


High Performance Embedded Systems
Assignment 1

Group #:	6
Final Grade:	97.50%
Additional comments	Git repo was checked out at commit 500f2b8 - Jun 26th 13:44:04

Part I: QEMU

Item	Description	Total	Grade	Comments
Full boot deliverables	- images/ - u-boot-qemu.log - linux-qemu.log - qemu_boot.log - video.txt	15%	15.00%	
Full boot execution	Correct execution of uboot, kernel and file system on QEMU based on video and local testing.	25%	25.00%	- It is not necessary to use the / with fatload in this case since the files are already at root directory.
Bonus	Add boot commands to u-boot environment	10%	0.00%	

Part II: Yocto & NEON

Item	Description	Total	Grade	Comments
Yocto meta-layer	Correct creation of the meta-tec layer and tec-image. Inherits from the console-image. Adds all new recipes into the tec-image and includes them in the final image.	5%	5.00%	- Packages are usually not listed into conf/layer.conf file since this avoids control over the image when creating custom image files. Packages should be defined in the image file.
Yocto recipes	Correct creation of the following recipes: - rgb2yuv_c - rgb2yuv_intrinsics This includes correct build and installation of the binaries into the file system. Also it includes the presence of the required PDF documentation.	10%	10.00%	- Great usage of repository for the recipes. - It is necessary to clean up some autogenerated files from autotools. 
Autotools usage	Correct usage of autotools for the compilation of the programs.	10%	10.00%	
Getopt implementation	Correct usage of getopt for the command line options	5%	5.00%	
Application implementation and functionality	The application meets with the requirements proposed and is totally functional providing the correct image conversion.	25%	25.00%	- Good code, good indentation and really simple to follow. - Always try to clean up your code, in general, any commented code should be removed since it is useless. Using control versioning makes this easier.

General

Item	Description	Total	Grade	Comments
Git control versioning	The delivery is correctly made using a git repository with the layout suggested and following the required work flow. User participation from all team members. Avoid using a unified commit or last hour commits (development must be continuous)	5%	2.50%	- Poor usage of git development flow. Direct commits to master should always be forbidden.