CHAPTER 10 FLOPPY DISK UNIT

CHAPTER 10 FLOPPY DISK UNIT

10.1 General

The TF-20 Terminal Floppy is an intelligent floppy disk unit which is connected to the HX-20 through a serial communication interface and transfers the data stored in a floppy disk to the HX-20 according to the commands received from the HX-20.
When the TF-20 is connected to the HX-20, the DBASIC.SYS (Disk BASIC System, which is an extended portion of BASIC) is loaded from the floppy disk into the RAM of the HX-20 upon start of BASIC. The DBASIC.SYS loaded into the RAM operates together with the interpreter on the ROM until control is returned to the MENU again. It processes the data input/output to and from the floppy disk and newly added commands, statements and functions. The interpreter on the ROM handles the conventional functions of the HX-20.
In DISK BASIC, a maximum of two TF-20 units can be connected to the HX-20. The first TF-20 unit is used as disk drives "A:" and "B:" and the second unit as disk drives "C:" and "D:". To distinguish between the first and second units, the DIP switch located in the TF-20 must be used. The 4-pin DIP switch (bits 1 to 4) of the TF-20 is factory-set to all "ON" for drives "A:" and "B:". When connecting a second TF-20 unit to the HX-20, the DIP switch setting of the second unit must be changed to "bits 1, 2, 3, 4 = ON, ON, ON, OFF" to indicate that the unit is used as drives "C:" and "D:".
Daisy-chaining method is used to interconnect an HX-20 and a TF-20 or two TF-20 units via cable set #707 (for daisy chaning). TF-20 (disk 1) and TF-20 (disk 2) can be interconnected in any order. Fig. 10-1 shows how two TF-20 units are connected to the HX-20.

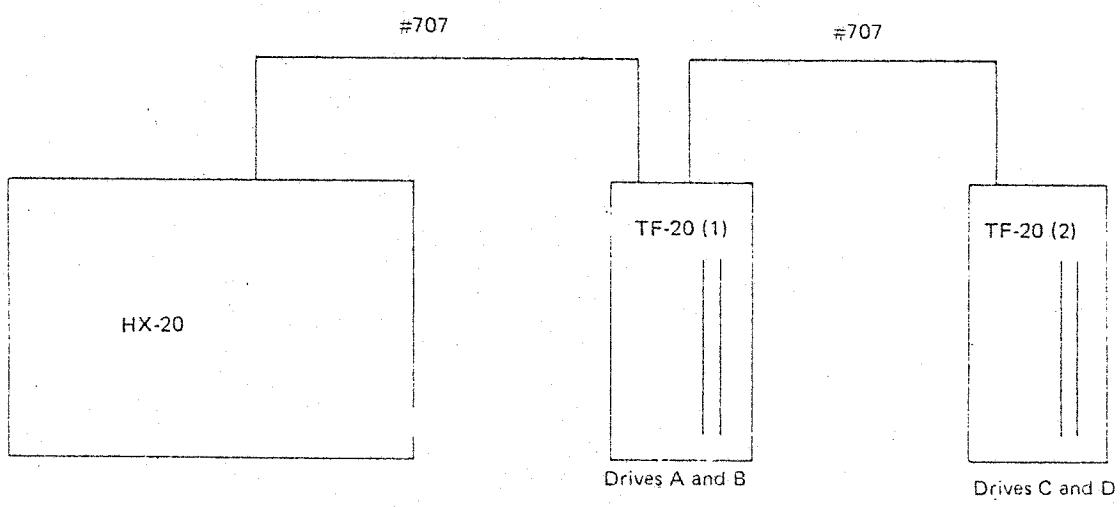


Fig. 10-1 Interconnection of HX-20 and Two TF-20 Units

10.2 Disk Format

Disk type:	Double-sided, double density (MFM)
No. of tracks:	80 tracks (40 tracks x 2 sides)
Track density:	48 TPI
No. of sectors:	16 sectors/track
Capacity per sector:	256 bytes

Total disk capacity: 320K bytes (256 x 16 x 80)
Access time between
tracks: 15 ms

Tracks and sectors are logically structured as shown below.

No. of tracks: 40 tracks (0 to 39)
No. of sectors: 64 sectors/track (1 to 64)
Capacity per sector: 128 bytes

Table 10.1 shows the relationship between the physical and logical specifications.

Table 10.1 Relationship between Physical and Logical Specifications

	Physical specifications	Logical specifications
Track	One track on one side + one track on the other side	One track
Sector	One sector (256 bytes)	Two sectors (128 bytes x 2)

10.3 System Disk and Non-system Disk

The floppy disks used in DISK BASIC can be divided into a system disk and a non-system disk. Either of these two disks must be initialized by the physical format of the TF-20 for generation of correct directories.
All the floppy disks supplied by EPSON have been initialized before shipment so that they can be used as non-system disks. Floppy disks other than those supplied by EPSON and those disks in which a read or write error has occurred must be initialized by the FRMAT command. The system disk refers to the disk which contains a system program for DISK BASIC, and must be inserted into drive "A:" when DISK BASIC is to be booted. The system disk is mapped as follows.

Track 0	Sectors 1 and 2:	Cold-start loader (loads a system contained in the system disk into the memory of the TF-20.)
	Sectors 3 to 18:	Unused
	Sectors 19 to 46:	BDOS (Basic Disk Operating System)
	Sectors 47 to 64:	BIOS (Basic Input/Output System) for the HX-20
Track 1	Sectors 1 to 42:	TFDOS (communication program with the HX-20)
	Sectors 43 to 64:	Unused
Tracks 2 and 3	Sectors 1 to 64:	Unused
Track 4	Sectors 1 to 16:	Directory area (for 64 directories max.)
	Sectors 17 to 64:	File area
Tracks 5 to 39	Sectors 1 to 64:	File area (278K bytes max.)

Two files "BOOT80.SYS" and "DBAISC.SYS" are secured for the system in the system disk. Since these files are write-protected, their filenames are not displayed even by executing the FILES command. Note that the user cannot use the same filenames as these two files. To duplicate a system disk, either copy all the contents of the existing system disk to a new floppy disk by COPY utility, or execute the SYSGEN command for a non-system disk.
"SYSGEN" copies not only the system area of the disk but also copies the system file whose file type is "SYS".

10.4 Interface with DISK BASIC

The DISK BASIC is broadly divided into the following 3 modules:
(1) BASIC interpreter (ROM version: HX-20 side)
(2) DBASIC interpreter (DBASIC.SYS: HX-20 side)
This interpreter is an extended portion of BASIC which is loaded from a disk to the RAM of the HX-20 upon start of the BASIC and handles the data input/output to and from the disk and the processing of commands and statements, together with the BASIC interpreter described in (1) above. This module consists mainly of a portion connected to the BASIC interpreter (i.e., a BASIC driver) and a portion interfacing with the TFDOS (i.e., EPSP driver).
(3) TFDOS (TF-20 side)
The TFDOS which is resident on the RAM of the TF-20 receives commands from the HX-20, opens and reads or writes files using the BDOS or the BIOS for the HX-20, and returns data and error codes to the HX-20.
Of the above 3 modules, the BASIC driver and EPSP driver of the DBASIC interpreter are interfaced with each other through the BSCINT (BASIC interface), while the EPSP driver is interfaced with the TFDOS through the EPSP (EPSON Serial Communication Protocol) as shown in Fig. 10-2.

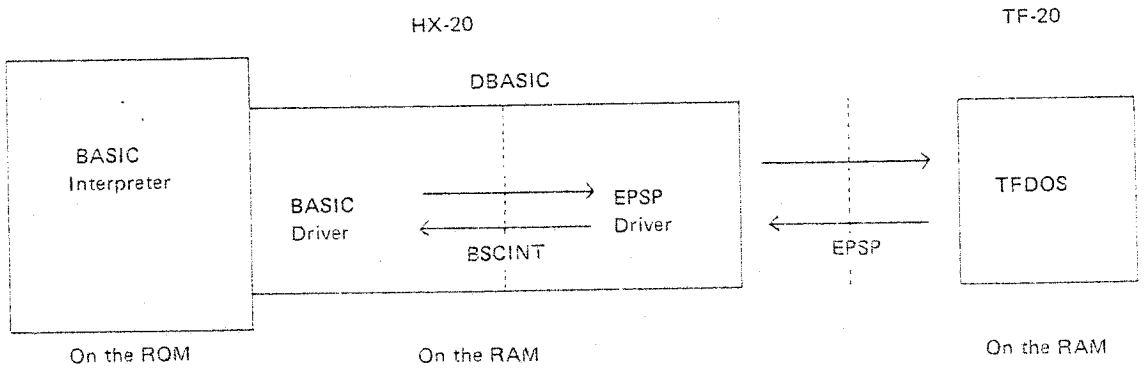


Fig. 10-2 Software Configuration of Disk BASIC

10.4.1 BASIC interface (BSCINT)
10.4.1.1 Functions of BSCINT
Interfacing of DBASIC with BASIC is supported by subroutine "BSCINT" (BASIC Interface) which has the following functions:

(1) File open
(2) File close
(3) Random read (128 bytes)
(4) Random write (128 bytes)
(5) File delete
(6) File rename
(7) File size calculation
(8) First directory search
(9) Next directory search
(10) Direct write into disk (DSKO$, 128 bytes)
(11) Disk formatting (FRMAT)
(12) Disk system reset (RESET)
(13) System disk generation (SYSGEN)
(14) Disk free area calculation (DSKF)
(15) Direct read from disk (128 bytes)
(16) Disk all copy

10.4.1.2 Subroutine call procedure
Subroutine "BSCINT" is called as follows:

(1) Setting the entry point for BSCINT
The contents at an address 3 bytes from addresses (0A3E and 0A3F) are "JMP BSCINT" (see Fig. 10.3). This means that the address specified by addresses (0A3E and 0A3F) is the entry point of the subroutine that includes BSCINT error processing.

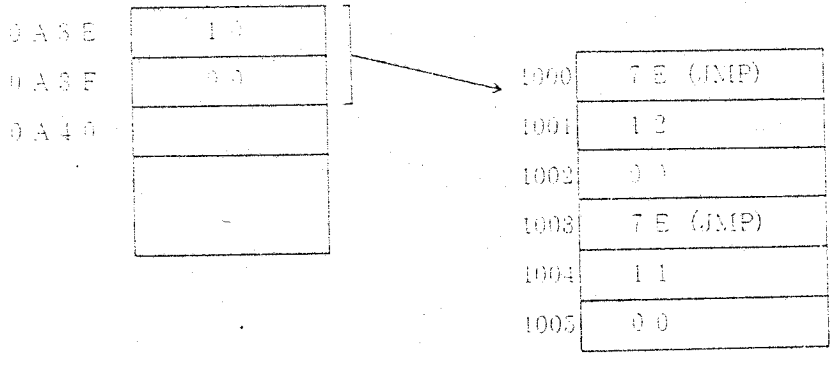


Fig. 10-3 BSCINT Entry Point

(2) Creation of a parameter packet
Parameters are created on memory, and are given in the order of the function code, return code, and data (see Fig. 10-4). The data string has a length of one or more bytes. For details of the functions and parameters, refer to the BSCINT parameter packet table.

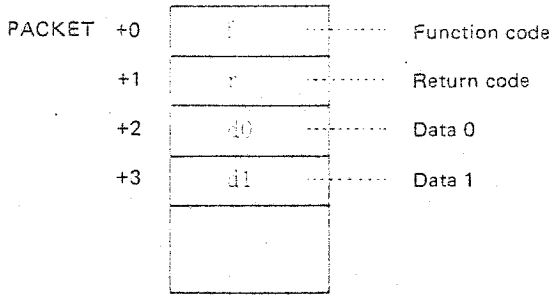


Fig. 10-4 Parameter Packet of Subroutine "BSCINT"

(3) Subroutine call
The first address of the parameter packet is set in the index register to call subroutine "BSCINT".
An example of opening a file is shown below:

[Example] File under the file descriptor "ABC.BAS" is opened in Sequential Output mode using drive "A".

```
        LDAA  #$7E      * (JMP instruction)
        STAA  BSENTR
        LDD   $A3E
        STD   BSENTR+1
        LDX   #CPOPC
        JSR   BSENTR
        LDAA  1,X
        BNE   ERROR
        RTS
          :
ERROR   EQU   *         * error procedure
          :
BSENTR  FCB   $7E       * (JMP BSCINT)
        RMB   2

CPOPC   FCB   $00
        FCB   $00
        FCB   $00
        FCC   /ABC△△△△△△/
        FCC   /BAS/
```

10.4.2 BSCINT parameter packet table

*All packet data numbers are decimal numbers.

No.	Function	Packet data No.	Description
1	File open		Opens the file in the specified drive according to the filename, file type, and file mode.
		00	00 (function code)
		01	Return code (set at return)
		02	File number (set at return)
		03	Drive number ("A", "B", "C" or "D")
		04 ~ 11	Filename (8 characters. If the filename is less than 8 characters, left-justify the filename and fill blank code(s) (20) in the remaining space.)
		12 ~ 14	File type (3 characters. If the file type is less than 3 characters, left-justify the file type and fill blank codes (20) in the remaining space.)
		15	Modes 10_{16}: Sequential input (M.SQI) 30_{16}: Sequential output (M.SQO) 40_{16}: Random access (M.RND) If no file exists in M.SQI or M.SQO mode, a new file is created. If no file exists in M.SQI mode, it is assumed that an error has occurred. If a file exists in M.SQO mode, the previous file will be deleted.
2	File close		Closes the specified opened file.
		00	01 (function code)
		01	Return code (set at return)
		02	File number (i.e., the number returned at a file open)
3	Random read		Reads the specified record of a file. (One record consists of 128 bytes.)
		00	02 (function code)
		01	Return code (set at return)
		02	File number (i.e., the number returned at a file open)
		03 ~ 04	Record number (binary value in the range of 1 to 65535. Must be entered in the order of high- and low-order bytes.)
		05 ~ 06	Buffer address (must be entered in the order of high- and low-order bytes.)
4	Random write		Writes the specified record of a file. (One record consists of 128 bytes.)
		00	03 (function code)
		01	Return code (set at return)
		02	File number (i.e., the number returned at a file open)
		03 ~ 04	Record number (binary value in the range of 1 to 65535. Must be entered in the order of high- and low-order bytes.)
		05 ~ 06	Buffer address (must be entered in the order of high- and low-order bytes.)

No.	Function	Packet data No.	Description
5	File delete		Deletes the specified file.
		00	04 (function code)
		01	Return code (set at return)
		02	Unused
		03	Drive name ("A", "B", "C" or "D")
		04 ~ 11	Filename (8 characters. If the filename is less than 8 characters, left-justify the filename and fill blank codes (20) in the remaining space.)
		12 ~ 14	File type (3 characters. If the file type is less than 3 characters, left-justify the file type and fill blank codes (20) in the remaining space.)
6	File rename		Rename the existing file.
		00	05 (function code)
		01	Return code (set at return)
		02	Unused
		03	Drive name ("A", "B", "C" or "D")
		04 ~ 11	Filename before change (8 characters)
		12 ~ 14	File type before change (3 characters)
		15 ~ 22	Filename after change (8 characters)
		23 ~ 25	File type after change (3 characters)
7	File size calculation		Returns the number of records of the specified file. (One record consists of 128 bytes.)
		00	06 (function code)
		01	Return code (set at return)
		02	File number (i.e., the number returned at a file open)
		03 ~ 04	Maximum value of a record number (the value must be in the range of 0 to 65535. 0 indicates the null state.)
8	First directory search		Returns the FCB (file control block) address and directory code on the disk of the file for which the filename and file type were specified. If the filename and file type are all specified by character '?', it is assumed that file matching has been completed for all files.
		00	07 (function code)
		01	Return code (set at return)
		02	Unused
		03	Drive name ("A", "B", "C" or "D")
		04 ~ 11	Filename (8 characters)
		12 ~ 14	File type (3 characters)
		15	Directory code (set at return)
		16 ~ 47	Directory FCB (set at return)

No.	Function	Packet data No.	Description
9	Next directory search		Searches the next directory. (This function is performed next to the function No. 8 above.) The method of specifying the filename and file type is the same as function No. 8.
		00	08 (function code)
		01	Return code (set at return)
		02	Unused
		03	Drive name ("A", "B", "C" or "D")
		04 ~ 11	Filename (8 characters)
		12 ~ 14	File type (3 characters)
		15	Directory code (set at return)
		16 ~ 47	Directory FCB (set at return)
10	Direct write into disk (DSK0$)		Writes data into the specified tracks and sectors of floppy disk.
		00	09 (function code)
		01	Return code (set at return)
		02	Unused
		03	Drive name ("A", "B", "C" or "D")
		04	Track number (binary value in the range of 0 to 39_{10})
		05	Sector number (binary value in the range of 1 to 64_{10})
		06 ~ 07	Buffer address (must be entered in the order of high- and low-order bytes)
11	Disk formatting (FRMAT)		Formats the floppy disk in the specified drive.
		00	0A (function code)
		01	Return code (set at return)
		02	Unused
		03	Drive name ("A", "B", "C" or "D")
12	Disk system reset		Enables a disk replacement. When the disk system is reset, all the disks can be read or written and disk drive "A" is selected.
		00	0B (function code)
		01	Return code (set at return)
		02	Unused
		03	Drive name ("A", "B", "C" or "D")
13	System disk generation (SYSGEN)		Copies the system area and file of the system disk set in drive "A", to the disk set in drive "B". After copying, the disk in drive "B" can be used as a system disk.
		00	0C (function code)
		01	Return code (set at return)

No.	Function	Packet data No.	Description
14	Disk free area calculation (DSKF)		Provides the free area size of the disk in the specified drive in 2K-byte units.
		00	0D (function code)
		01	Return code (set at return)
		02	Unused
		03	Drive name ("A", "B", "C" or "D")
		04	Free area size (binary value in 2K-byte units set at return)
15	Direct read from disk (DSKI$)		Reads data from the specified tracks and sectors of a floppy disk.
		00	0E (function code)
		01	Return code (set at return)
		02	Unused
		03	Drive name ("A", "B", "C" or "D")
		04	Track number (binary value in the range of 0 to 39_{10})
		05	Sector number (binary value in the range of 1 to 64_{10})
		06 ~ 77	Buffer address (must be entered in the order of high- and low-order bytes. In this case, however, the message work area of EPSP driver routine is used.)
16	Disk all copy		Copies all the contents of the floppy disk in the specified drive to the disk in the other drive of the same floppy disk unit. (i.e., from "A" to "B", from "C" to "D")
		00	0F (function code)
		01	Return code (set at return)
		02	Unused
		03	Drive name ("A", "B", "C" or "D")
			NOTE: With drives "A" and "B", disk copying must be from "A" to "B". With drives "C" and "D", disk copying must be from "C" to "D".

10.4.3 BSCINT return codes

Code (Hex)	Meaning
00	Normal completion of operation
01	The specified file is not found.
02	End of File (EOF) was detected during file input.
03	The file already exists.
04	The specified device is not found.
05	No directory area exists.
06	No disk area exists.
07	The specified record number is incorrect.
08	The disk is write-protected.
09	The file is not opened.
0A	The specified file number is incorrect.
0B	The specified file mode is incorrect.
0C	The specified file is already open.
0D	The number of opened files is too many.
0E	The specified file descriptor is incorrect.
0F	An error has occurred during a read operation.
10	An error has occurred during a write operation.

10.5 EPSP (EPSON Serial Communication Protocol)

10.5.1 EPSP functions

The EPSP is an interface between the EPSP driver and the TFDOS as described in Chapter 4. The EPSP on the TF-20 side has the following functions:

(1) Disk system reset
    Corresponds to Item (12) of paragraph 10.4.1.
(2) File open
    Corresponds to Item (1) of paragraph 10.4.1.
(3) File close
    Corresponds to Item (2) of paragraph 10.4.1.

(4) First directory search
Corresponds to Item (8) of paragraph 10.4.1.
(5) Next directory search
Corresponds to Item (9) of paragraph 10.4.1.
(6) File delete
Corresponds to Item (5) of paragraph 10.4.1.
(7) File creation
By this function, the directory and memory are initialized and a file empty of data is created.
(8) Random read
Corresponds to Item (3) of paragraph 10.4.1.
(9) Random write
Corresponds to Item (4) of paragraph 10.4.1.
(10) File size calculation
Corresponds to Item (7) of paragraph 10.4.1.
(11) Disk all copy
Corresponds to Item (16) of paragraph 10.4.1.
(12) Direct write (128 bytes) into disk (DSKO$)
Corresponds to Item (10) of paragraph 10.4.1.
(13) Disk formatting (FRMAT)
Corresponds to Item (11) of paragraph 10.4.1.
(14) System disk generation (SYSGEN)
Corresponds to Item (13) of paragraph 10.4.1.
(15) Disk free area calculation (DSKF)
Corresponds to Item (14) of paragraph 10.4.1.
(16) Direct read (128 bytes) from disk (DSKI$)
Corresponds to Item (15) of paragraph 10.4.1.
(17) Disk boot
By this function, file "BOOT80.SYS" is booted to the HX-20 from the system disk in the disk drive A of the TF-20. In other words, this function opens file "BOOT80.SYS", reads 128 bytes of data only and transfers them to the HX-20.
(18) Load open
By this function, file "DBASIC.SYS" contained in the system disk in the drive A of the TF-20 is opened and then loaded into the RAM of the TF-20. After the loading, the file is relocated on the RAM of the TF-20 using a relocatable flag (one of the load open parameters) and an ending or starting address. Return code "FF" if the corresponding file is not found, or return code "00" if found, is returned to the HX-20 together with the file size of "DBASIC.SYS".
(19) Load close
This function indicates that the transfer of file "DBASIC.SYS" has been completed. In this case, the TF-20 does not perform any function.
(20) Read one block
By this function, the file "DBASIC.SYS" opned, read, and relocated in item (18) above is transferred to the HX-20 in units of 128 bytes.
Return code "FF" indicates the end of file (EOF).

10.5.2 Subroutne "OUTSRL"

Subroutine "OUTSRL" handles the data transmission/reception of EPSP as follows:

(1) Creation of a parameter packet

Parameters are given in the form of a packet as shown in Fig. 10-5.

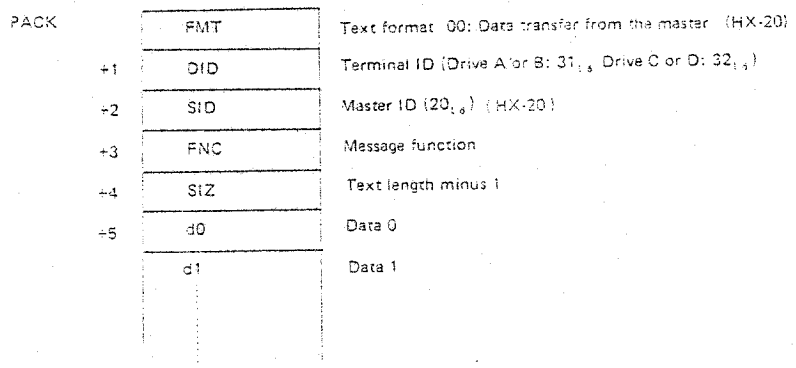


Fig. 10-5 Parameter Packet of Subroutine "OUTSRL"

(2) Subroutine call

The first address of the parameter packet is set in the index register to call subroutine "OUTSRL" (entry point: FF70). For details of the EPSP, refer to Chapter 4. For details of the EPSP functions on the TF-20 side, refer to the next page.

*EPSP side

Open file
Drive : "A", filename, file type: ABC, BAS
File mode: Sequential output "0"

```
OUTSRL  EQU  $FF70
        LDX  #PACKET
        JSR  OUTSRL      ... Routine for data output
                             to the serial interface
         :

PACKET  EQU  *
FMT     FCB  $00,$30,$20,$0F,$0E
MSG     FCB  $00,$01,$01,
        FCC  /ABC△△△△△/
        FCC  /BAS/
```

10.6 Function Table of Floppy Disk Unit

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	Extent 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	Extent 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.)

FMT	DID	SID	FNC	SIZ	Text data No.	Description of function and text
						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	Extent 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 (3 characters)
					0C	Extent number
					0D ~ 0F	Unused
					10	Drive code (1 or 2)
					11 ~ 18	Filename after change (8 characters)
					19 ~ 1B	File type after change (3 characters)
					1C	Extent 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	Extent 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

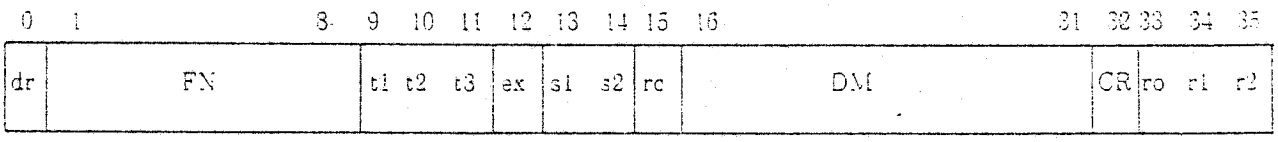
FMT	DID	SID	FNC	SIZ	Text data No.	Description of function and text
						Random data write
00	SS	MM	22	84	00	High-order byte of FCB address in HX-20
					01	Low-order byte of FCB address in HX-20
					02 ~ 81	Write data (128 bytes)
					82	R1)
					83	R1) Random record numbers
					84	R2)
01	MM	SS	22	02	00	Extent 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	Extent 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)

FMT	DID	SID	FNC	SIZ	Text data No.	Description of function and text
						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

FMT	DID	SID	FNC	SIZ	Text data No.	Description of function and text
						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 number
					01	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)

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).

The format of the file control block (FCB) used by the floppy disk unit is as follows:



dr: Disk drive code (00 to 16) (Use of code 05 to 16 will result in an error.)
00: A file is assigned to the standard disk drive.
01: Disk and disk drive A are selected automatically.
02: Disk and disk drive B are selected automatically.

16: Disk and disk drive P are selected automatically.

FN: Filename consisting of a maximum of 8 characters (in ASCII codes).
If no filename is given by the user, blanks (20) will be filled as the filename.

t1, t2, t3: File format (in ASCII codes)
As ASCII codes, bits in the upper row are selected and high-order bits set to 0 are used. These bits when represented by t1, t2 and t3 are as follows.
t1=1: Read only file
t2=1: No system file, FILES list

ex: File extent (Normally 0)
This is a number to indicate the current location of the logical extent, and is normally set to 00 by the user. This number must be a value in the range of 0 to 31 when a file input/output is to be performed.

s1:	Used within the system.
s2:	Used within the system. s2 is set to 0 when a file is to be opened, created, or called for search.
rc:	Record number of the logical extent indicated by "ex" and must be a value in the range of 0 to 128.
DM:	A value set and used by the system.
cr:	A value indicating the location of the record where data read/write is being performed in sequential file processing. This value is normally set to 0 by the user.
r0, r1, r2:	Random record number indicated by a value in the range of 0 to 65535. r0, r1 and r2 are used to configure 24 bits. r0 indicates the low-order digit, r1 the high-order digit, and r2 an overflow.

CHAPTER 11 SLAVE MCU COMMANDS

11.1 General

The interface between the master and slave MCUs consists of two signal lines. Serial communication is performed at 38.4K BPS. Slave MCU operations are performed in response to instructions (commands) sent from the main MCU. The master CPU uses the serial interface to communicate either with the slave MCU or externally.

The slave CPU supports the following functions:

(1) Operation of the microprinter
(2) Data reception via RS-232C port
(3) Data I/O for external cassette
(4) Data I/O and operation of built-in microcassette
(5) Output for piezoelectric speaker
(6) Control switches for serial, power supply and bar code reader power

11.2 Commands for Slave MCU Control

Commands are sent to the slave MCU via the 38.4K-BPS serial interface. Commands are one byte in length. However, for some commands, parameters are added. The standard communication procedure involves sending a command from the master MCU and receiving an ACK signal from the slave MCU in response. The sequence for commands sent with parameters is shown below.

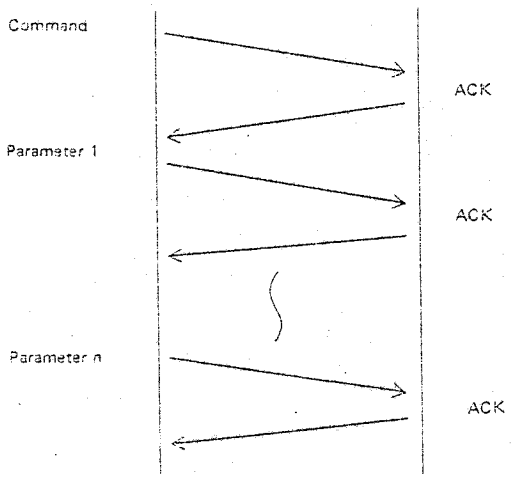


Fig. 11-1 Command Transmission Sequence

First, a 1-byte command is sent to the slave MCU. The SNSCOM subroutine (entry point FF19) is called to receive the ACK signal. For details of commands, see the command table.

For data reception from the RS-232C or cassette, the slave MCU sends serial input data to main MCU upon completion of command reception. Data received by the slave MCU under this condition are assumed to be commands and the current input mode is cancelled.

11.3 Cancelling a Command

The command being executed is cancelled if an overrun error occurs during serial communication. (For example, if overrun occurs when 100-line feed is specified for the microprinter, the current command is aborted and the system goes into WAIT status pending receipt of a fresh command.) If new data is received from main MCU while a command is being executed by the slave MCU, the data is set in the receive register but not processed. At this point, if new serial communication data is received, the data in the register is destroyed, causing an overrun error.

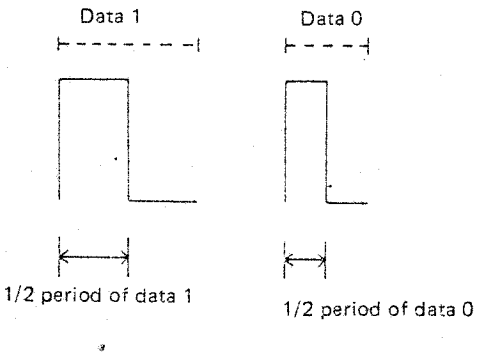
To cancel a command, the master MCU sends a series of BREAK commands to the slave MCU. Subroutine BREAKIO (entry point FFA3) is provided for this purpose.

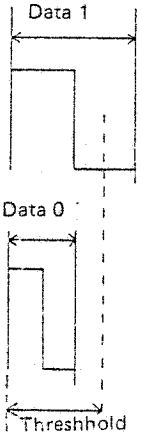
11.4 Slave MCU Command Transmission Subroutine

Subroutine name	Entry point	Description
SNSCOM	FF19	Transfers a command or 1 byte of data to the slave MCU via SCI. Parameters: At Entry (A) Transmit data (Command) At Return (C): Abnormal I/O flag (A): Return code (Transmit data from slave MCU) Registers retained (B), (X) Subroutines referenced None Variables used None

11.5 Commands to Slave MCU

Command	Master MCU data	Slave MCU response	Description
00	00	01 (ACK)	Slave MCU ready check. ACK is returned when the slave MCU is ready to receive a command. The slave MCU makes no response if it is not ready.
01	01	01 (ACK)	Sets the constants required by slave MCU in the field. The following values are set: Generated polynomial expressions, BCC register value, RS-232C bit rate, cassette (external or built-in microcassette), micro-cassette tape counter setting.
02	02	01 (ACK)	Initialization. The status of serial communication driver remains unchanged.
03	03 (Command) AA (Parameter)	01 (ACK)	Opens masks for special commands. Commands 06, 07, 08 and 0B cannot be executed unless the masks are opened. Any value other than AA indicates that the mask is closed.
04	04	01 (ACK)	Closes masks for special commands.
05	05 ah (Upper byte of address) al (Lower byte of address)	01 (ACK) 01 (ACK) d (Data)	Reads slave MCU memory. NAK (0F) is returned in response to 05 if the mask is not open.
06	06 ah (Upper byte of address) al (Lower byte of address) d (Data)	01 (ACK) 01 (ACK) 01 (ACK) 01 (ACK)	Stores data to the memory address specified by the slave MCU. 0F (NAK) is returned and command execution is aborted if the mask is not opened.
07	07 ah (Upper byte of address) al (Lower byte of address) d (Data)	01 (ACK) 01 (ACK) 01 (ACK) 01 (ACK)	Performs logical OR operation for the data at the memory address specified by the slave MCU and the specified data and stores the result in the specified address. 0F (NAK) is returned and command execution is aborted if the mask is not opened.
08	08 ah (Upper byte of address) al (Lower byte of address) d (Data)	01 (ACK) 01 (ACK) 01 (ACK) 01 (ACK)	Performs logical AND operation for the data at the memory address specified by the slave MCU and the specified data and stores the result in the specified address. 0F (NAK) is returned and command execution is aborted if the mask is not opened.

Command	Master MCU data	Slave MCU response	Description
09	09	01 (ACK)	Unused (In version 2, bar-code reader power ON)
0A	0A	01 (ACK)	Unused (In version 2, bar-code reader power OFF)
0B	0B ah (Upper byte of address) al (Lower byte of address)	01 (ACK) 01 (ACK) 01 (ACK)	Sets the program counter to a specified value. (Jumps execution to a specified address.) 0F is returned and command execution is aborted, if the mask is not opened.
0C	0C	02 (ACK for BREAK)	BREAK. Terminates processing and sets the system to command WAIT status.
0D	0D AA	01 (ACK) 01 (ACK)	Cuts OFF power supply. Command execution is aborted if parameter AA is omitted.
0E ~ 0F			Undefined
10	10 d (Data)	01 (ACK) 01 (ACK)	Activates the built-in printer. Prints out 6-dot data (Bit 0 to bit 5). One dot-line is printed by repeating this command procedure 24 times.
11	11 d (Number of lines)	01 (ACK) 01 (ACK)	Feeds the specified number of dot lines to the built-in printer.
12	12	01 (ACK)	Activates built-in printer motor for approx. 1.2 sec. (Paper feed operation)
13 ~ 1F			Undefined
20	20	21 (ACK)	Executes external cassette ready check. Code 21 is returned when the external cassette is ready. The external cassette makes no response if it is not ready.
21	21 d1 (Upper byte of time (MCU clocks) of 1/2 cycle for data '1') d2 (Lower byte of time (MCU clocks) of 1/2 cycle for data '1') d3 (Upper byte of time (MCU clocks) of 1/2 cycle for data '0')	01 (ACK) 21 (ACK) 21 (ACK) 21 (ACK)	Sets constants for the external cassette. 

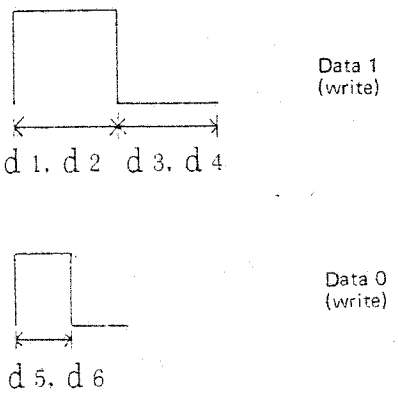
Command	Master MCU data	Slave MCU response	Description
	d4 (Lower byte of time (MCU clocks) of 1/2 cycle for data '0')	21 (ACK)	The times (in MCU clock pulses) for 1/2 cycle for data '1' and for data '0' are set as constants. The bit judgment threshold value for data read is also set as the number of MCU clocks (Fig. below).
	d5 (Upper byte of bit judgment threshold value between cycle times for '1' and '2')	21 (ACK)	This data represents the interblock gap length in tape stop mode (long gap) as the number of times that data FF is written to the tape.
	d6 (Lower byte of bit judgment threshold value)	21 (ACK)	
	d7 (Upper byte of interblock gap length (in bytes) in stop mode (tape head stops between blocks))	21 (ACK)	
	d8 (Lower byte of interblock gap length in stop mode)	21 (ACK)	
22	22	01 (ACK)	Turns the external cassette REM terminal ON.
23	23	01 (ACK)	Turns the external cassette REM terminal OFF.
24	24	01 (ACK)	Writes 1 block of data in EPSON format. After synchronizing pattern is sent, the number of bytes specified as the block length is written followed by 2 CRC bytes. For output data, only the number of bytes specified as the block length are required. If data has not been received from the master MCU when the slave attempts to write data to the cassette, the slave MCU returns 2F, activates the speaker (880Hz for 1 sec.) and terminates cassette output. Block write start mode values are as follows (d1) 00: 125-byte gap before the block (default valve). 01: 15-byte gap before the block. FF: 625-byte gap before the block. Block write start mode value (00, 01 or FF) is used as the block write end mode value at the completion of block write operation. In 00 and FF modes, the REM terminal is turned after completion of block write operation.
	d1 (Block write start mode)	21 (ACK)	
	d2 (Block write end mode)	21 (ACK)	
	d3 (Upper byte of block length)	21 (ACK)	
	d4 (Lower byte of block length)	21 (ACK)	
	W1 (Output data)	22 (ACK)	
	⟨	(2F(NAK))	
	Wm (Output data)	22 (ACK)	
		(2F(NAK))	

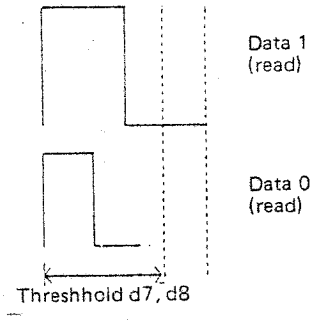
Command	Master MCU data	Slave MCU response	Description
25	25 d1 (Upper byte of number of FF patterns) d2 (Lower byte of number of FF patterns)	01 (ACK) 21 (ACK) 21 (ACK)	Outputs number of FF patterns specified by d1 and d2 to the external cassette. Writing of data FF is unrelated to blocking.
26	26 d1 (Block read start mode) d2 (Block read end mode) d3 (Upper byte of block length) d4 (Lower byte of block length)	01 (ACK) 21 (ACK) 21 (ACK) 21 (ACK) 21 (ACK) W1 W2 W3 { W84	Inputs files from an external cassette. Searches header block (EPSON format) and sends the contents of this block to the master MCU. Header block always begins with data H. In actual practice, however, d1 is ignored. REM is turned OFF after reading 1 block if d2 is 00. If d2 is 01, REM is left ON. If an error occurs during transmission of block data, data transmission is terminated and P34 (connected to P12 of the master MCU) is turned ON. Two CRC bytes are placed at the end of the block but are not transmitted.
27	27 d1 (Block read start mode) d2 (Block read end mode) d3 (Upper byte of block length) d4 (Lower byte of block length)	01 (ACK) 21 (ACK) 21 (ACK) 21 (ACK) 21 (ACK) W1 W2 W3 { W84	Inputs files from an external cassette. Seaches EOF block (EPSON format) and sends the contents of this block to the master MCU. EOF block always begins with data E. Parameters and execution result are identical to those for command 26.
28	28 d1 (Block read start mode) d2 (Block read end mode) d3 (Upper byte of block length) d4 (Lower byte of block length)	01 (ACK) 21 (ACK) 21 (ACK) 21 (ACK) 21 (ACK) W1 W2 W3 { W260	Inputs files from an external cassette. Inputs the next block (EPSON format) and sends the data to the master MCU. The block may beging with any data. Parameters and execution result are identical to those for command 26.
29			Undefined
2A			Undefined

Command	Master MCU data	Slave MCU response	Description
2B	2B d1 (Specifies the pulse mode)	01 (ACK) 21 (ACK)	Specifies the input signal for the external cassette and built-in micro-cassette. d1 Bit 3: When logic '1', the microcassette input signal is as defined by bit 2. When logic '0', the microcassette input signal is judged at input. Bit 2: When logic '1', the microcassette input signal is reversed. When logic '0', the microcassette input signal is normal. Bit 1: When logic '1', the external cassette input signal is defined by bit 0. When logic '0', the external cassette input signal is judged at input. Bit 0: When logic '1', the external cassette input signal is reversed. When logic '0', the external cassette input signal is normal. NOTE: In versions 1 and 2, the slave MCU assumes (Bit 3, Bit 2) = (1, 1) when bit 3 is logic '0'.
30	30 d1 (Tone) d2 (Duration)	01 (ACK)	Specifies the tone and duration and sounds the piezoelectric speaker. The specifications for tone are as follows: 0 = pause, 1, 2, 3 ... correspond to C, D, E ... . Values 1 to 28_{10} represent a 4-octave major scale (13 = 880Hz) and values 29 to 56_{10} a scale each tone of which is a half tone higher than that represented by 1 to 28. Duration is specified with 1 = 0.1 sec., 2 = 0.2 sec, etc. 0 specifies a pause (command not executed).
31	31 d1 (Upper byte of frequency specification) d2 (Lower byte of frequency specification) d3 (Upper byte of duration specification) d4 (Lower byte of duration specification)	01 (ACK) 31 (ACK) 31 (ACK) 31 (ACK) 31 (ACK)	Specifies the frequency and duration and sounds the piezoelectric speaker. Frequency is specified as the number of MCU clocks corresponding to 1/2 cycle. Example: 349_{10} for 880Hz. Specification of duration: 1 = 400μsec. (256 MCU clocks)
32	32	01 (ACK)	Sounds the speaker for 0.03 sec at tone 6 using command 30.
33	33	01 (ACK)	Sounds the speaker for 1 sec at tone 20 using command 30.

Command	Master MCU data	Slave MCU response	Description
34	34 d s1 d 11 d s2 d 12 d sn d 1n FF	01 (ACK) 31 (ACK) 31 (ACK) 31 (ACK) 31 (ACK) 31 (ACK) 31 (ACK) 31 (ACK)	Sets melody data in the slave MCU buffer. Buffer size is 48 bytes. The data set here can be output to the speaker using command 35. The format for data is the same as for command 30, i.e., tone, duration. As a pair, these data repeatedly specify the tone and duration. Due to the buffer size, the maximum number of data is 46_{10}. Data must end with FF. The data set in the buffer remains unchanged unless it is rewritten by command 34 or destroyed by a printer command. (This is because this buffer is also used by printer.)
35	35	01 (ACK)	Sounds the piezoelectric speaker in accordance with the melody data specified in command 34.
36 ~ 3F			Undefined
40	40	01 (ACK)	Turns the serial driver ON. RTS is set to low (OFF).
41	41	01 (ACK)	Turns the serial driver OFF.
42	42 d1 (Upper byte of bit rate) d2 (Lower byte of bit rate) d3 (Word length) d4 (Mode)	01 (ACK) 41 (ACK) 41 (ACK) 41 (ACK) 41 (ACK)	Selects RS-232C mode. Bit rate corresponds to bit time specified as the number of CPU clock cycles (for example, 800_{16}: 300 BPS). Word length (excluding parity bits) may be set at 5, 6, 7 or 8 bits. The significance of each bit of mode data (d4) is as follows. Bit 0, Bit 1: Number of stop bits (1 or 2) Bit 2: '0': Carrier check '1': No carrier check Bit 3: Controls RTS output. '0': low '1': high Bit 4: Undefined Bit 5: Undefined Bit 6, Bit 7: Parity control. '00': Even parity '01': Odd parity '10' or '11': None
43	43	V	Inputs RS-232C status maintained by the slave MCU. The significance of each bit of the status code is as follows. (Logic '1' indicates an error.)

Command	Master MCU data	Slave MCU response	Description
43	43	V	Bit 0: Carrier detect Bit 1: Parity Bit 2: Overrun Bit 3: Framing Bit 4: Undefined Bit 5: Undefined Bit 6: Undefined Bit 7: Undefined Error status bits are reset by a clear command (44) or when input is resumed (command 45).
44	44	01 (ACK)	Clears RS-232C error status.
45	45	01 (ACK) V1 V2	Starts RS-232C input. Input data is sent to the master MCU. If the word length of the data (including the parity bits) is less than 8 bits, the remaining bits (from MSB) are padded with data 0 (right-justified). P34 (connected to master MCU P12) is reset (logic '1')when input starts. P34 is set (logic '1') if an error (framing error, carrier OFF, etc.) occurs. Data reception terminates upon receipt of a new command from the master MCU.
46	46	01 (ACK)	Terminates RS-232C input initiated by command 45. (This is not the only way of terminating such input.)
47			Undefined
48	48 d1 (Upper byte of polynomial expression) d2 (Lower byte of polynomial expression)	01 (ACK) 41 (ACK) 41 (ACK)	Sets the polynomial expression used for CRC check. This polynomial expression can also be used for cassette files. Default value is 8408 ($1+X^5+X^{12}+X^{16}$).
49	49 d1 (Upper byte of BCC register value) d2 (Lower byte of BCC register value)	01 (ACK) 41 (ACK) 41 (ACK)	Sets BCC register values for CRC check. This value is used as the initial value when CRC calculation is performed at RS-232C input. However, the data in BCC register is lost when I/O operations to a cassette are performed.
4A	4A	V	Inputs upper byte of BCC register value.

Command	Master MCU data	Slave MCU response	Description
4B	4B	V	Inputs lower byte of BCC register value.
4C	4C	01 (ACK)	Activates the serial driver. In contrast to command 40 which turns RTS OFF, this command does not affect the status of RTS.
4D	4D	01 (ACK)	RTS high/low specification. Only bit 0 is significant. 0: low, 1: high
4E			Undefined
4F			Undefined
50	50	V	Identifies the plug-in option. Bit states of P46 and P20 are returned. Bit 0: Bit state of P46 Bit 1: Bit state of P20 Bit 2 to 7: 0 NOTE: Plug-in option power is turned OFF when this command is executed.
51	51	01 (ACK)	Turns power of plug-in ROM cartridge ON.
52	52	01 (ACK)	Turns power of plug-in ROM cartridge OFF.
53 ~ 5F			Undefined.
60	60	61 (ACK)	Executes ready check. Same MCU responds only if a microcassette command is executable. In all other cases, no response is sent.
61	61 d1 (Upper byte of signal low time of one cycle for data '1') d2 (Lower byte of signal low time of one cycle for data '1') d3 (Upper byte of signal high time of one cycle for data '1') d4 (Lower byte of signal high time of one cycle for data '1')	01 (ACK) 01 (ACK) 61 (ACK) 61 (ACK) 61 (ACK)	Sets the microcassette parameters. Parameters are specified using data d1 to d8. 

Command	Master MCU data	Slave MCU response	Description
	d5 (Upper byte of time of 1/2 cycle for data '0') d6 (Lower byte of d5) d7 (Upper byte of '0', '1' bit judgment threshold value) d8 (Lower byte of d7)	61 (ACK) 61 (ACK) 61 (ACK) 61 (ACK)	
62	62 d1 (Upper byte of number of gap bytes) d2 (Lower byte of number of gap bytes)	01 (ACK) 61 (ACK) 61 (ACK)	Specifies the number of gap bytes for each mode when stopping the microcassette between blocks.
63	63 d1 (Upper byte of the number of bytes sent) d2 (Lower byte of the number of bytes sent)	01 (ACK) 61 (ACK) 61 (ACK)	Advances the tape (in PLAY mode) for the specified number of bytes. The bit judgement threshold value is taken as the length of one bit and 9 bits are counted as one byte. This command does not perform data read.
64	64 d1 (Block write start mode) d2 ((Block write end mode) d3 (Upper byte of block length) d4 (Lower byte of block length) W1 (Data) ⟆ Wm (Data)	01 (ACK) 61 (ACK) 61 (ACK) 61 (ACK) 61 (ACK) 61 (ACK) (6F (NAK)) 61 (ACK)	Outputs one block to microcassette in EPSON format. Output file and command format and execution result are identical to command 24 (block output to external cassette).
65	65 d1 (Upper byte of number of bytes) d2 (Lower byte of number of bytes)	01 (ACK) 61 (ACK) 61 (ACK)	Outputs the number of bytes of data FF specified by d1 and d2 to the microcassette. Result is the same as command 25.
66	66 d1 (Block read start mode) d2 (Block read end mode) d3 (Upper byte of block length) d4 (Lower byte of block length)	01 (ACK) 61 (ACK) 61 (ACK) 61 (ACK) 61 (ACK) W1	Inputs files from microcassette. Command operation and parameters are identical to command 26.

Command	Master MCU data	Slave MCU response	Description
		W2 ⟆ W84	
67	67 d1 (Block read start mode) d2 (Block read end mode) d3 (Upper byte of block length) d4 (Lower byte of block length)	01 (ACK) 61 (ACK) 61 (ACK) 61 (ACK) 61 (ACK) W2 ⟆ W84	Inputs files from microcassette. Command operation and parameters are identical to command 27.
68	68 d1 (Block read start mode) d2 (Block read end mode) d3 (Upper byte of block length) d4 (Lower byte of block length)	01 (ACK) 61 (ACK) 61 (ACK) 61 (ACK) 61 (ACK) W1 W2 ⟆ W260	Inputs files from microcassette. Command operation and parameters are identical to command 28.
69			Undefined
6A			Undefined
6B			Undefined
6C			Undefined
6D	6D d1 (Upper byte of counter value) d2 (Lower byte of counter value)	01 (ACK) 61 (ACK) 61 (ACK)	Sets microcassette counter value in the slave MCU. The counter value is a 16-bit signed hexadecimal number.
6E	6E	V	Fetches microcassette counter value. Sends the upper 8 bits of counter value to the master MCU.
6F	6F	V	Fetches microcassette counter value. Sends the lower 8 bits of counter value of the master MCU.

Command	Master MCU data	Slave MCU response	Description
70	70	V	Executes microcassette write protect check. In write enable status, '0' is returned to the master MCU. In write protect status, 'FF' is returned to MCU.
71	71 d1 (Upper byte of counter value) d2 (Lower byte of counter value)	01 (ACK) 61 (ACK) 61 (ACK)	Rewinds microcassette tape to the tape counter value specified by d1 and d2. Speed of rewind is same as that of fast forward.
72	72 d1 (Upper byte of counter value) d2 (Lower byte of counter value)	01 (ACK) 61 (ACK) 61 (ACK)	Advances the microcassette tape (fast forward) to the counter value specified by d1 and d2.
73	73	01 (ACK)	Causes the microcassette to rewind up to the beginning of tape (fast rewind).
74	74	V	Inputs microcassette status to the slave MCU. Status is a one-byte code. The significance of each bit is as follows. (Logic '1' indicates an error.) Bit 0: Tape read error Bit 1: Undefined Bit 2: Header or EOF block not found Bit 3: Delay in data transmission from master MCU during data output Bit 4: Write protect Bit 5: Head error Bit 6: Microcassette not connected Bit 7: Undefined
75	75	01 (ACK)	Clears the microcassette status register.
76	76	01 (ACK) (6F (NAK))	Loads the microcassette head. If an error occurs during loading, the slave MCU returns '6F'.
77	77	01 (ACK) (6F (NAK))	Unloads the microcassette head. If an error occurs during unloading, the slave MCU returns '6F'.
78	78	01 (ACK) (6F (NAK))	Rewinds the microcassette tape. Rewind operation continues until the next command is received.
79	79	01 (ACK) (6F (NAK))	Advances the microcassette tape (fast forward). Fast forward continues until the next command is received.

Command	Master MCU data	Slave MCU response	Description
7A	7A	01 (ACK) 6F (NAK))	Advances the microcassette tape (slow forward). Slow forward continues until the next command is received.
7B	7B	01 (ACK) 6F (NAK))	Stops microcassette tape forward and rewind operations.
7C			Undefined
7D			Undefined
7E ~ 7F			Undefined
80	80 d1 (Upper byte of address) d2 (Lower byte of address) d3 (Bit position)	01 (ACK) 0F (NAK)) 01 (ACK) 01 (ACK) 01 (ACK)	Causes master MCU PLG2 port (Address 26, bit 5) value to be stored in the specified bit in the slave MCU. The PLG2 port value is stored in the bit specified by d3 at the slave MCU address specified by d1 and d2. This operation continues until the next command is received. As this is a special command, the mask must be opened prior to execution (command 03). This command will not be accepted if the mask has not been opened.
81	81 d1 (Upper byte of address) d2 (Lower byte of address) d3 (Bit position)	01 (ACK) (0F (NAK)) 01 (ACK) 01 (ACK) 01 (ACK)	Stores the value of the specified bit in the slave MCU to P12 of the master MCU. The slave MCU address is specified by d1 and d2 and the bit position is specified by d3. If any of the data at the position specified by d3 (1) is '1', '1' will be stored in P12. In all other cases, '0' will be stored in P12. Like command 80, this command is a special command.
81 ~ 8F			Undefined
90 ~ FF			Undefined

ERR SEQ LOC OBJECT PROGRAM SLAVE --- SEND SLAVE COMMAND ---

```
00001                          *
00002                                NAM    SLAVE
00003                                TTL    --- SEND SLAVE COMMAND ---
00004                                OPT    LOAD
00005                                OPT    PAGE=55
00006                          *  FILE NAME 'EX36'    BY K.A
00007                          *
00008                          * SEND COMMAND TO SLAVE MCU
00009                          * SEND MELODY PATTERN TO SLAVE MCU AND SEND COMMAND TO PLAY MELODY.
00010                          *
00011A 1000                          ORG    $1000
00012                          *
00013                          *
00014          0011  A         TRCSR  EQU    $11
00015          0012  A         SRDR   EQU    $12
00016          0013  A         STDR   EQU    $13
00017                          *
00018                          * SET SLAVE MCU THE MELODY (YANKEE DOODLE)
00019A 1000 CE 1022  A         PLAY   LDX    #MELTBL    * (X):THE ADDRESS WHERE MELODY DATA ARE STORED.
00020A 1003 86 34    A               LDA A  #$34       * SEND DATA TO SLAVE MCU.
00021A 1005 8D 0E 1015                BSR    SNDSLV     * COMMAND 34: SET MELODY DATA
00022A 1007 A6 00    A         SLV10  LDA A  0,X        * SET DATA
00023A 1009 8D 0A 1015                BSR    SNDSLV
00024A 100B 08                        INX
00025A 100C 81 FF    A               CMP A  #$FF       * LAST DATUM IS $FF.
00026A 100E 26 F7 1007                BNE    SLV10
00027                          *
00028                          * PLAY MELODY.
00029A 1010 86 35    A               LDA A  #$35
00030A 1012 8D 01 1015                BSR    SNDSLV
00031                          *
00032A 1014 39                        RTS
00033                          *
00034                          *
00035                          *
00036                          * SUBROUTINE
00037                          *  SEND COMMAND TO SLAVE MCU
00038                          *  PARAMETER
00039                          *   ON ENTRY
00040                          *    (A): COMMAND
00041                          *   ON EXIT
00042                          *    (A): RECIEVED CODE
00043                          *   REGISTER PRESERVE  (X),(B)
00044                          *
00045A 1015 73 2011  A         SNDSLV TIM    #$20,TRCSR * TX READY ?
00046A 1018 27 FB 1015                BEQ    SNDSLV
00047A 101A 97 13    A               STA A  STDR       ** SEND COMMAND
00048                          * RECIVE FROM SLAVE MCU
00049A 101C 7D 0011  A         SNDS10 TST    TRCSR      * RX READY ?
00050A 101F 2A FB 101C                BPL    SNDS10
00051A 1021 39                        RTS
00052                          *
00053                          *
00054                          * MELODY TABLE (YANKEE DOODLE)
00055A 1022    29    A         MELTBL FCB    41,10,41,10,15,10,16,10
```

11-16.

```
ERR   SEQ   LOC  OBJECT      PROGRAM  SLAVE      --- SEND SLAVE COMMAND ---

          A 1023    0A     A
          A 1024    29     A
          A 1025    0A     A
          A 1026    0F     A
          A 1027    0A     A
          A 1028    10     A
          A 1029    0A     A
     00056A 102A    29     A              FCB    41,10,16,10,15,10,11,10
          A 102B    0A     A
          A 102C    10     A
          A 102D    0A     A
          A 102E    0F     A
          A 102F    0A     A
          A 1030    08     A
          A 1031    0A     A
     00057A 1032    29     A              FCB    41,10,41,10,15,10,16,10
          A 1033    0A     A
          A 1034    29     A
          A 1035    0A     A
          A 1036    0F     A
          A 1037    0A     A
          A 1038    10     A
          A 1039    0A     A
     00058A 103A    29     A              FCB    41,20,13,10,11,10
          A 103B    14     A
          A 103C    0D     A
          A 103D    0A     A
          A 103E    0B     A
          A 103F    0A     A
     00059A 1040    FF     A              FCB    $FF
     00060                        *
     00061          0000   A              END
***** TOTAL ERRORS    0
```