embeddedworld2015
Exhibition&Conference
... it´s a smarter world

Short range wireless technologies BLE, Bluetooth and Wi-Fi are an essential part of any IoT effort

Gokhan Tanyeri; Trish Messiter
Clarinox Technologies Pty Ltd
Melbourne, Australia
firstname@clarinox.com

Paul Beckett; Glenn Matthews
Electrical and Computer Engineering
RMIT University
Melbourne, Australia
firstname.lastname@rmit.edu.au

Abstract*—The Internet of Things is changing the shape of the embedded systems industry and, in turn, the shape of the world. A key component of this change is the role that short range wireless – such as BLE, Wi-Fi and Bluetooth -- will play in new IoT scenarios.*

The focus of this paper will be on the many aspects of short range wireless connectivity for IoT applications. It will give developers (and their managers) greater understanding of the practicalities behind new scenarios involving short range wireless.

We will investigate questions such as: what short range wireless protocols are currently being used; what issues surround their use; what can be done to contain design costs; and how can developers deploy robust solutions in the face of increasing time to market pressures?

All of the popular technologies are rapidly evolving. The specifications are continuously upgraded and improved. Constant change poses design, development and maintenance issues, as does the increasing pressure of reduced time-to-market.

The pros and cons of off-the-shelf vs custom hardware/software solutions will be discussed in light of these challenges. The role of scalable, debuggable software infrastructure will also be discussed in the light of increasing time pressures. The advantages of a multi-faceted approach will be discussed.

The session will conclude with some examples of combined Bluetooth & Wi-Fi product developments. Many IoT designs require the integration of multiple wireless technologies within a single device (as if one technology wasn't challenging enough!), and this is a key frontier for developers.

Keywords—IoT, short range wireless, Bluetooth, BLE, Wi-Fi

I. INTRODUCTION

Since the emergence of the internet as a recognizable entity, it has been clear that it will continue to evolve towards greater reach and more complex connectivity. In this context, the Internet of Things (IoT) might be viewed as simply a logical extension of the current "internet of people". In [1], Gigaom Research defines the IoT as: *"An ultra-connected environment of capabilities and services, enabling interaction with and among physical objects and their virtual representations, based on supporting technologies such as sensors, controllers, or low-powered wireless as well as services available from the wider internet."*. On the other hand, a simpler definition is provided by Gartner [2] as a: *"…network of physical objects that contain embedded technology to communicate and sense or interact with their internal states or the external environment"*. The key theme in all definitions is efficient interaction and connectivity between devices.

Typically, these connected devices are wireless with many devices coming to market with multiple short range wireless technologies integrated into their design. The IoT ecosystem requires various elements. On the "Internet" (cloud) side, companies with skills of managing big data are required to process, categorize, and store incredible amounts of data. On the "Things" (device) side, embedded sensors, MCU, RTOS, middleware and connectivity providers are needed. In between these two sides exists the *"complicated and necessary processes … [that] make possible the distributed intelligence, autonomous operation, and real-time performance of the IoT"* [3]. This middle layer encompasses networking technologies including, protocols, gateways and communications.

These components are not necessarily new; rather what has changed is the availability of the necessary elements at a price point that is attractive for a myriad of applications. In other words: *"The relentless advance of Moore's law is bringing IoT*

into the mainstream, broadening the range of opportunities to monitor, connect, and control physical objects through the use of embedded technologies, either within existing products or as separate devices" [1].

The IoT ecosystem is thereby rapidly becoming both a subset of the traditional embedded systems domain and also a superset when the elements of cloud computing, big data and networking are taken into account. While it is true that the internet of things (IoT) can be a difficult concept to pin down simply, one thing is certain; it will be the most complex system developed in human history and will fundamentally change the way we interact. Whether machine to machine, machine to infrastructure, machine to environment, or machine to human, there is a huge explosion in smart connected things that is impacting upon every aspect of our interactions. A recent report by the UK Government Chief Scientific Adviser [4], estimates that:*"... the number of connected devices could be anywhere from 20 billion to 100 billion by 2020."*. These estimates are summarized in FIGURE 1.

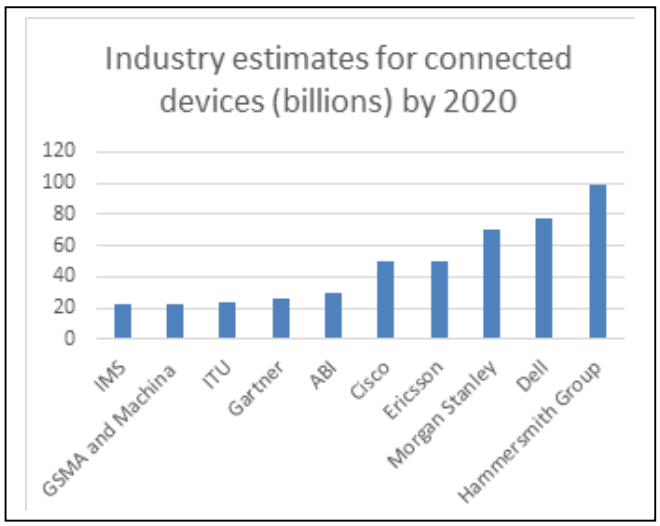

Figure 1 Estimates for numbers of connected devices [4]

II. WHAT SHORT RANGE WIRELESS PROTOCOLS ARE CURRENTLY BEING USED?

The Embedded Market Forecasters 2014 survey [5] obtained data from approximately 700 embedded developers and managers. The use of wireless technologies in designs over the previous 12 months across a range of industries obtained from this survey is shown in TABLE 1. It can be seen that the adoption of wireless standards varies widely between specific industry sectors. For example, whilst all listed industries have above industry average use of both Bluetooth and Wi-Fi technologies, only Consumer Electronics (CE) and Home Automation (Home) have above average use of ZigBee.

All of the industries in Table 1, except for automotive, have a reduced number of designs compared to the industry average that do not include wireless at all (shown as row label *None*). The use of wireless technologies in projects planned for the next 12 months in this same survey is shown in TABLE 2.

It can be seen that the trend for medical, automotive, consumer electronics and home automation to all have a higher uptake of both Bluetooth and Wi-Fi technologies than average can be expected to continue. In future designs across all industries the use of ZigBee, RFID and IrDA is planned to reduce, with ZigBee seeing the largest reduction in planned integration into new designs in the Consumer Electronics sector.

TABLE 1. COMPARATIVE CURRENT WIRELESS USE

TECH	Used in current designs				
	Industry	***Medical***	***Auto***	***CE***	***Home***
BTV2.0	16%	29%	20%	29%	17%
BTV3.0	7%	11%	13%	20%	17%
BTV4.0	13%	26%	18%	31%	17%
All Bluetooth	36%	66%	51%	80%	51%
802.11a	8%	13%	17%	18%	17%
802.11b	17%	22%	18%	27%	24%
802.11g	23%	29%	22%	31%	35%
802.11ac	5%	4%	5%	14%	10%
802.11ad	4%	5%	5%	6%	3%
802.11n	16%	28%	16%	18%	21%
ZigBee	20%	17%	20%	29%	24%
RFID	12%	15%	11%	12%	14%
NFC	5%	6%	9%	12%	10%
IrDA	7%	7%	7%	6%	10%
Own	17%	16%	16%	14%	17%
None	29%	12%	28%	12%	21%

TABLE 2. COMPARATIVE PLANNED WIRELESS USE

TECH	Planned for future designs				
	Industry	***Medical***	***Auto***	***CE***	***Home***
BTV2.0	13%	24%	10%	23%	15%
BTV3.0	9%	9%	16%	18%	22%
BTV4.0	21%	49%	25%	39%	22%
All Bluetooth	43%	82%	51%	80%	59%
802.11a	3%	6%	9%	5%	7%
802.11b	11%	12%	13%	13%	15%
802.11g	20%	22%	20%	18%	19%
802.11ac	5%	6%	6%	8%	7%
802.11ad	1%	2%	3%	3%	0%
802.11n	16%	21%	17%	15%	11%
ZigBee	14%	15%	16%	8%	11%
RFID	10%	6%	17%	10%	7%
NFC	5%	8%	9%	13%	0%
IrDA	3%	8%	4%	0%	0%
Own	16%	18%	13%	13%	11%
None	27%	12%	28%	13%	15%

III. IoT CHALLENGES

The IoT is ultimately about using technology to provide connected solutions that are sufficiently attractive to encourage purchase by the end users. *"To be successful, IoT use cases need to fit with the way technology is being adopted by today's agile businesses [and technology savvy consumers]"* [6]. Business value however will not be realized unless technology challenges are addressed. Whilst the impression is commonly provided that solutions are simple to develop; and processors, sensors and wireless technologies are cheap, this is underrepresenting the expertise and effort necessary to design and develop smart connected "things". The fact that more wireless embedded projects are behind schedule or cancelled than non-wireless projects is a testimony to the challenges in these designs [5].

The overall IoT ecosystem, as well as the individual technology components required, are still forming and adapting to the new paradigm. The constant change and continuous upgrade of standards is just one aspect that presents challenges and it is time consuming and expensive to keep pace.

The multiple short range wireless technology standards, whilst providing flexibility of choice can also cause confusion. From Tables I and II it appears that Bluetooth and Wi-Fi are now firmly established complementary technologies. Wi-Fi lends itself to higher bandwidth usage scenarios such as video steaming or large file transfer, Bluetooth for voice, music and (with the introduction of BLE) adaptable for wearable devices, sensors and home automation.

The issue of interoperability continues to be a complex problem as the solution is a balance between conflicting constraints with the desire for simple, light weight code on one side and backward compatibility and multiple use cases on the other. Code complexity is also challenging. One wireless technology is complex enough; implementing multiple technologies more so.

At the device level, the target platform support also becomes a challenge. RTOS and wireless protocol stack software must support the MCU as well as the wireless chipset and the transport mechanisms.

The fast pace of adoption of the IoT is also pushing the feature sets of the current standards to their limits, despite the continuous additions to these standards. In the past relatively simple use cases were often enough to differentiate products but now this may only be sufficient to meet basic expectations. Users are demanding increased sophistication and this can lead to a requirement to run simultaneous technologies and within Bluetooth simultaneous profiles and roles, greatly increasing complexity of design.

Finally, the perennial challenge in embedded design, i.e., difficulty of debug, will continue to be a major challenge and the trends to increasing complexity and shorter design cycles will further exacerbate this issue.

In summary, the main challenges are:

- Complexity;
- Changing standards;
- Interoperability;
- Co-existence;
- Target platform support;
- Range of features;
- Simultaneous roles and/or profiles;
- Debugging;

Embedded development for the IoT will face all of these issues simultaneously. The challenge is to ensure they are all managed to achieve successful projects and designs.

IV. WHAT CAN BE DONE TO CONTAIN DESIGN COSTS?

There is tradeoff between NRE costs and BOM costs so the initial decision is to prioritize these. From a top level approach, if NRE costs are the priority then pre-built and/or open source hardware and software modules may be suitable. If BOM costs are the priority, then single chip based and specialized vendors may offer the best solutions.

At a more detailed level, however, additional processes and tools will help to reduce costs and get robust designs to market faster.

There are a number of techniques used to address the management of design costs, these include:

- Off-the-shelf solutions
- Scalable Middleware infrastructure
- Single chip designs
- Code and design reuse
- Debug tools

A. Off-the-shelf vs Custom Hardware/Software Solutions

Future wireless system developments for the IoT market will have to face multiple wireless standards, along with incessant demands for increased energy efficiency, higher data rates, increased security and more. In common with the wider trends in embedded design, it appears that hardware is becoming more commoditized and there is a move towards the situation where the focus is largely on software [5, 7].

On the hardware side, current trends (e.g., [8]) indicate a slow but steady move from 8-bit towards 32-bit microcontroller hardware (and, interestingly, a move away from reconfigurable systems such as FPGAs). As a result, an increasing percentage of projects are likely to use formal operating systems and middleware to manage their design complexity and time-to-market pressures. However, it is clear that the demands of energy efficiency will prevent the uptake of "virtualization" in this domain for some time yet.

Middleware is a less processor-intensive alternative to virtualization that is still able to present a single environment to all applications. The idea is to encapsulate the API interfaces and services and provide a consistent view of the operating system to the software layer. This approach is already in common use for embedded communications[9]. For many years, informal embedded market estimates [10] have indicated that purpose-built middleware is very common in real-world applications, notwithstanding problems such as lack of scalability, cost of managing and supporting deployed systems

and their performance overheads. It also appears that most companies using middleware currently develop their own.

However, the additional demands placed on devices and systems for IoT requires a more unified approach to embedded tools, middleware and life-cycle management. In this way, an embedded development system needs to represent a complete solution, rather than as a loose group of isolated elements.

B. The Role of Scalable, Middleware Infrastructure

Formal middleware frameworks such as the ClarinoxSoftFrame® [9], offer a modular and open approach to embedded software development. Their intention is to encapsulate and hide any underlying hardware and software differences and provide a common interface to the higher-layer applications, focusing on just the application specific interfaces and using a standard API.

The benefits of such an approach include an ability to use a variety of suppliers depending on the requirements whilst allowing for a much higher degree of reuse of the intellectual property developed during a project. For example if for one project a low cost MCU with less resources is suitable, then the middleware port to support that MCU can be established, and the application layer developed. If the next project has similar application level functionality required, but needs a higher performance and more resourced MCU, then only the middleware requires update; and the majority of application layer is directly reusable. In this manner both the requirements of the individual projects, and the need to obtain increasing levels of productivity, can be achieved.

While at times all MCU manufacturers come up with brilliant new MCU families, to be able to benefit from a brand new architecture requires a software infrastructure catering for that family. At the same time, flexibility is required to avoid being forever locked into a single supplier and to protect the product's intellectual property. In a similar way, a different RTOS and different wireless chips may be deployed across a range of projects. These choices will be dependent on the pricing, performance or functionality of the specific item. For example, a particular RTOS might be certified for high reliability or for medical use. Ultimately, the deciding factor might simply be the familiarity of the project team with a particular set of products. Middleware enables such choices.

Another advantage of middleware is the flexibility of the upgrade and maintenance as the whole software is contained. The upgrades could be managed from a single point instead of multiple firmware upgrades. In addition, the integration and maintenance of firmware from a single vendor is generally preferable than dealing with multiple products across several vendors.

C. Development of single chip designs

The idea of increased hardware integration (thereby reducing component count) will generally be in conflict with the concept of off-the-shelf components. Prebuilt hardware modules will typically increase the total number of components deployed within a design (as individual modules contain multiple components). For many IoT designs it is possible to meet the design requirements with a single MCU and a wireless chipset (rather than OTS modules). By combining this architecture with suitably scaled software, a reduction in key components can be achieved.

A software solution based on a single MCU provides many advantages when considered over the full product lifecycle. By eliminating the number of MCUs used for the design and reducing the chipsets used for the wireless technologies, the overall BOM pricing will be reduced significantly. This provides huge competitive advantages to the OEM, especially for high volume products.

D. Code and design reuse

In principle, both code and design reuse aim to save the cost of time to implement and test new code by the practice of using code previously created. While code is a commonly reused element, other aspects of a design, such as testing tools, hardware designs and documentation may also be suitable for reuse. Whilst the theory is sound, practical issues may impede the amount of reuse possible in an embedded design due to the linkage between hardware and software. This will be especially difficult when the resource requirements of the hardware platform differ greatly between projects.

E. Debug tools

Debugging has always been amongst the biggest concerns of designers and embedded systems are becoming increasingly complex with unique constraints which make them increasingly hard to debug. As test and debug continues to consume the largest slice of the development and maintenance cycle [11], new approaches to debugging are required.

In contrast to the general purpose computer software design environment, a primary characteristic of embedded environments is the sheer number of different platforms available to the developers (CPU architectures, silicon vendors, operating systems and their variants). Embedded systems are, by definition, not general-purpose designs: they are typically developed for a single task (or small range of tasks), and the platform is chosen specifically to optimize that application. Not only does this fact make life tough for embedded system developers, it also makes debugging and testing of these systems harder as well, since different debugging tools are needed in different platforms [12].

Tools such as ClarinoxSoftFrame mentioned previously use a *"plug-in"* approach [11] that offers a common interface to the hardware and shields the development team from the complexities of the various CPU architectures, operating systems, debugging processes etc. There are many other component-based architectures [12], [13] that target heterogeneous multi-processor systems. Their common feature is that they attempt to abstract away the complex inner details of their hardware and software stacks and are therefore able to target different hardware and software configurations by communicating through well-defined interfaces that can be swapped or modified to suit. In this way, the design group can focus on the high-level behavior of the system rather than being bogged down in hardware details.

In our experience, visibility within the debug process can be greatly enhanced by the flexible integration of high and low level tools within a unified framework. This approach can offer the simplicity of a combined tool for faster debugging plus a

formal framework targeting a variety of operating systems and platforms, thereby allowing for smaller development teams.

V. EXAMPLES OF COMBINED BLUETOOTH & WI-FI PRODUCT DEVELOPMENTS.

Over the next 12 months we will see more smart technologies becoming common place – including wearable tech, smart packaging, highly connected appliances, and deep integration between many different types of sensors including city management systems that will offer up significant insight into behavior of complex, global events [14].

We can form our approach to these new challenges from experience in past designs. The following examples are drawn from a range of Bluetooth & Wi-Fi product developments with which Clarinox have project experience. The examples help to illustrate the breadth of deployments and product types that are incorporating these technologies.

A. An intelligent whiteboard

The concept for a futuristic approach to the whiteboard combined NFC, Bluetooth Low Energy, Bluetooth Classic and, in the future, will support Wi-Fi as well.

The design used the state of the art DSP processor to manage the complex requirements of a large scale camera/scanner. In addition to these highly innovative electronics for the actual task of the white board, the product had to offer seamless wireless connectivity.

The seamless connectivity was achieved by using NFC for an elegant and simple method to pair devices. The combination of both Bluetooth classic and Bluetooth low energy enabled the use of the simplest wireless protocols.

The future of such a product may include Wi-Fi as an additional protocol. These protocol choices make sure that any smart device can be connected to the white board for sharing the whiteboard contents.

Lessons for future designs: The use of the simplest communications methods to achieve the goals can result in the most elegant design.

B. Satelite positioning plus short range wireless

An innovation in GNSS receiver modules required support of scalable positioning options and low latency positioning with high data rates. The product brief was to not only use the satellite technology but Bluetooth and Wi-Fi as well.

A powerful multi-processor architecture was required to manage the variety of communication options as well as the required DSP processing for novel positioning algorithms. Such an architecture guaranteed the stability and accuracy of positioning whilst supporting a challenging communication architecture that included Wi-Fi and Bluetooth protocols.

Lessons for future designs: High end hardware to support complex high end software.

C. Clarinox Koala EVM

The goal was to address the gap between the low end MCU based solutions and the high end multiprocessor solutions. In the middle there is a place for a single MCU, single wireless chipset solution that was sufficient to support simultaneous Bluetooth Classic, Low Energy and Wi-Fi.

The STMicroelectronics Cortex-M4 ST32F429 MCU combined with the Texas Instruments WiLink-8 wireless chipset enables multiband Wi-Fi, Bluetooth Classic and Bluetooth Low Energy to run simultaneously. Wi-Fi data rates of up to 16Mbps can be achieved with 70% CPU loading.

Lessons for future designs: Single MCU, single wireless chip solutions can support many IoT scenarios

D. Advanced angler electronics

Portable fish finding instruments for anglers have entered into a new age with the incorporation of multiple short range technologies including Bluetooth.

A powerful ARM Cortex-A9 platform supports the traditional functionality and the new usages scenarios provided for by Bluetooth Classic and Low Energy technologies within a Linux environment.

Lessons for future designs: Even when working within an open source environment incorporation of some proprietary components can add functionality to the overall design.

VI. CONCLUSIONS

The IoT is “moving fast – far faster than anything we’ve seen before” with predictions that 2015 will be “the year of the IoT” [14]. There is no doubt that it is a very major challenge for engineers to organize the communication of the trillions of devices that will form the IoT. This year will bring “more functionality, more complex software and more design headaches” [15] with the IoT being a large contributor to these trends.

However it is the very essence of engineering to develop the required solutions. Many of the solutions currently, or with some measure of adaptation will, exist. These include tried and tested concepts such as use of OTS solutions, scalable middleware, single chip solutions, code/design reuse and improved debug tools.

It is definitely exciting to have a key role in this extraordinary development in the history of mankind. A period of history foreseen by Tesla in his prediction, “When wireless is perfectly applied the whole earth will be converted into a huge brain, which in fact it is, all things being particles of a real and rhythmic whole.” [16].

REFERENCES

[1] Gigaom Research. (2015, Accessed: January 2015). *The internet of things: a market landscape*. Available: http://research.gigaom.com/report/the-internet-of-things-a-market-landscape/

[2] Gartner. (January 2015). *The Internet of Things - IoT* Available: www.gartner.com/it-glossary/internet-of-things

[3] Embedded Computing Design. *The hidden world of the IoT: Between sensors and the cloud*. Available: http://embedded-computing.com/guest-blogs/the-

hidden-world-of-the-iiot-between-sensors-and-the-cloud

[4] UK Government Chief Scientific Adviser. (2014). *The Internet of Things: making the most of the Second Digital Revolution* [online report]. Available: https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/389315/14-1230-internet-of-things-review.pdf

[5] Embedded Market Forecasters. (January). *2014 Market Intelligence Program* [online]. Available: http://embeddedforecast.com/

[6] Wind River. (2014, Accessed: January 2015). *GIGAOM Research Report: Enabling IoT*. Available: http://www.windriver.com/whitepapers/gigaom-research-enabling-iot/White-Paper-Gigaom-Research-Enabling-iot.pdf

[7] S. Cadene. (2013, accessed: January, 2015). *2013 Embedded Market Study* [online]. Available: http://www.slideshare.net/StephanCadene/embedded-mar1913

[8] R. Merritt. (2014, accessed: January, 2015). *Slideshow: 10 Embedded Design Trends*. Available: http://www.eetimes.com/document.asp?doc_id=1322014&print=yes

[9] Clarinox Technologies Pty. Ltd. (2010, accessed: January 2015). *The Clarinox Softframe.* [online]. Available: www.clarinox.com

[10] J. Krasner. (2009, accessed: January 2015). *Forecast 2010: What Is in Store for Embedded Developers* [online]. Available: http://www.embeddedmarketintelligence.com/2009/12/02/forecast-2010-what-is-in-store-for-embedded-developers/

[11] S. Farokhzad, G. Tanyeri, P. Messiter, and P. Beckett. (2010). *Plug-in Based Debugging For Embedded Systems* [online]. Available: http://www.clarinox.com/docs/whitepapers/EmbeddedDebugger.pdf

[12] L. G. Murillo, J. Harnath, R. Leupers, and G. Ascheid, "Scalable and retargetable debugger architecture for heterogeneous MPSoCs," in *System, Software, SoC and Silicon Debug Conference (S4D)*, 2012.

[13] Z. Kebin, G. Yu, and C. K. Angelov, "Graphical Model Debugger Framework for embedded systems," in *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2010.

[14] G. Webb. (2014, accessed: January 2015). *Why 2015 will be the year of the Internet of Things* [online]. Available: http://www.businessspectator.com.au/article/2014/12/30/technology/why-2015-will-be-year-internet-things

[15] J. Ganssle. (2015, accessed: January 2015). *A look foreback* [online]. Available: http://www.embedded.com/electronics-blogs/break-points/4438216/A-look-foreback

[16] M. Novak. (2015, accessed: January 2015). *Nikola Tesla's Incredible Predictions For Our Connected World* Available: http://www.gizmodo.com.au/2015/01/nikola-teslas-incredible-predictions-for-our-connected-world/