Power-Loader Extensions for 64HDD

Overview

Power-Loader (Pwr-Load, for short) is a parallel cable interface for 64HDD. It allows data to be transmitted one byte, rather than one bit, at a time. Extremely fast transfer speeds are possible, but to use this functionality you need to attach a cable between your user-port and the 64HDD machine. Typical transfer rate is 30-50times faster than the standard IEC serial load. Support is available in all file modes, including the MSD file mode (including LFN support) and all supported images.

Introduction

Pwr-Load is a fully synchronous parallel transfer scheme for 64HDD. Being fully synchronous means that no data is lost, regardless of what graphics are being displayed.

The transfer scheme is possible with all Commodore computers that have a user port (and of course, an IEC serial port). That means only the C16 misses out. The transfer happens in one of four ways:

- Auto-run loader: used to load single part software. Loading the file with special partition prefix will initiate the load. For example: LOAD"990:filename",11,1 will Pwr-Load the file.

- Runtime loader: used to Pwr-Load software under program control. The loader is installed into the system input buffer using the normal IEC routines, a subsequent SYS call will bring in the rest of the file.

- Custom loader: using the example code as a starting point, Pwr-Load features can be added to your programs allowing files up to 16Mb to be streamed to the Commodore

- Modified Kernal (a C64 Kernal is now available): this allows seamless integration of Pwr-Load with all programs using Kernal LOAD functions. To use this functionality an EPROM needs to be burnt. Alternatively, a pre-programmed EPROM can be purchased by contacting the author.

Typically the parallel transfer occurs under software control via a "loader". The protocol is the same for all machines, but because the user-port is at different memory locations in different machines, a different loader is required for each machine.

Installation and Configuration

Cable and 64HDD settings:

Firstly, you will need a parallel cable. For details see the Cable specification section. The cable is relatively straight-forward to make (or buy if you already have a user-port printer port).

Secondly, you'll need to install the loaders. Installation is easy, just unzip the `PWR-LOAD.ZIP` using the `-d` option. Loaders for each Commodore platform will be placed in the 64HDD system directory. {Currently, loaders are only available for some systems. The others are still being developed.}

It is strongly recommended that your 64HDD system is configured with either a RAMdisk or disk caching system such as SmartDrv. The reason for this is more evident when Power-Loading from a disk image since the original file is first spooled to a temporary file in the 64HDD system directory before being sent in parallel mode.

If you will only be using 64HDD with one Commodore computer type, then you can configure 64HDD to know that. By default, 64HDD assumes you will be using a C64, and in that case no further action is required.

To configure the default for another machine, use the +pwr command line option (or add this to the GO64HDD batch file). Options are:

+pwr	990	(default) C64
+pwr	992	C16/Plus4
+pwr	994	VIC20
+pwr	996	C128

Configuring the default allows you to shorten the partition number to the just an exclamation mark, for example:

```
LOAD"990:filename",11,1
```

Can be typed as:

```
LOAD"!:filename",11,1
```

Or using a DOS wedge such as JiffyDOS:

```
%!:filename
```

To load the default runtime loader (pwr+1) use the following:

```
LOAD"!!:filename",11,1
```

With all case, partition and direct path support is now limited as the Pwr-Load function is actually using a partition number. Hence the following may not be valid:

```
LOAD"!!:99:filename",11,1
```

...and will probably result in an error. The work-around is to change to the appropriate drive first using $ or CP commands.

Notes: 1) with some systems, Power-Loader will only work reliably if the other drives on the daisy chain are switched on!

2) the drive "click" is sounded every 2k transmitted instead of every block as per IEC serial mode.

3) when Power-Loading from a disk image, there may be a brief pause before the parallel transmission begins. During this time the disk image file is spooled to a temporary file.

Kernal EPROM Replacement:

The replacement Kernal is compatible with the standard CBM Kernal. The main change has been a JMP wedge installed in the ROM at the normal LOAD routine to add "!!:" to the front of the filename and to load the appropriate runtime loader, checking it for validity before initiating a Power-Load.

If the runtime loader is not loaded (for example the drive is a real CBM drive and not a 64HDD drive) the normal LOAD routines will have loaded the file in any case.

The rules followed before pre-fixing the filename with "!!:" are as follows:

- No prefix is added when a directory load is issued, ie with a leading $
- No prefix is added when a direct file load with ":filename" is issued
- If a unit number is included, for example "0:testfile" the "0:" is replaced with "!!:"

Unfortunately, not all programs use Kernal calls for their loaders and in these instances the programs will load without and Pwr-Load assistance.

Also, when a real CBM drive is sent "!!:filename" it treats the name as "0:filename". This normally is not a problem except for when "!!:*" is sent as it will try to load the first file on the disk with "0:*", regardless of file type (and sometimes this may be a banner using type DEL).

Command: Loading software from “READY."
Applicability:

■ All ☐ MSDOS ☐ D64 ☐ D71 ☐ D81 ☐ H64 (Native) ☐ T64 ☐ LNX

Syntax:
```
LOAD”pwr:filename[,Xstart]”,drive,1
LOAD”!:filename[,Xstart]”,drive,1
```

Examples:
Pwr-Load file on a C64
```
LOAD”990:filename”,11,1
```

Where pwr:

990	(default) C64
992	C16/Plus4
994	VIC20
996	C128

The “,X” option allows the data stream to be forwarded to “start” bytes into the file, counting from after the first two bytes which define the load address. “start” is given as ASCII. The auto and runtime loaders use the original load address. This option is only available to the MSD file system.

The “,{shift-X}” option allows the data stream to be forwarded to “start” bytes into the file, counting from after the first two bytes which define the load address. “start” is given in binary as lo, middle, high bytes. The auto and runtime loaders use the original load address. This option is only available to the MSD file system.

Errors:
- Error is 0, if load is successful.
- Error [92] if the requested loader is not installed (or located in the system directory)

Notes:
- The software is loaded using the auto-run loader. If the file is being loaded into the current BASIC area, BASIC text pointers are adjusted so that file saves will work correctly.
- Instead of `READY` the prompt will read `PWR-LOADED!`, once loading is completed. The prompt will only appear when the load is requested from direct mode.
- Vectors from 806-817 will be over written with the standard CBM values, hence any patches applied after system “on” will be over-written.
- Data in the datasette buffer will be over–written by the loader.

Command: Loading software from within a running program
Applicability:

■ All ☐ MSDOS ☐ D64 ☐ D71 ☐ D81 ☐ H64 (Native) ☐ T64 ☐ LNX

Syntax:
```
LOAD"pwr:filename[,Xstart]",drive,1
LOAD"!:filename[,Xstart]",drive,1
```

Examples:

Pwr-Load file on a C64 using the auto-loader

```
0 IF A=0 THEN A=1:LOAD"990:filename",11,1
1 IF A=1 THEN SYS832
2 A=2
3 REM REST OF PROGRAM…
```

Where pwr:

990	(default) C64
992	C16/Plus4
994	VIC20
996	C128

Pwr-Load file on a C64 using the runtime loader

```
0 IF A=0 THEN A=1:LOAD"991:filename",11,1
1 IF A=1 THEN SYS512
2 A=2
3 REM REST OF PROGRAM…
```

Where pwr:

991	C64
993	C16/Plus4
995	VIC20
997	C128

Errors:

- Error is 0, if load is successful.
- Error [92] if the requested loader is not installed (or located in the system directory)

Notes:

- See notes for auto-run loader.
- The runtime loader is stored in the input buffer. Because of this, the SYS call needs to be made before the next system input is performed.
- The auto-run transfers data at a faster rate than does the runtime loader. The programmer can use whichever is works best with the structure of the existing program.

Command: Using a custom loader
Applicability:

■ All ☐ MSDOS ☐ D64 ☐ D71 ☐ D81 ☐ H64 (Native) ☐ T64 ☐ LNX
Syntax: `OPEN file,drive,channel,"999:filename[,Xstart]"`

Examples:

Pwr-Load file on a C64 using a custom loader at 49152

```
1 OPEN 1,11,2,"999:filename"
2 GET#1,A$: REM DUMMY CHARACTER READ TO SETUP PWR-LOAD
3 CLOSE 1
5 SYS49152: REM CUSTOM LOADER
6 REM REST OF PROGRAM…
```

Errors:

- Error is 0, if load is successful.
- Error [92] if the requested loader is not installed (or located in the system directory)

Notes:

- See coding examples included with PWR-LOAD distribution.
- The loader can be written in BASIC, however this will probably execute slower than a standard serial IEC load
- Opening to partition 999: sets the name of the file to be loaded.

Power-Loader Details:

Data Format:
Data sent from 64HDD always follows this format:

Byte 1	low byte of start address
Byte 2	high byte of start address
Byte 3	low byte of end count
Byte 4	mid byte of end count
Byte 5	high byte of end count
Byte 6….n	data stream

Because three bytes define the end count, data streams up to approximately 16MB can be handled. Such long data streams are not necessarily program files, as a custom loader could transfer audio, video, or other data using the parallel link.

For normal program loads however, only the low and mid bytes are significant, and should define the highest address +1 loaded.

It should be noted that any time an ATN signal is activated the data stream gets cancelled. The ATN signal is activated by any IEC serial activity, even to another device number. The reason for this is that the transfer scheme uses the CLK and DATA lines in addition to the 8bits on the user-port and so cannot co-exist.

Auto-Run Loader:

The Auto-run loader relies on the patching the CHROUT routine to autostart. The routine is stored on top of these vectors and through the datasette buffer on the C64 (location 806+). The routine adjusts the BASIC text pointers for files loaded into the BASIC area. Locations 251-255 are also temporarily used for pointers.

The Auto-run loader is requested by using one of the following partition numbers: 990, 992, 994 or 996.

The faster load routine built into this loader can be used as an alternative to the runtime loader (see below) from within a BASIC program by calling SYS832 (on the C64) after the loader portion has loaded.

Runtime-Loader:

The runtime loader is shorter in file length than the auto-run loader, but is not as quick as a result of the code relying on Kernal calls. The difference speed equates to about one second for a 154block file.

The runtime loader is loaded into the input buffer used by the Kernal and BASIC. This memory area is seldom used by programs for other purposes as key strokes will be written into the area on INPUT, or during other input

routines. However, the code loaded into the buffer is available if used before such an input occurs.

The runtime loader is requested by using one of the following partition numbers: 991, 993, 995 or 997.

Once loaded the transfer can be started from within a BASIC program by calling SYS512 (on the C64). Locations 251-255 are also used for temporary pointers.

Custom-Loaders:

These can be written following the example source provided for the auto-run, runtime or custom loaders. Using a custom loader has the benefit of saving about 1second during each Pwr-Load as the loader does not have to be transferred to the Commodore computer each time. It is also possible to add Pwr-Load capabilities to programs that may not work with the above options. Some considerations when writing you own loader are: 1) minimise JSR calls; 2) in-line code as much as possible; 3) use 2MHz mode if you want to cater for a C64/C128 machine; 4) blank screen; 5) directly code write to the IEC control lines and 6) disable IRQs. Nearly 40kB per second can be achieved in combination with the correct PC hardware and custom loader.

Transfer Specification and Speed:

Data transfer is governed by a synchronous protocol. The Commodore holds-off 64HDD by holding the DATA line low, whilst 64HDD holds-off the Commodore by holding the CLOCK line low. As a result, the speed is a little slower than an asynchronous link. The link needs to be synchronous to cope with a range of PC hardware that might be used (different speed CPUs and different speed drives within each system itself – floppies are slower than hard disks, etc). A synchronous link also allows the Commodore to perform normal IRQ functions, etc

The maximum transfer speed is about 20kilobytes per second when everything on the PC is at its optimum, even higher when running in 2MHz mode. On older hardware 13kilobytes per second can be reliably achieved.

Transferring the loader takes about 1second as the standard IEC protocol is used and the file has to be located and sent serially.

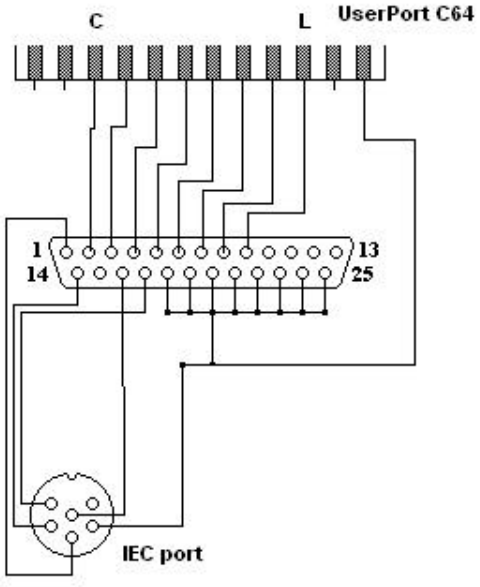
Power-Loader Cable Specification:

Parallel Connection Options:
In addition to the X1541 or XE1541 cable used with 64HDD, parallel loading support requires that a cable be connected from the 64HDD LPT port to the user-port on the Commodore computer. The link is very simple, requiring 8-data lines connected bit for bit, D0 to PB0, D1 to PB1, etc, plus a GND connection to ensure that the machines are grounded to the same signal level (although the IEC/X1541 cable has a GND connection, it is advisable to have one on the second cable should the system ever be powered on with only one of the cables connected).

There are several options for constructing the cable:

1) Direct method: 8-data lines, plus GND directly:

LPT Port	C64 UserPort	C128 UserPort	VIC20 UserPort	PLUS4 UserPort
2	C	C	C	?
3	D	D	D	?
4	E	E	E	?
5	F	F	F	?
6	H	H	H	?
7	J	J	J	?
8	K	K	K	?
9	L	L	L	?
19	A,N,1,12	A,N,1,12	A,N,1,12	?



2) LPT to LPT: for those that have a standard printer or GeoCable interface already for their Commodore, the connection is even simpler. All you need is a cable that provides pin-for-pin connection of the 9

wires, for example: 2-2, 3-3, 4-4, etc. Such a cable should be readily available or easily wired.

3) 64HDD Standardised Connection: personally I have chosen to use a female DB9 socket as the "drive" end of the parallel cable since it has the minimum number of pins needed to satisfy requirements. To assist construction the following pin-out has been used as it allows a ribbon cable to be connected to both ends easily.

LPT	Function	DB9
2	D0	1
3	D1	6
4	D2	2
5	D3	7
6	D4	3
7	D5	8
8	D6	4
9	D7	9
19	GND	5

4) XP cable: The XP cable used by StarCommander may be used providing you have a suitable DB15 to user-port adaptor.

Connecting 64HDD Pwr-Load and Centronics at the same time:
A userport Centronics printer and Power-Loader can be used on the same port with this simple hardware design. Each of the eight data lines from the 64HDD machine needs to be connected with a BAT85 diode (or other Schottky diode), cathode pointing towards the 64HDD port. These diodes will effectively decouple the 64HDD system when a parallel transfer is not occurring, thus allowing the CBM computer to send data to the printer. (This circuit modification was designed and tested by Jochen Adler).

Please note: the data is provided for guidance. I have successfully built this circuit, but you take responsibility for the modifications you make to your computer.

Electrical compatibility: The LPT data port is set to output once MSDOS has loaded, whilst the Commodore's User-Port is set to input when powered on. As a result it is ok for these ports to be connected. However, both ports should not be set to output at the same time as either /both ports may be damaged.