ECE 484W

Assignment #4 Tutorial
Communicating between the LCD Screen (VEEK-MT2S) and Computer

Due by date listed on Canvas

Turn in your report via Canvas (each individual needs to do it; accumulated 10 points deduction for every 12 hours window late submission)

Turn in your video clip (less than one minute; each individual needs to do it) via Canvas to demonstrate each required task (Accumulated 10 points deduction for every 12 hours window late submission.). If you video clip does not demonstrate each required task, you will lose 10 points for each missing task. If you do not submit your video clip, we will not grade your report.

.

If you submit your report/video clip multiple times, I will grade the last attempt and your submission time will be the time for your last attempt

Requirements:

Display picture data to the on-board LCD screen, in more detail:

1. On the Qt GUI:
   - Add buttons to transfer original and overlay images from the GUI.
     - » If “Transfer Original” is pressed, the original image uploaded by the user should be sent to the VEEK-MT2S board and displayed on the LCD in a similar manner to last assignment utilizing a router.
     - » The same functionality should be implemented for “Transfer Overlay”.
   - Add a button mechanism to turn the overlay image on the LCD screen on/off.
   - Enhance the functionality of your brightness/contrast sliders by having them communicate their values to the VEEK-MT2S board through the communication implemented last assignment.
   - Above requires C/C++ or Python coding on the VEEK-MT2S board alongside modifications to Qt program from your previous assignment

(Note that the VEEK-MTS may support a different resolution; it is fine if you restrict the images to this resolution)

2. Modify the on-board program in C/C++ or Python (No “open-source” code) to adjust the brightness and contrast of test image based on brightness and contrast values transmitted from your Qt program.

(In other words, you should implement image processing algorithm on VEEK-MT2S.)

This assignment may require a keyboard to be attached to the VEEK-MT2S Board

1. An alternative to this may work by establishing PuTTY connection with Serial as completed in the last assignment, running "ifconfig" and noting the IP address and then closing the Serial connection…
2. Then, reopen PuTTY, change the PuTTY connection type to SSH instead of the normal Serial, typing the noted IP in the space and sending commands to the board in this manner
   - If you receive "Gtk-WARNING **: cannot open display:", I recommend attaching a USB keyboard and running the commands directly on a LXTerminal opened on the LCD screen itself (this is caused by not having a display to connect to via PuTTY communication)

Re-flashing the SD Card to VEEK Image:

This will remove any programs and files installed from the DE10-Standard image

1. Install the "Linux LXDE Desktop" for the VEEK-MT2S board found here. *https://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&CategoryNo=1&No=1098&PartNo=4*
   Recommended to install the "VEEK-MT2S CD-ROM" from above link to study on-board example programs from your PC rather than through PuTTY

2. Insert the SD Card into your computer (there are adapters in the DE10 box if your computer doesn't support microSD)

3. Open Win32 Disk Imager and in the "Device" dropdown select the drive of the SD card defined by your computer ("E" in this case)

4. For "Image File" select the .img file **unzipped** from the previously installed SD Card Linux Image

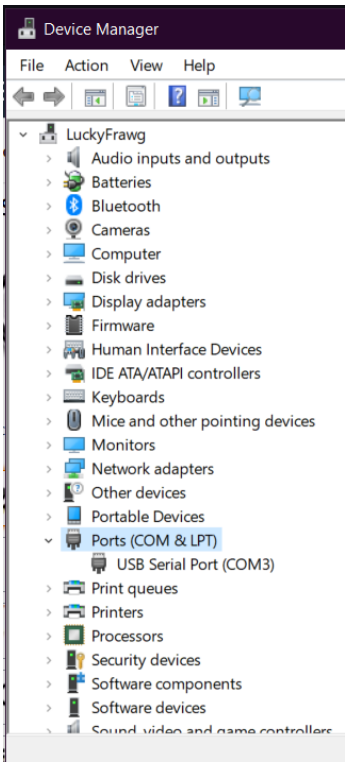
5. Click "Write" and wait for it to finish
   - Confirm that writing to a physical device can corrupt it
   - Should take ~ 10 mins
   - Ignore any file explorer windows that pop up saying to format the disk

6. Eject the SD Card and put it back in the VEEK-MT2S board

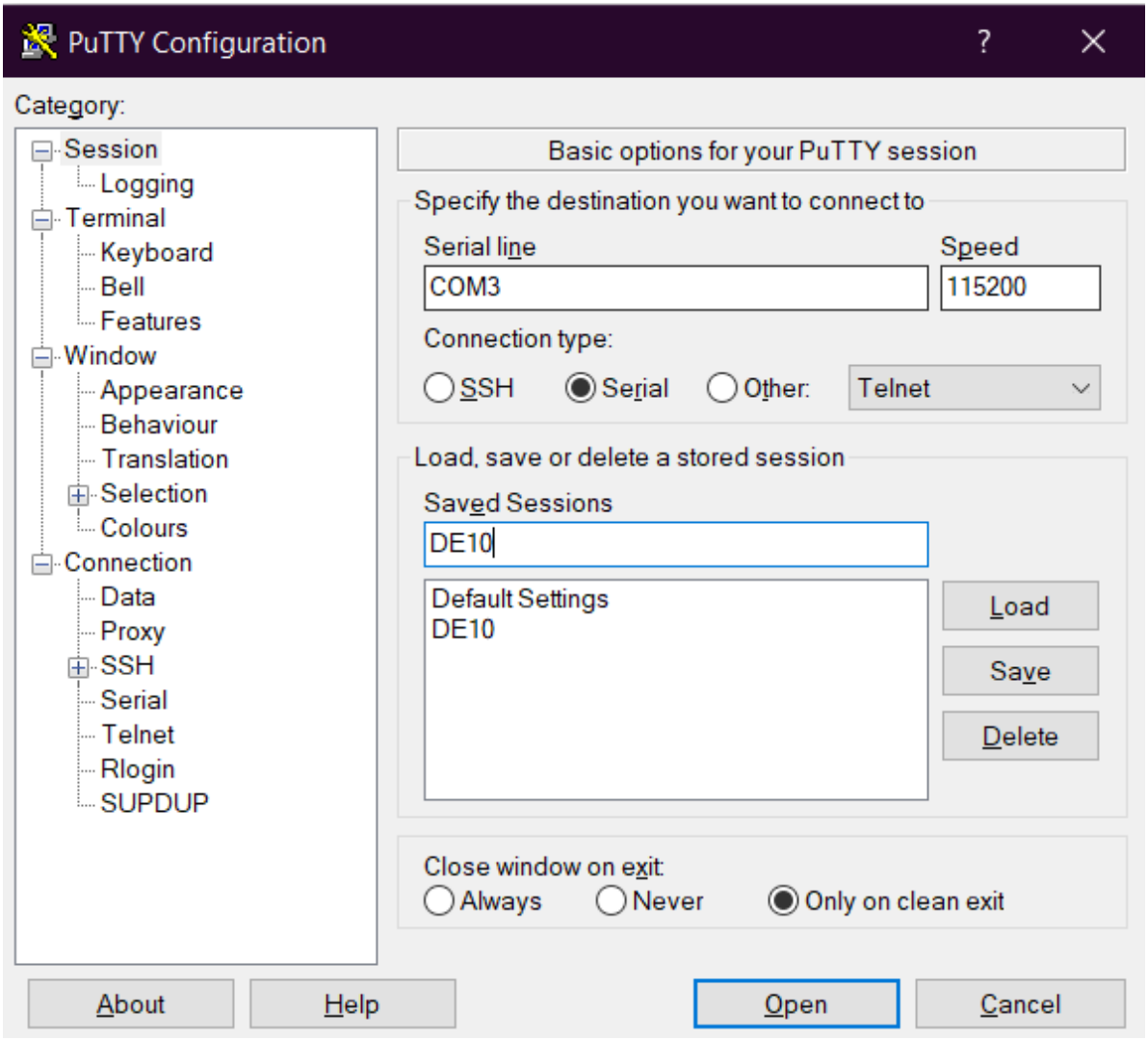
Connecting the RS232 Port/Establishing Connection from PC to Board:

Make sure to use the Mini-B to USB cable to connect the board to the PC

1. Set the MSEL Switches to “010100” to boot Linux to the LCD Screen
2. Press “Windows Key” + “X” → Device Manager → Ports(COM & LPT) – about midway down → expand it and note the name of the serial port (COM?)
3. Unzip and launch the FTDI driver setup, click next until successfully installed
4. Retry above and double check you are using the Mini-B cable and that the connections are secure



5. Open PuTTY, on the main screen change the Connection Type to Serial, change the Serial Line to the previously noted serial port name above (COM3 in this case), and change the baud rate to 115200
    - This will have to be done each time you want to establish connection from the PC to board.
        - Recommended: Name a “Saved Session” and save it to easily set the settings for future assignments.



6. Connect the DE10 board and click “Open”, it should open a terminal window

7. Press the red button on the board to power it on, you should see the LCD screen on the back light up, show the camera for a second and then the desktop should be loaded on the screen which can be interacted with through the touchscreen

You may need to stop the 5 second timeout on the initial u-boot menu like last assignment and run the commands “setenv eth1addr 00:11:22:33:44:55” and “saveenv” again

Installing Necessary Image Processing Libraries

- This assignment is essentially building upon the communication between the VEEK-MT2S Board and computer that was established in the previous assignment to permit transmitting and displaying images onto the board.
- The receiver Python/C++ program on-board should decode the data being received through the UDP socket from the Qt GUI and perform specific operations on that (receiving original or overlay image, adjusting brightness/contrast of the image(s), and turning the overlay on/off)

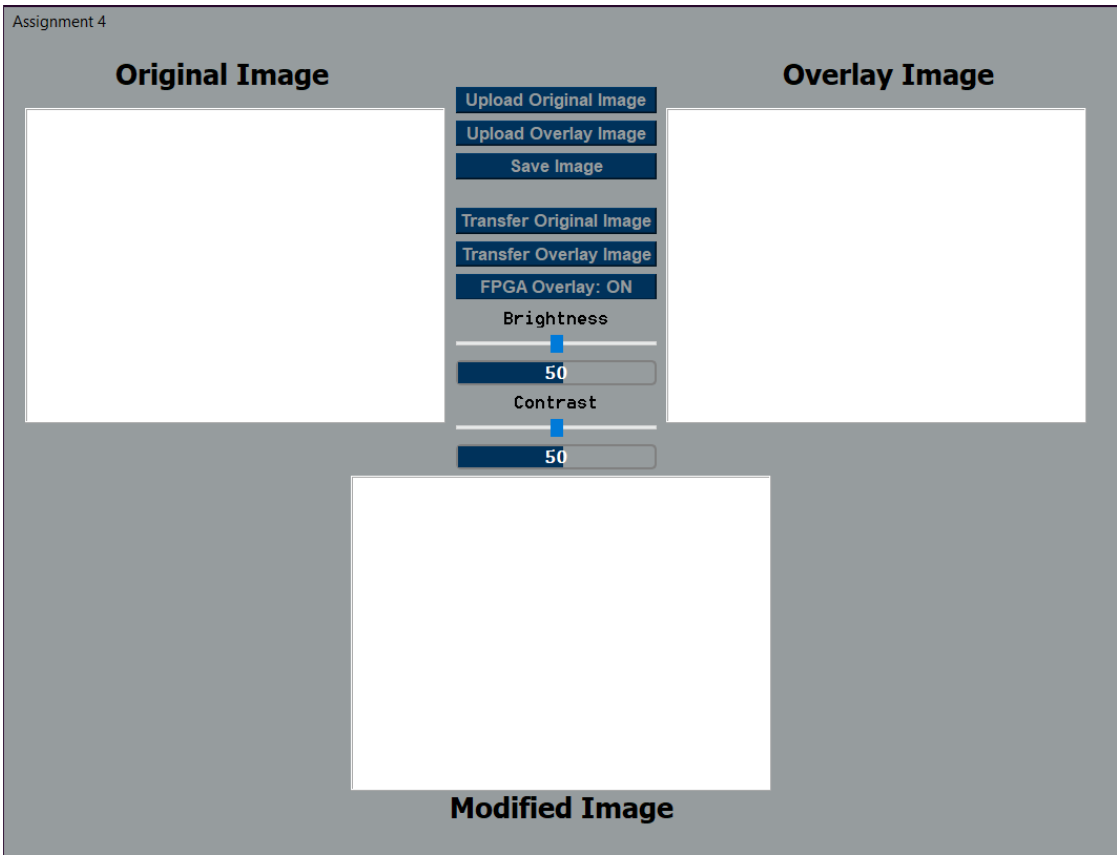
1. Send "ifconfig" to obtain the IP address of the board like last assignment, this will be necessary for the UDP protocol to transfer image data and should be adjusted in both programs each time the IP address changes
   - Any image data sent through a UDP socket must be divided into sizes of 1024 bytes since the UDP socket buffer supports byte transfer up to 1024 bytes.
   - Consider how the data should be distinguished between brightness/contrast/pixels/etc

2. Since this assignment is largely dependent on image processing, there are multiple libraries that can be used, the commands for some of them which can be utilized to meet the requirements are listed below:
   - apt-get update
   - sudo apt install python-opencv
   - sudo apt install python-pil
   - sudo apt-get install python-tk
   - sudo apt-get install python-imaging python-pil.imagetk
   - sudo apt-get build-dep python-imaging
   - sudo apt install imagemagick
     - » This installs a default image viewer as our Linux installation does not have one and is required for the pil/opencv implementation
   - For sanity sake, it is suggested to install all the libraries, research into each library and decide which would work the best with your design

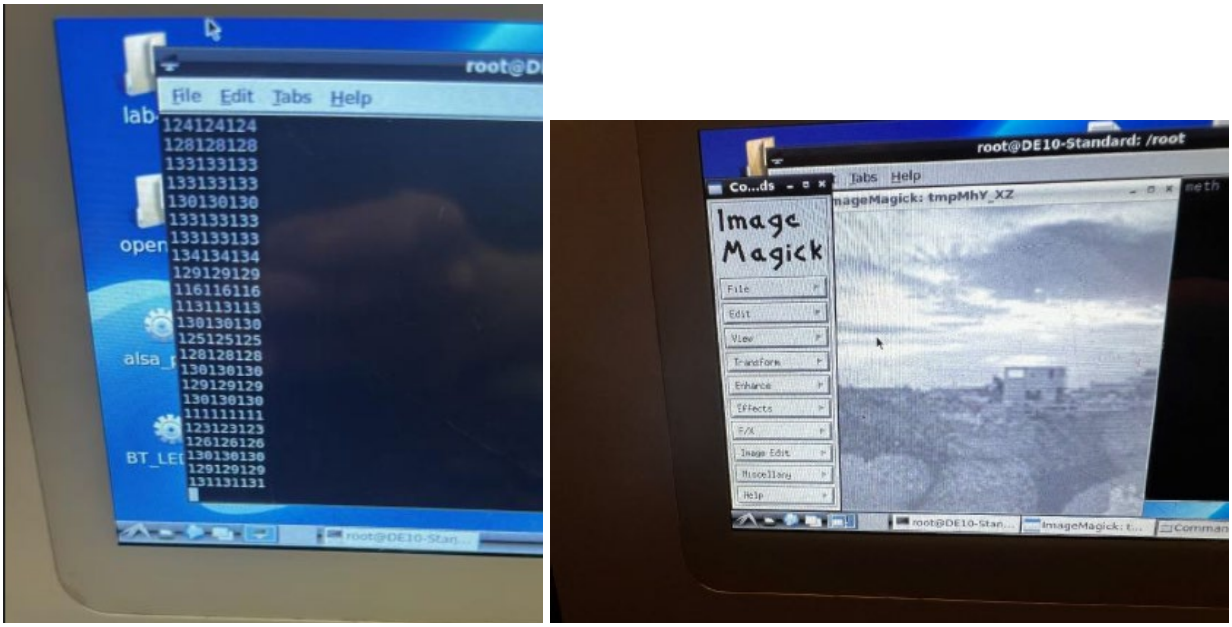
Each should bring up a prompt asking if you want to continue, type "Y" and it should connect to ports.ubuntu.com and begin installation of each package.

Running Programs On-Board

- Since this assignment requires the use of the LCD screen and our computers cannot access the physical peripherals on-board with a Serial connection through PuTTY, it is likely that a USB keyboard must be used to run the commands directly on the board itself
- It is possible the SSH connection method listed above "Re-flashing the SD Card to VEEK Image" will work as an alternative, but otherwise…
    - Use the touchscreen on board…
        - Click the menu in the bottom left corner → System Tools → LXTerminal
        - Using a USB keyboard, you can run your modified Python/C++ Python program with the same commands as before, "python lab4.py", after adjusting the IP address if necessary
- I highly recommend reviewing the manuals you may have downloaded above with the Linux desktop image
    - VEEK-MT2S_v.1.0.3_SystemCD/Manual
    - Additionally, researching image processing will prove useful and greatly benefit in completing the final assignment.
        - https://www.geeksforgeeks.org/python-opencv-cv2-imshow-method/
        - https://www.dfstudios.co.uk/articles/programming/image-programming-algorithms/image-processing-algorithms-part-5-contrast-adjustment/
- An example of how imshow() is used in an example program on-board can be found at VEEK-MT2S_v.1.0.3_SystemCD/Demonstration/SoC_Advanced/OpenCV/example
- On the Qt side, I recommend researching QStringLiteral, QBuffer, and QByteArray to implement image data transfer from the GUI to the receiver programs on-board
- On the Receiver side, I recommend looking into b64.decode or other data decoding functions, as stated previously, you must think of transferring the data at once and having the receiver understand what the numbers are (ex. 1st 3 digits corresponding to brightness, 2nd set of 3 digits correspond to contrast, transferring sets of RGB pixel data, etc)



Qt GUI Example



Python Receiver Program Printing the Decoded Data and Displaying Picture