mind
embedded development
an ESSENSIUM division

Buildroot

The open source way for streamlined custom embedded systems

ESSENSIUM

mind
embedded development
an ESSENSIUM division

Overview

- Open source toolchains
- Buildroot features
- Development process

Tools in development process

- toolchain
  - cross-compiler
  - assembler & linker
  - (filesystem) image generator
- boot loader / image writer
- kernel
- libraries
- services (shell, networking, http server, GUI, ...)
- ... and your own application!

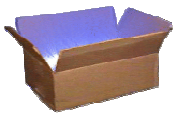
How to select your software

- Reduce development cost
  - Reuse existing software
  - Use easy-to-use software
  - Use reliable software
  - Use future-safe software
- Reduce per-unit cost
  - Avoid royalties
  - Use light-weight software

Why Open Source software?

- Reduce development cost
  - **Reuse existing software**
    - ➔ Adapt existing software to your needs
  - Use easy-to-use software
  - **Use reliable software**
    - ➔ Typically very frequent updates
    - ➔ Many debuggers
  - **Use future-safe software**
    - ➔ No vendor lock-in
- Reduce per-unit cost
  - **Avoid royalties**
    - ➔ Free to distribute!
  - **Use light-weight software**
    - ➔ Buildroot system can fit in a few MB

Open source alternatives

		
ecos		debian
200K	2M	200M
compile-time configuration	compile-time configuration	runtime configuration
cross-compile	cross-compile	compile on target

Buildroot: streamlined embedded system development

- Complete development chain in one box
- Buildroot = just the box
  - Buildroot itself = only 50MB
  - Packages fetched from Internet
  - Patches applied
  - Makefiles and configurations supplied
  - Configuration menu to select components
- Buildroot automates the entire process
  - Cross-compiler, assembler & linker are built
  - Kernel, standard library & required applications are built
  - Filesystem is built
  - Bootloader is built and installed
  - ➔ Just push the button, wait 12 hours and go!

http://buildroot.uclibc.org/

Buildroot features

- Cross-toolchain: gcc + GNU binutils + gdb
- Kernel: Linux (several versions + patches) / Hurd
- Bootloader: U-Boot / grub / syslinux / pxelinux
- Library: uClibc
- Filesystems: cramfs, ext2, iso, jffs2, romfs, squashfs, ubifs
- Additional software
  - busybox
  - POSIX / Linux / GNU tools
  - target toolchain
  - Database (mysql)
  - Editors
  - Audio
  - Graphics / Video
  - Networking (http and other servers, firewall, proxy, ...)
  - Scripting (lua, python, perl, ruby, php)
  - X windows system (X11 / Xorg / tinyX)
  - Qt / Qtopia
  - Many others...

Buildroot gives full flexibility

- You can edit the downloaded sources in-place
- You can add patches to each package
- You can add packages to the build
- You can add files to the filesystem

- You CAN NOT easily change the configuration of the running target.
  - There is no dynamic packaging system

Buildroot gives full control

- You have access to all the sources
- There are no binary dependencies
- You can start with any compiler
  - The cross-compiler itself is compiled by buildroot
- Once the system is built, you can replace any component by something else
  - No lock-in for anything, not even on buildroot itself

Buildroot is easy to use *for programmers*

- The first, unmodified build is out-of-the-box
  - Many boards are supported
- Existing makefiles are good examples
- Good documentation at uClibc.org

- No GUI-stuff (except for simple config menu)
- **Customization still takes an effort!**
  - Requires knowledge of the package being customized
  - Many different sources → different coding styles

Where to get help

- Start from a “known good” situation
  - Supported development board
  - Official release
- Search the internet
  - Many different mailing lists
  - Efficient searching requires a bit of expertise
- Hire help
  - Outsource bootstrapping process
  - Look for experts for developing specific features
    - additional drivers
    - board support
    - application porting
  - Hire an expert

Example development process with buildroot

1. Select a development board
   - Choose one supported by buildroot
2. Do an initial build
   - Sanity check that buildroot works for you
3. Develop your custom board
   - Remove interfaces, add ASICs, ...
4. Develop your application(s)
   - Usually in parallel with board development, on development board
   - Directly add applications to buildroot environment
5. Add drivers / board support to buildroot
6. Contribute your drivers to the community
   - This makes it easier for you to follow upgrades

Debugging and profiling

- Low-level debugger: gdb
  - target only has to run gdb_server
  - communicates with the real debugger on a PC
  - can be connected to a GUI (ddd)
- Low-level profiler: OProfile
  - Uses hardware performance counters
  - Kernel-level support
- No real fancy GUI tools
  - but you can combine with proprietary tools!

Conclusions

- Choose open source if you need customization
- Buildroot is a good choice if the system is
  - small
  - not x86
- Open source development process is different
  - Use the source
  - Use the 'net

www.mind.be

www.essensium.com

Essensium NV
Mind - Embedded Software Division
Gaston Geenslaan 9, B-3001 Leuven
Tel : +32 16-28 65 00
Fax : +32 16-28 65 01
email : info@essensium.com