development Tools

Graphics Processor, Resident Debugger

&

Dual UART RS-232 External Interface

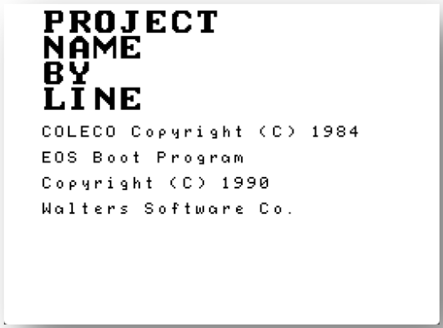
Technical Documentation and Analysis

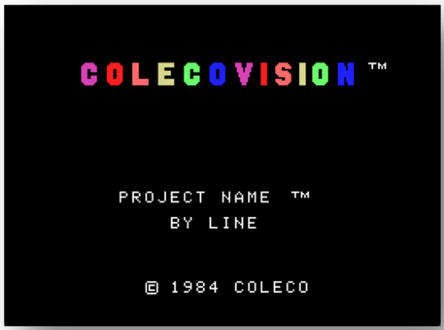
Document Version: 4.0
January 2026

Source: Firmware Reverse Engineering and Schematic Analysis

ColecoVision ADAM Archive
ColecoVisionADAM.com • AdamArchive.org

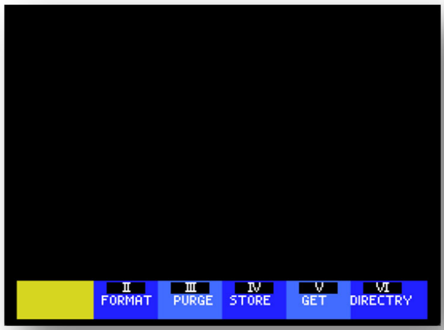
Table of Contents


PROJECT
NAME
BY
LINE
COLECO Copyright (C) 1984
EOS Boot Program
Copyright (C) 1990
Walters Software Co.


COLECOVISION ™

PROJECT NAME ™
BY LINE

© 1984 COLECO



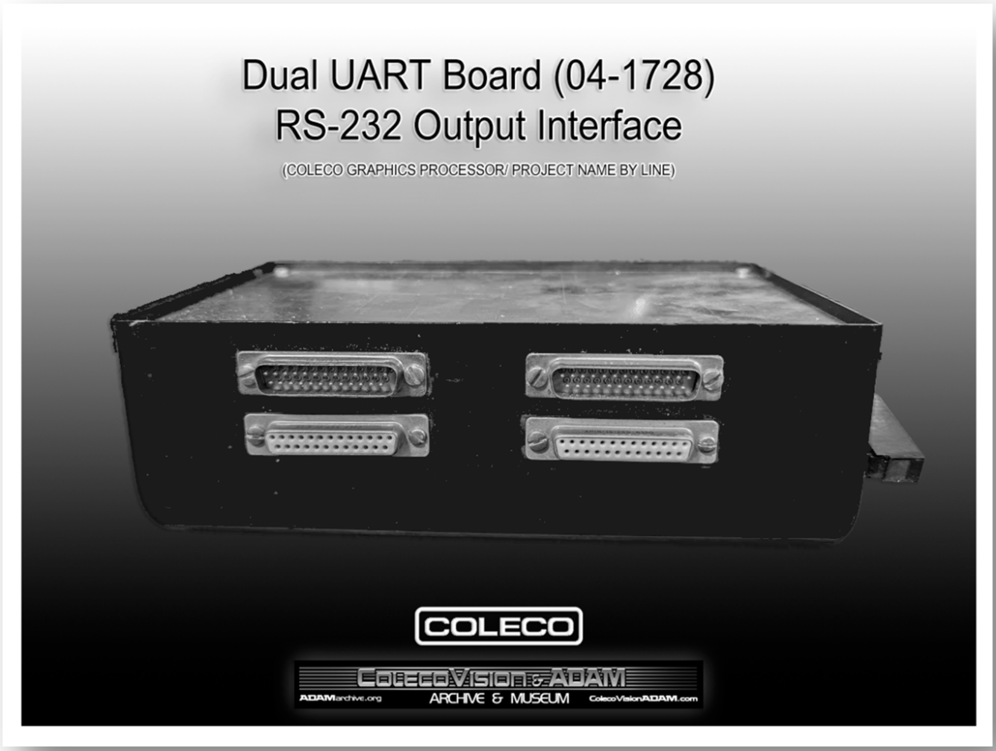
1. Executive Summary

This document covers professional development tools created by Coleco Industries in 1984 for internal use by game developers. These tools were used to create graphics and software for ColecoVision cartridges and ADAM computer products.

The development system consists of:

- Graphics Processor ROM (32KB) — Graphics creation, editing, and export
- Resident Debugger ROM (24KB) — Integrated development environment with debugging
- Dual UART RS-232 Interface Board — Hardware for serial communication
- Expanded RAM Disk Versions — Community-converted EOS bootable versions

These were internal development tools, not consumer products. Their preservation provides valuable insight into Coleco's software development process during the golden age of video games.



2. Historical Context

Understanding the era in which these tools were developed provides essential context for their significance.

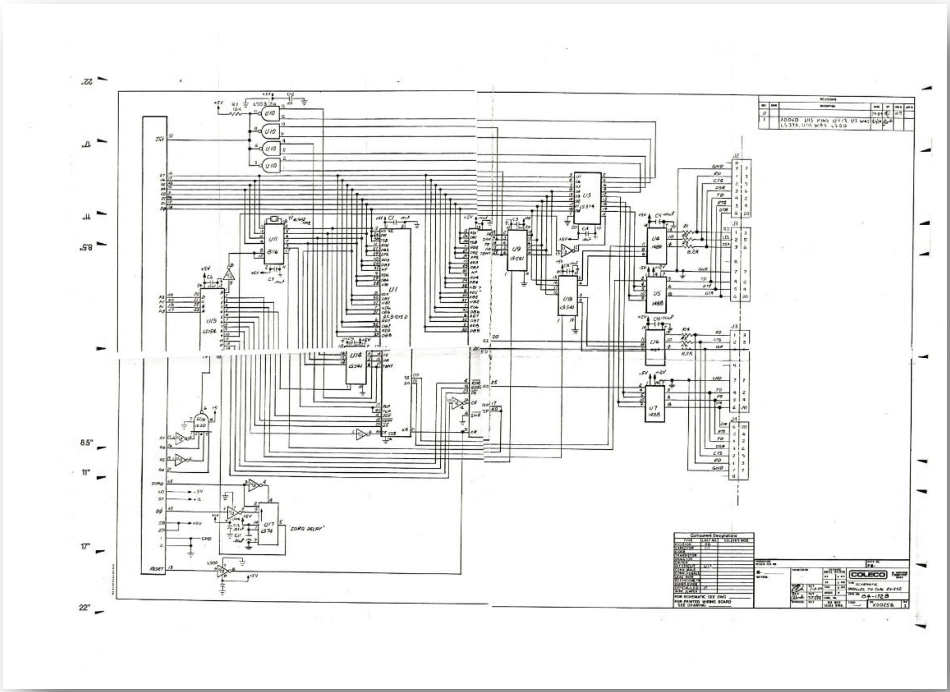
2.1 Timeline

Date	Event
August 1982	ColecoVision released
June 1983	ADAM Computer announced
October 1983	ADAM released to market
1984	Graphics Processor and development tools created internally
1984-1985	Peak game development period
January 1985	ADAM discontinued
1988	Coleco Industries files bankruptcy

2.2 Significance

The Graphics Processor represents professional-grade development tools from the early 1980s video game industry:

- VAX Integration — Indicates Coleco used DEC minicomputers for game development
- Intel HEX Support — Direct EPROM programming capability on the development floor
- Dual Serial Ports — Simultaneous output to multiple devices
- Complete Toolset — Pattern, sprite, screen, and object editing in one package



3. Software Tools

3.1 Graphics Processor ROM

Technical Specifications

Attribute	Value
Processor	Zilog Z80
ROM Size	32,768 bytes (32KB)
Load Address	$8000 - $FFFF
Entry Point	$9986 (varies by version)
ROM Signature	$55 $AA (Standard ColecoVision/ADAM)
Copyright	1984 Coleco Industries Inc.
Title String	"BY LINE / PROJECT NAME /1984"

ROM Versions

Four distinct versions have been identified, showing evolution from basic editor to full IDE:

Version	Entry	MD5 Checksum	Status
Original	$9986	fd51cdb2e619ed1b1a4f90ad442bd116	Base version
a1	$9986	831505090ec33617749b673017b9adc1	Padding cleanup
a2	$9AA6	848f5a9db15b210e431755968776f000	Scripting added
a3	$9FB8	d98a34f74327f129bbe614fe4ddd9b20	Full IDE

Feature Comparison by Version

Feature	Orig	a1	a2	a3
Graphics Editing	✓	✓	✓	✓
Drawing Tools	✓	✓	✓	✓
Intel HEX / VAX Export	✓	✓	✓	✓
RS-232 Configuration	✓	✓	✓	✓
TAPE Support	-	-	✓	✓
Programming Language	-	-	✓	✓
PIRATE Feature	-	-	✓	✓
TRACE / DEBUG Mode	-	-	-	✓
FIND / UNTIL / PAUSE	-	-	-	✓
SYMBOLS Table	-	-	-	✓

Version Details

Original / a1 — Base Graphics Editor

Full drawing and editing capabilities with Intel HEX and VAX/VMS export. DISK-based storage. Versions are functionally identical; a1 differs only in padding bytes ($00 vs $FF in unused ROM area).

Version a2 — Scripting Language Added

Major revision with built-in programming language. TAPE support replaces DISK. Added PIRATE feature. New entry point at $9AA6.

```
Keywords: begin, repeat, return, call, jump, quit, stop, thru, byte, word, data,
load, delay, process, trigger, show, sprites, bkgrnd, mode, overlay, softkey,
dupl
```

Version a3 — Full IDE with Debugging

Most advanced version with comprehensive debugging: TRACE mode, FIND command, UNTIL breakpoints, PAUSE execution, SYMBOLS table. Entry point at $9FB8.

```
Additional: TRACE, FIND, UNTIL, PAUSE, SYMBOLS, plot, print, color, erase, "High
address?", "Low address?", "Instruction?"
```

Functional Capabilities

Graphics Editing Modes:

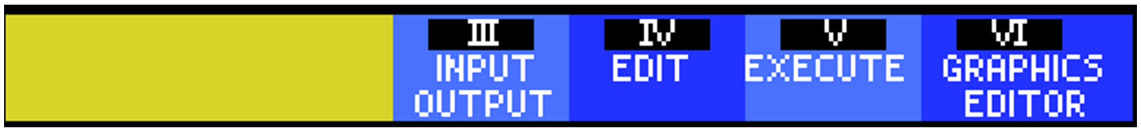
Mode	Function
Pattern Editor	Create/edit 8x8 pixel tile patterns
Sprite Generator Editor	Design sprite graphics (8x8 or 16x16)
Screen Layout Editor	Arrange patterns on screen (32x24 grid)
Color Editor	Set foreground/background colors per tile group
Object Definition	Create composite multi-sprite objects

Drawing Tools:

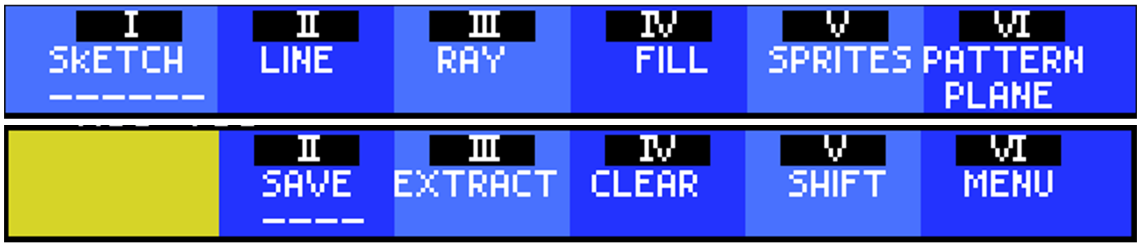
Tool	Description
SKETCH	Freehand pixel drawing
LINE	Bresenham line algorithm implementation
RAY	Ray/vector drawing
FILL	Flood fill algorithm

Menu Structure

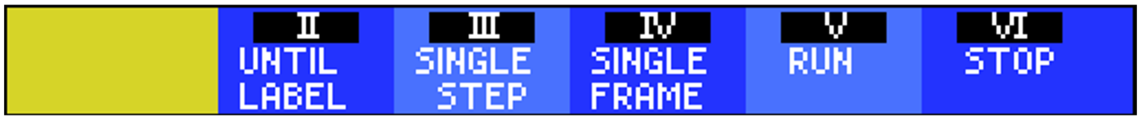
Main Menu:



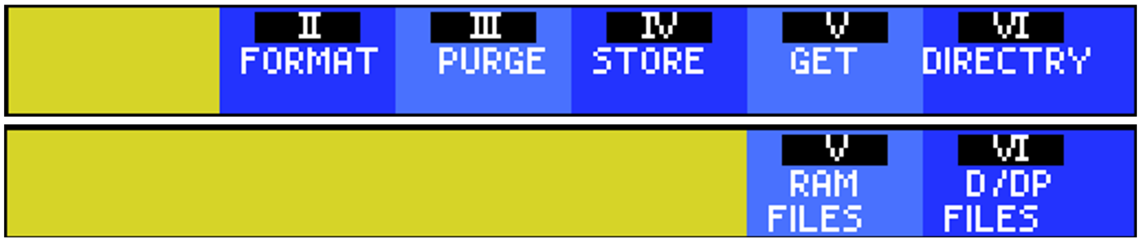
Graphics Sub-Menu:



Edit Sub-Menu:



File Sub-Menu



3.2 ADAM Resident Debugger Rev 2.0

A companion development tool combining graphics editing with debugging and testing capabilities. The "Resident" designation indicates it remained in memory during development sessions.

Specifications

Attribute	Value
ROM Size	24,576 bytes (24KB)
Load Address	$8000 - $DFFF
Entry Point	$A155
ROM Signature	$55 $AA (Standard ColecoVision/ADAM)
Header Title	"BY KONAMI / MONKEY ACADEMY" (placeholder)
Date	1984

Unique Debugger Features

Features present in the Resident Debugger but NOT in the Graphics Processor:

Feature	Purpose
MUSIC	Sound/music composition editor
SOFTKEYS	Programmable function key configuration
CONFIGURE	System configuration menu
LABEL	Symbol/label management for code debugging
ABSOLUTE/COMPLEX	Addressing and data structure modes
BUILD	Code assembly/build function
GROUP	Element grouping operations
WHERE	Memory/code location finder
FROM RAM / TO RAM	Direct memory transfer operations
ENABLE / DISABLE	Feature toggle controls

Comparison: Graphics Processor vs Resident Debugger

Feature	Graphics Processor	Resident Debugger
ROM Size	32KB	24KB
Graphics Editing	Full	Full
Intel HEX Export	Yes	No
VAX/VMS Output	Yes	No
Code Debugging	No	Yes
Music Editor	No	Yes
Softkey Config	No	Yes
Label Support	No	Yes

Development Workflow: Graphics Processor was used by artists for final graphics creation and export. Resident Debugger was used by programmers for integrated development, testing, and debugging with graphics preview capability.

3.3 Expanded RAM (ExpRAM) Disk Versions

Community-created disk versions allowing the Graphics Processor to run from disk/tape media instead of ROM cartridge.

Disk Images

Disk	Size	Contents
ExpRAM	160KB	Documentation + Conversion Utilities
ExpRAM a1	160KB	EOS-bootable Graphics Processor program

Community Contributors

Author	Organization	Contribution
Jim Walters	Walters Software Co.	EOS boot conversion, boot screen
Solomon Swift	Phoenix 2000	DaVINCI/C.G.P. Converters
Steve Pitman	Pitman Software	SmartBASIC V1.0 Loader
Ron Collins	Akron AUG BBS	Comprehensive Documentation

ExpRAM Enhancements

Features added in the disk version:

- SELFBOOT — EOS self-booting capability
- CRC Error Checking — Data integrity verification
- FILE EXISTS Protection — Overwrite confirmation
- GAME SAVE Integration — Compatible with GAME SAVE utility

Historical Note: These disks were distributed freely via BBS networks, including the Akron AUG BBS (216-882-4720). They represent the ADAM community's effort to make Coleco's internal tools accessible to hobbyists.

4. Hardware

4.1 Dual UART RS-232 Interface Board

Specifications

Attribute	Value
Coleco Drawing Number	04-1728
Part Number	700656
Function	Dual serial port interface
Bus Interface	Z80 I/O mapped
UART Type	AY-5-1015D (x2)
RS-232 Drivers	1488 (x2)
RS-232 Receivers	1489 (x2)

Component List

Ref	Part Number	Function
U1	AY-5-1015D	UART 1 - Primary serial port
U3	74LS374	Octal D flip-flop - Status latch
U4	1488	RS-232 line driver
U5	1488	RS-232 line driver
U6	1489	RS-232 line receiver
U7	1489	RS-232 line receiver
U8	74LS541	Octal buffer - Data bus
U9	74LS541	Octal buffer - Data bus
U11	-	Baud rate generator
U14	74LS541	Octal buffer - Data bus
U15	74LS154	4-to-16 line decoder

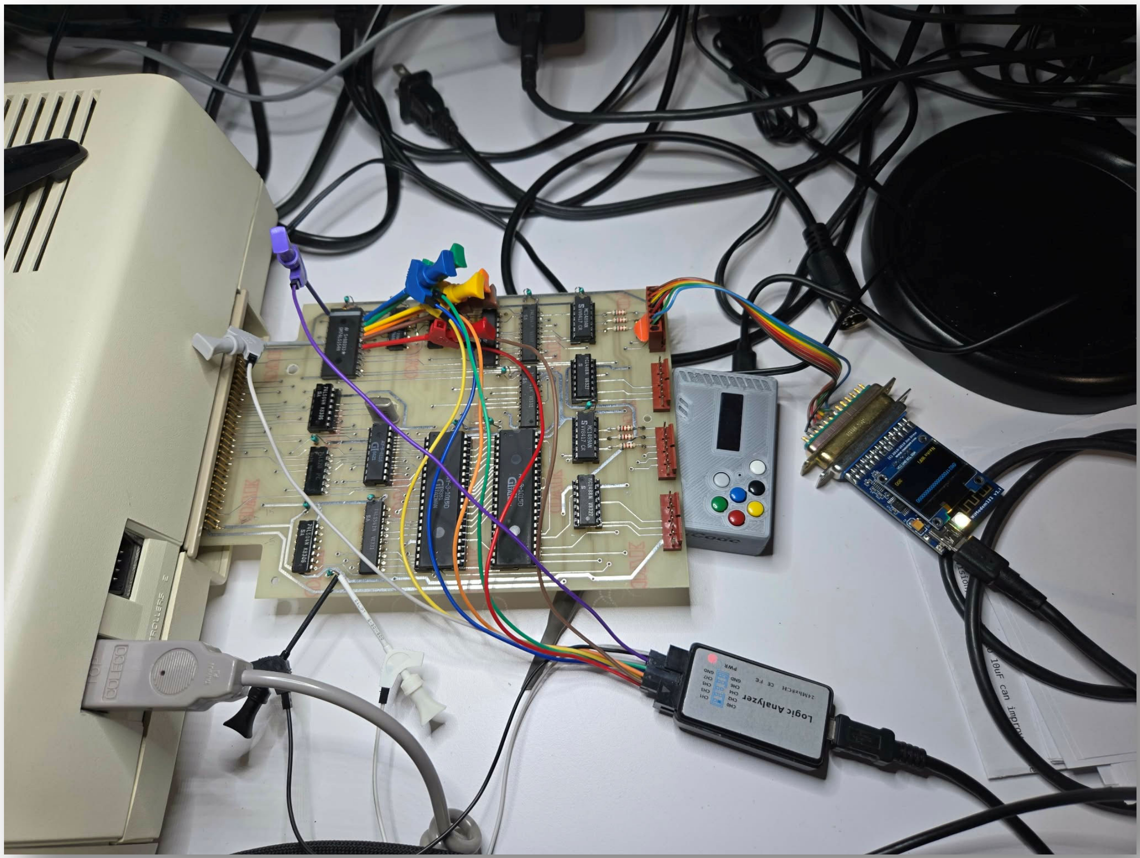
I/O Port Address Decoding

Decode Logic: Active when A4-A7 are low during /IORQ. Base address $80, sequential ports.

Port	Decoder	Binary	Function
$80	0	00001010	Baud rate generator set
$81	1	10001010	UART 2 status read (PE, DAV, FE, OR, TBMT)
$82	2	01001010	/RDE - UART 1 Receive Data Enable
$83	3	11001010	DSR/CTS modem status enable
$84	4	00101010	UART 1 status read (PE, DAV, FE, OR, TBMT)
$85	5	10101010	/RDAV - UART 1 Reset Data Available flag
$86	6	01101010	/DS - UART 1 Data Strobe (transmit byte)
$87	7	11101010	/CS - UART 1 Control Strobe
$88	8	00011010	/RDE - UART 2 Receive Data Enable
$89	9	10011010	Modem control latch (TD, RTS, DTR signals)
$8A	10	01011010	/RDAV - UART 2 Reset Data Available flag
$8B	11	11011010	/DS - UART 2 Data Strobe (transmit byte)
$8C	12	00111010	/CS - UART 2 Control Strobe

Signal Definitions

Signal	Full Name	Description
PE	Parity Error	Received byte failed parity check
DAV	Data Available	Receive buffer contains valid data
FE	Framing Error	Stop bit not detected at expected time
OR	Overrun	New data arrived before previous was read
TBMT	Transmit Buffer Empty	Transmitter ready for next byte
/RDE	Receive Data Enable	Gate received data onto bus
/RDAV	Reset Data Available	Clear DAV flag after reading data
/DS	Data Strobe	Load byte into transmit buffer
/CS	Control Strobe	Write to UART control register



5. Serial Communication & Data Transfer

This section explains how data was actually transferred between the Graphics Processor and external systems.

5.1 Software Port Implementation

Important Discovery: ROM analysis reveals the Graphics Processor uses ports in the $50-$59 range, NOT the Dual UART board's documented $80-$8C range.

Port	Direction	Function
$50	OUT	UART control
$52	IN	UART status
$54	IN	UART status/data
$55	IN	UART data receive
$56	OUT	UART data transmit
$59	OUT	UART control

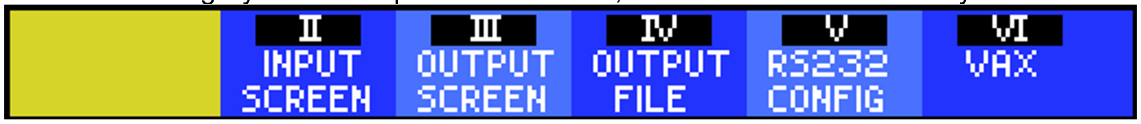
Implications:

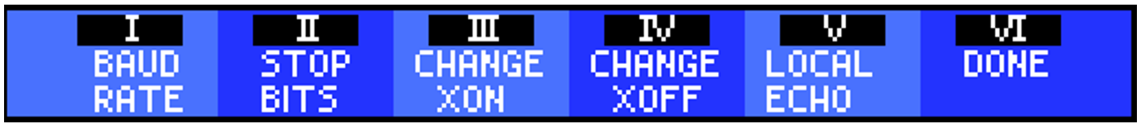
1. The Dual UART Board may have been designed to match these software port addresses
2. The ROM can be patched to work with standard Orphanware serial cards
3. Port addresses are similar to Orphanware-style serial interface

5.2 Actual Communication Workflow

Data was transmitted via OUTPUT FILE (SmartKey III), not the VAX key:

1. RS232 CONFIG (SmartKey IV) — Configure format (Intel HEX), protocol (XON/XOFF), baud rate
2. OUTPUT FILE (SmartKey III) — Transmit graphics data as Intel HEX ASCII text via RS-232
3. Receiving System — Captures text stream, converts Intel HEX to binary for use





5.3 Output Formats

Format	Status	Description
Intel HEX	Working	Standard ASCII format for EPROM programmers
I_CODE	Working	Internal project format
SmartBASIC	Working	DATA statements for ADAM BASIC programs
VAX	Stub	Never implemented - intended for VAX-specific format

5.4 VAX Function Status

The "VAX" SmartKey option is a stub function (just a RET instruction). However, this does NOT mean the unit couldn't communicate with mainframes.

Bottom Line: Coleco developers transferred data to VAX systems using Intel HEX format via RS-232, then converted it on the VAX side. The VAX SmartKey was probably intended for a native VAX format that was never coded.

5.5 Firsthand Technical Verification

Richard F. Drushel, Ph.D. provided firsthand verification of the serial features in the mid-1990s:

- Disassembled the CGP cartridge (aka "Project Name By Line")
- Discovered software was looking for Orphanware-type serial port at different base address
- Patched all port references using Norton Utilities block editor
- Connected via null modem to Tandy 2800HD laptop
- Captured Intel HEX output using Procomm Plus terminal with ASCII capture
- Successfully converted graphics data for use in game projects

Dr. Drushel confirmed that the "VAX" SmartKey option did nothing - it was just a RET instruction, which ROM analysis has verified.

Note: Dr. Drushel's patched binary was unfortunately lost in a disk crash and was never recreated.

6. Preservation Notes & Contributors

6.1 Source Materials

Item	Status
Graphics Processor ROM (32KB) - 4 versions	Preserved and analyzed
ADAM Resident Debugger ROM (24KB)	Preserved and analyzed
ExpRAM Disk Images (2 disks)	Preserved
Dual UART Schematic (04-1728)	Documented from original
Port decode information	Reverse engineered and validated

6.2 Documentation Produced

Document	Contents
This technical document	System overview and specifications
Disassembly listing	Annotated Z80 source code
Memory map	ROM organization
I/O port table	Hardware interface details

6.3 Contributors and Acknowledgments

Technical Analysis and Firsthand Knowledge:

Contributor	Contribution
Richard F. Drushel, Ph.D.	Firsthand technical verification, disassembly work, VAX stub confirmation, successful Intel HEX export testing

ExpRAM Disk Community Contributors (1980s-1990s):

Contributor	Organization	Contribution
Jim Walters	Walters Software Co.	EOS boot conversion, boot screen
Solomon Swift	Phoenix 2000	DaVINCI/C.G.P. Converters
Steve Pitman	Pitman Software	SmartBASIC V1.0 Loader
Ron Collins	Akron AUG BBS	Comprehensive documentation

Preservation and Analysis:

Organization	Contribution
ColecoVision ADAM Archive	ROM preservation, schematic documentation, firmware analysis, this document

Appendix A: Original User Documentation

The following documentation was written by Ron Collins, SYSOP of the Akron AUG BBS, and was distributed on the ExpRAM disk. It represents the only known user documentation for the Graphics Processor and provides invaluable operational instructions.

A.1 Introduction

"First, let me say that this program originated on a Coleco game cartridge. Somebody used the CP/M GAMESAVE.COM program to create this version of the software - a version that is able to be sent by modem or mail to just about anywhere in the world. Somebody else patched it to access disk drive one rather than your first digital data pack."

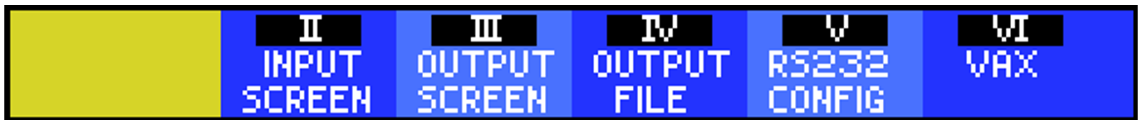
"What is the GRAPHICS PROCESSOR? This is a rather complex and sophisticated drawing program (for cartridge sized programs!) that many believe Coleco used to create all those nice looking ColecoVision games and Adam Super Games."

A.2 Loading the Program

To boot the program, pull the <COMPUTER RESET> switch on your ADAM Memory Console after inserting the data pack or disk into drive #1. Your Adam will load the C.G.P. into memory and execute it. Your screen will clear, turn black and come up with the COLECOVISION name in its multi-colored game format.

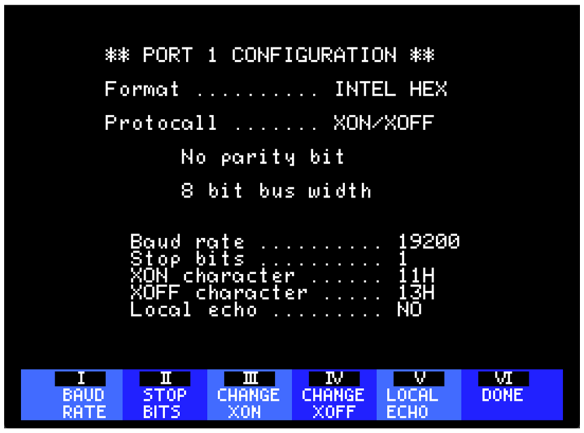
A.3 Main Menu Structure

At load time, you are presented with the SmartKey options:



- SmartKey II (INPUT SCREEN) — Reads in a screen, shows "READING IN SCREEN"
- SmartKey III (OUTPUT SCREEN) — Writes screen data with HEX counter
- SmartKey IV (OUTPUT FILE) — Prompts for filename to output
- SmartKey V (RS232 CONFIG) — Serial port configuration
- SmartKey VI (VAX) — "Does nothing at this time" (per original docs)

A.4 RS-232 Configuration Screen

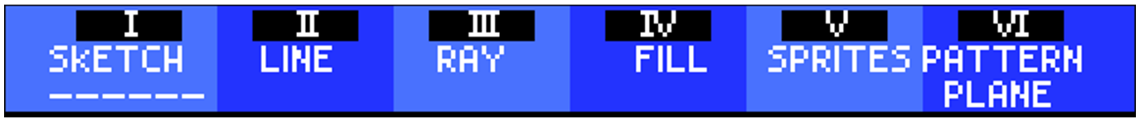


```
** PORT 1 CONFIGURATION **
Format .......... INTEL HEX
Protocall ....... XON/XOFF
      No parity bit
      8 bit bus width

  Baud rate .......... 19200
  Stop bits .......... 1
  XON character ...... 11H
  XOFF character ..... 13H
  Local echo ......... NO
```

A.5 The Graphics Editor

SmartKey VI (GRAPHICS EDITOR) provides the main drawing area:



SKETCH: Press HOME key to draw one white pixel. Hold left firebutton and move joystick for continuous line. Mouse also supported.

LINE: Move cursor to start, press fire. Move to end, press fire. Press third time to draw.

RAY: Draw sunbeam patterns. First point is apex, subsequent points connect back to create rays.

FILL: Flood fill between vertical lines. Move cursor between lines and press firebutton.

A.6 Keyboard Shortcuts

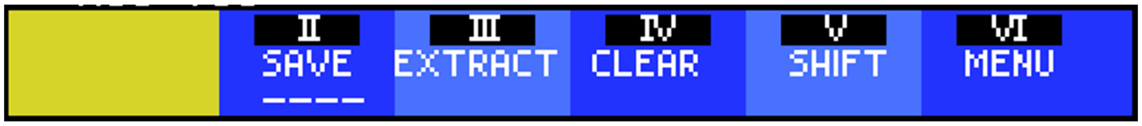
Key	Function
DELETE	Toggle erase mode (cursor turns red)
CLEAR	Access screen clearing options
INSERT	Change foreground/background colors
WILDCARD	Enter ZOOM mode for enlarged editing
MOVE/COPY	Access powerful move/copy features
STORE/GET	File operations (save/load)
HOME	Draw single pixel at cursor location
ESC	Return to previous menu

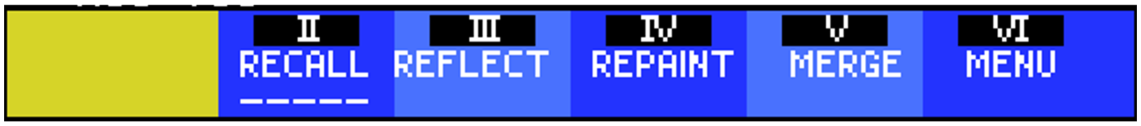
A.7 ZOOM Mode (WILDCARD Key)

Press WILDCARD key for enlarged editing. Cursor becomes corner brackets marking edit area. Move with joystick, press RETURN or fire button to select area. Screen shows enlarged copy for detailed pixel editing. Full drawing tools remain available in zoomed view.

A.8 MOVE/COPY Features

Press MOVE/COPY key for powerful editing operations:





Option	Description
SAVE	Capture screen area to memory using bracket cursor to define region
EXTRACT	Remove selected area from screen (leaves rectangular void), store for recall
RECALL	Stamp saved image at cursor location (flashing bracket shows placement)
REFLECT	Create mirror/negative image at new location
REPAINT	Change colors of placed graphics
SHIFT	Move graphics within defined area
CLEAR	Clear selected area

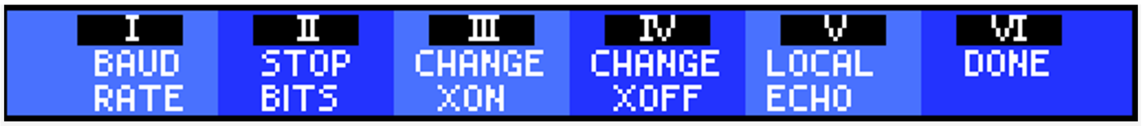
Note: *"For those of you familiar with POWERPAINT (copyright DEI), the move and copy features of that program, as powerful as they are, will pale by comparison to those of the GRAPHICS PROCESSOR!"*

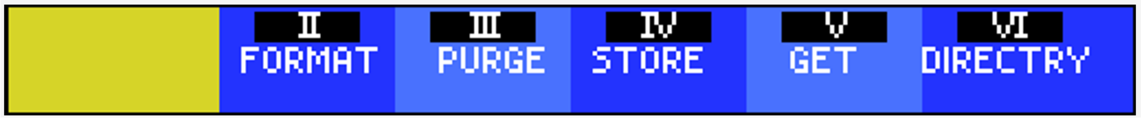
A.9 Color Codes (HEX)

Code	Color	Code	Color
0	Transparent	8	Dark Yellow (orange)
1	Black	9	Light Red
2	Dark Green	A	Yellow
3	Medium Green	B	Light Yellow
4	Dark Blue	C	Green
5	Medium Blue	D	Violet
6	Red	E	Grey
7	Cyan	F	White

These are the TMS9918A VDP color palette values used by ColecoVision and ADAM.

A.10 File Operations (STORE/GET Key)



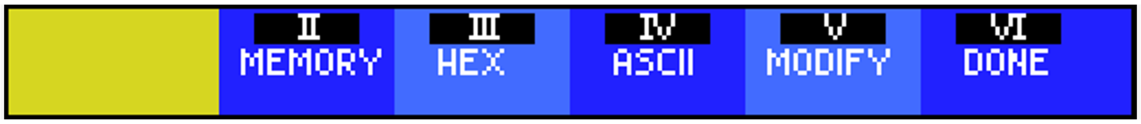


- FORMAT — Formats directory (not disk). Put freshly formatted disk in first.
- PURGE — Delete files (choose DISK FILES or RAM FILES)
- STORE — Save graphics to disk
- GET — Load graphics from disk
- DIRECTORY — List files (DISK FILES or RAM FILES)

Important: The graphics disk directory is stored in block 0, not the normal EOS location. This is why SmartWriter or FileManager won't see the directory.

A.11 The Edit Option (Memory Editor)

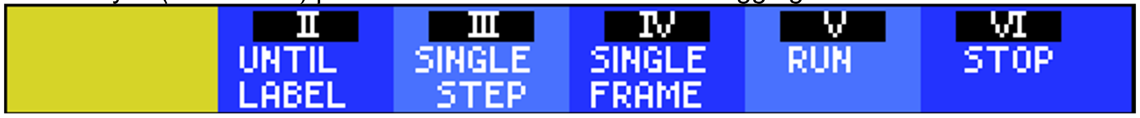
SmartKey IV (EDIT) provides direct memory access:



- MEMORY — Prompts "Where?" for address in Intel Hex format
- HEX — Display memory in hexadecimal
- ASCII — Display text representation of values
- MODIFY — Direct memory editing with cursor

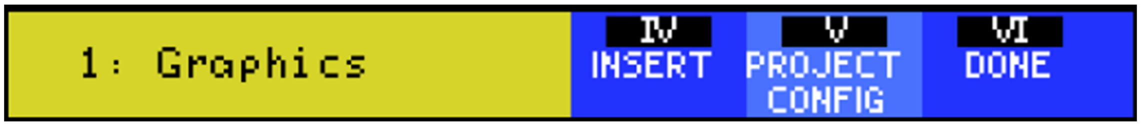
A.12 The Execute Option (Debugging)

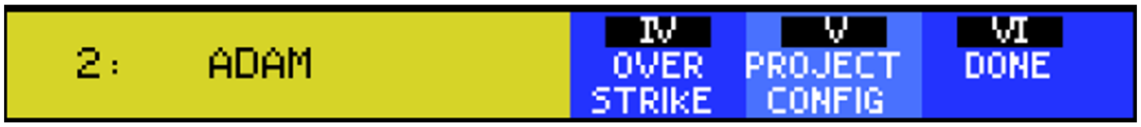
SmartKey V (EXECUTE) provides code execution and debugging:

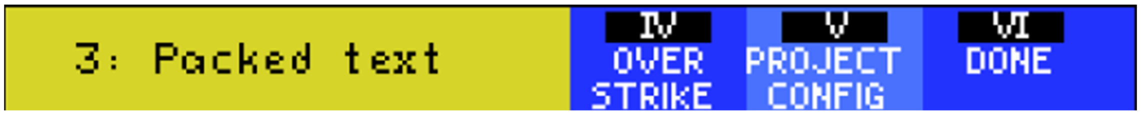


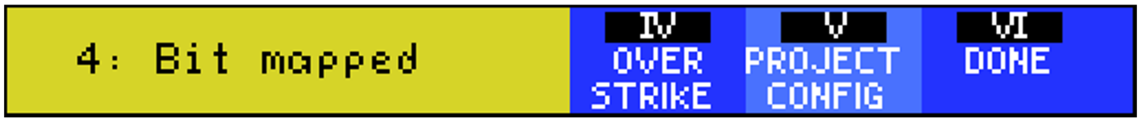
A.13 Project Configuration

SmartKey V (PROJECT CONFIG) offers four modes:









1. Graphics — Standard graphics mode
2. ADAM — ADAM-specific mode
3. Packed Text — Compressed text graphics
4. Bit Mapped — Bitmap graphics mode

— End of Original Documentation —

Documentation by Ron Collins, Akron AUG BBS
Extracted from ExpRAM distribution disk