

ECE 373 Assignment #4 Spring 2018

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 last week, 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 control file open.

Kicking around...

Here are your requirements:

- a) When the driver loads, it creates the `/dev/ece_led` character device file and prints to the logfile that says it was loaded. 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 the network port's 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 char dev file it should be given the current blink rate integer.
- e) If a positive integer is written to the char dev file, the driver should use that value as the new blink rate, just the same as in (c).
- 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 before or at the start of class on **Monday, 7-May-2018**:

1. Source code to your kernel module, plus your kernel module Makefile.
2. Your userspace program. This does not require a Makefile, just the source code.
3. A note from someone (anyone, but Brett is a good choice) saying they actually saw it work.

Thought I'd something more to say

If you understand how all of this works, you'll probably do just fine on the midterm.