BY ROBERT CRAVOTTA • TECHNICAL EDITOR

# Version inquiry

DANIEL VASCONCELLOS

I was supporting the system-build and -integration effort, in the field, of an R&D flight system. The system was an iterative evolution of a quick and dirty proof-of-concept vehicle that the team had demonstrated several months previously. Most of the boards, harnesses, and assemblies were handcrafted one- or two-offs (prototype quantities). The use of so many custom boards complicated the system-integration effort because anomalous behaviors encountered during the integration effort could reasonably be the result of flaws in the hardware or software design, build errors introduced during the hardware fabrication, or flaws in the test-setup configuration. In any case, it was critical for us to explicitly determine the cause of each anomalous behavior because the prototype system would be able to support only a single shot at the final test run.

To support parallel development of the software with the hardware build, we built a few of each of the electronics-control boards. At a minimum, we were developing the system on partially built systems—one in the lab and one in the field. Although this approach allowed us to perform more of the development effort in parallel, it increased the complexity of managing the configurations as we discovered and corrected flaws in the design or building of the system. In general, practicing version control of the boards and components minimized out-of-sync problems.

On one occasion, though, the version-control effort for the multiple development systems failed and cost the development team considerable time to troubleshoot. The design team predominantly comprised hardware designers, and the team considered replacing a physical PROM chip on the controller board as sufficient to manage the version of the software loaded in the system. The flight-control software had no user interface; it could control the flight vehicle and talk to the ground-control system through a serial link. Because the PROM sported a handwritten label that identified the software version loaded in it, the design team had not felt it necessary for the system software to include the ability to tell the ground-system controller what version of the software was loaded and executing on the flight system.

In this situation, the field system was exhibiting an intermittent anomalous behavior. The successful operation of the system in the lab suggested that we had a problem in the test-setup configuration. After many failed attempts to discover the problem in the test configuration, the design team tried to convince the team in the lab that the problem was not the test configuration. At some point in the troubleshooting effort, I asked whether the software loaded in the PROM could be an earlier version than that loaded in the lab. The label on the PROM indicated otherwise, but the intermittent-failure behavior suggested that we had a previous version of the software loaded. After we shipped the field board to the lab, we were able to duplicate the problem in the lab setup. It turned out that we had loaded the PROM with an earlier version of the software but had incorrectly labeled it.

At this point, we added a query function to the flight-system software so that the ground system could confirm what version of the software was loaded. One thing that contributed to the prolonged effort to correct this problem was that none of the system people who worked on the first iteration of the system were present in the field during the building of the second system. The system people on the second system were new to the project. The people with the experience to recognize the earlier version of the software were too far removed from the integration effort to recognize that a version mis-synchronization had occurred. This situation exacerbated an otherwise-simple-to-solve problem because of a break in the field-integration team's experience with the system. EDN