# ECEN 5013

## Assignment 7: Buildroot and Yocto Kernel Driver Builds

# Github Classroom Link

https://classroom.github.com/a/3UUtrVu-

# Suggested Reading:

1. Lecture 14 and 15.
2. Buildroot documentation, including:
    a. https://buildroot.org/downloads/manual/manual.html#_infrastructure_for_packages_building_kernel_modules
    b. https://buildroot.org/downloads/manual/manual.html#rootfs-custom

# Implementation:

1. Your assignment source code will start with the content of the ldd3 repository. Start by making a minor modification to include your github username in the hello.c module output inside the misc-modules folder.
2. Update your buildroot assignment submission repository for assignment 5 to include support for building and installing the misc-modules and scull components on your qemu target based on your fork of the ldd3 repository.
    a. Use git to tag your assignment 5 repository with “assignment 5 done” before making any modifications.
    b. Add a new package “ldd” to your assignment 5 base_external directory which supports building and installing the misc-modules and scull components into the root filesystem. Add this package to the image.
    c. Use a rootfs overlay in buildroot to add an init module script named “S98lddmodules”. **Be sure to specify a relative path to support builds outside your working directory**:
        i. run scull_load functions on startup to load the scull driver, and run scull_unload functions on shutdown to unload.
        ii. Run module_load functions on startup to load the faulty driver and run module_unload functions on shutdown to unload.

iii. Run modprobe on the hello module to load on startup and use rmmod to remove on shutdown.

d. Run *echo "hello_world" > /dev/faulty* from the command line of your running image and copy the resulting kernel oops to your assignment 7 repository in a assignments/assignment7/faulty-oops.md
   i. Add your analysis of this kernel oops in an assignments/assignment7/faulty-oops.md file, using the discussions in lecture 15 and the Mastering Markdown github instructions.

## Validation:

1. You should be able to clone your final buildroot assignment repository to a new directory, run ./build.sh to build the system image and ./runqemu.sh to start the image with no other interaction necessary.
   a. The image should include printk logs in /var/log/messages including your github user name from the hello module which is loaded automatically during startup.
   b. You should should have /dev/scull devices in the /dev directory as well as /dev/faulty
   c. Modules hello, faulty and scull should all be listed in lsmod after startup.
   d. Running /etc/init.d/S98lddmodules stop should unload all modules on the buildroot instance.
   e. Your analysis in your assignment 7 repository within assignments/assignment7/faulty-oops.md should explain the content and how you can use this to locate the faulty line in the kernel driver.

## Extra Credit (+10 points)

Suggested reading:

c. https://www.yoctoproject.org/docs/2.6/kernel-dev/kernel-dev.html#working-with-out-of-tree-modules
d. https://www.yoctoproject.org/docs/3.0/ref-manual/ref-manual.html#ref-devtool-reference

3. Update your yocto submission repository for assignment 6 to include support for building and installing the misc-modules and scull components on your qemu target.
   a. Tag your assignment 6 repository with "assignment 6 done"
   b. Add an "ldd" package to your meta-aesd layer which supports building and installing the misc-modules and scull components into the root filesystem. Add this package to the image. Here are some hints about yocto steps you can take in order to complete this task:
      i. Use devtool add to add the ldd3 repository to your build

ii. Modify the ldd3 makefile in the devtool working directory to include only scull and misc-modules components
iii. Use devtool finish to save the corresponding .bb and Makefile patch files to your repository as a “scull” recipe in the meta-aesd layer.
iv. Modify the task-install in the .bb file to use the correct module folder for the M argument
v. Add a “files” subfolder in your scull recipe and place an init script in this folder which does the portion of initialization specific to the scull module.
vi. Use the update-rc.d framework used in the previous assignment to install the script.
vii. Copy the scull recipe to a new “misc-modules” recipe.
viii. Modify the misc-modules recipe to include an install script for misc-modules components.
ix. Include both scull and misc-modules in your core-image-aesd image.

## Submission:

1. Your submission repository master branch should contain the minor changes to personalize the hello module, modified from the default sources of the ldd3 repository.
2. Your assignment 5 (and assignment 6 for extra credit) repository master branches should be updated to include the ldd3 components, and successfully build and run each image, as discussed in the validation section.
3. If submitting extra credit, please email the instructor and student assistants.
    a. Extra credit submissions are not allowed after the assignment due date.