

## M2M IWG Enables M2M Development With Open-Source Technologies

[Electronic Design](#)

[Benjamin Cabé](#)

Benjamin Cabé, Contributing Technical Expert

Fri, 2013-04-12 15:03

Today's machine-to-machine (M2M) communications market is held back by fragmentation in the industry, with too many basic functions remaining closed and proprietary. In an environment where M2M vendors develop their own proprietary implementations of the basic M2M communications stack, huge amounts of time and resources are wasted.

Why reinvent the same core functions over and over again for specific hardware and applications in ways that cannot be reused and repurposed? Why compete on something as fundamental as a basic M2M software stack? Wouldn't the industry advance more quickly if vendors agreed on a common stack and competed on hardware and software capabilities that are truly unique?

The Eclipse Foundation and Sierra Wireless established the M2M Industry Working Group (<http://m2m.eclipse.org/>) to address this problem and to develop a universal, open-source software stack for building M2M solutions. Now, the results of these efforts are bearing fruit. For the first time, developers can take advantage of an end-to-end open-source software stack for M2M embedded development, extending from the embedded device to the connection with the server infrastructure, with all of the tools necessary to develop M2M solutions.

This open-source software stack encompasses a large and growing set of open-source M2M tools and libraries, an embedded application framework on top of Linux and application programming interfaces (APIs) to provide a common platform for M2M applications, and a growing body of standardized and open-source communication protocols for embedded applications.

Together, these open-source blocks allow developers to create their own M2M applications and run them on various types of hardware for the first time. As more developers and vendors adopt and expand this open-source M2M platform, we believe the pace of M2M innovation will accelerate considerably.

### Related Articles

- [Use Lua To Develop M2M Embedded Applications](#)
- [The Future is Here...M2M. Machines are Taking Over!](#)
- [Design Considerations for M2M Products](#)

### M2M Industry Fragmentation

Most M2M technology solutions have been created as bespoke pieces of software, developed from the ground up for a specific purpose, usually with proprietary protocols, operating systems, and server implementations. In the early days of connected devices, this approach made sense. When M2M vendors and developers wanted to address a specific industry, the simplest approach was often to develop a solution optimized for that application. A provider of power meters, for

example, could offer an optimized solution by designing an M2M system built to meet the specific requirements of smart meter deployments.

However, this approach presents a number of limitations. The limited portability of such solutions creates impediments to developers and vendors, and ultimately to their customers. For instance, a solution optimized for power meters is difficult to integrate on a factory floor with other types of machine sensors, because those machines are also likely designed with proprietary protocols and technologies optimized for their specific application.

This fragmentation additionally prevents real interoperability between new solutions and those already deployed in the field—even within a single vendor, and certainly across different vendors. This makes it much more difficult to maintain and upgrade deployed solutions as M2M technology evolves.

Developers also have limited ability to reuse software. On one level, this means that far more time and effort are necessary to build a new solution than is the case in technology areas with more standardized software stacks, but the problems go even deeper than that.

Across the industry, embedded application developers' specializations have become so narrow that it has been difficult to establish a thriving community of embedded application developers who can continually share knowledge to expand their skills and marketability. This hinders the speed of advancement and innovation among individual developers and makes it harder for companies to find the developers they need with expertise in the specific languages, technologies, and APIs a given company uses.

If industry projections of 50 billion connected devices emerging in the next decade are correct, this will become a serious problem. There simply will not be enough developers with enough specialized expertise to meet the demand for new M2M applications.

### **Building An Open-Source M2M Platform**

Recognizing these issues, Sierra Wireless and the Eclipse Foundation, along with IBM and Eurotech, established the M2M Industry Working Group in November 2011. Under the umbrella of the Eclipse Foundation, the goal was to bootstrap an ecosystem for M2M developers that could provide all of the development tools, messaging protocols, and embedded runtime framework to support an end-to-end software stack for M2M development that was completely open-source and royalty-free.

Less than a year and a half later, the group has now established a complete, functional open-source M2M software stack and a common platform for developers to build embedded applications. This end-to-end M2M software stack is based on three open-source technology pillars:

- Koneki ([www.eclipse.org/koneki/](http://www.eclipse.org/koneki/)): The first pillar of the M2M IWG effort provides an ecosystem of tools and libraries for M2M developers, primarily in the form of an integrated development environment (IDE) for the Lua programming language. This includes debugging and simulation tools on a level that previously were available only for specific, proprietary hardware, if they were available at all.
- Paho: The second pillar of the M2M IWG effort, spearheaded by IBM, is the development of fault-tolerant, open-source M2M protocols for secure and bandwidth-efficient wireless communication. The Paho project ([www.eclipse.org/paho/](http://www.eclipse.org/paho/)) currently supports open-source implementations of common messaging and device management protocols. However, the project will continue to expand to support open-source implementations of various legacy and industry-standard protocols.

- **Mihini:** More recently, the M2M IWG created Mihini ([www.eclipse.org/mihini/](http://www.eclipse.org/mihini/)), the first open-source embedded application framework for developing embedded applications. Based on the Sierra Wireless ALEOS Application Framework, Mihini provides a broad set of functions, features, and APIs necessary to build M2M solutions. This includes remote device management over the air (OTA), OTA updates and maintenance, interaction with both physical assets connected to an M2M gateway and servers, and much more. These capabilities not only simplify development for seasoned M2M developers, they also can enable even non-experts in embedded systems to develop embedded applications relatively easily, for the first time.

### **Advantages Of An Open-Source M2M Software Stack**

With the introduction of the Mihini framework, the M2M IWG now has real source code and a concrete implementation of all elements of the software stack. As a result, developers now can take advantage of a complete, end-to-end open-source platform for M2M applications, from the embedded device to the connection with the server. This open-source platform presents a number of benefits:

- **Interoperability:** Any new M2M solution should be able to work with thousands of devices already deployed in the field. Open communication protocols and royalty-free implementations of the M2M communications stack can address this problem and potentially support new communication scenarios between devices that are not feasible today.
- **Expanding applicability:** As more developers take advantage of and expand this open-source platform, we expect to see entire ecosystems of libraries emerge that developers can draw on for ready-to-use implementations of various technology stacks. As the number of open-source libraries expands, the pace of M2M development and innovation will accelerate.
- **Increased hardware flexibility:** As more vendors adopt an open, standardized approach to M2M development, developers will have far more flexibility and choice in designing embedded systems. They will be able to pick and choose the optimal hardware for a given application in a way that is not possible today.
- **Accelerated innovation:** The more vendors and developers standardize on a basic software stack, the less time and resources they will need to devote to designing, testing, and updating discrete proprietary systems, and the more they will be able to reuse and build on their work. More than any other factor, this reusability will allow for the kind of experimentation and innovation necessary to meet the demands of the M2M market in the coming years.

### **Conclusion**

Participants in the M2M IWG have been presenting this platform at industry shows since the end of 2012, conducting live demos of a connected greenhouse using these open-source tools, open-source communication protocols, and even open-source hardware. The implication should be clear. This is not merely an idealized vision of the future. This is a tangible, functional development platform for working M2M applications that any developer can use right now.

So what comes next? Over time, we expect more developers and vendors to begin using and expanding on this open-source platform. The working group has proven that it's possible and has provided the foundation for a real open-source revolution in M2M applications. The more developers adopt and participate in this open-source M2M ecosystem, the more sophisticated, versatile, and useful it will become.



**Benjamin Cabé** is an open-source evangelist at Sierra Wireless. He has a longtime passion for Eclipse and its ecosystem and is a committer on several Eclipse projects and contributor to numerous other open source projects. In his day-to-day job, and as the chair of the Eclipse M2M Industry Working Group, he advocates the use of innovative technologies for the Internet of Things. He also co-leads Eclipse projects Koneki and Mihini. When not wandering on the Internets engaging M2M/IoT developers, he is building crazy machines with Arduino kits and Raspberry Pis. You can find him online on Twitter (@kartben) or on his blog, <http://blog.benjamin-cabe.com>. He can be reached at [bcabe@sierrawireless.com](mailto:bcabe@sierrawireless.com).

**Source URL:** <http://electronicdesign.com/communications/m2m-iwg-enables-m2m-development-open-source-technologies>