

IoT Frameworks/Development to Dominate Software Trends in 2016

Is this the year that the Internet of Things really takes hold? Advances in cloud technology and resolving security and quality issues in software could at long last tip the scales.

IT'S TIME TO STOP KICKING THE TIRES and take the plunge into the Internet of Things (IoT). Most IoT frameworks finally have some meat on the bone, and the plethora of options has become mind-boggling.

One major trend is the move to cloud-based development as tools become more complex, which increases setup and integration costs. Having someone else set up, support, and maintain the tools can offload these chores from a company, but developers need to be wary of issues like long-term support.

Also taking center stage are the matters of security and software quality. Though these areas have always been important, they're often neglected in the general embedded space due to cost and effort. Of course, no one wants insecure and buggy code, but where is the boundary between total security and bug-free code? Moreover, what actually gets delivered by most soft-

ware projects? For unconnected devices, security has become less of an issue.

SOFTWARE FOR THE INTERNET OF THINGS

The challenge with IoT has always been the large number of interfaces, connections, and entities involved in even the most basic designs (*see the figure below*). IoT offers many benefits and provides a potentially flexible environment, yet one must factor in its greater complexity with more potential points of failure, attack, and maintenance. The entire system needs to work together properly, otherwise problems become inevitable. And with possibly millions of end nodes and gateways involved, the ramifications could be catastrophic.

Organizations and vendors have been taking the wraps off IoT frameworks and environments, often expanding the func-



tionality, coverage, and partnerships. Many IoT standards organizations were closed societies made up of vendors looking to guide the design of the frameworks. This year they will be delivering their incarnation of these frameworks to developers, so that they can examine and utilize them to build IoT solutions.

The ability to create IoT solutions is not new, as many are being developed and deployed already. However, these cutting-edge implementations tend to be limited to larger companies or those willing to deal with emerging standards. It will now be easier for developers to deliver IoT solutions thanks to more-established standards and tools.

Nonetheless, one big issue continues to cast a shadow: The large number of choices in the IoT development space will limit interoperability.

WILL IOT BE SAFE AND SECURE?

Software quality and security are related because security breaches are often caused by bugs in the code. Both issues arise when it comes to IoT development because connectivity between devices opens the door to remote attacks.

Most IoT frameworks incorporate multiple security measures, from the use of Transport Layer Security (TLS) on communication links to secure boot. Encryption is a vital ingredient and more hardware with built-in hardware acceleration is being made available. Secure key storage and even anti-tamper support is becoming more common.

The challenge for developers continues to be how security measures can be incorporated into a design and subsequently contending with them, let alone developing the other functional features of an application. Framework and policy-based security support can help, but it doesn't eliminate security issues for application design. Beware of claims that a framework or system will handle all security needs. This is rarely the case, but it does make the job of securing a system significantly easier.

Lots of IoT development is being done in C, which has the reputation of allowing developers to shoot themselves in the proverbial foot. Adding security to the mix compounds the issue, as many problems often result from misunderstandings of how things work, or just due to carelessness. Many available tools, such as static analysis, often support standards like MISRA C++ and the new MISRA C:2012 standard.

Formal methods are moving out of areas such as military and avionics systems and into industrial applications. For example, companies in the transportation and automotive space are looking at programming languages, namely Ada and SPARK, to provide better development environments for embedded, real-time applications (see *"Best of 2015: CodePeer SPARKS Ada"* on electronicdesign.com). Functional programming languages that are also more amenable to formal proofs can be found in the cloud. In this space, they're being used in everything ranging from system control to financial calculations and analytics.

DEVELOPING IN THE CLOUD

Not everyone will jump to the cloud for development. Even so, vendors continue to push the technology, and developers are finding advantages in this approach.

There are literally dozens of cloud integrated-development-environment (IDE) systems/sites to choose from, and many IoT solutions providers now partner with some of these sites to augment their offerings. This follows from the success of collaboration and code-management sites like Github and Pastebin. Moreover, sites such as Cloud9, mbed, and even Arrow Cloud Connect provide embedded IoT support for the ARM Cortex-M0+.

Some, like Codevny, support online as well as workstation-based tools (e.g., Eclipse), taking advantage of cloud-based servers for compilation and build services. Wind River's Helix environment includes the Helix Lab Cloud, which allows developers to test their applications on hardware that resides in the cloud.

The advantages remain significant by offloading installation, management, and support. Developers can have access to the latest software. However, developers must be cognizant of issues such as maintaining toolsets and runtime support for specific versions of their own application.

Online development, which does give more control to the vendor, is not for everyone. This is especially the case when control over the code and tools is critical or required. The quality and performance of the Internet connection is also important. Cloud operation obviously works best with reliable, high-speed connections.

Cloud-based solutions tend to excel in collaboration support; such capability is often crucial for IoT applications spanning multiple hardware nodes. This type of collaboration tends to be a bit more difficult to integrate when the tools operate independently of each other.

MORE TOOL INTEGRATION FOR DEVELOPERS

Tool integration and support is essential to cloud-based environments, but vendors will not ignore conventional development toolchains. These are being enhanced well past providing IDEs and compilers by including operating systems and stacks. Though available in the past, the level of integration and support is now becoming much better. It's one way for vendors to lock in their customers by providing a better alternative than the competition.

Sometimes this integration is more focused. Take, for instance, Renesas' Synergy, which specifically targets its Synergy line of Cortex-M microcontrollers (see *"Dev Kits: Getting Synergy"* on electronicdesign.com). It's built around Express Logic's Thread-X RTOS. Microchip's Harmony is a more inclusive framework that features multiple third-party components (see *"Interview: Rich Hoefle Presents Microchip's Harmony"* on electronicdesign.com).

There are advantages to vendors and developers for both approaches. Tighter integration allows for better support but fewer choices. Developers will have to consider the indirect costs of using these frameworks and tools, because there's really no such thing as a free lunch or IDE. It's just that the costs may be hidden elsewhere, e.g., higher hardware costs.

OPEN-SOURCE SOFTWARE AND IoT

The interplay continues on between open-source software (OSS) and proprietary software. OSS is playing a critical part in IoT because it generally induces more collaboration among participants negotiating standards.

Vendors can get a better handle on where OSS fits within their offerings, ranging from support for Linux to providing some or all of their software with an open-source license. It's still not an easy choice, though, and vendors need to determine where they can best gain their advantage and income. Luckily, a multitude of developers are willing to pay for support and integration.

Like security, though, developers and companies might be forced to contend with OSS licensing as more standards are

built around OSS. Developers may need to utilize tools like Black Duck Software Suite to track compliance, security, and governance issues related to an application (see *"Is it GPL if it Quacks Like a Duck?"* on electronicdesign.com).

Making the choice to go with proprietary licenses is often a preferred alternative. However, choices might be more limited, especially when addressing IoT that incorporates many more components within an overarching IoT application environment.

DON'T FORGET POWER-MANAGEMENT SOFTWARE

The need for efficient power-management hardware and software ranks high on the list of issues surrounding mobile IoT devices. It's cropping up in more sophisticated hardware with more granularity in terms of power control and the types of peripherals available. Just managing the large number of power modes and peripherals is pushing chip vendors to provide power-management frameworks that work with applications and operating systems.


Battery-operated IoT devices usually operate in cycles where most time is spent in a low-power mode. Developers will need to examine all of the options when trying to minimize power requirements, from using non-volatile storage like FRAM, to choosing chips with ultra-low power peripherals, to taking advantage of intelligent, configurable peripheral interconnects that allow on-chip devices to operate independently of the CPU.

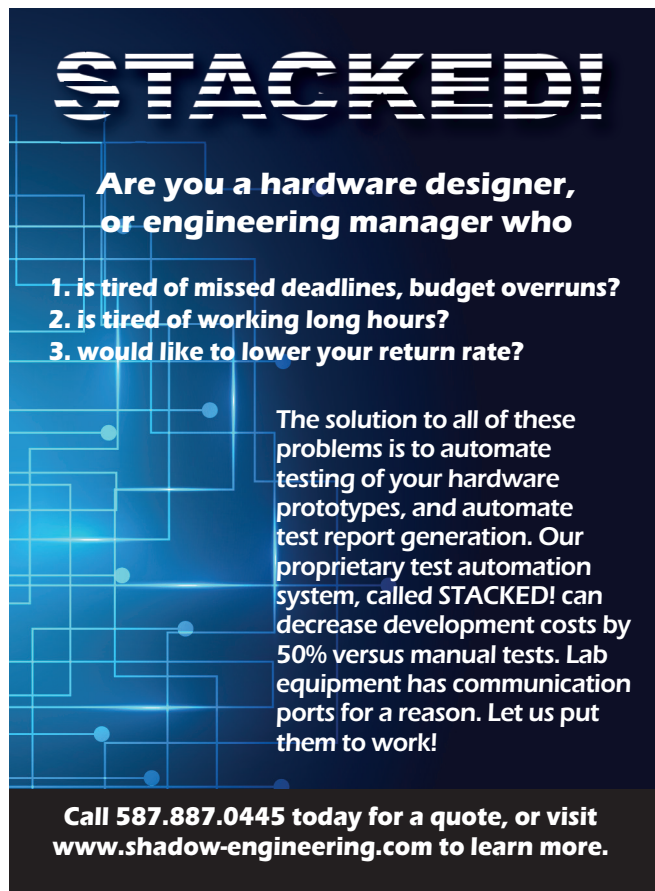
Keep an eye out for multicore, asymmetric microcontroller solutions that provide even more granularity for controlling power and performance. Although the downside will be a programming environment with heightened complexity, it should provide important advantages in both system isolation and security.

NOT EVERYTHING IS IoT

IoT continues to be a major theme for developers and vendors, but it's really just a part of the embedded software landscape. It just happens to be growing quickly and becoming more important to the long-term plans of many companies. It also makes companies think about where and how solutions will be monetized.

Still, IoT isn't everything or everywhere, and developers should not be espousing IoT solutions just because it's popular. On the other hand, many of the tools and support targeting IoT can be employed for more conventional types of embedded applications.

Cloud development tools work with the same efficacy for non-IoT development, too. Security hardware and software can play roles that are just as important for unconnected devices and networked devices that may not be part of a large cloud-based IoT solution. 



STACKED!

Are you a hardware designer, or engineering manager who

- 1. is tired of missed deadlines, budget overruns?**
- 2. is tired of working long hours?**
- 3. would like to lower your return rate?**

The solution to all of these problems is to automate testing of your hardware prototypes, and automate test report generation. Our proprietary test automation system, called **STACKED!** can decrease development costs by 50% versus manual tests. Lab equipment has communication ports for a reason. Let us put them to work!

Call 587.887.0445 today for a quote, or visit www.shadow-engineering.com to learn more.