

# How to set up uClinux for the Spartan 3E Starter kit

By consultant Søren Hansen (soren.hansen@teknologisk.dk)  
Copyright © 2007 Danish Technological Institute

## Contents

Preface.....	1
The Compiler tool chain .....	1
Acquiring the uClinux distribution .....	2
The Hardware layout file “auto-config.in” .....	2
Distribution setup.....	2
Setting the proper distribution .....	3
Customizing the kernel .....	4
Customizing the User programmes.....	13
Building the image.....	25
The Image itself .....	25
Saving and retrieving a given setup .....	27
Adding a user application .....	28
References.....	30

## Preface

This document describes in detail “howto” set up various options in uClinux, be they kernel or distribution options so that it is possible to configure and build an uClinux image which will run successfully on the FPGA in the Spartan-3E starter kit.

Prior to using “howto”, an appropriate configuration for the chosen board must be performed. It includes a MicroBlaze 4.0 for the soft-core and is in fact the core on which to run uClinux Linux. Alternatively, a howto can be downloaded which describes such a configuration, and it is actually the configuration with which howto has been tested [HOWTO-1].

To simplify the process, the document has been divided into different sections. Some may have completed them in advance and may thereby have enabled users to jump to the section or stage they wish to reach.

## The Compiler tool chain

The first initiative is to acquire the compiler tool chain. It can be found at the Petalogix web site (<http://www.petalogix.com/resources/downloads/mb-gcc>) together with a simple instruction explaining how to set up the compiler for the environment.

## Acquiring the uClinux distribution

The official uClinux web site is found at <http://www.uclinux.org> which also provides links to relevant information about distribution. Two different approaches can be pursued to acquire the distribution source. One is downloading a snapshot, the other is using cvs to download the current development version.

It is also the fastest way to acquire the source for a full source snapshot:  
<http://www.uclinux.org/pub/uClinux/dist/>.

Be prepared when using the cvs method - it takes a long time:

```
cvs -d:pserver:anonymous@cvs.uclinux.org:/var/cvs login
Press enter when prompted for a password.
cvs -z3 -d:pserver:anonymous@cvs.uclinux.org:/var/cvs co uClinux-dist
cvs -z3 -d:pserver:anonymous@cvs.uclinux.org:/var/cvs co uClinux-2.4.x
Make a symbolic mapping uClinux-2.4.x as uClinux-dist/linux-2.4.x
```

Note that the uClinux distribution used in this howto is the snapshot uclinux-dist-20060803.

## The Hardware layout file “auto-config.in”

At this point the compiler has been set up, and the distribution is ready for configuration. However, a proper working auto-config.in is required before distribution options can be set satisfactorily. The file contains a description of hardware setup and layout and is imperative to the Linux kernel. A proper working auto-config.in means the setup of HW as explained in [HOWTO-1].

The format of the auto-config.in file itself is the so-called DOS format and must be changed into UNIX format by using the command `dos2unix`<sup>1</sup>. The file is subsequently copied into the Linux kernel source tree to the following location:

```
linux-2.4.x/arch/microblaze/platform/uclinux-auto
```

We assume that the current path is the root of the distribution itself.

Note that only linux-2.4.x is known to work.

## Distribution setup

The next sections show the selected options for the setup which enables us to boot a Linux kernel on the Microblaze soft-core CPU in Spartan 3E FPGA.

When showing the selected setup or options, different dialogs will appear. However, in order to reduce the number of dialogs shown, only dialogs containing setup options are included. As a result, some dialogs are not included in their full form.

Finally it should be noted that the selected options are by no means a scientific choice, they are more a starting point to ensure a working system. The next step would therefore be to determine your particular needs and alter the appropriate options to ensure them.

---

<sup>1</sup> If you do not have this command, you can write as follows: `cat auto-config.in.org | tr -d '\r' > auto-config.in`

# Setting the proper distribution

## Intro

In order to properly configure the distribution according to your preference, enter the distribution path and start the configuration process. This process can be initiated in three ways<sup>2</sup>;

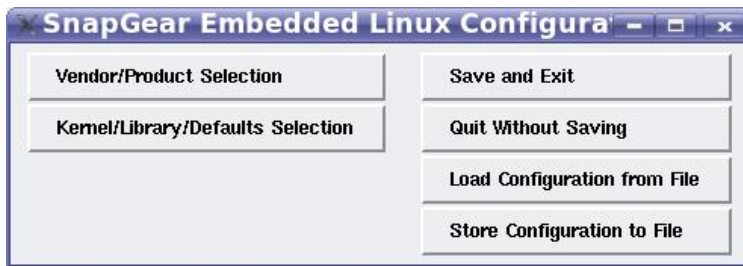
make config

make menuconfig

make xconfig

The first approach is seldom used and is not recommendable as it utilizes a simplistic non-intuitive interface. The other two are good, the latter being the preferred one. It should be noted, however, that it requires X-Windows.

## Configuring



In the main menu you have to enter both the Vendor/Product and the Kernel/Library/etc submenu once you have configured them accordingly. “Save and Exit”.

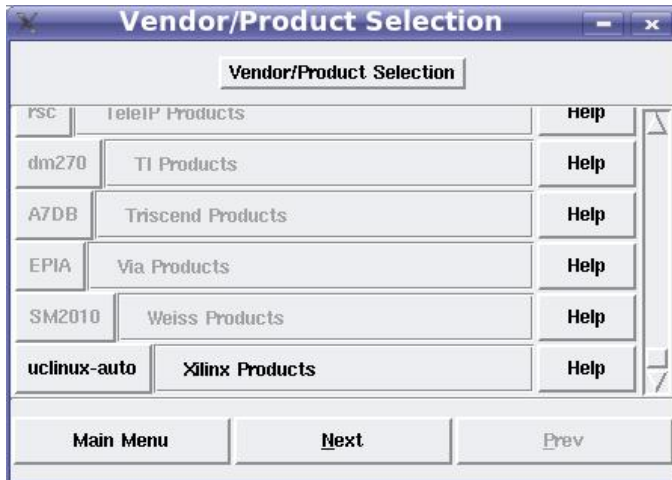
## Vendor/Product selection



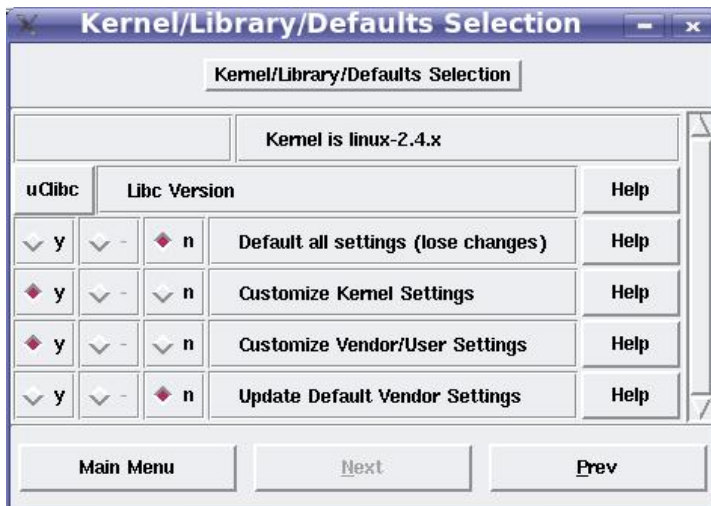
In the Vendor/Product menu you have to change the Vendor to Xilinx and select uclinux- auto as shown below.

---

<sup>2</sup> Note that all three config approaches achieve the same; it is just a question of usability.



## Kernel/Library/etc.



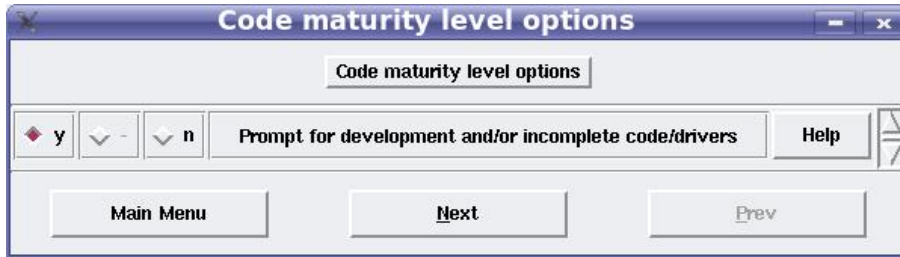
In this uClinux distribution, only the 2.4.x kernel has support for Microblaze. Support for Microblaze in version 2.6.x should be released soon or may already be released. However, this has not been verified.

Note also that we have to customize both the kernel and Vendor/User settings.

## Customizing the kernel

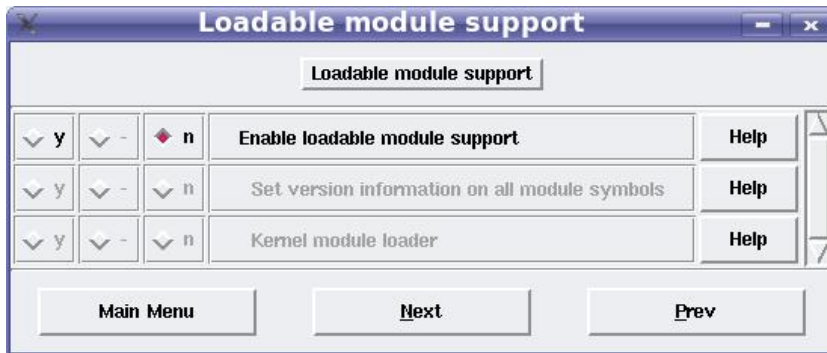
A significant point is that no networking support is included.

## Code maturity level



Normally you just say yes.

## Loadable module support



Including loadable module support makes it possible to load modules into a running kernel. It is not an important feature for our test and has therefore been left out. You may want to enable it.

## Processor type and features

Platform		
uclinux-auto	Platform	Help
RAM		
Memory model		Help
0x80000000	ROM base address (in hex)	Help
0x00100000	ROM size (in hex)	Help
66000000	CPU CLOCK Frequency	Help
▼ y ▼ - ▼ n	Use msrset/msrcr insns	Help
▼ y ▼ - ▼ n	Hardware Multiplier	Help
▼ y ▼ - ▼ n	Hardware Divider	Help
▼ y ▼ - ▼ n	Hardware Barrel Shift	Help
▼ y ▼ - ▼ n	Instruction Cache	Help
0x80000000	Instruction cache address base	Help
0x8000	Instruction cache byte size	Help
▼ y ▼ - ▼ n	Data Cache	Help
0x80000000	Data cache address base	Help
0x8000	Data cache byte size	Help
◆ y ▼ - ▼ n	Console on UARTLITE	Help
◆ y ▼ m ▼ n	GPIO driver	Help
▼ y ▼ m ◆ n	Ethernet driver	Help
▼ y ▼ m ◆ n	SystemAce driver	Help
◆ y ▼ - ▼ n	Kernel cmdline support (needs new bootloader)	Help
Debug Logging		
◆ y ▼ - ▼ n	Enable debug logging	Help

The “platform” option has to be uclinux-auto. In order to ensure interaction with the running

system, “Console on UARTLITE”<sup>3</sup> has to be enabled as this option ensures and configures the kernel to create a device whereto stdout is pointing.

## General setup



No comments.

---

<sup>3</sup> This is assuming that the option is available. In the uClinux distribution snapshot from 20060511 this was not the case.

## Memory Technology Devices (MTD)

Memory Technology Devices (MTD)				
<input checked="" type="radio"/> y	<input type="radio"/> m	<input type="radio"/> n	Memory Technology Device (MTD) support	Help
<input type="radio"/> y	<input type="radio"/> -	<input checked="" type="radio"/> n	Debugging	Help
0			Debugging verbosity (0 = quiet, 3 = noisy)	Help
<input checked="" type="radio"/> y	<input type="radio"/> m	<input type="radio"/> n	MTD partitioning support	Help
<input type="radio"/> y	<input type="radio"/> m	<input checked="" type="radio"/> n	MTD concatenating support	Help
<input type="radio"/> y	<input type="radio"/> m	<input checked="" type="radio"/> n	RedBoot partition table parsing	Help
<input type="radio"/> y	<input type="radio"/> -	<input checked="" type="radio"/> n	Include unallocated flash space	Help
<input type="radio"/> y	<input type="radio"/> -	<input checked="" type="radio"/> n	Force read-only for RedBoot system images	Help
<input type="radio"/> y	<input type="radio"/> m	<input checked="" type="radio"/> n	uCbootstrap partition table parsing	Help
<input type="radio"/> y	<input type="radio"/> m	<input checked="" type="radio"/> n	Command line partition table parsing	Help
<b>User Modules And Translation Layers</b>				
<input checked="" type="radio"/> y	<input type="radio"/> m	<input type="radio"/> n	Direct char device access to MTD devices	Help
<input checked="" type="radio"/> y	<input type="radio"/> m	<input type="radio"/> n	Caching block device access to MTD devices	Help
<input type="radio"/> y	<input type="radio"/> m	<input checked="" type="radio"/> n	Readonly block device access to MTD devices	Help
<input type="radio"/> y	<input type="radio"/> m	<input checked="" type="radio"/> n	FTL (Flash Translation Layer) support	Help
<input type="radio"/> y	<input type="radio"/> m	<input checked="" type="radio"/> n	NFTL (NAND Flash Translation Layer) support	Help
<input type="radio"/> y	<input type="radio"/> m	<input checked="" type="radio"/> n	INFTL (Inverse NAND Flash Translation Layer) support	Help
<input type="radio"/> y	<input type="radio"/> -	<input type="radio"/> n	Write support for NFTL (BETA)	Help
RAM/ROM/Flash chip drivers				
Mapping drivers for chip access				
Self-contained MTD device drivers				
NAND Flash Device Drivers				
Main Menu		Next		Prev

No comments.



## RAM/ROM/Flash chip drivers

RAM/ROM/Flash chip drivers				
<input checked="" type="checkbox"/> y	<input type="checkbox"/> m	<input type="checkbox"/> n	Detect flash chips by Common Flash Interface (CFI) probe	Help
<input checked="" type="checkbox"/> y	<input type="checkbox"/> m	<input type="checkbox"/> n	Detect JEDEC JESD21c compatible flash chips	Help
<input type="checkbox"/> y	<input type="checkbox"/> -	<input checked="" type="checkbox"/> n	Flash chip driver advanced configuration options	Help
NO	Flash cmd/query data swapping			Help
<input type="checkbox"/> y	<input type="checkbox"/> -	<input type="checkbox"/> n	Specific CFI Flash geometry selection	Help
<input type="checkbox"/> y	<input type="checkbox"/> -	<input type="checkbox"/> n	Support 8-bit buswidth	Help
<input type="checkbox"/> y	<input type="checkbox"/> -	<input type="checkbox"/> n	Support 16-bit buswidth	Help
<input type="checkbox"/> y	<input type="checkbox"/> -	<input type="checkbox"/> n	Support 32-bit buswidth	Help
<input type="checkbox"/> y	<input type="checkbox"/> -	<input type="checkbox"/> n	Support 64-bit buswidth	Help
<input type="checkbox"/> y	<input type="checkbox"/> -	<input type="checkbox"/> n	Support 1-chip flash interleave	Help
<input type="checkbox"/> y	<input type="checkbox"/> -	<input type="checkbox"/> n	Support 2-chip flash interleave	Help
<input type="checkbox"/> y	<input type="checkbox"/> -	<input type="checkbox"/> n	Support 4-chip flash interleave	Help
<input type="checkbox"/> y	<input type="checkbox"/> -	<input type="checkbox"/> n	Support 8-chip flash interleave	Help
<input checked="" type="checkbox"/> y	<input type="checkbox"/> m	<input type="checkbox"/> n	Support for Intel/Sharp flash chips	Help
<input checked="" type="checkbox"/> y	<input type="checkbox"/> m	<input type="checkbox"/> n	Support for AMD/Fujitsu flash chips	Help
<input type="checkbox"/> y	<input type="checkbox"/> m	<input checked="" type="checkbox"/> n	Support for ST (Advanced Architecture) flash chips	Help
<input checked="" type="checkbox"/> y	<input type="checkbox"/> m	<input type="checkbox"/> n	Support for RAM chips in bus mapping	Help
<input checked="" type="checkbox"/> y	<input type="checkbox"/> m	<input type="checkbox"/> n	Support for ROM chips in bus mapping	Help
<input type="checkbox"/> y	<input type="checkbox"/> m	<input checked="" type="checkbox"/> n	Support for absent chips in bus mapping	Help
<input type="checkbox"/> y	<input type="checkbox"/> -	<input checked="" type="checkbox"/> n	Older (theoretically obsoleted now) drivers for non-CFI chips	Help
<input type="checkbox"/> y	<input type="checkbox"/> m	<input checked="" type="checkbox"/> n	AMD compatible flash chip support (non-CFI)	Help
<input type="checkbox"/> y	<input type="checkbox"/> m	<input checked="" type="checkbox"/> n	pre-CFI Sharp chip support	Help
<input type="checkbox"/> y	<input type="checkbox"/> m	<input checked="" type="checkbox"/> n	JEDEC device support	Help
<input type="checkbox"/> y	<input type="checkbox"/> m	<input checked="" type="checkbox"/> n	ST PSD4256G compatible flash chip support	Help

No comments.

## Mapping drivers for chip access

Mapping drivers for chip access

<input type="checkbox"/> y	<input type="checkbox"/> m	<input checked="" type="checkbox"/> n	CFI Flash device in physical memory map	Help
0x8000000			Physical start address of flash mapping	Help
0x4000000			Physical length of flash mapping	Help
2			Bus width in octets	Help
<input type="checkbox"/> y	<input type="checkbox"/> m	<input checked="" type="checkbox"/> n	CFI Flash device mapped by Arcturus uCbootstrap	Help
<input type="checkbox"/> y	<input type="checkbox"/> m	<input checked="" type="checkbox"/> n	CFI Flash device mapped on Dragonix VZ	Help
<input type="checkbox"/> y	<input type="checkbox"/> m	<input checked="" type="checkbox"/> n	CFI Flash device mapped on SnapGear/SecureEdge	Help
<input type="checkbox"/> y	<input type="checkbox"/> m	<input checked="" type="checkbox"/> n	CFI Flash device mapped on SnapGear/GEODE boards	Help
<input type="checkbox"/> y	<input type="checkbox"/> m	<input checked="" type="checkbox"/> n	CFI Flash device mapped on SnapGear/SecureEdge (uClinux)	Help
8MB			FLASH size	Help
<input type="checkbox"/> y	<input type="checkbox"/> m	<input checked="" type="checkbox"/> n	CFI Flash device mapped on Microblaze/mbvanilla (uClinux)	Help
8MB			FLASH size	Help
<input checked="" type="checkbox"/> y	<input type="checkbox"/> m	<input type="checkbox"/> n	CFI Flash device mapped on MicroBlaze auto-config (uClinux)	Help
8MB			FLASH size	Help
<input type="checkbox"/> y	<input type="checkbox"/> m	<input checked="" type="checkbox"/> n	CFI Flash device mapped on Microblaze/ML401 (uClinux)	Help
8MB			FLASH size	Help
<input type="checkbox"/> y	<input type="checkbox"/> m	<input checked="" type="checkbox"/> n	CFI Flash device mapped on Microblaze/SUZAKU (uClinux)	Help
<input type="checkbox"/> y	<input type="checkbox"/> m	<input checked="" type="checkbox"/> n	CFI Flash device mapped on Key Technology devices	Help
<input type="checkbox"/> y	<input type="checkbox"/> m	<input checked="" type="checkbox"/> n	CFI Flash device mapped on SED SIOS III Core	Help
<input type="checkbox"/> y	<input type="checkbox"/> m	<input checked="" type="checkbox"/> n	CFI Flash device mapped on Motorola M5272C3	Help
<input type="checkbox"/> y	<input type="checkbox"/> m	<input checked="" type="checkbox"/> n	CFI Flash device mapped on Motorola M5208	Help
<input type="checkbox"/> y	<input type="checkbox"/> m	<input checked="" type="checkbox"/> n	CFI Flash device mapped on SNEHA CPU16B	Help
<input checked="" type="checkbox"/> y	<input type="checkbox"/> m	<input type="checkbox"/> n	Generic uClinux RAM/ROM filesystem support	Help
16MB			FLASH size	Help
<input type="checkbox"/> y	<input type="checkbox"/> m	<input checked="" type="checkbox"/> n	PCI MTD driver	Help
<input type="checkbox"/> y	<input type="checkbox"/> m	<input checked="" type="checkbox"/> n	PCMCIA MTD driver	Help

OK Next Prev

No comments.

## Block devices

Block devices				
<input type="checkbox"/> y	<input type="checkbox"/> m	<input type="checkbox"/> n	Promise SATA SX8 support	Help
<input checked="" type="checkbox"/> y	<input type="checkbox"/> m	<input type="checkbox"/> n	Loopback device support	Help
<input type="checkbox"/> y	<input type="checkbox"/> m	<input type="checkbox"/> n	Network block device support	Help
<input checked="" type="checkbox"/> y	<input type="checkbox"/> m	<input type="checkbox"/> n	RAM disk support	Help
4096			Default RAM disk size	Help
<input type="checkbox"/> y	<input type="checkbox"/> -	<input checked="" type="checkbox"/> n	Initial RAM disk (initrd) support	Help
<input type="checkbox"/> y	<input type="checkbox"/> -	<input type="checkbox"/> n	RAM disk data block compiled in	Help
<input type="checkbox"/> y	<input type="checkbox"/> m	<input checked="" type="checkbox"/> n	ROM disk memory block device (blkmem)	Help
NONE			FLASH type	Help
8MB			FLASH size	Help
AUTO			FLASH bit width	Help
<input type="checkbox"/> y	<input type="checkbox"/> -	<input type="checkbox"/> n	2nd flash in /dev/rom8	Help
<input checked="" type="checkbox"/> y	<input type="checkbox"/> -	<input type="checkbox"/> n	Per partition statistics in /proc/partitions	Help

No comments.

## File systems

Option	Yes (y)	Module (m)	No (n)	Description	Help
	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	OS/2 HPFS file system support	Help
	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	/proc file system support	Help
4096				Number of inodes in any given PROC directory	Help
	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	/dev file system support (EXPERIMENTAL)	Help
	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	Automatically mount at boot	Help
	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	Debug devfs	Help
	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	/dev/pts file system for Unix98 PTYs	Help
	<input type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>	QNX4 file system support (read only)	Help
	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	QNX4FS write support (DANGEROUS)	Help
	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	ROM file system support	Help
	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	Second extended fs support	Help
	<input type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>	System V/Xenix/V7/Coherent file system support	Help
	<input type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>	JDE file system support (read only)	Help

Main Menu      Next      Prev

No comments.

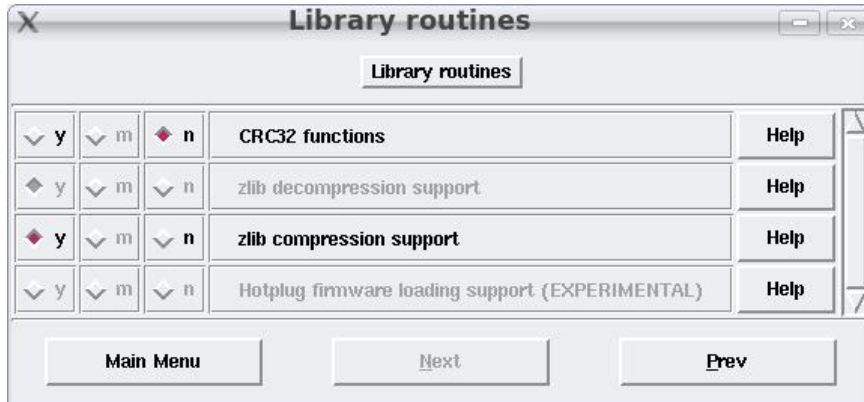
## Kernel hacking

Option	Yes (y)	Module (m)	No (n)	Description	Help
	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	Full Symbolic/Source Debugging support	Help
	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	Magic SysRq key	Help
	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	Kernel profiling support	Help
2				Profile shift count	Help
	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Use fast second timer for profiling	Help
	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	Suppress Kernel BUG Messages	Help

Main Menu      Next      Prev

No comments.

## Library routines



No comments.

When done: "save and exit".

## Customizing the User programmes

When the important changes to the kernel have been carried out, we finally have to modify certain options for the User programmes.

In the process of inspecting and configuring these options, you will observe that several programmes can be found both in Busybox menu and in the scatter in the other menus. Some preliminary experience resulted in programmes unable to compile if they were not Busybox programmes. The question is therefore which context is the most appropriate and why? This has not been ascertained at this point in time.

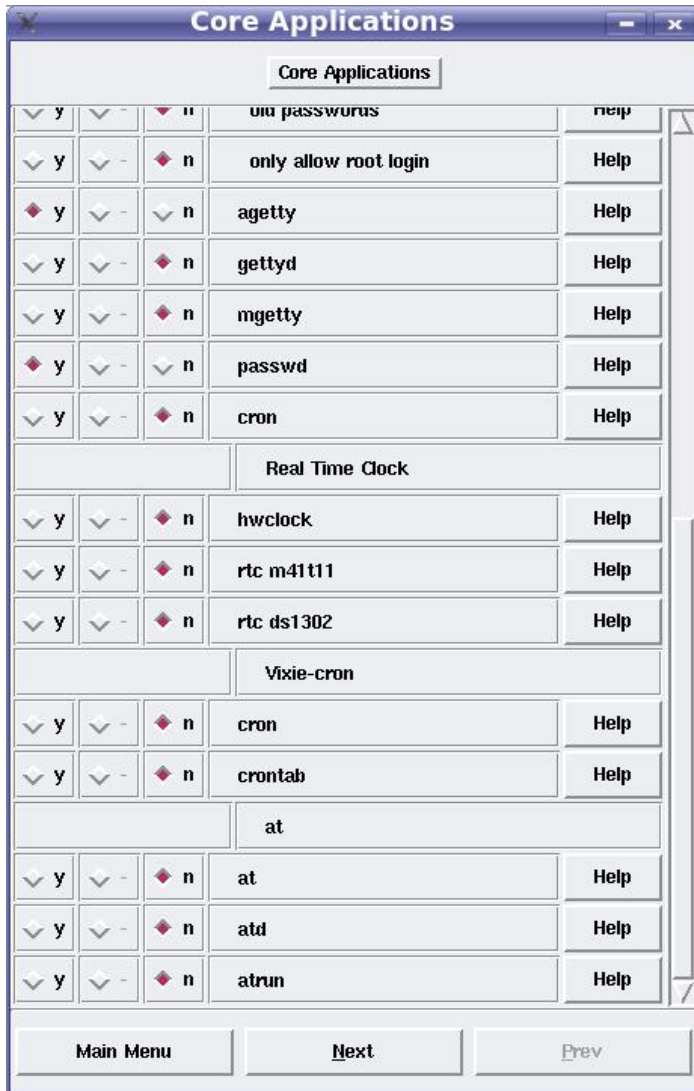
## Core application



When inspecting the default options set for this category, it was determined by trial and error that it is not possible to have both the options “agetty<sup>4</sup>” and “enable console shell”. You must choose one or the other but not both. Also note that the option “login” is necessary if “agetty” should work.

---

<sup>4</sup> Serial console support



No comments.

## Library Configuration



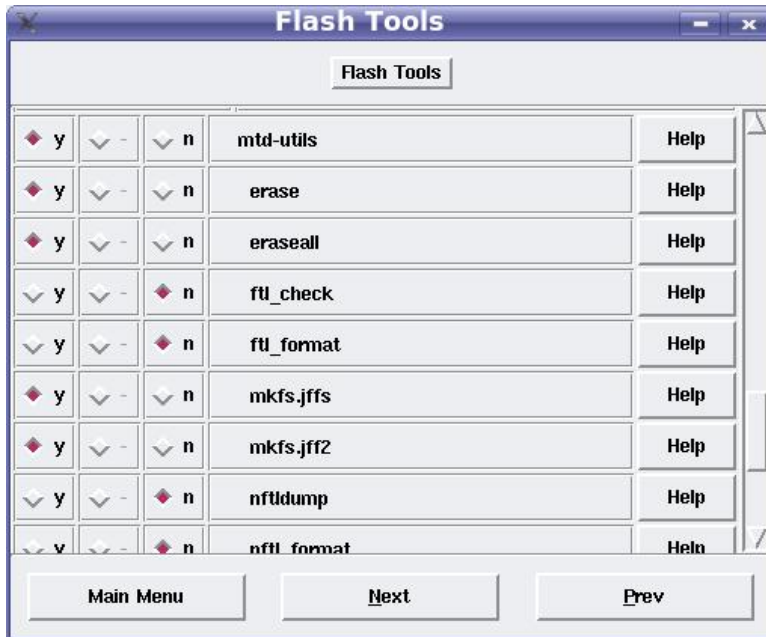
No comments.

## Flash Tools



No comments.





As far as I can determine, MTD utilities should be enabled in order to activate flatfsd<sup>5</sup>. It has, however, not been examined thoroughly.

---

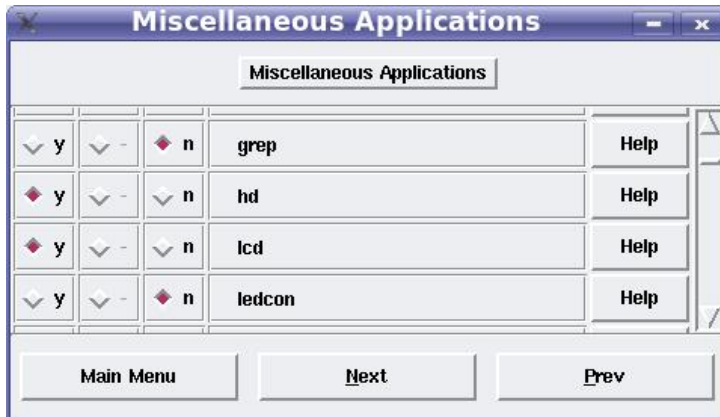
<sup>5</sup> Programme enabling to write configuration smoothly to flash[FLASH]

# Filesystem Applications

Selection	Description	Help
<input checked="" type="checkbox"/> y	flatfsd	Help
<input type="checkbox"/> Disklike	flatfsd storage type	Help
<input checked="" type="checkbox"/> y	Compress /etc/config to flash as required	Help
<input type="checkbox"/> y	Device has Real Time Clock	Help
<input type="checkbox"/> y	mount	Help
<input type="checkbox"/> y	umount	Help
<input type="checkbox"/> y	fdisk	Help
<input type="checkbox"/> y	sfdisk	Help
EXT2		
<input checked="" type="checkbox"/> y	e2fsck	Help
<input checked="" type="checkbox"/> y	mke2fs	Help
<input type="checkbox"/> y	badblocks	Help
<input checked="" type="checkbox"/> y	chattr	Help
<input type="checkbox"/> y	dumpe2fs	Help
<input type="checkbox"/> y	e2label	Help
<input checked="" type="checkbox"/> y	fsck	Help
<input type="checkbox"/> y	lsattr	Help

No comments.

## Miscellaneous Applications



No comments.



No comments.



No comments.

# Busybox



This option has to be enabled to enable Busybox.



No comments.



No comments.



No comments.



No comments.



No comments.

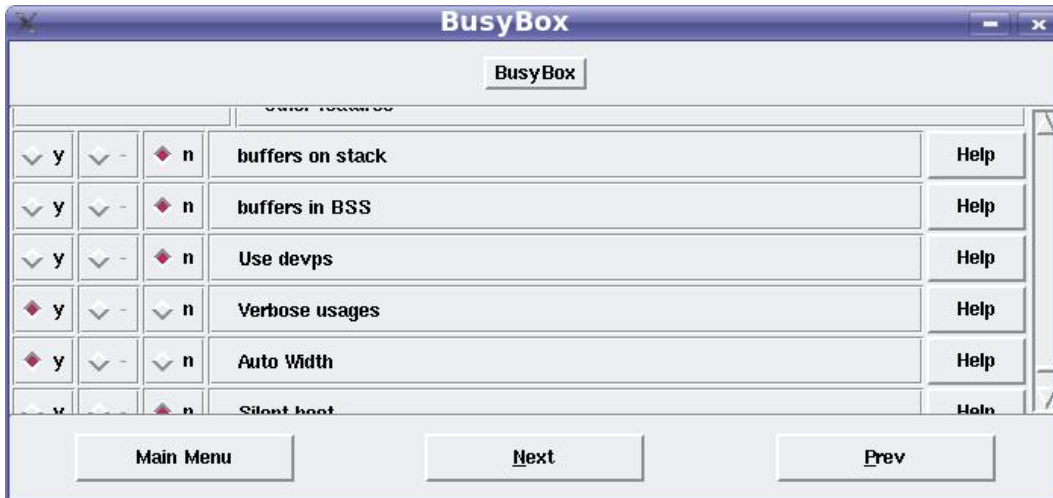


No comments.



No comments.





No comments.



No comments.

## Building the image

### The Image itself

At this point we are ready to make the image itself. It is a dual phase procedure involving the execution of two different commands.

The first checks and creates dependencies:

```
make dep6
```

The second builds the kernel along with the selected programmes and libraries:

```
make all
```

If the make process is successful, you can find the image in the directory “images” with the name image.bin. The define appended ensures that the created image is not copied to /tftpboot.

<sup>6</sup> It is assumed that the current path is the root path of the uClinux distribution.

## Common errors

### Compilation failure due to missing zlib.h

*Where:*

This is typically seen when compiling cramfs.

*Solution:*

For some unknown reason, the uClinux distribution depends on a file which is not within itself. This is obviously not a sound approach. Until the problem is solved it can be circumvented by installing a package onto your Linux distribution. For Debian and Ubuntu, the package in question is `zlib1g-dev`.

### Error, cannot copy image to /tftpboot

*Where:*

Last lines of the distribution compilation.

*Solution:*

There are two solutions; in the event there is a desire to have the image copied to `tftpboot`, it is then a simple matter of creating the directory in question. However, if this is not the case, simply append `"NO_BUILD_INTO_TFTPBOOT=n"` to `"make all"`. The compilation command thus becomes: `make all NO_BUILD_INTO_TFTPBOOT=n`

### Network dependent programmes and related problems

*Where:*

In this howto where network support has been removed from the kernel, certain network dependent programmes failed to compile.

*Solution:*

In this situation it was a question of determining in which context the errors were to be found and then remove them from the compilation.

The important lesson is that there may be dependencies which for no obvious reason at all result in programmes failing to compile but are relatively easy to find.

## Downloading and booting the image

The "normal" approach is carried out via the Xilinx Platform Studio, see [HOWTO-1] for a description.

### Accelerated version

In order to speed up the process of downloading bit, image and starting the application, the following scripts can be applied. Simply copy these lines into two separate files where one is a batch file and the other is `tcl`<sup>7</sup> file. These files should then be placed in the root of your project.

The batch file:

```
echo "Downloading bit"
xbash -q -c "cd /cygdrive/z/DemoKit/demo_uc_1/; /usr/bin/make -f system.make
download; exit;"
echo "Downloading image and starting it"
xmd -tcl run.tcl
```

---

<sup>7</sup> Assuming that Xilinx Platform Studio is installed with `tcl` support which is presumed to be standard.

The run.tcl file:

```
xload xmp system.xmp
connect mb mdm
stop
rst
dow -data images/image.bin 0x22000000
con 0x22000000
```

The above 0x22000000 is assuming that the start address of your image is placed here.

## Saving and retrieving a given setup

### Saving the current setup

Once you have configured your system and spent considerable time on making it work the way you want, the need to save the system configuration arises.

The whole configuration of a given image is placed in five different files<sup>8</sup>:

- linux-2.4.x/arch/microblaze/platform/uclinux-auto/auto-config.in
  - The HW configuration
- .config
  - The configuration describing the overall hardware layout. Addresses, port etc.
- linux-2.4.x/.config
  - The kernel configuration file
- config/.config
  - The Vendor/User setting configuration file. Note also that when you add your own files, the option for activating them is found in this file.
- uClib/.config
  - Saving the last file is deemed optional in most cases.

After copying these files, you can safely delete the rest or simply add them to your file versioning system of your choice.

### Retrieving a setup

At some point you may wish to create the image again or to adjust some simple options to suite new needs.

To recreate the image and files saved in the previous section, the following awkward steps must be taken in sequence.

1. First change directory to the distribution.
2. Run cmd: "make mrproper"
  - a. Cleans up the distribution, everything is cleaned.
3. Copy distribution configuration to ./config
4. Run cmd: ./config/setconfig defaults
  - a. Sets options to some predefined value. In some instances, however, user intervention is required and a suitable choice would be to press enter.

---

<sup>8</sup> It is assumed that the current path is the root path of the uClinux distribution.

- b. If you wish to speed things up, you can extend the command. The extended version just accepts everything, just like you would by pressing enter.
      - i. Run cmd: "yes ""|./config/setconfig defaults"
5. Overwrite Vendor/User settings config/.config file with your saved version.
6. Overwrite the kernel configuration file linux-2.4.x/.config with your saved version.
7. Overwrite uClibc/.config configuration file with your saved version.
8. The next step is to inform the system that we wish to use an old version of the configuration files. This is done by running the command: "make oldconfig"
  - a. Notice again that you "may" have to answer some questions. In the normal scenario they can just be answered by pressing the return button.
  - b. Yet again the process can be accelerated.
    - i. Run cmd: "yes ""| make oldconfig"
9. Finally we are ready to start building the kernel. As explained earlier, it is performed firstly by running "make dep" and lastly by running "make all" which creates the image.
10. The image can now be copied to your Xilinx project in the project

## Adding a user application

Certain steps must be taken in the event a user application is developed and should run on the target platform. Below is shown the entire howto with explanations of how it is done. The text has been pasted from the file Documentation/Adding-User-Apps-HOWTO which you can find in your distribution.

*Adding User Applications to the uClinux Distribution*

*D. P. Siddons  
9th Dec. 2001*

*This document gives simple instructions for adding a user-written application to the uClinux configuration system. Entries must be added to three files, and an appropriate Makefile must exist in the user application source directory, which must be put in user (all directory names here are given relative to the uClinux top directory. In my system this is /home/peter/uClinux-dist).*

*Files to edit:*

*user/Makefile*

*Add a line to the file like*

```
dir_$(CONFIG_USER_FOO_FOO)           += foo
```

*This adds the directory 'foo' to the list of directories to be built. I added mine in alphabetical order. The order doesn't seem to matter.*

*config/Configure.help*

*This file contains the text which is presented on request during the config.*

*Add a block like*

```
CONFIG_USER_FOO_FOO
```

*This program does foey things to your bars.*

The text must be indented two spaces, and there must be no empty lines. Lines should be <70 chars long.

config/config.in:

Add a line in the appropriate menu section (i.e. in the program group you want your app to show up in during 'make config'; I used 'misc'), like

```
bool 'foo'          CONFIG_USER_FOO_FOO
```

The repetition of FOO allows for directories which contain multiple executables. Thus, if the user directory 'foo' contained code to make 'foo' and 'bar', each gets its own config line if an additional entry is made like

```
bool 'bar'          CONFIG_USER_FOO_BAR
```

Next, there needs to be a proper /user/foo/Makefile. The Makefile should follow the following template:

```
-----  
EXEC = foo  
OBJS = foo.o  
  
all: $(EXEC)  
  
$(EXEC): $(OBJS)  
    $(CC) $(LDFLAGS) -o $@ $(OBJS) $(LDLIBS)  
  
romfs:  
    $(ROMFSINST)    /bin/$(EXEC)  
  
clean:  
    -rm -f $(EXEC) *.elf *.gdb *.o  
-----
```

If more than one executable is built in the foo directory, as above, then the Makefile should look like

```
-----  
EXECS = foo bar  
OBJS = foo.o bar.o  
  
all: $(EXECS)  
  
$(EXECS): $(OBJS)  
    $(CC) $(LDFLAGS) -o $@ $@.o $(LDLIBS)  
  
romfs:  
    $(ROMFSINST) -e CONFIG_USER_FOO_FOO          /bin/foo  
    $(ROMFSINST) -e CONFIG_USER_FOO_BAR          /bin/bar  
-----
```

More complex makefiles are of course possible. The reader is encouraged to browse the user tree for examples.

When all this is set up, doing the standard 'make xconfig; make dep; make' should build the app and install it in romfs and hence in the target system image.bin.

## References

[HOWTO-1] “Howto create a project for a simple uClinux ready MicroBlaze 4.0 design on XPS (Xilinx Platform Studio) for Spartan-3E” <http://www.teknologisk.dk/20356>

[FLASH] uCdot - “Using flatfsd to save persistent state”  
<http://www.ucdot.org/article.pl?sid=04/01/18/2312200&mode=thread>