Introduction to Buildroot

Advanced Embedded Linux Development

with **Dan Walkes**

University of Colorado **Boulder**

Learning objectives:

Build System Overview

Buildroot Overview

Linux Components

- Toolchain
- bootloader
- kernel
- root filesystem

Mastering Embedded Programming Chapter 6
https://opensource.com/article/18/6/embedded-linux-build-tools

What makes a Build System?

- How are you going to generate these?
- Options:
  - Buildroot, Yocto, OpenWRT
  - Roll Your Own (RYO)
  - Existing distribution (debian, Ubuntu, etc)

Why use a Build System over RYO?

- Why reinvent the wheel?
- Is there really something so specific about your project that means you can't leverage significant portions from other projects?
- What is your strategy for keeping up with security patches?

https://opensource.com/article/18/6/embedded-linux-build-tools

Build System vs Distribution

- Goals of a distribution are likely different than your project
  - Focused on users/desktops.
  - Upgrades presume someone is interacting with the system.
  - Package management not designed for embedded software development

https://opensource.com/article/18/6/embedded-linux-build-tools

Build System vs Distribution

- Reducing the image size may be challenging.
- Likely needs customization work to generate a production image.
- Binary compatibility on different upgrade paths (v1->v3 vs v1->v2->v3) may be challenging.

https://opensource.com/article/18/6/embedded-linux-build-tools

Steps Performed by a Build System

- Download source for common packages from upstream
- Apply patches for cross compilation, arch dependent bugs, etc
- Build components
- Assemble rootfs in staging area
- Create image files

Other Build System Features

- Add your own packages (applications or drivers)
  - Often using proprietary license for some/all of these.
- Select system profiles
  - with/without graphics
- Track open source licenses used
  - Help with open source license compliance

What is Buildroot?

- Builder of root filesystems for embedded devices
- Collection of Linux packages with build instructions
  - Host and target packages
- Started in 2001
- Focuses on simplicity

What is Buildroot?

- Build system is licensed GPLv2
  - You are expected to share changes to Buildroot source.
- Leverages popular and ubiquitous make files and kconfig/menuconfig
  - Same configuration mechanism used with the Linux kernel, busybox, ct-ng

Installing Buildroot

- Based on GNU make and utilities
- Need a collection of packages, most of which you already probably have installed.
  - See mandatory packages at https://buildroot.org/downloads/manual/manual.html#requirement-mandatory
  - Also suggest ncurses (libncurses5-dev)

https://buildroot.org/downloads/manual/manual.html#requirement-mandatory

Buildroot Packages

- Buildroot uses “packages” with build and install instructions utilizing GNU Make syntax.
- By default these packages are located directly in the buildroot tree under a “package” directory
  - Alternatively you can use your own tree for package builds, what Buildroot calls “br2-external” trees

https://buildroot.org/downloads/manual/manual.html#outside-br-custom

Buildroot Packages

- Why might it make sense to use an external tree?
  - For packages you can't share upstream (proprietary license)
- Packages can reference git repositories for source code.
  - Build your own custom applications from your own dedicated repositories.

https://buildroot.org/downloads/manual/manual.html#outside-br-custom
https://buildroot.org/downloads/manual/manual.html#adding-packages

Buildroot Packages

- Packages need at least two files:
  - Config.in - KConfig code adding the package to the config menu
  - <package_name>.mk
- Buildroot Package definitions do not contain code
  - Instead, they contain instructions to get code, compile, and add to rootfs