Raspberry Pi as Internet of Things hardware: Performances and Constraints

Mirjana Maksimović, Vladimir Vujović, Nikola Davidović, Vladimir Milošević and Branko Perišić

***Abstract*—The Internet of Things (IoT) ideology can be looked as a highly dynamic and radically distributed networked system composed of a very large number of identifiable smart objects. These objects are able to communicate and to interact among themselves, with end-users or other entities in the network. Entering the era of Internet of Things, the use of small, cheap and flexible computer hardware that allow end-user programming become present. One of them, considered in this paper, is the Raspberry Pi, fully customizable and programmable small computer board. Comparative analysis of its key elements and performances with some of current existing IoT prototype platforms have shown that despite few disadvantages, the Raspberry Pi remains an inexpensive computer with its very successfully usage in diverse range of research applications in IoT vision.**

***Index Terms*— Raspberry Pi; Internet of Things; Arduino; BeagleBone; Phidgets; Udoo.**

I. Introduction

The Internet of Things – IoT, can be looked as a highly dynamic and radically distributed networked system. In other words, it is a system composed of a very large number of smart objects which are identifiable, able to communicate and to interact, either among themselves, building networks of interconnected objects, or with end-users or other entities in the network [1]. The presence of smart devices able to sense physical phenomena and translate them into a stream of information data, as well as the presence of devices able to trigger actions, maximizes safety, security, comfort, convenience and energy-savings [1, 2]. Since IoT systems will be designed, managed and used by multiple stakeholders, driven by different business models and various interests, these systems should [3]:

- allow new applications to be built on top of existing systems,
- allow new systems to be deployed in parallel with existing systems,
- allow an adequate level of interoperability, so that innovative and competitive cross-domain systems and applications can be developed.

Mirjana Maksimović, Vladimir Vujović and Nikola Davidović are with the Faculty of Electrical Engineering, University of East Sarajevo, Vuka Karadžića 30, 71123 Istočno Sarajevo, Bosnia and Herzegovina (e-mail: mirjana@etf.unssa.rs.ba, vladimir_vujovich@yahoo.com, nikola.davidovic@gmail.com).

Vladimir Milošević and Branko Perišić are with the Faculty of Technical Sciences, University of Novi Sad, Trg Dositeja Obradovića 6, 21000 Novi Sad, Serbia (e-mail: tlk_milos@uns.ac.rs, perisic@uns.ac.rs).

Emerging trends of user programming give the opportunity to non-professional end-users of making additions to products, according to their specific needs. There are hundreds of products available today that allow end-user programming. Using inexpensive hardware and open source software, it is possible to programmatically control many devices in such a way that own solution meets user needs. Moreover, providing techniques to end-users and the possibility to shape products according to their needs is beneficial for both users and product developers. In this work, one of prototype platforms which enable end-user programming will be considered. An emphasis will be on Raspberry Pi computer board making a comparative study of its performances and constraints with current popular prototyping platforms [2]. The main goal of this research is to define and present advantages and disadvantages of Raspberry Pi and abilities of its usage in the development of the next generation of IoT.

The rest of this paper is organized as follows. The description of Raspberry Pi, its core components, and detail comparison with other available IoT platforms are presented in Section 2. The last section provides conclusion remarks summarizing Raspberry Pi's advantages and disadvantages as IoT hardware.

II. Comparison of The Raspberry Pi with other IoT hardware platforms

Smart objects play the central role in the IoT vision. Equipped with information and communication technology, these objects can store their context, they are networked together, they are able to access Internet services and they interact among themselves and with human beings [3].

Raspberry Pi is a small, powerful, cheap, hackable and education-oriented computer board introduced in 2012 (Fig. 1). It operates in the same way as a standard PC, requiring a keyboard for command entry, a display unit and a power supply. This credit card-sized computer with many performances and affordable for 25-35$ is perfect platform for interfacing with many devices. The vast majority of the system's components – its central and graphics processing units, audio and communications hardware along with 256 MB (Model A) – 512 MB (Model B) memory chip, are built onto single component. The Raspberry Pi board shown in Fig. 1 and Fig. 2 contains essential (processor, graphics chip, program memory - RAM) and other optional devices (various interfaces and connectors for peripherals). The processor of Raspberry Pi is a 32 bit, 700 MHz System on a Chip, which is built on the ARM11 architecture and can be overclocked for more power [4]. SD Flash memory serves as a hard drive to Raspberry Pi's processor. The unit is powered via the micro

USB connector while internet connectivity may be via an Ethernet/LAN cable or via an USB dongle (WiFi connectivity) [5, 6].

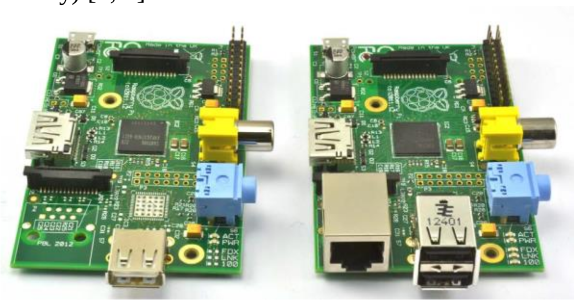


Fig. 1 The Raspberry Pi Model A (left) and Model B (right) board

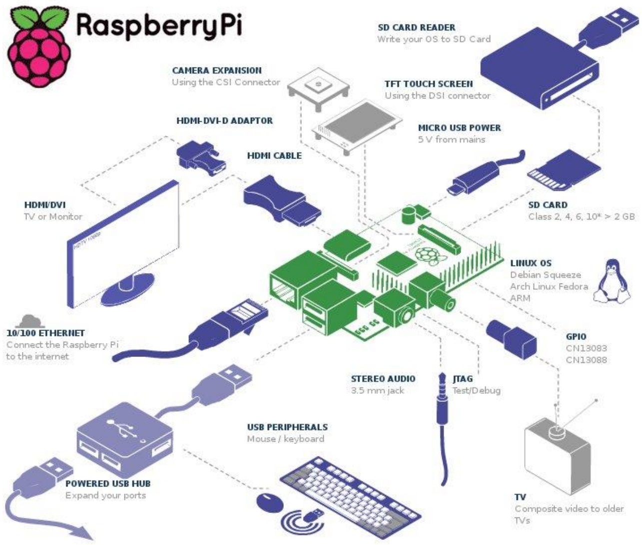


Fig. 2 The Raspberry Pi's core-components

The Raspberry Pi, like any other computer, uses an operating system. The Linux option called Raspbian is a great match for Raspberry Pi because it's free and open source, keeping the price of the platform low, and making it more hackable. There are also a few non-Linux OS options available [5].

One of the great things about the Raspberry Pi is that it has a wide range of usage. What enables it, as well as performances and constraints of Raspberry Pi will be described in rest of the paper.

The Raspberry Pi performances will be compared with following IoT prototype platforms (Fig. 3):

- Arduino – an open-source physical computing platform based on a simple microcontroller board, and a development environment for writing software for the board (Fig. 3 a). It can receive input from variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators. The microcontroller on the hardware board can be programmed using the Arduino programming language and the Arduino Integrated Development Environment (IDE). Arduino supports two working modes, stand-alone or connected to a computer via USB cable [3].
- BeagleBone Black – a single-board computer based on low-power Texas Instruments processors, using the ARM Cortex-A8 core (Fig. 3 b). It is a small credit card-sized computer which can run an operating system such as Linux/Android 4.0. The main difference between it and Arduino is that it can run a small operating system, thereby practically converting it into a minicomputer that can run programs on these operating systems. BeagleBone is designed to function at a much higher level and it has far more processing capacity than Arduino [7].
- Phidgets – a set of "plug and play" building blocks for interfacing the physical and the virtual worlds via low cost USB sensing and control from PC. Phidgets includes USB-based hardware boards for input (e.g., temperature, movement, light intensity, RFID tags, switches, etc.) and output actuators (e.g., servo motors, LED indicators, LCD text displays) (Fig. 3 d). Its architecture and API let programmers to discover, observe and control all Phidgets connected to a single computer. On the software side, all the required components are packed as an ActiveX COM Component. Each Phidget component requires a corresponding visual component, providing a visual on-screen interface for interactive end-user control. The system has an extensive library of APIs and can be used with a large number of applications, even with other toolkits in some cases. Using Phidgets enables programmers to rapidly develop physical interfaces without the need for extent knowledge in electronics design issues [3, 8].
- Udoo **-** a mini PC that can be used both with Android and Linux OS, with an embedded Arduino-compatible board. It is a powerful prototyping board for software development and design. Udoo embeds a micro-computer with the most common communication ports (Ethernet, WiFi, USB, HDMI, SATA, digital and analog input/output) and a microcontroller with a standard pinout for fast prototyping applications. Thus, Udoo is an open hardware, low-cost platform equipped with an ARM i.MX6 Freescale processor, and an Arduino Due compatible section based on ATMEL SAM3X ARM processor. It can be summarized that Udoo seeks to bring the best elements of Raspberry Pi and Arduino together into a single mini-PC. Udoo designers claim that the board will have the power of four Raspberry Pis. Udoo retail line up consists of three models, sharing most of the features and different only for connectivity and i.MX6 processor used [9]: Udoo Quad, Udoo Dual and Udoo Dual Basic.

A. Size and Cost

Ease and cost of platform deployment is directly influenced by the physical size and cost of each platform. In other words, smaller components, used as sensor nodes, can be placed in more locations and used in more scenarios. On the other side, one of the main goals of every network is to collect data from as many locations as possible without exceeding fixed budget. A reduction in per-platform cost will result in the ability to purchase more of them, to deploy a collection network with higher density, and to collect more data [7]. Table I presents size, weight and cost of Raspberry Pi compared to above mentioned prototype platforms.

TABLE I
COMPARISON OF PLATFORMS' SIZE, WEIGHT AND COST

Name	Size (mm)*	Weight (g)*	Cost per node US$*
Raspberry Pi	85.6 x 53.98 x 17	45	**25-35**
Arduino (Uno)	**75 x 53 x 15**	**~30**	30
BeagleBone Black	86.3 x 53.3	39.68	45
Phidgets	81.3 x 53,3	60	50-200
Udoo	110 x 85	120-170	99-135

**(The smaller value is better)*

B. *Power and Memory*

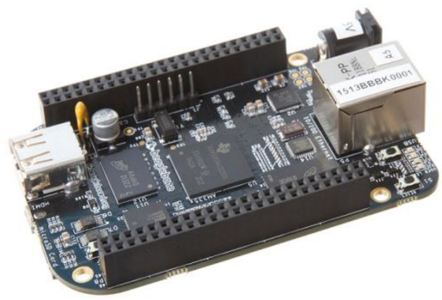
The main goal of proposed platforms is low power consumption in order to meet the multiyear application requirements. Ultra-low-power operation can only be achieved by combining both low-power hardware components and low duty-cycle operation techniques. In addition, algorithms and protocols must be developed to reduce radio activity whenever possible what can be achieved by using localized computation to reduce the streams of data being generated by sensors and through application specific protocols. One of the solutions is to combine together events from multiple sensor nodes by a local group of nodes and then transmit a single result across the sensor network [10].

The main component of the Raspberry Pi is CPU which is responsible for carrying out the instructions of a computer program via mathematical and logical operations. The ARM-based BCM2835 processor, which is cheap, powerful, and it does not consume a lot of power is the reason why the Raspberry Pi is able to operate on just the 5V 1A power supply provided by the onboard micro-USB port.

The Raspberry Pi has four distinct power modes [12]:

- The run mode - the CPU and all functionality of the ARM11 core are available and powered up.
- The standby mode – the CPU components that process instructions are shut down although the power circuits on the core are still active. In this mode the core can be quickly woken up by a process generating a special call to the CPU called an interrupt. In this mode, known as Wait for Interrupt, any current processing will be stopped and asked calling process will be performed.
- The shutdown mode - there is no power.
- The dormant mode - the core is powered down and all caches are left powered on.

a) An Arduino Uno board



b) A BeagleBone Black board

c) A Phidgets board

d) An Udoo board

Fig. 3 A prototype IoT hardware platforms

As it already stated, the Raspberry Pi for operating requires up to 700mA [11]. The unit of Raspberry Pi can be powered using a range of power sources (assuming they are able to provide enough current ~700mA) like [6]:

- computer USB Port or powered USB hub (will depend on power output),
- special wall warts with USB ports,
- mobile phone backup battery (will depend on power output),
- solar charger for cell phone,
- alkaline batteries (six rechargeable AA batteries and a voltage regulator [11]).

The Raspberry Pi's biggest limitation regarding the power supply is that no external device should draw more than 100mA from any of its USB ports [4].

Regarding the storage, the device should have sufficient memory in order to store the collected data. In addition to the amounts of storage, the program memory should also be enough to perform simple computations and transmit only the required data and routing information if the device is part of a network. It is important to note there's no hard drive on the Raspberry Pi and everything is stored on a Secure Digital (SD) Card. The minimum required size of SD card is 2 GB although large SD cards holding 32 GB, 64 GB or more are available but at the same time they are often prohibitively expensive. This storage can be expanded by using devices that provide an additional hard drive upon using the USB ports. Known as USB Mass Storage (UMS) devices, these can be physical hard drives, solid-state drives (SSDs) or even portable pocket-sized flash drives [11]. Comparative analysis of platforms' CPU, memory and power is given in Table II.

TABLE II
COMPARISON OF PLATFORMS' CPU, MEMORY AND POWER

Name	Processor	RAM*	Power
Raspberry Pi	ARM BCM2835	256-512 MB	5V/ USB
Arduino	ATMEGA8, ATMEGA168, ATMEGA328, ATMEGA1280	16-32 KB	7-12V /USB
BeagleBone Black	AM335x 1GHz ARM® Cortex-A8	512 MB	5V
Phidgets	PhidgetSBC	64 MB	6-15V
Udoo (Quad)	Freescale i.MX6Quad, 4 x ARM® Cortex™-A9 core Atmel SAM3X8E ARM Cortex-M3 CPU	**1 GB**	6-15V

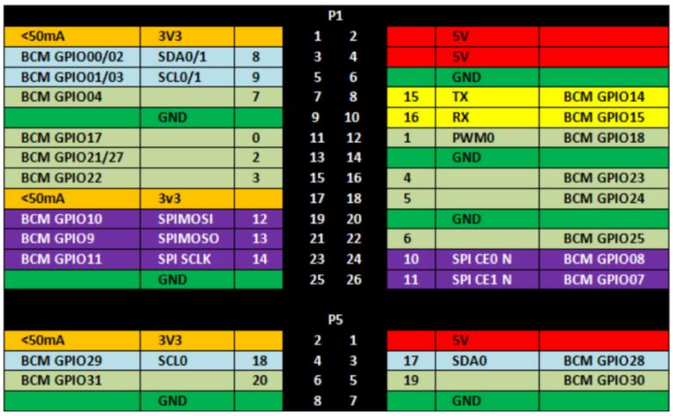
**(The higher value is better)*

C. Flexibility

The architecture must be flexible and adaptive in order to be used in wide range of applications. In addition for cost reasons it must make it easy to assemble just the right set of software and hardware components. Thus, these devices require an unusual degree of hardware and software modularity while simultaneously maintaining efficiency [10].

However, the strength of any device is its flexibility and universality. One of the great things about the Raspberry Pi is that it is very flexible and there's no single way to use it. For example, it can be used for: general purpose computing, learning to program or integrate it with electronics projects [5]. What enable a wide range of its usage are next core-components shown in Fig. 2 [2]:

- Two USB 2.0 ports allows connecting peripherals and storage devices while one micro USB serve for powering device.
- The 3.5mm analog audio jack allows connecting headphones and speakers to the Raspberry Pi what is especially useful for audio and media player based projects.
- Composite RCA port for attaching the yellow video cable from TV allows using TV as a monitor.
- The High Definition Multi-media Interface (HDMI) port allows the Raspberry Pi to be hooked up to high-definition televisions and monitors that support the technology. The HDMI port provides digital video and audio output. 14 different video resolutions are supported, and the HDMI signal can be converted to DVI, composite or SCART [5].
- Support for DSI (Display Serial Interface) - Raspberry Pi can be expanded with display.
- Support for CSI (Camera Serial Interface) - Raspberry Pi can be expanded using camera.
- The GPIO (general purpose input and output) - The Raspberry Pi has 26-pin GPIO port arranged in two rows containing 13 pins each, located on the top-left of the Pi's printed circuit board. The one row contains the even-numbered pins, and the other row contains the odd-numbered pins (Fig. 4).



			P1				
<50mA	3V3		1	2		5V	
BCM GPIO00/02	SDA0/1	8	3	4		5V	
BCM GPIO01/03	SCL0/1	9	5	6		GND	
BCM GPIO04		7	7	8	15	TX	BCM GPIO14
	GND		9	10	16	RX	BCM GPIO15
BCM GPIO17		0	11	12	1	PWM0	BCM GPIO18
BCM GPIO21/27		2	13	14		GND	
BCM GPIO22		3	15	16	4		BCM GPIO23
<50mA	3v3		17	18	5		BCM GPIO24
BCM GPIO10	SPIMOSI	12	19	20		GND	
BCM GPIO9	SPIMOSO	13	21	22	6		BCM GPIO25
BCM GPIO11	SPI SCLK	14	23	24	10	SPI CE0 N	BCM GPIO08
	GND		25	26	11	SPI CE1 N	BCM GPIO07
			P5				
<50mA	3V3		2	1		5V	
BCM GPIO29	SCL0	18	4	3	17	SDA0	BCM GPIO28
BCM GPIO31		20	6	5	19		BCM GPIO30
	GND		8	7		GND	

Fig. 4 Raspberry Pi' P1 and P5 GPIO connectors [12]

The GPIO pins can accept input and output commands and thus can be programmed on the Raspberry Pi. It is important to note that there is a certain difference between GPIO pins schedule among model A and model B of the Raspberry Pi. Some of GPIO pins can be used as digital inputs/outputs and as interfaces for embedded protocols. Thus, the GPIO enables the Raspberry Pi to communicate with other components and circuits, and allows it to act as a controller in a larger electronic circuit (GPIO pins can be accessed for controlling hardware such as LEDs, motors, and relays (examples of outputs) and used to read the status of buttons, switches, and dials, or to read sensors like temperature, light, motion, proximity sensors and many other (examples of input) [4]). In other words, the GPIO port, are the main way of connecting with other electronic boards and through them it's possible to communicate to other computing devices using a variety of different protocols including Serial Peripheral Interface (SPI) and Inter-Integrated Circuit (I²C):

- I²C – low-speed interface – Inter-Integrated Circuit (I²C) is a serial bus interface which supports multiple devices and only requires two wires for communication. It's work on relatively low speeds [14].
- SPI – Serial Peripheral Interface Bus (SPI) is a synchronous full-duplex (two ways) serial connection [15].

In addition to standard GPIO port, Raspberry Pi Model B Rev 2 has an expanded set of connectors [13]. It is important to mention P5 header which is made up of 8 pins (+3.3 V, +5 V, two ground pins and four GPIO pins that can provide the second I²C protocol) and P6 header with two pins – their short circuiting provides soft reset of BCM2835.

Table III analyses the expansion connectors of Raspberry Pi and other platforms what make easy their connection to a wide variety of external peripherals.

TABLE III
COMPARISON OF PLATFORMS' EXPANSION CONNECTORS

Name	Analog inputs	Digital I/O pins	USB ports
Raspberry Pi	0	14	1-2
Arduino	6	14	1
BeagleBone Black	6	14	1
Phidgets	8	8I+8O	6
Uddo (quad)	14	62+14	5

D. *Communication*

The Raspberry Pi's Ethernet port is main gateway for communication with other devices. It is auto-sensing which means that it may be connected to a router or directly to another computer (without the need for a crossover cable) [5, 6]. The model B has a standard RJ45 Ethernet port while model A doesn't, but can be connected to a wired network by an USB Ethernet adapter. USB Ethernet adapter has two-speed mode, 10 Mb/s and 100 Mb/s (Table IV). With a cable connected, the Raspberry Pi will automatically receive the details it needs to access the Internet when it loads its operating system through the Dynamic Host Configuration Protocol (DHCP). This assigns the Raspberry Pi an Internet Protocol (IP) address on network, and tells it the gateway it needs to use to access the Internet (typically the IP address of router or modem). The disadvantage of Raspberry Pi lies in lack of integrated WiFi module but this support can be added by USB dongles. In such a way the Raspberry Pi can be used for creating ad-hoc networks or to connect to a wide range of wireless networks, including those running on the latest 802.11n high speed standard [11]. Raspberry Pi can it serves as static websites, but it can also generate dynamic content using databases and web applications. In addition, it can even provide access to its GPIO ports via web technologies. Also, Raspberry Pi can be used as Sensor Web node by connecting it to a network so it can be accessed from other computers [4].

TABLE IV
COMPARISON OF COMMUNICATION INTERFACES

Name	LAN (MBit)	WiFi Module
Raspberry Pi	10/100	-
Arduino	-	-
BeagleBone Black	10/100	-
Phidgets	-	-
Uddo (quad)	10/100/1000*	+*

**(except Udoo Dual Basic model)*

E. *Operating systems and programming languages*

Regardless of the hierarchical approach each device, as part of Internet of things, still needs a program, and the most common approaches to programming each smart device, is to either program it using some form of operating system or to choose a higher level of abstraction. The operating systems vary from traditional operating systems in terms of goals and technique and each system differs substantially in the approach to memory protection, dynamic reprogramming, thread model, real-time features, etc. [12].

The Raspberry Pi uses an operating system called Raspbian (Table V) based on Linux but there are also a few non-Linux OS options available. The preferred OS are Linux distribution (Debian, Fedora Remix and Arch Linux) because they are free, but mainly because they run on the Raspberry Pi's ARM processor [4]. There are several reasons for deciding to go with the Raspbian operating system [12]:

- Raspbian has a desktop environment similar to Windows and Mac called Lightweight X11 Desktop Environment (LXDE), so it provides an easy transition for those not familiar with Linux.
- It comes pre-installed with software useful for writing codes.
- The operating system has been tailored to run on the Raspberry Pi. The code compilation is optimized for on-chip floating-point calculations (hard-float) rather than a slower software-based method.
- There is wide spread community support for the operating system.

TABLE V
OPERATING SYSTEMS AND PROGRAMMING LANGUAGES

Name	Board operating system	Programming language
Raspberry Pi	Raspbian, Ubuntu, Android, ArchLinux, FreeBSD, Fedora, RISC OS	C, C++, Java, Phyton
Arduino	/	Arduino
BeagleBone Black	Linux Angstrom	Arduino
Phidgets	Linux	Visual Basic, VB.NET, C#, C/C++, Flash/Flex, Java, Labview, Matlab, ActionScript 3.0, Cocoa
Udoo	Ubuntu, Android, Linux, ArchLinux	Arduino, C, C++, Java

III. Raspberry Pi's performances and constraints

Raspberry Pi advantages can be summarized as follows:

- The Raspberry Pi is a small independent computer that runs on the various distribution of Linux operating system and can be programmed as needed.
- It has a very large working memory (RAM memory).
- It has expandable memory to store the data (up to 64GB).
- It works on multi operating processor (supports a set of instructions).
- It operates at speeds from 700 MHz to 1000 MHz.
- It has support for USB 2.0 which allows its expansion with a large number of peripherals.
- Depending of the needs it is possible to expand the Raspberry Pi with WiFi and Bluetooth adapters (power and range can be changed by changing the adapter).
- Expansion and communication with network devices over a LAN adapter are possible.
- It can be expanded with various prototype shields (Pi-Face, GSM/GPRS & GPS, GPIO expansion board, GertBoard)
- It is possible to form an expandable system with various electronic components (sensors and electronic circuits) using digital inputs and outputs, I^2C or SPI protocols.
- C, Python or object oriented languages such as C++ and Java can be used for programming of Raspberry Pi.
- It can be powered form battery or sollar cell.
- It can be run in server mode.
- A various web server can be installed and running on Raspberry Pi.

Based on the above mentioned it can be noted that for small amount of money the Raspberry Pi comes with a lot of nice things, but it also lacks some useful features. The main disadvantages of Raspberry Pi are [4]:

- It does not have a real-time clock (RTC) with a backup battery but it can easily work around the missing clock using a network time server, and most operating systems do this automatically.
- It does not have a Basic Input Output System (BIOS) so it always boots from an SD card.
- It does not support Bluetooth or WiFi out of the box but these supports can be added by USB dongles.
- Unfortunately, most Linux distributions are still a bit picky about their hardware, so it should be first checked whether flavor of Linux supports particular device.
- It doesn't have builtin an Analog to Digital converter. External component must be used for AD conversion.
- It has a relatively small number of digital I/O, but it can be expanded with external logic devices.

IV. Conclusion

Raspberry Pi's performances are compared with some popular boards and development platforms on a general level by computing power, size and overall costs of the solutions. Based on performed analysis, it can be stated that Udoo has the best performances among considered IoT hardware platforms, but at the same time its price is quite high. On the other side the detail analyses of Raspberry Pi have shown that as ultra-cheap-yet-serviceable computer board, with support for a large number of input and output peripherals, and network communication is the perfect platform for interfacing with many different devices and using in wide range of applications. In other words, the Raspberry Pi brings the advantages of a PC to the domain of sensor network, what makes it the perfect platform for interfacing with wide variety of external peripherals. Coupling it with WiFi and providing access to the Internet it is possible to set it up for a remote communication, what the Raspberry Pi makes very suitable for applications in IoT concept. Thus, the advantage of Raspberry Pi lies in its flexibility and endless possibility of its usage enabling at the same time end-users to program it according their needs and budgets.

Even there are large differences between stated platforms in their ideal use cases, energy requirements, OS, etc., it can be noted that all of them can be very successively applied as IoT hardware components. Choosing between one and the other will depend on project's requirements.

References

[1] D. Miorandi et al., "Internet of things: Vision, applications and research challenges," Ad Hoc Networks 10 (2012) 1497–1516

[2] A.K. Dennis, Raspberry Pi Home Automation with Arduino, Packt Publishing, USA, 2013

[3] D. Uckelmann, M. Harrison, F. Michahelles, Architecting the Internet of Things, Springer 2011

[4] M. Schmidt, Raspberry Pi – A Quick Start Guide, The Pragmatic Bookshelf, 2013

[5] M. Richardson and S. Wallace, Getting started with Raspberry Pi, O'Reilly, USA, 2013

[6] Raspberry Pi Getting Started Guide, RS Components, Vsn 1.0, 2012

[7] Beagle Board, Available: http://beagleboard.org/Products/BeagleBone%20Black, [21.1.2014]

[8] Phidgets, Available: http://www.phidgets.com/, [21.1.2014]

[9] UDOO Starting manual (beta) v04, (2013), Available: www.udoo.org, [21.1.2014]

[10] J. L. Hill, System Architecture for Wireless Sensor Networks, PhD Thesis, University of California, Berkley 2003

[11] E. Upton and G. Halfacree, Raspberry Pi User Guide, Wiley, 2012

[12] B. Horan, Practical Raspberry Pi, Apres, USA, 2013

[13] Raspberry PI Rev2 – P1 Connector, Available: http://www.combinatorialdesign.com/boards/Raspberry_Pi/P1, [1.12.2013]