INSTRUCTION MANUAL

PETDISK II

6502PDS

CGRS MICROTECH

P.O. Box 102 • LANGHORNE, PA 19047

(215) 757-0284

PETDISK II

LOW COST – HIGH PERFORMANCE FLOPPY DISK

5¼" MINI FLOPPY DISK SYSTEMS:

Model 540-1 Single Drive, Double Density (143K)
Model 540-2 Dual Drive, Double Density (286K)
Model 580-1 Single Drive, Quad Density (286K)
Model 580-2 Dual Drive, Quad Density (572K)

8" MINI FLOPPY DISK SYSTEMS:

Model 877-1 Single Drive, IBM standard (295K)
Model 877-2 Dual Drive, IBM standard (590K)

CGRS MICROTECH

P.O. BOX 102, LANGHORNE, PA 19047 U.S.A.
215-757-0284

PEDISK II is a high performance mass storage peripheral to enhance your computer's storage capability. Total storage to 850K bytes is available. The PEDISK II system consists of a small disk controller electronic board that mounts inside the computer and an external disk drive assembly. PEDISK II offers the fastest mass storage system available for the Commodore PET. With a data transfer rate of 250000K bits per second, the PEDISK II loads data directly to memory. This is up to four times faster than any 488 bus-type mass storage device.

One, two, or three drives connect to the PEDISK controller board. The user can choose 5¼" or 8" disk drives. An important feature of the PEDISK II system is its ability to exchange data with other computers. The standard 8" IBM 3740 format allows data exchange with large computers. Data files, mail lists, etc. available on large computers can be processed by the PET/PEDISK system. Remote PET/PEDISK computers can provide data entry for large machines.

PDOS II software links directly to the standard BASIC and operates with BASIC-type commands for easy interfacing. A full set of disk utility commands completes the powerful disk operating firmware.

SPECIFICATIONS:

MODEL 540	MODEL 580	MODEL 877
Single or Dual drives	Single or Dual drives	Single or Dual drives
143K bytes per 1¼" diskette	286K bytes per 5¼" diskette	295K bytes per 8" diskette
Single-sided	Single-sided	Single-sided
Tracks 40	Tracks 80	Tracks 77
Sectors 28	Sectors 28	Sectors 30
Bytes per sector 128	Bytes per sector 128	Bytes per sector 128
Direct memory interface	Direct memory interface	Direct memory interface
MFM encoding: IBM System 34	MFM encoding: IBM System 34	Standard IBM 3740 encoding
PDOS II Firmware	PDOS II Firmware	PDOS II Firmware

PHYSICAL SIZE, POWER REQUIREMENTS

Model 540, 580	6" Height, 3½" Width (per drive), 12" Depth 110 VAC, 47-63 Hz.
Model 877	9½" Height, 4½" Width (per drive), 19" Depth 110 VAC, 60 Hz. (50 Hz. optional)

FIRMWARE:

The PDOS II firmware package is supplied with PEDISK II. It links to the BASIC language by adding a full repertoire of disk commands.

ROM – 2K bytes
Sequential File Handling
Relative File handling
Program load and save
RAM usage – 2K bytes
Automatic error retry
Full error check logic
Overlay capability
Full disk utilities – format, read, write, patch, load, save, copy, delete, directory, collect, header, scratch.

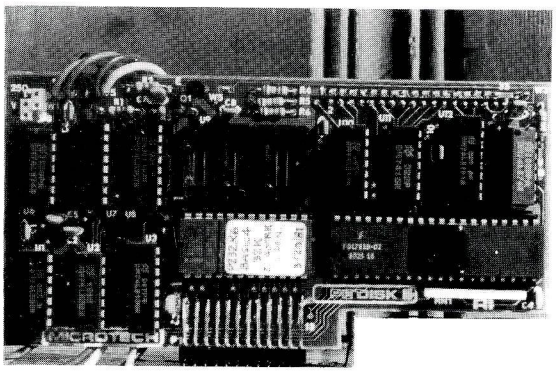
DISK COMMANDS:

BASIC

!LOAD – reads a program file to the computer memory.
!SAVE – stores a BASIC program file to the disk.
!OPEN – forms a sequential or relative data file on the disk.
!INPUT – reads a data record from a file on the disk.
!PRINT – stores a data record to a file on the disk.
!CLOSE – ends a sequential or relative data file on the disk.
!LIST – displays a directory of all files on the disk.
!RUN – reads a program file to the computer memory and executes.

MONITOR-DOS

D – displays contents of memory or diskette.
G – go to program and execute.
H – help user with listing of all commands.
K – kill a file on the diskette (erase file).
L – read program to the computer memory.
M – memory examine and change monitor.
N – name a file differently (rename).
P – print directory of all files on the disk.
R – return to BASIC mode.
S – save program or data from memory to the disk.
U – utility: format, copy, compress, patch diskette.
X – execute program after loading.



ORDERING INFORMATION

☐ ROCKWELL AIM
☐ SYNERTEK SYM
☐ COMMODORE PET
Basic 3.0 ☐ Basic 4.0 ☐
16K ☐ 32K ☐

DEALER

P.O. BOX 102, LANGHORNE, PA 19047 U.S.A.
215-757-0284

ROM SOURCE LISTING

A complete source listing is available for the PEDISK II ROM software. The ROM contains:

1. COMPLETE SYSTEM MEMORY MAP
2. DISK COMMANDS
3. INITIALIZATION ROUTINES
4. MAIN INTERFACE ROUTINES
5. SCREEN I/O UTILITIES
6. DISK PRIMITIVE I/O ROUTINES
7. FILE NAME ROUTINES
8. DISK LOAD ROUTINE
9. MEMORY EXAMINE AND CHANGE ROUTINES

The source listing is available only to registered PEDISK II owners. It is available for the purpose of interfacing user software to the PEDISK II System and understanding PEDISK II System operation. All PEDISK II software is written in assembly language.

The cost of the ROM SOURCE LISTING is $50.00. To obtain your copy, carefully read and sign the attached SOURCE LISTING AGREEMENT. Send the signed agreement along with your check or money order to:
CGRS MICROTECH, INC., P.O. Box 102, Langhorne, Pa. 19047

SOURCE LISTING AGREEMENT

AGREEMENT made and entered into between CGRS MICROTECH, INC. of Langhorne, Pa., hereinafter designated as "Company", party of the first part, a

_____name

_____company

City of_____

State of_____

hereinafter designated as "Customer", party of the second part.
COMPANY agrees to grant Customer use of the source listing of the following program:_____
to be delivered upon signing of the this agreement, subject to the following terms and conditions:

WITNESSETH:

1. This Agreement grants the use of the aforementioned program source listing for his own, personal or single computer installation use. Under no circumstances may the program, in whole or in part, nor any copies as expressed below, be used on other than Customer's own computer installation. In effect, there may be only one copy of the program in use at any given time.

2. No right to print or copy the aforementioned program, either machine readable or otherwise, in whole or in part, is granted to Customer except as hereinafter expressly provided:

a) BACKUP or ARCHIVAL COPIES. Customer may make backup or archival copies of the machine readable program provided they are for Customer's own, individual use.

b) MODIFICATIONS. The source may be entered, in whole or in part, into Customer's system for the purpose of modification or for merging with other programs as desired, provided the resulting software will be used SOLELY by Customer for Customer's own, individual use. Furthermore, any updated software which contains part or all of the aforementioned source listing, whether modified or unmodified, shall remain subject to the terms of this agreement.

3. The original source listing supplied and any machine readable copies thereof, in whole or in part, which are made by the customer, shall remain the property of the Company.

4. This agreement and the program or source listing to which it applies may not be assigned or otherwise transferred by Customer without prior consent from Company.

5. Company makes no warranties, expressed or implied, with regard to performance of the aforementioned program. Furthermore, Company cannot be held responsible nor liable for any consequential damages or loss resulting from operation thereof.

6. This agreement shall be governed by the laws of the State of Pennsylvania.

7. The Customer agrees to be responsible for all reasonable attorney fees and expenses incidental to breaching any of the terms and conditions of this agreement as set forth above.

The parties hereunto set their hands and seal.

_____for Company _____date

_____for Customer_____date

PEDISK II DISK OPERATING SYSTEM

GETTING STARTED!

I. INTRODUCTION.

Congratulations on your purchase of a Microtech PEDISK II System! You have purchased a system that has more processing power than any other disk system available for the Commodore PET (TM), but still manages to contain the easiest to use instruction set.

A complete PEDISK II System consists of:

1) a PEDISK II controller board.
2) a flat cable assembly to connect controller to drive.
3) disk drive assembly with power supply in an enclosure.
4) a cable clamp to secure the flat cable.
5) the PDOS ROM and diskette with your software.

When the PEDISK II controller card is installed and connected to the disk drive you will have the fastest and most versatile disk system available for the PET.

This manual will help you get started using your PEDISK. Detailed instructions on using the Microtech PEDISK II and the PDOS Operating System are given in the PDOS Operating System Users Guide.

II. INSTALLING THE MICROTECH PEDISK.

SETUP (Refer to Fig.1)

The PEDISK II controller card is jumperable to match several different disk system configurations. Although all jumpers are factory installed, a brief check would be in order. Jumpers are setup as follows:

JUMPER W1: PEDISK II was designed to control either 5 1/4" or 8" disk drives. Jumper W1 selects 5 1/4" operation in the upper position and 8" operation in the lower position.

JUMPER W2: PEDISK II can operate at two different data transfer rates- 125K or 250K bits/second. Jumper W2 selects 250K in the upper position and 125K in the lower position. The rate used depends on the system configuration as follows:

1. 5 1/4" single density - 125K (FM)
2. 5 1/4" double density - 250K (MFM)
3. 8" single density - 250K (FM)

The disk size as well as the software configuration will determine the rate used. With PDOS software, the 250K rate should always be selected.

JUMPER W3: This jumper connects J1 (the disk drive connector) pin 16 to common. It should be installed on 5 1/4" systems and left open on 8" systems.

JUMPER W4: This jumper enables the "READY" input on the controller. This signal is available on 8" drives. This jumper should be left open on all 5 1/4" systems and may be optionally installed on all 8" systems.

JUMPER W5: This jumper connects the side select output on double sided systems. It should be installed on 5 1/4" double sided systems and left open on all other systems. Standard PDOS systems are single-sided and W5 should be open.

ROM INSTALLATION (Refer to Fig. 1)

Remove the Commodore ROM #D8 (UD7-8032) from its socket and carefully insert into the PEDISK controller socket U9.
Refer to Fig.2 for location of #D8
Be careful not to bend pins on the ROM when removing or reinstalling. Be careful to orient pin 1 correctly. If the ROM is installed backwards, it will probably be destroyed.
NOTE: For PET 8000 series machines, replace ROM #UD7.

Install the PDOS ROM in the PEDISK controller board, socket U4. Again, be careful to orient pin 1 correctly. Check both ROMs for bent pins and double check for proper orientation of pin 1.

INSTALLATION- 2001 and 4000 series, LARGE KEYBOARD ONLY

1.Verify that all jumpers are properly installed per instructions above. (See SETUP)

2.Verify that the two ROMs are installed properly.

3.Check all I.C.s and verify proper orientation of pin 1. Make sure all parts are seated in their sockets. It is possible for I.C.s to become loose during shipment.

4.Remove the backing strip and affix the white cable clamp to the right inside wall of the PET. The cable clamp is used for stress relief of the disk drive cable so that the PEDISK II controller board will not be pulled from its socket by pulling on the disk drive cable.

5.Make sure the power for the PET is OFF!
Carefully plug the PEDISK II controller board into the socket vacated by the D8 ROM. Make certain that pin 1 on the controller plug P1 is plugged into pin 1 of the socket. When facing the component side of the PEDISK II controller board, pin 1 of P1 is on the left beside capacitor C3. Refer to Fig. 1.
NOTE: ROM D8 occupies memory from $E000-$E7FF (57344-59391).

6. Clip the orange wire to pin 2 of I.C. #G5 (74191) on the PET main logic board. Refer to Fig.3. Make sure the clip is securely in place and that is does not short any p.c. runs. PEDISK II obtains its 4MHZ clock from this point.

7. On PEDISK II units with a Western Digital 1793 controller chip the red wire should clip to the non-band (rear) end of diode CR12 (Test Point 6). Be especially careful about this point. This is the +12 volt power source. Check it with a voltmeter if you are not sure. Be very careful. If the red wire is connected to a higher voltage source, the entire PEDISK II controller board could be destroyed. If you are not absolutely confident see your dealer to have the PEDISK II controller board installed. NOTE:There is no red wire on PEDISK II controller boards with a Synertek 1793 controller chip.

8. Clip the black wire (RESET) to the memory expansion connector J4 pin 22. Refer to Fig.2.

9. Clip the green wire (R/W) to the memory expansion connector J9 pin 22.

10. Clip the yellow wire (Phase 2) to the memory expansion connector J9 pin 21.

11. Plug the flat disk drive cable into J1 of the PEDISK II controller board. Be careful to orient pin 1 of the cable (yellow dot) with pin 1 of J1 (to the front of the board nearest the ROMs)

12. Route the flat cable through the white cable clamp and clamp the cable. Route the flat cable out the side of the Pet where it connects to the disk drive.

13. Plug the card edge connector end of the flat cable onto the back of the disk drive. Again, be sure to line up pin1 of the cable with pin 1 of the disk drive. The disk drive pin numbers are screened on the disk drive p.c. board.

INSTALLATION FOR 8000 SERIES PETS.

Follow the procedures listed above with the following exceptions:

STEP 5: Replace Commodore ROM UD7 instead of D8.

STEP 6: Clip the orange wire to I.C. #UD3 pin ~~9~~ pin 4 74393 (New logic) (74177).

STEP 7: Clip the red wire to the non-band end of CR5 (Test Point 2).

III. SYSTEM CHECK OUT.

When you have your system fully connected, take a minute to double check your system. Are all of your connectors in solid? Are they aligned in the right direction? Are the wire clips from the PEDISK II controller board securely attached and not causing short circuits? Is the PEDISK II controller board plugged securely into the ROM socket? (D8 or Is it facing in the right direction?

The following step by step procedures will insure that your system is functioning properly before attempting to use it.

STEP 1.

Turn on the power to the Disk Drive (but not the PET). At this time the drive motor should be turning but the select light should not be on. If the select light is on, it is an indication that the flat cable is plugged onto the disk drive backwards.
The "select light" is the red light on the front of the disk drive.

STEP 2.

Turn on the power to the PET. At this point, the memory available should be exactly the same as before and the PET should operate exactly as normal.

If this is not the case, turn off the power to both the disk drive and the PET. Recheck the orientation of the Commodore ROM D8 or UD7 in the PEDISK II controller board. Make sure the PEDISK II controller board is firmly seated in the D8 ROM socket.
Check to make sure that none of the pins on the bottom of the PEDISK II board were not bent under the plug.
Check to make sure none of the pins on the Commodore ROM were bent under during insertion. Check to make sure the wire clips are connected to the proper places and that are not causing shorts on the p.c.board runs.

If everything looks correct and your system is not working properly, see your dealer. He may spot something that you are missing.

STEP 3.

Insert the system disk supplied with your system in disk drive #0. Be sure the disk is entered correctly and the door on the drive is shut. The disk is to be inserted with the label to the door. The Master Disk provided with the system is write protected. Verify this. For 5 1/4" disks a tab should cover a slot on the right side of the disk. For 8" disks a notch should be open in the bottom of the disk. If your master disk is not write protected, correct the situation.

STEP 4.

Enter the following command:

SYS59904 (SYS45056 for PEDISK I/EXS100 systems)

The red light on disk drive 0 should light momentarily and the message PEDISK II SYSTEM, CGRS MICROTECH, LANGHORNE, PA. 19047 C1981 should appear in the corner of the screen. After that the "READY" message should reappear and your PEDISK II system is initialized and ready to go. If this does not happen, please refer to the troubleshooting appendix in this manual for assistance in finding the problem.

STEP 5.

Enter the following command:

PRINT FRE(0)

The amount of free memory should now be approximately 2000 bytes less than when the PET was first turned on.

Enter the command: !SYS

Note that all PDOS commands start with an exclamation mark to differentiate them from PET BASIC commands. The screen should now show a greater than (>) sign with a cursor beside it. If not, it indicates that the system disk is not in disk drive 0 correctly.

STEP 6.

The Master Disk should not be used for normal operation of your PEDISK. "Backup" disks should be made for normal operation and the Master Disk should be stored in a safe place. The following procedure will create a duplicate of your master known as a Backup.

a. After the ">" prompt, enter the "U" key.

b. A disk utility will be displayed:

PEDISK II
DISK UTILITY SELECT

1 COMPRESS DISK FILES
2 COPY DISK OR FILE
3 INITIALIZE DISK
4 DISK SECTOR READ & WRITE

Enter the "3" key.

c. Remove the Master Disk from the disk drive and insert an unused disk. All data, if any, previously on the disk will be destroyed.

d. After the DEVICE? prompt, enter a "0" key. (zero, not O)

e. After the SURE? (Y-YES) prompt, enter a "Y" key.

f. The computer will format each track on the disk and display FORMAT TRACK _ . After the last track, PDOS will prompt with "NAME?". Enter a name with eight characters.

Example: BACKUP#1

g. PDOS will record disk name and a blank directory on the disk.
h. When the ")" prompt returns, place the Master Disk back in the drive.
i. Enter a "U" key. When the disk utility menu returns, enter a "2".
j. The disk copy utility will display a "COPY FROM DRIVE #" DEVICE? prompt. Enter a "0 zero" key.
k. The disk copy utility will display a "COPY TO DRIVE #" DEVICE? prompt. Enter a "0" key again.
l. A "PUT ORIGINAL IN DRIVE, HIT R KEY" prompt will be displayed. Enter an "R" key. Your Master Disk is still in disk drive 0 and will be used as the original to be copied from.
m. PDOS copy utility will read several tracks of the Master Disk.
n. It will then prompt "PUT COPY IN DRIVE, HIT R KEY". Remove the Master Disk from the drive and insert the diskette that we initialized in steps above. When the new diskete is inserted, enter a "R" key.
o. PDOS copy utility will record several tracks onto the new disk When finished, PDOS will return to the ")" prompt. Your new "Backup" diskette is finished. Store the Master Disk in a safe place. You may wish to make additional backups now.

Use your Backup diskette for all testing from now on.

STEP 7.

Enter the character "R" (but do not follow it with a RETURN character). The system should now return with the READY message and the blinking cursor.

You have now completed the check-out procedure and are ready to have some fun with your new Microtech PEDISK system.

IV. STARTING YOUR SYSTEM.

The following procedures should be followed in the order shown each time you turn on your system:

STEP 1. Turn on the power to the Disk Drive Assembly.
STEP 2. Turn on the power to your PET.
STEP 3. Insert the system disk in disk drive 0 and close the drive door.
STEP 4. Enter the following command:
SYS59904 (SYS45056 for PEDISK I/EXS100 systems)

Your system is now initialized and ready to go!

V. SAVING AND LOADING PROGRAMS.

STEP 1.
Load your favorite program from a program tape (or type it in) as instructed in your PET manual. If it is a long program, have patience - it is the last time you will have to wait long minutes to run that program! Save the program tape somewhere for emergency back-up; you probably won't ever need it again.

STEP 2.
Pick out a good name for the program that is six or less characters long. For this demonstration, we will use the name GAME.

STEP 3.
Enter the following command:
!SAVE "GAME:0"
The disk light will go on for a second or two and the system will respond with the READY message. Your entire program is now saved on disk!

STEP 4.
Enter the following comand:
NEW
You have now removed the program from memory. Remember how long it took to load it from cassette? Well, watch what happens now!

STEP 5.
Enter the following command:
!LOAD "GAME:0"
Your program will now be re-loaded into memory from the disk in just a second or two! Sorry, you no longer have time to grab a sandwich while your program is being loaded.

STEP 6.
Using steps 1 thru 3, save one or two more programs on your disk. Take turns loading them into memory and running them. Don't spend too much time playing your favorite games, though - there's more coming.

VI. USING PEDISK DATA FILES.

Saving and loading programs on disk saves a lot of time, but it is the ability of the system to save and use data on the disk that gives the PET the power to be much more than just a complex game machine. Business applications, engineering jobs, word processing, and other computer uses all require the power of a high speed, random access disk system.

The cers Microtech PEDISK II system with its PDOS operating system provides the PET user with the unique ability to process data both sequentially (one after the other) and randomly (access any record on the disk file at any time) with easy to use commands. The Users Guide contains complete instructions on using data files; this section will help you get started.

STEP 1.

Enter the following command:

```
!OPEN "TEST:0" NEW LEN 20
```

You have now created a data file on disk drive 0 with the name TEST that can hold up to 20 records. (A record is data that has been combined into a single string variable.)

STEP 2.

Enter the following command:

```
PRINT FC%
```

FC% is a special variable that will contain an error code in case anything went wrong during the execution of the previous disk command. If anything but zero is printed here, look at the list of error codes in the Users Guide for the meaning of the error.

After every disk operation in this section, use this command to look at FC% to be sure that there were no errors.

STEP 3.

You are now going to save a list of ten names on the disk. For each name, enter the following commands:

```
R$ = "name"  (Replace 'name' with an actual name)
!PRINT "TEST:0" R$
```

Don't forget to check FC% after each !PRINT command to be sure no errors occured.

STEP 4.

Enter the following command:

```
!CLOSE "TEST:0" END
```

You have now written an end of file indicator (so the system knows how many records are active on the disk) and taken the file out of ready status. The only command that will now be accepted for this file is another !OPEN command. Up to 4 files may be open (in ready status) at one time.

STEP 5.

Enter the following command:

```
!OPEN "TEST:0"
```

The file has now been re-openned. (Don't forget to check FC% after each disk command)

STEP 6.

Keep repeating the following sequence of commands until FC% contains the value 255.

```
!INPUT "TEST:0" R$
PRINT FC%
PRINT R$
```

You should be getting back the same sequence of names that you entered in step 3. When FC% returned a value of 255, it meant you had retrieved the last record on the file.

STEP 7.

Enter the following command:

```
FI% = 5
```

FI% is a special variable you can use whenever you want to retrieve a record other than the next record in sequence.

STEP 8.

Enter the following command:

```
!INPUT "TEST:0" POS R$
PRINT FC%
PRINT R$
```

Notice that FC% is no longer 255 (end of file), but once again is zero. Notice too that R$ now contains the 5th name that you entered. By using the parameter POS with the !INPUT command, you instructed the system to get a record based on the value in FI% (referred to as its index) rather than the next record in sequence.

STEP 9.

Now repeat the instructions in step 6. Notice that you are now getting the names following record 5 again. This means that you can use FI% to start retrieving records sequentially on a disk at any point in the file as well as use it to retrieve individual records.

STEP 10.

Repeat steps 7 through 9 again, but this time add the following command after each !INPUT command:

```
PRINT FI%
```

You can see that after each !INPUT command (and after each !PRINT command), FI% contains the index (position of the record in the file) of the last record that was read or written.

STEP 11.

Enter the following commands:

```
FI% = 4
R$ = "REPLACE RECORD # 4"
!PRINT "TEST:0" POS R$
```

You have just replaced record 4 with a new value. Do you know how to check that the record was replaced? (Hint - step 7 through 9 with a different initial value of FI%)

STEP 12.

You are now going to update each record by adding a telephone number after each name. Enter the following commands:

```
FI% = 1
!INPUT "TEST:0" POS R$
R$ = R$+"phone-no"  (Replace phone-no with an actual phone number)
!PRINT "TEST:0" POS R$
```

You have now updated the first record on the file.

STEP 13.

Repeat the following commands until FC% returns a value of 255 after the !INPUT command:

```
!INPUT "TEST:0" R$
PRINT FC%
R$ = R$+"phone-no"
!PRINT "TEST:0" POS R$
```

Notice that you didn't have to change the value in FI% for each update. That's because the value in FI% was set by each !INPUT command to the correct value for the !PRINT command.

STEP 14.

Repeat steps 12 and 13 again, but add the following command before the !INPUT command is step 13:

```
FI% = 100
```

You will notice that the new value of FI% did not affect the results. This is because the !INPUT or !PRINT command without the POS parameter always gets the next record in sequence regardless of the value of FI%. This can be very important, espescially when more than one file is open at a time.

STEP 15.

Enter the following command:

```
!CLOSE "TEST:0"
```

STEP 16.

Try to write programs using the disk commands that duplicates the instructions listed above.

If you have followed all of the steps listed above, you now know how to use the PEDISK to load and save programs and save and restore data on the disk drives. There are many other operations that you may want to do with your disk system, including initializing diskettes for use with your PEDISK, copying diskettes for back-up, and many other utilities that can help make the PET a more powerful system. The following chapter will describe the use of the PDOS DOS-mode instructions.

VII. USING THE PDOS OPERATING SYSTEM.

Besides the added BASIC commands supplied with the PEDISK system, PDOS contains a complete disk operating system and monitor. This operating system is much more powerful than the MONITOR program supplied with your PET system. The monitor portion of PDOS provides a more convenient method of performing the functions of the PET MONITOR and also includes many disk oriented commands that can give you more control over your PEDISK disk system.

Now that you have a few programs and a data file saved on the disk, you can use some of the features of the PDOS Operating System. Start by entering the following command: (Be sure that the system disk is in drive 0)

!SYS

The screen will clear and a greater than sign (>) will appear (1771 for PEDISK I) disk controller for the meaning of these codes. at the top of the screen. You are now operating in what is known as PDOS DOS-mode. This is a complete disk operating system that is implemented on many other 6502 based computer systems.

The following sections will describe some of the features of PDOS. See the Users Guide for a complete description of all of the features of the PDOS Operating System.

A. GETTING HELP.

Enter the character H. (Do not follow any PDOS commands with a RETURN key). A menu containing all of the PDOS commands will now be displayed on the screen. The greater than (>) sign will appear for the next PDOS command

B. DISPLAYING THE DISK DIRECTORY.

Enter the character P. The system will respond with the following message:

DEVICE?

Type in the character 0 (zero). A complete directory of all files on the disk will be displayed on the screen. Take particular note of the part of the heading thay contains 'SECTORS REMAINING'. That's the number of disk sectors (in hex) you have left on the disk for additional files.

You will see a number of files on the disk with names that start with asterisks. These are system files used by the operating system and should not be disturbed.

If the message MORE... appears at the bottom of the screen, that means that there are more entries than can fit on the screen. Enter any character to continue the listing, or the STOP key to return for the next PDOS command.

C. INITIALIZING NEW DISKETTES.

New diskettes that have never been used in a PEDISK system must be initialized before they can be used. Initialization consists of formatting (writing disk ID marks) each sector on the disk, and creating a blank table of contents for the PDOS file handling system.

While in PDOS DOS-mode (with the greater than sign (>) at the top of the screen), enter the character U. The following menu will now appear on the screen:

```
      PEDISK II
   DISK UTILITY SELECT

1. COMPRESS DISK FILES
2. COPY DISK OR FILE
3. INITIALIZE DISK
4. DISK SECTOR READ & WRITE

ENTER SELECTION NUMBER -
```

You can now enter the character 3 to select the Initialize disk utility. The following message will appear:

```
      PEDISK II DISK FORMAT

       DEVICE?
```

If you have more than one disk drive, place the disk to be initialized in drive 1 and open the door on drive 0 (for safety's sake). If you only have one drive, replace the system disk in drive 0 with the disk to be initialized. Enter the number of the drive (0 or 1) in which you placed the disk to be initialized. The system will then ask:

```
   SURE? (Y-YES)
```

Respond with a Y key if you are sure. Since a disk initialize destroys all data on the disk, be sure you are right.
As each track is formatted, the track number will be displayed to the screen.

After approximately 45 seconds, the system will display the following message:

```
      NAME?
```

Every diskette can be assigned a name which it will then always keep and display with the directory. Enter any 8 characters as the diskette name. All eight characters must be entered. If your name is less than eight characters, type spaces to fill in the remainder.

D. COPYING DISKETTES.

You can now copy the system diskette onto another diskette for back-up purposes. First enter the U command. The same menu will appear that you used when you initialized the diskettes. This time, you enter a 2 to start the Copy Disk Utility. The following message will then appear:

```
PEDISK II COPY UTILITY
COPY FROM DRIVE#
 DEVICE?
```

Enter a 0 to indicate that you will be copying from disk drive 0. The system will respond with the following message:

```
COPY TO DRIVE#
DEVICE?
```

If you only have one drive in your system, enter a 0. If you have more than one drive, place the diskette on which you are going to copy to into drive 1 and enter a 1.

The system will respond with the following message:

```
PUT ORIGINAL IN DRIVE
ENTER "R" KEY TO CONTINUE
```

Type in the character R. If you are using 2 drives, the entire diskette will be copied without further intervention. If there is only one drive, the system will tell you when to swap diskettes in disk drive 0.

After the copy is complete, the system will return to the PDOS command mode and display the ")" prompt.

The copy utility duplicates an entire diskette. It will copy all files from the original to the copy. Any existing files on the copy diskette prior to the copy will be destroyed.

E. RETURN TO BASIC.

Enter the character R to return to BASIC mode.

There are many more features of the PDOS Operating System that you can use, including the ability to examine and change memory bytes, dump disk or memory, compress diskettes, delete files, and control machine or Assembly language programs. Complete instructions on using these and other features of the PDOS Operating System are contained in the accompanying Users Guide.

A special manual is available for those who wish to modify or write extensions to PDOS. This manual, THE PEDISK II PROGRAMMER'S MANUAL is available from Microtech. It is intended for assembly language programming. Call or write for additional information.

Have fun with your new Microtech PEDISK II System!

APPENDIX A
ERROR CODES AND MESSAGES

There are two different types of error messages that may develope while using PEDISK II, hard disk errors and logical errors. Hard disk errors are generally caused by a bad diskette, bad installation, or by improper disk specification while in the PDOS DOS-mode. Logical errors are errors caused by BASIC disk handling commands. Persistant hard errors may be caused by hardware malfunction although unlikely.

1. HARD DISK ERRORS.

Hard disk errors cause an interuption of whatever process is running, and the printing of an error message. PDOS will attempt to return to the mode from which it was called after an error. However, sometimes the control is returned to the BASIC mode if an error occurs in the DOS-mode.
The format of the hard disk error message is as follows:

** DISK ERROR cc mm dd tt ss st cm cn

cc = Error code (See table below)
mm = Status mask
dd = Disk device
tt = Track
ss = Sector
st = Status
cm = Disk command
cn = Remaining sector count

Error codes mm, st, and cm are normally used only by service personnel. See the technical description of the Western Digital 1793 (1771 for PEDISK I) disk controller for the meaning of these codes.

The following table lists the possible errror codes (cc):

CODE	MEANING
01	Unknown Basic disk command. Occurs if an exclamation mark occurs before a BASIC keyword that is not a valid PDOS BASIC command.
02	Invalid track number requested. This could be the result of a bad track number on disk or bad RAM.
03	File name is not a vaild string - This is usually caused by lack of quotes.
04	File name greater than 6 characters
05	Attempt to save a file with a duplicate name, use a different name and try again.
06	No room on disk for file. Compress the disk or insert a new disk.
08	CRC error. (Data lost on disk)

10	Initialization error. If code cm = 16, then the error occured during a seek, otherwise the sector ID could not be found. This message will occur if a disk is is not properly initialized. This message also occurs when your PEDISK II card is not properly installed, check the green and yellow wires. Also check the cable between the disk and the controller card.
11	Invalid track number specified.
13	Disk drive NOT READY. Usually caused by disk drive door open or disk inserted upside down. This error can only occur on 8" drives. If error occurs on 5", PDOS ROM or system RAM is probably bad.
14	Invalid drive number requested.
15	Invalid track number requested.
16	Busy timeout error at end of disk command.
17	Busy timeout error at beginning of disk command. Usually caused by bad disk controller chip. Power down the computer and restart the entire system.
40	Hard read error-data lost. Usually caused by bad disk.
50	Hard write error-data lost. Usually caused by bad disk.

Basic logical errors do not cause a suspension of operations. Instead, an error code is inserted in variable FC%. Therefore, the variable FC% should be checked after every disk operation.

The following table lists the values of codes that may occur in variable FC% after a disk operation.

CODE	MEANING
7	File is not open.
8	Value in FI% in POS option command is zero or greater than the number of records in a file.
9	File or record name not a string variable.
10	Record string is greater than 127 characters. If this message occurs during a !INPUT command with a POS option, no record was written in that position.
48	File is already open.
49	4 files are already open.
50	!OPEN file not found.
51	!OPEN NEW file is a duplicate name.
52	!OPEN NEW LEN variable is not integer
53	!OPEN NEW LEN variable is not numeric
255	End of file.

APPENDIX B
MEMORY MAP OF PEDISK II SYSTEM

00034-00039 Temporary Zero Page Registers
00084-00103 Temporary Zero Page Registers
00183-00184 Disk Address transfer registers-"BLCKAD"
59648 ($E900) Disk drive select latch
59776 ($E980) Disk controller command/status register
59777 ($E981) Disk controller track register
59778 ($E982) Disk controller sector register
59779 ($E983) Disk controller data register

Note: the select latch and the disk controller are not fully decoded

59904-61439 PDOS ROM addresses

In addition to the above, PDOS takes 2K of memory from the top of system RAM for use as transient storage. This area is used by several overlay programs loaded from the disk. It is also used for critical disk storage.

DISK STORAGE:

WORKAREA = HIMEM-256 HIMEM is the top of PET RAM.
DRIVE# = WORKAREA+145
TRACK# = WORKAREA+146
SECTOR# = WORKAREA+147
SECTOR COUNT = WORKAREA+150

WORKAREA+151 thru WORKAREA+159 are temporary storage for disk transients.

PDOS start = HIMEM-2048
TRANSIENTS = HIMEM-1024
INDEX = WORKAREA

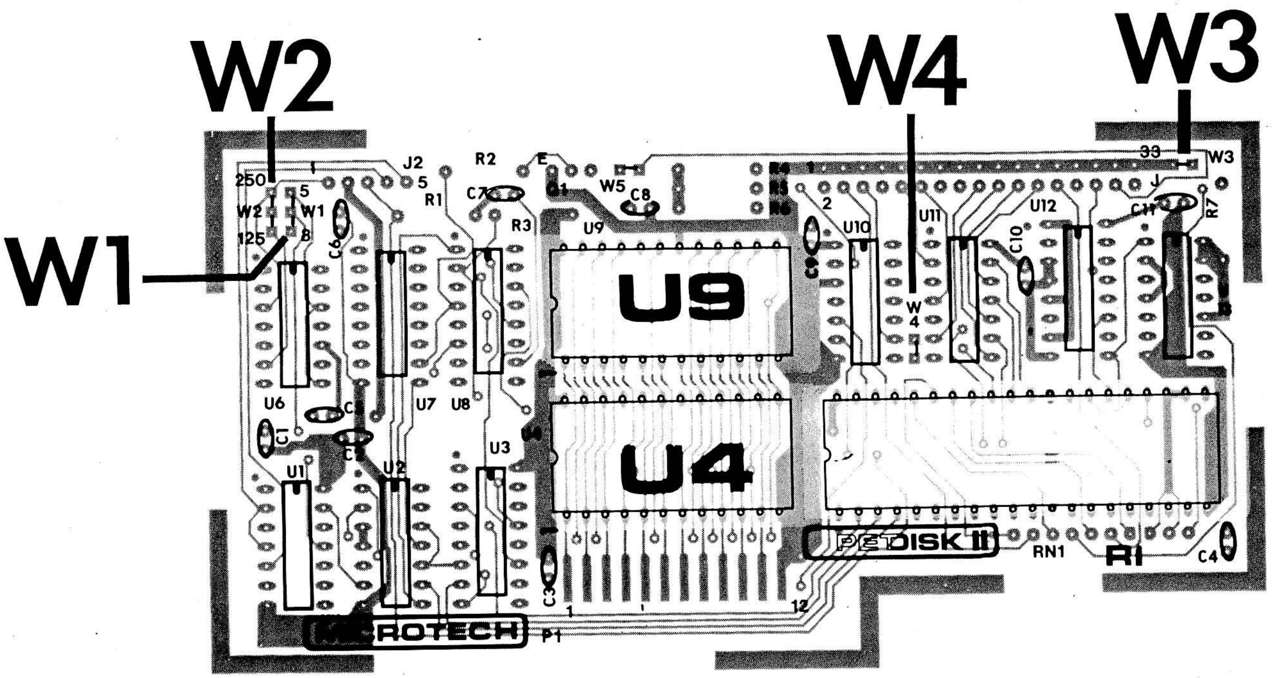

W2
W1
W4
W3
U9
U4
PETDISK II
ROTECH
RN1
R1
C4
P1
U1
U2
U3
U6
U7
U8
U10
U11
U12
W5
C8
C7
R2
R3
J2
250
125
33

FIG.1

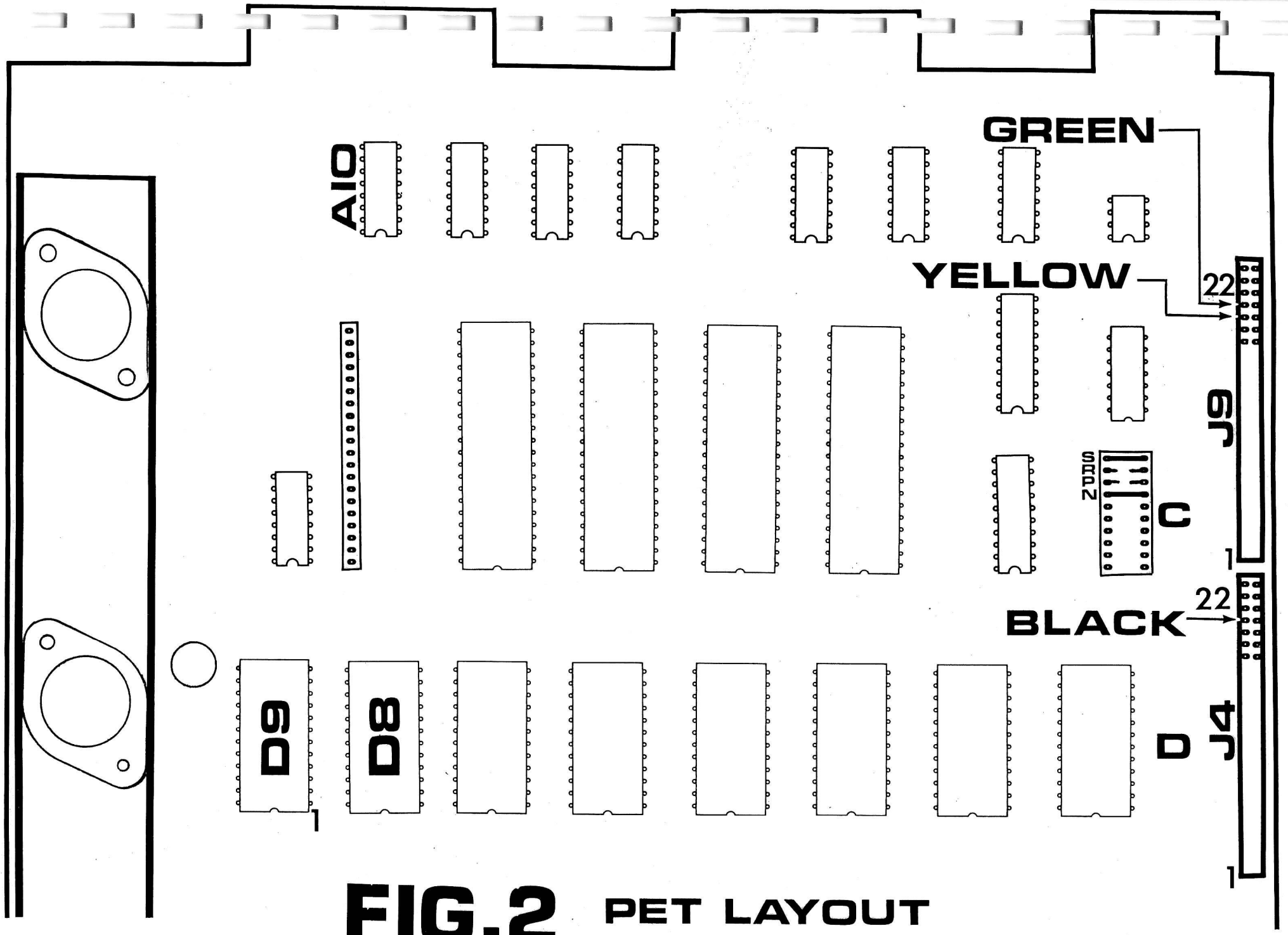


FIG.2 PET LAYOUT

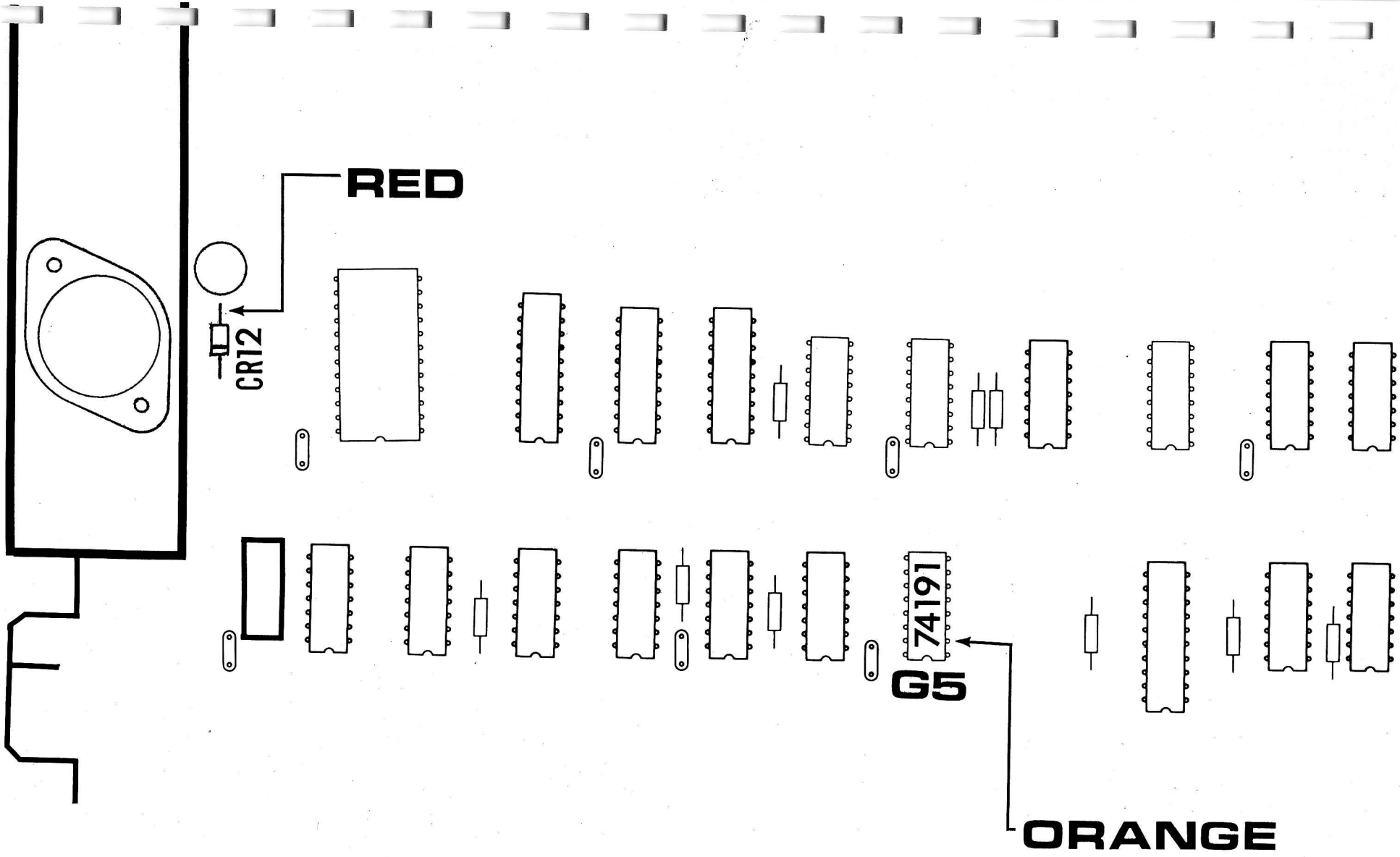


FIG. 3 LOGIC BOARD

PDOS II DISK OPERATING SYSTEM
USERS MANUAL

I. INTRODUCTION

PDOS II is a comprehensive disk operating system for the MICROTECH PEDISK II Disk Controller for the Commodore PET. The system was designed to be easy to use, but provide powerful file handling capabilities. PDOS also greatly facilitates the use of programming languages other than BASIC in the Commodore PET, including a conditional disk Editor/Assembler and fullFORTH+ available for csrs Microtech, Inc.

This manual contains complete details of all BASIC mode and PDOS mode commands. The manual Getting Started that accompanies this manual contains step by step instructions on using most of the features of the PEDISK II system. The contents of this manual will be better understood if you go through the exercises in the Getting Started manual before reading this manual.

The use of PDOS II does not interfere with any feature of the PET. All PET commands and keywords function exactly as they do in a PET without PDOS. Disk functions are provided by adding new commands to PET BASIC for disk program and data file control. Additional disk control is provided by an interface to the PDOS 6502 Operating System in use in other 6502 based microprocessor systems.

PDOS operates in two modes. In BASIC mode, the system functions exactly as under the standard PET operating sytem with additional commands added to provide disk file processing. In PDOS mode, the system provides commands for processing machine language programs and additional disk utilities.

Many of the PDOS commands are processed by transient routines contained on disk. This allows the more experienced programmer to modify these routines for special purposes. Also, the user can program his own routines and add them to the PDOS command set.

II. SYSTEM INITIALIZATION

PDOS is initialized by issuing the following command:

SYS59904

The system will then change the PET end of memory pointer to reserve 2K of memory for use by PDOS, and load components of the operating system from the system disk in disk drive 0. The system disk must be in disk drive 0 if transfer is made to PDOS mode, but may be replaced by data disks during BASIC program runs.

PDOS routines make use of the upper 2K bytes of memory. Therefore, no user program can access this area of memory either by use of a POKE instruction or by loading a machine language program that uses this area.

III. PDOS DISK DEFINITIONS.

The following definition of terms will help the user to understand the discussion of PDOS disk handling procedures that follow.

SYSTEM DISK.
The system disk is a disk initialized for use in PDOS that contains all the disk routines required for operation of PDOS. A copy of a system disk is supplied with the PDOS system. This disk should be copied and saved in a safe place for back-up. The system disk uses less than 5 tracks for the system utilities and can be used to hold users programs and files. If a system uses only 1 disk drive, it is good practise for all disks to be copies of the system disk.

FILE.
A file is a collection of data contained on the disk related to a specific purpose. For example, a file may contain the personnel data for all employees in a payroll system, or all the names and addresses for a label system, or a file may contain a program to be executed.

LOAD FILE.
A load file is a special purpose file that contains a machine or Assembly language program to be executed. The load file contains the program exactly byte for byte as it will exist in memory when executed. Load files are sometimes referred to as core-image files.

SOURCE FILE.
A source file contains BASIC source language files. They are contained on the disk in a compacted format to reduce disk storage requirements. Source files can be loaded and saved by BASIC commands described in chapter IV.

FILE NAME.

Every file is assigned a file name which may be 1 to 6 characters in length. The file name may consist of any combination of ASCII characters except a colon(:). File names starting with an asterisk (*) are not recommended as all system files start with that character.

When file names are referred to in BASIC mode, the file name is a literal and must be enclosed within quotes. Any place in BASIC mode that a file name is required, the file name as a literal contained within quotes or a string variable containing the file name may be used.

DEVICE NUMBER.

PDOS will process in a system with one or two or three! disk drives. They are numbered 0 and 1. Disk drive 0 must always be present and contain the system diskette. Disk drive 1 is optional and can be loaded with any disk initialized by the PDOS disk initialization utility routine.

FILE DESIGNATION.

The file designation is used to refer to a specific file on a specific disk drive. The file designation consists of the file name, a colon(:), and the device number. The following examples show valid file designations:

```
"BJACK:0"
"PAYROL:2"
"A/P:2"
"#PARTS:1"
```

RECORD.

All files consist of a set of individual records. Records can be any length from 1 to 127 bytes long and each record in a file may contain a different number of bytes. The length of each record is determined by the length of the string variable used to create the record.Records are used to hold individual data items in a file. For example, in a payroll file a record may hold the data for a single employee, or in a label file a record may hold a single name and address.

TRACK.

The disk consists of a fixed number of divisions called tracks. The number of tracks on a disk is usually 40 or 80 in a 5 1/4" diskette and 77 in a 8" disk.

SECTOR.

Each track on a disk is further sub-divided into sectors. In the PDOS Operating System, each sector is fixed at 128 bytes in length. When using the BASIC file handling routines, processing at the track and sector level is invisible to the programmer.

SEQUENTIAL FILES.

Sequential files contain records that are placed in the file, one after another, in the order in which they were written to the file. The records are read back in the same order in which they were written.

INDEXED FILES.

Indexed files contain records that are placed on the disk in a position dictated by an index supplied by the program. For example, a program may specify that a record to be written is to be written on the 56th position on the file, then the next record in the 41st position, etc. The records can be read back in any sequence, depending on the index supplied by the program. Therefore an indexed file can be read back in a different sequence than when it was written.

Files may be created as sequential files and then be read back as indexed records.

VTOC (Volume Table Of Contents).

The VTOC is an index containing the names, locations, and other data of all files on a disk. The format of the PDOS VTOC os shown in Table 1. A blank VTOC (no files) is created on each disk initialized by the PDOS disk initialization utility routine.

TABLE 1
VTOC DESCRIPTION

DISK DATA (BYTES 1-16)	
1 - 8	DISK NAME (8 CHARACTERS, SET WHEN INITIALIZED)
9	NUMBER OF ACTIVE FILES
10	NEXT OPEN TRACK
11	NEXT OPEN SECTOR
12 - 16	RESERVED
FILE INDEX (REMAINDER OF 8 SECTORS)	
63 ENTRIES OF 16 BYTES FOR EACH FILE	
1 - 6	FILE NAME
7 - 8	RECORD LENGTH(DATA) OR ENTRY ADDRESS(LOAD FILES)
9 - 10	BLOCK SIZE(DATA) OR LOAD ADDRESS(LOAD FILES)
11	FILE TYPE (SEE TABLE 2)
12	RESERVED
13	STARTING TRACK OF FILE
14	STARTING SECTOR OF FILE
15 - 16	TOTAL NUMBER OF SECTORS IN FILE

FILE TYPE.

PDOS recognizes 8 types of files. The file type is coded by number and is in byte 11 of the file description entry in the VTOC. Table 2 shows the 8 file types.

TABLE 2
LIOCS FILE TYPES

CODE	DESCRIPTION
0	SEQUENTIAL FILE
1	INDEXED FILE
2	INDEXED SEQUENTIAL (FOR FUTURE USE)
3	BASIC TEXT (PROGRAM) FILE
4	ASSEMBLY TEXT FILE
5	PROGRAM LOAD FILE
6	MISC. TEXT FILE
7	PROGRAM OBJECT FILE (SEE ASSEMBLER MANUAL)

OPEN.

Before any file can be processed by a program, it must first be openned. For existing files, the open routine searches the VTOC for the file, moves data from the VTOC to memory, and performs other housekeeping functions required for reading records.

For new files, the open searches the VTOC to insure that the file does not contain a duplicate name of an existing file, creates a new entry in the VTOC for the file, and reserves space on the disk for the file.

PDOS permits 4 data files to be opened at the same time.

CLOSE.

When a program is finished processing a disk file, it must be closed. Closing an input file restores the file to the unopen state and unloads the head on the disk drive (unless another file on the same drive is still open). All open files must eventually be closed.

Closing a new file performs the same functions as for pre-existing files, but in addition may write an end of file indicator on the disk.

All open files are automatically closed when the power is turned off from the PET or the system is re-initialized with the SYS59904 command.

IV. BASIC MODE PROCESSING.

A. DESCRIPTION.

BASIC mode processing consists of the entire PET instruction set plus additional commands for disk file handling. All commands added to the PET command list start with an exclamation mark to differentiate them from PET BASIC commands. Other than the exclamation mark, the syntax for the additional commands is the same as PET BASIC commands.

The following is a list of commands added to PET BASIC. Parameters may be literals or variables. Optional parameters are enclosed in brackets. Keywords are shown in capital letters, string parameters are shown in quotes, and numeric parameters are shown in lower case.

1. Program Load Command.

   !LOAD "file-desig"

2. Program Save Command.

   !SAVE "file-desig"

3. File Open Command.

   !OPEN "file-desig" [NEW LEN rec-count]

4. Record Input Command.

   !INPUT "file-desig" "rec-variable"

5. Record Output Command.

   !PRINT "file-desig" "rec-variable"

6. File Close Command.

   !CLOSE "file-desig" [END]

7. Enter PDOS Mode Command.

   !SYS

8. Program Run Command.

   !RUN "file-desig"

9. List Directory on disk Command.

   !LIST

B. LOADING AND SAVING PROGRAMS.

Programs are saved and loaded from the disk in much the same way as they are loaded and saved from cassette. The file designator as defined in Chapter III identifies the program and disk device that contains the program. If an attempt is made to load a program that does not exist on the specified disk, the system will print six question marks and return to BASIC. An attempt to save a program with the same name as a program already on the specified disk will cause the system to generate disk error 05.

If any other problem develops as a result of the attempt to save or load a program, an error message as described in Appendix A will be printed and control return to BASIC.

The following examples demonstrate saving and loading programs:

```
!SAVE "PAYROL:0"
```

This command will save the program "PAYROL" on disk drive 0. It is important to add the ":" and the disk drive number to each file name.

```
X$="PAYROL:0":!SAVE X$
```

As you can see, a string variable can also be used in PDOS commands. Remember, the ":" and the drive number must always be included as part of the string.

```
!LOAD "BRKOUT:1"
```

The program "BRKOUT" is loaded from disk drive 1. In PDOS; 0, 1, and 2 are valid drive numbers if your system has that many drives. In single drive systems, the only valid drive number is 0.

Loading a program always deletes any pre-existing programs and variables from memory as if a NEW and CLR instruction were executed prior to the load.

The !Load command is only valid as a direct command and cannot be issued from a program. If overlaid programs are desired, the !RUN command should be used.

The !SAVE command is valid form either a program or in the direct mode.

C. PDOS DATA FILES.

The PDOS Operating System provides both sequential and relative files (see Chapter III for a definition of all terms used in this section). All records are variable length. The size of each record is determined by the size of the string variable used to create the record. Records may be updated on the disk and the record length changed.

Four BASIC commands are provided by PDOS for use with the file handling: !OPEN- open a file, !INPUT- read (input) a record from the file, !PRINT- write (print) a record to the file, and !CLOSE- close the file.

!OPEN Command.

To use a data file, it must first be "opened". The !OPEN command does this. New data files must be opened before they are written to and existing files must be opened before they are read or written to. To open an existing file, you must specify only the name. For example: !OPEN "TEST:0" will open an existing file named "TEST" on disk drive 0.
To open a new file, the keyword NEW is added to the !OPEN command. For example: !OPEN "TEST:0" NEW will open a new file named "TEST" on disk drive 0. When a new file is opened, the expected length of the file must be specified. This is done with the keyword LEN. For example: !OPEN "TEST:0" NEW LEN 50 will open a new file named "TEST" on disk drive 0 with a length of 50 records. If the LEN option is not specified, it is assumed to be 100. The length may be specified with an integer variable. For example: L%=50 : !OPEN "TEST:0" NEW LEN L% will open a new file named "TEST" on disk drive 0 with a length of L% records (50).

For existing files, the !OPEN command causes a search of the disk directory. If the file is found, its directory entry is moved into an INDEX buffer and the index is set up for data file manipulation. For a new file, the !OPEN command creates a directory entry. The search is initiated to check for duplicate name. If none is found, the directory entry is written on the disk and an INDEX buffer is set up. PDOS provides space for four INDEX buffers, four files may be open at one time.

Remember that files remain open until closed with the !CLOSE command or by re-initializing the system with the SYS59904 (SYS45056 PEDISK I)

!INPUT Command.

After a data is opened, records of data may be read into the computer with the !INPUT command.

For sequential files, the command is similiar to all PDOS commands. For example: !INPUT "TEST:0" A$ will read the next record from file "TEST" on disk drive 0 and put it into string variable A$. The variable must have been previously defined in the BASIC program. The first time the !INPUT command is used, it will read the first record from the file and so on.

WARNING: The PDOS file handler reads data records one at a time. It sets the string variable pointer to an I/O buffer that contains the record. Any further processing of that file may change the data in the I/O buffer and therefore change the data pointed to by the string variable. The string variable will appear to have mysteriously changed during the program. For example consider the following program:

```
570 !INPUT "FILE:0" R1$
580 GOSUB 9000 :REM CHECK FC% FOR ERROR
590 !INPUT "FILE:0" R2$
```

After execution of statement 570, R1$ will contain a valid record from the file. The execution of statement 590 will cause a new record to be read and the value of R1$ will be overwritten. A simple method of preventing this is to equate the variable to itself. This will cause the BASIC to add the record to its normal string variable stack. Of course this is not required if the record is processed before another record is input. In the above program the addition of the following statement will prevent loss of record R1$:

```
585 R1$ = R1$
```

The PDOS file handling package also includes relative files. Relative files are accomplished by the use of a special variable called FI%. When a file is opened, PDOS establishes a variable FI% for that file and sets it equal to zero. After each !INPUT or !PRINT command a record counter is incremented and put into FI%. FI% will always contain the record count for the file that was just processed. This feature allows the powerful capability of relative files.

A relative file allows a record to be read based on its relative position in the file. Records do not have to be read sequentially. The POS option on an !INPUT command allows this. To read any record based on its relative position, simply set FI% equal to the relative record desired and add the keyword POS to the !INPUT command. For example:

```
520 FI%=2 : !INPUT "FILE:0" POS R1$
```

This statement will read the second record from "FILE" on disk drive 0. An additional feature of PDOS is that FI% will still be incremented when using !INPUT or !PRINT without the POS option. Combination relative-sequential file techniques can be used since sequential records can be read after a relative record without the user having to keep track of the record number.

ERROR CHECKING

PDOS checks for logical errors as well as hard disk errors. Logical errors do not stop processing as do hard disk errors but are transferred to another special string variable called FC%. After every PDOS operation, FC% should be checked to make sure it is zero. Any other value indicates a problem. A complete list of the error codes is contained in Appendix A.

Since disk access requires that the variables FI% and FC% as well as string variables used for records to be defined prior to use, a statement similiar to the following should be in the beginning of every program that uses disk files.

```
10 FI%=0,FC%=0
```

!PRINT Command.

The !PRINT command is used to write records to a data file. It is similar in format to the !INPUT command. Both sequential and relative files can be written.

For example: !PRINT "TEST:0" A$ will write the variable A$ to next record in file "TEST" on disk drive 0. If no records have been written to the file since opened, FI% will equal 1 and A$ will be written to record 1.

Relative records are written with the !PRINT command the same way relative records are read with the !INPUT command.

For example: 540 FI%=4 : !PRINT "FILE:0" POS A$

This statement will write A$ to the fourth record in the file "FILE" on disk drive 0.

This is especially useful in updating records. Suppose, for example, a file is to be read sequentially with each record updated (changed) and re-written on the disk. The records can be read sequentially by using the !INPUT command without the POS option. Since each !INPUT command will return the position of the record in FI%, a !PRINT command with the POS option following the !INPUT command will cause the last record read to be re-written. The next example will demonstrate this.

```
100 F$="DAFILE:0" : REM USE VARIABLE FOR CONVENIENCE
110 !OPEN F$:REM       OPEN EXISTING FILE
120 IF FC% = 0 THEN GOTO 140:REM  TEST FOR ERROR
130 PRINT "***  ERROR CODE # ";FC%:STOP
140 INPUT F$ R$:REM    READ RECORD INTO R$
150 IF FC% = 0 THEN GOTO 180:REM TEST FOR ERROR
160 IF FC% <> 255 THEN GOTO 130:REM TEST END OF FILE
170 PRINT "PROGRAM END OK":STOP
180 R$ = LEFT$(R$,4)+"X"+MID$(R$,6)
190 REM   RECORD IS UPDATED BY REPLACING THE 5TH CHARACTER WITH "X"
200 !PRINT F$ POS R$:REM  RE-WRITE THE LAST RECORD READ
210 GOTO 120:REM TEST ERROR CODE AND PROCESS NEXT RECORD
```

When a new file is to be created, the NEW parameter is used in the open command. The LEN parameter is used to specify the maximum number of records in the file. If the LEN parameter is omitted in a new file, the default value of 100 records is used.

The END option of the !CLOSE command is used to create an end of file indicator after the last record processed. This option is usually used when a new file is created sequentially. It insures that the end of file status (255) will be returned in FC% when reading back the file sequentially.

!CLOSE Command.

Files are closed with the !CLOSE command. It is used to clear INDEX buffer space so that new files can be opened. The format of the !CLOSE command is similiar to other PDOS commands. For example: !CLOSE "TEST:0" will close file "TEST" on disk drive 0. The END option is used to create an end of file indicator after the last record processed. This option is used to insure file status of 255 will be returned in FC% when reading back the file sequentially.

D. !SYS COMMAND.

The !SYS command transfers control of the system to PDOS mode. A system disk must be in drive 0 when the !SYS command is executed. The !SYS command is only valid as a direct command; it cannot be issued from a program.

E. !RUN Command.

The !RUN command allows the PDOS user to load and run programs with a single command. It can be used in the program mode or the immediate mode. It offers the capability to "overlay" programs. A BASIC program could be loaded into memory. The last statement of that program could be !RUN "NEXT:0" and the new program "NEXT" on disk drive 0 would be loaded over the original and executed. This is the manner by which large programs can be run from computers with small memory. Program segments can run, do their particular job, and then call the next segment.

F. !LIST Command.

The PDOS system allows an instant display of all programs contained on a disk with use of the !LIST command. The command has no parameters. For example: !LIST

After the command is interpreted, the PET will ask "DEVICE ?" The user must enter a 0, 1, or 2. No other numbers will be accepted.

The computer will then display all programs including system utilities, BASIC program files, data files, and assembly object files. Each entry has the following format:

```
 NAME TYPE TRACK SECTOR #SECTORS

NAME = File name, 6 characters or less
TYPE = type of file; SEQ-sequential
                     IND-indexed (relative)
                     ISM-indexed sequential
                     BAS-BASIC program file
                     ASM-assembly file
                     LD -loadable object file
                     TXT-text file
                     OBJ-relocatable object file
TRACK =starting track on the disk
SECTOR=starting sector on the disk
#SECTOR=number of sectors in the file
```

V. PDOS MODE.

The BASIC command !SYS is used to enter PDOS mode from BASIC mode. PDOS mode provides the power of the full PDOS Operating System implemented on the AIM (Rockwell), SYM (Synertek), the 6502 Professional Developement System from cers Microtech, Inc. and other 6502 systems to the PET user.
Using the PDOS mode, machine or Assembly language programs may be saved or loaded from the disk, other languages such as the EDT/ASM Assembler may be executed, memory bytes may be easily examined and changed, disk sectors can be directly read or written, and a number of other disk utility functions may be used.

A system disk must be in drive 0 at all times while in the PDOS mode.

PDOS mode may in some cases cause a deletion of BASIC programs in memory prior to entering PDOS mode. It is good practice to save any BASIC program in memory on disk (if they are not already on disk) before entering PDOS mode.

When entering PDOS mode, the screen will clear and a greater-than sign (>) will appear. The greater than sign is used to indicate that the system is ready to accept PDOS commands. All PDOS commands are one character. A complete list of the commands can be displayed by typing in the command "H" for HELP. PDOS commands are processed as soon as the command is entered; the user does not type a RETURN following the command.

Return to BASIC mode from the PDOS mode is accomplished by issuing the command "R".

NOTE: Some error conditions may cause an automatic return to BASIC mode. If this should occur, it is possible that the system may not be completely reset to BASIC mode. Therefore, the user should issue a !SYS command followed by a PDOS "R" command or re-initizatize the system by issuing the SYS59904 initialization command if that should happen.

The following is a list of all commands implemented in PDOS. The command is executed by entering the single character shown. Commands marked with an asterisk are executed by routines contained in the Read Only Memory (ROM) containing the PDOS II System and can be executed with any disk ready. Those commands without an asterisk are executed by transient routines contained on the system disk and will not be accepted if the system disk is not ready in drive 0.

A. (MACROASSEMBLER-EDITOR)

The A command will cause a special interface routine to be loaded. This routine can be used with the MACROASSEMBLER-EDITOR (MAE) program from Eastern House Software.

D. (DUMP)

The D command will cause a formatted dump of either main memory or disk sectors to be printed. The dump will show the hex and alpha contents of memory or disk. The prompts for controlling the dump will be listed after entering the D command.

The dump can be stopped and return made to PDOS command mode at any time by entering the STOP key. After filling the screen, the dump will stop and wait. The dump can be continued by entering any character.

*G. (Go)

The G command will execute a machine or Assembly language program already in memory. The system will prompt the user for the entry address of the program.

H. (Help)

The H command will cause a list of all PDOS commands to be displayed on the control console.

*K (Kill)

The K command is used to delete a file from the disk.

*L (Load)

The L command will load a program or file from disk without executing it. Control will be returned to PDOS command mode after loading.

*M (Memory)

The M command is used to examine and change individual bytes in memory. After the M command is input, the system will ask for an address. A 4 byte hexadecimal address should then be entered. The system will respond by listing 8 bytes at that address and positioning the cursor under the first byte.

The user can then type in any two digit hexadecimal number which will replace the byte in memory and proceed to the next byte. Typing a space will leave the contents of the byte unchanged and proceed to the next byte. Entering a carriage return causes the system to ask for a new address.

After spacing or changing the 8 bytes listed, the next 8 bytes in sequence are listed and processing as described above proceeds. Return to PDOS command mode is accomplished by entering the STOP key at any time.

Memory bytes changed by the user are read back by the system to verify that the byte to be changed was in available RAM and the change was accomplished. If the change was not made for any reason, the system will print a new line beginning with the bad byte. Question marks are printed if the user enters any character but a valid hexadecimal digit.

N (Name)

The N command is used to rename files on a disk. THe system will ask for the name of the file to be changed and the new name. If a file already exists with the new name a message is printed and the user can either delete the existing file with the new name or input a different name for the file.

P (Print VTOC)

THe P command is used to print a directory of all files on a disk. System files containing transient routines all start with an asterisk (*).

*R (Re-enter BASIC)

The R command is used to return to BASIC mode. The system disk must be in drive 0 when issuing the R command.

*S (Save)

The S command is used to save machine or Assembly language programs on disk.

After the S command is given, the system will ask for the address range to be saved on disk, and an entry address. The entry address is the address that the system will branch to after loading the program from an X command.

U (Utility Menu)

The U command is used to enter the PDOS Utility Sub-System. The Utility Sub-System is used to perform a number of house-keeping chores related to the disk drive.

Following the execution of the U command, the Utility Selection menu will be listed as follows:

```
        PEDISK II
    DISK UTILITY SELECT

1. COMPRESS DISK FILES
2. COPY DISK OR FILE
3. INITIALIZE DISK
4. DISK SECTOR READ & WRITE

ENTER SELECTION NUMBER -
```

By entering the appropriate menu selection number, the following functions can be performed:

1. COMPRESS DISK FILES

This Utility is used to compress a disk and free up empty space caused by deleted and truncated files. During the running of the disk compression, the keyboard is disabled to avoid accidental interupt of the system. If the compression run is interupted by a power failure or by execution of the system RESET key, files on the disk may be lost. Therefore, it is a good idea to use the file copy utility to create a back-up of the disk before compressing it.

If any file is moved during a compress, a message will be printed listing the file name. This could be useful if hard error interrupts the compress. The new VTOC is not written until after all files are moved.

If a file is larger than the memory, it will be divided and the message will be printed for each section.

2. COPY DISK

This utility is used to copy entire disks. Complete instructions on the use of this utility will be listed after the utility has been called. A copy (backup) diskette can indeed be copied with only one drive implemented on the system, but the user may have to change the disks a number of times depending on the amount of memory available and the amount of space used on the disk. Protection is built into the system to prevent writing to any disk but the one being copied to.

The disk copy will erase all files that exist on the disk being copied to, and place the files from the original disk in the same position on the new disk. The new disk must contain a valid PDOS VTOC. It will retain the name assigned to it when it was initialized, and will not be re-assigned the name from the original disk.

3. INITIALIZE DISK.

This Utility is used to initialize a new disk for use in the PEDISK II system. It will completely initialize all tracks on the disk and create a blank (no files) VTOC. After initializing the system, the user will be asked to enter the disk name which can be any 8 ASCII characters.

Care should be used with this utility as it will erase any data previously recorded on the disk.

4. DISK SECTOR READ & WRITE

This Utility can be used to read any number of sectors on a disk to any memory address, or to write from memory to any sectors on the disk. This Utility can be used, for example, to load a load file from the disk to an address other than its normal load address.

Since the utility bypasses the normal PDOS file processing system, extreme care must be executed or the disk VTOC or existing disk files may be destroyed.

Following the execution of this Utility, the system will respond by requesting the entry of the function (read or write) to be performed, the disk drive, track number, sector number, the number of sectors, and the memory address.

*X (eXecute)

The X command loads a file from disk to memory and executes it. The system will prompt the user for the name of the file and drive number.

VI. MEMORY TEST

The PDOS II system disk includes a memory test program that can be used to verify your computer's memory. The program is called MTEST:0. To operate, initialize the system with the SYS59904. When the "READY" message reappears, go into the DOS-mode with the !SYS command. The MTEST program may be executed with the X command. Enter the X after the) prompt. The PET will ask for the file name. Enter the name MTEST:0
The program will load and execute displaying a message:

```
PET MEMORY TEST
CGRS MICROTECH,INC.
START ADDRESS?
```

Enter the starting address in hexadecimal. Do not start the program below address $0600.

After the start address is entered, the PET will ask for the end address. " END ADDRESS?"
Enter the ending address as a four digit hexadecimal number. The program will begin testing the range of memory specified. Verify proper operation by observing a changing character in the upper left corner of the screen. The test requires about 1 second for each 4K of memory.

If an error is detected the test will print a string of numbers. They are: Test# 1 or 2, memory location of the failure, data expected, data in memory, test type 0 or 1. A failure would be:

```
01 3580 55 57 01
```

This location failed test 1, memory location $3580, data should have been $55 but was $57 (data bit 2 is bad), test type 1.

If you are having trouble with the disk copy routines or the disk initialize routines you should run the test on the top 4K of your memory.

For example: START ADDRESS? 7000
END ADDRESS? 8000

This will test the top 4K in a 32K PET.

VII. BUILD - GET PROGRAMS

The PDOS system disk contains two programs called "BUILD:0" and "GET:0". These programs build and get a series of data files. They are includes both as test programs and examples of how to use the PDOS data file capabilities.

To operate the "BUILD:0" program, initialize your system using the SYS59904. When the READY reappers, the program is loaded with the !LOAD command. Enter the following:

```
!LOAD "BUILD:0"
```

LIST the program and examine it. Try to follow the program flow.

```
Enter RUN
```

The program will ask for a file name, enter "TEST:0".
Do not forget the quotes, they must also be entered.
The "BUILD:0" program will build a data file of alphabetic characters starting with a different letter.
Use the !LIST command to display a disk directory. The program "TEST:0" should be listed.

Now load the "GET:0" program. Again examine the program and attempt to understand the program flow. To operate, again enter

```
RUN
```

The "GET:0" program will ask for a file name. Enter the "TEST:0" name. Each record from the file will be displayed on the screen. This not only tests the file handling capabilities of your PDOS software but also provides an example of how things are done.

Enjoy your PEDISK II. You have an easy to use yet powerful disk.