ECE 373 Assignment #4
Spring 2022

Ticking away the moments...

Once the basics of a driver are in place, we can continue to add more features. This week, we'll add a bit of code that has the LED blinking on a timer. From HW 3, you should already have the LED control connected to the cdev callbacks, so most of the work is already done. Here, the aim is to have a driver that blinks the LED as long as some user process has the device control file open.

Kicking around...

Here are your requirements:

a) When the driver loads, it creates the **/dev/ece_led** character device file (no mknod this time) and prints to the system log that says it was loaded (maybe dmesg?). It also checks for a module parameter **blink_rate** that gives a new default blinks-per-second rate, otherwise it has a default of 2.

b) When a user program opens the device file, LED0 starts to blink on a 50% duty cycle at the given rate per second. This blink should be controlled by a timer object.

c) If a new value is written to the module parameter by writing into the parameter entry in **/sys/module/<driver_name>/parameter/blink_rate**, the blink rate will change.

d) If the user program reads the device file (not the sysfs file) it should be given the current blink rate integer.

e) If a positive integer is written to the device file, the driver should use that value as the new blink rate, just the same as in (c). Note, this is the `/dev` entry, not the sysfs file.

f) If the data written is not a positive integer, the write callback should return the error -EINVAL. Also, make sure nothing bad happens if the program writes a 0.

So you run and you run...

Turn these materials in to Github Classroom by **Friday**, **6-May-2022 at 11:00pm:**

1. All the code you used (including makefiles and scripts)