

Microblaze uClinux Demo Package

Release 1.00e

22nd August 2003

John Williams
School of ITEE, University of Queensland
Brisbane, AUSTRALIA

jwilliams@itee.uq.edu.au
<http://www.itee.uq.edu.au/~jwilliams/mblaze-uclinux>

1. Introduction

This package is an easy-to-use demonstration of the uClinux operating system port to the Xilinx FPGA-based Microblaze soft processor. It provides precompiled hardware and software binaries plus scripts to download them onto an FPGA evaluation board. The pre-built demo is currently limited to the Insight/Memec V2MB1000 evaluation board.

uClinux is a version of the Linux operating system for processors without a memory management unit (MMU). From a user and application perspective, there is very little difference between uClinux and Linux. Most of the applications that comprise this demo are part of the regular Linux distribution. You can find more information about uClinux at <http://www.uclinux.org>.

Yes, you really can have an OS on an FPGA, and what's more, it's Free Software!

2. Disclaimer

The contents of this package are intended for use only on a Memec/Insight Electronics Virtex2 1000 system board (V2MB1000), revisions 1 or 3 (there is no revision 2 board). Their use on any other platform may cause damage to your hardware. If you are not sure what you are doing, do not attempt to use this demo. The authors accept no liability for any damage, inconvenience or lost productivity caused by the information and data contained in this package.

3. License

The following statement applies to the uClinux binary kernel image distributed as part of this demo package. The file COPYING contains the full text of the GNU General Public License.

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

The source code for this GPL software is available for download from <http://cvs.uclinux.org>. All other files in this package, including documents, scripts and binary files are the copyrighted work of the author, unless otherwise stated. The demo package may be re-distributed freely, provided that this notice is left intact.

4. Hardware / Software Requirements

- Memec/Insight V2MB1000 prototyping board (revision 1 or 3)
- Memec/Insight P160 Communications Module
- Windows PC with Xilinx EDK 3.2 SP2 installed
- Two COM ports (or one COM port and a second terminal such as a laptop or Linux box) and serial cables
- Parallel JTAG cable (serial or USB Multilyn cable will require editing of `_impact.cmd` – not described here)

5. Demo Outline

The demo consists of two principal components:

- 1) A kernel and operating system image (`image.bin`)
- 2) FPGA and PROM configurations (`demo.bit / demo.mcs` and `demo_selfboot.bit/demo_selfboot.mcs`)

The kernel image is uploaded to the board with the XMD utility, and may be optionally written into flash memory. Similarly, the FPGA configuration is uploaded using the iMPACT software, and optionally may be written into the configuration PROM. Automated scripts are provided to perform these actions.

For convenience it is recommended that both the image be copied to flash, and the FPGA configuration written into the PROM. This is referred to here as the “Persistent Demo”, and avoids the time-consuming process of uploading images and configurations in between demo sessions. Optionally, after flash programming a self-booting configuration may be loaded into the PROM, to permit the board to automatically boot to the uClinux kernel upon reconfiguration.

However, some users may prefer not to write to the flash or configuration PROM. This option is also supported, and is called the “Single Use Demo”. Instructions for both are provided below.

6. Terminal Requirements

The demo requires that two serial connections are made to the V2MB1000 board.

COM1 is connected to the serial port on the V2100MB P160 communications module, for use by `xmd` to download kernel and bootloader images.

A second COM port is required to act as the uClinux console. This must be connected to the “primary” serial port on the V2MB1000 main board (just beneath the DIP switches). Terminal emulation software, such as HyperTerminal or TeraTerm on Windows, or `minicom` on Linux, is required. It should be connected to this port configured at 57600 baud, no parity, software (XON/XOFF) flow control.

7. Single-Use Demo

The following instructions permit a “once-off” demonstration. The advantage of this is that no changes are made to the Flash memory or configuration PROM on the V2MB1000 board. However, it requires that the image and FPGA configuration be downloaded every time.

To begin, open a Xygwin shell, and change to the directory where you installed the uClinux demo.

7.1. Configuring the FPGA

Run the “configure_fpga” script:

```
$ ./configure_fpga.sh
```

IMPORTANT

The script will ask you which revision Insight/Memec board you are using – enter “1” or “3” as appropriate¹.

After several moments the software should report that the FPGA is programmed successfully. If not, check your JTAG cable connection to the board and try again.

7.2. Downloading the Image

First, ensure that xmd.ini is correct for your platform. In particular, make sure that the correct COM port is identified. Do not change the buad rate – that is fixed by the hardware target.

Next, run the “download_image” script:

```
$ ./download_image.sh
```

This will take several minutes, and should complete without reporting any errors.

7.3. Prepare the Terminal

Ensure that your terminal emulation software (e.g. HyperTerminal or TeraTerm on Windows, or minicom on Linux) is running and properly configured. See the “Terminal Requirements” section for details).

7.4. Launching the Demo

Finally, run the “run_bootloader” script:

```
$ ./run_bootloader.sh
```

The bootloader menu should appear in the terminal window:

```
J-Boot Menu
-----
1.   Execute Flash image
2.   Execute RAM image
3.   Erase Flash
4.   Write image into Flash

Make your choice>
```

¹ In general, revision 1 boards have red power LEDs, while revision 3 boards are green. Check your V2MB1000 user manual if you are unsure.

If the menu does not appear, check your terminal settings and try again.

Finally, select option “2. Execute RAM image” from the menu. Read the Section “Using the Demo”.

8. Persistent Demo

Note: This demo will copy FPGA configuration files and binary kernel image data into the configuration PROM and Flash memory respectively. These changes are persistent, but not permanent. If you are unsure what this means, use the “Single Use Demo”, described later in this document.

8.1. Configuring the FPGA / PROM

Jumper J1 on the V2MB1000 board must be properly set to place the FPGA in Master Serial Mode. This requires that all jumper pairs in J1 are closed. Refer to the V2MB1000 reference manual for details. No other changes are required from the default jumper settings of the V2MB1000 board.

To begin, open a Xywin shell and change to the directory where you installed the uClinux demo.

Next, run the “configure_prom” script:

```
$ ./configure_prom.sh
```

IMPORTANT

The script will ask you which revision Insight/Memec board you are using – enter “1” or “3” as appropriate².

This should take several moments, and finish without reporting any errors.

8.2. Downloading the Image

First, ensure that xmd.ini is correct for your platform. In particular, make sure that the correct COM port is identified. Do not change the baud rate – that is fixed by the hardware target.

Next, run the “download_image” script:

```
$ ./download_image.sh
```

This will take several minutes, and should complete without reporting any errors.

8.3. Prepare the Terminal

Ensure that your terminal emulation software (e.g. HyperTerminal or TeraTerm on Windows, or minicom on Linux) is running and properly configured. See the “Terminal Requirements” section for details).

8.4. Launching the Bootloader

The bootloader is used to copy the image from RAM (where it currently resides) into the Flash. Run the “run_bootloader” script:

```
$ ./run_bootloader.sh
```

The bootloader menu should appear in the terminal window:

² In general, revision 1 boards have red power LEDs, while revision 3 boards are green. Check your V2MB1000 user manual if you are unsure.

```
J-Boot Menu
-----
1.    Execute Flash image
2.    Execute RAM image
3.    Erase Flash
4.    Write image into Flash

Make your choice>
```

If the menu does not appear, check your terminal settings and try again.

First, you must erase the Flash. Select option “3 Erase Flash” and then press “Y” to confirm.

After the flash is erased, the menu is redisplayed.

IMPORTANT

If you are using a *revision 3 board*, it is necessary to press the “Program” button on the V2MB1000 main board, and re-run the “run_bootloader.sh” script before continuing. The bootloader is unable to write the Flash until this is done.

Finally, choose option “4. Write Image into Flash”, and again press “Y” to confirm. This will take several minutes.

8.5. Launching the Demo

Once the Flash and PROM are properly programmed, press the “Program” button on the V2MB1000 board to force a reconfiguration of the FPGA.

The demo may now be launched by re-running the “run_bootloader” script from within a Xygwin shell, and selecting option “1. Execute Flash Image”. It will normally be necessary to first press the “Program” button on the V2MB1000 between each demo session.

See the section “Using the Demo” for more information.

8.6. Self-booting (optional)

Once the flash is properly programmed, you can optionally reconfigure the PROM in such a way that, upon reconfiguration, the board will automatically boot into the bootloader (stored in BRAM), from which you may directly enter the kernel. To do so, simply run the “configure_prom_selfboot” script:

```
$ ./configure_prom_selfboot.sh
```

Once again, you will be asked which revision Insight/Memec board you are using – enter “1” or “3” as appropriate.

Upon completion, the bootloader menu should be displayed in the terminal window. Selection option “1 Boot from Flash” to start the uClinux kernel.

9.2. Alternate Login

After the kernel has started, close XMD and connect a second terminal to the COM port (probably COM1) instead. Set it for 115200 baud, software (XON/XOFF) flow control, and press return a few times. You'll get a login prompt, try root, or bin, or games (check /etc/passwd) to see which users are defined. Welcome to multi-user computing, Microblaze uclinux style!

9.3. GPIO test

The hardware has a GPIO core connected to the DIP switches and 7 segment LEDs on the V2MB1000 board. This is mapped as a character device in the kernel (/dev/gpio), and can be exercised using the 'gpio_test' program. Try 'gpio_test A7'.

9.4. Unix/Linux Utilities

Try the following:

- cat
- ls
- cd
- pwd
- uptime
- ps
- ...

Look in /bin to get the full list.

9.5. Games

The old Un*x Classic "advent4" is there, give it a try!

9.6. The /proc File System

The proc file system (/proc) is a window into the internal operation of the kernel. Try some of the following:

- cat /proc/interrupts
- cat /proc/version
- cat /proc/meminfo
- cat /proc/mounts
- cat /proc/devices
- cat /proc/tty/driver/serial
- cat /proc/cpuinfo

You can also find info about a process by examining the files in the directory /proc/*pid*, where *pid* is the process ID as reported by 'ps'.

10. Acknowledgements

The author would like to acknowledge the support of the University of Queensland, and specifically his supervisor Professor Neil Bergmann, for financially and morally supporting the port of uClinux to the Microblaze.

Thanks also to the participants on the Microblaze uClinux mailing list, and in particular the volunteers who helped in testing this demo package.

Finally thanks to Xilinx and the EDK team for developing such a cool little processor and being open to the idea that it could handle a “proper” operating system.

11. Package Contents

<u>File</u>	<u>Purpose</u>
download_prom.sh	Script to download configuration PROM contents
download_prom_selfboot.sh	Script to download a self-booting PROM configuration
download_fpga.sh	Script to download FPGA bitstream
run_bootloader.sh	Script to download and run the uclinux bootloader
image.bin	uClinux kernel and file system image
demo.bit	Configuration bitstream for the uClinux demo hardware
demo.mcs	PROM contents file for the uClinux demo hardware
demo_selfboot.bit	Self-booting FPGA configuration
demo_selfboot.mcs	Self-booting configuration PROM contents
bootloader_xmdstub.elf	ELF image of the uClinux bootloader
xmd.ini	Default XMD startup file
download_prom_rev[1 3].cmd	iMPACT batch script to download the PROM file. Different files for revision 1/3 boards.
download_prom_selfboot_rev[1 3].cmd	iMPACT batch script to download the self-booting PROM file. Different files for revision 1/3 boards.
download_fpga_rev[1 3].cmd	iMPACT batch script to download the FPGA configuration. Different files for revision 1/3 boards.
COPYING	The GNU General Public License Document