

Assingment 1: Bump it up!

Group #: 4

Grade: 93

Item	Description	Percentage	Grade	Comments
Yocto meta-layer	Correct creation of the meta-tec layer.	10	10	
Yocto recipes	Correct creation of the recipes. This includes correct build (including doxygen generation) and installation of the binaries into the file system.	10	10	Everything worked great!
Autotools, GNU make or Cmake usage	Correct usage of autotools, GNU make or Cmake for the compilation of the program.	10	10	
Git control versioning	The delivery is correctly made using a git repository with the layout suggested and following the required work flow.	10	10	
Getopt implementation	Correct usage of getopt for the command lines options	5	5	On informative switches such as -h you should exit the application and do not continue. Otherwise you will have error messages such as in your case.

Application plementation functionality	im- and	The application meets with the requirements proposed and is totally functional providing the correct image conversion.	45	45	<ul style="list-style-type: none"> • One of the cleanest source codes I have seen on this project! • Great indentation! Personally I like spaces instead of tabs. Always indent C code. See https://www.gnu.org/software/indent/manual/indent.html • If something is commented out then you don't need it. • Try to not mix camelCase style with underscore_case style. C usually uses underscore. • Just as expected! Great job! <pre> root@raspberrypi2:~/g4# ./rgb2yuv_c -i image.rgb -o c.yuv -Info- RGB to YUV tool will use the image image.rgb as the source image -Info- RGB to YUV tool will store the image resultant in the file c.yuv -Info- The elapsed time of the function 'rgb2yuv' was: 257.914000 ms root@raspberrypi2:~/g4# ./rgb2yuv_intrinsics -i image.rgb -o intrinsics.yuv -Info- RGB to YUV tool will use the image image.rgb as the source image -Info- RGB to YUV tool will store the image resultant in the file intrinsics.yuv -Info- The elapsed time of the function 'rgb2yuv' was: 66.780000 ms root@raspberrypi2:~/g4# ./rgb2yuv_neon -i image.rgb -o neon.yuv -Info- RGB to YUV tool will use the image image.rgb as the source image -Info- RGB to YUV tool will store the image resultant in the file neon.yuv -Info- The elapsed time of the function 'rgb2yuv' was: 43.402000 ms root@raspberrypi2:~/g4# ls -lah *.yuv -rw-r--r-- 1 root root 600K Apr 25 23:14 c.yuv -rw-r--r-- 1 root root 600K Apr 25 23:15 intrinsics.yuv -rw-r--r-- 1 root root 600K Apr 25 23:15 neon.yuv </pre>
--	------------	--	----	----	--

Documentation	The documentation is complete and correct according with the requirements. Including the sample images.	10	3	<ul style="list-style-type: none">• Nice try to use Doxygen, sadly it comes pre-regerated (not generated by autotools).• -7 points: Only rgb2yuv-c comes with full documentation. The others seems to be incomplete (no images) and also the reported times are all the same (likely you forgot to update them).
Bonus	Doxygen integration	5	-	
Total	-	100	93	-