Year 2000 Leaves Programmers Unruffled

Embedded Systems Largely Immune to Angst Surrounding New Millennium

by Jim Turley

While the computer industry at large and the general press are treating the year 2000 as the Apocalypse, a number of designers and programmers of embedded systems are largely ignoring the coming millennium. Or, more specifically, their systems are ignoring it. Far from failing catastrophically on 1/1/00, many embedded systems will be blissfully unaware of the change of millennium (or, for the detail-oriented, the last year of this millennium).

The concern centers around the fact that most computers and microprocessor-based systems do not store more than the least-significant two decimal digits of the year (e.g., 97 instead of 1997). When 2000 arrives, many systems will wrongly perceive the year as 1900, causing comparisons that involve dates to produce incorrect results. The canonical examples are insurance systems that believe policy holders haven't been born, or banks that miscalculate interest schedules for payments received 100 years in advance.

Shortsighted Chip Designs at Root of the Problem

Most real-time-clock chips (as well as microprocessors and microcontrollers that include an RTC) store the date, month, year, hour, minute, and second in individual bytes of nonvolatile memory. Except for the year, none of these values can ever exceed 59, so one byte provides ample accuracy. Values are usually stored in binary; some RTCs store time and dates using BCD notation for programming convenience. Either way, the concept is the same.

For single-byte years, the maximum value is 256 (or 99 for BCD format). This is generally interpreted as the last two decimal digits of the actual calendar year, in other words, the number of years since 1900. Unless the software is designed to check for "illegal" dates, once the counter rolls over, the year 2000 will look like the year 1900.

Some operating systems, including MS-DOS, maintain a separate software counter. The counter is incremented every day, with all dates measured relative to January 1, 1980. This simple counter will fail after December 31, 2099.

Reader Feedback

If you have experience with an embedded system that suffers from Year 2000 phobia, we'd like to hear from you. We encourage system designers, engineers, and programmers to write us with examples of problems. Address your letters to *editor@mdr.zd.com.*

Most Embedded Failures Benign Annoyances

While many embedded systems track the time, the date, or the day of the week, most ignore the year because it's not relevant to their operation. Elevator controllers, for example, often behave differently on weekdays than on weekends. They may also distinguish between business hours, evening hours, and lunchtime. They do not, as a rule, behave differently in different years.

One subtle failure that may appear in such systems is if the day of the week is incorrectly calculated. January 1, 1900 was a Thursday; January 1, 2000 will fall on a Saturday. For systems that calculate the day of the week using the calendar, rather than simply incrementing a day-of-the-week counter, confusion may result. In the example cited above, the elevator may follow its weekday programming rather than the weekend program—hardly a catastrophic situation.

Other systems that may be affected are traffic-light controllers, industrial-automation systems, and office heating, ventilation, and air-conditioning (HVAC) systems. These, too, may incorrectly use weekday presets instead of their weekend settings. Other embedded systems—such as those in automotive driveline, climate, or comfort systems—are extremely unlikely to be aware of, much less affected by, changes in the calendar. Moreover, even for those systems that are affected, the failure is quite benign. It seems quite unlikely that the elevator, traffic-light controller, or automobile engine would simply stop working, as they are explicitly designed to avoid such failures.

Maintenance Is the Larger New Year's Headache

Although these failures might be mere annoyances, they will ultimately require a patch or fix, and that may be the biggest problem. Embedded systems are, almost by definition, secluded, awkward to reach, and difficult to reprogram. For low-end microcontrollers that have all their program code stored in on-chip ROM, reprogramming is impossible. Field updates are not an option, because the program ROM is often permanently soldered into the system, making it neither reprogrammable nor removable. Ultimately, replacing an entire printed-circuit board or assembly may be necessary.

The real issue, then, will be the mundane one of scheduling the maintenance, not the dramatic one of dealing with catastrophic fallout from system failures. Whatever the solution, it seems likely that many problems won't be resolved—or even identified—until well into the next decade. On the bright side, perhaps engineers in China will have a little extra time to deal with the problem: Chinese New Year 4700 isn't due to come until January 28, 2002. M