RaspiStillYUV modified

The RaspiStillYUV.c file provided is a modified version of the original RaspiStillYUV.c. It was made to be used in a homemade personal project with the intention of providing a faster way of taking and processing photos with the raspberry pi. It is by no means official or 100% reliable. The user might need to refine some code outside the processing section in order to make it work.
IT IS RECOMMENDED THE READING OF THE COMMENTED SECTIONS OF THE CODE AS THEY CONTAIN USEFUL INFORMATION (especially regarding how to access data according to each color scheme).

Program Description:

Inside the main() function there will be a section (line 1462) were the processing should be made. The user can then use that section to process the image as desired. Image data can be accessed through "(*state.camera_pool→header)→data[pix]", were "pix" will be the pixel the user wants to access according to the desired color scheme.
The program provided contains a section that is meant to threshold darkness in pixels and print it onto a file (to do so we must add ">file.txt" at the end of the program command, like so "raspiyuv -h 2464 -w 2464 >file.txt")

This program also allows the usage of ivport camera multiplexers. It will be required that the user reads ivport's documentation regarding it's particular board and usage. Also download and read the code provided at https://github.com/ivmech/ivport to enable i2c communication (it may be required to run some python scripts at boot before running your program), and also download wiringPi libraries to use GPIO.

Compiling the program:

TO SETUP EVERITHING BEFORE COMPILING HEAD TO:
https://thinkrpi.wordpress.com/2013/05/22/opencvpi-cam-step-1-install/
FOLLOW EVERY STEP UNTIL YOU ARE ABLE TO SUCCESFULLY COMPILE THE ORIGINAL PROGRAM!

To compile the modified program, the user must replace the original file at /userland/host_applications/linux/apps/raspicam/RaspiStillYUV.c with the file provided. If the user doesn't need GPIO, delete all wiringPi sections and simply run "./userland/buildme". If the user needs GPIO, add the following line to the makefile in the same folder "set (MMAL_LIBS mmal_core mmal_util mmal_vc_client -lwiringPi)" (added -lwiringPi) and then run "./userland/buildme" to compile.

To learn about raspberry i2c communication:
https://raspberry-projects.com/pi/programming-in-c/i2c/using-the-i2c-interface

To download and learn about wiringPi:
http://wiringpi.com/