



Contents lists available at ScienceDirect

## Computers and Electrical Engineering

journal homepage: [www.elsevier.com/locate/compeleceng](http://www.elsevier.com/locate/compeleceng)

# Raspberry Pi as a Sensor Web node for home automation <sup>☆</sup>

**Vladimir Vujović\***, Mirjana Maksimović

Faculty of Electrical Engineering, Vuka Karadžića 30, 71123 East Sarajevo, Bosnia and Herzegovina

**ARTICLE INFO****Article history:**

Received 15 April 2014

Received in revised form 29 January 2015

Accepted 29 January 2015

Available online xxxx

**Keywords:**

Home automation

Raspberry Pi

Sensor Web

RESTful

Internet of things

**ABSTRACT**

The world of home automation is an exciting field that has exploded with new technologies and today is known as an area where "The internet of things" (IoT) vision becomes reality. The primary advantages that stem from this concept include how each device forms a small part of the Internet, by which the advanced system is able to interact and communicate, maximizes safety, security, comfort, convenience and energy-savings. This paper proposes an implementation of Sensor Web node as a part of IoT using a Raspberry Pi – inexpensive, fully customizable and programmable small computer with support for a large number of peripherals and network communication. Using this technology, in an example of monitoring and determining the confidence of fire in building, a full system, based on Sensor Web elements, is created and developed starting from a scratch. The given example confirms the advantage of Raspberry Pi – flexibility and extensive possibility of its usage.

© 2015 Elsevier Ltd. All rights reserved.

## 1. Introduction

The "Internet of Things" – IoT, can be viewed as a highly dynamic and widely distributed network system. In other words, it is a system comprising many identifiable components that are able to communicate and to interact, either among themselves or with end-users or other entities in the network [1]. Home-automation, by way of smart devices that sense physical occurrences and translate them into a stream of information data, maximizes safety, security, comfort, convenience and energy-savings (Fig. 1) [1,2]. The building elements of home automation are sensor networks and Building Management System (BMS). Using Sensor Web elements, as a part of IoT, in opposition to the standard sensor elements, the communication infrastructure rapidly changes from local to global, moving a BMS to global level too. Moreover, home automation system in a whole becomes ubiquitous – appears everywhere and anywhere.

Thus, home automation can be defined as a mechanism removing as much human interaction as technically possible and desirable in various domestic processes and replacing them with programmed electronic systems. Ultimately it is a system that aims to heighten quality of life with the automation of household activity that may be controlled over the Internet or telephone.

Originally, this system was used to control HVAC (Heating, Ventilation and Air-Conditioning) as well as fire safety and security, controlled by a central computer [3]. With the development of modern technologies home automation includes various features for security, surveillance, lighting, energy management, access control, entertainment-appliances, interfaces and software. The home automation system should be easy to understand and operate, as well as providing for easy

☆ Reviews processed and approved for publication by the Editor-in-Chief.

\* Corresponding author. Tel./fax: +387 57 342 788.

E-mail addresses: [vladimir.vujovic@etf.unssa.rs.ba](mailto:vladimir.vujovic@etf.unssa.rs.ba) (V. Vujović), [mirjana@etf.unssa.rs.ba](mailto:mirjana@etf.unssa.rs.ba) (M. Maksimović).

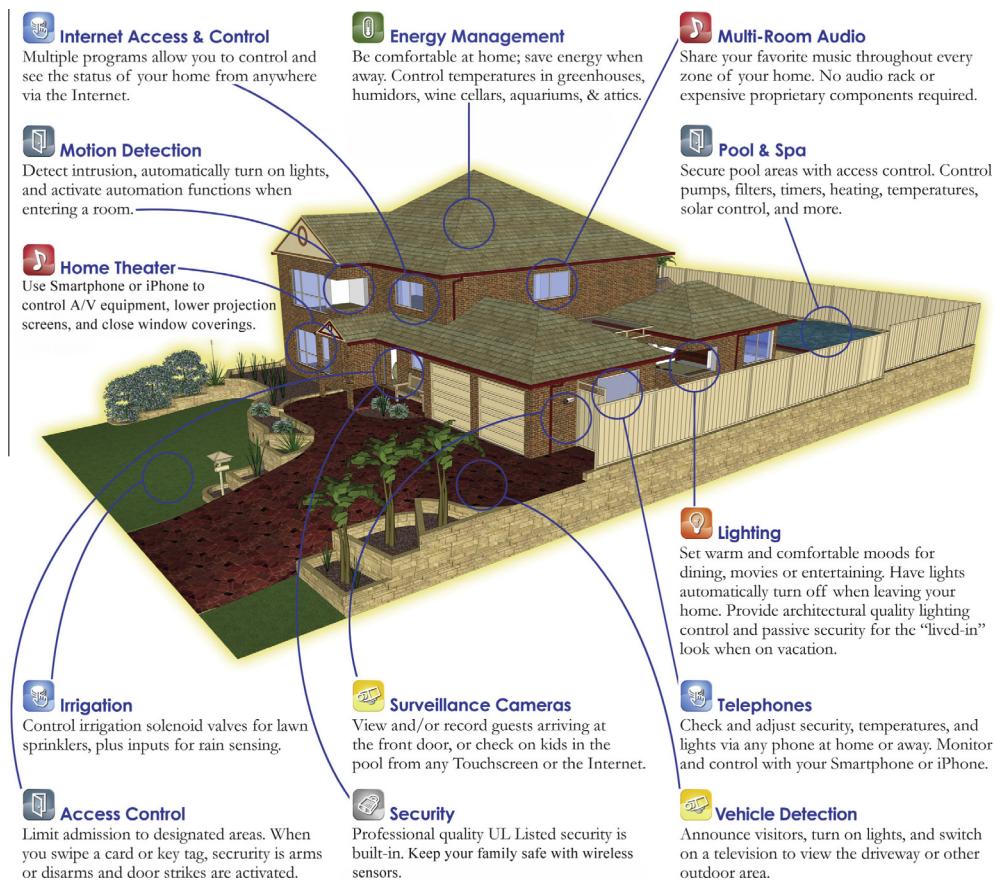


Fig. 1. Integrated home automation system and its benefits [3].

expansion according to users' needs and individual budget. More importantly, home automation provides for an alternative solution to the reduction of energy consumption, cost expenditure as well as increased convenience [3].

Research performed by Berg Insight [4] states that at the end of 2012, in the countries EU27 + 2, there were a total of 1.06 million smart home systems in use. This demonstrates that the European market for smart home systems is approximately three years behind North America in terms of penetration and market maturity. Berg Insight forecasts that the installed smart home systems in EU27 + 2 will grow and reach 17.4 million systems by 2017 (Fig. 2).

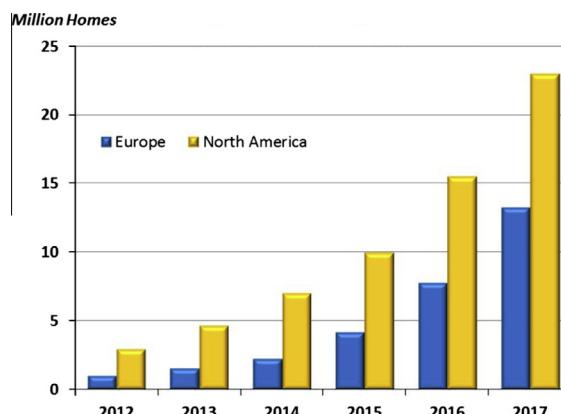


Fig. 2. Total number of smart homes (Europe and North America 2012–2017) [4].

There are three types of home automation systems:

- Individual control devices,
- Distributed-control systems and
- Centrally controlled systems.

Individual devices control only one appliance or function, examples include; programmable thermostats, motion detectors, occupancy sensors, photocell lighting controls and timers. A distributed-control system allows for individual appliances to communicate internally with each other over the existing electrical wiring without a central controller (although keyboard entry is possible using telephones or personal computers). A centrally controlled communication system routes signals between a central computer and appliance controllers or environmental sensors. The leading advantage of this system is that it can control some “dumb” appliances as well as “smart” appliances but if the controller fails, the whole system fails [5].

It is important to note that an automated home can be a very simple grouping of controls, or it can be heavily automated whereby any appliance that is plugged into electrical power is remotely controlled. There are hundreds of products available today that allow control over the devices automatically, either by remote control; or even by voice command. But commercial home automation systems are usually expensive (costs mainly include equipment, components, furniture and custom installation) and user can't modify anything to tailor the system to its own home. On the other side, by using inexpensive hardware and open source software, it is possible to programmatically control many devices in a building in such a way that own solution meets user's specific needs. This is known as Do-It-Yourself (DiY) category in home automation where customer creates its own source code, has the measurements, the metrics, the methods, the knowledge to build and to troubleshoot, repair, and enhance its own solution. In this way it is possible to realize the system which will include everything user would expect from a commercial home automation system. Therefore, 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 the prototype platforms which enable end-user programming in order to build a complete open-source home automation system is considered. The aim is to show how to make devices, components and appliances web aware so that managing and monitoring can be done by web browser. In addition, it is possible to make the IP (Internet Protocol)-enabled devices and components to be also aware of other components on the network and to be capable to interact directly with each other without human interference or a central management computer. In other words, different devices and the appliances in the home, as home environment, being connected to the Internet, can be controlled remotely or continuously monitored. This contributes to overall cost reduction and energy saving which is one of the main concerns of today.

This paper presents Raspberry Pi usage as a Sensor Web node for home automation with a fire-safety example (monitoring and determining the confidence of fire in a building). Raspberry Pi computer, which was introduced in 2012, is currently a mainstream system subject to widespread availability that can be used in home automation. As a small and powerful tool it enables the home automation enthusiast, programmer and electronic hobbyist to enhance their homes with sensors and software [2]. In order to justify the practicality of Raspberry Pi within home automation a comparison with other sensor nodes and development platforms is performed [6,7]. The subsequent analysis has shown that Raspberry Pi is an ultra-cheap-yet-serviceable computer board with support for a large number of input and output peripherals. This makes it the perfect platform for interaction with many different devices and usage in a wide range of applications. Combining Raspberry Pi with Wi-Fi it can communicate remotely, therefore increasing suitability for the construction of wireless sensor nodes and Sensor Web nodes. Raspberry Pi can also be used as a processing node in Wireless Sensor Networks (WSN), not just as a sensor node but also as a controller. In addition, data processing and decision-making can be based on artificial intelligence, what is performed in developed testbed on the example of monitoring and determining the confidence of fire.

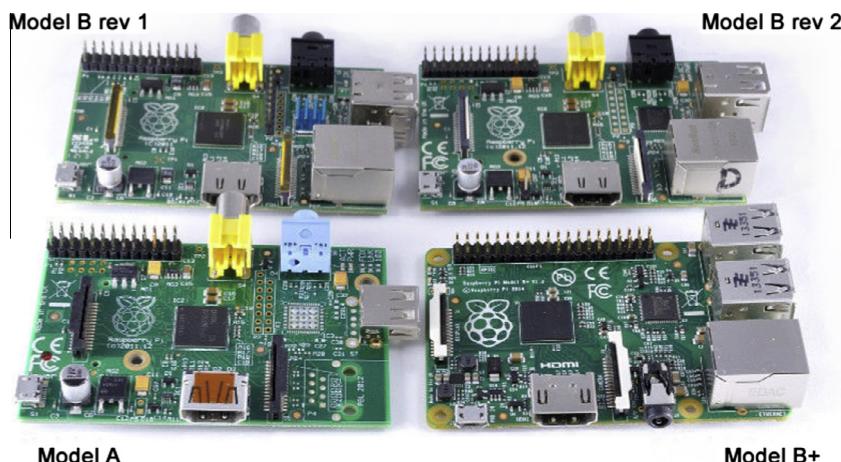
The rest of this paper is organized as follows; Section 2 highlights the State of the Art, Section 3 presents an implementation of three components used for building a Raspberry Pi as a Sensor Web node, Section 4 provides an example of client application for data visualization and processing as one possible way of application. Finally, Section 5, based on the performed research and obtained results, provides the concluding remarks and outlines potential directions for future work.

## 2. State of the art

Raspberry Pi (Fig. 3) is a small computer with the dimensions; 85.60 mm × 53.98 mm × 17 mm, weighing only 45g and affordable for 25–35\$. This makes it perfect for home automation, where a small device can easily be placed in a case and mounted inside an electrical box, or replace an existing thermostat device on a wall [2]. The Raspberry Pi board contains a processor and graphics chip, program memory (RAM – Random-Access Memory) and various interfaces and connectors for external devices. Some of these devices are essential, others are optional, but all Raspberry Pi models have the same processor, a system on a chip (SoC) named BCM28351. It is cheap, powerful, and it does not consume a lot of power [8].

The unit of Raspberry Pi can be powered using a range of power sources (assuming they are able to provide enough current ~700 mA) [9]:

- Computer USB (Universal Serial Bus) Port or powered USB hub (will depend on power output),
- Special wall warts with USB ports,



**Fig. 3.** Raspberry Pi Models.

- Mobile Phone Backup Battery (will depend on power output),
- Solar charger for cell phone.

The Raspberry Pi can be also powered from alkaline batteries. The most suitable pack, which fulfils the relatively high power requirement, is made of six rechargeable AA batteries and a voltage regulator [10].

The Raspberry Pi has four distinct power modes [11]:

- The run mode – the central processing unit (CPU) and all functionality of the ARM11 core are available and powered up.
- The standby mode – the main core clocks are shut down (the parts of the CPU that process instructions are no longer running) although the power circuits on the core are still active. In this mode, known as “Wait for Interrupt” (WFI) mode, the core can be quickly woken up by a process generating a special call to the CPU called an interrupt. This interrupt will stop any current processing and do what the calling process has asked for.
- The shutdown mode – there is no power.
- The dormant mode – the core is powered down and all caches are left powered on.

Raspberry Pi operates as a standard personal computer (PC), requiring a keyboard for command entry, a display unit and a power supply, and as a Web Server, when peripherals (keyboard and display) are not needed. In other words, Raspberry Pi fits perfectly as a server device which is working with whole network of single purpose clients' devices.

Secure Digital (SD) flash memory card is configured in such a way that it mirrors a hard drive to Raspberry Pi's processor. Although large SD cards holding 32 GB, 64 GB or more are available, they are often prohibitively expensive, but the minimum required size of an SD card is 2 GB depending on the distribution demands of the operating system. If possible, the SD card class 10 should be used. The storage can be expanded, by using devices that provide an additional hard drive upon using the USB ports. These are known as USB Mass Storage (UMS) devices and can be physical hard drives, solid-state drives (SSDs) or even portable pocket-sized flash drives [12]. It is also important to note that the Raspberry Pi Model A has 246 MB of RAM while the Models B and B+ have 512 MB.

Internet connectivity may be via an Ethernet/LAN cable. Model B has a standard RJ45 Ethernet port, while Model A does not, however it can be connected to a wired network through an USB Ethernet adapter. The Raspberry Pi Model B Ethernet port 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). Wi-Fi connectivity via an USB dongle is an alternative option [13]. Using such a device, 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 [12].

The Raspberry Pi, like any other computer, also uses an operating system (OS) and the “stock” OS is a flavor of Linux called Raspbian. Linux is a great match for Raspberry Pi because it is a free and open source. On the one hand, it maintains a low price for the platform, but on the other hand it is easier to hack. There are also a few non-Linux OS options [13].

One of the positive aspects to the Raspberry Pi is that it is very flexible and can be used for a variation of purposes. For example, it can be used for general purpose computing, learning to program or integration with electronics projects [13]. The following core components enable the wide range of its usage [2] (Fig. 4):

- Two USB 2.0 ports allow connecting peripherals and storage devices while one micro USB serve for powering the device.
- The 3.5 mm analog audio jack allows for the connecting of headphones and speakers to the Raspberry Pi, which is particularly useful for audio and media player based projects.

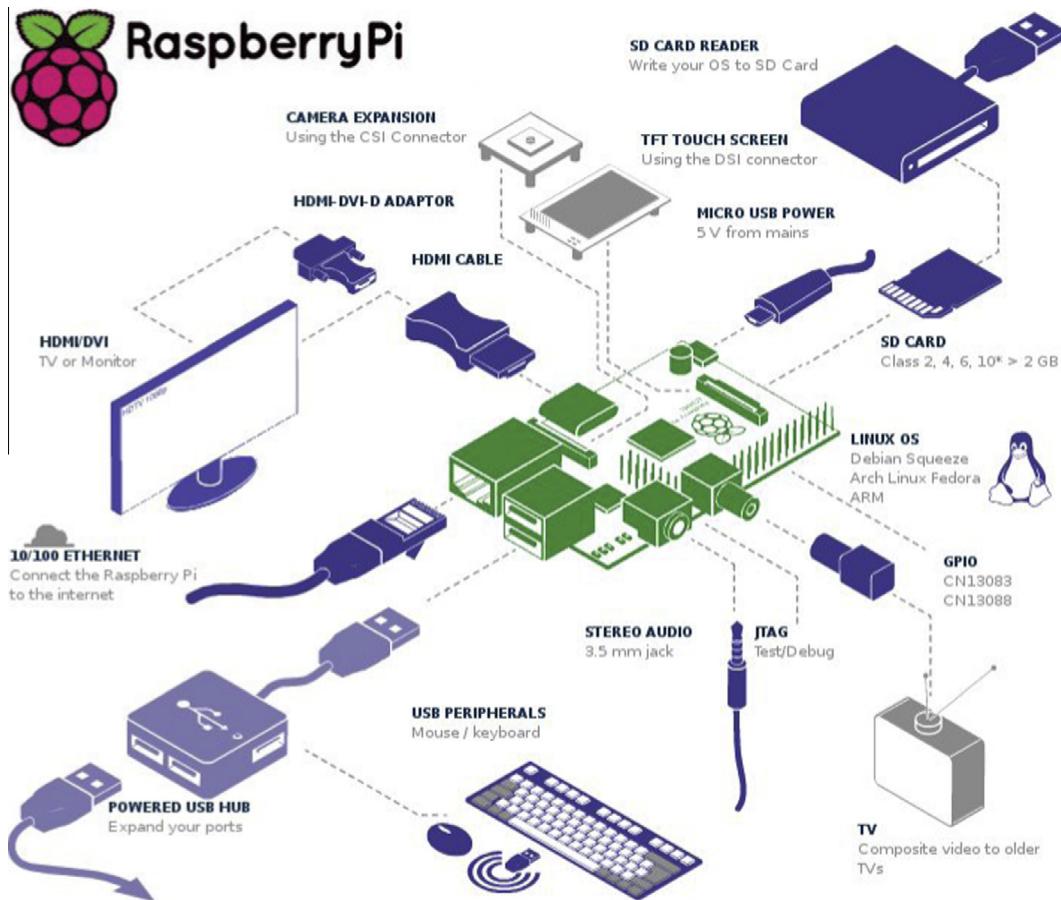


Fig. 4. Raspberry Pi Model B components.

- 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. It is also used for streaming video and audio from the web to TV.
- 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) pins on the Raspberry Pi are the principal manner for connecting with other electronic boards. Raspberry Pi Model B, which is used in this paper, consists of 26 pins arranged in two rows containing 13 pins each. The first row contains the even-numbered pins, and the other row contains the odd-numbered pins (Fig. 5). There is a specific difference between the orders of the GPIO pins in Model A to the order of those in the Raspberry Pi's Model B. The GPIO pins can accept input and output commands and thus can be programmed on the Raspberry Pi. They can be accessed for controlling hardware such as LEDs, motors, and relays, which are all examples of outputs. As for inputs, Raspberry Pi can read the status of buttons, switches, and dials, or it can read sensors like temperature, light, motion, or proximity sensors (among many others) [8].

Some of GPIO pins can be used as digital inputs/outputs and as interfaces for embedded protocols. Two of the most important protocols based on their widespread use are:

- I<sup>2</sup>C – low-speed interface – Inter-Integrated Circuit (I<sup>2</sup>C) is a serial bus interface, which supports multiple devices and only requires two wires for communication. It's work on relatively low speeds.

- SPI – Serial Peripheral Interface Bus (SPI) is a synchronous full-duplex (two ways) serial connection.

- Extended GPIO – in addition to standard GPIO port Raspberry Pi Model B Rev 2 has an expanded set of connectors. It is important to mention the 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 I<sup>2</sup>C protocol) and the P6 header with two pins – their short-circuiting allows for the soft reset of BCM2835.

It is important to note that in July 2014, a Model B+, the higher-spec variant of the Raspberry Pi is presented. Compared to the Model B used in this paper it has: more GPIO (40 vs. 26) (Fig. 5), more USB (4 USB 2.0 ports vs. 2), micro SD, better audio, neater form factor and the most important: lower power consumption and the same price.

## GPIO Numbers

Raspberry Pi B			Raspberry Pi A/B			Raspberry Pi B+			
Rev 1 P1 GPIO Header			Rev 2 P1 GPIO Header			B+ J8 GPIO Header			
Pin No.			Pin No.			Pin No.			
3.3V	1	2	5V			3.3V	1	2	5V
GPIO0	3	4	5V			GPIO2	3	4	5V
GPIO1	5	6	GND			GPIO3	5	6	GND
GPIO4	7	8	GPIO14			GPIO4	7	8	GPIO14
GND	9	10	GPIO15			GND	9	10	GPIO15
GPIO17	11	12	GPIO18			GPIO17	11	12	GPIO18
GPIO21	13	14	GND			GPIO27	13	14	GND
GPIO22	15	16	GPIO23			GPIO22	15	16	GPIO23
3.3V	17	18	GPIO24			3.3V	17	18	GPIO24
GPIO10	19	20	GND			GPIO10	19	20	GND
GPIO9	21	22	GPIO25			GPIO9	21	22	GPIO25
GPIO11	23	24	GPIO8			GPIO11	23	24	GPIO8
GND	25	26	GPIO7			GND	25	26	GPIO7

**Fig. 5.** Raspberry Pi Model A, B and B + GPIO connectors

**Table 1**

## The differences between Raspberry Pi models.

 Model B+	Model B	Model A	
Chip	Broadcom BCM2835 SoC full HD Multimedia Applications Processor	Broadcom BCM2835 SoC full HD Multimedia Applications Processor	Broadcom BCM2835 SoC full HD Multimedia Applications Processor
RAM	512 MB SDRAM @ 400 MHz	512 MB SDRAM @ 400 MHz	256 MB SDRAM @ 400 MHz
Storage	MicroSD	SDCard	SDCard
USB 2.0	4× USB Ports	2× USB Ports	1× USB Port
Power draw/voltage	600 mA up to 1.8 A @ 5 V	750 mA up to 1.2 A @ 5 V	600 mA up to 1.2 A @ 5 V
GPIO	40	26	26

After performed analysis, the differences between the Raspberry Pi models presented in Fig. 3 can be summarized (Table 1).

Analysis of the Raspberry Pi illustrates that it has expansion connectors which support analog inputs, digital I/O, I<sup>2</sup>C, SPI and UART (Universal Asynchronous Receiver/Transmitter). This makes their connection to a wide range of external peripherals and simultaneously Raspberry Pi's applications much easier. It is necessary to outline the advantages and disadvantages of Raspberry Pi, alongside the abilities of its usage as a sensor node. In order to do so a comparative study of its performances and constraints with current popular wireless sensor nodes and prototype platforms is performed. This involves a comparison, on a general level, by computing power, size and overall costs of the solutions (presented in Tables 2 and 3) [6,7].

Based on the comparative performed it is possible to summarize Raspberry Pi's advantages and disadvantages for usage in home automation projects, looking to those of other systems [6,7]:

- Advantages: processing power, memory, connectivity, multipurpose usage (USB), Linux OS, programming in high-level object oriented languages, the community around which is much bigger than the other platforms' communities and price.
  - Disadvantages: power consumption and weight.

With regards to sizing, power and pricing presented in this section it can be concluded that the Raspberry Pi is qualified for an application in home automation projects (which constitutes as an important use within the ten uses for the Raspberry Pi stated in [14]). The additional hardware and software requirements can be achieved through already existing hardware modules and open source software. Thus, it can be summarized that Raspberry Pi is powerful but still easy enough for beginners to use, compared to other considered prototype platforms which can't be used as a standalone computer or are targeted more at advanced users and serious developers. One of the main reasons, besides above mentioned Raspberry Pi's advantages, is the fact that the Raspberry Pi is best suited for projects that require a graphic interface or the Internet. This advantage makes Raspberry Pi perfect for usage in Internet of Things vision.

**Table 2**

Comparison of Raspberry Pi's performances with current existed wireless sensor nodes.

Name	Processor	RAM	External memory	Operating system	Size (mm)		Weight (g)	Cost per node US\$
Raspberry Pi	ARM BCM2835	256–512 M	2–64 G	Raspbian	85.6 × 53.98 × 17		45	25–35
MicaZ	ATMEGA128	4 K	128 K	Tiny os, Mote runner	58 × 32 × 7		18	99
TelosB	TI MSP430	10 K	48 K	Tiny os, Sos, Mantisos	65 × 31 × 6		23	99
Iris	ATMEGA1281	8 K	128 K	Tiny os, Mote runner	58 × 32 × 7		18	115
Cricket	ATMEL128L	4 K	512 K	Tiny os	58 × 32 × 7		18	225
Lotus	ARM NXP LPC1758	64 K	512 K	Rtos, Tiny os	76 × 34 × 7		18	300

**Table 3**

Comparison of Raspberry Pi's performances with similar prototype platforms.

Name	Processor	RAM	Power	Operating system	Programming	Analog inputs	Digital I/O pins	USB ports	LAN (MBit)	WiFi Module	Size (mm)	Weight (g)	Cost per node US\$
Raspberry Pi	ARM BCM2835	256–512 MB	5 V USB	Raspbian, Ubuntu, Android, ArchLinux, FreeBSD, Fedora, RISC OS	C, C++, Java, Phyton	0	8–27	1–4	10/100	–	85.6 × 53.98 × 17–85 × 56 × 17	42–45	25–35
Arduino	ATMEGA8, ATMEGA168, ATMEGA328, ATMEGA1280	16–32 KB	7–12 V USB	/	Arduino	6	14	1	–	–	75 × 53 × 15	~30	30
BeagleBone Black	AM335x 1 GHz ARM® Cortex-A8	512 MB	5 V	Linux Angstrom	Arduino	6	14	1	10/100	–	86.3 × 53.3	39.68	45
Phidgets	PhidgetSBC	64 MB	6–15 V	Linux	Visual Basic, VB.NET, C#, C/C++, Flash/Flex, Java, Labview, Matlab, ActionScript 3.0, Cocoa	8	8I + 8O	6	–	–	81.3 × 53.3	60	50–200
Udoo (quad)	Freescale i.MX6Quad, 4 x ARM® Cortex™-A9 core Atmel SAM3X8E ARM Cortex-M3 CPU	1 GB	6–15 V	Ubuntu, Android, Linux, ArchLinux	Arduino, C, C++, Java	14	62 + 14	5	10/100/ 1000	+	110 × 85	120–170	99–135

Author of [10] identifies a wide range of techniques for using the Raspberry Pi with sensors, displays, motors, and so on. In the workshop [15] a Raspberry Pi is used to read sensors (inputs), store their values in a database for historical trending and turn relays (outputs) on and off when a sensor value goes outside of a certain range. In other words, a system that is built whereby a light is turned on if the door is opened or motion is detected in a given area. Also, the temperature is recorded into the database by which any increase in temperature outside of a set range triggers a fan to be turned on to reduce the temperature. Several home automation projects using Raspberry Pi are also presented in [16], in which wireless sensors and a Raspberry Pi are identified as opening up a world of possibilities. The alternative projects include:

- Home alarm system project,
- Temperature sensor project,
- Webcam Surveillance project,
- Siren project.

Authors of [17] further highlight that sensor networks are just one example of how these small, powerful, and inexpensive components can be used. Using the example of XBee–ZigBee Mesh network authors demonstrate how to build your own sensor network with just moderate skill levels. Furthermore, the possibility of using a Raspberry Pi board as a ZigBee home automation server is presented in [14]. This can be set up to support a range of tasks, including remote controlled air conditioning and lighting, and checking whether doors are open or closed. Works [2,10,17–20] propose combining the Raspberry Pi with the power of the open source Arduino platform, and present numerous home automation projects. For this purpose authors use different technologies and languages. In works [10,18] Phyton is used for programming the Raspberry Pi, while a multitude of technologies including X10, C-Bus, ZWave, ZigBee, and Hue as well as use of Perl, PHP, C++, and Bash are presented in [19]. Alongside the informative aspect to the projects, the authors also give readers ideas for new projects.

To deal with main challenges of home automation (high cost of ownership, inflexibility, poor manageability and difficulty achieving security) in this paper is presented an approach how to design and to implement a cheap and open source home automation system. This system would be capable of controlling and automating appliances through an easy manageable web interface to run and maintain the home automation system. In this paper, Raspberry Pi is used as a Sensor Web node for home automation purposes, where a whole system is built from the beginning.

The first difference between our approach and approaches presented in previous mentioned works is in choosing Java for the prototype implementation which is one of the leading modern programming languages. For the creation of the driver of the sensors the open source Pi4J library is used. This is a bridge between the native and Java libraries for full access to the Raspberry Pi. A Java web service was preinstalled on the processing unit in order to make a Raspberry Pi as a Sensor Web node. In this instance it is Apache Tomcat 7 server that enables Java services to run. The Apache Tomcat 7 has a very simple configuration as well as optimal utilization of hardware. Thus, the developed testbed advance is in its modularity – developed drivers written in Java provide multiple use of written software in future projects. In other words, the Linux operating system delivers additional advantages for using Raspberry Pi as a Sensor Web node. Programming in high-level languages such as C, C++, Python, or Java, solution implementation is quite simple and it is accessible to a large number of users, as opposed to micro controller programming which usually depends on the development kit.

In paper [10] the communication between sensors and Raspberry Pi over SPI interface is presented. In our prototype communication is performed via the I<sup>2</sup>C interface because this requires only two wires to connect devices even if it has several devices to interface, while SPI needs  $3 + n$  wires (where  $n$  is number of connected devices). Currently the usage of the I<sup>2</sup>C interface is widespread and numerous electronics have implemented this interface. In addition, the method of connecting analog sensors and the system development through A/D (analog-to-digital) converters and associated electronics is explained in detail.

For the interaction between Raspberry Pi and clients authors of presented papers used a Web Server. On construction and implementation of a survey, we discovered that the RESTful service concept is subject to widespread use in Sensor Web technology. Following this assessment, a framework Jersey 2.2 [21], which is running and executing on Apache Tomcat server, is used in prototype building process for the implementation of RESTful services of Raspberry Pi. Applying service-oriented technologies enables the easy development of distributed systems and their integration from the broker system into the larger information systems.

In addition to all above mentioned it is important to emphasize that WSNs are typically used to monitor some parameters of environmental processes which are complex, ambiguous and vagueness embedded in their nature. Considering this, sensor readings can be imprecise and unreliable so using crisp values to describe WSN parameters sometimes is not the most suitable approach. Considering that WSNs are envisioned as an integral part of the future Internet – IoT vision, supporting its extension to the physical world, the incorporation of soft computing technologies in sensor nodes may lead to potential network performance improvements, since it provides effective parameter combination and can be directly executed by the sensor nodes. Integration of soft computing technologies (fuzzy logic, neural networks, fuzzy rule-based systems, data mining techniques ...) in sensor nodes is an example of an application adapted to WSNs. In this way using fuzzy values instead of crisp ones significantly improves the accuracy of event detection (it is believed that crisp values can't adequately handle the often imprecise sensor readings). Therefore, WSN empowered by fuzzy logic detection mechanism work as an intelligent and power efficient sensing network. This fact is used in the development of our prototype with the aim of increasing the accuracy of the system, reducing false alarms and power consumption (using adequate rule base reducing techniques).

It can be summarized that the principal advantage of the developed testbed put forth in this paper is the low energy and resource consumption. This is because the Raspberry Pi eliminates many elements deemed unnecessary from its system. The Raspberry Pi runs in server mode and all graphical user interfaces as well as the third part of the application software are removed. By installing the Web Server on the unit and providing access to the Internet, Raspberry Pi becomes a complete and ideal system (hardware and software) for building Sensor Web nodes. In addition, application of fuzzy logic in the process of reasoning minimizes data transfer and increases system reliability. Hence, involving fuzzy logic in decision making process leads to improving the overall system performance what is the main advantage of proposed system compared to other presented in literature.

### **3. Raspberry Pi based Sensor Web testbed**

For the control of intelligent home via the Internet in real-time, a simple and convenient infrastructure is constructed based on the client/server architecture. For the purposes of chosen home automation project the rest of the paper presents, in detail, three realized basic elements:

- Raspberry Pi connection with sensors that gather information from the environment,
- Creation of the driver with which sensors would be accessed and thereby the access and operation be facilitated, and
- Raspberry Pi conversion to the RESTful service, which will be available via Internet.

#### *3.1. Raspberry Pi as a sensor unit*

As it already stated, Raspberry Pi can be directly used in electronics projects mostly because it has GPIO pins in addition to digital I/O support, I<sup>2</sup>C and SPI protocols (used to perform communications with elements of sensing units). Most of the modern electronic devices, logic circuits and microcontrollers support one of the two protocols, as such representing the advantage of the Raspberry Pi.

The sensor node prototype created in this work is composed of a sensing unit, which includes temperature sensors and an A/D converter, and a processing unit, which includes a processing and communications unit. Raspberry Pi type B is used as the processing unit and rest of the paper describes a way of transforming this into a sensing unit.

The most common sensor elements in the case of fire monitoring and detection are temperature sensors, which according to the type of implementation can be [22]: Thermistor, Resistive Temperature Device (RTD), Thermocouple and Integrated Silicon Linear Sensor, while the mode can be analog or digital.

There are quite a few different ways for reading an external temperature from the Raspberry Pi's GPIO pins. Many different types of sensors and different connection methods can be used. Depending on the chosen temperature sensor (analog or digital), a simple thermistor, or something more advanced, based on the 1-wire protocol (DHT11, DHT22) can be used. Otherwise, if analog sensors are used, then it is necessary to convert the measured value into digital one so that Raspberry Pi is able to register and understand it. The biggest drawback to Raspberry Pi is the lack of a built-in A/D and D/A converter, and for use with an analog value, there is a need to use logic circuits that provide translation of analog values into digital. For this purpose logic circuits which communicate with the Raspberry Pi via I<sup>2</sup>C protocol (PCF8591) or SPI protocol (MCP3004/3008) are used in the developed testbed. It is important to state that both of these circuits represent an 8-bit A/D converter, which provides 256 steps. In the case of requiring a higher precision system, the used A/D converter can be replaced with any model of higher precision, such as a 10-bit, 16-bit, and so on.

In this paper, prototype sensor unit is composed of two analog temperature sensors (thermistor models: B57045k10 and 10kNTC chosen based on their price, availability and precision) and I<sup>2</sup>C-bus integrated circuit 8-bit A/D and D/A converter (model: PCF8591). An electronic circuit scheme of proposed prototype has been given in Fig. 6.

Sensors characteristics are shown in Table 4.

$T_0$  and  $R_{T0}$  parameters in Table 4 are reference temperature and resistance, respectively, at which  $B$  parameter is calculated. For both sensors that temperature is 25 °C, and resistance is 10,000 Ω.  $B$  parameter can be calculated empirically, but usually all three parameters,  $T_0$ ,  $R_{T0}$  and  $B$  are given in a sensor data sheet.  $\Delta R_R/R_R$  is a standard sensor deviation. Since thermistors work on the principle of temperature–resistance proportion, and only the known value at the input side of thermistor is power supply, it is necessary to correlate these two values. In other words, the thermistor resistance value has to be calculated from the known value of power supply. The electrical resistance of the thermistor can be obtained using a voltage divider consisting two resistors, one with a variable and unknown resistance and the other with a fixed and known resistance [23]. In the prototype model fixed resistance of 10,000 Ω, sensor and A/D converter power supply in the range of 3.27–3.3 V are used.

Fig. 7 shows the voltage divider:

- $V_s$  – Supply voltage (3.27–3.3 V)
- $V_0$  – Measured voltage
- $R_T$  – Resistance of thermistor
- $R$  – Known resistance (10 kΩ)
- $I$  – Current

Next two equations describe voltage divider:

$$V_s = I(R + R_T) \quad (1)$$

$$V_0 = IR \quad (2)$$

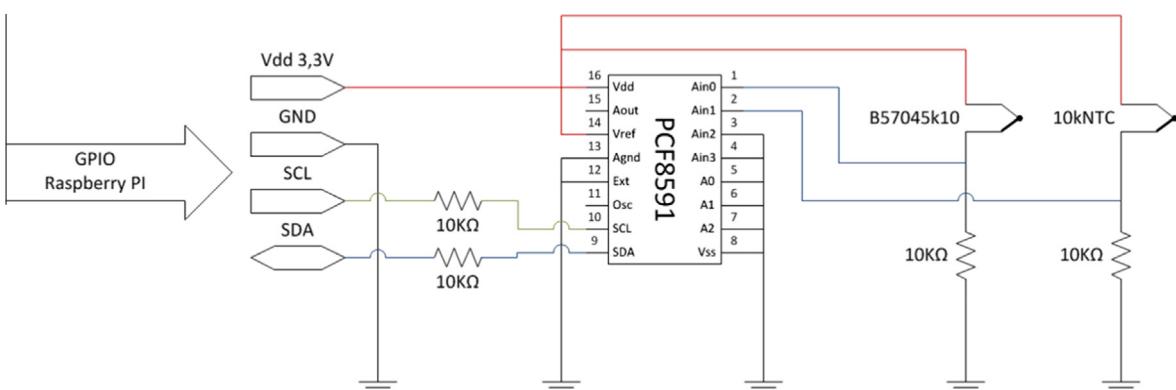
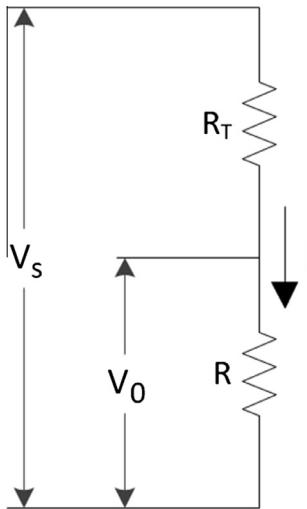


Fig. 6. Electronic circuit scheme of Raspberry Pi as a sensing unit.

**Table 4**  
Sensors' parameters.

	B57045k10	10kNTC
$T_0$	25 °C	25 °C
$R_{T0}$	10,000 Ω	10,000 Ω
$B_{(25/100,25/50)}$	4300 K	3950 K
$\Delta R_R/R_R$	±10%	±1%
Range	-55–125 °C	-20–105 °C
Additionally	-	Waterproof



**Fig. 7.** Voltage divider.

Eq. (3) is derived from Eqs. (1) and (2):

$$\frac{V_s}{R + R_T} = \frac{V_0}{R} \quad (3)$$

And the final equation for thermistor' resistance is:

$$R_T = \left( \frac{V_s}{V_0} - 1 \right) R \quad (4)$$

The thermistor temperature for NTC Thermistor can be computed with the Steinhart–Hart equation [24] with the  $B$  parameter:

$$\frac{1}{T} = a + b \ln(R_T) + c \ln^3(R_T) \quad (5)$$

For

$$a = \left( \frac{1}{T_0} \right) - \frac{1}{B} \ln(R_0), \quad b = \frac{1}{B}, \quad c = 0 \quad (6)$$

Steinhart–Hart equation is:

$$\frac{1}{T} = \frac{1}{T_0} + \frac{1}{B} \ln \left( \frac{R_T}{R_0} \right) \quad (7)$$

Temperature  $T$  is expressed in Kelvin. If parameters  $T_0$ ,  $R_0$ ,  $B$  and current resistance of thermistor are known, then temperature can be calculated.

After defining all details and parameters for calculation of required values, there is a need to create a driver on processing unit that allows reading information from the A/D converter and converting them to a form suitable for further processing.

### 3.2. Building a driver for operating with a sensor

The previous chapter described how the sensors and the A/D converter are connected to the processing unit via GPIO ports – over the I<sup>2</sup>C protocol, where the thermistor voltage is read. In order for the Raspberry Pi to read the data from the sensor or A/D converter it is necessary to create software that has the ability to access the integrated circuit. After reading the data, depending on the type of sensor, the value has to be converted to the appropriate data, e.g. in the case of the thermistor it is the temperature. This software is called the driver and it enables much easier access to information, whereby the driver itself takes responsibility for the communication with the sensor unit as well as the data reading and writing.

For the creation of the driver, the Pi4J library is used. This is a bridge between the native and Java libraries for full access to the Raspberry Pi. The Pi4J library is highly dependent on Wiring Pi Native Library and Oracle JDK. Since Wiring Pi is a part of the Pi4J library, only Oracle JDK must be installed on the Raspberry Pi operating system, which in our case is the Linux-Debian "Wheezy".

The principal issues with drivers are:

- How to access the bus and select the device (each device on the bus has a unique address)?
- How to access the selected device and write/read the information?

To solve the first issue there is a need to know the address of the device and the number of the bus (as Raspberry Pi Model B rev 2 has two I<sup>2</sup>C buses). To find the required data the next command at the prompt of Linux should be used:

```
sudo i2cdetect -y 1
```

The result of executing this command will be similar like the one shown in Fig. 8.

The number presents the address of the devices connected on I<sup>2</sup>C bus 1 (in this case, device address is 48).

To solve the second issue it is necessary to know the architecture of the device, in our case, the I<sup>2</sup>C-bus integrated circuit 8-bit A/D and D/A converter (model: PCF8591). PCF8591 uses a control register, which is required to control the device function. Control register structure (available in integrated circuit datasheet) is shown in Fig. 9.

With the case presented in this work the most important two LSB bits in the control register represent A/D channel number and allow selection of one of the four A/D converter inputs. As it shown in Fig. 6, the thermistor B57045k10 is connected with the Ain0 input of the A/D converter while the Ain1 input is connected with the 10kNTC thermistor. The selection of the desired input is performed by entering the values in the last two bits. Since all remaining bits are 0, the selection of desired input can be done by sending the byte values of 0 × 00, 0 × 01, 0 × 02, 0 × 03.

After the input voltage value is read from one of the A/D converter inputs it is converted to an 8-bit binary code. The result from the conversion is placed into the A/D converter data register and awaits transmission. The first byte transferred in a waiting cycle mode contains the results of the previous measurement, which means that in the first reading there is no data, but it sends a control byte – for PCF8591 it is the hexadecimal value 80. Reading speed from PCF8591 is limited by the I<sup>2</sup>C bus speed. In Fig. 10 is presented a bus protocol for read mode, A/D conversion.

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
00:	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
10:	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
20:	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
30:	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
40:	--	--	--	--	--	--	--	--	48	--	--	--	--	--	--	--
50:	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
60:	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
70:	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

Fig. 8. Output from i2cdetect command.

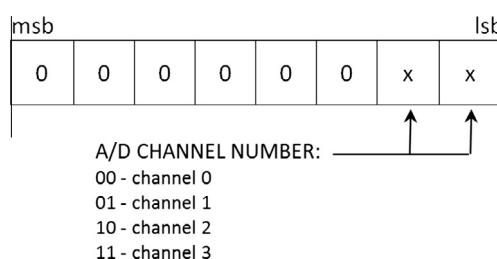
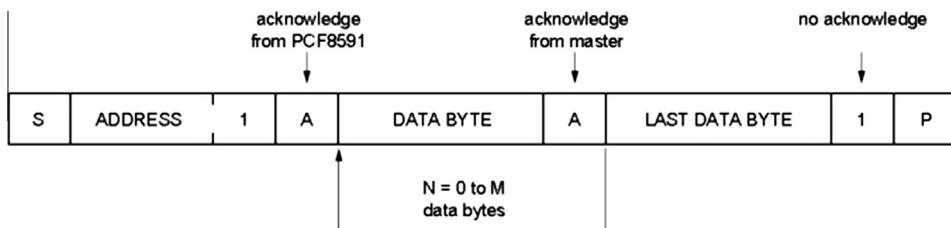


Fig. 9. Control register on PCF8591 AD/DA converter.

**Fig. 10.** Bus protocol for read mode, A/D conversion.

Using Java (Oracle) object-oriented programming language five objects are created (due to the size of the code only certain parts will be shown):

*I2CDeviceClass* – class responsible for connecting and communicating with devices on the I<sup>2</sup>C bus. Using objects and methods from the Pi4J library, an instance of *I2CBus* object is created, and used to access the I<sup>2</sup>C bus. The *I2CDevice* object, which is responsible for the operation of the device (writing and reading from devices) is created by forwarding the address of the device.

```
I2CDevice b57045k10 = I2CFactory.getInstance(1).getDevice(Integer.decode("0x00").intValue());
```

*ADConversion* – possesses a method for calculating the thermistor resistance based on the output voltage of the A/D converter. Eq. (4) is used for the calculation. It is necessary to know the reference voltage value and the resolution of the A/D converter.

```
public double resistance(int adVal, int resistance)
{
    vRef = 3.269;
    resolution = 255;
    refVoltage = vRef/resolution;

    return ((vRef/(refVoltage*adVal))-1)*resistance;
}
```

*Thermistor* – possesses a method for calculating the temperature in °C based on Eq. (7). For the calculation it is necessary to know the parameters of the thermistor (given in Table 4).

```
public double temperatureCelsius(double resistance)
{
    double t0 = 298.15;
    double r0 = 10000;
    double b = 4300;

    double resVal = resistance/r0;
    double tmpVal = (1/t0)+((1/b)*Math.log(resVal));

    return (1/tmpVal) - 273.15;
}
```

*B57045K10* and *NTC10K* (because of the similarity only *B57045K10* object will be presented) – the class is responsible for creating all of the previously mentioned objects: for creating connections (*I2CDeviceClass*) and supporting classes (*ADConversion*, *Thermistor*). Class also provides methods for reading the temperature from the sensor.

The first step is selecting the channel on the A/D converter by setting bits in the Control Register:

```
b57045k10.write(Integer.decode("0x00").byteValue());
```

Next step is reading values from the Data register:

```
b57045k10.read(buffer, 0, 2);
...
byte[] buffer = new byte[2];
int bufl = buffer[1] & 0xFF;
```

and the value conversion:

```
Thermistor B57045K10 = null;
ADCConversion adconversion = null;
...
DecimalFormat format = new DecimalFormat("###.##");
...
String data = format.format(B57045K10.temperatureCelsius(adconversion.resistance(buf1, 10000)));
```

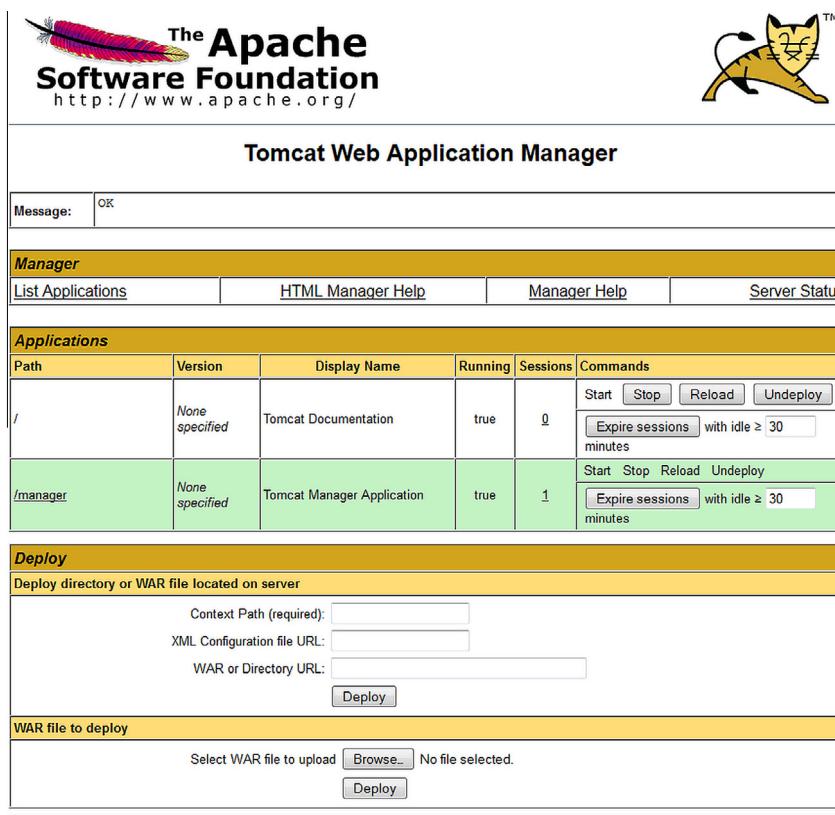
Access to the values of the driver is available by calling the method `getTemperature()` from class `B57045K10`.

### 3.3. Raspberry Pi as a RESTful service

The processing unit is a key element for the construction of a Raspberry Pi based sensor node. In addition to the sensor data processing it is necessary for the processor unit to have the ability to run and execute a Web Server, on which services will be deployed. In this work, a REST (Representational State Transfer) based service, which will be the mediator between the sensor driver and an end-user or a customer, is considered. Web Servers can be lightweight, but also very heavyweight in terms of resource use in the CPU [25]. Depending on the needs of the sensor node and CPU power, it may be desirable to perform server optimization.

In this paper Java is chosen for the prototype implementation – to make a Raspberry Pi as a Sensor Web node prototype. Apache Tomcat 7 server, which allows for Java service to run, is preinstalled on the processing unit and it is chosen because of its very simple configuration and optimal utilization of the hardware [26]. Framework Jersey 2.2 is used for the implementation of RESTful services.

After creating a Web application, deployment on Apache Tomcat is executed via Web oriented interface shown in Fig. 11. The simplicity delivered by the management console of the Apache Tomcat server is suitable for a large number of users because there is no need for additional tools and knowledge to perform the administration. Applications come after creating “packed” in WAR (Web application ARchive) files, which, after deployment to the server, become executed.



**Fig. 11.** Apache Tomcat Administration console.

A REST [27] is an architectural model for how distributed applications are built, and are founded on three concepts: representation, state, and transfer:

- Representation: Data or resources are encoded as representations of the data or the resource and these representations are transferred between the clients and servers.
- State: All of the necessary state required to complete a request must be provided with the request. The clients and servers are inherently stateless. A client can't rely on any state to be stored in the server, and the server can't rely on any state stored in the client. This doesn't, however, pertain to the data stored by servers or clients, only to the connection state needed to complete transactions.
- Transfer: The representations and the state can be transferred between client and servers.

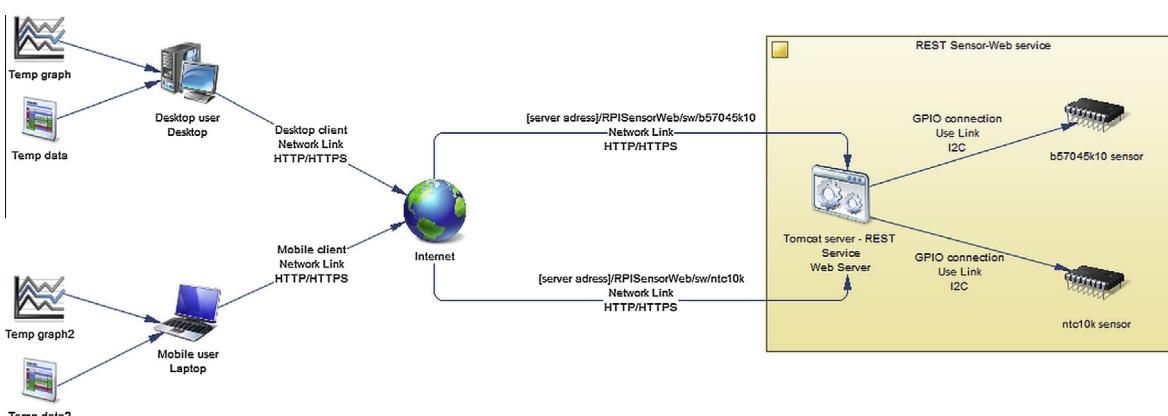
REST is an architectural model that can be efficiently implemented as a combination of the Hypertext Transfer Protocol (HTTP) and Transmission Control Protocol/Internet Protocol (TCP/IP). With this instantiation of REST, HTTP requests are used to transfer representations of resources between clients and servers. Each element represents a resource that has a unique Uniform Resource Identifiers (URIs) address, the access method [28], the parameters and the type of sent data. During the creation of Web Services for a sensor node, it is necessary to define all the available resources and manners for how to access them, as well as additional elements that affect the performance of the sensor and the whole system. At the end of the sensor node, resources represent sensors and the information they collect, but also resources can be aggregated from all sensors to the inclusion of information that is already processed at the sensor node.

The infrastructure diagram of prototype model is shown in Fig. 12. The diagram is divided into four main components: the clients (mobile or desktop user), the communication channel (Internet or LAN – Local Area Network), the service (Raspberry Pi as a Sensor Web element) and the sensing units. With regards to the client it is important to note that it is possible to use any client that supports an execution of URI address call in order to access a Sensor Web. In our case, that is the Java programming language. The majority of the currently used systems have this support by default. The communication channel used to access the Sensor Web is the Internet itself – based on the TCP/IP protocol. Finally, the service, the Sensor Web element includes all remaining elements, communication unit (which owns the IP address), the CPU (which is able to collect, process and disseminate data), sensors and A/D converter.

Using two temperature sensors described in Fig. 6, a prototype of RESTful service that implements a method for obtaining data from individual sensors, as well as methods that can display data collectively from all sensors in the sensor nodes, are created. Due to the fact that both sensors connected to the sensor node in the prototype do not require input parameters, but only read data, for all the methods the GET access method is used. The requirement for reducing the amount of transferred data is very important for the realization of Sensor Web node. This requirement, as the primary choice for communications between the service and the client, implies using the JSON (JavaScript Object Notation) because of its simplicity, processing speed and low permeability, while communication itself is realized via HTTP/HTTPS protocol [29].

For reading data from both sensors of Sensor Web node two methods are given:

```
[server_address]/RPISensorWeb/sw/b57045k10
[server_address]/RPISensorWeb/sw/b57045k10/data
and
[server_address]/RPISensorWeb/sw/ntc10k
[server_address]/RPISensorWeb/sw/ntc10k/data
```



**Fig. 12.** The infrastructure diagram of prototype model.

Although JSON is used for the information exchange, the reason for existence of two addresses is to reduce the amount of transferred data. By calling the first address the client receive JSON, which contains only information about the measured value:

```
{"temp": "23.55"}
```

The second address call implies forwarding a JSON with information of sensor type and sensor model in addition to the measured value.

```
{"type": "thermistor", "model": "b57045k10", "temp": "23.87"}
```

Aggregated information about the sensors can be accessed via three addresses:

```
[server_address]/RPISensorWeb/sw/info
[server_address]/RPISensorWeb/sw/info/data
[server_address]/RPISensorWeb/sw/info/number
```

The first address provides to the client information of the sensor number, data of all sensors and the measured value.

```
"sensor_number": 2, "data": [{"type": "thermistor", "model": "b57045k10", "temp": "23.23"}, {"type": "thermistor", "model": "ntc10k", "temp": "23.07"}]
```

The second address returns only information about sensors and measured values:

```
{"data": [{"type": "thermistor", "model": "b57045k10", "temp": "22.58"}, {"type": "thermistor", "model": "ntc10k", "temp": "22.37"}]}
```

while the third address sends only data of sensor number in the sensor node:

```
"sensor_number": 2 .
```

Implementation of b57045k10 class of RESTful Services is given with:

```
@Path("/b57045k10")
public class b57045k10
{
    @Context
    private HttpServletRequest request;
    @Context
    private ServletContext context;

    @GET
    @Produces(MediaType.APPLICATION_JSON + ";charset=utf-8")
    public String getSensorData()
    {
        B57045K10 b57045k10 = new B57045K10("thermistor", "b57045k10", "1", "0x48");

        return b57045k10.getTemperature().toString();
    }

    @GET
    @Path("/data")
    @Produces(MediaType.APPLICATION_JSON + ";charset=utf-8")
    public String getAllSensorData()
    {
        B57045K10 b57045k10 = new B57045K10("thermistor", "b57045k10", "1", "0x48");

        return b57045k10.getAllData().toString();
    }
}
```

As it can be seen, getting the information from the sensors is done by calling created drivers of individual sensors. After data processing, the information is sent to the user as above described JSON which is a lightweight data-interchange format. It is easy for human beings to read and write and simpler for machines to parse and generate messages than using Extensible Markup Language (XML).

#### 4. Raspberry Pi as a Sensor Web in action

With the threats, risks and dangers that exist today, fire and security systems in particular have come to play an increasingly significant role in life-safety operations. Fire can cause severe damage to the infrastructure and internal area, as well as life threatening conditions. Therefore, early and accurate fire detection is important for the prompt extinguishing of the fire, and the reduction of damages and loss of life. It is obvious, thus, that fire-safety systems are essential. Their integration within a home automation system, through fast delivery of data, quick response, access control, intrusion, video surveillance, fire detection and alarm, and emergency communications, enables effective incident management in turn making people feel safer in their workplaces and their homes.

For these reasons, an analysis of data sensed at the Raspberry Pi used as a Sensor Web node, fire monitoring and detection client components that perform real-time data collection, processing and visualization of data from the sensor, are required. Generally, data visualization is the process in which computing and digital technologies are used to display abundant information – quantitative and qualitative information, mathematical or scientific models, and measured, statistical or calculated data. The importance of data visualization can be found in many different fields. In real-time systems visualization has the role of translating the observed values into animated and interactive 2D and 3D graphical models.

Depending on the nature of the problem to which the observed sensor node belongs, it is possible to provide detailed data processing and analysis, as well as their collection and storage in the database with a purpose of subsequent processing and analysis. This will be presented in rest of the paper. With reference to the presented prototype applications, raw data is current information that has been read from the above described temperature sensors at Raspberry Pi Sensor Web node (obtained by exposing proposed two sensors to artificial fire). The next two components are created with the purpose of showing the functionality of our solution: Data Grapher – the component that performs the visualization of measured data in function of time, and Data Processing – the component that performs processing of the sensor node measured data.

##### 4.1. Data Grapher

The visualization of the measured data using the Data Grapher component is presented in Fig. 13. The Data Grapher architecture is based on RESTful client and is designed for the visualization of the data obtained from a given RESTful Sensor Web node prototype. The collected data is in JSON format (plain format), thus the whole data processing is done on the client's side. As a result the client has the option of manipulating a wide range of settings and the data output can be customized. Since the data from the sensors is observed in real-time, the primary goal is to set a time frame in which the measurements are performed, as well as an explanation and elaboration of the measurement range and sampling interval.

In the settings of the Data Grapher is offered the possibility of selecting a sensor type from a range, in order to cover the different types of sensors. An essential element of visualization is also the choice of the ratio in which the measured data are presented and the selection of the visual identity of the chart.

##### 4.2. Data Processing

The function of the Data Grapher component is for the good visualization of the measured sensor data, while the Data Processing component processes them. Data processing component can be based on simple, predefined rules and at the same time on complex mathematical models or decision-making based on rules in the domain of artificial intelligence.



**Fig. 13.** Data Grapher – Graphical visualization of sensor data.

Legend:						
#####	- Very low critical					
####	- Low critical					
####	- Medium critical					
####	- High critical					
#####	- Very high critical					
Date of measurement: 01.08.13						
No.:	Time:	PrevTemp:	CurrTemp:	Delta:	Appr.1:	Appr.2:
<hr/>						
	Start at: 22:14:27					
1.	22:14:28	19,40	19,40	+ 00,00	19,98	25,00
2.	22:15:27	19,40	21,95	+ 02,55	22,03	21,71
3.	22:16:27	21,95	23,87	+ 01,92	23,96	23,84
4.	22:17:27	23,87	31,51	+ 07,64	36,37	48,83
5.	22:18:27	31,51	35,44	+ 03,93	40,35	41,01
6.	22:19:27	35,44	46,54	+ 11,10	61,41	80,86
7.	22:20:27	46,54	55,49	+ 08,95	75,53	86,78
8.	22:21:27	55,49	62,08	+ 06,59	81,37	81,37
9.	22:22:27	62,08	53,17	- 08,91	69,49	70,45
10.	22:23:27	53,17	44,27	- 08,90	58,60	58,60
11.	22:24:27	44,27	38,47	- 05,80	45,83	46,08
12.	22:25:27	38,47	35,07	- 03,40	40,53	40,53
13.	22:26:27	35,07	32,56	- 02,51	37,57	37,55
14.	22:27:27	32,56	27,45	- 05,11	31,97	30,74
15.	22:28:27	27,45	22,58	- 04,87	22,85	25,00
16.	22:29:27	22,58	24,19	+ 01,61	24,26	24,20
17.	22:30:27	24,19	48,96	+ 24,77	64,17	83,25
18.	22:31:27	48,96	79,22	+ 30,26	91,38	92,20
19.	22:32:27	79,22	84,84	+ 05,62	92,20	91,96
20.	22:33:27	84,84	42,11	- 42,73	55,10	55,10
21.	22:34:27	42,11	32,56	- 09,55	37,57	37,57
22.	22:35:27	32,56	29,80	- 02,76	35,36	35,00
23.	22:36:27	29,80	28,45	- 01,35	32,70	32,52
24.	22:37:27	28,45	28,79	+ 00,34	32,92	32,92
25.	22:38:27	28,79	25,16	- 03,63	25,51	25,52
26.	22:39:27	25,16	23,23	- 01,93	23,43	25,00
Stop at: 22:40:05						

**Fig. 14.** Data processing – processing data at client side.

Creating clients with different strategies of sensor node data processing implies the advantage of access where the processing unit and the sensor are located at the same place or are directly linked. Access to the sensor data over the Internet – via REST services, provides competitive performance of more than one client where each can be based on a different strategy as discussed earlier.

Data Processing component of the created prototype collects information of previous and current measured temperature values of the sensor. The sensed air temperatures are included in the system of decision-making. Decision-making is executed according to two predefined algorithms for estimating the likelihood of fire based on fuzzy logic principles [30] and the results of both algorithms are presented in Fig. 14.

The use of fuzzy logic in the decision-making process is chosen because fuzzy logic allows the combination and evaluation of diverse parameters in an efficient manner. Fuzzy logic is a very promising approach because the sensor nodes easily support the requirements for execution. This improves the overall network performance.

The Data Processing client can work independently or in accordance with the Data Grapher. In our case sensed values are indexed according to the index marker placed on the chart in order to make easier recognizing the moment when the decision is made. As Fig. 14 shows, different colours marked the different conclusions of fire confidence. This presentation gives a detailed insight into the operation of the algorithm for decision-making. When a signal is detected by temperature sensors placed in the smart home, and decision about fire confidence is made based on predefined fuzzy logic rules, the alarm is activated and information for the user can be provided on the user interface.

#### 4.3. Summary

The example of system development for monitoring and determining the confidence of fire in a building presented in this paper shows the power of Raspberry Pi's usage in home automation. Hence, the developed Raspberry Pi prototype Sensor Web node is based on RESTful services and created in order to build the infrastructure that supports fast critical event signalling and remote access to sensor data via the Internet (the detection of critical events is performed by using fuzzy logic).

Both shown client methods for data processing are just one possible way of application. Use of the presented approach with Raspberry Pi as a Sensor Web node makes its applications endless.

As it can be observed, the recent advances in technology pave the way for the realisation of the “Internet of Things” (IoT). With the integration of intelligent objects and the Internet becoming a reality, an unprecedented exchange of information is enabled. The DIY category in home automation is more active today than ever before. The combination of inexpensive electronics with low-cost networked computers makes this option extremely attractive. In other words, use of inexpensive hardware, like proposed Raspberry Pi, and open source software, makes it possible to programmatically control many devices offering in such way solutions to meet user's own needs opposite to commercial solutions which are usually generalized to make products appeal to a large consumer base. The Raspberry Pi brings the advantages of a PC to the domain of a sensor network, contributing to its strength as a platform for interfacing with wide variety of external peripherals. Subsequently, the Raspberry Pi is good for any projects which require a graphic interface (display), and especially for any projects user wants to connect to the Internet. Thus, by coupling it with Wi-Fi and providing access to the Internet it is possible to set it up for a remote communication. In addition, Raspberry Pi can be used as a small energy efficient server that will host middleware, frontend and other more upper level software and aggregate all data for more complex systems. From all above mentioned it can be concluded the Raspberry Pi is very suitable for applications in IoT concept. Thus, the advantage of Raspberry Pi lies in its flexibility and extensive possibility of its usage, enabling at the same time end-users to program it according their needs and budgets.

Our approach differs from others presented in literature with regards to Java language used for programming and controlling, which leads to system modularity. Subsequently developed drivers written in Java provide multiple uses of written software in future projects. In other words, programming in high-level languages such as Java heighten the simplicity of solution implementation and accessibility to a wide market of users, as opposed to micro controller programming which usually depends on the development kit.

Alongside this the communication between sensors and the Raspberry Pi is performed via I<sup>2</sup>C interface whereas for the communication between Raspberry Pi and the clients RESTful service is used. As a result the Raspberry Pi is running in server mode and all graphical user interface and additional application software are unnecessary. This underlines the advantage of the model based on its low energy usage and resource consumption. In other words, such a developed system, based on the data collected by a Sensor Web, enable remote access to sensor data, their processing and visualization via Internet possible.

As leading advantage of proposed system, what makes this project different from others presented in literature, is the role it takes on as an appliance of artificial intelligence in a decision-making process. Fuzzy logic is chosen because it can tolerate unreliable and imprecise sensor readings, it is much closer to human way of thinking than crisp logic and compared to other classification algorithms based on probability theory, fuzzy logic is much more intuitive and easier to use. Thus, fuzzy logic is a very promising approach because the execution requirements can be easily supported by sensor nodes improving the overall network performance.

All these above stated choices are made in order to make the whole system a low cost solution to the home automation. As a result of this study, a web-based software and hardware application for smart home automation system has been realized. Security (fire detection) of the smart home is provided by means of remote control.

In other words, this paper presents the design and implementation of a low cost but yet flexible and secure Internet based home automation system, which allows user to create his/her own solution without need for advanced programming skills.

## 5. Conclusion

The application of the IoT technology, in the context of the home automation means integration of all electrical devices and their monitoring, controlling and alerting in ways not possible before. This works to provide many advantages including, safety, security, improved comfort, energy and cost savings.

In order to address the issues of flexibility and functionality, a novel, standalone, flexible and low cost home controlling and monitoring system using RESTful based Web services as an interoperable application layer for communicating between the remote user and the home devices, have been designed and implemented. Performed research and experimental results have shown that by using the Raspberry Pi and open source software it is possible to programmatically control many devices in a building in such a way that user can create his/her own solution customized to meet his/her individual needs. Thus, the proposed system is better from the scalability and flexibility point of view than the commercially available home automation systems.

It can be concluded that the discussed system design and architecture, and proposed prototype present the basic level of home appliance control and remote monitoring while the required goals and objectives of home automation system have been achieved. Results have shown that the Raspberry Pi is an inexpensive computer with a lot of potential, which enables its utilization, not just in home automation project, but also in a diverse range of research applications.

Directions of future work will be to move Raspberry Pi usage to outdoor, such as the monitoring of temperature and humidity, forest fire, radiation etc. For that purpose Raspberry Pi will be expanded with GSM/GPRS module and independent power supply (battery or solar cells). By testing it in such environment the possibility of usage of Raspberry Pi in monitoring natural phenomena will be shown.

## References

- [1] Miorandi D, Sicari S, De Pellegrini F, Chlamtac I. Internet of things: vision, applications and research challenges. *Ad Hoc Netw* 2012;10:1497–516.
- [2] Dennis AK. *Raspberry Pi home automation with Arduino*. USA: Packt Publishing; 2013.
- [3] Dupré J. Integrating fire and security systems with home automation; 2010 <<http://hiddenwires.co.uk/resourcesarticles2010/articles20100705-02.html>> [25.12.13].
- [4] Kurkinen L. Smart homes and home automation, Berg Insite M2M Research Series; <<http://www.berginsight.com/ReportPDF/ProductSheet/bi-sh2-ps.pdf>> [16.01.14].
- [5] Ashi, Home automation system. Reporter, vol. 19, no. 2, February 2002.
- [6] Vujović V, Maksimović M. Raspberry Pi as a wireless sensor node: performances and constraints. The 37th International ICT Convention – MIPRO 2014, ISSN 1847-3938, ISBN 978-953-233-078-6, Opatia, Croatia 2014. p. 1247–52.
- [7] Maksimović M, Vujović V, Davidović N, Milošević V, Perišić B. Raspberry Pi as Internet of things hardware: performances and constraints, IcETRAN; 2014, Vrnjačka banja, Serbia.
- [8] Schmidt M. *Raspberry Pi – a quick start guide*. The Pragmatic Bookshelf; 2013.
- [9] Home automation. <<http://www.leviton.com>> [07.01.14].
- [10] Monk S. *Raspberry Pi Cookbook*. CA: O'Reilly Media; 2014.
- [11] Horan B. *Practical Raspberry Pi*. USA: Apress; 2013.
- [12] Upton E, Halfacree G. *Raspberry Pi user guide*. Wiley; 2012.
- [13] Richardson M, Wallace S. *Getting started with Raspberry Pie*. USA: O'Reilly; 2013.
- [14] Heath N. 10 coolest uses for the Raspberry Pi, European Technology; 2012 <<http://www.techrepublic.com/blog/european-technology/10-coolest-uses-for-the-raspberry-pi/505/>> [21.12.13].
- [15] Introduction to home automation with the Raspberry Pi. <<https://vancouver.hackspace.ca/wp/2013/11/03/introduction-to-home-automation-with-the-raspberry-pi-sunday-november-24th-2013/>> [21.12.13].
- [16] Home monitoring projects for Raspberry Pi. <<http://www.projects.privateeyepi.com/home>> [22.12.13].
- [17] Bell CA. Beginning sensor networks with Arduino and Raspberry Pi. Apress Media; 2013.
- [18] Warner TL. *Hacking Raspberry Pi*. USA: Que Publishing; 2013.
- [19] Goodwin S. *Smart Home Automation with Linux and Raspberry Pi*, 2nd ed., Apress Media; 2013.
- [20] Electronic project – Simple home automation with the Raspberry Pi. <<http://www.penguintutor.com/electronics/simple-homeautomation-raspberrypi>> [22.12.13].
- [21] Jersey 2.2 User Guide, <<https://jersey.java.net/documentation/latest/index.html>> [09.09.13].
- [22] Yang Q. Temperature sensors; 2011 <[http://www.eng.hmc.edu/NewE80/PDFs/TemperatureSensors\\_QYang2011.pdf](http://www.eng.hmc.edu/NewE80/PDFs/TemperatureSensors_QYang2011.pdf)> [28.12.13].
- [23] Recktenwald G. Temperature measurement with a Thermistor and an Arduino; 2013 <<http://web.cecs.pdx.edu/~eas199/B/howto/thermistorArduino/thermistorArduino.pdf>> [02.11.13].
- [24] Steinhart JS, Hart SR. Calibration curves for thermistors. *Deep Sea Res Oceanogr* 1968;15(4):497–503.
- [25] Pop B, Medeşan R, Jurca I. Performance comparison of Java application servers, Buletinul Stiintific al Universitatii Politehnica din Timisoara, Seria Automatica si Calculatoare Transactions on Automatic Control and Computer Science. <<http://citeserx.ist.psu.edu/viewdoc/download?doi=10.1.1.7.4033&rep=rep1&type=pdf>> [09.09.13].
- [26] Web Servers survey. <<http://isea.nitk.ac.in/PMISprojects/reports/WebserversReport.pdf>> [20.01.14].
- [27] Gomes MC, Paulino H, Baptista A, Araújo F. Dynamic interaction models for web enabled wireless sensor networks, parallel and distributed processing with applications (ISPA); 2012.
- [28] Burke B. RESTful Java with JAX-RS, first ed., O'Reilly Novembar 2009.
- [29] Vujović V, Maksimović M, Kosmajac D, Milošević V, Perišić B. Web integration of REST enabled wireless sensor networks for fire detection, International Conference on Applied Internet and Information Technologies, Zrenjanin, Serbia; 2013. p. 30–5.
- [30] Maksimović M, Vujović V, Perišić B, Milošević V. Developing a fuzzy logic based system for monitoring and early detection of residential fire based on thermistor sensors, *Comput Sci Inform Syst* 12(1):63–89 ISSN 1820-0214 (Print) 2406–1018 (Online), 2015, doi: <<http://dx.doi.org/10.2299/CSIS140330090M>>.

**Vladimir Vujović** works at the Department of Computer science and informatics, Faculty of Electrical Engineering in East Sarajevo, Bosnia and Herzegovina. As a Ph.D student at the Faculty of Technical Sciences in Novi Sad, Serbia, he is pursuing research in the field of Software engineering, Information systems and Artificial intelligence (Model Driven Software Development, Information System Design, Home Automation, Sensor Networks, etc.)

**Mirjana Maksimović** received the M.Sc. and Ph.D degree in Electrical Engineering from University of East Sarajevo, Bosnia and Herzegovina. She works at the Departments of Automatics and Electronics, Electroenergetics and Computer science and informatics, Faculty of Electrical Engineering in East Sarajevo. Her current research and teaching interests extend to a range of topics in Telecommunications, Automation and Electronics.