

Lab 1

Yocto Installation and in-class work. Provide a report (PDF).

Lab 2

For this week's lab, your task will be to create a systemd service that functions each time your embedded Linux system boots up; you can prototype your service on your host machine. When your service information comes up via the systemctl status command, be sure to include the FIRST NAME of EACH of your group members in the description (show this output in your lab report).

Lab 3

Demonstrate operation of your updated Linux Firmware on Beaglebone.

Lab 4

Show the working implementation of your udev/systemd files for automatically mounting an external micro-SD "drive" on your filesystem

Lab 5

Modify (if necessary) your uEnv.txt file to accommodate the external touchscreen display for your embedded device.

Lab 6

Create a test application, and show the following:

- 1) your cross tool can build an application which can be sent to the embedded Linux target via sftp and executed successfully there;
- 2) you are able to debug your application, stepping through the code on your host, line by line, while the program executes on your embedded Linux machine (Beaglebone).

Lab 7

Connecting a USB camera to the Beaglebone. If you are using a USB hub, please be sure to power it, or you may get an access error with your Beaglebone! Connect to your USB camera as shown in class (follow github notes), and take some footage. Display on your Beaglebone's GUI display.

Lab 8

Look through the various references provided to familiarize yourself with Qt, including: 1) Exploring Beaglebone, Qt Primer, pp. 615-629, and 2) Mastering Qt 5, available via amazon.ca at: https://www.amazon.ca/Mastering-Qt-stunning-cross-platform-applications-ebook/dp/B07DH9YK9Q/ref=sr_1_1?dchild=1&keywords=mastering+qt&qid=1608566592&s=digital-text&sr=1-1, 3) Design Patterns in C++ with Qt (by Ezust & Ezust), available in the github archive, Widgets chapter; report on your understanding of Qt implementation for GUI applications.