

From: HAND-HELD-COMPUTER
EPSON HX-20 Software
ASSEMBLER
DISASSEMBLER
page '70'

HC-20 PROTOCOL

CREATED BY KENJI AKAHANE
REVISION 8-2
DATE 03.04.1982
UPDATE 04.20.1982
UPDATE 11.19.1982
FILE PROTOCOL HX1D

NOTE. UPDATE MARK IS 72 COLUMN '%'. REVISION B
NOTE. UPDATE MARK IS 72 COLUMN '@'. REVISION C

CP/NET LOGICAL MESSAGE SPECIFICATION

NOTES: MM = MASTER ID
SS = SLAVE ID
XX = DON'T CARE BYTE
NN = VALUE SPECIFIED

ALL NUMERIC VALUES ARE IN HEXADECIMAL

FMT	DID	SID	FNC	SIZ	MSG / FUNCTION NAME

SYSTEM RESET:					
00	MM	SS	00	00	00-00 = XX
01	SS	MM	00	00	00-00 = 00
CONSOLE INPUT:					
00	MM	SS	01	00	00-00 = XX
01	SS	MM	01	00	00-00 = 00
CONSOLE OUTPUT:					
00	MM	SS	02	00	00-00 = XX
01	SS	MM	02	00	00-00 = CHARACTER INPUT
RAW CONSOLE INPUT:					
00	MM	SS	03	00	00-00 = MASTER CONSOLE #
01	SS	MM	03	00	00-00 = CHARACTER INPUT
RAW CONSOLE OUTPUT:					
00	MM	SS	04	00	00-00 = MASTER CONSOLE #
01	SS	MM	04	00	00-00 = CHARACTER TO OUTPUT
LIST PUTPUT:					
00	MM	SS	05	00	00-00 = MASTER LIST #
					01-NN = CHARACTER TO LIST DEVICE
					(NN = 01 TO 80)
01	SS	MM	05	00	00-00 = 00
DIRECT CONSOLE I/O:					
00	MM	SS	06	00	00-00 = XX
01	SS	MM	06	00	00-00 = 00
GET I/O BYTE:					
00	MM	SS	07	00	00-00 = XX
01	SS	MM	08	00	00-00 = 00

00	MM	SS	08	00	SET I/O BYTE:
01	SS	MM	08	00	00-00 = XX
					00-00 = 00
00	MM	SS	09	00	PRINT STRING:
01	SS	MM	09	00	00-00 = XX
					00-00 = 00
00	MM	SS	0A	00	READ CONSOLE BUFFER:
01	SS	MM	0A	00	00-00 = XX
					00-00 = 00
00	MM	SS	0B	00	GET CONSOLE STATUS:
01	SS	MM	0B	00	00-00 = MASTER CONSOLE #
					00-00 = CONSOLE STATUS BYTE
00	MM	SS	0C	00	RETURN VERSION NUMBER:
01	SS	MM	0C	00	00-00 = XX
					00-00 = 00
00	MM	SS	0D	00	RESET DISK SYSTEM:
01	SS	MM	0D	00	00-00 = XX
					00-00 = 00
00	MM	SS	0E	00	SELECT DISK:
01	SS	MM	0E	00	00-00 = SELECTED DISK
					00-00 = RETURN CODE
00	MM	SS	0F	0E	OPEN FILE:
					00-01 = FCB ADDRESS IN SLAVE
					02-02 = DRIVE CODE
					03-0A = FILE NAME
					0B-0D = FILE TYPE
					0E-0E = EXTEND NUMBER
01	SS	MM	0F	00	00-00 = DIRECTORY CODE
00	MM	SS	10	01	CLOSE FILE:
01	SS	MM	10	00	00-00 = FCB ADDRESS IN SLAVE
					00-00 = DIRECTORY CODE
00	MM	SS	11	0C	SEARCH FOR FIRST:
					00-00 = DRIVE CODE
					01-08 = FILE NAME
					09-0B = FILE TYPE
					0C-0C = EXTEND NUMBER
01	SS	MM	11	00	00-00 = DIRECTORY CODE
					01-20 = DIRECTORY FCB ENTRY
00	MM	SS	12	01	SEARCH FOR NEXT:
01	SS	MM	12	00	00-00 = XX
					00-00 = DIRECTORY CODE
					01-20 = DIRECTORY FCB ENTRY

00	MM	SS	13	0C	DELETE FILE:
					00-00 = DRIVE CODE
					01-08 = FILE NAME
					09-0B = FILE TYPE
					0C-0C = EXTEND NUMBER
01	SS	MM	13	00	00-00 = DIRECTORY CODE

READ SEQUENTIAL:

00	MM	SS	14	03	00-01 = FCB ADDRESS IN SLAVE
					02-02 = EXTEND NUMBER
					03-03 = CURRENT RECORD
01	SS	MM	14	82	00-00 = EXTEND NUMBER
					01-01 = CURRENT RECORD
					02-81 = SECTOR OF DATA READ
					82-82 = RETURN CODE
00	MM	SS	15	83	WRITE SEQUENTIAL:
					00-01 = FCB ADDRESS IN SLAVE
					02-02 = EXTEND NUMBER
					03-03 = CURRENT RECORD
					04-83 = SECTOR OF DATA TO WRITE
01	SS	MM	15	02	00-00 = EXTEND NUMBER
					01-01 = CURRENT RECORD
					02-02 = RETURN CODE
00	MM	SS	16	0E	MAKE FILE:
					00-01 = FCB ADDRESS IN SLAVE
					02-02 = DRIVE CODE
					03-0A = FILE NAME
					0B-0D = FILE TYPE
					0E-0E = EXTEND NUMBER
01	SS	MM	16	00	00-00 = DIRECTORY CODE
00	MM	SS	17	1F	RENAME FILE:
					00-00 = DRIVE CODE
					01-08 = FILE NAME
					09-0B = FILE TYPE
					0C-0C = EXTEND NUMBER
					0D-0D = S1 (NOT USED)
					0E-0E = S2 (NOT USED)
					0F-0F = RECORD COUNT (NOT USED)
					10-10 = DRIVE CODE
					11-18 = FILE NAME
					19-1B = FILE TYPE
					1C-1C = EXTEND NUMBER
					1D-1D = S1 (NOT USED)
					1E-1E = S2 (NOT USED)
					1F-1F = RECORD COUNT (NOT USED)
01	SS	MM	17	00	00-00 = DIRECTORY CODE

00	MM	SS	18	00	RETURN LOGIN VECTOR:
					00-00 = XX
01	SS	MM	18	00	00-00 = 00
00	MM	SS	19	0C	RETRUN CURRENT DISK:
					00-00 = XX
01	SS	MM	19	00	00-00 = 00
00	MM	SS	1A	00	SET DMA ADDRESS:
					00-00 = XX
01	SS	MM	1A	00	00-00 = 00
00	MM	SS	1B	00	GET ALLOCATION VECTOR ADDRESS:
					00-00 = XX
01	SS	MM	1B	00	00-00 = 00
00	MM	SS	1C	00	WRITE PROTECT DISK:
					00-00 = XX
01	SS	MM	1C	00	00-00 = 00

00	MM	SS	1D	00	GET R/O VECTOR:
01	SS	MM	1D	00	00-00 = XX
					00-00 = 00
00	MM	SS	1E	0C	SET FILE ATTRIBUTES:
					00-00 = DRIVE CODES
					01-08 = FILE NAME
					09-0B = FILE TYPE
					0C-0C = EXTEND NUMBER
01	SS	MM	1E	00	00-00 = 00
00	MM	SS	1F	00	GET DISK PARAMETER ADDRESS
01	SS	MM	1F	00	00-00 = XX
					00-00 = 00
00	MM	SS	20	00	SET/GET USER CODE:
01	SS	MM	20	00	00-00 = SET/GET CODE
					00-00 = CURRENT CODE (IF GET)
00	MM	SS	21	04	READ RANDOM
					00-01 = FCB ADDRESS IN SLAVE
					02-04 = R0,R1,R2 RANDOM RECORD #
01	SS	MM	21	82	00-00 = EXTEND NUMBER
					01-01 = CURRENT RECORD
					02-81 = SECTOR OF DATA READ
					82-84 = RETURN CODE
00	MM	SS	22	84	WRITE RANDOM:
					00-01 = FCB ADDRESS IN SLAVE
					02-81 = SECTOR OF DATA TO WRITE
					82-84 = R0,R1,R2 RANDOM RECORD #
01	SS	MM	22	02	00-00 = EXTEND NUMBER
					01-01 = CURRENT RECORD
					02-02 = RETURN CODE
00	MM	SS	23	01	COMPUTE FILE SIZE:
01	SS	MM	23	05	00-01 = FCB ADDRESS IN SLAVE
					00-00 = EXTEND NUMBER
					01-01 = CURRENT RECORD
					02-04 = R0,R1,R2 RANDOM RECORD #
					05-05 = RETURN CODE
00	MM	SS	24	03	SET RANDOM RECORD
					00-01 = FCB ADDRESS IN SLAVE
					02-02 = EXTEND NUMBER
					03-03 = CURRENT NUMBER
01	SS	MM	24	03	00-02 = R0,R1,R2 RANDOM RECORD #
					03-03 = 00

00	MM	SS	25	01	RESET DRIVE:
01	SS	MM	25	00	00-01 = DRIVE VECTOR
					00-00 = RETURN CODE
00	MM	SS	26	01	ACCESS DRIVE:
01	SS	MM	26	00	00-01 = DRIVE VECTOR
					00-00 = 0
00	MM	SS	27	01	FREE DRIVE:
01	SS	MM	27	00	00-01 = DRIVE VECTOR
					00-00 = 0
					WRITE RANDOM WITH ZERO FILL:

00	MM	SS	28	84	00-01 = FCB ADDRESS IN SLAVE
					02-81 = SECTOR OF DATA TO WRITE
					82-84 = R0,R1,R2 RANDOM RECORD #
01	SS	MM	28	02	00-00 = EXTEND NUMBER
					01-01 = CURRENT RECORD
					02-02 = RETURN CODE

HC-20 SERIAL NETWORK PROTOCOL

1. MASTER - SLAVE HANDSHAKE

SOURCE	DESTINATION	COMMENT
		ENQUIRE TO DESTINATION DEVICE
(EOT)	---	
P1	---	
DID	---	
SID	---	
ENQ	---	
	<---	ACK
SOH	---	SEND HEADER (FUNCTION)
FMT	---	
DID	---	
SID	---	
FNC	---	
SIZ	---	
HCS	---	
	<---	ACK (NAK), (WAK)
STX	---	
DB0	---	
DB1	---	
.	---	
.	---	
DBN	---	
ETX	---	
CKS	---	
	<---	ACK, (NAK)
(EOT)	---	

2. NET WORK RS232C 8-BIT STANDARD PROTOCOL

FMT	DID	SID	FNC	SIZ	MSG
-----	-----	-----	-----	-----	-----

FMT = MESSAGE FORMAT CODE
 DID = MESSAGE DESTINATION PROCESSOR ID
 SID = MESSAGE SOURCE PROCESSOR ID
 FNC = FUNCTION CODE
 SIZ = DATA FIELD LENGTH - 1
 MSG = ACTUAL MESSAGE, SIZ + 1 BYTES LONG

MESSAGE FIELD LENGTH TABLE							
FMT CODE	FMT	DID	SID	FNC	SIZ	MSG	COMMENT
00	1	1	1	1	1	1-256	PREFERRED FORMAT
01	1	1	1	1	1	1-256	RETURNED RESULT
02	1	1	1	1	2	1-65536	(NOT SUPPORTED)

(EOT)

(B): FROM SLAVE TO MASTER

1 ACK = IF RECIEVED CORRECTLY, AND IS ABLE TO RECIVE TEXT THEN 'ACK'
IF RECIEVED CORRECTLY, BUT ISN'T ABLE TO RECIVE TEXT THEN 'WAK'
IF RECIEVED NOT CORRECTLY, THEN 'NAK'.

(C): FROM MASTER TO SLAVE

IF RECIEVED CODE FROM SLAVE IS 'ACK' GOTO 'SEND TEXT'
IF RECIEVED CODE IS 'WAK', WAIT 100 MILI SEC, THEN GOTO (A),
IF RECIEVED CODE IS 'NAK', GOTO (A)
IF RECIEVED CODE IS NOT 'ACK' EITHER 'NAK' OR NOT RECIEVED ANSWER,
SEND SLAVE 'ENQ', THEN SLAVE SEND (ACK: 'NAK', 'ACK') TO MASTER
AGAIN.

NOTE.

TIME OUT = 100 MILI SEC.

ONCE MASTER CPU SELECT SLAVE CPU, UNTIL 'EOT' CODE IS SEND TO SLAVE
CPU, MASTER-SLAVE CONNECTION IS NOT CUT.

IF CURRENR HEADER IS SAME AS LAST HEADER, CURRENT HEADER IS ABLE TO
BE OMIT.

HX-20 ORIGINAL COMMANDS

NOTE.

HX-20 ORIGINAL FNC COMMANDS ARE FROM 80H TO FFH.

IF THE BIT6 OF FNC IS 0, THE COMMAND HAS TO RECIEVE THE ANSWER
FROM DEVICE. IF BIT6 IS 0, DESTINATION DEVICE MAY OMIT THE ANSWER.

(1) DISK

SEE TF FUNCTION

(TF function not in booklet)

4 CRT COMMAND

SEE SCREEN DOCUMENT.

DOCUMENT OF SCREEN COMMAND

FILE NAME	SCREEN HX1D
CREATED	06.10.1982
AUTHOR	K. AKAHANE, M. HANAOKA
UPDATE	06.23.1982 K. A

SCREEN COMMAND

ENTRY POINT : DISCON (#FF5E)

PRAMETER : 1 ACCX : PACKET TOP ADDRESS.

2 PACKET

0 : FUNCTION

1 : DATA 1

.

.

N : DATA N

3 NOTE

IF YOU USE CRT THEN PACKET NEED MORE 4 BYTES AREA
FROM FUNCTION-4 TO FUNCTION-1. BUT THEN ACCX POINTS
FUNCTION PACKET ADDRESS.

RETURN : 1 PACKET
DATA IS SET FROM NEXT ADDRESS OF FUNCTION.

REGISTER : PRESERVD ACCX AND ACCB

FMT	DID	SID	FNC	SIZ	MSG / FUNCTION NAME (L:LCD C:CRT)

					SCREEN DEVICE SELECT(L.C)
00	MM	SS	84	00	00-00 = DEVICE NO. (CRT:30 LCD:22)
01	SS	MM	84	00	00-00 = ERROR CODE. 00 : NON ERROR. FE : DEVICE NOT READY. FF : DEVICE NAME NOT CORRECT.
					INITIALIZE SCREEN DEVICE.(C)
00	MM	SS	85	00	00-00 = XX
01	SS	MM	85	00	00-00 = ERROR CODE. 00 : NON ERROR. FF : I/O ERROR.
					CHECK SCREEN DEVICE AND GET SOME PARAMETERS. (L.C)
00	MM	SS	86	00	00-00: XX
01	SS	MM	86	00	00-00: SCREEN DEVICE NO. 01-02: SCREEN TOP ADDRESS.(CRT) 03-04: MAX SCREEN SIZE.(CRT)
					SET SCREEN SIZE (L.C)
00	MM	SS	87	03	00-00 = SCREEN WIDTH OF VIRTUAL SCREEN 01-01 = SCREEN DEPTH OF VIRTUAL SCREEN 02-03 = TOP ADDRESS OF VIRTUAL SCREEN (NOT USED)
01	SS	MM	87	00	00-00 = ERROR CODE. 00 : NON ERROR. FF : SCREEN SIZE IS NOT CORRECT. FE : ADDRESS OF TOP OF SCREEN IS NOT CORRECT.
					READ SCREEN SIZE (L.C)
00	MM	SS	88	00	00-00 = XX
01	SS	MM	88	01	00-00 = SCREEN WIDTH 01-01 = SCREEN DEPTH
					GET PHYSICAL SCREEN SIZE (L.C)
00	MM	SS	89	00	00-00 = XX
01	SS	MM	89	01	00-00 = SCREEN WIDTH 01-01 = SCREEN DEPTH
					SET THE PYSICAL SCREEN POINTER ON THE VIRTUAL SCREEN (L.C)
00	MM	SS	C0	01	00-00 = COORDINATE (X) 01-01 = COORDINATE (Y)
01	SS	MM	C0	00	00-00 = XX
					GET THE PYSICAL SCREEN POINTER ON THE VIRTUAL SCREEN (L.C)
00	MM	SS	8A	00	00-00 = XX

01	SS	MM	8A	01	00-00 = COORDINATE (X) 01-01 = COORDINATE (Y)
00	MM	SS	C1	01	?SET READ POINTER (L.C) 00-00 = COORDINATE OF X IN THE VIRTUAL SCREEN
01	SS	MM	C1	00	01-01 = COORDINATE OF Y 00-00 = XX
00	MM	SS	8B	00	?GET READ POINTER (L.C) 00-00 = XX
01	SS	MM	8B	01	00-00 = COORDINATE OF X IN THE VIRTUAL SCREEN
					01-01 = COORDINATE OF Y
00	MM	SS	C2	01	SET CURSOR POSITION ON THE VIRTUAL SCREEN (L.C) 00-00 = COORDINATE (X)
01	SS	MM	C2	00	01-01 = COORDINATE (Y) 00-00 = XX
00	MM	SS	8C	00	GET CURSOR POSITION ON THE VIRTUAL SCREEN (L.C) 00-00 = XX
01	SS	MM	8C	01	00-00 = COORDINATE (X) 01-01 = COORDINATE (Y)
00	MM	SS	C3	00	SET CURSOR MARGIN (L.C) 00-00 = MARGIN
01	SS	MM	C3	00	00-00 = XX
00	MM	SS	8D	00	GET CURSOR MARGIN (L.C) 00-00 = XX
01	SS	MM	8D	00	00-00 = MARGIN
00	MM	SS	C4	01	SET SCROLL STEP (L.C) 00-00 = HORIZONTAL SCROLL STEP
01	SS	MM	C4	00	01-01 = VERTICAL SCROLL STEP 00-00 = XX
00	MM	SS	8E	00	GET SCROLL STEP (L.C) 00-00 = XX
01	SS	MM	8E	01	00-00 = HORIZONTAL SCROLL STEP 01-01 = VERTICAL SCROLL STEP
00	MM	SS	C5	00	SET LIST FLAG (L.C) 00-00 = XX
01	SS	MM	C5	00	00-00 = XX
00	MM	SS	C6	00	RESET LIST FLAG (L.C) 00-00 = XX
01	SS	MM	C6	00	00-00 = XX
00	MM	SS	C7	04	SET POINT TO DISPLAY (L.C) 00-01 = COORDINATE (X) 02-03 = COORDINATE (Y)
01	SS	MM	C7	00	04-04 = COLOR CODE 00-00 = XX
					GET POINT ON THE DISPLAY (L.C)

00	MM	SS	8F	03	00-01 = COORDINATE (X)
					02-03 = COORDINATE (Y)
01	SS	MM	8F	00	00-00 = COLOR CODE
DRAW LINE TO DISPLAY (L.C)					
00	MM	SS	C8	08	00-01 = COORDINATE (X) OF START POINT
					02-03 = COORDINATE (Y) OF START POINT
					04-05 = COORDINATE (X) OF END POINT
					06-07 = COORDINATE (Y) OF END POINT
					08-08 = COLOR CODE (C)
01	SS	MM	C8	00	00-00 = XX
?READ CHARACTER ON THE READ POINTER (L.C)					
(ERASED 1982/05/08)					
00	MM	SS	90	00	00-00 = XX
01	SS	MM	90	00	00-00 = READ CHARACTER
READ THE EXTEND OF CURRENT LINE (L.C)					
00	MM	SS	91	00	00-00 = XX
01	SS	MM	91	03	00-00 = FIRST LINE NUMBER WHICH HAS CURRENT LINE.
					01-01 = LAST LINE NUMBER WHICH HAS CURRENT LINE.
					02-02 = LAST COORDINATE (X)
					03-03 = LAST COORDINATE (Y)
SET LINE TERMINATE POSITION (L.C)					
00	MM	SS	C9	00	00-00 = LINE NUMBER
01	SS	MM	C9	00	00-00 = XX
WRITE ONE CHARACTER TO VIRTUAL SCREEN (L.C)					
00	MM	SS	92	00	00-00 = CHARACTER CODE
01	SS	MM	92	01	00-00 = CURSOR POSITION X
					01-01 = CURSOR POSITION Y
CLEAR THE GRAPHICS SCREEN. (L.C)					
00	MM	SS	CA	00	00-00 = BACK GROUND COLOR (C)
01	SS	MM	CA	00	00-00 = XX
SET SCROLL SPEED (C)					
00	MM	SS	CB	00	00-00 = SCROLL SPEED (0-9)
01	SS	MM	CB	00	00-00 = XX

00	MM	SS	93	02	**SET DISPLAY MODE (C)	%
					00-00 = CHARACTER MODE.	%
					(00:NOT USE 01:ACTIVE)	%
					01-01 = GRAPHIC MODE.	%
					(00:NOT USE 01:GRP0 01:GRP1)	%
					02-02 = BACKGROUND COLOR	%
					(0:GREEN	%
					1:YELLOW	%
					2:BLUE	%
					3:RED	%
					4:WHITE	%
					5:SYIAN	%
					6:MAGENTA	%
					7:ORANGE)	%
01	SS	MM	93	00	00-00 = ERROR CODE	%
					(00:NON ERROR FF:ERROR)	%

?SET CURSOR POSITION

00	MM	SS	CC	01	ON PHISICAL SCREEN (C)	
					00-00 = COORDINATE (X)	
					01-01 = COORDINATE (Y)	
01	SS	MM	CC	00	00-00 = XX	
						?GET CURSOR POSITION
					ON PHYSICAL SCREEN (C)	
00	MM	SS	94	00	00-00 = XX	
01	SS	MM	94	01	00-00 = COORDINATE (X)	
					01-01 = COORDINATE (Y)	
						WRITE ONE CHARACTER ON ACC-POSITION
					ON PHYSICAL SCREEN (C)	
00	MM	SS	CD	01	00-00 = CHARACTER CODE	
					01-01 = COLOR CODE.	
01	SS	MM	CD	00	00-00 = XX	
						SET ACC-POSITION ON PHYSICAL SCREEN(C)
					00-00 = COORDINATE (X)	
					01-01 = COORDINATE (Y)	
00	MM	SS	CE	01	00-00 = XX	
01	SS	MM	CE	00		
						READ ONE CHARACTER ON ACC-POSITION
					ON PHYSICAL SCREEN (C)	
00	MM	SS	95	00	00-00 = XX	
01	SS	MM	95	01	00-00 = CHARACTER CODE	
					01-01 = COLOR CODE.	
						COLOR SET SELECT (C)
00	MM	SS	CF	00	00-00 = COLOR SET CODE.	%
					(0:COLOR SET 0 1:COLOR SET 1)	%
01	SS	MM	CF	00	00-00 = XX	%
						SET CURSOR MODE (C)
					00-00 = CURSUR MODE	
					(0:ON 1:OFF)	
00	MM	SS	D0	00	00-00 = XX	
01	SS	MM	D0	00		
						?READ ONE LINE'S (32 BYTES) CHARACTERS ON
					READ POINTER (C)	
00	MM	SS	96	00	00-00 = XX	
01	SS	MM	96	20	CHARACTER CODE	
						?SET GRAPHIC CHARACTER FONT (C)
00	MM	SS	D1	06	00-00 = CHARACTER CODE	
					01-06 = CHARACTER FONT PATTERN	
01	SS	MM	D1	01	00-00 = XX	
						?DRAW CIRCLE (C)
00	MM	SS	D2	0?	00-0? =	
01	SS	MM	D2	00	00-00 = XX	
						?PAINT (C)
00	MM	SS	D3	0?	00-0? =	
01	SS	MM	D3	00	00-00 = XX	
*** SCREEN NEW COMMAND (1982/05/08)						%
						%
						%
						%
00	MM	SS	97	03	READ CHARACTERS FROM VS. (L,C)	%
					00-00 = START X-COORDINATE.	%
					01-01 = START Y-COORDINATE.	%
					02-03 = READ CHARACTERS NO.	%

01	SS	MM	97	NN	00-NN = CHARACTERS WHICH ARE RED.	%
						%
					WRITE ONE CHARACTER TO VS AND	%
					GET EXTEND OF NEW CURSOR POSITION.(L,C)%	%
00	MM	SS	98	00	00-00 = CHARACTER CODE.	%
01	SS	MM	98	03	00-00 = CURSOR POSITION X.	%
					01-01 = CURSOR POSITION Y.	%
					02-02 = FIRST LINE NO.	%
					03-03 = LAST LINE NO.	%
						%
						@

From HX-20 technical doc:

FMT	DID	SID	FNC	SIZ	Text data No.	Description of function and text
						Terminal floppy reset
00	SS	MM	0E	00	00	XX
01	MM	SS	0E	00	00	Return code 00
						File open
00	SS	MM	0F	0E	00	High-order byte of FCB address in HX-20
					01	Low-order byte of FCB address in HX-20
					02	Drive code (1: Drive A or 2: Drive B)
					03 ~ 0A	Filename
					0B ~ 0D	File type
					0E	Extend number (Normally 0)
01	MM	SS	0F	00	00	Return code
						BDOS error (See Note at the end of this table.)
						FF: File not found.
						Codes other than the above: Normal
						File close
00	SS	MM	10	01	00	High-order byte of FCB address in HX-20
					01	Low-order byte of FCB address in HX-20
01	MM	SS	10	00	00	Return code
						(The same return code as that at file open.)
						First data search
00	SS	MM	11	0C	00	Drive code (1 or 2)
					01 ~ 08	Filename
					09 ~ 0B	File type
					0C	Extend number (Normally 0)
01	MM	SS	11	20	00	Return code
						(The same return code as that at file open.)
					01 ~ 20	Directory FCB entry
						(The FCB of the found directory is entered.)
						Next data search
00	SS	MM	12	00	00	XX
01	MM	SS	12	20	00	Return code
						(The same return code as that at file open.)
					01 ~ 20	Directory FCB entry
						(The FCB of the found directory is entered.)
						File creation
00	SS	MM	16	0E	00	High-order byte of FCB address in HX-20
					01	Low-order byte of FCB address in HX-20
					02	Drive code (1 or 2)
					03 ~ 0A	Filename
					0B ~ 0D	File type
					0E	Extend number (Normally 0)
01	MM	SS	16	00	00	Return code
						(The same return code as that at file open.)

						File Rename
00	SS	MM	17	1F	00	Drive code (1 or 2)
					01 ~ 08	Filename before change (8 characters)
					09 ~ 0B	File type before change (8 characters)
					0C	Extend number
					0D ~ 0F	Unused
					10	Drive code (1 or 2)
					11 ~ 18	Filename after change (8 characters)
					19 ~ 1B	File type after change (8 characters)
					1C	Extend number
					1D ~ 1F	Unused
01	MM	SS	17	00	00	Return code
						(The same return code as that at file open.)
						Random data read
00	SS	MM	21	04	00	High-order byte of FCB address in HX-20
					01	Low-order byte of FCB address in HX-20
					02	R0 \
					03	R1 Random record numbers
					04	R2 /
01	MM	SS	21	82	00	Extend number
					01	Current record number
					02 ~ 81	Read data (128 bytes)
					82	Return code
						BDOS error (See Note at the end of this table.)
						Codes other than the above: Normal
						Random data write
00	SS	MM	22	84	00	High-order byte of FCB address in HX-20
					01	Low-order byte of FCB address in HX-20
					02 ~ 81	Write data (128 bytes)
					82	R0 \
					83	R1 Random record numbers
					84	R2 /
01	MM	SS	22	02	00	Extend number
					01	Current record number
					02	Return code
						BDOS error (See Note at the end of this table.)
						Codes other than the above: Normal
						File size calculation
00	SS	MM	23	01	00	High-order byte of FCB address in HX-20
					01	Low-order byte of FCB address in HX-20
01	MM	SS	23	05	00	Extend number
					01	Current record number
					02	R0 \
					03	R1 Random record numbers
					04	R2 /
					05	Return code (Always 0)
						Disk all copy
01	SS	MM	7A	00	00	Drive code (1 or 2)
01	MM	SS	7A	02	00	High-order byte of currently copied track number
					01	Low-order byte of currently copied track number
						0 to 39
						FFFF : End
					02	Return code (BDOS error or 0)
						Direct write into disk
00	SS	MM	7B	82	00	Drive code (1 or 2)
					01	Track number (0 to 39)

					02	Sector number (1 to 64)
					03 ~ 82	Write data (128 bytes).
00	MM	SS	7B	00	00	Return code (BDOS error or 0)
						Disk formatting (FRMAT)
00	SS	MM	7C	00	00	Drive code (1 or 2)
01	MM	SS	7C	02	00	High-order byte of currently formatted track number
					01	Low-order byte of currently formatted track number
						0 to 39
						FFFF : End
					02	Return code (BDOS error or 0)
						New system disk generation (SYSGEN)
00	SS	MM	7D	00	00	XX
01	MM	SS	7D	02	00	\ 0000 : Not end
					01	/ FFFF : End
					02	Return code (BDOS error or 0)
						Disk free area calculation (DSKF)
00	SS	MM	7E	00	00	Drive code (1 or 2)
01	MM	SS	7E	01	00	Free area size (in 2K-byte units)
					01	Return code (BDOS error or 0)
						Direct read from disk (DSKI\$)
00	SS	MM	7F	02	00	Drive code (1 or 2)
					01	Track number (0 to 39)
					02	Sector number (1 to 64)
01	MM	SS	7F	80	00 ~ 7F	Read data (128 bytes)
					80	Return code (BDOS error or 0)
						Disk boot
00	SS	MM	80	00	00	Application ID
						(in BASIC 80 ₁₆ ... BOOT80.SYS)
01	MM	SS	80	FF	00	Return code
						00 : Normal
						FF : File not found.
					01 ~ FF	Read data
						Load open
00	SS	MM	81	0D	00 ~ 07	Filename (the filename of DISK BASIC is "DBASIC".)
					08 ~ 0A	File type (the file type of DISK BASIC is "SYS".)
					0B	Relocate flag
						00: Do not relocate.
						01: Relocate from the starting address.
						02: Relocate from the ending address.
					0C ~ 0D	Ending or starting address
01	MM	SS	81	02	00	Return code
						00: Normal
						FF: File not found.
					01	High-order byte of file size
					02	Low-order byte of file size
						Load close
00	SS	MM	82	00	00	XX
01	MM	SS	82	00	00	Return code (Always 0)
						Read one block
00	SS	MM	83	01	00	High-order byte of current record

					01	number
						Low-order byte of current record number
01	MM	SS	83	82	00	High-order byte of current record number
					01	Low-order byte of current record number
					02 ~ 81	Read data
					82	Return code (00: Normal; FF: End)

Reverse engineered from epspd:

						Sector read
00	SS	MM	77	01	00	Track number
					01	Sector number
01	MM	SS	77	82	00	Track number
					01	Sector number
					02 ~ 81	Read data
					82	Return code (00: Normal; FF: End)
						Sector write
00	SS	MM	78	82	00	Track number
					01	Sector number
					02 ~ 81	Write data
01	MM	SS	78	00	00	Return code (00: Normal; FF: End)

NOTE: The term "BDOS error" used in the above table refers to one of the following errors: a read error (error code:FA), a write error (error code: FB), a drive select error (error code: FC), and a write protect error (error code: FD or FE).

Some useful IDs (SS, MM):

- HX-20: 20h
- PX-8: 22h
- PX-4: 23h
- First TF-20 (first & second disk): 31h
- Second TF-20 (third & fourth disk): 32h

Commands used by CP/M:

- FDC_RESET_M (0E)
- FDC_READ (77)
- FDC_WRITE (78)
- FDC_WRITEHST (79)

Commands used by user programs:

- COPYDISK: FDC_COPY (7A) & FDC_FORMAT (7C)