Short Range Wireless Usage, Concerns, Directions and Debugging in the IoT Era

By Trish Messiter and Sameera Rajaratne, Clarinox Technologies

The world of embedded is changing. It is being impacted by one of the major growth markets of our time – the Internet of Things (IoT). The IoT demands innovative solutions to a new set of requirements, impacting upon the decisions, issues and preferences of engineers tasked with wireless embedded designs.

The fundamental requirement of IoT designs is efficient interaction based upon connectivity between devices. Mostly the preferred connectivity mechanism is wireless. The choice amongst wireless technologies however is large as is presented in **Table 1**. It lists the technologies used in designs current in the 2014 EMF survey [1] and then compares the planned and actual uptake of technology for the following year. For all standards based technologies the actual number of device designs incorporating wireless technologies was greater than planned. The same period saw a significant reduction in designs that did not include any wireless; as well as a reduction in designs incorporating proprietary wireless indicating a strong trend to the inclusion of wireless technologies into more designs.

Technology	Reported as used, 2014 survey	Planned use, 2014 survey	Actual use as reported, 2015 survey
Bluetooth	36%	43%	51.5%
802.11a	8%	3%	11.8%
802.11b	17%	11%	18.4%
802.11g	23%	20%	23.4%
802.11ac	5%	5%	10.6%
802.11ad	4%	1%	5.4%
802.11n	16%	16%	18.7%
Zigbee	20	14%	22%
RFID	12%	10%	14.2%
NFC	5%	5%	8%
IrDa	7%	3%	10.6%
Own	17%	16%	14.4%
None	29%	27%	11.8%

Table 1. Average Reported Technology use across all countries

Source of Bluetooth Stack	Industry	IoT
Provided with chip (e.g. CSR, Broadcom, TI)	26.4%	22.1%
Provided by Bluetooth vendor module	35.2%	28.9%
Provided by RTOS vendor	22.4%	26.7%
Provided by Stack Software Vendor	16.0%	22.2%

Table 2. Industry / IoT Bluetooth stack sources

Frequency	Total Industry	Bluetooth Stack from h/w vendor	Bluetooth Stack from s/w vendor
Never	23.8%	35.0%	10.0%
Yearly	42.9%	45.0%	60.0%
Quarterly	21.9%	20.0%	20.0%
Monthly	4.8%	0.0%	10.0%
More often than monthly	6.7%	0.0%	0.0%

Table 3. ***Q.*** **Once deployed, how often is your current embedded project likely to be updated in the field?**

In numbers

Among wireless protocols used for embedded developments, it can be seen from Table 1 that Bluetooth is extremely popular in the 2015 survey with actual industry wide use reported as 51.5%. This technology was investigated in further detail by looking into several industry groups as illustrated in **Figure 1**. The area of medical device developments reported the very high level of 82.7% of designs incorporating Bluetooth technology. Interestingly approximately half of developers within Automotive and Medical classified their projects as IoT projects. And within consumer electronics (CE) this number hit nearly 75%. Within this group of developers that classified themselves to be working on IoT projects the use of Bluetooth technology across all industries was 73%, up from 51.5% for non-IoT projects showing the importance of Bluetooth technology for IoT designs.

When it comes to embedded Bluetooth designs, most developers have traditionally relied on the Bluetooth stack bundles provided by chip or module vendors (i.e. hardware vendors) but is this still the right decision for IoT designs? Can software vendor-offered Bluetooth stack sources prove to be superior alternatives in some circumstances? The Sources of Bluetooth stacks and usage within the embedded market are shown in **Table 2**.

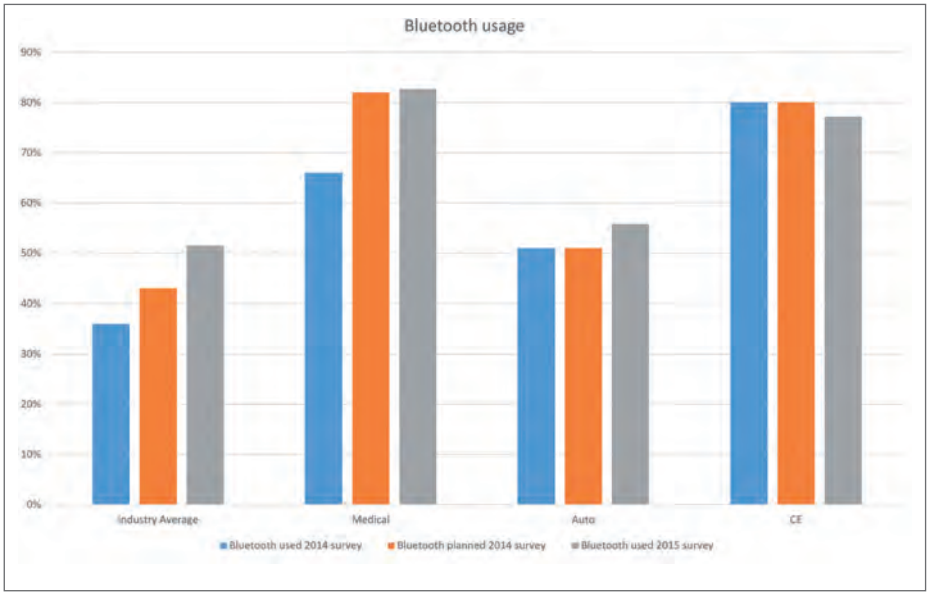

Figure 1.

Interestingly while the industry average shows acquisition of stack software from either chip or module vendor was 61.6%; for IoT developments this was the lower figure of 51.1%. This represents a shift from hardware towards the software vendors for IoT developments; but why?

The survey results were then analysed to look for the characteristics in common between projects that were part of this trend. Respondents were asked how frequently they upgraded the software in their deployed systems and **Table 3** summarises the responses according to the source of Bluetooth stacks employed in their designs.

For the developer who intends on performing updates to their embedded Bluetooth projects after deployment, Table 3 indicates that a software vendor is the more likely choice for the stack source. In fact, 2015 EMF survey results reveal that 90% of embedded developers using a software vendor stack solution in their current projects intend on dispatching updates to their Bluetooth systems once deployed, with 10% of developers intending to service their projects regularly at monthly intervals.

These figures would indicate that there is a perception that Bluetooth systems from hardware vendors are more difficult

to upgrade. Stack software bundled with Bluetooth SoCs or provided by Bluetooth module vendors often introduce unwanted dependencies in the form of hardware/platform-specific support capabilities and no abstraction layers to separate applications from hardware. Such stack solutions then become suited to the zero-maintenance, deploy-and-forget Bluetooth projects.

Meeting expectations

In the analyses, it is of considerable importance to review the performances of protocol software with regards to meeting the requirements asked of embedded engineers for their projects. Respondents were asked to select the three most disappointing aspects, or biggest issue of their protocol software selection. The results are presented in **Table 4**.

	Total Industry	Bluetooth Stack from h/w vendor	Bluetooth Stack from s/w vendor
Expectation of better support	46.0%	50.0%	45.2%
Ease of purchasing	9.8%	9.7%	19.0%
Better technical solution	37.2%	41.9%	38.1%
Product integration	40.7%	51.6%	40.5%
Corporate standardization	6.8%	8.1%	7.1%
Better price	24.9%	29.0%	21.4%
Open Source	18.6%	9.7%	19.0%
Other	4.3%	1.6%	4.8%

Table 4. ***Q.*** **What were the most disappointing aspects of your protocol software selection?**

There is a surprising rate of dissatisfaction among respondents using hardware vendor stack source alternatives with regards to the lack of adequate product integration and a slightly higher dissatisfaction with the technical solution for their products or systems as compared to the software vendors. The lack of features, complexity, debugging tools and support are the likely problems linked to stacks from these vendors. This is not surprising; hardware vendors develop their stack solutions to meet the demands of the generic or majority use case developer; hence these are solutions may be lacking many of the advanced and complex capabilities and not tuned to a specific application's requirements.

The purchasing process is more of an issue for software vendor solutions. In comparison to the relative easiness for embedded developers working with hardware vendor based solutions; those stacks tend to come already integrated in the chip or module for immediate development. There may be an initial time outlay to search for vendors, evaluate, engage and then install, customise and test the software vendor stack. These additional efforts are being taken on with increasing frequency with the shift to an increased software focus within IoT developments and could be reduced by closer relationships between the hardware and software vendors such as is provided by partner programs.

Developers were also asked, based on their current design efforts, approximately how many end units are expected to be shipped over the life of the product, the result of the questionnaire is shown in **Table 5**.

It may be surprising to see how frequently software vendor based stack solutions are employed for products with shipment expectations exceeding 500,000 units over the life of the product; with 15.2% of uses of software vendors stacks in 500,000+ units category compares to 3.9% of developments in this volume category using hardware vendor-sourced stacks. We conclude from this that software vendor based Bluetooth stack solutions present themselves to be the preferred choice for large production quantities over the life of the system. Ease of maintenance, scalability of upgrading, and a higher volume over which to amortise the licensing expenses are the likely reasons for the preference.

Debugging

For highly complex and speedy wireless developments, the availability and effectiveness of debugging features becomes a significant concern and a point of comparison. There are of course many aspects, tools and techniques available for debugging. As shown in **Table 6**, in general it was found that developers using stacks from hardware vendors relied more heavily on hardware aligned techniques such as blinking LEDs on the board and were more likely to use a combination of debug tools. That said however wireless embedded developers deploying hardware vendor stack sources will often have more limited tool-chain support and possibly ones with minimal software based debugging capabilities. Software vendors, on the other hand, especially independent stack vendors make it a priority to include rich debugging features that enables wireless developers to quickly progress through the development phase of the product cycle. With software vendor sourced stacks, developers will often find the basic debugging features available with additional tools such as protocol analysers and memory leak analysers to complement a more comprehensive debugging suite for wireless applications. Protocol and memory leak analysers provide an in-depth analysis of the wireless application and enables fast debugging even the most complex applications. These higher level wireless focused tools complement to the lower level debug capabilities of a debug focused toolchain together with tools such as spectrum and logic analyzers.

Units	Total Industry	Bluetooth Stack from h/w vendor	Bluetooth Stack from s/w vendor
under 50	15.7%	11.7%	15.2%
50 to 99	5.4%	3.9%	6.5%
100 to 199	4.7%	5.2%	6.5%
200 to 299	3.1%	0.0%	0.0%
300 to 399	1.7%	2.6%	4.3%
400 to 499	1.2%	0.0%	0.0%
500 to 999	7.2%	3.9%	10.9%
1,000 to 9,999	18.4%	24.7%	17.4%
10,000 to 49,999	12.2%	15.6%	6.5%
50,000 to 99,999	5.4%	6.5%	4.3%
100,000 to 499,999	6.0%	11.7%	10.9%
500,000 to 999,999	1.9%	1.3%	4.3%
1 million or more	7.1%	2.6%	10.9%
Don't know	9.9%	10.4%	2.2%

Table 5. Based upon your current design effort, approximately how many end units do you expect your company to ship over the life of the product?

Method	Total Industry	Bluetooth Stack from h/w vendor	Bluetooth Stack from s/w vendor
Using a debugger	80.6%	89.4%	70.5%
Blinking LEDs on the board	58.2%	78.9%	61.4%
Using printf( ) statements in the code	54.3%	69.7%	59.1%
Logic analyzer	45.9%	61.8%	65.9%
Simulation	38.1%	31.6%	29.5%
Protocol analyzer	35.6%	51.3%	38.6%
Trace module	24.3%	32.9%	25.0%
Host adapter	20.6%	30.3%	15.9%
Using a real-time visualization tool	14.3%	11.8%	18.2%
Spectrum analyzer	9.9%	18.4%	15.9%
Other	4.2%	6.6%	4.5%

Table 6. How do you debug your software?

Conclusions

From this analysis it appears that for low volume, less complex and more traditional (or projects the developers consider not to be IoT developments) then the stack obtained from a hardware vendor appears the more popular choice.

For the other end of the spectrum, very high volume, high complexity and new or IoT developments then there is a move towards selecting a software stack from a software vendor.

If a product is to remain in the field for many years, End of Life (EOL) issues must be dealt with. Choosing software that is provided with a chip then means that the software will also need to be changed when the hardware come EOL. This is double trouble. Much better to pick a Bluetooth stack that has an abstraction layer so the same upper layer software will run despite changes in hardware. Increase quality and performance by choosing an independent stack.

More IoT developments are using stacks from software vendors and in most cases IoT developments will be new developments. It can speculated that new developments may also be pushing towards increased complexity as market demands for increasingly sophisticated features puts increased pressure on new project requirements. The flexibility of software based solutions lend themselves more to catering for unique and complex requirements.

Other than the dependency on the hardware vendor there is the issue of meeting requirements. What if the required RTOS is not supported? What if the required functionality is not supported? What if the required quality is not provided? Hardware vendors must cater for the mainstream. But what if you want to create functionality that is beyond mainstream? The choice in these situations is to pick a stack vendor that is independent; specializes in Bluetooth technology and takes the time to provide for new innovative functionality.

For high complexity and speedy IoT developments effectiveness debugging is a significant concern. Software vendors are

The Authors

Sameera Rajaratne is currently undertaking B.Eng. degree in electrical and computer systems and B.Comm. degree in finance from Monash University, Clayton, Australia. His experience at Clarinox, along with past research in the RF microwave field, has driven his passion for the wireless embedded industry, particularly for the ongoing developments and innovations in IoT technologies. Sameera is also currently a research assistant at Monash University, working on biomedical IoT projects. He is a student member of IEEE.

Trish Messiter is co-founder and CEO for Clarinox with experience across many fields including research, engineering design, computer modelling, software debugging and technical support as well as technical sales and marketing. Trish has an engineering degree from the University of NSW, Australia and a certificate in sales and marketing. She has provided presentations on short range wireless technologies at a number of conferences including Embedded World. She has published work on embedded software and wireless technologies in collaboration with RMIT.

more likely to include rich debugging features available with additional tools such as software based protocol analysers and memory leak analysers to complement the toolchain debugging capabilities.

For the embedded industry and the IoT developer, the imperative question is whether their designs are best served by incorporating Bluetooth on hardware or purchased independently from a dedicated software vendor (either RTOS or stack vendor).

The findings provided in this report support the idea that although the chips and modules themselves may or may not be particularly expensive, the cost of replacing and upgrading Bluetooth on such systems may be more difficult (and possibly more expensive) than by upgrading software only. Given the large percentage of systems upgrades, developers and managers should consider the costs between deploying chip or module-based and software vendor based Bluetooth developments.

www.clarinox.com

Reference

[1] EMF, Embedded Market Forecasters (http://embeddedforecast.com), Market Intelligence and Research for the Embedded Systems Industry