TAND®

ServiceManual

26-3808

S

SOFTWARE MANUAL FOR PORTABLE DISK DRIVE

Catalog Number: 26-3808

10 9 8 7 6 5 4 3 2 1

CUSTOM MANUFACTURED FOR RADIO SHACK, A DIVISION OF TANDY CORPORATION

CONTENTS

INTRODUCTION

Your Portable Disk Drive can be used with any of the Tandy® portable computers: the TRS-80® Model 100, the Tandy 102, or the Tandy 200.

This manual describes the Portable Disk Drive's software, beginning with the format in which the drive stores data on the diskette. It includes information about the three modes of operation available with the drive, and provides details about the commands you can use in these modes.

Use this manual hand-in-hand with the Portable Dist Drive Operation Manual that comes with the drive.

CHAPTER 1/STRUCTURE OF DISK DATA

This chapter describes the structure of data stored with a Portable Disk Drive. First, it looks at the physical division of the diskette. Then, it looks at the logical construction of a file.

DISK SECTORS

At the factory, the Portable Disk Drive diskettes are formatted to have 40 rings, or tracks, numbered 0-39. Each track is divided into two sectors, numbered 0 and 1. The sector is the smallest unit of a track that the disk drive can access (read from or write to) directly. In all, there are 80 sectors on a diskette (two sectors per track × 40 tracks per diskette), as shown in the following illustration:

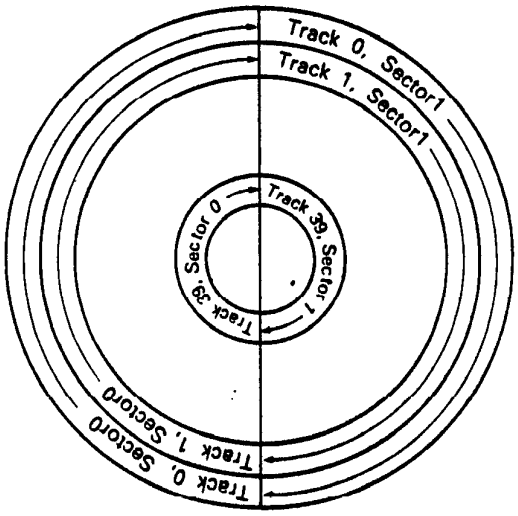


Figure 1-1. Disk Sectors

The sectors on Portable Disk Drive diskettes are called hard sectors because they are permanent. They are established at the factory by the index hole on the spindle of the disk drive.

LOCATING SECTORS

Whenever the computer needs to access data, it consults the disk directory to learn which sector the data is in. For example, the directory might indicate that the data is in Track 4, Sector 1. Given this information, the computer positions the disk drive head to the correct track (in this case, Track 4).

As the track spins beneath the drive head, the computer uses an index signal to recognize the appropriate sector when that sector comes along. For example, if the computer is looking for Sector 0 on the track illustrated in Figure 1-2, the index signal looks for a 7.6 ms timing gap on the spinning diskette. When the drive head reads that gap, the computer knows it has found Sector 1.

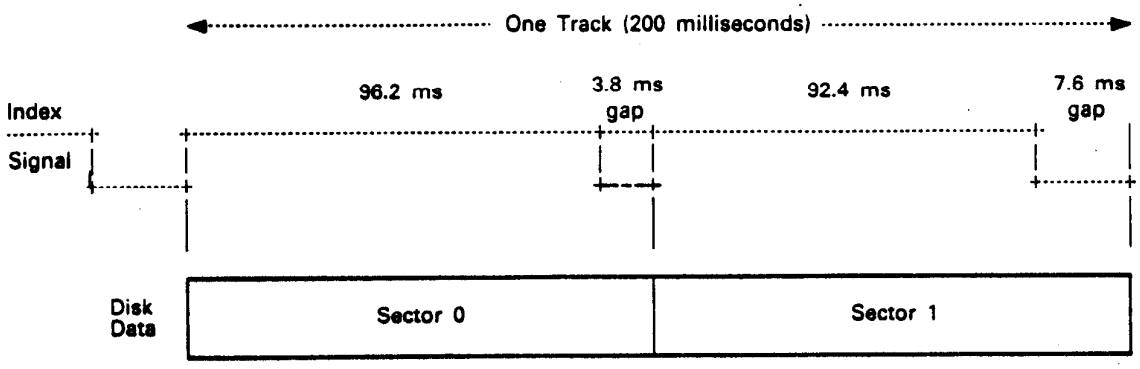
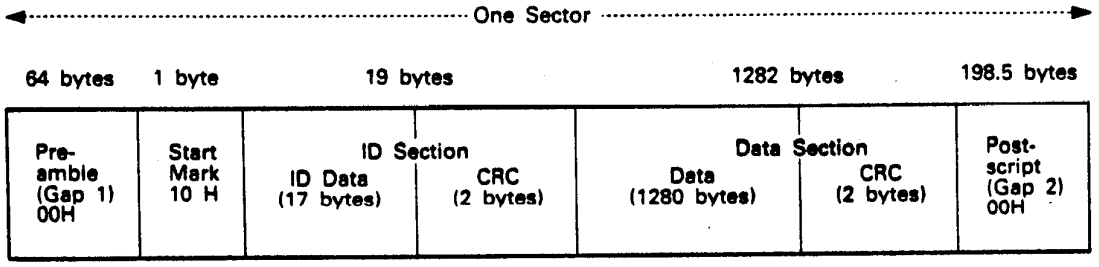


Figure 1-2.
The Relationship Between the Index and the Hard Sector

Note: The segments in Figure 1-2 and in other illustrations in this manual are not drawn to scale. To get a clear understanding, always note the true measurements provided.

PARTS OF A SECTOR

A sector is divided into three parts, devoted to the following information: data, disk ID information, and disk control information. The control information consists of the preamble, start mark, and postscript. See Figure 1-3 for the breakdown.



H = hexadecimal notation
CRC = cyclic redundancy count

Figure 1-3. A Sector

Figure 1-4 illustrates the format of the ID section of a sector:

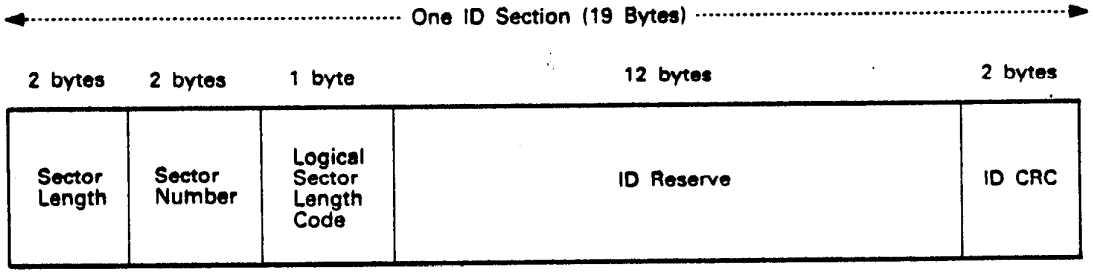


Figure 1-4. The ID Section of a Sector

LOGICAL CONSTRUCTION OF THE DISK AND FILES

The first part of this section analyzes the various parts of the disk directory and how they pertain to a given file. The second part illustrates the structure of the file itself.

THE DIRECTORY

The disk directory is located on Track 0, Sector 0. It consists of 40 file control blocks (FCBs) and one space management table (SMT).

Each file is managed by its own FCB. The free sectors are managed by the SMT.

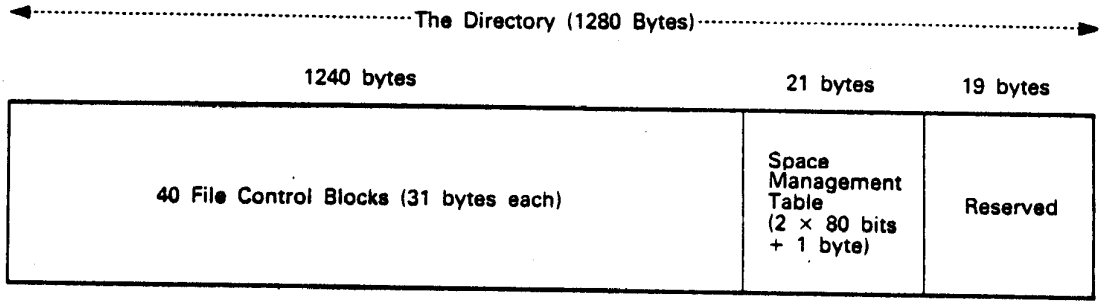


Figure 1-5. The Disk Directory

The File Control Blocks

One File Control Block consists of the following:

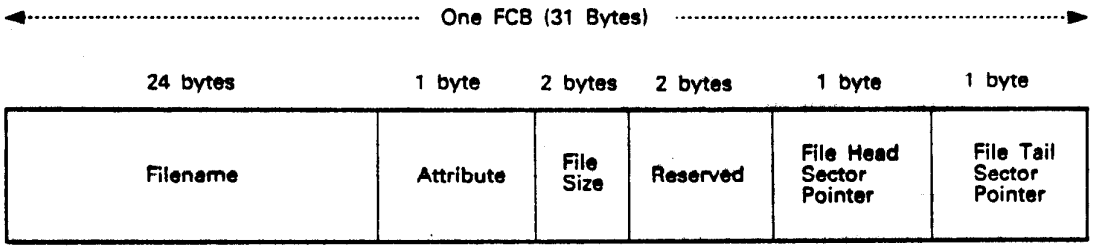


Figure 1-6. An FCB

Filename. The filename is a text line consisting of a maximum of 24 characters. Figure 1-7 illustrates the format of a filename.

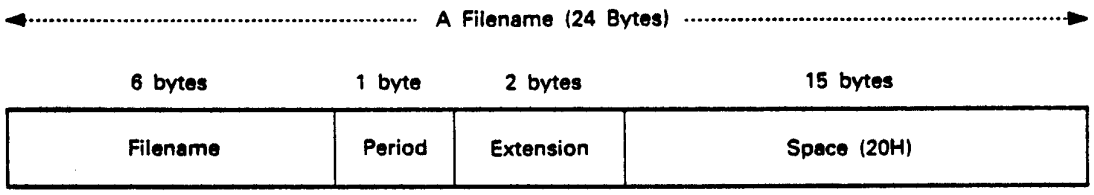


Figure 1-7. Filename Format

SPACE MANAGEMENT TABLE

Normally, when the system searches for a file, the designated file attribute has priority. However when the Portable Disk Drive 2 is used with Model 100, Tandy 102, or Tandy 200, the byte is always filled with the ASCII value of the character F, which instructs the system to ignore the file attribute.

File Size. The file size takes up two bytes of the FCB. For a text file, the size is the length of the file, in bytes, minus the EOF byte. For a BASIC program, the size is the length of the pseudo code of BASIC file.

Sector Pointes for File Head and Tail. These pointes indicate the sectors at which the file begins and ends. Each pointer uses one byte of the FCB. The first seven bits are for the track number. The eighth bit is for the sector number.

The Space Management Table

In the SMT, two bits are assigned to each sector, and are used for file management. The first bit shows the status of the sector: 0 = not used, 1 = used. The second bit is reserved.

In all, 20 bytes are used for file management (2 bits per sector × 80 sectors per disk = 160 bits, or 20 bytes).

The 21st byte of the SMT contains the used-sector counter, which keeps track of the number of used sectors.

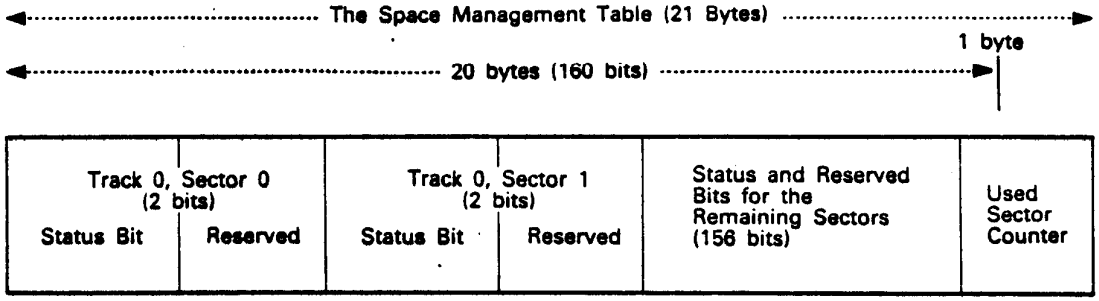


Figure 1-8. The SMT

CONSTRUCTION OF A FILE

Here is a summary of the specifications for sequential files on a diskette in the Portable Disk Drive:

maximum length of file	=	65534 bytes
length of filename	=	24 bytes
maximum number of files	=	40 files
total disk capacity	=	101.12 kilobytes

The construction of a file is as follows:

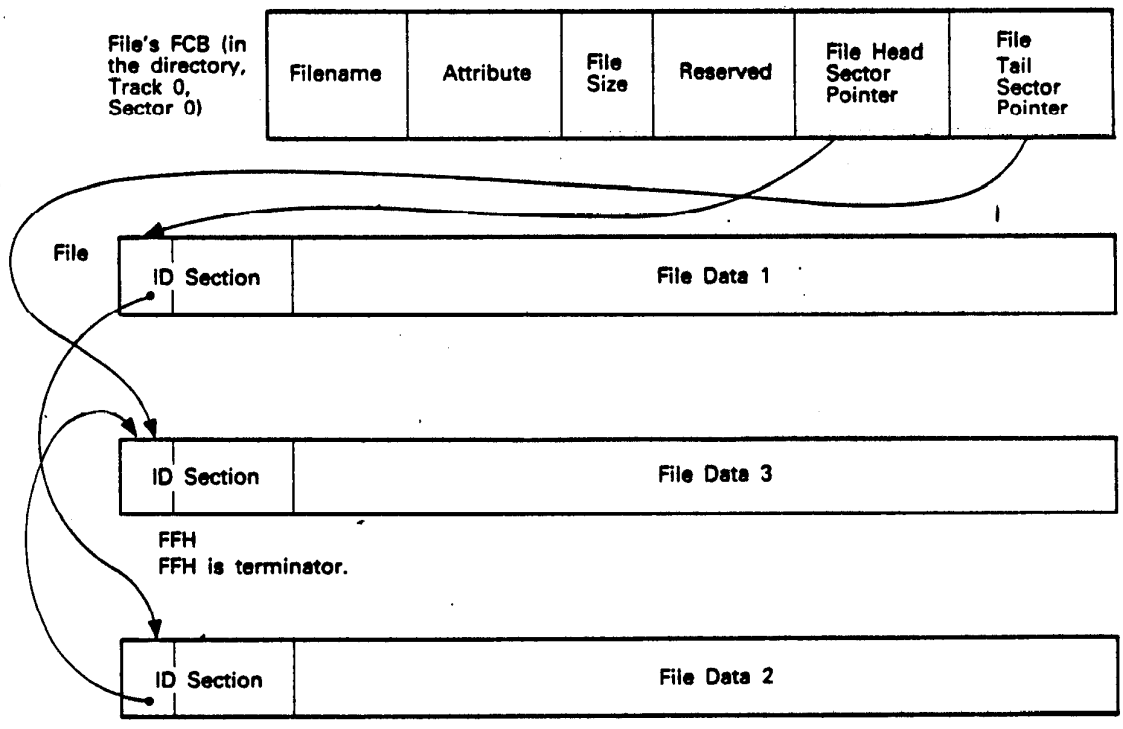


Figure 1-9. Construction of a File

CHAPTER 2/INTERNAL SOFTWARE OPERATION

The key to the software operation of the Portable Disk Drive is a device driver called FLOPPY. CO. This driver is created during the initialization of the disk drive.

A device driver is any program that enables a computer to take advantage of a particular device, such as a printer or disk drive. In particular, FLOPPY. CO is the File Manager for the Portable Disk Drive. It provides the computer with the information that the computer requires for management of the drive's disk files.

MODES OF OPERATION

The Portable Disk Drive uses three modes:

- Initialization mode, which allows the drive to load and initialize the File Manager.
- Operation mode, which the File Manager uses to operate on the data in individual disk files. This mode supports simple DOS. The FILE, LOAD, SAVE, and KILL commands all use this mode.
- FDC-emulation mode, which allows the File Manager to function like a floppy disk controller. Commands that operate on the entire disk or directory--rather than on individual files--use this mode. These commands include BKUP, NAME, and FORMAT.

Note: See the Portable Disk Drive Operation Manual for more information on the commands available with the drive.

SELECTION OF MODES

Before starting up the drive, the various modes are selectable via the four DIP switches on the Portable Disk Drive. Figure 2-1 shows the locations of the switches. The table following the figure shows the settings required for each mode.

The default setting--all switches on--selects the initialization mode at a baud rate of 9600 bps (bits per second). The drive must be in this mode when you install the File Manager.

Once you install the File Manager, the drive must be in either the operation mode or the FDC-emulation mode so that you can use the File Manager. As mentioned earlier, some disk commands use the operation mode, and others use the FDC-emulation mode. You can change the drive's default mode from the initialization mode to the operation mode by changing the DIP switch settings.

Note: Using DIP switches to select modes is effective only when the drive's power is on. While the Drive is operating, its mode may be switched between the FDC and the operation modes depending upon the command used. However, the File Manager automatically changes the mode internally (from the operation mode to the FDC-emulation mode and vice versa), according to the function selected.

The following sections describe how the software operates when the drive is in the various modes.

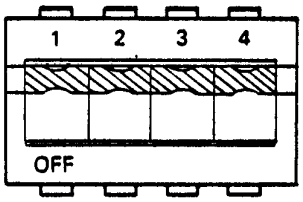


Figure 2-1. DIP Switch Locations

Switch Settings for Mode Selection

Mode	Baud Rate (bps)	Dip Switch Settings 1	2	3	4
init	9600	on	on	on	on *
init	19200 *	off	on	on	on *
init	19200 *	on	off	on	on
FDC	150	off	off	on	on
FDC	300	on	on	off	on
FDC	600	off	on	off	on
FDC	1200	on	off	off	on
FDC	2400	on	off	off	off
FDC	4800	on	on	on	off
FDC	9600	off	on	on	off
FDC	19200	on	off	on	off
FDC	38400	off	off	on	off
FDC	76800	on	on	off	off
operation	4800	off	on	off	off
operation	9600	on	off	off	off
operation	19200	off	off	off	off

First 19200 is for initialization. Second 19200 is for boot sequence (operation mode).

INITIALIZATION MODE

After you connect the Portable Disk Drive, you must initialize the drive before you can use it. To begin, put the drive in the initialization mode by setting all the DIP switches to on. The drive operates at 9600 bps.

The initialization mode has two functions. The first is to allow the drive to accept instructions sent in MOTOROLA® "S" code. This is called the received serial program function. The second is to allow the drive to send initial program loading code to the computer, once it receives your instruction to do so. This is called the IPL function. Combined, these functions enable you to install the File Manager.

FILE MANAGER INSTALLATION

Step-by-step instructions for loading the File Manager are given in the Portable Disk Drive Operation Manual. The following is simply a summary of what goes on "behind the scenes" during the procedure.

1. The drive receives the following "S" format program, which you send as described in the operation manual:

   S10985157C00AD7EF08B3AS901FE

   This program tells the drive to instruct the computer to load and execute a boot program from the Utility Diskette.

Note: For more information on the "S" format, see Appendix A.

2. The computer loads the program from Track 0, Sector 1 into the drive's random access memory.

3. The boot program causes the drive to send the first IPL code in BASIC to the computer.

4. Once you save and execute the IPL code (as IPL.BA), it determines the model of the computer (Model 100, Tandy 102, or Tandy 200). It then instructs the drive to send the second IPL code.

5. The drive sends the second IPL code to the computer, in accordance with the model.

6. The computer loads the second program (sent in Step 5), and passes control to that program.

7. Following instructions from the program, the drive sends the main code of the File Manager to the computer.

8. The second IPL code receives the main code of the File Manager, and creates the file FLOPPY.CO.

Up to this point, the drive works in the initialization mode. However, it is substantially under the control of the boot program stored in the drive's RAM.

OPERATION MODE

To use the File Manager after installing it, you must have the Portable Disk Drive in the operation mode. When all DIP switches are in the off position, and the drive is powered on, the operation mode works at 19200 bps. The operation mode provides all operations except IPL with file management commands.

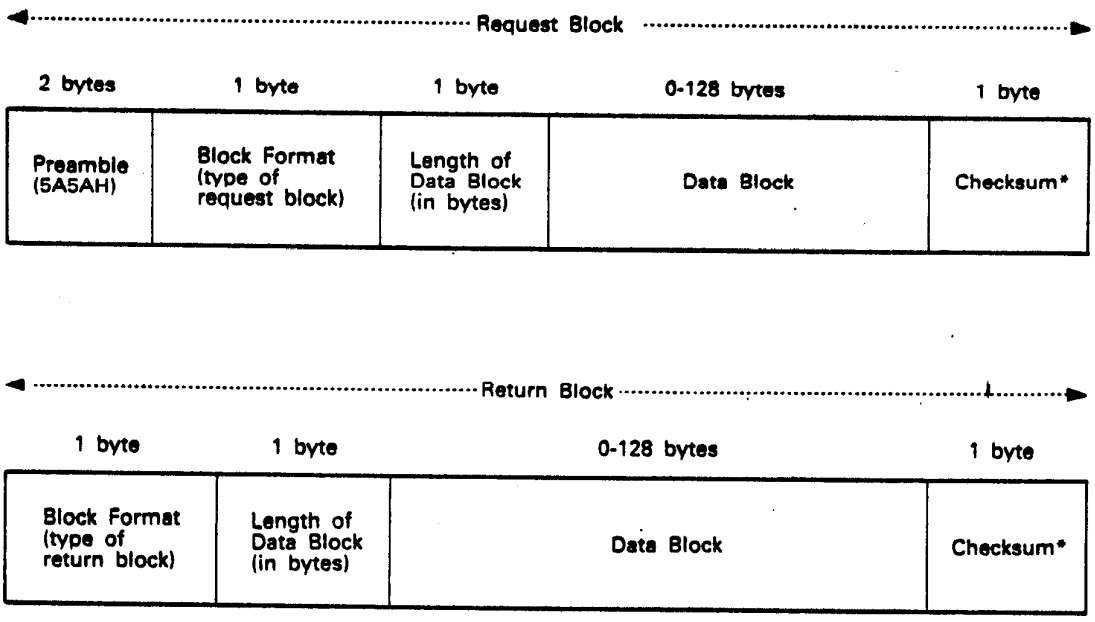
BLOCK COMMUNICATION

The computer controls the drive via block communication through the serial (RS-232) interface. The computer sends a request block of data strings to the drive. The drive sends a return block of data strings to the computer in response.

This exchange of information is termed a half-duplex block transfer because messages are sent in two directions, but in only one direction at a time. Drive control can be performed by either one transfer of blocks or by a combination of transfers. The determining factor is the type of instruction that the computer is giving.

Formats of the Communication Blocks

The formats of the request and return blocks are illustrated in Figure 2-3. See Chapter 3, "Communication Reference," for information on specific requests and responses.



* The checksum is the one's complement of the least significant byte of the number of bytes from the block format through the data block.

Figure 2-3. Formats of the Communication Blocks

FDC-EMULATION MODE

The FDC-emulation mode enables the computer to directly read from and write to a specified sector. Under normal use, the drive enters this mode automatically whenever you issue a command that affects either the directory or the entire disk.

The information about the FDC-emulation mode is in Chapter 4. It is intended for users who choose to enter the FDC-emulation mode manually. It's quite possible, however, that you will never want to do this. If you do use this mode, be careful. When accessing by sectors, you always have the danger of overwriting information in the middle of a file.

To enter the FDC-emulation mode, send block format 08H to the drive while in the operation mode. (See "Change to FDC-Emulation Mode," in Chapter 3.)

CHAPTER 3/
OPERATION MODE COMMAND REFERENCE

This chapter is a reference section to the various block transfers available with the Portable Disk Drive. It lists the communications alphabetically, according to their functions, as shown in this summary:

Function	Request Block Format	Return Block Format
Change to FDC-Emulation Mode	08H	—
Close a File	02H	12H
Create/Access a Directory Reference	00H	11H or 12H
Delete a File	05H	12H
Format a Disk	06H	12H
Get the Drive's Status	07H	12H
Open a File	01H	12H
Read Data from a File	03H	10H or 12H
Write Data to a File	04H	12H

ABOUT THE DESCRIPTIONS IN THIS CHAPTER

Each function's description contains the complete format to use for the request block. If the request requires a message in return, the format of the response block is also shown.

Note: In the event of an error, the drive sends a response block in the 12H format. See Appendix B, "Error Codes."

The following abbreviations are used in the illustrations of the communication blocks:

P = Preamble (always contains the value 5A5AH)
Fmt = Block format
Len = Data block length
C = Checksum (defined in Chapter 2)

The values for the block format and the data block length vary according to the particular communication. In each communication block, they are given in hexadecimal (H) format, in parenthesis. Other items that are specific to a particular block are explained in the text.

ORDER OF BLOCK TRANSFERS

The order in which you perform certain communications is critical. For example, you must create a reference about a file before you can open or delete that file. If you do not, a Sequence Error (30H) occurs. See the sample sequences in the Appendix C, "Flowcharts."

CHANGE TO FDC-EMULATION MODE

Changes the Portable Disk Drive from the operation mode to the FDC-emulation mode.

REQUEST BLOCK

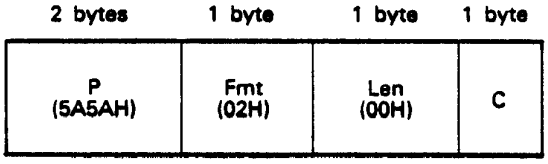
2 bytes	1 byte	1 byte	1 byte
P (5A5AH)	Fmt (08H)	Len (00H)	C

No return block is returned.

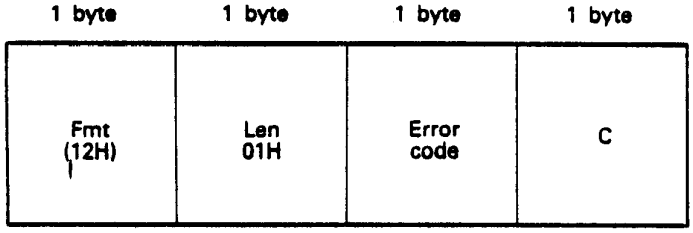
CLOSE A FILE

Closes the open file.

REQUEST BLOCK



RETURN BLOCK



CREATE/ACCESS DIRECTORY REFERENCE

Instructs the drive to create or access a directory reference about a file. You must create a reference before you can open or delete a file.

REQUEST BLOCK

2 bytes	1 byte	1 byte	24 bytes	1 byte	1byte	1 byte
P (5A5AH)	Fmt (00H)	Len (1AH)	Filename	File Attribute	Search Form	C

Request Block Items

filename — Specifies the file for which you want the directory reference. The filename can be a maximum of 24 bytes long. You can use any alphanumeric characters in the filename, as long as you do not use the value 00H in the first byte.

file attribute — Specifies the attribute of the file. The directory reference for the file with the given attribute is outputted before all others. Use any letter for the attribute.

search form — Specifies the form of the directory reference. Use any of the following:

- 00H to create a reference for a file to be opened or deleted.
- 01H to request the first directory information.
- 02H to request information for the next file.

RETURN BLOCK

- When operation is complete.

1 byte	1 byte	24 bytes	1 byte	2 bytes	1byte	1 byte
Fmt (11H)	Len (1CH)	Filename	File Attribute	File Size	Free Sectors	C

Return Block Items

filename — The name of the file specified in the request. If you specified no file, the filename is filled with the value 00H. If you asked for directory information on the next file, and the reference past the end of the directory, the filename is also filled with the value 00H.

file attribute — Specifies the attribute of the file. If the filename is invalid (00H), the attribute also contains 00H.

file size — The size of the file. The preceding byte is significant. If the filename is invalid, the size is 0000H.

free sectors — The number of hard sectors available on the diskette. (One sector contains 1280 bytes.) The correct number returns even if the filename is invalid.

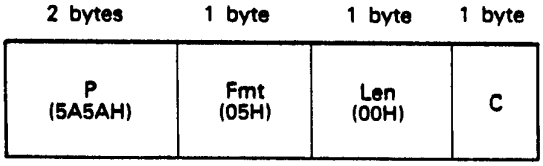
- When operation is not complete, please refer to 12H block on page 22.

DELETE A FILE

Deletes the file specified in a prior Create Directory Reference communication.

Note that the search form used in the Create/Access Directory Reference communication must be 00H. (See "Create/Access Directory Reference.")

REQUEST BLOCK



RETURN BLOCK

Refer to 12H block (page 22)

FORMAT A DISK

Formats a disk and initializes the directory.

REQUEST BLOCK

2 bytes	1 byte	1 byte	1 byte
P (5A5AH)	Fmt (06H)	Len (00H)	C

RETURN BLOCK

Refer to 12H block (page 22)

GET DRIVE STATUS

Confirms that the disk drive is ready for use.

REQUEST BLOCK

2 bytes	1 byte	1 byte	1 byte
P (5A5AH)	Fmt (07H)	Len (00H)	C

RETURN BLOCK

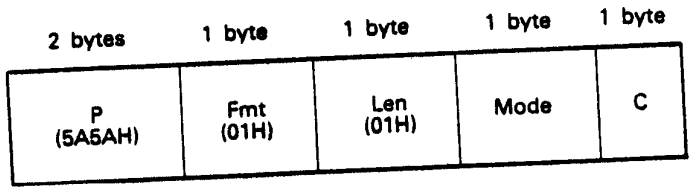
Refer to 12H block (page 22)

OPEN A FILE

Opens the file specified in a prior Create Directory Reference communication. You can open only one file at a time.

Note that the search form used in the Create/Access Directory Reference communication must be 00H. (See "Create/Access Directory Reference.")

REQUEST BLOCK



Request Block Items

mode — Specifies the mode in which you want the file to be opened. Use any of the following:

- 01H to open the file (new) in write mode.
- 02H to open the file (existing) in append/write mode.
- 03H to open the file (existing) in read mode.

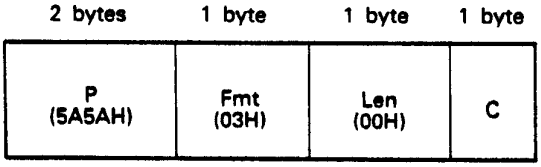
RETURN BLOCK

Refer to 12H block (page 22)

READ FILE DATA

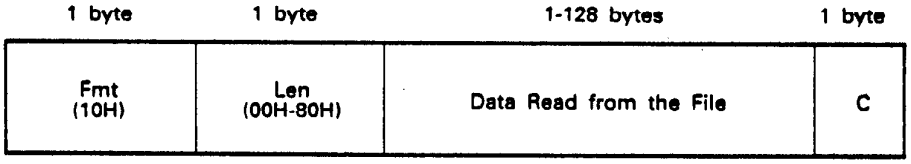
Reads the data in the open file.

REQUEST BLOCK



RETURN BLOCK

- When operation is complete.



Request Block Items

len — Specifies the number of bytes of data read from the file. If the amount of data remaining is more than 128 bytes, the data block length (len) is 128 bytes. If the amount is equal to or less than 128 bytes, len is the same as the amount.

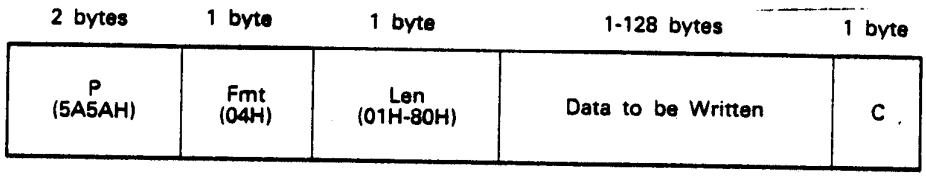
- When operation is not complete.

Refer to 12H block (page 22)

WRITE FILE DATA

Writes data to the open file.

REQUEST BLOCK



Request Block Items

len — Specifies the number of bytes in the data to be written.

In the event of an error, the system returns an error code. Otherwise, no block is returned.

RETURN BLOCK

Refer to 12H block (page 22)

CHAPTER 4/FDC-EMULATION MODE REFERENCE

This chapter is a reference section to the various commands available in the FDC-emulation mode. It lists the commands alphabetically, according to function, as shown in this summary:

Function	Command Letter
Change Modes	M
Check Device Condition	D
Format With/Without Verify	F (with verify) G (without verify)
Read ID Section	A
Read One Logical Sector	R
Search ID Section	S
Write ID Section With/Without Verify	B (with verify) C (without verify)
Write One Logical Sector With/Without Verify	W (with verify) X (without verify)

ABOUT THE DESCRIPTIONS IN THIS CHAPTER

In the FDC-emulation mode, each command is represented by a letter of the alphabet, as shown in the preceding chart. To enter a command, send a string that includes the command letter, followed by any parameters, which you give as decimal codes. End the string with a carriage return. If a command includes multiple parameters, separate the parameters with a comma (,). You can use a space between the command and the first parameter, but the space is not required.

PARAMETER DEFINITIONS

In the command descriptions that follow, the parameters are underlined. The two most common parameters are physical sector number and logical sector number:

- The physical sector number is a number in the range 0 to 79. If you omit this value, the command operates on Physical Sector 0.
- The logical sector number is a number in the range 1 to the number of logical sectors in a physical sector. If you omit this value, the command operates on Logical Sector 1.

All other parameters are defined as they occur in the descriptions.

RESULT CODE

After the drive receives a command in FDC-emulation mode, it transmits 8-byte characters which represent 4 bytes of status code in hexadecimal.

- The first and second bytes contain the error status. A value of '00' indicates that no error occurred.
- The third and fourth bytes usually contain the number of the physical sector where data is kept in the buffer.

  For the D, F, and S commands, the contents of these bytes are different. See the command descriptions in these cases.
- The fifth-eighth bytes usually show the logical sector length of the data kept in the RAM buffer, except the third and fourth digits are 'FF'.

  In the case of an S, C, or M command--or an F command that ends in an error--the bytes contain '0000'.

CHANGE MODES

Switches the drive between the FDC-emulation mode and the operation mode.

M mode

Parameters

mode	Specifies the mode you want. Use 0 (for the FDC-emulation mode) or 1 (for the operation mode).

CHECK DEVICE CONDITION

Sends the condition of the drive, in the third and fourth bytes of the result code.

D

The meanings of the result code bits are as follows:

Bit 7 1 = The diskette is not inserted.
0 = The diskette is inserted.

Bit 6 1 = The diskette has been removed from the drive.
0 = The diskette has not been removed.

Bit 5 1 = The diskette is write-protected.
0 = The diskette is not write-protected.

Bit 4 Always 0.

FORMAT WITH/WITHOUT VERIFY

Formats the diskette, using the specified logical sector length.

G logical sector length code (without verify)
F logical sector length code (with verify)

Parameters

logical sector length code — Can be a number in the range 0 to 6, where:

0 = 64 bytes
1 = 80 bytes
2 = 128 bytes
3 = 256 bytes
4 = 512 bytes
5 = 1024 bytes
6 = 1280 bytes

The default is 3, or 256 bytes.

Notes: The amount of data read or written at one time during FDC-emulation mode is one logical sector. The size of the logical sector might vary from one diskette to another, depending on the format of the diskette.

The third and fourth bytes usually contain the physical sector number, where data is kept in the buffer. But in the error case (when the first and second bytes are not '00'), the third and fourth bytes contain the physical sector number where error occured.

READ ID SECTION

Reads the ID section of the specified physical sector.

A physical sector number

Parameters

physical sector number	Can be a number in the range 0 to 79. if you omit this value, the command reads Physical Sector 0.

The procedure is similar to that for the Read Sector command except you receive ID data instead of Sector data.

READ ONE LOGICAL SECTOR

Reads data from the specified logical sector.

R physical sector number, logical sector number

Parameters

physical sector number	Can be a number in the range 0 to 79. If you omit this value, the command reads Physical Sector 0.
logical sector number	Can be a number in the range 1 to the number of logical sectors in a physical sector. If you omit this value, the command reads Logical Sector 1.

Notes:

If the logical sector length is:	Then the number of logical sectors per physical sector is:
64	20
80	16
128	10
256	5
512	2
1024	1
1280	1

When the drive receives this Read command, the drive reads the data and sends the result code. If the code shows an error, the command is terminated. If it shows no error, the computer must send a carriage return code to receive the read data. If the read data is not necessary, the computer must send a code other than a carriage return.

SEARCH ID SECTION

Searches the ID section of a physical sector, comparing it with the specified string.

S

The procedure is similar to that for the Write Sector command, except that you send a search string instead of data to be written.

WRITE ID SECTION WITH/WITHOUT VERIFY

Writes the ID section of the specified physical sector.

B physical sector number (with verify)
C physical sector number (without verify)

Parameters

physical sector number	Can be a number in the range 0 to 79. If you omit this value, the command reads Physical Sector 0.

Notes: When the drive receives this command, the drive sends the result status. If the code does not show an error, the computer must send the data to be written. The drive sends the status again after it writes the data to the disk.

WRITE ONE LOGICAL SECTOR WITH/WITHOUT VERIFY

Writes data to a logical sector.

W physical sector number, logical sector number
(with verify)
X physical sector number, logical sector number
(without verify)

Parameters

physical sector number	Can be a number in the range 0 to 79. If you omit this value, the command reads Physical Sector 0.
logical sector number	Can be a number in the range 1 to the number of logical sectors in a physical sector. If you omit this value, the command reads Logical Sector 1.

Notes: When the drive receives this command, the drive sends the result status. If the code does not show an error, the computer must send the data to be written. The drive sends the status again after it writes the data to the disk.

APPENDIX A/MOTOROLA "S" FORMAT

The MOTOROLA "S" format is a hexadecimal data form. Figure A-1 illustrates a data record and an a file-end record (S9 record) in this format.

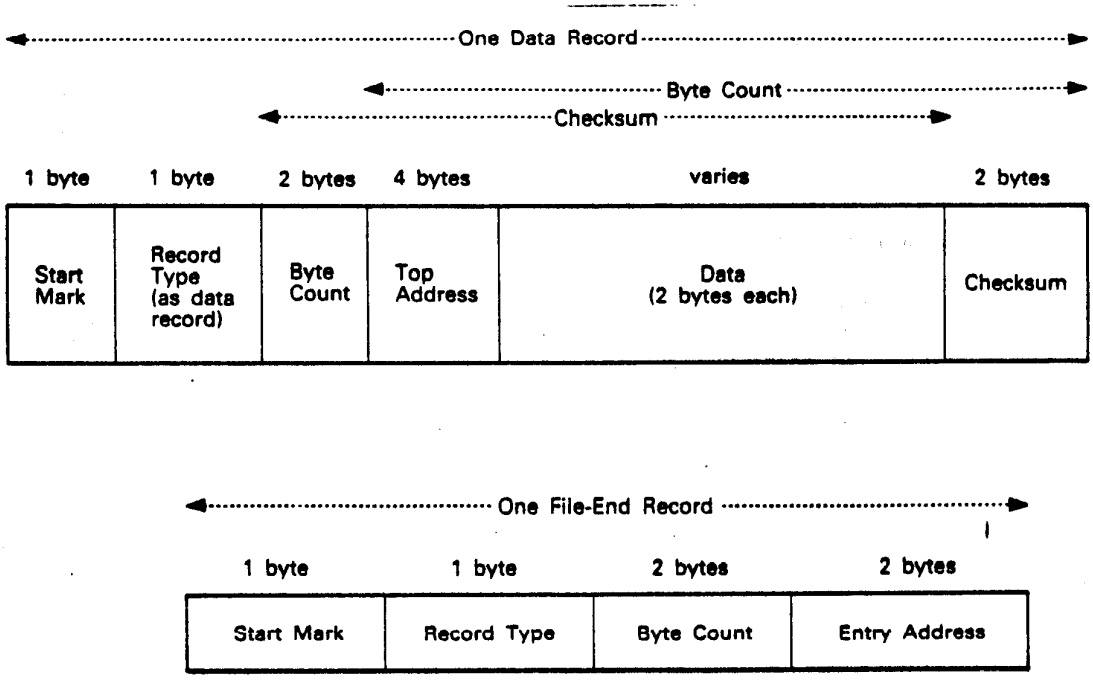
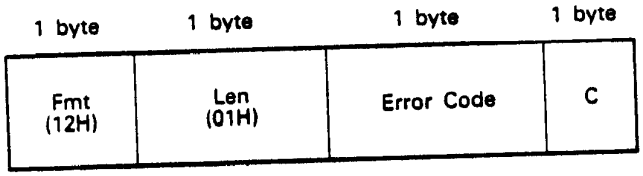


Figure A-1. "S" Format Records

APPENDIX B/ERROR CODES

If an error occurs while the Portable Disk Drive is in operation mode, the drive sends an error code in the following format:



The error codes and their meanings are:

00H	Operation complete
10H	File not found
30H	Command parameter error, sequence error
4XH	Read error
50H	Write-protected diskette
60H	Disk full
7XH	Diskette insertion error
8XH	Hardware fault

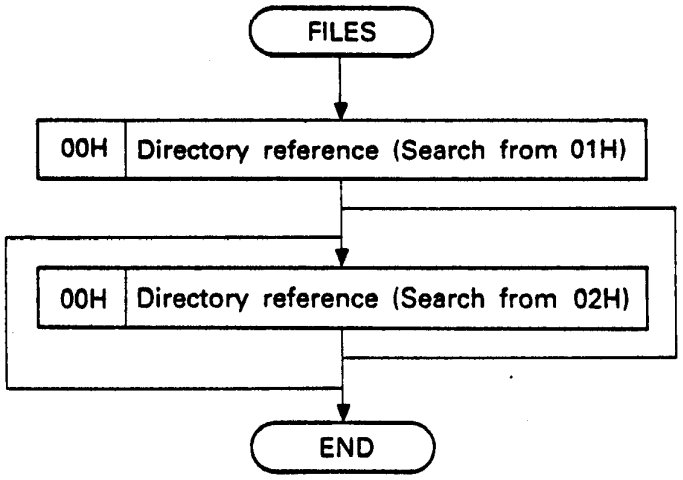
X is a variable hexadecimal value in the range 0 to F.

APPENDIX C/FLOWCHARTS

The following flowchart illustrates some basic file operations you might perform in the operation mode. Use them as a guide to the order in which to send communications to the drive.

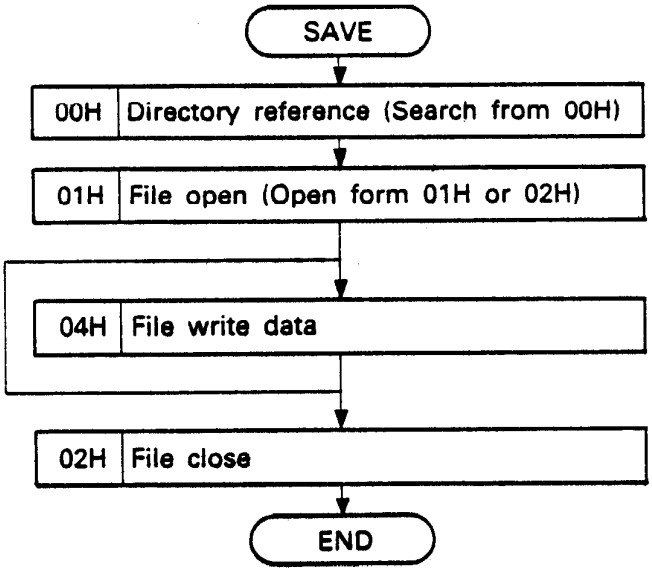
If any error other than a comrnunication error occurs in the middle of a sequence, you need to start the sequence again.

CREATE/ACCESS A DIRECTORY REFERENCE



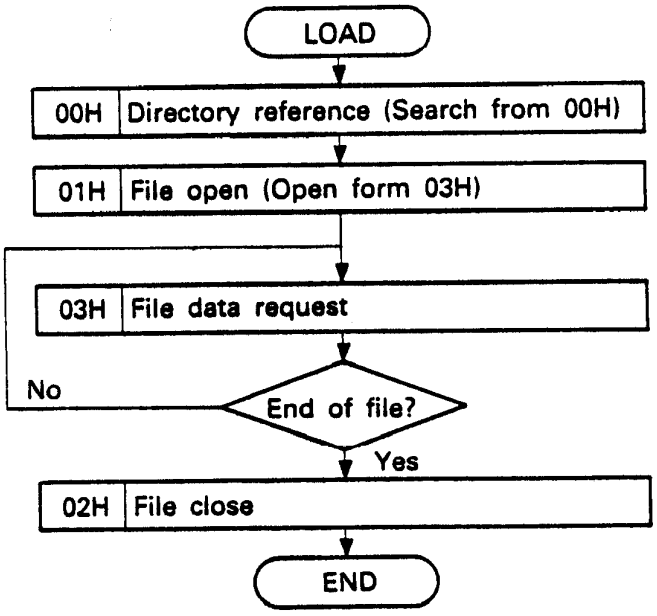
1. Start a directory reference.
2. Refer to the next directory.

WRITE DATA TO A FILE



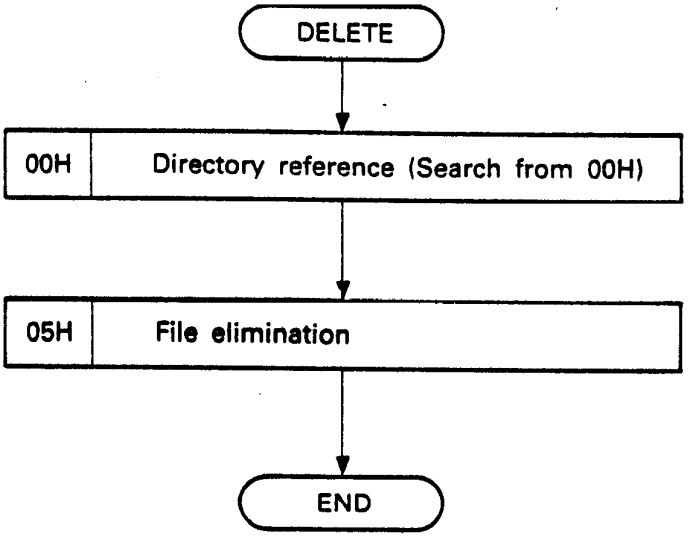
1. Specify the file to open.
2. Open the file in write mode.
3. Send the data to be saved.
4. Close the file.

READ DATA FROM A FILE



1. Specify the file to open.
2. Open the file in read mode.
3. Receive the data to load.
4. Close the file.

DELETE A FILE



1. Specify the file to delete.

2. Delete the file.

RADIO SHACK
A Division of Tandy Corporation
Fort Worth, Texas 76102