Embedding Systems

Alix Paultre

The impact of convergence on the design community is not just increasing in scope, it's also speeding up. Consolidation of functionality and integration of subsystems has meant that the very definition of what an embedded system is becomes more elusive. The problem begins at the bottom of the circuit food chain, where the situation has gotten to the point that "ordinary" semiconductors like MOSFETS and diodes are lumped in with passives in the new product sections, and even complex devices like switches and op amps aren't even really considered ICs anymore. On the flip side (no pun intended), what used to be considered a microcontroller or microprocessor is now more a computer System-on-Chip than what we used to think of as a 'normal' IC. As complex technologies get mature and get integrated into more complex products, the vey way we think about system architecture has to change.

For example, I recently attended the Embedded Systems Conference in Boston, and the exhibits and activity there underscored both the tremendous challenges facing today’s design engineers due to disruptive technology migration and the great opportunities it presents. In every meeting, at every booth, in every product demonstration was the reminder that hardware design will never be the same again. Nobody making processors now would even think of not offering at least one device with integrated controllers for displays, touchscreens, power management, and lots of I/O. some of the more impressive in the field only require power, software, and a display with a touchscreen to become a product.

One of the greatest challenges today in embedded systems (and the products they appear in) involve communications and interface with the user and the cloud. Engineers must design products that interact with the world on multiple levels using a variety of communications protocols with an assortment of system interfaces. As mentioned earlier, since many of today’s microcontrollers are really computer SoCs with all required subsystems for core features incorporated on-chip, communicating with other peripheral SoCs, the user, and the Internet to achieve its complete functionality set is critical.

Luckily there are lots of tools and resources to aidd the designer. From FAEs to White Papers, resources abound, and many hardware and software development

tools like LabView and the Beagleboard become so powerful that they become platforms for products in their own right. There are a surprising number of designs that go straight from the development board (or software design suite) right into producti0n today.

That smart washing machine, or refrigerator, or electric meter must not only communicate with the user, it must communicate internally between intelligent embedded systems, each providing a core required functionality (wireless, power, signal, interface) and a requirement for proper setup to ensure maximum performance. Beyond that level may be at least one and possibly several additional layers of functionality either enabled by supporting embedded systems peripheral to primary functionality or software-driven app-based features provided by the cloud.

This technology progression is incredibly empowering, as the tools and devices are so sophisticated today that an engineer with a good system architecture integration strategy can create almost any product with unheard-of levels of functionality. The key to remember is that with a proper perspective, you can get on top of the challenges and prosper in this disruptive environment.

Source URL (retrieved on *12/08/2013 - 11:49pm*):
http://www.ecnmag.com/articles/2011/10/embedding-systems