

## EOS Routines

### Get Vram

A = table code (see above)  
DE = starting index into table  
HL = address of user buffer  
IY = block size (or byte count)

### Calc. Offset

D = Y-Pat-Pos  
E = X-Pat-Pos  
Exit = offset in register DE

### Px-To-Ptrn-Pos

DE = 16 bit signed number  
Exit = E = number

### Load Ascii

None

### Put Ascii

HL = character to load  
BC = number of characters to load  
DE = address in Vram

### Switch Memory

A = number of memory configuration

### Write Sprite Attribute

DE = address of sprite table  
HL = address of priority table  
A = number of sprites (divided by 4)

### Set Read / Set Write

HL = Vram address  
Exit = register C contains the Vdp Data Port Value

## Text Entry Standard (49944) - (24-195) Setup From (50022)

Upon exit register BC = start of buffer  
Use this routine after (50022)  
Example at (34166) in demo program

---

## Text Entry Routine (49978) - (58-195) A = Key Value

ASCII values returned with carry set  
Used by both Standard and Small Text entry

---

## Device Selector (49675) - (11-194) No Setup Required Routine

Upon exit Device Number stored at (52223) - (255-203)  
Upon exit a (NZ) condition = Previous Menu  
Example at (34297) in demo program

---

## Error Routine (49635) - (227-193) A = Error Number

---

**Text to Message Line (49654) - (246-193) Multiple SmartKey Keyboard Routine**  
DE = Message Line  
HL = SmartKeys  
B = Amount of SmartKeys  
C = Key Value

---

**Keyboard Routine (49657) - (249-193)**  
B = Amount of Smartkey  
C = Key Value

---

**Smartkey Final Test (49660) - (252-193)**  
B = Amount of SmartKeys  
C = Key Value

---

**Catalog Block Loader** (49613) - (205-193) No Setup Required

Loads Catalog Block at (54272)  
Block Number in Register A

---

**Folder Erase Routine** (49504) - (96-193) No Setup Required

Catalog Block Number at (64985)

---

**High Light Routine** (49491) - (83-193) HL = Location

---

**Erase Routine** (49487) - (79-193)

---

**Folder Routine** (48723) - (83-190) No Setup Required

Device Number at (52223) Selected by Device Selector Routine

JR C = Previous Menu  
OR A and Error  
JR NZ

---

Current Device Location (52223) - (255-203)  
File Number Location (52222) - (254-203)  
Print Buffer (52214) - (246-203)  
Line Buffer (52206) - (238-203)  
Line Store Location (52204) - (236-203)

---

For more information on the extended routines, disassemble the routines from (48723) to (51199).

## EOS Routines

### Write Vram

BC = number of bytes to be written  
DE = starting Vram address to be written to  
HL = address of buffer containing the data

### Read Vram

BC = number of bytes to be read  
DE = starting Vram address to be read from  
HL = address of buffer to receive the data

### Write Register

B = register number to write to  
C = data byte value to be written

### Read Register

Exit = data byte value in register A

### Fill Vram

A = constant byte value to be written  
DE = number of bytes  
HL = starting Vram address

### Init Table

A = table code  
HL = table address  
Table Code  
0 = sprite attribute table  
1 = sprite generator table  
2 = pattern name table  
3 = pattern generator table  
4 = pattern color table

### Put Vram

A = table code (see above)  
DE = starting index into table  
HL = address of user buffer  
IY = block size (or byte count)

**ADAM**

**EOS Programing Kit**

**Copyright (C) 1991**

**Walters Software Co.**

**User Manual**

**For the Coleco Adam Computer**

## Helpful Suggestions for Beginners

I would suggest you obtain a copy of an ADAM technical manual, preferably an EOS 6 version. This version in my opinion is the best one. The ColecoVision sound, and graphics manuals would also be helpful if you are going to work in these areas. In order to program in EOS you must have a understanding in Z80 machine programing. You must understand what each command does, the book programing the Z80 by Rodnay Zaks is a good place to start. Next study the demo program until you understand how each routine works. When you are ready to start, I would suggest you work out a simple program on paper. This program could do some basic tasks, like key sounds, drawing lines, ect. The program should also loop back to the beginning of the program. Type in your program and save it to the correct blocks on the EOS Programing Kit media. Test the program by pushing keys ect., if your program continues to perform without errors, you are ready to go on to a more complex program. If your program crashes, go back and search for the problem. Don't give up to easy, most of the time it is something simple. A computer crash can't hurt your computer, but can destroy the data on your disk or data pack. Always backup your work, sometimes I keep two backups plus my work disk. For more EOS examples you might want to disassemble some of Coleco's P.D. software. EOS programing is not hard and can be a lot of fun, all you need is a little basic knowledge and I'm sure you'll come up with a great little program. Have fun.

## EOS Programing Kit

This programing kit will open up a completely new outlook on the EOS operating system for the Adam computer. The EOS consists of routines to perform different functions on the Adam computer. Most of these routines require some additional information in order to perform it's job. Additional routines added to this system by Walters Software Co. are SmartKey, Device, Error, Line and box plotter, text entry, key sounds, screen fill and clear, file card, and more. These additional routines help make program development much easier on the Adam computer.

This programing kit does not intend to teach Z80 programing, but how to use the EOS. For more information about Z80 programing I would recommend the book Programing the Z80 by Rodnay Zaks.

## Contents

Eos Operating System, Eos Routines Examples -----	1
Writing Routines	
Eos Programing Media Layout Chart -----	3
Block Contents	
Writing a Program, Booting the Demo -----	4
Eos Extended Routines -----	6
Eos Routines -----	13
Eos Jump Table -----	21
ASCII Chart -----	24
Color Codes, Keyboard Codes -----	25
Helpful Suggestions for Beginners -----	26

## COLOR CODES

0 = Transparent  
1 = Black  
2 = Medium Green  
3 = Light Green  
4 = Dark Blue  
5 = Light Blue  
6 = Dark Red  
7 = Cyan  
8 = Medium Red  
9 = Light Red  
10 = Dark Yellow  
11 = Light Yellow  
12 = Dark Green  
13 = Magenta  
14 = Gray  
15 = White

(0 to 15) \* 16 = Text Color

(0 to 15) = Background Color

Use (0 to 15) for box and line colors

### Some of the more common Keyboard Codes

128 = Home  
129 = I  
130 = II  
131 = III  
132 = IV  
133 = V  
134 = VI  
137 = Shift + I  
138 = Shift + II

139 = Shift + III  
140 = Shift + IV  
141 = Shift + V  
142 = Shift + VI  
144 = Wild Card  
145 = Undo  
146 = Move  
147 = Store  
148 = Insert  
149 = Print  
150 = Clear  
151 = Delete  
152 = Shift + Wild Card  
153 = Shift + Undo  
154 = Shift + Move  
155 = Shift + Store  
156 = Shift + Insert  
157 = Shift + Print  
158 = Shift + Clear  
159 = Shift + Delete  
160 = Up Arrow  
161 = Right Arrow  
162 = Down Arrow  
163 = Left Arrow  
164 = Control Up  
165 = Control Right  
166 = Control Down  
167 = Control Left  
168 = Up + Right  
169 = Right + Down  
170 = Down + Left  
171 = Left + Up  
172 = Home + Up  
173 = Home = Right  
174 = Home + Down  
175 = Home + Left  
184 = Shift + Backspace  
185 = Shift + Tab

## ASCII CHART

32 -	64 - @	96 - `
33 - !	65 - A	97 - a
34 - "	66 - B	98 - b
35 - #	67 - C	99 - c
36 - \$	68 - D	100 - d
37 - %	69 - E	101 - e
38 - &	70 - F	102 - f
39 - '	71 - G	103 - g
40 - (	72 - H	104 - h
41 - )	73 - I	105 - i
42 - *	74 - J	106 - j
43 - +	75 - K	107 - k
44 - ,	76 - L	108 - l
45 - -	77 - M	109 - m
46 - .	78 - N	110 - n
47 - /	79 - O	111 - o
48 - 0	80 - P	112 - p
49 - 1	81 - Q	113 - q
50 - 2	82 - R	114 - r
51 - 3	83 - S	115 - s
52 - 4	84 - T	116 - t
53 - 5	85 - U	117 - u
54 - 6	86 - V	118 - v
55 - 7	87 - W	119 - w
56 - 8	88 - X	120 - x
57 - 9	89 - Y	121 - y
58 - :	90 - Z	122 - z
59 - ;	91 - [	123 - {
60 - <	92 - \	124 -
61 - =	93 - ]	125 - }
62 - >	94 - ^	126 - ~
63 - ?	95 - _	127 -

## The EOS Operating System

EOS is made up of routines that perform different tasks on the Adam. A routine is a group of Z80 machine code instructions. These instructions when sent to the Z80 micro processor chip in the Adam, tell it what job to perform.

### Example EOS Z80 Routine

Call 64620

This routine, (Read Keyboard) gets one key push from the keyboard. The ASCII code is placed into register (A).

### A Simple keyboard routine using an EOS routine

```

256 Call 64620
259 CP 134
261 JP Z, 32768
264 CP 133
266 JP Z, 32778
269 CP 132
271 JP Z, 32788
274 CP 131
276 JP Z, 32798
279 CP 130
281 JP Z, 32808
284 CP 129
286 JP Z, 32818
289 JR 256
    
```

This keyboard routine gets a key push from the keyboard and checks for smartkeys 1 - 6, if the key push was not a smartkey then the routine will jump back to the beginning to get another key push. If the key push was a smartkey, the routine will jump to a location corresponding to the key push. Example: if the key push was smartkey I, (129), then the routine would jump to location (32818).

## An EOS Block Load Routine

1. LD	A, 4	Device number
2. LD	BC, 0	Block number
3. LD	DE, 1	Block number
4. LD	HL, 54272	Buffer
5. CALL	64755	EOS routine load block
6. JP	NZ, 32768	Jump if error (register (A))

In line 1 we load register (A) with the device number, in this case (4) is disk drive number one. This means we will be getting our block load from disk drive number one. Lines 2 and 3 contains our block number that we want to load into memory. Start with register (DE), when (DE) is full, (255/255), use register (BC). Line 4 contain the memory location were the block load will start, this is called the buffer. Line 5 is the EOS routine that will get the block from the device, (disk drive number 1). Lines 1 - 4 sets up the EOS routine, this is the information that the EOS routine needs inorder to perform it's job correctly. Line 6 tests the routine to see if everything went OK. If a (NZ) condition exists this means an error occurred, the error code is placed in register (A). These error tests are very important and should always be used.

### Writing routines

I think the best way of starting out, is to work your routine out on paper. When you have your routine exactly the way you want it, you will want to save it for later use. There are several ways of doing this. I started out using the SmartDSK TOOLKIT program and saved the routine to a block(s) on a disk. I could later add it to my program by saving the last block of my program to a block in the ramdisk. Next copy the routine block to the next block after the one that contains the program. Use the move option to move the routine into the program block. Save the program block back to your work disk and that's all there is to it. There are several programs for the Adam both in EOS and CP/M that will work. Choose the program that suits you best.

## Eos Jump Table

Eos 2	(64776) - ( 8-253)
Eos 3	(64779) - (11-253)
CVA	(64782) - (14-253)
Get In/out Ports	(64785) - (17-253)
Bank Switch Memory	(64788) - (20-253)
Put ASCII to VDP	(64791) - (23-253)
Write Vram	(64794) - (26-253)
Read Vram	(64797) - (29-253)
Write VDP	(64800) - (32-253)
Read VDP	(64803) - (35-253)
Fill Vram	(64806) - (38-253)
Init Vram	(64809) - (41-253)
Put Vram	(64812) - (44-253)
Get Vram	(64815) - (47-253)
Calculate offset	(64818) - (50-253)
Point to Pattern	(64821) - (53-253)
Load ASCII to VDP	(64824) - (56-253)
Write Sprite Table	(64827) - (59-253)
Read Game Controller	(64830) - (62-253)
Update Spinner	(64833) - (65-253)
Decrement Low Nibble	(64836) - (68-253)
Decrement High Nibble	(64839) - (71-253)
High to Low	(64842) - (74-253)
Add A to HL 16 bit	(64845) - (77-253)
Sound Init	(64848) - (80-253)
Sound Off	(64851) - (83-253)
Start Song	(64854) - (86-253)
Sound	(64857) - (89-253)
Effect Over	(64860) - (92-253)

## EOS Jump Table

Start Print Buffer	(64668) - (156-252)
Start Print Character	(64671) - (159-252)
Start Read One Block	(64674) - (162-252)
Start Character Device	(64677) - (165-252)
Start Read Keyboard	(64680) - (168-252)
Start Write One Block	(64683) - (171-252)
Start Character Device	(64686) - (174-252)
Synchronize Z80 and 6801	(64689) - (177-252)
Write One Block	(64692) - (180-252)
Write Character Device	(64695) - (183-252)
Init File Manager	(64698) - (186-252)
Init Tape Directory	(64701) - (189-252)
Open File	(64704) - (192-252)
Close File	(64707) - (195-252)
Reset File	(64710) - (198-252)
Make File	(64713) - (201-252)
Read Directory For File	(64716) - (204-252)
Set File	(64719) - (207-252)
Read File	(64722) - (210-252)
Write File	(64725) - (213-252)
Set Date	(64728) - (216-252)
Get Date	(64731) - (219-252)
Rename File	(64734) - (222-252)
Delete File	(64737) - (225-252)
Read Device Depend Status	(64740) - (228-252)
Goto Word Processor	(64743) - (231-252)
Read EOS	(64746) - (234-252)
Trim File	(64749) - (237-252)
Check FCB	(64752) - (240-252)
Read Block	(64755) - (243-252)
Write Block	(64758) - (246-252)
Mode Check	(64761) - (249-252)
Scan For File	(64764) - (252-252)
File Query	(64767) - (255-252)
Position File	(64770) - ( 2-253)
EOS 1	(64773) - ( 5-253)

On the EOS programming media you will find a program file containing an EOS operating system, (SmartDSK), Expanded EOS, two fonts, two main buffers, 16 K of work space expandable to 48K.

Block	Location	Contents
0	51200	Boot Block
1		Catalog
2	0	Work Space
3	1024	Program Setup
4	32768	Work Space
5	33792	" "
6	34816	" "
7	35840	" "
8	36864	" "
9	37888	" "
10	38912	" "
11	39936	" "
12	40960	" "
13	41984	" "
14	43008	" "
15	44032	" "
16	45056	" "
17	46080	" "
18	47104	" "
19	48128	" "
19	48723	Expanded EOS
20	49152	" "
21	50176	" "
22	51200	Work Space
22	51763	ASCII Data
23	52224	Fonts
24	53248	Fonts
	54272	Buffer
	55296	"
25	56320	SystemSetup
26 - 33	57344 - 65535	EOS



### Block 0

This block contains the routines to copy the program into memory. In order to add more program workspace, you must change the routines to load half of the program, move it into place, then load the rest of the program. To make the program workspace smaller, you must modify the amount of bytes copied to memory in the read file routine.

### Block 1

This block contains the file information, (catalog). If you are changing the size of the workspace don't forget to change the blocks used and blocks reserved.

### Block 3

This block contains the Startup routine, which moves the program and EOS into the right memory locations. If you are changing the size of the program you must change this routine.

### Blocks 4 - 18

These blocks are reserved for your program. There is also some space in block 19 (48128 - 48722) and block 22 (51200 - 51763).

### Blocks 19, 20, 21

These blocks contain the expanded EOS routines.

### Blocks 23 and 24

These are the expanded EOS fonts.

### Block 25

This block contains the EOS startup.

### Blocks 26 - 33

These blocks contain the SmartDSK EOS system, (9).

The **System buffers** are at 54272, 55296, and 56320.

After Bootup memory locations 0 - 32767 can be used as a **Program buffer**.

## EOS Jump Table

EOS Start	(64560) - (48-252)
Console Display	(64563) - (51-252)
Console Init	(64566) - (54-252)
Console / Display Character	(64569) - (57-252)
Delay	(64572) - (60-252)
End Print Buffer	(64575) - (63-252)
End Print Character	(64578) - (66-252)
End Read One Block	(64581) - (69-252)
End Read Character	(64584) - (72-252)
End Read Keyboard	(64587) - (75-252)
End Write One Block	(64590) - (78-252)
End Write Character Device	(64593) - (81-252)
Find DCB	(64596) - (84-252)
Get DCB Address	(64599) - (87-252)
Get PCB Address	(64602) - (90-252)
Hard Init / Cold Start	(64605) - (93-252)
Hard Reset Net	(64608) - (96-252)
Print Buffer	(64611) - (99-252)
Print Character	(64614) - (102-252)
Read One Block	(64617) - (105-252)
Read Keyboard	(64620) - (108-252)
Read Keyboard Return Code	(64623) - (111-252)
Read Printer Return Code	(64626) - (114-252)
Read Return Code	(64629) - (117-252)
Read Tape Return Code	(64632) - (120-252)
Relocate PCB	(64635) - (123-252)
Request Status	(64638) - (126-252)
Request Keyboard Status	(64641) - (129-252)
Request Printer Status	(64644) - (132-252)
Request Tape Status	(64647) - (135-252)
Scan Adam Net	(64650) - (138-252)
Soft Init	(64653) - (141-252)
Soft Reset Device	(64656) - (144-252)
Soft Reset Keyboard	(64659) - (147-252)
Soft Reset Printer	(64662) - (150-252)
Soft Reset Tape	(64665) - (153-252)

## Eos Routines

### EOS Start

None

### Read One Block

A = device number  
HL = address of buffer  
BCDE = block number

No Error = Z  
Error = NZ, A = error code

### Write One Block

A = device number  
HL = address of buffer  
BCDE = address of buffer

(same as Read One Block)

### Start Read One Block

A = device number  
HL = address of buffer  
BCDE = block number

(same as Read One Block)

### End Read One Block

A = device number

C = command complete  
NC = not complete  
Z = no errors  
NZ = error, A = error code

**Start Write One Block** (same as Start ReadOne Block)

**End Write One Block** (same as Start Read One Block)

## Writing a program.

To start writing a program you must decide what program to use. I started using TOOLKIT from SmartDSK III, this program is called a block editor. You can also use a macro assembler, editor, or any program that lets you enter code and save it to a block. Using TOOLKIT set the offset to the location you wish your program to start at. Load in the amount of blocks you want to work with. Type in your code, when finished save it back to the blocks that were loaded in.

The extended EOS is setup to start a program at 33792, if your program starts in a different location you must change the startup location. If you wish to display a title screen a location is provided. To change these locations you must modify block 25. Location 56744 - 56746 is reserved for a title screen, locations 56785 - 56787 are reserved for the program start.

## Booting the Demo Program.

Insert the EOS Programming Kit media into one of your drives, and pull the computer reset switch. The program will boot in a few moments. The first screen contains the options to choose a full ramdisk or to reserve one bank. The option to insert a title screen before the ramdisk screen is included. The next screen contains the title of the program, and four key push sounds. Push smartkey (VI) to continue. The third screen is a demo of displaying text in three sizes, small, standard, and large. The next screen shows you how to draw a line and a box. The fifth screen fills an area on the screen. The next screen is a demo of the standard text entry and the small text entry. The next screen shows the "ONE MOMENT PLEASE", error routine, device select, and file card. The last screen shows three clear routines. Pushing continue from the last screen will return you to the first demo screen. Pushing the escape key will also return you to the first demo screen. This entry can be changed to suit your program. A disassembly of the demo program is included for your convenient. The disassembly is provided by another W.S.C program, Decimal Disassembler.

Name of Routine	Call	Setup Information
Back Up Sound	(51185) - (241-199)	No Setup Required Saves all Registers
Error Sound	(51163) - (219-199)	No Setup Required Saves all Registers
Ding Sound	(51114) - (170-199)	No Setup Required Saves all Registers
Tink Sound	(51081) - (137-199)	No Setup Required Saves all Registers
Clear Smartkeys	(51052) - (108-199)	No Setup Required Uses A, DE, HL
Clear Message Line	(51042) - (98-199)	No Setup Required Uses A, DE, HL
Clear Display	(51033) - (89-199)	No Setup Required Uses A, DE, HL
Display Text Modified	(50959) - (15-199)	A = Color HL = Location DE = Buffer BC = Font Table
You must exx registers and load HL with the scale before calling this entry. This is a custom routine.		
Display Text Large X 2	(50955) - (11-199)	A = Color HL = Location DE = Buffer
Routine Setup Example		
205 - 50 - 198		
33 - ( ) - ( )		
17 - ( ) - ( )		
62 - 23		
Continued on next page		

## EOS Routines

**Read Keyboard** No Error = Z  
None Error = NZ, A = error code  
Exit = register A contains key value

**Print Character** (same as Read Keyboard)  
A = character to be printed

**Print Buffer** (same as Read Keyboard)  
HL = address of buffer (buffer must end with a (3))

**Setup Print Buffer** (same as Read Keyboard)  
A = Ascii character

**Start Print Character** (same as Read Keyboard)  
A = Ascii character

**End Print Character** C = command complete  
None NC = not complete  
No Error = Z  
Error = NZ, A = error code

**Start Print Buffer** (same as Read Keyboard)  
HL = address of buffer (must end in (3))

**End Print Buffer** (same as End Print Character)  
None

**Console Init**  
B = number of columns  
C = number of lines  
D = X coord. of upper left corner  
E = Y coord. of upper left corner  
HL = pattern name table

**Console Out/ Console Disp**  
A = character  
Optionally - E = Y location, D = X location

## EOS Routines

### Rename File

A = device number  
DE = address of old file name  
HL = address of new file name

No Error Z = 1, A = 0  
Error Z = 0, A = error code

### Delete File

A = device number  
HL = address of file name

(same as Rename file)

### Read Block

A = device number  
HL = address of buffer  
BCDE = block number

No Error = Z  
Error = NZ, A = error code

### Write Block

A = device number  
HL = address of buffer  
BCDE = block number

(same as Read Block)

### Trim File

A = device number  
DE = address of file name

(same as Read Block)

### Init Directory

A = device number  
C = number of blocks in directory  
DE = number of blocks in device  
HL = address to volume name

(same as Read Block)

### Find DCB

A = device number  
Exit = address of DCB in register IY

(same as Read Block)

### Request Status

A = device number

(same as Read Block)

229  
213  
205 - 11 - 199  
209  
225  
36  
8  
121  
198 - 8  
79  
8  
205 - 11 - 199

---

### Display Text Standard X 1

(50949) - ( 5 - 199)

A = Color  
HL = Location  
DE = Buffer  
BC = Font Table

This is a Custom Routine

---

### Display Text Standard X1

(50946) - ( 2 - 199)

A = Color  
HL = Location  
DE = Buffer

The routine loads font table at (52224)

---

### Display Text Standard X1

(50944) - ( 0 - 199)

HL = Location  
DE = Buffer

This routine displays text black on red

---

### Display Text Message line Standard X1

(50941) - (253-198)

DE = Buffer

This routine displays text on the message line

---

**Clear Message Line** (50933) - (245-198) DE = Buffer  
**Clear SmartKeys**  
**Display Text on Message Line**

---

**ASCII to Decimal Converter** (50868) - (180-198) HL = Number  
BC = Buffer

---

**File Loader Custom** (50842) - (154-198) A = Device Number  
HL = File Name  
DE = Buffer  
BC = Amount

Amount = number of Bytes

---

**1 K File Loader** (50830) - (142-198) HL = File Name

Device must be at (52223) (255-203)  
File is loaded at Buffer (56320) ( 0-220)

---

**4 K File Loader** (50816) - (128-198) HL = File Name

Device must be at (64879) - (111-253)  
File is loaded at (40960) - ( 0-160)

---

**One Moment Please** (50807) - (119-198) No Setup Required

Displays Text on Message Line

---

**Test for SmartKeys** (50781) - ( 93-198) No Setup Required

Calls (50764) - ( 76-198) keyboard routine  
Returns with key number in register C and register A, (1-6)

---

**Read Keyboard** (50764) - ( 76-198) No Setup Required

Test For Escape Key (27) and jump to (33792) - ( 0-132) This jump can be changed to suit your program by changing location (50779 - 50780) Returns with untested key press in register A

---

## EOS Routines

**Write File** No Error Z = 1, A = 0  
A = file number Error Z = 0, A = error code  
HL = address of buffer  
BC = number of bytes

**Set Date**  
B = day  
C = month  
D = year

**Get Date** Error Z = 0, A = error code  
B = day B = 0  
C = month C = 0  
D = year D = 0

**File Manager Init**  
DE = address to place FCB buffer  
HL = address to place FCB header

**Scan For File** No Error Z = 1, A = 0  
A = device number Error Z = 0, A = error code  
HL = address of file name  
Exit = file start block in BCDE

**Check if Directory** Z = directory exists  
IX = points to volume entry NZ = no directory  
A = error code

**Strcmp** No Error Z = 1, A = 0  
HL = address of name Error Z = 0, A = error code  
DE = address of file name in directory

**Mode Check**  
IX = FCB address  
HL = directory entry (file name)  
Exit = Z=1 OK Z=0 NO

## EOS Routines

### Update Spinner

None

### Query File

A = device number  
DE = address of file name  
HL = address of buffer  
Exit = directory entry in buffer (if no error)  
BCDE = file start block

No Error Z = 1, A = 0  
Error Z = 0, A = error code

### Set File

A = device number  
DE = address of file name  
HL = address of buffer

No Error Z = 1, A = 0  
Error Z = 0, A = error code

### Make File

A = device number  
HL = address of file name  
BCDE = file size (in bytes)

Z = no errors  
NZ = error, A = error code

### Open File

A = device number  
HL = address of file name  
B = mode  
Exit = file number in register A

No Error Z = 1, A = file number  
Error Z = 0, A = error code

### Close File

A = file number

No Error Z = 1, A = 0  
Error Z = 0, A = error code

### Reset File

A = file number

(Error same as Close File)

### Read File

A = file number  
HL = address of buffer  
BC = number of bytes

(Error same as Close File)  
BC = number of bytes transferred

**Reset Keyboard** (50761) - (73-198) No Setup Required

This routine uses the Read keyboard at (50764)

**Font Enlarger** (50738) - (50-198) No Setup Required

Returns with Font Table location in register BC

**Display Text Small** (50643) - (211-197) DE = Screen Location  
HL = ASCII Buffer

**SmartKey Maker** (50545) - (113-197) B = SmartKey  
HL = ASCII Buffer

**Text to Message Line and Multiple SmartKey** (50526) - (94-197) DE = Message Line  
HL = SmartKeys  
B = Amount of SmartKeys

6 = 1 SmartKey displayed (VI)  
5 = 2 SmartKeys displayed (VI & V)  
4 = 3 SmartKeys displayed (VI, V & IV)  
3 = 4 SmartKeys displayed (VI, V, IV & III)  
2 = 5 SmartKeys displayed (VI, V, IV, III & II)  
1 = 6 SmartKeys displayed (VI, V, IV, III, II & I)

**Plotter** (50185) - (9-196) A = Color

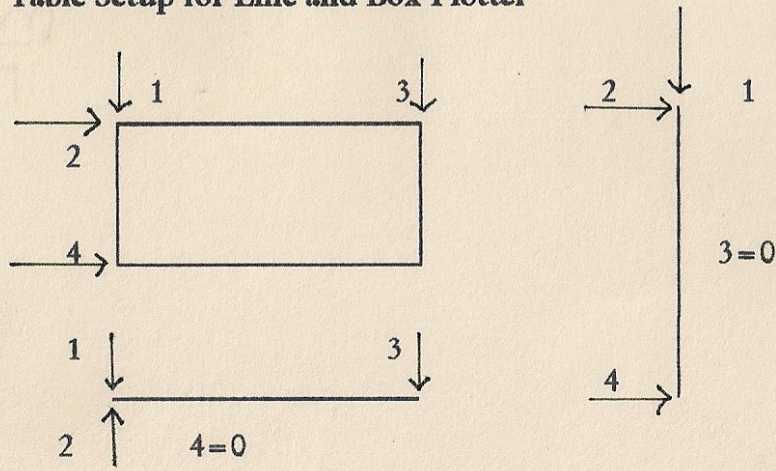
BC = Coordinates  
DE = Coordinates

**Line or Box Plotter from Table** (50149) - (229-195) A = Color  
HL = Table

**Multiple Box or Line Plotter** (50133) - (213-195) A = Color  
HL = Table

Use zero to terminate table

## Table Setup for Line and Box Plotter



Line Example at (33086) Box example at (33091)

---

**Fill Area on Display** (50115) - (195-195) B = Number of lines  
 DE = Length  
 HL = Location  
 A = Color

---

**Setup for Text Entry Small** (50053) - (133-195) A = Color  
 HL = Buffer  
 DE = Location  
 BC = Length

---

**Setup for Text Entry Standard** (50022) - (102-195) A = Color  
 HL = Buffer  
 DE = Location  
 BC = length

---

**Text Entry Small** (49961) - (41-195) Setup from (50053)

Upon exit register BC = start of buffer  
 Use this routine after (50053) Example at (34222) in Demo program

## EOS Routines

Decoder (game controller)  
 A = controller number (0-1)  
 Exit  
 H = fire button data  
 L = joystick data  
 B = arm button data  
 C = raw data for segment 0  
 D = keyboard data  
 E = spinner count data  
 A = raw data for segment 1

## Poller

IX = pointer to first byte of users controller map  
 A = controller enable bit  
 Exit  
 IX+0 = joystick 0  
 IX+1 = fire 0  
 IX+2 = arm 0  
 IX+3 = keyboard 0  
 IX+4 = spinner count 0  
 IX+5 = joystick 1  
 IX+6 = fire 1  
 IX+7 = arm 1  
 IX+8 = keyboard 1  
 IX+9 = spinner count 1

## Debounce

HL = pointer to debounce buffer  
 IX = pointer to user controller map  
 A = joystick data  
 B = arm data  
 C = fire data  
 D = keyboard data  
 Exit  
 HL = pointer to next buffer  
 IX = pointer to next data area