Rob Funk's Reference Cards

This is the first time Rob Funk's Reference Cards are made publically available. Although he hasn't finished them all yet, these can already be quite useful! Enjoy them! For a straight ASCII file, see the link on the main page. To get a nice Postscript file, use the print option of Mosaic and the print command 'cat > filename'.

Rob Funk (Funk.17@osu.edu) ***or*** *Ivo van Poorten (ipoorten=www@cs.vu.nl)*

BASIC Commands

Commands

Cmd	Abbr/Ex.	Comment
BYE	B.	Goes to memo pad or self-test mode
CLR	CLR	Clears all variables
CONT	CON.	Continues execution after or STOP
DIM	DI. A$(30)	Reserves 30 bytes for A$
	DI. B(100)	Defines an array with 101 positions
	DI. C(17,3)	Defines an 18 x 4 array
END	END	Closes files, turns off sound, halts program
LET	LE. A=B	Assigns variable A to the value of variable B
	C$=D$	(the word LET can be omitted)
LIST	L.	Lists a program
	L. 400,500	Lists program lines 400 through 500
NEW	NEW	Erases program and variables from memory
POKE	POKE Y,X	Writes value X to memory address Y
REM	. program comment	Comment
RUN	RU.	Begins execution of program in memory
	RU. "D1:MYPROG	Loads MYPROG from disk and runs it
STOP	STO.	Halts program without closing files

Program Statements

FOR, TO, STEP...	F. X=3 TO 9 STEP 2	STEP may be omitted for step value of 1 3, 9, and 2 may be arithmetic expressions
NEXT	N. X	
GOSUB... RETURN	100 GOS. 300:___ 300 ? 400 RET.	RETURN goes to the statement following the colon.
GOTO	G. X	X may be a variable or line number
IF/THEN	IF Y=5 THEN 500	Conditional branch
	IF X THEN Y=6	X=0 is false, X>0 is true.
	IF A$="Y" THEN A=1:?	False goes to next line number; true executes the rest of the line
ON/GOSUB	ON X GOS. 20,30,40	If X<1 or X>3, it goes to the next line
ON/GOTO	ON X G. 20,30,40	

```
POP       POP                 Use when RETURN or NEXT is bypassed
TRAP      T. 1000             Identifies the line to GOTO if error occurs
          T. 32768             If TRAP >32767, previous TRAP is cleared
```

I/O Commands

```
CLOAD     CLOA.               Loads a program from cassette
CLOSE     C. #2               Closes a file (no error if file is not open)
CSAVE     CS.                 Saves a program to cassette
DOS       DO.                 Goes to DOS (if DOS not loaded, same as BYE)
ENTER     E. "D1:MYPROG       Loads a program saved with LIST
GET       GE. #5,Y            Input a single byte
INPUT     I. Y$               Prints "?", then receives data from keyboard
          I. #1,A              Receives data from file #1
          I. #16,Y$            Recieves data from keyboard with no "?"
LIST      L. "D1:MYPROG       Lists a program to a file
LOAD      LO. "D1:MYPROG      Loads a program save with SAVE
LPRINT    LP. A$              Prints a line to printer
NOTE      NO. #2,A,B          Detects sector and byte within file
OPEN      O. #2,N,0,"D1:FILE  Open FILE; N=4 - input   N=6 - directory
                                         N=7 - dir w/ <> (DOS 2.5)
                                         N=8 - output  N=9 - append
                                         N=12 - input and output
POINT     P. #3,A,B           Position on sector A, byte B within a file
PRINT     ? A,B;"HERE"        Comma tabs, semicolon appends
PUT       PU. #5,Y            Output a single byte
READ      REA. A,B            Assigns data values from DATA statements
DATA      D. 5,10,16,3        Hold data values for READ
RESTORE   RES. 350            Data for next READ is on line 350
SAVE      S. "D1:MYPROG       Saves a program to be loaded with LOAD or RUN
STATUS    ST. #3,A            Sets A to the device status value
XIO       XIO cmd,#5,aux1,aux2,"S:"  (See XIO Command Codes)
```

BASIC Functions

Cmd	Abbr/Ex.	Comment

Arithmetic Functions

ABS	Y=ABS(X)	Absolute value - |X|
CLOG	Y=CLOG(X)	Base 10 logarithm
EXP	Y=EXP(X)	Inverse of LOG - e^X
INT	Y=INT(X)	Integer, rounds down: INT(-4.5) = -5
LOG	Y=LOG(X)	Natural logarithm, e=2.71...
RND	Y=RND(X)	Random number between 0 and 1 (value of X is irrelevent)
SGN	Y=SGN(X)	Evaluates sign of X; Y=-1,0,1
SQR	Y=SQR(X)	Square root

Trig Functions (others are derived)

ATN	Y=ATN(X)	Inverse tangent (Arctan)
COS	Y=COS(X)	Cosine
SIN	Y=SIN(X)	Sine
DEG	DEG	Use degrees for angle measurements
RAD	RAD	Use radians for angle measurements (default)

Special Functions

ADR	Y=ADR(X$)	Memory address of string
FRE	? FRE(0)	Remaining free space in RAM (in pages)
PEEK	Y=PEEK(X)	Contents of memory at address X
USR	Y=USR(X)	Result of machine language program at memory address X

String Functions

ASC	Y=ASC(X$)	ATASCII code of first byte of X$
CHR$	Y$=CHR$(X)	Character with ATASCII value X
LEN	Y=LEN(X$)	Length of string
STR$	Y$=STR$(X)	Convert number to string - STR$(12) = "12"

VAL	Y=VAL(X$)	Convert string to number - VAL("12") = 12
substring	Y$=X$(5,8)	Y$ contains the 5th through 8th character of X$

Graphics/Sound (X increases right, Y increases down)

GET #6	GE. #6,A	Input data from screen
GRAPHICS	GR. M	Reset graphics mode to mode M
COLOR	C. 3	Set color number for PLOT or DRAWTO
DRAWTO	DR. X,Y	Draw line to screen coordinate
LOCATE	LOC. X,Y,A	Set A to the COLOR number of coordinate X,Y
PLOT	PL. X,Y	Plot a graphics point
POSITION	POS. X,Y	Set cursor position
PUT #6	PU. #6,A	Output data to screen
SETCOLOR	SE. R,H,L	Change Hue and Luminance of color Register
SOUND	SO. V,P,D,L	Turn on sound: Voice (0-3), Pitch (0-255), Distortion (0-14 even), Loudness (0-15)
XIO 18	X. 18,#6,0,0,"S:"	Fill bounded area of screen with color in memory location 765

Controller Functions

PADDLE	Y=PADDLE(X)	Paddle value, Y=0 to 228 400/800: X=0 to 7 XL/XE: X=0 to 3
PTRIG	Y=PTRIG(X)	Paddle trigger, Y=0 if pressed, 1 if not 400/800: X=0 to 7 XL/XE: X=0 to 3
STICK	Y=STICK(X)	Joystick position 400/800: X=0 to 3 XL/XE: X=0 or 1
STRIG	Y=STRIG(X)	Joystick trigger, Y=0 if pressed, 1 if not 400/800: X=0 to 3 XL/XE: X=0 or 1

Color Register Values

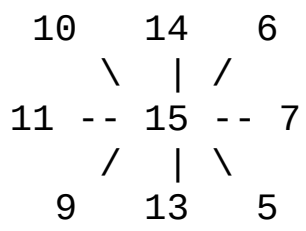
Color	Setcolor Hue	ADD-VALUE
gray	0	0
light orange	1	16
orange	2	32
red-orange	3	48
pink	4	64
purple	5	80
purple-blue	6	96
blue	7	112
blue	8	128
light blue	9	144
turquoise	10	160
green-blue	11	176
green	12	192
yellow-green	13	208
orange-green	14	224
light orange	15	240

```
for SETCOLOR A,B,C the contents of
color reg. A  =  (ADD-VALUE of B) + C
              =  (B * 16) + C
```

Symbolic Device Names

Symbol	Device	IOCB
C:	Cassette tape unit	#7
D1:-D9:	Disk units 1-9 (number available depends on DOS)	
E:	Screen editor	#0
K:	Keyboard	
P: (P1:)	Printer	
P2:-P8:	Alternate printer; e.g. most parallel are P2:	
R1:-R4:	RS-232 interfaces; requires handler to be loaded	
S:	Screen	#6
Z:	R-Time-8 Clock; requires handler to be loaded	

Joystick Movement (STICK values)



bits:

3	2	1	0
R	L	D	U
i	e	o	p
g	f	w	
h	t	n	
t			

6502 Assembly Language Mnemonics

Code	Operation
ADC	ADd memory to accumulator with Carry
AND	AND memory with accumulator
ASL	(Arithmetic) Shift Left one bit
BCC	Branch on Carry Clear
BCS	Branch on Carry Set
BEQ	Branch on result EQual to zero
BIT	test BITs in accumulator with memory
BMI	Branch on result MInus (negative)
BNE	Branch on result Not Equal to zero
BPL	Branch on result PLus (positive)
BRK	force Break
BVC	Branch on oVerflow flag Clear
BVS	Branch on oVerflow flag Set
CLC	CLear Carry flag
CLD	CLear (binary-coded) Decimal mode
CLI	CLear Interrupt disable flag
CLV	CLear oVerflow flag
CMP	CoMPare memory and accumulator
CPX	ComPare memory and index X
CPY	ComPare memory and index Y
DEC	DECrement memory by one
DEX	DEcrement index X by one
DEY	DEcrement index Y by one
EOR	Exclusive OR memory with accumulator
INC	INCrement memory by one
INX	INcrement index X by one
INY	INcrement index Y by one
JMP	JuMP to new location
JSR	Jump to new location, Save Return address (Jump to SubRoutine)
LDA	LoaD Accumulator from memory
LDX	LoaD index X from memory
LDY	LoaD index Y from memory
LSR	(Logical) Shift Right one bit
NOP	No OPeration
ORA	OR memory with Accumulator
PHA	PusH Accumulator on stack
PHP	PusH Processor status on stack

```
PLA       PulL Accumulator from stack
PLP       PulL Processor status from stack
ROL       ROtate Left one bit
ROR       ROtate Right one bit
RTI       ReTurn from Interrupt
RTS       ReTurn from Subroutine
SBC       SuBtract memory and borrow from accumulator (SuBtract with Carry)
SEC       SEt Carry flag
SED       SEt (binary-coded) Decimal mode
SEI       SEt Interrupt disable flag
STA       STore Accumulator in memory
STX       STore index X in memory
STY       STore index Y in memory
TAX       Transfer Accumulator to index X
TAY       Transfer Accumulator to index Y
TSX       Transfer Stack pointer to index X
TXA       Transfer index X to Accumulator
TXS       Transfer index X to Stack pointer
TYA       Transfer index Y to Accumulator
```

Peek/Poke Adresses Frequently Used

(multi-byte values are LSB-MSB)

Label	Decimal	Hex	Description
DOSVEC	10-11	A-B	Start vector for disk software, normally to start of DUP.SYS routine
POKMSK	16	10	POKEY interrupt shadow. To disable BREAK key, poke this and 53774 ($D20E) w/ 112.
			BIT DECIMAL ENABLED INTERRUPT
			7 128 break key
			6 64 "other key"
			5 32 serial input data ready
			4 16 serial output data required
			3 8 serial out xmit finished
			2 4 POKEY timer 4
			1 2 POKEY timer 2
			0 1 POKEY timer 1
RTCLOCK	18-20	12-14	24-bit TV frame counter/real-time clock
SOUNDR	65	41	Noisy I/O flag: 0=quiet, default=3
ATTRACT	77	4D	0 to suppress attract mode, 128 to start; incremented every time loc. 19 increments
LMARGIN	82	52	Left screen margin, default 2
RMARGIN	83	53	Right screen margin, default 39
GRMODE	87	57	(BASIC) Graphics mode number
RAMTOP	106	6A	Top of RAM in 256-byte pages
STOPLIN	186-7	BA-B	Line number of STOP or TRAP (BASIC)
ERRSAV	195	C3	(BASIC) Error number
FR0	212-3	D4-5	Value returned by USR (BASIC)
VBREAK	518-9	0206-7	Software break vector for 6502 BRK instruction
SDMCTL	559	022F	Shadow for DMA control register
			Playfield size: 1=narrow, 2=standard,
			3=wide. Missile DMA=4, Player DMA=8.
			Player resolution: 0=double line,
			16=single line.
			DMA enable=32
SDLSTL	560-1	0230-1	Points to display list
COLDST	580	0244	Nonzero = reboot when RESET pressed

KEYDIS	621	026D	0=keyboard enabled, 255=disabled (XL/XE)
FINE	622	026E	0=course scroll, 255=fine scroll (XL/XE)
GPRIOR	623	026F	Player/Missile/PlayField priority: 1 = P0-3, PF0-3, BAK 2 = P0-1, PF0-2, P2-3, BAK 4 = PF0-3, P0-3, BAK 8 = PF0-1, P0-3, PF2-3, BAK 16 = allow 5th player (from missiles) 32 = allow 3rd color
PADDL0-7	624-631	0270-7	values of paddles 0-7, 0-228
STICK0-3	632-5	0278-B	values of joysticks 0-3
PTRIG0-7	636-643	027C-0283	values of paddle buttons 0-7
STRIG0-3	644-7	0284-7	values of joystick triggers 0-3
TXTROW	656	0290	Text cursor row
TXTCOL	657-8	0291-2	Text cursor column
INVFLG	694	02B6	Inverse character flag, XOR'ed w/ATASCII code 0=normal, 128=inverse; other values result in weirder stuff
SHFLOK	702	02BE	0=lowercase (no shift), 64=uppercase (shift) 128=control key
BOTSCR	703	02BF	Number of text rows available for printing
PCOLR0	704	02C0	Color of player/missile 0
PCOLR1	705	02C1	Color of player/missile 1
PCOLR2	706	02C2	Color of player/missile 2
PCOLR3	707	02C3	Color of player/missile 3
PF0	708	02C4	Color register 0
PF1	709	02C5	Color register 1
PF2	710	02C6	Color register 2
PF3	711	02C7	Color register 3
BAK	712	02C8	Color register 4
KRPDEL	729	02D9	Jiffies before key begins repeating (XL/XE) 0=no repeat
KEYREP	730	02DA	Jiffies between key repeats (XL/XE) 0=repeat only once
NOCLIK	731	02DB	Non-zero=no key click (XL/XE)
HELPFG	732	02DC	HELP key status: 17=HELP pressed alone, 81=SHIFT-HELP, 145=CTRL-HELP; must be manually cleared. (XL/XE)
RUNAD	736-7	02E0-1	Run address of binary file
INITAD	738-9	02E2-3	Initialization address of binary file
MEMTOP	741-2	02E5-6	Top of free memory, below screen memory

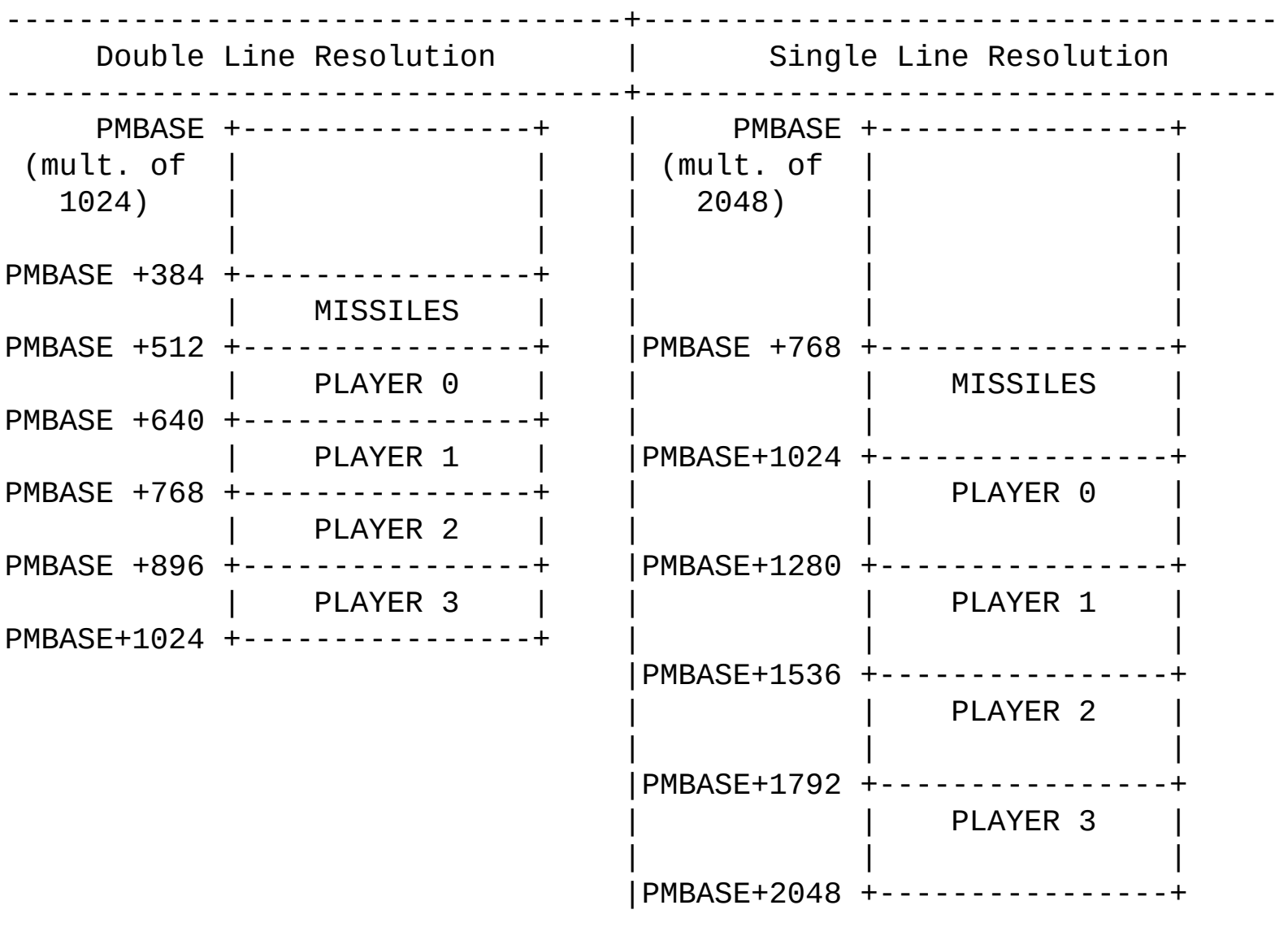
MEMLO	743-4	02E7-8	Bottom of free memory
CRSINH	752	02F0	Cursor inhibit: 0=on, otherwise off
CHACT	755	02F3	Character mode register, controls how characters are displayed:

VALUE	INVERSE CHARS	ORIENTATION	CURSOR
0	inverse	upright	absent
1	blank	upright	absent
2	normal	upright	transparent
3	solid	upright	opaque
4	inverse	inverted	absent
5	blank	inverted	absent
6	normal	inverted	transparent
7	solid	inverted	opaque

CHBAS	756	02F4	Pointer to page of character set (even): default 224; 226=lowercase/graphics; 204=international (XL/XE)
CH	764	02FC	Keyboard code of last key pressed (255 to clear)
FILLDAT	765	02FD	Color data for XIO FILL command
DSPFLG	766	02FE	0=control characters do their function, otherwise control characters display as characters (as if ESC pressed first)
SSFLAG	767	02FF	Start/stop display flag: 255=stop, 0=go; normally manipulated with CTRL-1
HPOSP0	53248	D000	(W) Horizontal position of player 0
M0PF	"	"	(R) Missile 0/playfield collision
HPOSP1	53249	D001	(W) Horizontal position of player 1
M1PF	"	"	(R) Missile 1/playfield collision
HPOSP2	53250	D002	(W) Horizontal position of player 2
M2PF	"	"	(R) Missile 2/playfield collision
HPOSP3	53251	D003	(W) Horizontal position of player 3
M3PF	"	"	(R) Missile 3/playfield collision
HPOSM0	53252	D004	(W) Horizontal position of missile 0
P0PF	"	"	(R) Player 0/playfield collision
HPOSM1	53253	D005	(W) Horizontal position of missile 1
P1PF	"	"	(R) Player 1/playfield collision
HPOSM2	53254	D006	(W) Horizontal position of missile 2
P2PF	"	"	(R) Player 2/playfield collision
HPOSM3	53255	D007	(W) Horizontal position of missile 3
P3PF	"	"	(R) Player 3/playfield collision
SIZEP0	53256	D008	(W) Size of player 0, 1=2X, 3=4X
M0PL	"	"	(R) Missile 0 to player collision
SIZEP1	53257	D009	(W) Size of player 1, 1=2X, 3=4X
M1PL	"	"	(R) Missile 1 to player collision

SIZEP2	53258	D00A	(W) Size of player 2, 1=2X, 3=4X
M2PL	"	"	(R) Missile 2 to player collision
SIZEP3	53259	D00B	(W) Size of player 3, 1=2X, 3=4X
M3PL	"	"	(R) Missile 3 to player collision
SIZEM	53260	D00C	(W) Missile size, 1=2X, 3=4X
P0PL	"	"	(R) Player 0 to player collision
P1PL	53261	D00D	(R) Player 1 to player collision
P2PL	53262	D00E	(R) Player 2 to player collision
P3PL	53262	D00F	(R) Player 3 to player collision
GRACTL	53277	D01D	(W) 1=Missile DMA, 2=Player DMA
HITCLR	53278	D01E	(W) clear collision registers
PMBASE	54279	D407	(W) Player/Missile base address
WSYNC	54282	D40A	(W) Wait for horizontal sync
VCOUNT	54283	D40B	(R) Vertical TV scan line counter
NMIEN	54286	D40E	(W) Non-maskable interrupt enable (192 for DLI)

Player/Missile Area Layout



Sound Command Pitch Values

Note	Octave: / Distort:	-3 / 12	-2 / 12	-1 / 10	0 / 10	+1 / 10	+2 / 10	+3 / 10
B		67	33	128	64	31		
A#		72	36	136	68	33	16	
A		75	37	144	72	35		
G#		82	40	153	76	37	18	
G		85	42	162	81	40		
F#		90	45	173	85	42		
F		98	48	182	91	45		
E		102	51	193	96	47	23	
D#			55	204	102	50		
D			57	217	108	53	26	
C#			60	230	114	57		
C			63	243	121*	60	29	14

* Middle C

XIO Command Codes

code	operation
3	OPEN
5	GET RECORD
7	GET CHARACTERS
8	PUT RECORD
11	PUT CHARACTERS
12	CLOSE
13	STATUS REQUEST
17	DRAW LINE
18	FILL
32	RENAME
33	DELETE
34	CREATE DIRECTORY (MyDOS)
34	LOCK DISK (SpartaDOS 2.3/3.2)
35	LOCK FILE
36	UNLOCK FILE
37	POINT
38	NOTE
39	GET FILE LENGTH (SpartaDOS)
40	LOAD BINARY FILE
41	SAVE BINARY FILE (Atari DOS, SpartaDOS 2.3/3.2)
41	CHANGE DIRECTORY (MyDOS)
42	CREATE DIRECTORY (SpartaDOS)
43	DELETE DIRECTORY (SpartaDOS)
44	CHANGE DIRECTORY (SpartaDOS)
45	SET BOOT FILE (SpartaDOS)
46	UNLOCK DISK (SpartaDOS 2.3/3.2)
49	SET FILE ATTRIBUTES (SpartaDOS X)
253	FORMAT SINGLE DENSITY (DOS 2.5)
254	FORMAT DISK (default format)