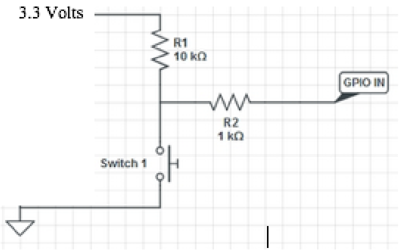
Homework 2

1. What is the difference between a Linux pipe and a FIFO (or a named pipe)? What can you do with a FIFO (named pipe) that you cannot do with a pipe?
2. List the key differences between using ‘bash’, ‘source’ and executing directly using ‘./’ when running a shell script. Your answer should specify how they interact with the shell's environment, their required permission and whether they create a new process.
3. List all the pins on the R-PI GPIO connector used by the piTFT screen. Explain what each pin is used for by the piTFT. Hint: read through the piTFT information at adafruit.com
4. For the R-Pi 4, Model B, list all possible GPIO pins that may be used for projects and labs. Identify the maximum set (when not using any special functions). Also, list the minimum set, when all special functions (and the Adafruit 2.8-inch piTFT) are used.
5. In Lab1, a shell script named “start_video” was created to run mplayer and video_control.py by using a single command. What is the correct ordering of operations within this script for this shell script to terminate correctly? Comment on the order of the calls within the script as well as which operation should run in the foreground, and which should run in the background. Hint: Correct shell script operation should return to a command prompt (the $ in a command window) if everything completed in the correct order.
6. If you run the date command on the ECE5725 server, the date and time are accurate. The RPi does not have a battery-backed real-time clock so how does the RPi maintain accurate time? When would it be appropriate to add an external, battery-backed real-time clock to the RPi?

7. For the following RPi GPIO circuit, describe why R2 is necessary in the figure:



Describe a possible 'software situation' that would damage the GPIO without R2. How does R2 prevent the problem? Why is the value of 1k ohm selected for R2?

8. In Lab1 and Lab2, you use a python code video_control.py, for example, to respond to buttons connected to the RPi GPIO pins. video_control.py passes these events to an instance of mplayer, controlling the video under playback. The two processes communicate using video_fifo. mplayer and video_control.py are launched from the bash script, start_video.

Describe what would happen if start_video is run WITHOUT first creating the fifo video_fifo. In this special case, if you press a button, what it the response of video_control.py? What is the response of mplayer? Will video_fifo be correctly created? Will control of mplayer be correct? Hint: Try this experiment on your pi or the class server!

9. While using the PiTFT buttons to control video, you may have encountered a problem with some of the buttons. It might appear that you had several 'hits' of the buttons when you pressed the button only once. There are several causes for this issue; what is one problem that could cause this issue? (hint: there are at least three possibilities observed by teams in the lab and discussed in class).

10. Describe pygame screen elements including a surface and a rect. How are these used to animate an image? Draw a step-by-step diagram illustrating the process you would use in PyGame to animate 2 frames of on the PiTFT.