

# Koneki Open-Source Development Tools Simplify M2M Development

[Electronic Design](#)

[Benjamin Cabé](#)

Benjamin Cabé, Contributing Technical Expert

Mon, 2013-07-15 09:18

Machine-to-machine (M2M) communications is one of the most exciting and fastest-growing technology areas today, with a projected 50 billion connected devices deployed by 2020.<sup>1</sup> And yet M2M technology is still not evolving as quickly as it could because too many basic development functions remain closed and proprietary.

## Related Articles

- [M2M Client-Side Challenges Emerge In Mobile Embedded System Updates](#)
- [M2M IWG Enables M2M Development With Open-Source Technologies](#)
- [Use Lua To Develop M2M Embedded Applications](#)

Most M2M technology solutions were created as custom pieces of software, built for a specific purpose, usually with proprietary protocols, operating systems, and server implementations. This industry fragmentation prevents developers from freely reusing software and skills or from tapping into the knowledge of a large community of fellow developers working on similar problems.

Indeed, when it comes to M2M development, it barely makes sense to classify it as a single field. Developers may become experts in one vendor's development environment and hardware, but there have been very few areas where that knowledge or expertise could be ported from one closed environment to another.

The M2M Industry Working Group (M2M IWG – <http://m2m.eclipse.org>), established by the Eclipse Foundation and Sierra Wireless, aims to change this by developing a universal, end-to-end open-source software stack for building embedded applications. The M2M IWG effort encompasses several areas,

including:

- Koneki: a large and growing set of open-source M2M developer tools and libraries designed to simplify M2M development ([www.eclipse.org/koneki/](http://www.eclipse.org/koneki/))
- Mihini: an open-source embedded application framework on top of Linux to provide a common platform for M2M applications ([www.eclipse.org/mihini/](http://www.eclipse.org/mihini/))
- Paho: a growing set of standardized, open-source communication protocols for embedded applications ([www.eclipse.org/paho/](http://www.eclipse.org/paho/))

Together, these tools will help to standardize and accelerate embedded application development and, ideally, spur new innovation in M2M technology. By making it possible for virtually anyone to build M2M solutions, the M2M IWG hopes

to inspire a new generation of developers to turn their creativity and talents to building unique new connected applications. Koneki M2M development tools are the first and most advanced of these M2M open-source efforts.

### **The Pressing Need For M2M Development Tools**

As the first pillar in the M2M IWG effort and the initiative that spawned the entire project, Koneki aims to solve one of the biggest problems for M2M developers: a lack of effective tools for developing embedded applications.

Fundamentally, implementing an M2M solution means programming an embedded device—linking a sensor or other connected device to the Internet or another network. And historically, there have been no good, common tools available to simplify that process. Instead, developers have had to rely on low-level editors for basic code development or closed, proprietary integrated development environments (IDEs) built for a specific type of hardware.

For an application developer, especially one seeking to implement a basic connected solution for the first time, there was no way to reuse familiar programming tools and environments. Instead, building an embedded application meant installing and learning an entirely new set of complicated tools from the ground up. Thus, the birth of Koneki (and of the M2M IWG itself) can be boiled down to a simple plea: we need tools! We need to make it easier to develop, debug, and simulate embedded applications.

### **Koneki Tools**

Koneki provides an ecosystem of tools for M2M developers, primarily in the form of an IDE for the Lua programming language, built for the Eclipse open-source framework. This includes debugging and simulation tools on a level that previously were available only for specific, proprietary hardware, if they were available at all. By providing an IDE and a growing set of M2M tools for Eclipse, Koneki will make it easy for even first-time M2M developers to create embedded applications. Koneki includes:

- **Lua development tools:** Koneki development tools are designed to support Lua, a simple, easy-to-learn scripting language (see [“Use Lua To Develop M2M Embedded Applications”](#)). Seasoned developers can master Lua in a few hours. But even more importantly, by embracing Lua, embedded applications can become accessible to a much larger group of potential developers. For example, someone may want to connect a greenhouse to be able to remotely check temperature and humidity. This person may not be an advanced programmer. He just wants a simple way to connect some sensors to the Internet. With a Lua-based IDE running in Eclipse, he can write a few lines of script and get his application online. The Lua IDE provided within Koneki also includes advanced code completion features, debuggers, source control systems integration such as Git, and a variety of other functions that operate transparently under the hood to make it easier to develop an application.
- **Simulation tools:** People developing an embedded application, especially those attempting one for the first time, may not have access to all the hardware that the application will use when they begin development. Koneki helps simulating the various sensors and communication scenarios the application will use.
- **Remote development:** Koneki makes it much easier to write source code and deploy it on an embedded device for testing, a process that was historically very complex. With Koneki and Eclipse, developers can write their application and click a button, and under the hood, the application is transferred to the remote system and launched automatically.

Ultimately, the combination of these tools will create a “one-stop shop” for developing M2M applications. It also will allow developers to cultivate and build on their embedded application skills and knowledge in a way that wasn’t possible

before.

## Capitalizing On Eclipse

Early on in the process of trying to create a body of common M2M development tools, our team at Sierra Wireless decided that the Eclipse Foundation was the right home for an open-source M2M IDE. Eclipse is not only an open-source foundation, but historically it also has been a platform built to support common development tools.

From the beginning, Eclipse was meant to provide an extensible IDE that would support a large, constantly growing body of plug-ins, debuggers, source code management systems, and other tools across a variety of development environments to make it easier to develop applications. This is exactly what has been missing for embedded applications.

When developers rely on proprietary tools from a manufacturer or hardware vendor, they can't use many of the common tools and enhancements that they can apply to other types of development to work more productively. Eclipse makes that possible by supporting all manner of plug-ins and making M2M tools extensible for the first time. With an embedded application IDE and tools within Eclipse, developers can develop M2M applications in much the same way they develop any other application, using common tools, within a single development framework.

A developer can create an entire M2M solution—for example, an embedded application that communicates with a back-end application that's accessible through a mobile user interface (UI) in an Android app—within the same Eclipse environment. And, because Eclipse is an open environment, it will continue to evolve to support more tools and enhancements, making it faster and easier to create powerful embedded applications.

## Laying The Groundwork For More M2M Innovation

Koneki development tools provide an important first step in creating a common, open, and easy to use IDE for supporting developers of M2M applications. Today, any developer interested in using these tools can visit the Eclipse Web site at [www.eclipse.org/koneki/](http://www.eclipse.org/koneki/) and download and begin using these tools.

In fact, the Lua development tools within Koneki are now mature enough that we recently released an official 1.0 version of Lua Development Tools, available for download right now. Developers can also download an OMA-DM simulator for simulating the popular device management protocol from the Open Mobile Alliance.

As more developers embrace these open-source tools, as well as other elements of the M2M IWG initiative—the Mihini open-source framework and a growing body of open-source M2M protocols—we believe we can begin to unlock the full potential of M2M applications.

## Reference

“The Internet of Things,” Cisco, <http://share.cisco.com/internet-of-things.html>

*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.*

**Source URL:** <http://electronicdesign.com/communications/koneki-open-source-development-tools-simplify-m2m-development>