

Dark side of the light



I was trying to get rid of the last nasty bugs in my electronics system. My company builds laser systems, and this system was the latest in a series of small, handheld laser designators for a military customer. I had designed a small control board to operate the laser, and the software was giving me a lot of trouble. I implemented background interrupts for timing-driven control loops, user-interface devices for control and configuration, and a data-communications SPI (serial-peripheral interface) connecting the microcontroller, an FPGA, and a few peripheral devices. Getting everything to behave had been difficult, but I thought I had solved most of the problems and believed I was in the home stretch.

I was wrong. Some intermittent behavior that I had attributed to earlier bugs refused to go away, and I was under pressure to finish the system so we could package and ship it out for a demonstration. I would start the device up, and, within minutes, some odd behavior would begin. An LED that

displayed the operating mode of the controller would suddenly start blinking in a pattern that didn't correspond to any normal state. More alarmingly, the system would sometimes emit laser pulses when it wasn't supposed to. The variety of bad behaviors indicated a problem with the SPI, which was responsible for programming the FPGA registers that we used to configure and operate the laser.

I was resigned to having my electronics technician tack-solder a bunch of leads to various SPI-related signals and connecting them to a digital oscil-

loscope to capture and analyze the SPI traffic. If I could find a corrupt message, I could trace it back to its source code and find the bug. There were a lot of SPI messages routinely zipping back and forth on the SPI bus, so I knew it would be a search for the proverbial needle in a haystack.

The morning after my technician had prepared the board for its date with the oscilloscope, we started to talk about the debugging plan. He told me that he thought he had identified the source of the problem. "It's the lamp we're using on the lab bench," he said. "The light is giving off noise and messing up your software." I was excited to learn he had a theory, but its unlikelihood immediately let me down.

"It's not the light. How could the light be causing the software to get confused?" I said, somewhat scornfully. I turned on the light and shined it at the board, and the LED pattern immediately changed from normal to anomalous. I was stunned. I put a piece of paper between the light and the board, cycled power on the board, and watched the LED blink happily. I removed the paper, and the LED pattern went bad again.

By cutting a hole in the paper about the size of an IC, I was able to selectively direct the light onto each component on the board and soon identified the problem: Light shining on a Bluetooth-interface IC caused the corruption. The Bluetooth chip was connected to the SPI bus for programming, although the connection was unnecessary because the chip's default settings worked fine. I solved the problem by cutting the SPI lead connections to the Bluetooth part, and the controller buzzed along peachily ever after. **EDN**

Edward Sullivan is an engineer at Fibertek Inc (Herndon, VA).

➦ Go to www.edn.com/090806tales and click on Feedback Loop to post a comment on this article.