

# Re: careers in embedded engineering

---

*Source:* <http://coding.derkeiler.com/Archive/General/comp.arch.embedded/2006-04/msg00573.html>

---

- *From:* "larwe" <[zwsdotcom@xxxxxxxxxx](mailto:zwsdotcom@xxxxxxxxxx)>
  - *Date:* 12 Apr 2006 08:30:31 -0700
- 

Steve at fivetrees wrote:

Also agreed. Furthermore, my experience has been that PC programmers have far less of a clue about the rigours of true embedded design, which includes

God, yes. From section 4-1, in chapter 4 ("Teaching Yourself, Top-Down (Large Embedded Systems)"):

"The previous chapter dealt with people who have hardware experience and want to start learning about microcontroller programming, or a "bottom-up" approach to embedded engineering. At the other end of the spectrum, we find people who have considerable experience programming application software in high-level languages, and who now want to extend their reach into embedded systems. These people will typically have what I would describe as "IT" qualifications (database programming, HTML design, Java development and so forth), rather than engineering experience. Computer science majors quite frequently fall into this category.

There is an immediate difficulty in adapting these people to embedded environments, and this is that pure software projects for mass-market operating systems (Windows, MacOS and so forth) are typically designed with an acceptance of enormous variations in performance margins due to variations in customer hardware. Furthermore, PC hardware is sufficiently expandable and inexpensive that it is realistic for an application software developer to require that the user provide special hardware features such as additional memory, 3D accelerated graphics, and so on. Neither of these two assumptions are remotely acceptable in embedded environments.

The embedded software developer must be able to:

- a) characterize and control the resource utilization of the software exactly. This includes being able to specify how much RAM and secondary storage space the software will require under all conceivable execution conditions, with explicit safeguards in place to prevent the software

Re: careers in embedded engineering

from overrunning those limits under unusual input circumstances. In most cases, it is also necessary to provide some guarantee of real-time performance (though the limits here are