



Building PPCBoot

for the

Compact Flash Computer

Version 1.0

24 Nov 2004

## Notice

The information in this document is subject to change without notice.

**THE SOFTWARE AND DOCUMENTATION ARE PROVIDED “AS IS” WITHOUT WARRANTY OF ANY KIND INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. FURTHER, C Data Solutions Ltd DOES NOT, GUARANTEE, OR MAKE ANY REPRESENTATIONS REGARDING USE, OR THE RESULT OF THE USE, OF THE SOFTWARE OR WRITTEN MATERIAL IN TERMS OF CORRECTNESS, ACCURACY, RELIABILITY, OR OTHERWISE.**

# Table of Contents

|  |   |
|--|---|
| Installing the 'LittleBoy' distribution..... | 4 |
| Building PPCBoot.....                        | 4 |
| FLASHING PPCBoot.....                        | 4 |
| DOWNLOADING WITH GDB.....                    | 5 |

## Installing the 'LittleBoy' distribution

The 'Littleboy' distribution CD contains all the tools and sources needed to start developing applications.

To install the tool chain

```
sh $path_to_cdrom/m68k-elf-tools-20030314.sh
```

To install the ppcboot distribution

```
tar xzvf $path_to_cdrom/ppcboot.tar.gz
```

This will install the distribution in a folder "ppcboot" in the current location.

## Building PPCBoot

In the PPCBoot directory type

```
[root@pengu ppcboot]# make littleboy_config  
[root@pengu ppcboot]# make
```

This will build the complete ppcboot tree, generating ppcboot.bin.

## FLASHING PPCBoot

Start the terminal emulator (preferably kermit) in the ppcboot directory.

The defaults for the Compact Flash Computer are 19200 baud, 8 bits, 1stop bit, no handshake.

Boot the Compact Flash Computer.

At the prompt type

```
Littleboy>loadb 20000
```

Enter the local mode of Kermit Ctrl-\ c, and send the ppcboot.bin to the Compact Flash Computer.

```
(/usr/local/littleboy/ppcboot/) C-Kermit>send ppcboot.bin
```

When the download is finished, return to the terminal emulator mode by typing **c**.

The **ppcboot.bin** image can now be programmed into flash. First the protection on the **FLASH** area to program is removed.

```
LittleBoy>protect off ff800000 ff81ffff
Un-Protected 9 sectors
LittleBoy>
```

**The FLASH is erased.**

```
LittleBoy>erase ff800000 ff81ffff
Erase Flash from 0xff800000 to 0xff81ffff
Erasing sector 0 ... ok.
Erasing sector 1 ... ok.
Erasing sector 2 ... ok.
Erasing sector 3 ... ok.
Erasing sector 4 ... ok.
Erasing sector 5 ... ok.
Erasing sector 6 ... ok.
Erasing sector 7 ... ok.
Erasing sector 8 ... ok.
Erased 9 sectors
LittleBoy>
```

**The FLASH is programmed by coping the program to the FLASH area.**

```
LittleBoy>cp.b 20000 ff800000 20000
Copy to Flash... done
LittleBoy>
(/usr/local/littleboy/ppcboot/) C-Kermit> send ppcboot.bin
```

## DOWNLOADING WITH GDB

You will need a bdm patched version of gdb.

In the ppcboot directory type

```
[root@pengu ppcboot]# ./bdm/load
GNU gdb 5.0
Copyright 2000 Free Software Foundation, Inc.
GDB is free software, covered by the GNU General Public License, and you are
welcome to change it and/or distribute copies of it under certain conditions.
Type "show copying" to see the conditions.
There is absolutely no warranty for GDB. Type "show warranty" for details.
This GDB was configured as "--host=i686-pc-linux-gnu --target=m68k-bdm-elf"...
Coldfire debug module version is 0 (5206(e))
0x10000010: 0x4440301d
0x10000180: 0x00004303
(gdb) Quit
(gdb)
```

Now type c. This instructs gdb to continue with program execution. The script for loading ppcboot.bin also sets up the start location of the program.