Lab 3: LKMs and Shift Registers

Overview:
So far you have been working with peripherals through prepackaged system drivers. Your LCD display code from the previous lab could be described as driver software for the LCD. However, it was in some ways very inefficient. Not only did you have a large number of GPIOs devoted to a single peripheral, but the software your driver used to access the physical hardware needed to go around a long, complex chain of other software to do the job.

The primary function of this lab will be to get around all that and build a single kernel-mode driver which accesses fewer GPIOs to do the same job, and does so in a highly direct manner. The result will be driver code which is more efficient and which more closely resembles actual industrial software drivers.

Using a Shift Register to Eliminate GPIO Usage:
GPIOs are a precious commodity in a digital system. Any time a task starts to use up a lot of them, someone is likely to suggest a shift register. A shift register may be used in either serial in, parallel out or parallel in, serial out, and as such can be used both to cut down on input and output pins. For example, if you need to write to 200 LEDs, you could make a 200-bit shift register and control all of them with a mere 3 pins: clock, data, and latch. Your task will be simpler: you just need to replace the GPIO usage from Lab 2 with a shift register, freeing up a lot of pins in the process!

Kernel Mode:
This part is harder, you'll need to write a kernel mode driver or LKM which will allow you to write software which mimics the driver you wrote before, but which uses the shift register, and does so from the kernel. To help you with this, we have provided some documentation on the sample case where you were doing Lab 2 with LKMs.

More Complexity:
In addition to paring down the expensive GPIO usage of the LCD display, you should try adding something interesting to the mix. There are a lot of extra peripherals we haven't used yet, and you will now have more free pins to work with...