Published in Future Intelligent Educational Environments 2013, Athens, Greece 16-17th July 2013

Putting the Buzz Back into Computer Science Education

Vic Callaghan[a,1], Malcolm Lear[a], Anasol Peña-Rios[a] Victor Zamudio[b], Ping Zheng[c]

[a] University of Essex, UK
[b] Instituto Tecnológico de León, México
[c] Canterbury Christ Church University, UK

Abstract This paper describes a rapid-prototyping system, based around a set of modularised electronics, Buzz-Boards, which enable developers to quickly create a wide variety of products ranging from intelligent environments, through robots to smart-phones peripherals to be built and deployed. In this paper we introduce readers to Buzz-Board technology, illustrating its use through three examples, a desktop robot (BuzzBot), a desktop intelligent environment (BuzzBox) and an Internet-of-Things application using a Raspberry Pi adaptor (BuzzBerry). As part of this paper we provide a general overview of Computer Science curriculum developments and explain how Buzz-Boards technology can provide a highly motivating and effective focus for computer science practical assignments. This paper adds to earlier BuzzBoard publications by describing support for Raspberry Pis and intelligent environments, together with reviewing the latest developments in computer science curricula in the USA and UK.

Keywords. Educational Technology, Computer Science Curriculum, Internet-of-Things, Embedded Computing, Buzz-Boards, Intelligent Environments, Robotics, Smart-Phones. ACM-IEEE Curricula, DfE National Curriculum for Computing.

1. Introduction

Enthusing students is the key to engaging students in education. Part of this is providing students with coursework that is relevant to their lives and engages with their imagination and, if possible, creates a general '*buzz*' in the class. In this paper we discuss this challenge in relation to teaching computer science and present an approach, BuzzBoards, which we believe can be used as a vehicle for simultaneously providing a motivating theme while acting as a teaching tool to illustrate important computing principles. BuzzBoards are a rapid prototyping kit of hardware and software components that enable students and developers to quickly create Internet-of-Things, Pervasive Computing and Intelligent Environments products. In the following section we will introduce the problem we are solving and, by way of comparative examples, describe some other solutions available to educators.

[1] Corresponding author: vic@fortito.com

1.1. Rapid Prototyping kits for education

Computer Science teachers and students face a particular problem in relation to their practical assignments; a shortage of time! Typically, student practical work is organised so students have only a few hours to design, assemble and test a system. Generally that means computer science students have to use pre-designed hardware which limits their scope for creative design. Systems that enable developers (students or professional designers) to quickly create working prototypes are called *rapid prototyping* systems. Buzz-Boards are one such rapid-prototyping system and adopt the principle of utilising modularised plug-together boards in order to enable a wide variety of products to be created quickly, leaving the students more time to focus on creative design elements and programming the systems. For educators wishing to procure tools to support such educational processes, there are a number of systems available, which are described in more detail elsewhere [1] [2] but, for convenience, we now present a précis of the most relevant examples. One of the most widely used prototyping kits is the Arduino system (www.arduino.cc). This is an open-source physical computing platform based on a choice of two processors, ARM and Atmel's ATmega328 microcontrollers. There is a huge user base and a good choice of add-on boards. Programming is an implementation of Wiring which is a Java based platform and IDE (but can be expanded through C++ libraries). The mbed is a rapid prototyping platform developed through collaboration between ARM and Philips which is popular with commercial developers, which supports ARM based product design (http://mbed.org/); ARM being the most widely deployed embedded processor in the world being, for example, the processor of choice in smart-phones. It has a practical dual-inline form, allowing it to be plugged into electronic boards in much the same way as an integrated circuit (making it easy to integrate into prototypes). It can be developed in various ways but one attractive option is an online C/C++ compiler and IDE which provides highly productive collaboration support. Again there is a large user base that share software. Finally, there is the Raspberry Pi which is the newest, cheapest and most popular ARM based educational computer kit available, having shipped over a million units is the first 12 months (www.raspberrypi.org). It differs from the majority of bare-board platforms in that its functionality is closer to a data processing computer (ie desktop computer functionality) rather than an embedded computer. As a consequence the RPi IO is somewhat limited. It was originally intended that Python would be its main programming language but the massive RPi user base have ensured it can run numerous OSs and languages.

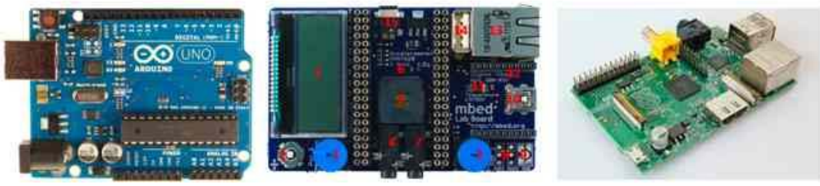

Fig. 1. (left to right): Arduino, mbed and Raspberry PI.

The Arduino, mbed RPi devices are shown in figure 1. A cursory glance reveals they are very different to regular personal computers and, as one might expect from such a bare technological appearance, the architectural functionalities are more obvious and the workings more basic, all of which are used as an educational advantage.

However, these devices and similar systems also have educational shortcomings. For example they frequently need additional hardware to create useful applications, requiring either third part add-ons or a high level of electronic competency to construct the required hardware. While for electronics engineering, there may be advantages in undertaking electronic design, for most computing students, needing to focus on programming, it's a drawback. BuzzBoards overcome these limitations by providing the add-ons these systems need, and offering a flexible plug-and play functionality that makes them simple and quick to use for students. The rest of this paper will examine some of these issues in more detail.

2. Overview of Buzz-Board Technology

2.1. Buzz-Board Markets

As explained earlier, Buzz-boards are primarily a technology for *rapid prototyping* that have applications in a wide range of areas where there is a need to assemble working computer based systems quickly. The primary areas targeted by Buzz-Boards are Education (student assignments), Industry (prototyping) and Maker Activities (arts, crafts & hobbyist). Buzz-boards can be used to create a divers set of applications such as smart-homes, smart phone apps, medical systems, pet-care, toys, internet-of-things gadgets etc. There are over 30 Buzz-Board modules for developers to choose from. Unfortunately, there is not the space to list them in this paper, but they are all listed on the FortiTo website (www.FortiTo.com) and papers [1] [2]. In the following sections we introduce the technology together with some exemplar applications.

2.2. Buzz-Boards

The key principle underpinning Buzz-Boards is modularity (both hardware and software) together with plug-and-play functionality (boards are identified to the system, and to each other, as they are plugged in) based on a common bus (Buzz-Bus). This provides a highly flexible, reconfigurable modular system that can be seen as an ideal infrastructure solution for rapid-prototyping and construction of pervasive and intelligent environments from full scale down to the desktop size environments.

2.2.1. Buzz-Bus

The flexibility of the Buzz-Board system is largely due to the Buzz-Bus board interconnect. Most sensor/affectors and other peripheral devices utilize I^2C, SPI, or RS232 serial buses. The Buzz-Bus uses these standards along with general purpose IO to allow Buzz-Boards to be reconfigured and interconnected to create novel products. For example the Buzz-Medi (ECG, EEG or EMG) and Buzz-Sense (humidity, temperature, barometric pressure) boards could be connected to the Buzz-Free (remote IO) to create a product that could log cardiac or muscular activity along with environmental data wirelessly onto a smart phone for later analysis.

2.2.1. Main Processor

The key component of any computing system is the processor. There are, however, a plethora of processors on the market suitable for embedded systems, each with a particular target market in mind. The Buzz-Board system could have adopted a specific processor but this solution would have been at odds with FortiTo's philosophy of flexibility and rapid prototyping without limiting component selection. In addition, FortiTo's marketing strategy is not to compete with existing and popular commercial offerings, but rather to support and augment them. With this in mind a processor agnostic solution was adopted. The main processor board adopted two methods of interfacing a wide range of processors. The first was a 40 pin dil socket designed to accept modules that have little more than a processor on-board, thus keeping cost to a minimum. This socket was based on the already existing mbed processor module thus immediately allowing full mbed compatibility [1]; see the right-hand image in figure 2. This processor agnostic socket links to the main boards Buzz-Bus sockets and on-board OLED display, push buttons, LED's etc. The possibility of remote wireless processing using a smart phone for example can also realized by plugging the Buzz-Free module into this socket. This option will be discussed later in this paper. The second processor interface is a dedicated Raspberry Pi connector; see the left-hand image in figure 2. The Raspberry Pi is a very popular low cost ARM processor board widely used in education. Unfortunately it suffers from limited peripheral IO interfaces, however it does support the I^2C and SPI serial buses and with the use of some Buzz interfaces, can be made fully Buzz-Bus compatible.

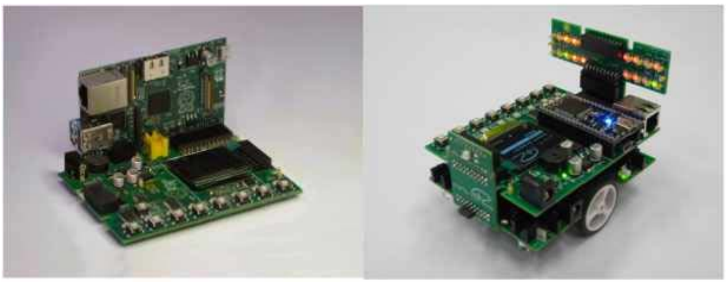

Figure 2: (left to right): Rasberry-Pi BuzzBerry and Buzz-Bot

3. Buzz Applications

3.1. Buzz-Bot – building a desktop robot

Buzz-boards enable the user to assemble them in different combinations; allowing the student to create and develop a wide variety of projects in a very simple 'plug-and-play' way. Mobile robots have long been recognized as a highly motivating and thorough way to cover computer science curricula [3] so, in this section, we describe how students can build a desktop robot Buzz-Bot (see Figure 3). The Buzz-Bot can be programmed to perform different tasks such as line following, light seeking, and maze escapes (which are the classic robotic challenges) among others. The Buzz-Bot includes 8 IR Range Finders, 5 line following sensors, 2 light following sensors, Lithium-Ion Battery, 2 dual mode motors, motor load monitoring, quadrature motor feedback, and USB and external DC charging. This example uses a mbed which can be programmed using C/C++. Programming the Buzz-Bot is very simple, using the online tools and software available on the mbed site. The program is compiled online, generating a .BIN file. The Buzzbed is connected to a PC via a USB (which behaves like a USB pen drive) allowing the user to ‘drag and drop’ the compiled program onto the “pen drive”. Figure 3 shows students from the Instituto Tecnologico de Leon (Mexico) programming and testing the Buzz-Bot as a line follower.

Figure 3. Students programming and testing the Buzz-Bot.

3.2. Buzz-Box – emulating intelligent environments

In simple terms, Intelligent Environments are high-tech environments, filled with numerous networked computers embedded into everyday things we use. Examples include smart -cities, -buildings, -hospitals, factories, -aircraft, -cars, -clothing or even space habitats. They are the forerunners of a new era of digital living where computers will be embedded into most aspects of our lives raising almost limitless possibilities. As such this is “hot topic” in research and teaching. Typically such facilities utilize a full size living space, such as an apartment, equipped with a rich selection of sensors/effectors along with associated embedded processing units. Whilst this approach has many advantages in terms of evaluating real life functionality and practicalities, it does not however lend itself to the classroom scenario involving several students, each requiring exclusive use of the environment. The Buzz-Box (figure 4) addresses this problem by the deconstruction and scaling down of the component parts of the larger environment to create a desktop environment. Key components are the interconnecting 250mm square Buzz-Panels fabricated using PCB material. This construction allows for a selection of essential environmental sensor/effectors to be pre-fitted to a range of panels. The various Buzz-Panels can be connected in any configuration (and size) to create the Buzz-Box using connectors that both mechanically hold the box form and distribute power and data to the sensor/effectors. The panels are manufactured as generic entities, and are then customised to create the specific functionalities illustrated in figure 4. For example, some panels act as the processor host, another might host media services, others a lighting or heating system, another as a controller or status indicator etc.

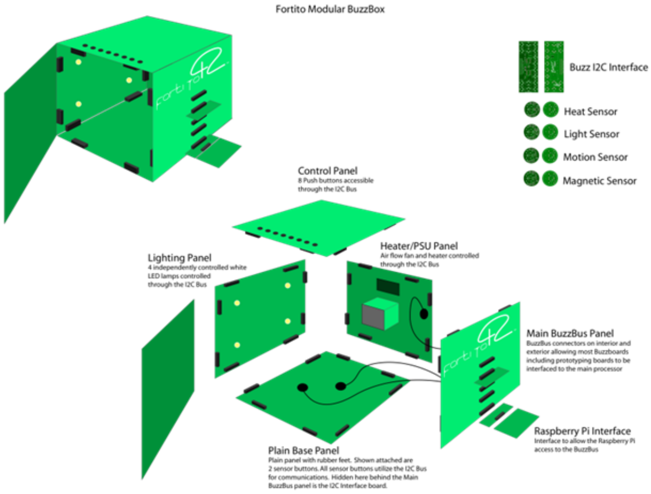

Figure 4 - The Buzz-Box Concept

3.3. Buzz-Free Wireless IO Link

The Buzz-Free wireless IO link (see Fig. 5) offers a new way to connect the processing power of a smart phone, or any computer, directly and wirelessly to hardware. The vision behind this product was to make it easy for smart-phone App developers to create applications to interact with the physical world. The difficulty is, while most embedded processors used inside smart phones have a rich set of peripheral IO interfaces such as I^2C, SPI, serial and general purpose IO, these interfaces are usually used exclusively within the phone and are not available for direct connection to the outside world. The FortiTo Buzz-Free system solves this restriction by breaking-out some of these interface to a small module called Buzz-Free. Access and control of the IO from smart phone processor is through a Bluetooth wireless link. The Buzz communication protocol is fast and simple to utilize once a Bluetooth channel is opened up.

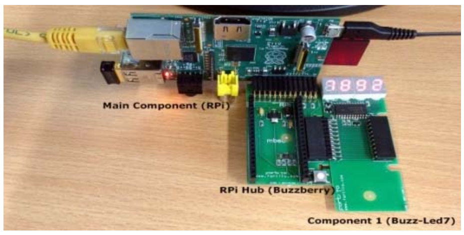

Figure 5 - Buzz-Free (Wireless IO)   Figure 6 – RPi + BuzzBoards (IoT Application)

For example, if an I^2C temperature sensor was connected to the Buzz-Free module, a smart phone could issue a simple ASCII command requesting data from the sensors I^2C address. Whilst the Buzz-Free module does have a processor, the protocol and firmware are fixed and the Buzz-Free module can be thought of as an IO breakout for the embedded processor, be it in a smart phone, Raspberry Pi or some other computer.

3.4. BuzzBerry – A RPi Interface for Internet-of-Things applications

The BuzzBerry (RPi hub) [1] is an interface board to the Raspberry Pi (RPi). As was mentioned in the introductions, the RPi is a small low-cost arm GNU/Linux computer which was developed to teach programming and computing fundamentals (http://www.raspberrypi.org/). The RPi uses an ARM processor and an SD card as a hard drive and the container of a customised Linux distribution for the device. In terms of peripherals, it allows the use of HD screens with a HDMI video output port and connectivity with an Ethernet port and two USB ports. It also includes RCA video output (for use with analogue televisions) and a 3.5 audio output jack. Finally the RPi includes General Purpose Input Output (GPIO) pins, which allow interfacing it to other devices in the real world. However, the IO is rather difficult to use as it requires a good understanding of electronics to interface to the physical world, which is why we have created the Buzz-Berry, as a way to simplify building RPi application that interact with the physical world.

A topical and highly motivating application for students is the *Internet-of-Things*, which in its early years had a variety of names, including the *Embedded-Internet* [4]. Sundmaeker [5] defined the Internet-of-Things (IoT) as "*a dynamic global network infrastructure where physical and virtual "things" have identities, physical attributes, virtual personalities, intelligent interfaces and are seamlessly integrated into the information network*". This creates smart objects capable of generating and collecting data autonomously using diverse sensors and actuators. In this example we use a mashup between RPi and BuzzBoard toolkit, a combination that allows the implementation of innovative projects by assembling diverse modules/functionalities in various combinations. To facilitate this, BuzzBerry uses RPi's GPIO pins to interface it with different Buzz-Boards modules allowing the creation of Internet-of-Things (IoT) applications of the students' choice (fig. 6). Discovery and communication using the Inter-Integrated Circuit bus (I^2C) allows the RPi to control a network of devices and permits the wider Internet to be notified which boards are plugged together, identifying board status and services. Mashups between the RPi and BuzzBoard Toolkit can be used in different domains such as; in health by monitoring vital signs in babies or elderly people using the BuzzMedi board, or tracking activity levels combining BuzzNav's compass and accelerometer with the Global Positioning System (GPS) in BuzzGPS. A different example is the use of the environmental sensors (humidity, temperature and barometric pressure) in BuzzSense to allow remotely monitoring and managing of intelligent environments, which can be applied to energy efficiency or assisted living. An innovative application is the use of BuzzBoards on immersive education and mixed reality laboratories [6], where geographically dispersed students can use immersive technology combined with IoT-based laboratory activities in collaborative learning sessions. The learning activities in these virtual laboratories are based on BuzzBoard modules, which have both real and virtual forms and where components can be created and moved between any of the connected virtual and real worlds. In terms of programming the RPi, the Raspberry Pi Foundation proposes the use of Python, an open-source multiplatform language that has become very popular in the teaching of programming fundamentals (http://www.python.org). However it is possible to create programs in other light-weight languages (e.g. C++). To use BuzzBoards with the RPi, the first step is to setup an SD card with RPi's linux-based operating system. To access BuzzBoard devices a programmer can utilise I^2C libraries that are provided for most popular languages. However, first the programmer will need to identify the I^2C address of the BuzzBoards using a command-line function. Finally a program can be written using a text editor and compiled using the console window or created using an open source light-weight IDE (e.g. Geany - http://www.geany.org/), generating an executable file (.py - phython or an object file for C) that can be invoked on command line.

4. Pedagogy, Computer Science and Buzz-Boards

4.1. Pedagogical Views

The nature of education involves both sides of the learning equation, the acquisition of concepts and theories by the learners and the use of this knowledge in real life situations to solve specific problems. In Computer Science education, the application of this knowledge to real world activities is an essential skill that the learners need to develop, and one of the reasons why educational institutions include laboratory activities in their curriculum programs. These laboratory activities follow the ‘learning-by-doing’ vision of Problem-based Learning (PBL), a constructivist pedagogy that encourages learners to build on their own knowledge by solving real problems co-creatively [7]. Papert et al. defined that the acquisition of this knowledge is generated by the interaction between knowledge, personal experiences and ideas in active behaviour resulting in the construction of meaningful tangible objects [8].

The application of problem-solving strategies is not limited to academic settings as, emerging technology is encouraging (or maybe requiring) people to adopt more life-long learning behaviours. An interesting example of this is the so-called *hackerspace* or *makerspace*. *Hackerspaces* have been defined as “*physical locations with tools and diverse experts who can help collaborate on projects in a wide range of scales, but it connotes a philosophy of doing things with no particular preference to empirical or theoretical methods*” [9]. One issue in maker and hacker spaces is the availability of suitable prototyping tools, which is an area that Buzz-Boards support. In a similar way, companies utilise rapid prototyping technology to design new products that meet the market requirements. Collins & Halverson [10] suggest that the use of new technologies in learning moved education from apprenticeship (where the student learns through observation and repetitive practice guided by a coach), to didacticism, (classroom-based education where knowledge is transmitted from teacher to student), to the current era where learning involves interacting with a rich technological environment. Buzz-Boards can support learning of these new technologies in both formal settings (universities) and informal (maker or personal spaces). Finally, as most science and engineering is grounded in the physical world, it is especially important, for educators, institutions and policy makers to consider how practical work should be integrated into a modern curriculum.

4.2. Computer Science Curriculums

There is no doubt that computing education is witnessing the need for huge changes to keep up with the evolving skill needs of industry which, in turn, affects computer science curricula. Fortunately, now is an especially timely moment to be considering computer science curricula as 2013 is the year that the Association of Computing Machinery (ACM) and the Institute of Electrical and Electronic Engineers (IEEE) publish their once-a-decade international curricular guidelines for undergraduate programs in computing [11]. The last complete volume was published in 2001 so the publication of 'Computer Science Curricula 2013' allows our discussion to take advantage of the most current insights into computer science curricula. In addition, the UK is currently going through a major metamorphous in pre-university computer education, with the UK Governments' Education Secretary, Michael Gove, having recently announced (January 2012) the government's intention to replace the existing ICT curriculum with a more academic Computer Science curriculum [12]. The reasons for this stem from both academia and industry, both of which have been concerned to introduce a computer science curriculum that keeps abreast of the needs of a modern technology based economy. For example, the current Engineering-UK annual report [13], identified that, in order to meet the future UK demand for engineers with Level 4+ skills (top end of pre-University), the UK needs to roughly double its output of students via HEIs and FECs. This assertion was scrutinised by the UK Department of Business, Innovation and Skills (BIS) and has become part of their Industrial Strategy. In support of UK Government policy, the Department for Education released a specification for their new Computer Science Programmes of Study for pre-University school students that was defined in terms of attainment targets which collectively aimed to teach students "*how digital systems work, how they are designed and programmed, and the fundamental principles of information and computation*" [14]. This is still in the process of being implemented so, at the time of writing, there are few clear templates for what the content may look like but, the indications are, it will entail students understanding and using computational abstractions, key algorithms, programming languages (linked to computational problem solving), Boolean logic, hardware and software architecture, networks and some system level work, including the role of specification and evaluation in design (both machine and user).

AL - Algorithms and Complexity	IAS - Information Assurance and Security	PD - Parallel and Distributed Computing
AR - Architecture and Organization	IM - Information Management	PL - Programming Languages
CN - Computational Science	IS - Intelligent Systems	SDF - Software Development Fundamentals
DS - Discrete Structures	NC - Networking and Communications	SE - Software Engineering
GV - Graphics and Visual Computing	OS - Operating Systems	SF - Systems Fundamentals
HCI - Human-Computer Interaction	PBD - Platform-based Development	SP - Social Issues and Professional Practice

Table 1 - ACM-IEEE 'Computer Science Curricula 2013' Knowledge Areas

In support of this, the influential UK 'Computing at School' Working Group released a report in March 2012, endorsed by such industry giants as Microsoft and Google, that provided a much more detailed interpretation, which identified the key concepts as being Languages, Machine and Computation; Data and Representation; Communication and Coordination; Abstraction and Design; and wider issues such as Intelligence and Ethics etc. [15]. From this it is clear that there is a move towards invariant principles which is a significant improvement on the earlier situation. As, at the time of writing (May 2013), the UK proposals are a little fluid, so we will focus on the more mature ACM-IEEE 'Computer Science Curricula 2013' [11]. This curricula is organized into a set of 18 Knowledge Areas (KAs), shown in table 1, which correspond to areas of study in computing. The authors go to great length to point out that they would not expect these knowledge areas to translate directly into equivalent courses, but rather that a good degree programme would have these topics integrated across its offerings in a way that suits the particular institution. Clearly this is a wide ranging curricula and the following section will discuss where Buzz-Boards can best support these aims.

4.3. Buzz-Boards and Computer Science

All the above reports argue, a computer science curriculum needs to be supported by substantial body of theoretical and practical knowledge. Buzz-Boards provide a practical means to implement the theory. Moreover the UK DfE stresses the importance of creative processes and projects, which the versatile system composition method employed in Buzz-Boards supports (reconfiguration via re-plugging). In more concrete computer science terms, Buzz-Boards are networked processors with a rich set of IO. Because the hardware structure is highly visible, through the process of plugging in functional units, the architecture principles are made more evident to students. The processors run an OS which can be programmed at various levels, from machine code through high-level languages such as C++ to end-user programming. Clearly there is not sufficient space to go through the ACM-IEEE Knowledge Areas, item by item, but even a cursory glance quickly reveals it would be difficult to find an area that Buzz-Boards could not support. To give just a few examples of how the ACM-IEEE curricula (see Table 1) might be supported by Buzz-Boards, AR and SF can be illustrated using the highly modularised architectural functionalities of Buzz-Boards; OS, SDF, PL, might be supported using the rich variety of programming environments afforded by Buzz-Boards; AL can be supported via the rich set of Buzz-Board sensors which provide huge opportunities for algorithms; the wide set of communications supported by Buzz-Boards, such as IP, Bluetooth, WiFi and I^2C, provide good mechanisms for NC. Beyond the basic Buzz-Board functionalities there are a huge variety of applications they can be used to illustrate. Some popular topics are the Internet-of-Things, Pervasive Computing and Intelligent Environments. All these topics are characterised by intensive use of networks, distributed computing and real-time operation. Clearly choosing application areas is also an important consideration both to support the underlying computer science and open the door to a longer term job market.

5. Buzz-Boards Value Proposition & Relevance to Wider Business

This paper has discussed the educational value of Buzz-boards in a university context explaining how they can perform as an effective pedagogical tool for teaching by meeting the needs of modern curricula and promoting an action-oriented learning environment to nurture future innovation. Undoubtedly Buzz-Boards can be recognised as the muse to inspire individuals and engineers to develop their creative ability in computing science. However, the potential values of Buzz-Boards are far more than its application to university education, as it has substantial commercial value both as a rapid-prototyping system for ICT industries worldwide, together with some potential to service a rapidly growing consumer market for educational toys and smart-phone apps (to give just some examples). Value creation can be defined from different perspectives using strategy and organisation behaviour literature; marketing and economics; or entrepreneurship theory. The core of the value creation process, regards it as adding value to market offerings of the customer, the wealth creation of the business organisation and advancement to the industries [16] [17]. Buzz-Boards can directly contribute to a manufacturers value creation by enabling them to generate prototypes of their product ideas quickly (quicker that other methods, and their competitors). In addition, the use of effective IT tools and computer AI can result in radical changes on how work is performed and managed (see case study evidence from Zheng [18]). In education it enables institutions to offer more attractive and effective options to students and staff, improving their image and impacting on their revenue generation.

We now live in a digital world with interconnected networks where embedded computing is all pervasive. The concept of Buzz-Boards has captured this emerging trend and can be further developed into a series of portfolio products serving different emergent market gaps. For instance, Buzz-free can be applied to various types of electronic consumer goods, such as smart phones for mass consumer market and BuzzBerry can support the RPi market which has massive educational and hobbyist user groups. In the long run, Buzz applications can create multiple commercial values in a more complex business environment for both B2C (business to consumer) and B2B (business to business) markets. Needless to say Buzz-Boards are full of unique and exploratory added values that have great potential to add value to a number of enterprises, especially the education market that this paper addresses.

6. Summary

In this paper we have addressed the issue of how new computer application paradigms such as the Internet-of-Things, Pervasive Computing and Intelligent Environments can be harnessed to form a highly motivating theme for teaching the latest computer science curricula. To enable this we have introduced a modular computer science laboratory kit, BuzzBoards that enables students to build a variety of products and environments, of their own design. We also explained how BuzzBoards are processor agnostic and can work with virtually any processor or system. We illustrated the potential for BuzzBoards using three examples starting with a desktop Robot (BuzzBot) assembled from a mbed ARM based processor. Robots are popular with students and have been used widely to illustrate most computer science principles, such hardware, IO, OS programming, data structures, communication and AI. We then described how a desktop intelligent environment could be built (BuzzBox) that enabled students to experiment with building, for example, smart-homes or pet-care applications. Finally, we discussed how BuzzBoards can support experimentation with the Internet-of-Things using a Raspberry Pi. In addition we highlighted our latest member of the Buzz-Board range, Buzz-Free, which is a board that enables connections to any Blue Tooth device, such as a smart-phone, enabling students to create innovative Apps that interact with the real world. While the BuzzBoard range of over 30 boards can be used for rapid prototyping of new products in companies, its roots lie in education and we therefore concluded the paper with a discussion on pedagogies and computer science curricula to show how these products might support computer science teaching. Finally, to further reinforce that discussion, we ended by presenting the value proposition that Buzz-Boards represent. More information is given in the cited references and on the BuzzBoards website www.FortiTo.com. The aim of the BuzzBoard range is not just to provide the most versatile educational technology kit for teaching computer science, but also to be the most creative and motivating approach, with the hope that, with your help, we can "*Put the Buzz Back into Computer Science Education*".

7. References

[1] V. Callaghan, "*Buzz-Boarding; Practical Support For Teaching Computing, Based On The Internet-Of-Things*", 1st Annual Conference on the Aiming for Excellence in STEM Learning and Teaching, Imperial College, London & The Royal Geographical Society, 12-13 April 2012

[2] M. Wang, V. Callaghan, M. Lear, M. Colley "*Teaching Next Generation Computing Skills; The Challenge of Embedded Computing*", Intelligent Campus 2011 (iC'11), Nottingham 26th July 2011

[3] V. Callaghan, R. McCarthy (eds), "*Introduction To Robotics In Computer Science And Engineering Education*", Special Issue of Journal of Computer Science Education, Vol 7, Issue 2, JAI Press, 1996

[4] J. Chin, V. Callaghan, '*Embedded-Internet Devices: A Means of Realising the Pervasive Computing Vision*', Proceedings of the IADIS International Conference WWW/Internet 2003, ICWI 2003, Algarve Portugal, 2003

[5] H. Sundmaeker, P. Guillemin, P. Friess, S. Woelfflé, "*Vision and Challenges for Realising the Internet of Things*", Cluster of European Research Projects on the Internet-of-Things (CERP-IoT) - European Commission, 2010.

[6] A. Peña-Ríos, V. Callaghan, M. Gardner, M. J. Alhaddad, "*Remote mixed reality collaborative laboratory activities: Learning activities within the InterReality Portal*" on Web Intelligence and Intelligent Agent Technology (WI-IAT), 2012 IEEE/WIC/ACM International Conferences - The Intelligent Campus International Symposium (IC'12), vol.3, no., pp.362,366, 4-7, Macau, China, 2012.

[7] K. Ngeow, Y. Kong. "*Learning to learn: Preparing teachers and students for problem-based learning*" [ERIC digest] 2001.

[8] S. Papert, I. Harel, "*Constructionism*", Ablex Publishing Corporation, 1991

[9] M. Altman, "*Founding a Hackerspace: An Interactive Qualifying Project Report*". (Interview). 2010.

[10] A. Collins. R. Halverson. "*Rethinking Education in the Age of Technology: The Digital Revolution and the Schools*". Distance Education, (fall), p.1-11. 2009. Available at: http://mit.kku.edu.sa/courses/media-arts-and-sciences/mas-714j-technologies-for-creative-learning-fall-2009/readings/MITMAS_714JF09_read03_coll.pdf

[11] ACM-IEEE, "*Computer Science Curricula 2013*", Ironman Draft (Version 1.0), The Joint Task Force on Computing Curricula (ACM & IEEE-Computer Society) February 2013.

[12] M. Gove "*Harmful' ICT curriculum set to be dropped to make way for rigorous computer science*", Press release, Rt Hon Michael Gove MP 'UK Department for Education (DfE),11 January 2012

[13] EngineeringUK, "*Engineering UK 2012: The state of engineering*" (a report), Weston House, 246 High Holborn, London WC1V 7EX,www.EngineeringUK.com

[14] DfE, "*Computing: Programmes of study for Key Stages 1-4*", Dept for Education, February 2013

[15] CAS, "*Computer Science: A curriculum for schools*" Computing at School Working Group, http://www.computingatschool.org.uk, March 2012

[16] D. F. Kuratko, R. M. Hodgetts, 7th ed., "*Entrepreneurship: Theory, Process, Practice*", Thomson South-Western, Mason, USA, 2007.

[17] P. Burn, *Entrepreneurship and Small Business: start-up, growth and maturity*, 3rd ed., Palgrave Macmillan, Basingstoke, UK, 2001.

[18] P. Zheng, "*Entrepreneurial Growth and Ownership under Market Socialism in China - a longitudinal study of small business growth*", *Journal of General Management*, Vol. 38, No. 2, Winter 2013