Assignment 4: Buildroot Builds

Advanced Embedded Linux Development

with **Dan Walkes**

University of Colorado **Boulder**

Learning objectives:

Git Submodules

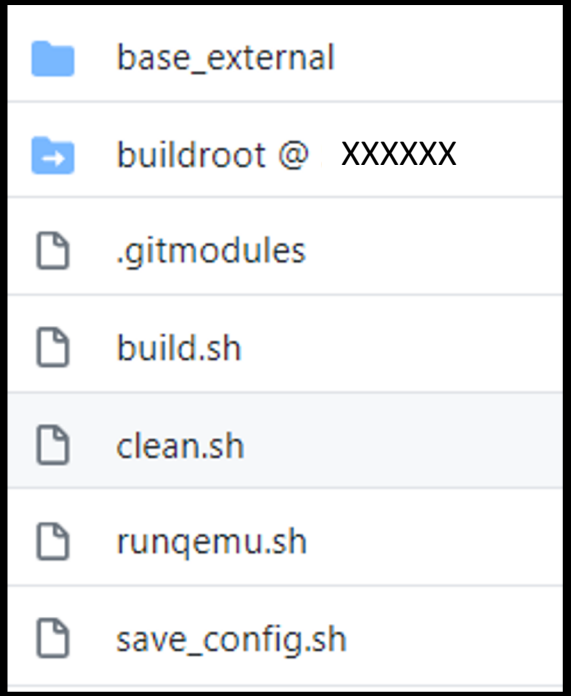
Overview of Buildroot Example Project

Git Submodules

- Add a git repository as a subdirectory of another repository.
- Keep two projects separate but reference one from the other.
  - Keep your project specific changes out of a shared project.
- We will use buildroot as a submodule for Assignment 4

https://git-scm.com/book/en/v2/Git-Tools-Submodules

Git Submodules



- Your assignment 4 repository will contain buildroot as a submodule
- @ XXXX references a specific commit hash corresponding to release specified in the assignment

https://git-scm.com/book/en/v2/Git-Tools-Submodules

Buildroot Git Submodule

- Add as a submodule in the root using git submodule add <buildroot-repo-url>
- Check out the branch corresponding to the release referenced in the assignment
- git add buildroot to add to staging
  - connects buildroot commit to this directory
- Keep your changes in the base project
  - Easy to retarget to a new buildroot release

https://git-scm.com/book/en/v2/Git-Tools-Submodules
https://git.busybox.net/buildroot/

Buildroot Packages

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

Building with Buildroot

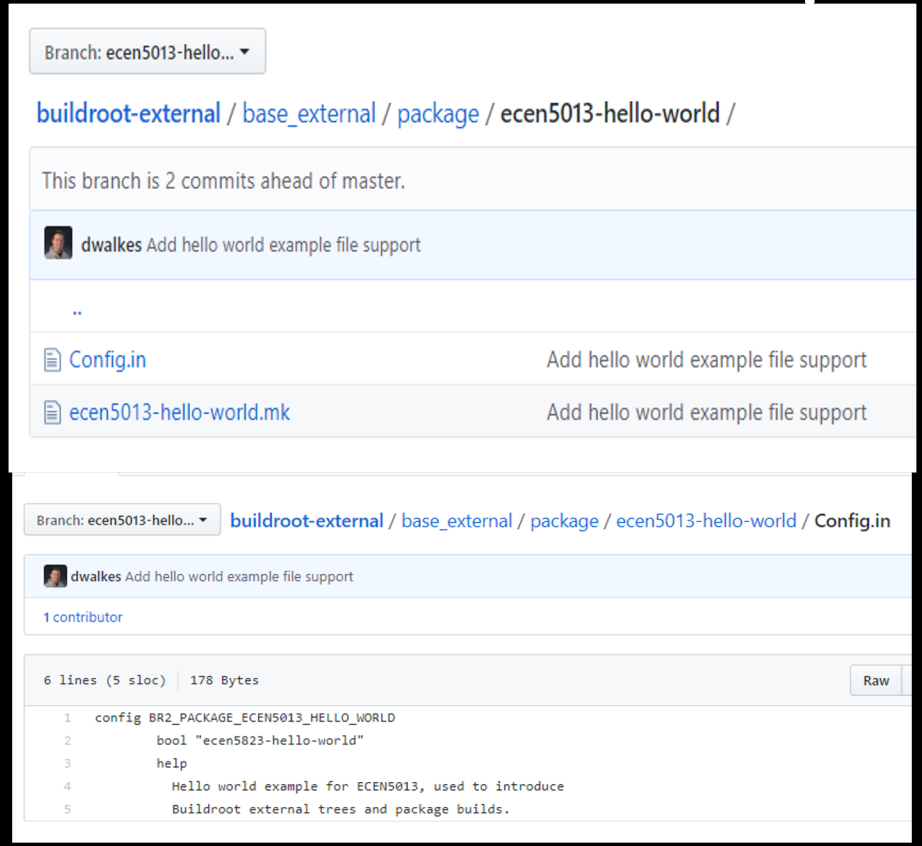
- Create a aesd-assignments package folder under a base_external package directory
- Add Config.in file
- Create an aesd-assignments.mk file
  - (Leverage Assignment 4 starter code)

https://buildroot.org/downloads/manual/manual.html#customize-packages
https://buildroot.org/downloads/manual/manual.html#adding-packages
https://buildroot.org/downloads/manual/manual.html#generic-package-tutorial

Example/Starter Code Overview

- See the buildroot-assignments-base repository for starter code https://github.com/cu-ecen-5013/buildroot-assignments-base
- See working hello-world example at https://github.com/cu-ecen-5013/buildroot-external/tree/ecen5013-hello-world

Buildroot Example Project




```
################################################################################
#
# ecen5013-hello-world
#
################################################################################

ECEN5013_HELLO_WORLD_VERSION = 452e6e18d1323df20a309e1e55300063d893777e
# Note: we use an https URL here because the repository is public
# If it were private, we'd use an SSH URL instead.
ECEN5013_HELLO_WORLD_SITE = https://github.com/cu-ecen-5013/ecen5013-hello-world.git
ECEN5013_HELLO_WORLD_SITE_METHOD = git

define ECEN5013_HELLO_WORLD_BUILD_CMDS
        $(MAKE) $(TARGET_CONFIGURE_OPTS) -C $(@D) all
endef

define ECEN5013_HELLO_WORLD_INSTALL_TARGET_CMDS
        $(INSTALL) -m 0755 $(@D)/hello-ecen5013 $(TARGET_DIR)/bin
endef

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

https://github.com/cu-ecen-5013/buildroot-external/tree/ecen5013-hello-world/base_external/package/ecen5013-hello-world

aesd-assignments.mk file setup

- $(eval $(generic-package))
  - Cross compiles packages for the target
  - Knows how to use LIBFOO_XXXX variables where LIBFOO is replaced with your package name. See the manual for variable details
- $(INSTALL) maps to the install utility - see https://linux.die.net/man/1/install

https://buildroot.org/downloads/manual/manual.html#generic-package-reference

Setting up base external

- Add base_external/Config.in, base_external/external.mk, and base_external/external.desc files to allow you to use package directories outside buildroot
- Use “project_base” as your external name in external.desc
  - See buildroot link below for details
  - See also example ecen5013-hello-world implementation

https://buildroot.org/downloads/manual/manual.html#outside-br-custom
https://github.com/cu-ecen-5013/buildroot-external/tree/ecen5013-hello-world

Setting up aesd-assignments

- aesd-assignments.mk needs to reference your assignment 3 repository:
  - Reference the **ssh** repository URL to work properly with the autograder.
  - Needs to represent a specific commit to build.
- It also needs to complete installation of scripts in the /bin directory
- See link below for template

https://github.com/cu-ecen-5013/buildroot-assignments-base/blob/master/base_external/package/aesd-assignments/aesd-assignments.mk

build.sh template

- See build script linked below
- This script
  - Handles git submodule initialization/update
  - Ensures .config file exists, using
    - Project specific config if it exists OR
    - qemu_aarch64_virt_defconfig as fallback from buildroot

https://github.com/cu-ecen-5013/buildroot-assignments-base/blob/master/build.sh

Initial Build

- After setting up base_external and aesd-assignments, run ./build.sh the first time
  - This creates your modified QEMU defconfig file

```
if [ ! -e buildroot/.config ]
then
```

```
if [ -e ${AESD_MODIFIED_DEFCONFIG} ]
then
```

```
# The defconfig from the buildroot directory we use for qemu builds
QEMU_DEFCONFIG=configs/qemu_aarch64_virt_defconfig
# The place we store customizations to the qemu configuration
MODIFIED_QEMU_DEFCONFIG=base_external/configs/aesd_qemu_defconfig
# The defconfig from the buildroot directory we use for the project
AESD_DEFAULT_DEFCONFIG=${QEMU_DEFCONFIG}
AESD_MODIFIED_DEFCONFIG=${MODIFIED_QEMU_DEFCONFIG}
AESD_MODIFIED_DEFCONFIG_REL_BUILDROOT=../${AESD_MODIFIED_DEFCONFIG}
```

```
else
        echo "Run ./save_config.sh to save this as the default configuration in ${AESD_MODIFIED_DEFCONFIG}"
        echo "Then add packages as needed to complete the installation, re-running ./save_config.sh as needed"
        make -C buildroot defconfig BR2_EXTERNAL=${EXTERNAL_REL_BUILDROOT} BR2_DEFCONFIG=${AESD_DEFAULT_DEFCONFIG}
fi
```

https://github.com/cu-ecen-5013/buildroot-assignments-base/blob/master/build.sh
https://github.com/cu-ecen-5013/buildroot-assignments-base/blob/master/shared.sh

Saving Configuration

- Then run ./save_config.sh as instructed by the build script
- You should see a new configuration file at base_external/configs/aesd_qemu_defconfig

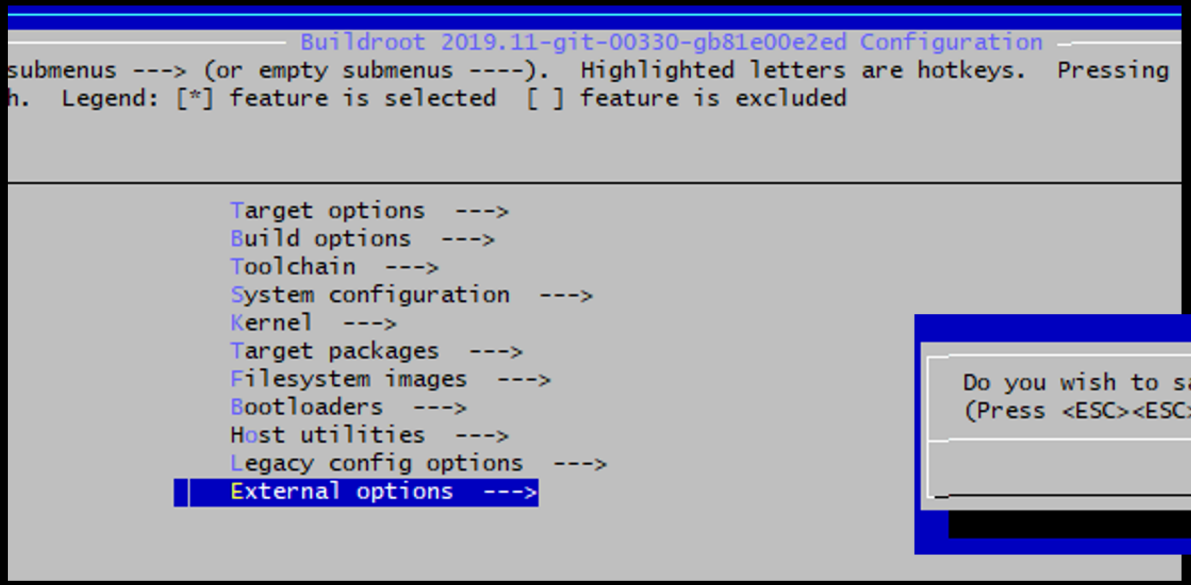
```
source shared.sh
mkdir -p base_external/configs/
make -C buildroot savedefconfig BR2_DEFCONFIG=${AESD_MODIFIED_DEFCONFIG_REL_BUILDROOT}
```

https://github.com/cu-ecen-5013/buildroot-assignments-base/blob/master/save_config.sh
https://buildroot.org/downloads/manual/manual.html#customize-store-buildroot-config

Adding your package

- You should see your package after running make menuconfig in the buildroot directory
- Select the package and then run ./build.sh
- Use save_configs.sh to save your configuration with the project external tree
  - Your coworkers (SAs) will need to delete their buildroot/.config file, then run ./build.sh to pick up config changes

Adding your package



Saving Configuration

- Run save_config.sh again
- You should see your buildroot config at base_external/configs/aesd_qemu_defconfig to include the aesd-assignments package

```
15  BR2_PACKAGE_AESD_ASSIGNMENTS=y
```

Adding your new package

- Now you ready to build your image with your custom package
  - Use ./build.sh
  - Or use “make” from the buildroot directory
  - Or use make aesd-assignments from the buildroot directory

https://buildroot.org/downloads/manual/manual.html#make-tips