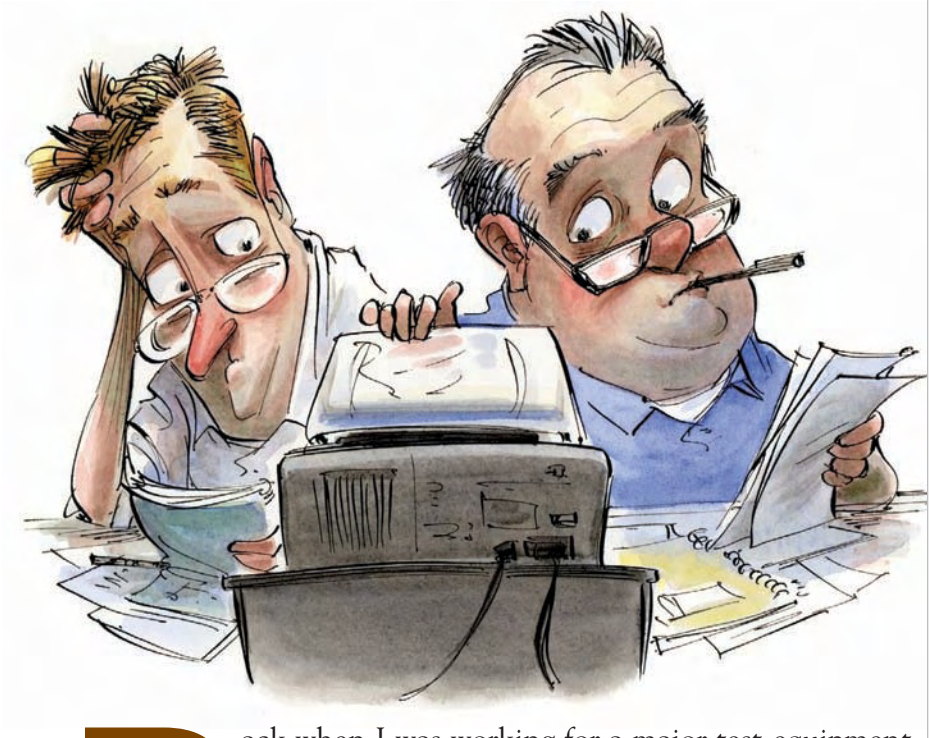
BILL FURCH • FUTUREPLUS SYSTEMS CORP

A probe without a paper trail



Back when I was working for a major test-equipment manufacturer, another engineer and I were asked to design and assemble a remotely controlled test system for data-acquisition probes. The system contained signal sources, voltmeters, frequency counters, and other test equipment that could communicate over IEEE-488 (HP-IB).

The controller was one of the first desktop computers, with the program stored on a cassette tape and written in BASIC. We had 8k of memory for storing the program, which was compiled, or "interpreted," each time it was run. With no memory space to spare, we didn't spend much time commenting our code.

The system ran a series of electrical tests automatically on each data-acquisition probe as it came off the assembly line. The operator—a skilled assembly worker rather than a fully trained technician—would plug the probe under test into the test system, initiate the program, and then simply watch as the program caused the signal and voltage sources to be applied to the probe, monitoring the results acquired by the voltmeters, frequency counters, and so on.

When the desktop monitor indicated that a probe under test had passed, the operator would box up the probe and send it to final packaging and then into stock. When a probe failed, the desktop computer printed out a short message indicating which test had failed and what the failure parameters were. The operator would tear off the failure report from the printer (a thermal model with a paper roll), staple the message to the probe, and hand off the unit to a senior technician for debug and repair.

We tested thousands of probes on this system, finding and fixing problems as they arose, until we began to think we had troubleshot every possible problem. One day, though, the operator said she was having a problem getting some of the probes to complete the tests; the program would get to a certain point and then "hang." I went out to the manufacturing line and verified the problem. Because we were running so many tests automatically, it was difficult to test one of the probes manually. I pored over the program—which, you'll recall, didn't have a lot of comments—trying to remember what each section did.

After a couple of hours, I approached the engineer who had co-designed the system for a fresh perspective. He came out to the manufacturing line, and we both sweated for a couple more hours. The manufacturing manager came by a few times and glared at us for having shut down his test line. It was a Friday afternoon, and neither of us looked forward to coming in on a Saturday.

Suddenly, the other engineer looked at the desktop printer and said, "What does that amber light mean?"

The printer had two LEDs—one green, for power on, and another, unlabeled indicator that I had not noticed or cared about until now. My colleague reached over and opened the paper-access door, revealing a cardboard tube that had once held paper.

We had found our "intermittent" problem: When a probe under test passed, the computer didn't need to print a failure report; when a probe failed, the computer would try to print out the failure data but couldn't. The printer would send the computer an interrupt reporting that it was out of paper, but we had failed to include the scan for that interrupt in the program.

We looked at each other. We had designed a sophisticated automated test system and had used it to test thousands of probes successfully. But we hadn't accounted for the simple inevitability of the printer running out of paper.

We revisited the program and found a bit of memory space to accommodate the "out of paper" interrupt. Then we told the manufacturing manager we had his test line back up, and we went home that afternoon looking forward to a peaceful weekend away from the office. EDN

Bill Furch is vice president of marketing and sales at FuturePlus Systems Corp. He holds a bachelor of science degree in electrical engineering from the University of Denver.

DANIEL VASCONCELLOS