# >>> Operating Systems And Applications For Embedded Systems
# >>> Build System

Name: Mariusz Naumowicz
Date: 26 marca 2019

>>> Plan

1. Build system
   - Open source build systems
   - Abilities
   - Other useful features
   - Open source build systems
2. Buildroot
   - Buildroot
   - Installing
   - Configuring
   - Running
   - License compliance
   - Adding source
     - package/Config.in
     - package/hello/Config.in
     - package/hello/hello.mk
     - package/hello/src/Makefile
3. The Yocto Project
   - Components
   - Installing the Yocto Project
   - Configuring
   - Building
   - Running
   - BitBake and recipes
   - Interesting features

# >>> Open source build systems

* The toolchain
* The bootloader
* The kernel
* The root filesystem

## >>> Abilities

* Download a source from upstream, either directly from the source code control system or as an archive, and cache it locally
* Apply patches to enable cross compilation, fix architecture-dependent bugs, apply local configuration policies, and so on
* Build the various components
* Create a staging area and assemble a root filesystem
* Create image files in various formats ready to be loaded onto the target

## >>> Other useful features

* Add your own packages containing, for example, applications or kernel changes
* Select various root filesystem profiles: large or small, with and without graphics or other features
* Create a standalone SDK that you can distribute to other developers so that they don't have to install the complete build system
* Track which open source licenses are used by the various packages you have selected
* Allow you to create updates for in-field updating
* Have a user-friendly user interface

## >>> Open source build systems

1. Buildroot: An easy-to-use system using GNU make and Kconfig (http://buildroot.org)
2. EmbToolkit: A simple system for generating root filesystems; the only one at the time of writing that supports LLVM/Clang out of the box (https://www.embtoolkit.org)
3. OpenEmbedded: A powerful system which is also a core component of the Yocto Project and others (http://openembedded.org)
4. OpenWrt: A build tool oriented towards building firmware for wireless routers (https://openwrt.org)
5. PTXdist: An open source build system sponsored by Pengutronix (http://www.pengutronix.de/software/ptxdist/index_en.html)
6. Tizen: A comprehensive system, with emphasis on mobile, media, and in-vehicle devices (https://www.tizen.org)
7. The Yocto Project: This extends the OpenEmbedded core with configuration, layers, tools, and documentation: probably the most popular system (http://www.yoctoproject.org)

## >>> Buildroot

The Buildroot project website is at http://buildroot.org.

Current versions of Buildroot are capable of building a toolchain, a bootloader (U-Boot, Barebox, GRUB2, or Gummiboot), a kernel, and a root filesystem. It uses GNU make as the principal build tool.

There is good online documentation at http://buildroot.org/docs.html, including The Buildroot User Manual.

## >>> Installing

1. git clone git://git.buildroot.net/buildroot
2. cd buildroot

## >>> Configuring

1. cd buildroot
2. make qemu_arm_versatile_defconfig
3. make

>>> Running

```
qemu-system-arm -M vexpress-a9 -m 256 -kernel output/images/zImage -dtb
output/images/vexpress-v2p-ca9.dtb -drive file=output/images/rootfs.ext2,if=sd
-append "console=ttyAMA0,115200 root=/dev/mmcblk0" -serial stdio -net
nic,model=lan9118 -net user
```

## >>> License compliance

```
make legal-info
```

## >>> Adding source

1. modify file package/Config.in
2. create project directory in package directory
3. in project directory create files: Config.in project_name.mk and directory src
4. move your source files into src directory and create Makefile

## >>> Adding source

```
menu "Misc"
source "package/hello/Config.in"
endmenu
```

>>> package/hello/Config.in

```
config BR2_PACKAGE_HELLO
    bool "hello"
    help
Hello world package.

http://example.com
```

>>> package/hello/hello.mk

```
################################################################################
#
# hello
#
################################################################################

HELLO_VERSION = 1.0
HELLO_SITE = ./package/hello/src
HELLO_SITE_METHOD = local

define HELLO_BUILD_CMDS
$(MAKE) CC="$(TARGET_CC)" LD="$(TARGET_LD)" -C $(@D)
endef

define HELLO_INSTALL_TARGET_CMDS
$(INSTALL) -D -m 0755 $(@D)/hello $(TARGET_DIR)/usr/bin
endef

$(eval $(generic-package))
```

>>> package/hello/src/Makefile

```
CC = gcc

.PHONY: clean

hello: hello.c
$(CC) -o '$@' '$<'

clean:
rm hello
```

## >>> Components

1. Poky: The reference distribution
2. oe-core: The core metadata, which is shared with OpenEmbedded
3. BitBake: The task scheduler, which is shared with OpenEmbedded and other projects
4. Documentation: User manuals and developer's guides for each component
5. Hob: A graphical user interface to OpenEmbedded and BitBake
6. Toaster: A web-based interface to OpenEmbedded and BitBake
7. ADT Eclipse: A plug-in for Eclipse that makes it easier to build projects using the Yocto Project SDK

## >>> Installing the Yocto Project

```
git clone -b fido git://git.yoctoproject.org/poky.git
```

## >>> Configuring

1. cd poky
2. source oe-init-build-env

```
source oe-init-build-env build-qemuarm
```

## >>> Building

* core-image-minimal: A small console-based system which is useful for tests and as the basis for custom images.
* core-image-minimal-initramfs: This is similar to core-image-minimal, but built as a ramdisk.
* core-image-x11: A basic image with support for graphics through an X11 server and the xterminal terminal app.
* core-image-sato: A full graphical system based on Sato, which is a mobile graphical environment built on X11, and GNOME. The image includes several apps including a terminal, an editor, and a file manager.

```
bitbake core-image-minimal
```

## >>> Running

To run the QEMU emulation, make sure that you have sourced oe-init-build-env and then just type:
runqemu qemuarm

## >>> BitBake and recipes I

* recipes: Files ending in .bb. These contain information about building a unit of software, including how to get a copy of the source code, the dependencies on other components, and how to build and install it.
* append: Files ending in .bbappend. These allow some details of a recipe to be overridden or extended. A.bbappend file simply appends its instructions to the end of a recipe (.bb) file of the same root name.
* include: Files ending in .inc. These contain information that is common to several recipes, allowing information to be shared among them. The files may be included using the include or require keywords. The difference is that require produces an error if the file does not exist, whereas include does not.
* classes: Files ending in .bbclass. These contain common build information, for example how to build a kernel or how to build an autotools project. The classes are inherited and extended in recipes and other classes using the inherit key word. The class classes/base.bbclass is implicitly inherited in every recipe.
* configuration: Files ending in .conf. They define various configuration variables that govern the project's build process.

## >>> BitBake and recipes II

The default task is do_build, so that you are running the build task for that recipe. You can list the tasks available in a recipe by running bitbake core-image-minimal like this:

```
bitbake -c listtasks core-image-minimal
```

The -c option allows you to specify the task, missing off the do_ part. A common use is -c fetch to get the code needed by a recipe:

```
bitbake -c fetch busybox
```

You can also use fetchall to get the code for the target and all the dependencies:

```
bitbake -c fetchall core-image-minimal
```

## >>> Interesting features

* dbg-pkgs: installs debug symbol packages for all the packages installed in the image.
* debug-tweaks: allows root logins without passwords and other changes that make development easier.
* package-management: installs package management tools and preserves the package manager database.
* read-only-rootfs: makes the root filesystem read-only.
* x11: installs the X server.
* x11-base: installs the X server with a minimal environment.
* x11-sato: installs the OpenedHand Sato environment

## >>> Further reading

* The Buildroot User Manual, http://buildroot.org/downloads/manual/ manual.html
* Yocto Project documentation: there are nine reference guides plus a tenth which is a composite of the others (the so-called Mega-manual") at https://www.yoctoproject.org/documentation
* Instant Buildroot, by Daniel Manchón Vizuete, Packt Publishing, 2013
* Embedded Linux Development with Yocto Project, by Otavio Salvador and Daianne Angolini, Packt Publishing, 2014

# >>> References

C. Simmonds.
*Mastering Embedded Linux Programming.*
Packt Publishing, 2015.