

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

1
1
2
2
2
3
4
3
5
5
7
8
0

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 (<u>http://www.petalogix.com/resources/downloads/mb-gcc</u>) 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 <u>http://www.uclinux.org</u> 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: <u>http://www.uclinux.org/pub/uClinux/dist/</u>.

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 dos2unix¹. 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.

¹ 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 ways²; 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

Vendor/Product Selection	Save and Exit
Kernel/Library/Defaults Selection	Quit Without Saving
	Load Configuration from File

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

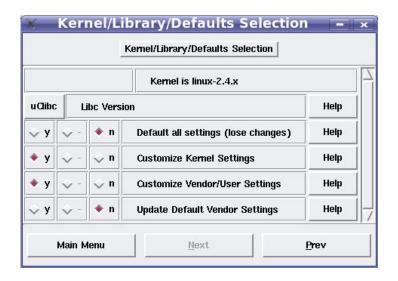
		Vendor/Product Selection	
		Select the Vendor you	wish to target
Xilinx	Vendor		Help
		Select the Product you	u wish to target
PalmIIIx	3com Pro	oducts	Help
DPCM	Actiontec F	Products	Help
Covote	ADI Produ	ints	Help

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

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

	Vendor/Product Select	tion
rsc	TeleTP Products	нер
dm270	TI Products	Help
A7DB	Triscend Products	Help
EPIA	Via Products	Help
SM2010	Weiss Products	Help
uclinux-	auto Xilinx Products	Help

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

	Code maturity level options		
🔶 y 😓 - 🔯 n Prompt	for development and/or incomple	te code/drivers	Help
Main Menu	Next	Pre	<i></i>

Normally you just say yes.

Loadable module support

			Loadable module support	
💸 у	÷ -	◆ n	Enable loadable module support	Help
¢у	÷-	\diamondsuit n	Set version information on all module symbols	Help
¢ y	÷-	↓ n	Kernel module loader	Help

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	
uclinu:	x-auto	Plat	form	Help
RAM	Ме	mory m	odel	Help
0008x0	0000		ROM base address (in hex)	Help
)×0010	0000		ROM size (in hex)	Help
860000	00		CPU CLOCK Frequency	Help
vу	÷ -	$\hat{\mathbf{v}}^{n}$	Use msrset/msrclr insns	Help
V Y	÷ -	\$ n	Hardware Multiplier	Help
ŵУ	÷ -	$\hat{\mathbf{v}}^{n}$	Hardware Divider	Help
\$ У	÷ -	\diamond n	Hardware Barrel Shift	Help
¢у	÷ -	🗢 n	Instruction Cache	Help
0008x0	0000		Instruction cache address base	Help
)×8000			Instruction cache byte size	Help
\$ У	÷ -	\diamond n	Data Cache	Help
0×8000	0000		Data cache address base	Help
0x8000			Data cache byte size	Help
♦ y	÷ -	🗢 n	Console on UARTLITE	Help
♦ у	∲ m	💠 n	GPIO driver	Help
💸 у	💠 m	🔶 n	Ethernet driver	Help
⇒ у	🔶 m	🔶 n	SystemAce driver	Help
 у 	÷ -	🗢 n	Kernel cmdline support (needs new bootloader)	Help
			Debug Logging	
♦ у	~-	🕹 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"³ has to be enabled as this option ensures and configures the kernel to create a device whereto stdout is pointing.

General setup

			General setup	
ф у	÷-	◆ n	PCI support	Help
\$ у	÷-	🔹 n	Networking support	Help
≎ у	÷-	🔶 n	Disk hardware support	Help
ý y	÷-	♀ n	PCI device name database	Help
⇒ у	÷-	🔹 n	Support for hot-pluggable devices	Help
			PCMCIA/CardBus support	
⇒ у	÷-	🔶 n	System V IPC	Help
⇒ у	÷-	🔶 n	BSD Process Accounting	Help
≎ у	÷-	🔶 n	Syscil support	Help
ELF	Ken	nel core	(/proc/kcore) format	Help
◆ у	🔷 m	💠 n	Kernel support for flat binaries	Help
🔶 у	÷-	💠 n	Enable ZFLAT support	Help
≎ у	÷-	🔹 n	Non power-of-2 kernel allocator (EXPERIMENTAL)	Help
¢у	÷-	• n	include /proc/mem_map	Help
\$ у	÷-	🔶 n	Allow allocating large blocks (> 1MB) of memory	Help
	Main	Menu	Next Pro	ev

 $^{^3}$ This is assuming that the option is available. In the uClinux distribution snapshot from 20060511 this was not the case.

Sec.		Me	mory Technology Devices (MTD) Memory Technology Devices (MTD)	
♦ y	🕹 m	🤝 n	Memory Technology Device (MTD) support	Help
\$ у	÷-	◆ n	Debugging	Help
0	<u>.</u>	<u></u> _	Debugging verbosity (0 = quiet, 3 = noisy)	Help
🔶 у	🖍 m	💠 n	MTD partitioning support	Help
⇒ у	🕹 m	🔶 n	MTD concatenating support	Help
⇒ у	🔷 m	🔹 n	RedBoot partition table parsing	Help
¢у	÷-	◆ n	Include unallocated flash space	Help
у у	÷-	• n	Force read-only for RedBoot system images	Help
⇒ y	🗢 m	🔹 n	uCbootstrap partition table parsing	Help
⇒ у	🕹 m	🔶 n	Command line partition table parsing	Help
			User Modules And Translation Layers	
🔶 у	文 m	🗢 n	Direct char device access to MTD devices	Help
🔶 y	ф m	🗢 n	Caching block device access to MTD devices	Help
¢у	\$ m	• n	Readonly block device access to MTD devices	Help
⇒ у	🔷 m	🔶 n	FTL (Flash Translation Layer) support	Help
⇒ у	🗢 m	🔶 n	NFTL (NAND Flash Translation Layer) support	Help
≎ у	🔷 m	🔹 n	INFTL (Inverse NAND Flash Translation Layer) support	Help
ŵу	÷-	\diamond 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	
	Mai	n Menu	Next Prev	20

Memory Technology Devices (MTD)

RAM/ROM/Flash chip drivers

y	i v m∣	v n	Detect flash chips by Common Flash Interface (CFI) probe	Help
• y	v m	v n	Detect JEDEC JESD21c compatible flash chips	Help
y y	× -	◆ n	Flash chip driver advanced configuration options	Help
NO	Flas	h cmd/qu	iery data swapping	Help
~ y	- V	<u>v</u> n	Specific CFI Flash geometry selection	Help
у у	v -	✓ n	Support 8-bit buswidth	Help
v y	v -	v n	Support 16-bit buswidth	Help
v y	Ŷ-	↓ n	Support 32-bit buswidth	Help
✓ y	÷-	\$ n	Support 64-bit buswidth	Help
\$ у	÷ -	◇ n	Support 1-chip flash interleave	Help
v y	Ŷ-	🕆 n	Support 2-chip flash interleave	Help
v y	÷-	◇ n	Support 4-chip flash interleave	Help
¢ y	÷ -	\sim ⁿ	Support 8-chip flash interleave	Help
♦ у	🔷 m	💠 n	Support for Intel/Sharp flash chips	Help
♦ y	🔷 m	🗢 n	Support for AMD/Fujitsu flash chips	Help
⇒ у	🔶 m	🔶 n	Support for ST (Advanced Architecture) flash chips	Help
◆ у	🕹 m	🗢 n	Support for RAM chips in bus mapping	Help
♦ у	🔷 m	💠 n	Support for ROM chips in bus mapping	Help
⇒ у	🔷 m	🔷 n	Support for absent chips in bus mapping	Help
⇒ у	÷ -	🔶 n	Older (theoretically obsoleted now) drivers for non-CFI chips	Help
¢у	◇ m	• n	AMD compatible flash chip support (non-CFI)	Help
¢у	◇ m	• n	pre-CFI Sharp chip support	Help
у у	◇ m	◆ n	JEDEC device support	Help
V Y	🔶 m	• n	ST PSD4256G compatible flash chip support	Help

Κ,			Mapping drivers for chip access Mapping drivers for chip access	
≎ y	💸 m	◆ n	CFI Flash device in physical memory map	Help
×8000	0000		Physical start address of flash mapping	Help
)×4000	0000		Physical length of flash mapping	Help
2			Bus width in octets	Help
ŵ у	🕹 m	🔹 n	CFI Flash device mapped by Arcturus uCbootstrap	Help
\$ у	💠 m	🔶 n	CFI Flash device mapped on Dragonix VZ	Help
⇒ у	🔷 m	🔶 n	CFI Flash device mapped on SnapGear/SecureEdge	Help
≎ у	◇ m	🔶 n	CFI Flash device mapped on SnapGear/GEODE boards	Help
≎ у	🕹 m	◆ n	CFI Flash device mapped on SnapGear/SecureEdge (uClinux)	Help
8MB	F	LASH si	ze	Help
🕹 у	🔷 m	🔶 n	CFI Flash device mapped on Microblaze/mbvanilla (uClinux)	Help
8MB	F	LASH si	ze	Help
♦ у	🕹 m	🤝 n	CFI Flash device mapped on MicroBlaze auto-config (uClinux)	Help
8MB	F	LASH si	ze	Help
V Y	🔶 m	♦ n	CFI Flash device mapped on Microblaze/ML401 (uClinux)	Help
8MB	F	LASH si	ze	Help
¢У	🕹 m	• n	CFI Flash device mapped on Microblaze/SUZAKU (uClinux)	Help
⇒ у	\$ m	🔶 n	CFI Flash device mapped on Key Technology devices	Help
⇒ у	◇ m	🔶 n	CFI Flash device mapped on SED SIOS III Core	Help
⇒ у	◇ ^m	🔶 n	CFI Flash device mapped on Motorola M5272C3	Help
<i></i> у	≎ m	🔹 n	CFI Flash device mapped on Motorola M5208	Help
⇒ у	\$ m	🔶 n	CFI Flash device mapped on SNEHA CPU16B	Help
♦ у	◇ m	💠 n	Generic uClinux RAM/ROM filesystem support	Help
16MB		FLASH s	size	Help
ŵ у	Ý m	• n	PCI MTD driver	Help
¢ у	◇ m	• n	PCMCIA MTD driver	Help
				1
		ок	<u>N</u> ext <u>P</u> rev	

Mapping drivers for chip access

Block devices

		Block devices	
		Block devices	
✓ m	• n	Promise SATA SX8 support	Help
◇ m	◇ n	Loopback device support	Help
🗸 m	♦ n	Network block device support	Help
\$ m	💠 n	RAM disk support	Help
,		Default RAM disk size	Help
Ŷ -	n	Initial RAM disk (initrd) support	Help
÷ -	♦ n	RAM disk data block compiled in	Help
🗢 m	◆ n	ROM disk memory block device (blkmem)	Help
ŀ	LASH t	Abe	Help
FL	ASH siz	8	Help
F	LASH b	it width	Help
× -	\$ n	2nd flash in /dev/rom8	Help
ŷ -	💠 n	Per partition statistics in /proc/partitions	Help
	lain Ma	u Next	Prev
		m m n m m n m m n m n n i - i m - n i - i m i n i FLASH trip FLASH siz FLASH biz i - i i - i i - i i - i	Image: model of the second

File systems

⇔ y ↓ m ♦ n	0.9/2 LIDES file quetem cumment	Help
y y w	OS/2 HPFS file system support	neih
◆ y 💠 - 💠 n	/proc file system support	Help
4096	Number of inodes in any given PROC directory	Help
💸 y 🔹 - 🔶 n	/dev file system support (EXPERIMENTAL)	Help
ŵ y ŵ - ♠ n	Automatically mount at boot	Help
👽 y 🔤 🔶 n	Debug devfs	Help
ý y	/dev/pts file system for Unix98 PTYs	Help
◇ y ◇ m ◆ n	QNX4 file system support (read only)	Help
ŵ y ŵ - ♠ n	QNX4FS write support (DANGEROUS)	Help
🔶 y 🔯 💠 n	ROM file system support	Help
🔶 y 🔯 🐝 m	Second extended fs support	Help
↓ y ↓ m ▲ n	System V/Xenix/V7/Coherent file system support	Help
🗸 v 🖾 m l 🔶 n	UDE file system sunnort (read only)	Hein

No comments.

Kernel hacking

			Kernel hacking	
🔶 у	÷ -	🗢 n	Full Symbolic/Source Debugging support	Help
⇒ у	÷-	🔶 n	Magic SysRq key	Help
🕹 у	÷-	🔶 n	Kernel profiling support	Help
2			Profile shift count	Help
÷У	÷-	\sim n	Use fast second timer for profiling	Help
⇒ у	÷-	🔹 n	Suppress Kernel BUG Messages	Help

Library routines

		Library routines	
\$ m	◆ n	CRC32 functions	Help
◇ m	◇ n	zlib decompression support	Help
🔶 m	💠 n	zlib compression support	Help
∽ m	🗢 n	Hotplug firmware loading support (EXPERIMENTAL)	Help
		v m v n v m v n	m n CRC32 functions m n zlib decompression support m n zlib compression support

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

Core Applications					
			Core Applications		
♦ у	÷ -	🗢 n	init	Help	
\$ У	÷-	🔶 n	enable console shell	Help	
💸 у	v -	🔷 n	execute firewall rules	Help	
🗢 у	÷ -	◆ n	process init.conf	Help	
minix-shell Shell Program		Help			
¢у	÷-	◇ n	simple history (sash)	Help	
у у	÷-	\$ n	sash ps	Help	
⇒ у	÷-	🔶 n	reboot (sash)	Help	
≎ у	÷ -	🔶 n	SnapGear reboot script	Help	
⇒ у	÷-	🔶 n	shutdown (sash)	Help	
🗢 У	÷-	🔶 n	expand	Help	
~ У	÷ -	\$ n	expand should not write zeroes	Help	
◆ у	÷ -	🗢 n	version	Help	
♦ у	÷-	💠 n	login	Help	
💸 у	÷-	🔶 n	old passwords	Help	
~ у	÷-	🔶 n	only allow root login	Help	
 у 	÷-	🗢 n	agetty	Help	
<u>.</u> v		🔺 n 🛛	nottud	Holn	

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 "agetty⁴" 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.

⁴ Serial console support

	_	C	ore Applications	<u>e</u>
			Core Applications	
<i>y y</i>	× -	•	uu passwurus	пен
> y	Ŷ -	🔶 n	only allow root login	Help
) y	÷-	◇ n	agetty	Help
> у	÷-	🔶 n	gettyd	Help
> у	÷-	🔶 n	mgetty	Help
▶ y	Ŷ -	🗢 n	passwd	Help
> y	÷ -	🔶 n	cron	Help
			Real Time Clock	
> у	÷ -	🔶 n	hwclock	Help
> у	÷ -	🔶 n	rtc m41t11	Help
> y	÷ -	🔶 n	rtc ds1302	Help
			Vixie-cron	
> у	÷ -	🔶 n	cron	Help
> у	÷ •	🔶 n	crontab	Help
			at	
> у	÷-	🔶 n	at	Help
> у	÷-	🔶 n	atd	Help
> y	÷-	◆ n	atrun	Help
	Main M		Next	Prev

Library Configuration

			Library Configuration	
¢у	÷ -	\sim n	PAM TACACS+	Help
⇒ у	÷-	🔹 n	Build libPCAP	Help
⇒ у	÷-	🔶 n	Build libPCRE	Help
🔶 у	÷ -	💠 n	Build libZ	Help
ф у	÷ -	🔶 n	Build libBZIP2	Help
⇒ у	÷ -	🔶 n	Build libATM	Help
⇒ у	÷ -	🔷 n	Build libpng	Help
⇒ у	÷ -	🔶 n	Build libjpeg	Help
\$ у	÷-	🔶 n	Build libupnp	Help
	Main M	lenu	Next	Prev

No comments.

Flash Tools

♦ y	v-	💠 n	flashw	Help
⇒ у	÷ -	◆ n	netflash	Help
¢ y	÷-	\$ n	flash	Help
с у	÷-	🗢 n	netflash uses FTP	Help

			Flash Tools	
♦ у	÷-	💠 n	mtd-utils	Help
♦ у	÷-	🗢 n	erase	Help
+ у	÷-	🗢 n	eraseali	Help
⇒ у	Ŷ-	🔶 n	ftl_check	Help
\$ у	÷-	🔹 n	ftl_format	Help
◆ у	÷-	💠 n	mkfs.jffs	Help
+ у	÷-	💠 n	mkfs.jff2	Help
⇒ у	Ŷ -	🔶 n	nftidump	Help
⇒ v	- N	🔹 n 🛛	nftl format	Help
	Main M	denu	Next	Prev

As far as I can determine, MTD utilities should be enabled in order to activate flatfsd⁵. It has, however, not been examined thoroughly.

⁵ Programme enabling to write configuration smoothly to flash[FLASH]

Filesystem Applications

			Filesystem Applications	
♦ y	÷ -	💠 n	flatfsd	Help
Diskli	ke	flatfsd	storage type	Help
♦ у	÷ -	💠 n	Compress /etc/config to flash as required	Help
ŵ У	÷.	🔶 n	Device has Real Time Clock	Help
≎ у	Ŷ -	🔶 n	mount	Help
\$ у	÷-	🔶 n	umount	Help
\$ у	÷ -	🔶 n	fdisk	Help
⇒ у	÷-	🔶 n	sfdisk	Help
			EXT2	
♦ у	÷ -	💠 n	e2fsck	Help
♦ у	÷ -	🗢 n	mke2fs	Help
⇒ у	÷.	🔶 n	badblocks	Help
♦ y	Ŷ -	🗢 n	chattr	Help
\$ у	Ŷ-	🔹 n	dumpe2fs	Help
💸 у	÷-	🔶 n	e2label	Help
🔶 y	÷ -	💠 n	fsck	Help
≎ у	Ŷ -	🔶 n	Isattr	Help
	Main	Menu	Next	<u>P</u> rev

Miscellaneous Applications

			Miscellaneous Applications	
🗢 у	÷ -	🔶 n	grep	Help
♦ y	Ŷ-	💠 n	hd	Help
♦ y	Ŷ -	💠 n	lcd	Help
≎ у	÷-	🔶 n	ledcon	Help

No comments.

Miscellaneous Applicatio	ons
	Help
🕈 y 🔯 - 🔯 n grep	Help
ŷy ŷ - ♦ n I	Help
♦ y 👽 - 🔯 n 🛛 In	Help
ýy ý - ♦ n Is	Help
🕈 y 🔯 - 🔯 n 🛛 mkdir	Help
💸 y 👽 - 🔶 n 🛛 mkfifo	Help
🔆 y 🔍 - 🔶 n mknod	Help
ŷy ŷ- ♦ n more	Help
◆ y 😳 - 😓 n mv	Help
◆ y 😳 - 🔯 n 🛛 m	Help
🔆 y 🔍 - 🔶 n mdir	Help
ÿy v - ♦ n sync	Help
◆ y 😳 - 😓 n touch	Help
Shutils tools	

		IISCE	Ilaneous Applications Miscellaneous Applications	
♦ y	~ -	💸 n 🛛	basename	Help
▶ y	÷-		date	Help
⇒ у	ŵ -	* n	dimame	Help
▶ y	÷-	🕹 n	echo	Help
♦ y	Ŷ-	💠 n	false	Help
> у	÷-	◆ n	logname	Help
≎ у	Ŷ -	🔶 n	printenv	Help
♦ у	÷-	💠 n	pwd	Help
• y	Ŷ-	🔅 n	true	Help
▶ y	÷-	💠 n	uname	Help
> y	÷ -	🔶 n	which	Help
у	÷-	🔶 n	whoami	Help
> у	Ŷ-	🔶 n	yes	Help
			Sysutils tools	
▶ y	÷-	💠 n	reboot	Help
▶ y	÷-	💠 n	shutdown	Help
• y	Ŷ-	🔅 n	df	Help
▶ y	Ŷ-	💠 n	free	Help
▶ y	Ŷ-	💠 n	hostname	Help
) y	÷-	💠 n	kill	Help
▶ y	÷-	🔅 n	ps	Help
	Main M	lonu	Procee toole	Prev

Busybox

×	BusyBox	— x
	BusyBox	
🔶 y 👽 - 😺 n 🛛 BusyBox		Help
Applets		
Main Menu	Next	Prev

This option has to be enabled to enable Busybox.

×			BusyBox	
			BusyBox	
💸 у	÷ -	◆ n	cal	Help
🔶 y	÷-	💠 n	cat	Help
ŵ у	÷-	🔶 n	chgrp	Help
🔶 y	Ŷ -	🗢 n	chmod	Help
🔶 у	÷-	💠 n	chown	Help
🕹 у	÷ -	🔶 n	chroot	Help
⇒ у	÷-	🔶 n	chvt	Help
ŵ у	÷ -	🔶 n	clear	Help
🔶 у	÷-	💠 n	cmp	Help
🔶 у	÷ -	💠 n	ср	Help
⇒ γ	÷-	🏼 🔹 n	co: Preserve hard links	Help
	1	Main Mei	nu <u>N</u> ext <u>Prev</u>	/

	BusyBox	- x
	BusyBox	
- 🕹 n	date: Enable ISO date format output (-1)	Help
/ - 🔶 n	dc	Help
- 🗼 n	dd	Help
- 🔶 n	deallocvt	Help
	<pre>- * n - * n</pre>	r - v n date: Enable ISO date format output (-1) r • n dc r · v n dd

×				BusyBox		-
				BusyBox		
√ У	× -	🔶 n	iptunnel			Help
🔶 y	Ŷ -	💠 n	kill			Help
🔶 y	Ŷ -	💠 n	killall			Help
💸 y	Ŷ -	🔶 n	klogd			Help
1				1	1	()()()()()()()()()()()()_{(1,1)}())_{(1,1)}()_{(1,1)}())_{(1,1)}())_{(1,1)}()_{(1,1)}()))_{(1,1)}()))_{(1,1)}()))_{(1,1)}()))_{(1,1)}()))_{(1,1)}()))))))))))))))))))))))))))))))))
	0	Main Mei	<u> </u>	<u>N</u> ext	<u>P</u> rev	

No comments.

			BusyBox		
~ y	v -	• n	losetup	неір	IE
♦ y	ŵ -	💠 n	ls	Help	1
е у	÷-	💠 n	ls: show usernames	Help	1
• y	÷-	💠 n	ls: show timestamps	Help	1
♦ y	÷ -	🗢 n	ls: file types	Help	1
♦ y	÷-	💠 n	ls: sort files	Help	1
◆ y	÷ -	💠 n	ls: recursive	Help	1
• y	÷ -	💠 n	ls: follow links	Help	Ī
~ у	÷ -	🔶 n	ls: Use color to identify file types	Help	1

			BusyBox	
¢У	÷ -		more: use termios	Help
◆ y	÷ -	💠 n	mount	Help
у	÷ -	🔶 n	mount: loop devices	Help
⇒ y	÷-	🔶 n	mount: support /etc/mtab	Help

6	_	_	BusyBox	(199)
		1 11	BusyBox	
¢у	÷ -	\$ n	telnetd: Support call from inetd only	Help
♦ y	\$ -	💠 n	test	Help
> у	÷-	🔶 n	test: Extend test to 64 bit	Help
y y	÷ -	🔹 n	tftp	Help
у у	÷ -	🗢 n	tftp: put	Help
> у	÷ -		tftp: get	Help
у у	÷-	◇ n	tftp: Enable "blocksize" command	Help
~ y	÷ -	💠 n	tftp: Enable debug	Help
> у	÷-	◆ n	time	Help
▶ y	÷ -	💠 n	top	Help
▶ y	Ŷ -	💠 n	top: Support showing CPU usage percentage (add 2k bytes)	Help
> у	÷ -	🔶 n	touch	Help
> у	÷-	◆ n	tr	Help
> у	÷ -	🔷 n	traceroute	Help
¢ у	Ŷ -	\$ n	traceroute: Enable verbose output	Help
▶ y	÷-	💠 n	true_false	Help
> у	÷-	🔶 n	tty	Help
		Main Mei	nu Next Pr	rev

6			BusyBox	
			BusyBox	
¢ у	÷-	🔶 n	update	Help
▶ y	÷-	🗢 n	uptime	Help
<i>y</i>	÷-	🔶 n	usleep	Help
y	÷-	🔹 n	uudecode	Help
> y	÷-	🔶 n	uuencode	Help
y	÷-	🔶 n	unzip	Help
<i>y</i>	÷-	🔶 n	vconfig	Help
y	÷-	💠 n	vi	Help
y	÷-	💠 n	vi: colon commands	Help
y	÷-	💠 n	vi: yank / put / mark	Help
y	÷-	🗢 n	vi: search / replace	Help
у	÷-	🔶 n	vi: catch signals	Help
> y	÷-	🔶 n	vi: previous command	Help
y	÷-	💠 n	vi: enable read only	Help
y	÷-	💠 n	vi: setable options	Help
y	÷-	🗢 n	vi: allow :set	Help
y	÷-	◇ n	vi: window resizes	Help
1			· · · · · ·	1 1
		dain Mei	nu <u>N</u> ext <u>P</u> rev	

		BusyBox	
◇ y ◇ -	🔶 n	buffers on stack	Help
◇ y ◇ -	🔶 n	buffers in BSS	Help
◇ y ◇ -	🔶 n	Use devps	Help
♦ y 💠 -	💠 n	Verbose usages	Help
♦ y 💠 -	💠 n	Auto Width	Help
- v l	🔺 n	Silont hoot	Heln

X			BusyBox	<u> </u>
			BusyBox	
* *				
🔶 у	ŵ -	💠 n	Readable sizes	Help
⇒ у	÷ -	🔶 n	Locale Support	Help
🔶 у	÷-	💠 n	Top CPU usage	Help
		Main Mer	u Next	Prev
		Main Mei		

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.

⁶ 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 /tftboot

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 tcl⁷ 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

⁷ 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 files⁸:

- 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
 - o 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.

⁸ 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 apropriate 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 fooey 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 apropriate menu section (i.e. in the program group you want your app to show up in during 'make config'; I used 'misc'), like CONFIG_USER_FOO_FOO bool '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 = fooOBJS = foo.oall: \$(EXEC) \$(EXEC): \$(OBJS) \$(CC) \$(LDFLAGS) -0 \$@ \$(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) -0 \$@ \$@.0 \$(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" <u>http://www.teknologisk.dk/20356</u>

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