

# Re: real time embedded system requirement on processor architecture

---

*Source:* <http://coding.derkeiler.com/Archive/General/comp.arch.embedded/2008-03/msg00287.html>

---

- *From:* Tim Wescott <[tim@xxxxxxxxxxxxxxxxxxxx](mailto:tim@xxxxxxxxxxxxxxxxxxxx)>
  - *Date:* Thu, 06 Mar 2008 22:54:32 -0600
- 

On Thu, 06 Mar 2008 17:35:43 -0800, joggingsong wrote:

Hi,

For a real time embedded system, is there any requirement on the processor architecture to meet the real time requirement?

"Real time" is pretty much accepted to mean that if an event happens, and your system doesn't respond to it quick enough, then the system isn't meeting spec. Some folks separate this into "hard real time" which means that if even one deadline is missed by even one nanosecond then disaster strikes, and "soft real time" which means that some deadlines can be missed, with some sort of graceful failure that gives degraded performance long before things crash and burn.

So the only requirement on a processor architecture to be suitable for some real time application is that it be deterministic, that is that you can always calculate the maximum possible time for the processor to carry out a given task. Once you've established that, then you can determine if it'll be fast enough to meet your particular real-time requirement.

Does realtime system depend on realtime OS to meet the realtime requirement?

In general, no, although for large complex systems a real time OS can make the development easier. (note that a real time OS also makes the development more complex and the problems more obscure -- developers of really critical applications tend to shy away from RTOS's for this reason).

Regardless of what framework you're doing it in, real-time software development requires that you pay attention to details from one end to the other. It's not the right place to jump in and play without knowing what you're doing.

Re: real time embedded system requirement on processor architecture

—  
Tim Wescott

Control systems and communications consulting

<http://www.wescottdesign.com>

Need to learn how to apply control theory in your embedded system?

"Applied Control Theory for Embedded Systems" by Tim Wescott

Elsevier/Newnes, <http://www.wescottdesign.com/actfes/actfes.html>

.