

Case Study: IoT Data Integration for Higher Education Institution

Krišjānis Pinka¹, Jānis Kampars², Vladislavs Minkevičs³
^{1–3} Riga Technical University

Abstract – Nowadays the Internet of Things (IoT) is one of the most trending technologies. It is expected that by year 2020 there will be 50 billion of Internet-connected devices. Fields like smart cities and smart homes largely rely on IoT phenomena by using a wide variety of sensors for data collection, analysis and corresponding actions. The paper describes how this trending and relatively new technology is applied at Riga Technical University for educational purposes.

Keywords – Internet of Things, data integration, cloud computing.

I. INTRODUCTION

The idea of the Internet of Things (IoT) suggests that rather than having a small number of powerful computing devices (e.g. laptop, tablet) you have a large number of devices, which are less powerful (e.g., bracelet, connected light bulb).

Name itself implies that the IoT consists of two key parts – network and various devices connected to it. The “Internet” part could take various forms and serves as the medium for sending and receiving the data, while the devices (referred to as “Things”) act as data sources and actuators.

Explosive growth of smartphones and tablet PCs brought the number of Internet-connected devices per person to more than 1 device in year 2010 for the first time in history [1]. In year 2015 the total number of connected devices reached 18.2 billion [2]. Cisco predicts that there will be a total of 50 billion Internet-connected devices by year 2020.

The IoT devices are present physically in the real world, in your home, work, and car or worn around your body. This means that they receive inputs from the physical world, transform them into data and produce outputs that can be collected and processed for further action, e.g., processing and analysis [3]. The sensor originating data are turned into information, knowledge, and, ultimately, wisdom. In this context, the IoT becomes immensely important in enriching our daily experience.

Even now the IoT has already made the Internet sensory with real-time data about measurable properties, such as temperature, pressure, vibration, light and moisture. This serves as a driving force for creating cyber-physical systems that have the potential of dramatically improving the way people live, learn, work and entertain themselves. In addition, the Internet is expanding into places that until now have been unreachable. Patients are ingesting IoT devices into their own bodies to help doctors diagnose and determine the causes of certain diseases. Extremely small Internet-connected sensors can be placed on plants, animals, geologic features and in buildings [1]. Those sensors provide data that give real-time insights and can also be

used for predictive analysis. Lorence Heikell [4] has documented an example from a dairy farm where IoT based monitoring is used for improving milk production, smoothing calving process and ensuring good health of the cows. The German farmer has acknowledged that cow farming has become much easier thanks to the applied IoT solution. The time for evaluating the health of the animals has gone down from 2–3 hours per day to a few minutes that can be spent while looking at IoT based reports in a mobile application. Another field of application is the building monitoring systems that present data to experts in real-time allowing them to evaluate the current state of a building, predicting possible damage or even collapsing risks. Experiments documented in [5] show that such IoT based building monitoring systems perform well and meet experts’ requirements.

IoT solutions can be composed of a large number of devices that should be provided with effective ways of data exchange. Maximising the inter-device communication distance and minimising the power consumption are an important challenge in the area of IoT, and new technologies are emerging to address it [6]. One of such technologies is the Wireless Sensor Network (WSN). It is an architectural model containing a number of sensor nodes and a central node (hub). In the WSN, the collected data from the sensor nodes are transferred to the central node for processing and further actions. One of the communication protocols that can be applied for IoT node communication is the Bluetooth low energy (BLE) wireless communication protocol. Another option providing low energy and relatively wide communication range is the Zigbee wireless communication standard. Zigbee supports creation of IoT networks with up to 64 000 devices [7]. The difference between BLE and Zigbee lies in the frequency, channel bandwidth, number of cell nodes and supported range.

The objective of this paper is to present the IoT data integration solution that was developed at Riga Technical University (RTU) for demonstrating the IoT capabilities to the new generation of IT professionals. The first version of our solution is ready for deployment and we are planning to engage our students in extending it. To make it more appealing for the students we have integrated our IoT solution with social media. Design science is chosen as the research method for this paper.

The organisation of this paper is as follows. Section II presents a literature review of related studies. Section III describes the IoT data integration solution developed at RTU. Section IV concludes with final remarks and a summary of our plans for expanding the current IoT solution.

II. RELATED RESEARCH

When IoT devices are connected to a network, they work together in cooperation to achieve a common goal – to provide a specific service. This can be beneficial for many real-life applications and services like smart-homes where some devices collect data (sensors), while others (actuators) use it for counteraction triggering.

The IoT can be applied to create extremely large-scale solutions like smart cities consisting of thousands of sensors and actuators [8]. In smart cities, administration and inhabitants can get access to the real-time data about the surrounding environment. This can then be transformed into information and knowledge used for making better decisions on a daily basis and urban planning [9].

Concepts like smart homes, which are more private IoT networks at a smaller scale, are gaining popularity because sensors and boards (microcontrollers and microcomputers) are available at low prices and a sufficient number of educational materials are freely available on the Internet. Smart homes can be defined as home automation systems consisting of various sensors and actuators used for improving the comfort for residents. BLE, Wi-Fi and RFID are typically used as the communication technologies. Sensors such as temperature, light, power meter are installed and connected wirelessly to a board following the WSN architectural concept. Usually the board is also connected to the Internet making data available to ones with appropriate access level. Smart homes are often complemented with a mobile application, thus taking the advantage of modern smartphone capabilities and delegating some of the control functions to the resident's smartphone. In this way, the monitoring data can be easily accessed from any place with an Internet connection [10].

Integrating the IoT with social media brings more possibilities in information sharing and reaching larger audiences. This can be specifically beneficial for smart-cities and other scenarios with a large user base. In this concept, the sensor data are sent to a central node for analysis and after that it is made available to the general public through the Internet and various social networks. Results of the research by Lee et al. [11] show that the combination of IoT and social network services not only helps with introduction of IoT products to general public but also brings a potential business value. Sensor information from the surrounding area can be correlated with user activities in social networks. Using this kind of information, it is easier to predict the places that a user might like and also the contextual conditions (e.g., high temperature, no rain) [12], [13].

The concept that is most related to this research is smart universities. The data are made publicly available and gathered by a wide range of various sensors. Typically these are:

- environment related sensors measuring properties, such as temperature, light, air quality and humidity;
- security sensors detecting motion, opening and closing doors or windows;
- safety sensors detecting smoke or other gases.

The architectural model described in [14] shows that all these devices are connected to a hardware board (Arduino, Raspberry Pi, Banana Pi) that wirelessly transmits data to a central node, which is another hardware board. Similar to smart homes, this also follows the WSN architectural principle. The received data are sent to a web server and database server that are deployed locally or in the cloud. Cloud provides the flexibility of scaling the IoT database and computing nodes according to the load of the system and provides a good quality of service with a minimal number of resources [15].

The architect of the IoT solution can choose different platforms for end-users. The most common ones are smartphones and tablets; however, these are not the only options. Smart TVs can also be used as devices for displaying the sensor originating information in real time [16]. This is particularly suitable for smart university use cases. The screen of a Smart TV performs well, presenting data for a large number of people at once. High resolution allows showing various graphical charts and data, which would be harder to see on a small screen of a smartphone. If the TV is mounted in a social place, it will certainly attract people and popularise the IoT technology.

III. IOT DATA INTEGRATION SOLUTION AT RTU

This section looks at the technology, methods and devices that were used to implement the IoT data integration solution at RTU. Sensors are used for data collection and are connected to the boards, which integrate and transmit the data to a central cloud-deployed node containing a database and REST (Representational State Transfer) endpoints. Smart TV is used for visualising data, and Twitter integration provides ways for interacting with the IoT solution.

A. Architecture

Various technologies and their combinations can be used to implement an IoT project and very similar results can be achieved using different boards and sensors. The devices used in the research were chosen based on their availability in Latvia's retail stores and popularity among developers. The objective of our solution is to educate the students and present the possibilities of IoT. Social media (Twitter) and Smart TV are used for making the solution interactive. The list of the used devices and general architecture of the solution is given in Fig. 1.

The central node of the IoT solution is deployed to RTU's cloud computing platform, thus providing persistent storage and virtually unlimited computing power. Currently, it is a virtual machine with random access memory (RAM) 1 GB and 1 x 2.39 GHz central processing unit (CPU). These resources are sufficient for running a web server and a database server, which are needed for aggregating the sensor data and transmitting them to Smart TV. If more resources are needed, database and web-server can be deployed to separate nodes with larger memory and CPU capacity. The first release of IoT solution has Apache 2 used as the web-server, while MySQL performs the role of the relation-database. Web server uses PHP server-side scripting language for processing REST

web-service requests. JSON (JavaScript Object Notation) is used as the data format for transmitting data between devices. Microcomputer Raspberry Pi and microcontroller Arduino are connected to each other with a universal serial bus (USB) cable, and Arduino is used for processing dust and CO sensor (analogue) data, while Raspberry Pi gathers (digital) data from other sensors and transmits them over Wi-Fi to the previously mentioned cloud node.

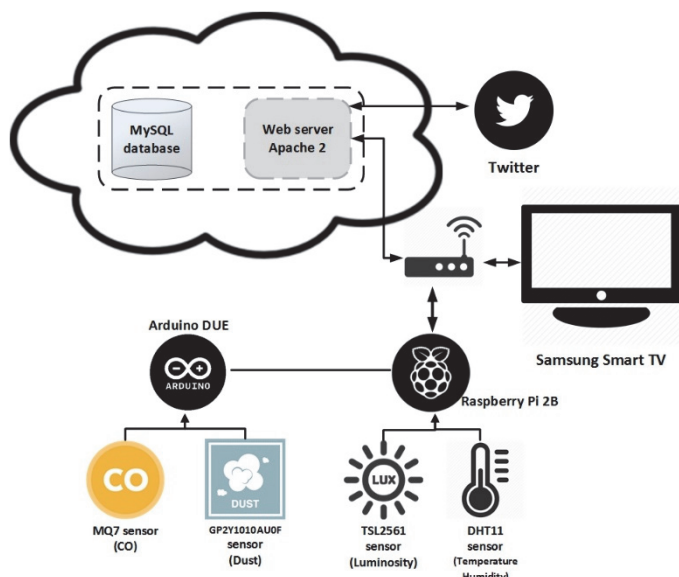


Fig. 1. The IoT Architecture used for the case study.

Both of the hardware boards are currently located at the premises of the Institute of Information Technology, RTU; however, they can be moved to any other location with a Wi-Fi connection. The devices can be powered from a standard USB cable or a portable power bank (generally used for charging mobile phones while being on the move), so it is possible to collect data in places with no power sources.

The Arduino program is written in C programming language. It processes measurements from CO and dust sensors and transfers the results to Raspberry Pi, which receives these data and then collects measurements from temperature, humidity and light sensors. A Python based program is used for implementing the data processing and transmission logic. If connection cannot be established due to network problems, the data collection continues and another connection attempt is performed later.

Samsung Smart TV is used for data visualisation and allows users to interact with the IoT data integration solution using the remote control of Smart TV. TV is connected to the Internet through Wi-Fi and is supplied with near real-time sensor data updates from the cloud-based node. Twitter integration provides additional options for users and makes our solution available from one of the leading social networks.

By tweeting a specific word in the text (e.g., HumidityDay #RTU-IoT), users can change the data to be displayed on the Smart TV screen. The cloud-based node is continuously scanning tweets under specific hashtag and searching for commands. If a syntactically valid command is detected, the

central cloud-based node contacts the Smart TV and changes the data being displayed.

The approximate budget of our solution is 750 euro where the Smart TV takes around 80 % of expenses. The cost can be significantly reduced by using a cheaper TV. Components that were used to implement the IoT solution are further described in the following sections.

B. Sensors

DHT11 sensor is used for temperature and humidity measurements, TSL2561 sensor provides luminosity measurements, GP2Y1010AU0F sensor serves as an optical dust sensor and MQ-7 measures carbon monoxide level. Figure 2 illustrates the wiring to hardware boards for all the used sensors.

DHT11 Temperature & Humidity Sensor has a digital output function and uses exclusive techniques for measuring physical conditions and converting the resulting samples into digital numeric values. High reliability and long-term stability are ensured by calibration of each DHT11 element in the laboratory environment. The single-wire serial interface makes system integration quick and easy. The small size of the sensor ensures low power consumption and long distance signal transmission. Sensor measurement range for humidity is 20–90 % RH (accuracy ± 5 % RH) and for temperature 0–50 °C (accuracy ± 2 °C). The required power supply is 3–5.5 V DC [17].

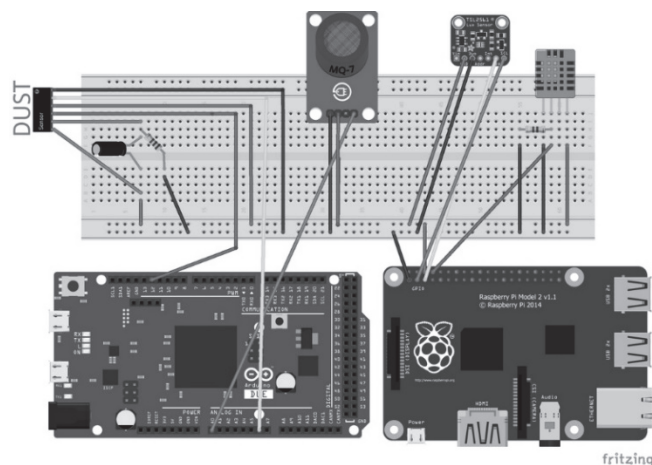


Fig. 2. Sensor wiring to hardware boards.

The TSL2561 luminosity sensor is an advanced digital light sensor, which can be used for measuring a wide range of light situations. Its accuracy allows performing exact Lux calculations, and it can be configured for different gain and timing ranges to detect light ranges from 0.1 to 40,000+ Lux. One of its advantages is that it contains both infrared and full spectrum diodes allowing for separate measurement of infrared, full-spectrum or human visible light. The sensor has a digital inter-integrated circuit (i2c) interface and built-in analogue-to-digital converter (ADC), i.e., it can be used with a wide range of boards (even ones having no analogue inputs like Raspberry Pi). The current draw is extremely low making it power-effective [18].

Sharp's GP2Y1010AU0F is an optical air quality sensor designed to sense dust particles. An infrared emitting diode and a phototransistor are diagonally arranged allowing one to detect the reflected light of dust in the air. It is especially effective in detecting very fine particles and is commonly used in air purifier systems. The sensor has a very low current consumption and can be powered with up to 7 VDC. The output of the sensor is analogue voltage proportional to the measured dust density, with a sensitivity of 0.5 V / 0.1 mg/m³. Due to the analogue output, this sensor can be used only with boards that have analogue inputs (e.g., Arduino) [19].

MQ-7 Carbon Monoxide (CO) sensor is suitable for sensing CO concentrations in the air. It can detect the highly poisonous CO-gas concentrations in parts-per million (ppm). This sensor has a high sensitivity and fast response time. The sensor output is analogue resistance, so it can be used only together with boards that have analogue inputs [20].

C. Boards

Data collection is provided by two boards: Arduino Due and Raspberry Pi 2B. The Raspberry Pi is a single-board computer created by the Raspberry Pi Foundation. It is based on the Broadcom BCM2836, which is a multimedia application processor geared towards mobile and embedded devices. The usage of ARMv7 processor allows Raspberry Pi 2B to run full range of advanced RISC machine (ARM) GNU/Linux distributions, including Snappy Ubuntu Core, as well as Microsoft Windows 10. It also contains several other components to support USB, RCA connections and SD cards as a means of persistent data storage [21].

The key features of Raspberry Pi 2 are:

- a 900 MHz quad-core ARM Cortex-A7 CPU;
- 1 GB RAM;
- 4 USB ports;
- 40 GPIO pins;
- full HDMI port;
- Ethernet port;
- combined 3.5mm audio jack and composite video;
- camera interface (CSI);
- display interface (DSI);
- Micro SD card slot;
- Video Core IV 3D graphics core.

An excerpt from Raspberry's Python based sensor data collection program is given below:

```
if __name__ == '__main__':
    try:
        instance = dht11.DHT11(pin = 14)
        while True:
            dht11()
            tsl = TSL2561(debug=True)
            lux = tsl.lux()
            CO_DUST()
            mysql()
            time.sleep(10)
    except error:
        GPIO.cleanup()
        destroy()
```

```
def dht11():
    result = instance.read()
    if result.is_valid():
        global temp, hum
        temp = result.temperature
        hum = result.humidity
```

The code shows that dht11 sensor (temperature and humidity) is connected to pin number 14. The data are read and if the validity test has been passed, the values of global variables temp and hum are updated. The main program transmits the data to the cloud-based server once in every 10 seconds.

Arduino is an open source single-board microcontroller hardware [22] that is coupled with a programming language and an Integrated Development Environment (IDE) [21]. It is based on the Atmel AVR processor and provides many inputs and outputs in only one self-sufficient piece of hardware. Arduino has become one of the most popular and successful examples of bringing the open source concept to the hardware world [22]. In this project we use the Atmel SAM3X8E ARM Cortex-M3 CPU based Arduino Due microcontroller board. It is the first Arduino board based on a 32-bit ARM core microcontroller. The key features of Arduino Due are:

- 32-bit core;
- 84 MHz clock speed;
- 512 KB of Flash memory for code;
- 54 digital input/output pins;
- 12 analogue inputs;
- 4 universal asynchronous receiver/transmitter (UARTs) hardware serial ports;
- USB OTG capable connection;
- 2 digital to analogue (DAC) pins.

The Arduino Due is compatible with all Arduino shields that work at 3.3V and are compliant with the 1.0 Arduino pinout [23].

An excerpt from Arduino's C based sensor data collection program is given below:

```
int measurePin = 6;
float voMeasured = 0;
float calcVoltage = 0;
float dustDensity = 0;

void loop(){
    Serial.print(MQGetGasPercentage(MQRead(MQ_PIN)/Ro,
    GAS_CO));
    voMeasured = analogRead(measurePin);
    calcVoltage = voMeasured * (3.3 / 1024);
    dustDensity = (0.17 * calcVoltage - 0.1) * 100;
    Serial.print(' ');
    Serial.println(dustDensity);
    delay(1000);
}
```

The code begins with defining variables for holding the values from a dust sensor that has been wired to pin 6. It is followed with a loop that takes CO and dust measurements once per second. At the end of the loop, the values are sent to the Raspberry Pi 2 via the serial connection.

Samsung J5500 5 series Flat Full HD Smart LED TV is used for presenting the data in a graphical format. Samsung J5500 5 series Samsung TV has built-in Wi-Fi that allows interacting with the cloud-deployed node. The data retrieval and presentation are implemented as a web-based application running on the application engine of the Smart TV. The developed application is basically a web page consisting of Hypertext Markup Language (HTML), Cascading Style Sheets (CSS) and JavaScript.

D. Cloud-Based Server

The server node is used for receiving, processing and storing sensor data and communicating with the Smart TV application. The board program sends a request for inserting sensor data in the database, while TV application is polling the aggregated sensor data for updating the user interface with current and historical sensor data measurements.

The cloud-based server is deployed to the Apache CloudStack open-source cloud computing platform. Ubuntu 14.04 LTS was chosen as the operating system. It is built on the foundation of Linux, which is a member of the UNIX family. UNIX is one of the oldest types of operating systems, and together with Linux has provided reliability and security for professional applications for almost half a century. Many servers around the world storing data for popular websites (such as YouTube and Google) run some variant of Linux or Unix [24].

Apache 2.4 was chosen as the web server software. It is open-source software that was developed in 1995. It is available for a variety of operating systems and is hosting half of world's active websites and, therefore, it is the most widely used web server of today [25]. In case of larger number of sensors and presentation devices, Apache can be substituted with alternatives providing a higher number of concurrent connections and better performance like NodeJs.

MySQL server 5.5 is used as the relation database management system. The advantages of MySQL are reliability, high-performance, easy installation and use, good support in a wide range of programming languages. Similarly to Apache, it can be substituted with other alternatives in case of a higher number of read/write operations. NoSQL database systems and specifically MongoDB would be a suitable option in such a scenario.

E. Smart TV Program

The Smart TV program is written in JavaScript and provides good performance and a wide range of functionality to front-end users. The program makes an HTTP request to the cloud-based node in order to get the sensor data, which are retrieved from the database. The data are then returned in JSON format and visualised by the Smart TV. The user-interface of the program is shown in Fig. 3.

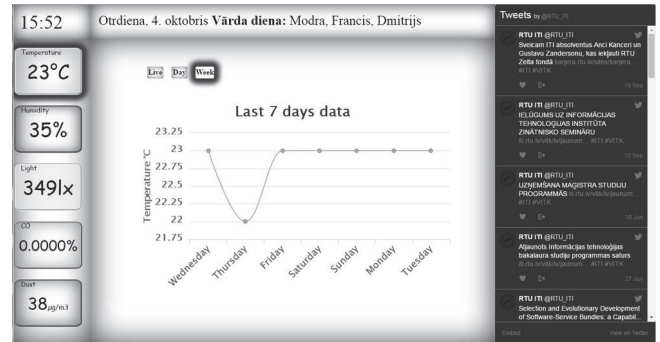


Fig. 3. Smart TV user interface.

The source code of the JavaScript function pulling data from the cloud-based server is given below:

```
function getData(){
$.ajax({
url: "http://iot-server.rtu.lv/rest",
type: "POST",
dataType: 'json',
success: function (data) {
Temp = data.Temp;
Hum = data.Hum;
Lux = data.Lux;
CO = data.CO;
Dust = data.DUST;
},
error: function(jqXHR,error, errorThrown) {
logData(jqXHR);
setTimeout(getData,3000);
}
});
}
```

The function retrieves the current temperature, humidity, light, CO and dust measurements and saves them into the local variables. In case of a communication error, the details are logged and request is repeated in 3 seconds.

I. Security of the IoT Solution

According to the latest news about Botnet made of IoT devices, such as CCTV cameras and DVR system attacks, the security measures should be in place such as regular security updates. The operating system should be hardened and unneeded services should be switched off [26]. The current security of the system includes physical protection of devices by putting them in RTU premises under video surveillance. This way people without the needed permission are unable to physically access the devices, and the risk of damaging the IoT solution is minimised.

Authentication and access lists are used for securing REST web-service endpoints. The same applies to Raspberry Pi and Arduino, where authentication is required for changing device settings and main program.

Smart TV is connected to Wi-Fi through Wi-Fi Protected Access version 2 (WPA2) security protocol allowing only for authorized use of the wireless network and safe data transmission.

IV. CONCLUSION

The IoT data integration solution is ready for deployment and is planned to be installed by the end of year 2016. Currently our solution can perform measurements only in a single location since it uses a single set of boards and sensors; however, this can be easily extended by adding more components. The unique feature of our IoT data integration solutions is the integration with Twitter, which is planned to be extended with support for other social media networks, such as Instagram. Student feedback will be gathered serving as input for specifying additional requirements for future improvements. The current data integration solution is only at its early stages and supports data gathering from a single location; therefore, work on further development will be related to expansion of system capabilities.

We are planning to extend the Twitter integration with additional commands that can be used in tweets. For example, users will be able to find out the data about real time conditions at the university by just tweeting a syntactically valid command (e.g., @rtu-iot get temperature) to the IoT solution's Twitter account and getting the response in form of a tweet. This would provide easy access to the IoT data from tablets, smart phones and desktop PCs.

The existing range of sensors can be extended with a sound level detection sensor. Increasing the number of devices will allow placing them in several rooms in such way gathering data from a larger area. Our IoT solution will also be supplemented with actuators performing various actions under certain context conditions. For example, the conditioner in one of our labs could be powered by a signal from an infrared (IR) transmitter in case of temperature going above a predefined margin.

We are also planning to add a Wi-Fi adapter in the monitoring mode that will be able to collect all broadcasted SSIDs (Service Set Identifiers) that mobile phones or laptops are broadcasting while trying to connect to Wi-Fi access points that they have already been connected to previously. The result will be displayed on Smart TV raising the awareness that our devices are broadcasting excessive information. Bluetooth adapter can also be used to collect information about mobile phones or laptops that have their Bluetooth enabled.

Another potential function of the IoT data integration solution is people counting using cameras and image processing programs that are deployed to the cloud. However, in this case the privacy aspects should be considered based on EU Regulation on the protection of natural persons with regard to the processing of personal data and on the free movement of such data.

Currently only Wi-Fi is used as a means of communication; however, we are planning to experiment with other more perspective communication technologies. In next versions of our solution, the connected sensors could be based on AVR MCU and sensors could send information to the board wirelessly. In this case, we also have to think about security issues, for example, whether the information will be sent in plaintext or encrypted form.

After setting up the number of sensors and monitored rooms, we are planning to display a live map of the sensor locations on the Smart TV. We are also planning to install customised

Bluetooth beacons for demonstrating the in-room positioning capabilities and tracking the movement of users, who have established a Bluetooth connection with our beacons.

REFERENCES

- [1] D. Evans, *The Internet of Things. How the next evolution of the Internet is changing everything*. Cisco Internet Business Solutions Group, 2011.
- [2] "Internet of Things (IoT): number of connected devices worldwide from 2012 to 2020 (in billions)," The Statistics Portal, August 2015. [Online]. Available: <https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/> [Accessed: Sept. 4, 2016].
- [3] A. McEwen and H. Cassimally, *Designing the Internet of things*. Chichester, UK: John Wiley and Sons, 2014, pp. 10–11.
- [4] L. Heikell, "Connected cows help farms keep up with the herd," [Online]. Available: <https://news.microsoft.com/features/connected-cows-help-farms-keep-up-with-the-herd/#sm.001f8tey8169bdtwt31clgcb586g/> [Accessed: Sept. 28, 2016].
- [5] W. Sun, X. Chen, F. Liu and L. Zhao, "Collection System for Pressure Information of Buildings Based on the IoT and CAN Bus," in *2012 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discover, CyberC2012*, Sanya, China, 2012, pp. 274–277. <https://doi.org/10.1109/cyberc.2012.53>
- [6] K. Nair, J. Kulkarni, M. Warde, Z. Dave, V. Rawalgaonkar, G. Gore and J. Joshi, "Optimizing Power Consumption in IoT based Wireless Sensor Networks using Bluetooth Low Energy," in *2015 International Conference on Green Computing and Internet of Things, ICGCIoT*, Greater Noida, Delhi, India, 2015, pp. 589–593. <https://doi.org/10.1109/icgciot.2015.7380533>
- [7] E. D. N. Ndih and S. Cherkouli. "On Enhancing Technology Coexistence in the IoT Era: ZigBee and 802.11 Case," *IEEE Access*, vol. 4, pp. 1835–1844, 2016. <https://doi.org/10.1109/ACCESS.2016.2553150>
- [8] T. Yashiro, S. Kobavashi and N. Koshizuka, "An Internet of Things (IoT) architecture for embedded appliances," in *Humanitarian Technology Conference, R10-HTC*, Sendai, Japan, 2013, pp. 314–319. <https://doi.org/10.1109/r10-htc.2013.6669062>
- [9] J. Jin, J. Gubbi, S. Marusic and M. Palaniswami, "An Information Framework for Creating a Smart City Through Internet of Things," *IEEE Internet of Things journal*, vol. 1, no. 2, pp. 112–121, Apr. 2014. <https://doi.org/10.1109/JIOT.2013.2296516>
- [10] R. Piyare and S.R. Lee, "Smart Home-Control and Monitoring System Using Smart Phone," in *1st International Conference on Convergence and its Application, ICCA*, vol. 24, 2013, pp. 83–86.
- [11] H. Lee and J. Kwon, "Survey and Analysis of Information Sharing in Social IoT," in *2015 8th International Conference on Disaster Recovery and Business Continuity, DRBC*, Jeju Island, Korea, 2015, pp. 15–18. <https://doi.org/10.1109/drbc.2015.13>
- [12] Y. Lu, J. Zhou and F.Hao, "The implementation method and value analysis of IoT based social network," in *2013 3rd International Conference Consumer Electronics, Communications and Networks, CECNet*, Xianning, China, 2013, pp. 680–683. <https://doi.org/10.1109/cecnet.2013.6703422>
- [13] B. Guo, Z. Yu, X. Zhou and D. Zhang, "Opportunistic IoT: Exploring the Social Side of the Internet of Things," in *2012 IEEE 16th International Conference on Computer Supported Cooperative Work in Design*, Wuhan, China, 2012, pp. 925–929. <https://doi.org/10.1109/cscwd.2012.6221932>
- [14] M. Cata, "Smart university, a new concept in the Internet of Things," in *2015 14th RoEduNet International Conference – Networking in Education and Research, RoEduNet NER*, Craiova, Romania, 2015, pp. 195–197. <https://doi.org/10.1109/roedunet.2015.7311993>
- [15] A. Botta, W.D. Donato, V. Persico and A. Pescape, "Integration of Cloud computing and Internet of Things: A survey," *Future Generation Computer Systems*, vol. 56, pp. 684–700, March 2016. <https://doi.org/10.1016/j.future.2015.09.021>
- [16] M. Yusofov and I. Kornilov, "Roles of Smart TV in IoT-environments: a Survey," in *Proceeding of the 13th conference of Fruct association*, Petrozavodsk, Russia, 2013, pp. 163–168.
- [17] "DHT11 Humidity & Temperature Sensor," D-Robotics UK. [Online]. Available: <http://www.micropik.com/PDF/dht11.pdf> [Accessed: Sept. 15, 2016].
- [18] Lady ada, "TSL2561 Luminosity Sensor," Adafruit Industries, Sept. 8, 2016. [Online] Available: <https://cdn-learn.adafruit.com/downloads/pdf/tsl2561.pdf> [Accessed: Sept. 15, 2016].

- [19] "Optical Dust Sensor – GP2Y1010AU0F," Feb. 2, 2010. [Online] Available: <https://www.sparkfun.com/products/9689> [Accessed: Sept. 16, 2016].
- [20] "Carbon Monoxide Sensor - MQ-7," Sept. 30, 2009. [Online]. Available: <https://www.sparkfun.com/products/9403> [Accessed: Sept. 16, 2016].
- [21] A. K. Dennis, *Raspberry Pi Home Automation with Arduino*. Birmingham, UK: Packt Publishing, 2013, pp. 9–10.
- [22] J. Bayle, *C Programming for Arduino*. Birmingham, Birmingham, UK: Packt Publishing, 2013, pp. 7–10.
- [23] "Arduino Due," Arduino. [Online]. Available: <https://www.arduino.cc/en/Main/ArduinoBoardDue> [Accessed: Sept. 10, 2016].
- [24] "Getting Started with Ubuntu 14.04 – Second edition," The Ubuntu Manual Team. [Online]. Available: http://files.ubuntu-manual.org/manuals/getting-started-with-ubuntu/16.04/en_US/screen/Getting%20Started%20with%20Ubuntu%2016.04.pdf [Accessed Sept. 15, 2016].
- [25] "Apache HTTP Server," Sept. 16, 2016. [Online]. Available: https://en.wikipedia.org/wiki/Apache_HTTP_Server [Accessed: Sept. 16, 2016].
- [26] P. Paganini, "150,000 IoT Devices behind the 1Tbps DDoS attack on OVH," Sept. 2016. [Online]. Available: <http://securityaffairs.co/wordpress/51726/cyber-crime/ovh-hit-botnet-iot.html> [Accessed: Sept. 29, 2016].

Krišjānis Pinka has earned his B. sc. ing. (in 2015) degree in Information Technology from Riga Technical University, Latvia. He has been working at the Faculty of Computer Science and Information Technology, Riga Technical University since 2015. The current positions are Junior System Administrator and Junior Researcher. He has participated in Cisco two-day IoE Creathon for social good. His areas of interest are IoT technologies and cloud computing.
E-mail: Krisjanis.Pinka@rtu.lv

Jānis Kampars obtained his Doctoral degree in Information Technology from Riga Technical University in 2012. His areas of interest are application integration, data integration, web development and cloud computing. Jānis Kampars works as an Assistant Professor and Researcher at the Faculty of Computer Science and Information Technology, RTU, Latvia. He has been working at RTU since 2004. Recently he has participated in FP7 project Capability as a Service. Jānis Kampars also delivers study courses on web programming, data analysis and processing, cloud computing, business system programming at RTU.
E-mail: Janis.Kampars@rtu.lv

Vladislavs Minkevičs works as a Chief IT Security Officer at Riga Technical University. His areas of interest are IT security, microcontrollers, and script programming for IT security implementation purposes. Vladislavs Minkevičs has been working as a Lecturer at the Faculty of Computer Science and Information Technology, RTU for more than 10 years and delivers study courses on IT security related material. Vladislavs Minkevičs has more than 15 years of experience in IT security area and has obtained CISA and CISSP certification.
E-mail: Vladislavs.Minkevics@rtu.lv