

**EPYX™**

**FOR  
COMMODORE  
64™**

**Fast  
Load™  
CARTRIDGE**

Program Designed by Scott Nelson

**INSTRUCTION MANUAL**

# GETTING STARTED

- 1) Set up your Commodore 64™ Computer as shown in the Owner's Manual.
- 2) Make sure your computer is turned OFF.
- 3) Insert the FAST LOAD CARTRIDGE™ into the cartridge slot of your computer.
- 4) Turn the computer ON.
- 5) Turn the disk drive ON.
- 6) Insert your program disk.

NOTE: There is no need to remove the FAST LOAD CARTRIDGE once inserted. FAST LOAD will work on most protected disks and is intended for use with the Commodore 64™ 1541 disk drive.

## USING FAST LOAD LOADING

After turning the computer ON, the word "FASTLOAD" will appear just below the Basic READY prompt.

- To list a directory without erasing the program in memory, type \$ or >\$.
- To run most disk software, hold down the **Commodore** key (C =) and press the **RUN/STOP** key. This will eliminate typing LOAD "\*" ,8,1 and then typing RUN each time you load.
- To load a basic program, type /**FILENAME**. This will eliminate typing LOAD "FILENAME" ,8.
- To save a basic program, type ←**FILENAME**. This will eliminate typing SAVE "FILENAME" ,8.
- To load a machine language file, type %**FILENAME**. This will eliminate typing LOAD "FILENAME" ,8,1.
- To send a command to the disk drive, type @**COMMAND** or >**COMMAND**. This will eliminate typing OPEN 15,8,15,"COMMAND":CLOSE 15.
- To read an error channel (when the red light on the disk drive is flashing) type ">" or "@".

## DISK TOOLS

To run the disk tool, type "£". This will bring up a menu screen. Press the appropriate letter for your desired function:

- A – DIRECTORY**
- B – RETURN TO BASIC**
- C – COPY**
- D – DISABLE FASTLOAD**
- E – EDIT DISKETTE**
- F – FILE UTILITY**

- A – DIRECTORY:** This will show a directory of the diskette.
- B – RETURN TO BASIC:** Will return to Basic, leaving FAST LOAD intact.
- C – COPY:** Selecting this option will give you another menu:
  - A – DIRECTORY:** This will list the directory.
  - B – RETURN TO THE FIRST MENU:** This will return you to the first menu.
  - C – COPY ENTIRE DISKETTE:** Will copy every sector of one diskette to another diskette. Note: The new diskette must be formatted for this option to work properly.
  - D – BAM COPY:** Will copy only those sectors on a diskette which have been allocated by the DOS. This is much faster than copying the entire diskette. Note: The new diskette must be formatted for this option to work properly.
  - E – COPY A FILE:** Will copy a program file from one diskette to another. Wild card characters ("\*" or "?") are allowed in the file name.
  - F – FORMAT DISKETTE:** Will erase a diskette. This option should be used before a BAM copy or entire disk copy is attempted. Note: This will erase any files on the disk.
- D – DISABLE FASTLOAD:** This will disable FAST LOAD and return you to Basic. Once this option has been selected, the only way to use FAST LOAD is to turn the Commodore 64 Computer Off and On again.
- E – EDIT DISK:** Selecting this option will give you a new menu:

```
EDIT DISKETTE      TRACK 12    SECTOR 02
READ WRITE        QUIET
```

First, you must enter the track and sector you wish to edit in hex. (If you prefer decimals, type a "#" before the number). Then you will see that sector displayed before you. To change a byte, enter the new value in hex. To move within a sector, use the cursor keys. To read a sector, type "R". To write a sector, type "W". To quit, type "Q".

**F – FILE UTILITY:** Selecting this option will give you a new menu:

**A – DIRECTORY:** This option will list the directory.

**B – RETURN TO THE FIRST MENU:** This will return you to the first menu.

**C – COPY A FILE:** This will copy a program file from one diskette to another.

**D – DELETE A FILE:** Will remove a file from the directory.

**E – LOCK A FILE:** Will “lock” a file, making it impossible to delete the file without first unlocking it. A locked file will have a “<” after it in the directory.

**F – UNLOCK A FILE:** This will unlock the file, making it possible to delete it.

**G – RENAME A FILE:** Allows you to change the name of a file.

## ADVANCED PROGRAMMING

For advanced programmers, the following are assembly monitor instructions:

### SMON

In addition to the above disk tools, the FAST LOAD CARTRIDGE contains a powerful monitor. Since the major use of a monitor is to “debug” assembly language programs, if you are not familiar with this, you may want to skip this section. To move into the monitor, type “! ”.

To exit the monitor, type “%”.

### SYNTAX

A command in SMON is usually entered as several arguments followed by a single command character. The command character can usually be placed anywhere on the line, therefore **800,850\*** is the same as **800\*850**, or **\*800 850**. Commas, periods, and spaces may be typed anywhere, and serve only to separate numbers. Numbers may be entered in hex (the default), decimal (by preceding it with “#”), or as ASCII by typing a single quote before it, or as a lot of ASCII by surrounding many characters with double quotes. Numbers may also be combined with any of the five following operators: “&” (logical and), “|” (logical or), “^” (logical exclusive or), “+” (plus), “-” (minus), “(” (left parenthesis), and “)” (right parenthesis). For example:

```
(11 + 4) & (5 | 1E ? 1) - #10 = A
(15) & (1E)           - A = A
14                   - A = A
```

**0000G:** Begin execution of M.L. code at location 0000 via JSR.

**0000 1111H22 33 44 55 . . . :** This command hunts for a sequence of bytes. 0000-1111 is the memory which will be searched. 22 33 sets a range between which the first byte will be accepted, 44 55 the second, and so on. For example: 0 10000HA9 A9 80 90 would search for an A9 followed by something between 80 and 90. Up to 10 bytes can be searched for.

**0000J:** Begins execution of M.L. code at location 0000 via JMP.

**00 11 2222 3333I:** Will disassemble memory at 2222-333 and print all immediates between 00 and 11.

**0000 1111L:** Will disassemble memory at 0000-1111 and print it to the screen.

**0000 1111 2222M:** Will move the block of memory at 1111-2222 down to 0000. The move instruction always moves the lowest byte (1111) first so some moves may not work correctly. For example: 4001 4000 8000M will not move the block up one byte, instead 4000 will be moved to 4001, then 4001 to 4002. Thereby filling 4001-8001 with the byte in 4000.

**N:** No Operation.

**0000 1111 2222 3333 4444Q:** To relocate. Takes the code at 3333-4444 and changes all absolute addresses in the range 1111-2222 by 0000 (subtracts 0000 from it).

**0000 11 R“SSSSSSSS”:** Binary load from device 11. If 0000 is specified, then this will load the file SSSSSSSS into memory at location 0000. If not specified, then it will load where it was saved. The “R” MUST be located just before the filename.

**0000S:** Executes one M.L. OP-CODE, then displays the contents of the registers. 0000 is the address of the OP-CODE to execute, or just “S” will continue with the next instruction. For example: **600S:** will execute the instruction at 600. **S:** Will execute the one after 600.

**0000T:** Begin execution at 0000 and display registers after each instruction.

**0000 1111 2222 V:** Checks the block 1111-2222 against the memory starting at 0000 if the location contents differ. Then it will print both contents.

**0000 1111 22 W“SSSSSSSS”:** Binary save. Saves the memory 0000-1111 (inclusive) in the file SSSSSSSS on device 22.

**00 11 2222 3333X Y @:** Disassembles memory from 2222-3333 and prints indirects between 00-11. **X** will only print indirect **X** (**X**), **Y** will only print indirect **Y** (**Y**), **@** will print all indirects.

**0000 1111 2222 3333Z:** Disassembles block at 2222-3333 and prints all absolute or zero page references in the range 0000-1111. This includes references from branches.

< . . . : These are separators, and are ignored.

**0000 1111\*:** Displays memory from 0000 to 1111 in HEX and ASCII.

**0000 1111 ↑:** Prints the ASCII of memory from 0000-1111.

**£AA XX YY PP SP:** Displays registers. If followed by numbers, then those numbers are stored in those registers. To change **X**, it is necessary to change **A** as well.

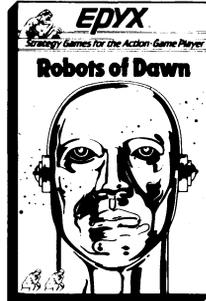
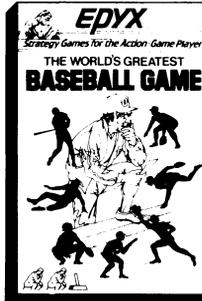
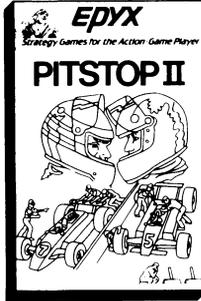
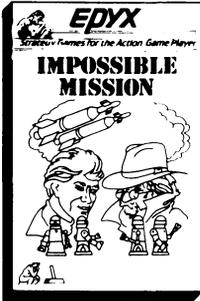
**%:** Returns you to basic.

**>“COMMAND”:** Eliminates the basic command, OPEN 15,8,15,“COMMAND”.

**\$:** Prints a directory.

**0000 =:** Prints the HEX, DECIMAL, BINARY, and ASCII value of 0000.

# NEW EPYX GAMES THAT WORK WITH FAST LOAD CARTRIDGE™



## LIMITED WARRANTY

Epyx, Inc., warrants to the original purchaser of this Epyx software product that the medium on which this computer program is recorded is free from defects in materials and workmanship for a period of ninety (90) days from the date of purchase. This Epyx software program is sold "as is," that is without express or implied warranty of any kind, and Epyx is not liable for any losses or damages of any kind resulting from use of this program. Epyx agrees for a period of ninety (90) days to either repair or replace, at its option, free of charge, any Epyx software product, postage paid, with proof of date of purchase, at its Factory Service Center.

This warranty is not applicable to normal wear and tear. This warranty shall not be applicable and shall be void if the defect in the Epyx software product has arisen through abuse, unreasonable use, mistreatment or neglect. THIS WARRANTY IS IN LIEU OF ALL OTHER EXPRESS WARRANTIES AND NO OTHER REPRESENTATION OR CLAIMS OF ANY NATURE SHALL BE BINDING ON OR OBLIGATE EPYX. ANY IMPLIED WARRANTIES APPLICABLE TO THIS SOFTWARE PRODUCT, INCLUDING WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, ARE LIMITED TO THE NINETY (90) DAY PERIOD DESCRIBED ABOVE. IN NO EVENT WILL EPYX BE LIABLE FOR ANY SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGE RESULTING FROM POSSESSION, USE OR MALFUNCTION OF THIS EPYX SOFTWARE PRODUCT.

Some states do not allow limitations as to how long an implied warranty lasts and/or the exclusion or limitation of incidental or consequential damages so the above limitations and/or exclusions or limitation of liability may not apply to you. This warranty gives you specific legal rights, and you may also have other rights which vary from state to state.

**EPYX**  
COMPUTER SOFTWARE

1043 Kiel Court, Sunnyvale, CA 94089

Commodore 64 is a trademark of Commodore Business Machines, Inc.

© 1984 EPYX, Inc.

PROOF OF PURCHASE  
PRODUCT NO.  
A107R

Part No. A1007R-60